

Base de données IDS/II

Manuel de référence IDS/II

Logiciel

Sujet : Description complète de la fonction de base de donnée IDS/II.

Observations :

Version du logiciel : GCOS7-LS version V1
GCOS7-MS Version V1
GCOS7-ES Version V1

Date : Septembre 1985

Bull Electronics Angers S.A.
CEDOC
Atelier de reprographie
331, Avenue Patton
49004 ANGERS Cedex 01
FRANCE

Bull HN Information Systems Inc.
Publication Order Entry
FAX: (508) 294-7411
MA02/423S
Technology Park
Billerica, MA 01821
U.S.A.

Copyright © Bull S.A., 1985

Toutes les marques citées sont la propriété de leurs titulaires respectifs.

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin du présent manuel.

La loi du 11 mars 1957, complétée par la loi du 3 juillet 1985, interdit les copies ou reproductions destinées à une utilisation collective. Toute représentation ou reproduction intégrale ou partielle faite par quelque procédé que ce soit, sans consentement de l'auteur ou de ses ayants causes, est illicite et constitue une contrefaçon sanctionnée par les articles 425 et suivants du code pénal.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull S.A. en cas de dommages résultant de son application. Des corrections ou modifications au contenu de ce document peuvent intervenir sans préavis ; des mises à jour ultérieures les signaleront éventuellement aux destinataires.

Préface

INTRODUCTION

Le présent manuel accompagne deux autres tomes consacrés à IDS/II: IDS/II User Guide et IDS/II Data Base Administrator Guide tous deux référencés ci-dessous. Son but est de fournir une base de référence sur le sujet et d'aider à l'utilisation des deux manuels cités plus haut.

IDS/II ("Integrated Data Store II" = "Stockage Intégré Des Données") constitue un ensemble de techniques, d'utilitaires et de langages spécialisés qui permet à l'utilisateur de définir et de mettre en oeuvre sa propre base de données quel que soit son degré de complexité, puis d'y accéder et de la mettre à jour.

Les principaux outils à disposition de l'utilisateur sont: le langage DDL qui sert à décrire le schéma de la base de données, le langage DMCL qui sert à la gestion des supports d'enregistrement, le langage DML pour l'interface avec les langages de programmation standards (par exemple COBOL) et les fonctions de logiciel/micro logiciel utilisées par de tels langages. Les pages qui suivent sont consacrées à la description de ces outils et de leurs fonctions associées.

La lecture des manuels dont la liste suit est également recommandée :

- IDS/II Data Base Administrator Guide (réf: 47 A2 13UD) qui fournit des informations complémentaires sur la manière de créer et de gérer les bases de données.
- IDS/II User Guide (réf: 47 A2 12UD) qui constitue le guide pratique de mise en oeuvre d'IDS/II en fonction des besoins spécifiques de l'utilisateur.
- IOF Terminal User's Guide (3 tomes: 47 A2 01 UJ/02 UJ/03 UJ) pour les informations sur la langage GPL (langage de gestion spécifique à GCOS).
- JCL User Guide (réf: 47 A2 12UJ) et JCL Référence Manual (47 A2 11UJ) pour plus d'informations sur le langage de gestion de travaux JCL.
- COBOL User Guide (réf: 47 A2 02UL) pour plus de détails sur l'utilisation du COBOL en IDS/II.
- LINKER (réf: 47 A2 10UP) pour la construction et l'édition de liens des programmes utilisateurs.
- General Access Control (réf: 47 A2 11UF) qui décrit la gestion générale des accès (aussi appelé GAC).

Manuel de référence IDS/II

- TDS Programmer's Manual (réf: 47 A2 03UT) pour la description du système transactionnel TDS.

La plupart de ces manuels existent en français ou sont en cours de traduction. Pour plus de précisions, veuillez consulter le "Catalogue de la Documentation" et le fascicule "Documents Nouveaux", qui portent tous deux la référence 00 F4 7210.

Table des matières

1.	Langage DDL	1-1
1.1	VUE DU SCHEMA	1-1
1.1.1	Indépendance programme - données	1-2
1.1.2	Description du DDL de schéma	1-2
1.1.3	Format origine DDL	1-3
1.2	REPRESENTATION	1-4
1.3	JEU DE CARACTERES	1-6
1.4	PONCTUATION	1-7
1.4.1	Délimiteurs	1-7
1.4.2	Séparateurs	1-7
1.5	MOTS	1-8
1.5.1	Généralités	1-8
1.5.2	Mots réservés	1-8
1.6	NOMS	1-10
1.6.1	Représentation synonymique	1-11
1.6.2	Représentation unique	1-11
1.6.3	Maximum de représentations autorisées	1-12
1.7	LITTERAUX	1-13
1.7.1	Littéraux alphanumériques	1-13

1.7.2	Littéraux numériques	1-13
1.7.3	Littéraux hexadécimaux	1-14
1.8	COMMENTAIRES	1-15
1.9	NOM-DE-DONNEE (OU NOM-DE-ZONE) -UTILISATEUR.....	1-15
1.10	IDENTIFICATEURS-DE-DONNEE (OU IDENTIFICATEURS-DE-ZONE)	1-16
1.10.1	Indicage.....	1-16
1.10.2	Qualification.....	1-17
1.11	PARAMETRES BASE DE DONNEES	1-18
1.12	CONDITIONS	1-19
1.12.1	Utilisation des crochets.....	1-21
1.12.2	Exemples de conditions.....	1-21
1.13	SYNTAXE DU DDL DE SCHEMA.....	1-22
1.14	RUBRIQUE SCHEMA	1-24
1.15	RUBRIQUE AREA.....	1-24
1.16	RUBRIQUE RECORD	1-25
1.17	CLAUSE RECORD.....	1-26
1.18	CLAUSE LOCATION	1-27
1.19	CLAUSE WITHIN	1-29
1.20	CLAUSE CHECK (SOUS-RUBRIQUE ARTICLE)	1-31
1.21	CLAUSE NOM-DE-DONNEE-UTILISATEUR	1-32
1.22	CLAUSE TYPE.....	1-33

Préface

1.23	CLAUSE OCCURS.....	1-36
1.24	CLAUSE CHECK (SOUS-RUBRIQUE DONNEE).....	1-37
1.25	RUBRIQUE SET.....	1-39
1.26	CLAUSE SET.....	1-41
1.27	CLAUSE OWNER.....	1-42
1.28	CLAUSE ORDER.....	1-43
1.29	CLAUSE MEMBER.....	1-45
1.30	CLAUSE INSERTION RETENTION.....	1-46
1.31	CLAUSE DUPLICATES.....	1-48
1.32	CLAUSE KEY.....	1-49
1.33	CLAUSE SELECTION.....	1-51
2.	Langage DMCL.....	2-1
2.1	VUE DE LA MEMOIRE.....	2-1
2.1.1	Langage DMCL.....	2-2
2.1.2	Fonction PREALLOC.....	2-2
2.1.3	Indépendance Programme-Données.....	2-3
2.2	STRUCTURE DE L'AIRE MEMOIRE.....	2-5
2.2.1	Numéro d'aire.....	2-5
2.2.2	Caractéristiques des aires.....	2-5
2.2.3	Structure de page.....	2-7
2.3	ADRESSAGE D'ARTICLE MEMOIRE.....	2-9
2.3.1	Définition.....	2-9

2.3.2	Taille de pointeur local	2-11
2.3.3	Taille de pointeur global	2-12
2.4	MISE EN OEUVRE D'UN ENSEMBLE PAR CHAINE DE POINTEURS	2-14
2.4.1	Numéro d'ensemble	2-14
2.4.2	Nombre de pointeurs	2-14
2.4.3	Ensembles locaux	2-14
2.4.4	Ensembles globaux	2-16
2.5	STRUCTURE D'ARTICLE MEMOIRE	2-17
2.5.1	Numéro d'article	2-17
2.5.2	Structure d'article	2-17
2.5.3	Exemple	2-18
2.6	SELECTION DU PARAMETRE PAGE-SIZE	2-20
2.7	SELECTION DU PARAMETRE LINES-PER-PAGE	2-22
2.8	SELECTION DE L'INTERVALLE CALC	2-23
2.9	SYNTAXE DMCL	2-24
2.9.1	Rubrique schéma	2-25
2.9.1.1	Clause SCHEMA	2-25
2.9.1.2	Clause EXTEND NUMBER OF AREAS	2-26
2.9.1.3	Clause EXTEND GLOBAL POINTERS	2-27
2.9.1.4	Clause NO LOCAL POINTERS	2-27
2.9.1.5	Clause BUFFER POOL	2-28
2.9.1.6	Clause NUMBER OF BUFFERS	2-29
2.9.2	Rubrique AREA	2-30
2.9.2.1	Clause AREA NAME	2-31
2.9.2.2	Clause AREA INTERNAL FILE NAME	2-31
2.9.2.3	Clause NUMBER OF PAGES	2-32
2.9.2.4	Clause LINES-PER-PAGE	2-33
2.9.2.5	Clause PAGE-SIZE	2-33
2.9.2.6	Clause CALC-INTERVAL	2-34
2.9.2.7	Clause EXTEND LOCAL POINTERS	2-34
2.9.3	Rubrique RECORD	2-35
2.9.3.1	Clause RECORD	2-36
2.9.3.2	Clause RANGE	2-36
2.9.3.3	Clause MIGRATION	2-38
2.9.4	Rubrique SET	2-39
2.9.4.1	Clause SET	2-39
2.9.4.2	Clause NO LOCAL POINTERS	2-40

3.	Traitement du schéma	3-1
3.1	FONCTIONS DU PROCESSEUR DDLPROC.....	3-1
3.2	TRAITEMENT DES ELEMENTS SL ET DD.....	3-4
3.3	JCL ETENDU DU DDLPROC.....	3-8
3.3.1	Description des paramètres DDLPROC.....	3-8
3.4	JCL DE BASE DE DDLPROC.....	3-9
3.4.1	Description des paramètres.....	3-9
3.5	LANGAGE DE COMMANDE DDLPROC.....	3-10
3.5.1	Mots réservés.....	3-10
3.5.2	Commandes.....	3-11
3.6	COMMANDE TRANSLATE SCHEMA DDL.....	3-12
3.7	COMMANDE TRANSLATE DMCL.....	3-18
3.8	COMMANDE PRINT SCHEMA.....	3-21
3.9	COMMANDE DELETE SCHEMA.....	3-39
3.10	COMMANDE MOVE SCHEMA.....	3-40
3.11	COMMANDE STATUS.....	3-41
3.12	EXEMPLES DE TRAITEMENT DU DDLPROC.....	3-42
3.13	MISE AU POINT DE DDLPROC.....	3-45
4.	Réservation d'aire mémoire	4-1
4.1	UTILITAIRE PREALLOC.....	4-1

4.2	RESERVATION D'ESPACE	4-2
4.3	JCL ETENDU DE PREALLOC	4-4
4.4	DESCRIPTION DES PARAMETRES PREALLOC	4-5
4.5	EXEMPLES	4-8
5.	Langage DML de manipulation des données	5-1
5.1	GENERALITES	5-1
5.1.1	Langage Hôte	5-1
5.1.2	Langages de mise en oeuvre	5-2
5.1.3	Langages de haut niveau (COBOL)	5-2
5.2	LANGAGE DE MANIPULATION DES DONNEES COBOL	5-3
5.2.1	Représentation	5-4
5.2.2	Règles de syntaxe	5-5
5.2.2.1	Syntaxe COBOL	5-5
5.2.2.2	Mots réservés	5-5
5.2.2.3	Qualification de zone d'article	5-5
5.2.3	Section SUB-SCHEMA dans la DATA DIVISION	5-6
5.3	ELEMENTS DB-KEY	5-10
5.4	REGISTRES SPECIAUX IDS	5-11
5.4.1	DB-STATUS	5-11
5.4.2	DB-REALM-NAME	5-11
5.4.3	DB-RECORD-NAME	5-11
5.4.4	DB-SET-NAME	5-12
5.4.5	DB-DETAILED-STATUS	5-12
5.5	INDICATEUR D'ETAT DE LA BASE DE DONNEES	5-13
5.5.1	Code d'instruction et code d'état	5-13
5.5.2	Combinaisons des codes d'instruction et d'état	5-13
5.5.3	Utilisation des registres spéciaux dans la détection d'erreur	5-13
5.5.3.1	DB-REALM-NAME.....	5-13
5.5.3.2	DB-RECORD-NAME.....	5-14
5.5.3.3	DB-SET-NAME	5-14

5.5.3.4	DB-DETAILED-STATUS	5-14
5.6	INDICATEURS DE POINT COURANT	5-17
5.6.1	Point courant d'unité d'exécution.....	5-17
5.6.2	Point courant de type d'ensemble	5-17
5.6.3	Point courant de domaine	5-18
5.6.4	Point courant de type d'article	5-18
5.6.5	Valeurs de point courant.....	5-18
5.6.6	Répercussion des instructions du DML sur les indicateurs de point courant .	5-19
5.7	INSTRUCTIONS DML	5-27
5.7.1	ACCEPT.....	5-28
5.7.2	CONNECT.....	5-31
5.7.3	DISCONNECT.....	5-34
5.7.4	ERASE	5-36
5.7.5	FIND.....	5-39
5.7.6	FINISH.....	5-50
5.7.7	GET.....	5-51
5.7.8	Instruction IF de test de condition	5-52
5.7.9	MODIFY	5-55
5.7.10	READY	5-62
5.7.11	STORE.....	5-64
5.7.12	USE.....	5-68
5.8	FONCTIONS D'IMPLANTATION DE POINTS DE REPRISE ET DE POINTS DE CONSOLIDATION.....	5-70
5.8.1	Comparaison entre points de reprise et points de consolidation	5-70
5.8.2	Fonction système de mise en oeuvre d'un point de reprise	5-71
5.8.3	Fonction système de mise en oeuvre d'un point de consolidation.....	5-72
5.9	ACCES AU FICHER DE JALONNEMENT IDSTRACE.....	5-73
5.10	SIMULATION D'ALGORITHME CALC (HASHING)	5-77
6.	Compilation et édition de liens du COBOL ET DU DML.....	6-1
6.1	ELEMENTS SL ET DD.....	6-1
6.2	JCL DU COBOL ETENDU	6-2
6.3	COMPILATION COBOL.....	6-2

6.4	EDITION DE LIENS	6-3
6.5	EXEMPLE	6-4
7.	Exécution du programme utilisateur	7-1
7.1	CADRE D'EXPLOITATION AU MOMENT DE L'EXECUTION	7-1
7.1.1	Conditions préalables	7-1
7.1.2	Modes de traitement	7-2
7.2	OBJETS DE REFERENCE AU MOMENT DE L'EXECUTION	7-3
7.3	JCL DE BASE DE L'UTILISATION	7-6
7.4	ACCES CONCURRENTS ET PROTECTION DES AIRES	7-8
7.4.1	Partage de la base de données	7-8
7.4.2	Mode d'utilisation des aires	7-9
7.5	REGLES CONCERNANT LES RESTRICTIONS DE READY PAR ASSIGN	7-12
7.5.1	Protection des aires par les journaux	7-13
7.5.2	Cohérence des niveaux de protection d'aire pour le même utilisateur en traitement par lots et IOF	7-14
7.6	COMMANDES UTILISABLES A L'EXECUTION	7-15
7.6.1	Mots réservés	7-15
7.6.2	Commandes	7-16
7.6.2.1	Commande SCHEMA	7-17
7.6.2.2	Commande IGNORE TRANSIENT	7-18
7.6.2.3	Commande BUFFER POOL	7-19
7.6.2.4	Commande NUMBER OF BUFFERS.....	7-20
7.6.2.5	Commande SAVE CURRENCIES.....	7-21
7.6.2.6	Commande SYNCHRO.....	7-22
7.6.2.7	Commande NO WARNING.....	7-23
7.6.2.8	Commande STATISTICS.....	7-24
7.6.2.9	Commande TRACE	7-27
7.6.2.10	Clause TRACE	7-28
7.6.2.11	Clause WHEN RECORD	7-29
7.6.2.12	Clause WHEN AREA.....	7-29
7.6.2.13	Clause WHEN DB-STATUS.....	7-30
7.6.2.14	Clause WHEN THRESHOLD	7-31
7.6.2.15	Clause WHEN PROGRAM	7-31
7.6.2.16	Clause PRINT.....	7-32

Préface

7.6.2.17	Clause IDSTRACE	7-36
7.6.2.18	Clause OPEN	7-36
7.6.2.19	Clause IAC	7-37
7.6.2.20	Commande CHECK PAGE	7-39
7.6.2.21	Commande INTERNAL FILE NAME	7-40
7.7	INTERPRETATION DES MESSAGES DU JOR	7-41
7.7.1	Format des messages	7-41
7.7.2	Codes de fonction du SGBD	7-42
7.7.3	Catégories des codes d'erreurs du SGBD.....	7-43
7.8	INTERPRETATION DES INFORMATIONS STATISTIQUES ET CHRONOLOGIQUES	7-45

Annexes

A.	Aide-mémoire des syntaxes DDL, DMCL et DML.....	A-1
A.1	STRUCTURE DU DDL DE SCHEMA	A-1
A.1.1	DDL de schéma - Rubrique schéma	A-2
A.1.2	DDL de schéma - Rubrique aire	A-2
A.1.3	DDL de schéma - Sous-rubrique article	A-2
A.1.4	DDL de schéma - Sous rubrique donnée	A-2
A.1.5	DDL de schéma - Sous-rubrique ensemble.....	A-3
A.1.6	DDL de schéma - Sous-rubrique détail.....	A-4
A.2	STRUCTURE DU DMCL	A-5
A.2.1	DMCL Rubrique schéma.....	A-6
A.2.2	DMCL - Rubrique article	A-6
A.2.3	DMCL - Rubrique ensemble	A-6
A.2.4	DML dans la DATA DIVISION	A-7
A.2.5	DML dans la PROCEDURE DIVISION (Partie Déclarative)	A-7
A.2.6	DML dans la PROCEDURE DIVISION (Partie non déclarative)	A-8
B.	Différences entre GCOS64 1E et GCOS 7	B-1
B.1	NOUVELLES SPECIFICATIONS DANS PREALLOC	B-1
B.1.1	Compatibilité.....	B-1
B.2	NOUVELLES SPECIFICATIONS COBOL/DML POUR LA COMPILATION ET L'EDITION DE LIENS.....	B-2
B.2.1	Compatibilité.....	B-3

Préface

B.3	NOUVELLES SPECIFICATIONS DANS L'EXECUTION DU PROGRAMME UTILISATEUR.....	B-4
B.3.1	Compatibilité.....	B-4

Illustrations

Figures

1-1	Echantillon DDL de schéma origine	1-22
1-2	Echantillon DDL de schéma origine	1-23
2-1	Vue de la mémoire entre vue du schéma et supports.....	2-1
2-2	Indépendance Programme-Données.....	2-4
2-3	Découpage de l'aire mémoire	2-6
2-4	Structure de page	2-8
2-5	Structure des pointeurs locaux, globaux et des éléments COBOL DB-KEY.....	2-10
2-6	Pointeurs locaux et globaux dans des aires de tailles différentes	2-13
2-7	Ensemble local mono-aire.....	2-14
2-8	Ensemble local multi-aire.....	2-15
2-9	Ensemble global	2-16
2-10	Structure de l'article mémoire	2-17
2-11	Exemple d'agencement d'un article mémoire.....	2-19
2-12	Echantillon DMCL origine.....	2-24
3-1	Fonctions du Processeur DDLPROC.....	3-2
3-2	Accès du processeur DDLPROC aux bibliothèques SL et aux bibliothèques BIN	3-7
3-3	Etat imprimé de traduction du DDL de schéma	3-15
3-4	Etat imprimé de traduction DMCL	3-20
3-5	Etat imprimé STORAGE sans option COMPACT.....	3-23
3-6	Etat imprimé STORAGE avec option COMPACT	3-24
3-7	Listage DML.....	3-25
3-8	Traitement du schéma PERSONNEL.....	3-43
3-9	Traitement du schéma de test PERSONNEL	3-44
4-1	Fonction PREALLOC	4-1
5-1	Accès utilisateur à IDSTRACE (un programme).....	5-74
5-2	Accès utilisateur à IDSTRACE (plusieurs programmes)	5-74
7-1	Objets référencés à l'exécution	7-5
7-2	Statistiques de session.....	7-26
7-3	Listage du jalonnement externe	7-35

Préface

Tableaux

1-1	Mots Reservés du langage DDL.....	1-9
1-2	Comparaison de deux identificateurs-de-donnée.....	1-21
1-3	Comparaison d'un identificateur-de-donnée et d'un littéral	1-21
1-4	Correspondance entre types de données DDL et COBOL	1-35
2-1	Taille de pointeur local	2-11
2-2	Taille de pointeur global.....	2-12
2-3	Adressage par pointeur global.....	2-13
2-4	Valeurs conseillées pour la taille de page.....	2-21
5-1	Codes d'instruction.....	5-14
5-2	Codes d'état de base de données	5-15
5-3	Combinaisons des codes d'instruction et d'état.....	5-16
5-4	Utilisation et mise à jour des points courants.....	5-21
7-1	Contexte du mode d'utilisation des aires	7-10
7-2	Mode d'utilisation final (au niveau fichier) après annulation de READY par ASSIGN 7-12	
7-3	Cohérence des niveaux de protection des aires pour un même utilisateur en modes par lots et IOF.....	7-14
7-4	Codes des fonctions du SGBD dans le JOR.....	7-43
7-5	Catégories des codes d'erreur du SGBD dans le JOR.....	7-44

1. Langage DDL

1.1 VUE DU SCHEMA

Le langage de description des données du schéma (Schema Data Description Language) ou Langage DDL permet de décrire la structure logique de l'information que le programmeur voit lorsqu'il écrit des programmes destinés à la base de données. Il existe deux notions essentielles à la compréhension de la vue du schéma :

- Les articles et leurs zones associées qui représentent des éléments externes comme par exemple les commandes, les factures, les clients, etc.
- Les ensembles qui établissent des relations entre ces éléments, comme par exemple la relation qui existe entre une commande spécifique et un client particulier...

Une notion complémentaire est désignée par le terme aire qui s'utilise pour découper les articles sur une base logique (une aire représentant par exemple une implantation géographique de l'entreprise) mais peut répondre également à d'autres utilisations :

- lorsque, pour des raisons pratiques, il n'est pas conseillé d'avoir la totalité de la base de données en ligne,
- lorsque des zones contiennent des informations "sensibles" (dossiers et paye du personnel par exemple).

1.1.1 Indépendance programme - données

Si le terme données désigne la vue de la base de données qui est fournie par le schéma, il est évident que la logique du programme dépend de la structure décrite. Si la structure des données est modifiée, la logique du programme en est affectée.

Si le terme données indique l'ensemble des chaînes de bits apparemment non structurées sur un disque, alors on peut parler de l'indépendance programme-données dans deux domaines :

- Il existe plusieurs options de projection de la structure du schéma sur la chaîne de bits. Par exemple, si dans une version future, pour des raisons d'efficacité, un ensemble est mis en oeuvre à l'aide d'index au lieu de l'actuelle chaîne de pointeurs, la logique de la séquence NEXT/PRIOR/OWNER est conservée et la logique du programme individuel n'en est pas affectée.
- L'utilisation du même schéma sur plusieurs représentations de la base de données permet d'employer les mêmes programmes. Les tailles respectives de ces différentes représentations peuvent ainsi changer (par exemple différence de taille entre une version de test et la taille réelle de la base de données).

1.1.2 Description du DDL de schéma

Un schéma écrit en langage DDL se subdivise en quatre types de rubriques qui expriment :

- L'identification du schéma (rubrique schéma).
- La définition des aires (rubrique aire).
- La définition des articles (rubrique article).
- La définition des ensembles (rubrique ensemble).

Dans un schéma, chaque aire, type d'article et type d'ensemble nécessite une rubrique distincte. Chaque schéma ne comporte qu'une seule rubrique schéma. La définition de la séquence des divers types de rubriques d'un schéma doit se conformer aux règles suivantes :

- La rubrique schéma doit être la première rubrique.
- Une rubrique aire doit précéder la rubrique article pour tous les articles contenus dans cette aire.
- Une rubrique article doit précéder une rubrique ensemble qui contient cet article.
- Une rubrique article doit comprendre une sous-rubrique article suivie d'un zéro et d'une ou plusieurs sous-rubriques zone.
- Une rubrique ensemble doit comprendre une sous-rubrique ensemble suivie au moins d'une sous-rubrique détail (MEMBER).

Une séquence rubrique-sous-rubrique est composée d'une ou plusieurs clauses qui en décrit les attributs. L'écriture des clauses peut se faire dans n'importe quel ordre à condition que la première clause désigne la rubrique-sous-rubrique. Toutes les rubriques et sous-rubriques doivent se terminer par un point. Chaque séquence rubrique-sous-rubrique est décrite de la manière suivante dans ce manuel :

- Description de la fonction de la séquence rubrique-sous-rubrique.
- Structuration en clauses de la séquence rubrique-sous-rubrique.
- Formats généraux de la séquence rubrique-sous-rubrique, c'est-à-dire le format général de chacune des clauses qui peuvent être spécifiées dans une séquence rubrique-sous-rubrique.
- Description propre à chaque clause.

Chaque clause est décrite de la manière suivante :

- Fonction.
- Format général.
- Règles de syntaxe.
- Règles générales.

Un format général décrit l'agencement des éléments qui constituent une clause. Les "locutions" sont les subdivisions d'une clause et doivent être écrites dans l'ordre indiqué ici. Une règle de syntaxe peut amplifier ou restreindre l'usage d'un élément du format général. Une règle générale peut amplifier ou restreindre les fonctions attribuées à un format général ou à ses éléments constitutifs.

1.1.3 Format origine DDL

Le DDL de schéma est un langage qui n'exige pas de format particulier. Il n'existe pas de zone spéciale pour numérotation de séquence, repérage de commentaire ou identification de programme. L'écriture DDL n'a pas pour base la ligne. Toutefois la fin de ligne est un séparateur lorsqu'elle n'apparaît pas à l'intérieur d'une chaîne délimitée (voir plus loin dans ce chapitre).

Pour être acceptées par le traducteur DDL, les rubriques de description d'entrées et les unités de bibliothèque contenant du DDL doivent être du type DATA ou DATASSF.

1.2 REPRESENTATION

La représentation des formats et les règles qui s'y appliquent sont indiquées ci-dessous :

- Les éléments constitutifs d'une clause sont des mots majuscules, des mots minuscules, des symboles et des caractères spéciaux.
- Les mots majuscules soulignés constituent les mot-clés du format. Exemple: SCHEMA NAME IS nom-schéma.
- Les mots majuscules non soulignés sont facultatifs. Dans l'exemple précédent, NAME et IS sont facultatifs.
- Les mots minuscules sont des termes génériques qui doivent être remplacés par des noms ou des valeurs appropriés. Dans l'exemple précédent nom-schéma doit être remplacé par le nom défini par l'utilisateur qui désigne ce schéma.
- L'encadrement d'une partie d'un format général entre des symboles spéciaux a, pour ce qui concerne les choix à opérer, la signification suivante :

Facultatif.

[a] aucun
[b] ou
[c] maximum 1
Obligatoire.

{a} au moins 1
{b} et
{c} maximum 1
Obligatoire.

a au moins 1
b et
c maximum tous une fois

Une élipse, représentée par (...) indique qu'une répétition est possible. La partie du format qui peut être répétée doit être encadrée par des crochets gauche et droite ({ }) ou des accolades gauche et droite qui s'apparient logiquement et doivent précéder la représentation de l'élipse.

Langage DDL

Exemples :

Format	Représentations possibles
[<u>POINTERS</u>]	espace (ou blanc) ou POINTERS
{ <u>UPDATE</u> } { <u>RETRIEVAL</u> }	UPDATE ou RETRIEVAL
<u>ACCEPT</u> <u>CONNECT</u> <u>DISCONNECT</u>	ACCEPT ou CONNECT ou DISCONNECT ou ACCEPT CONNECT ou ACCEPT DISCONNECT ou CONNECT DISCONNECT ou ACCEPT CONNECT DISCONNECT
[nom-aire] ...	espace (ou blanc) ou AR-4 ou AR-2 AR-6 AR-3
{nom-article} ...	REC-2 ou REC-4 REC-8

1.3 JEU DE CARACTERES

Les lettres de l'alphabet, des chiffres et des symboles constituent le jeu de 52 caractères du langage DDL. L'ensemble des caractères qui peuvent être utilisés dans l'écriture d'un schéma sont indiqués ci-dessous :

Caractères	Nom
0,1, ...,9	Chiffres
A,B, ...,Z	Lettres
	Espace (ou blanc)
+	Plus
-	Moins
,	Virgule
.	Point (marque également la virgule décimale)
"	Guillemets
(Parenthèse gauche
)	Parenthèse droite
'	Aphostrophe
\$	Dollars
/	Barre de fraction (ou barre oblique)
*	Astérisque
=	Egal
<= 256>	Plus grand que (ou supérieur à;)
	Plus petit que (ou inférieur à;)

L'utilisation de caractères supplémentaires, choisis parmi le jeu complet de caractères du DPS7, est autorisé dans les chaînes délimitées (littéraux non numériques, commentaires) ou dans la forme "échappement" (\$...\$) des noms.

1.4 PONCTUATION

Il existe deux types de caractères de ponctuation: délimiteur et séparateur. La fin d'une ligne est considérée comme un séparateur.

1.4.1 Délimiteurs

Un délimiteur s'utilise pour indiquer le début et la fin d'une chaîne délimitée. Une chaîne délimitée est constituée d'un ensemble quelconque de caractères autorisés dans le jeu du DPS7. Elle doit être encadrée par des délimiteurs. Un caractère délimiteur peut être utilisé dans le corps d'une chaîne. Toutefois il doit être répété. Les délimiteurs sont :

- les guillemets qui s'utilisent pour délimiter les littéraux alphanumériques et hexadécimaux ainsi que les commentaires ;
- l'apostrophe, qui s'utilise pour délimiter la forme synonyme des noms.

1.4.2 Séparateurs

Un séparateur s'utilise pour distinguer les mots et les littéraux et pour indiquer certaines constructions du langage. Dans une chaîne délimitée un séparateur n'a pas de signification. S'il est représenté par un caractère il devient partie intégrante de la chaîne. S'il constitue une fin de ligne il est ignoré. Les séparateurs sont :

- L'espace. Un ou plusieurs espaces constituent un séparateur. Un ou plusieurs espaces peuvent constituer un complément (avant ou après) à tout autre séparateur.
- La virgule. La virgule est un séparateur obligatoire entre les indices dans le format d'identificateur de donnée. Ailleurs il est facultatif.
- Le point virgule. Un point virgule peut s'utiliser comme un espace. Il sera d'ailleurs traité comme tel.
- Le point. La fin d'une rubrique ou d'une sous-rubrique doit être marquée par un point suivi d'un espace ou d'une fin de ligne.
- La paire de crochets (gauche et droite). Elle est nécessaire pour délimiter un indice d'identificateur de données. La paire de crochets s'utilise également dans l'écriture des conditions.
- La fin de ligne. La fin de ligne suit les mêmes règles et a le même effet qu'un espace.

1.5 MOTS

1.5.1 Généralités

Un mot est constitué d'une suite de 30 caractères maximum. Chaque caractère est choisi dans le jeu alphabétique (A à Z), numérique (0 à 9), ou le signe -. Une exception est constituée cependant par le fait que le signe moins (ou tiret) ne peut apparaître comme premier ou dernier caractère d'un mot. Un mot doit toujours commencer par une lettre.

Exemples :

Valable	Non Valable
A, A-B Z9999,	-A A\$B, A+B

1.5.2 Mots réservés

L'utilisation d'une liste de mots réservés est autorisée. Toutefois ces mots réservés ne peuvent être utilisés comme noms définis par l'utilisateur. Ils comprennent les mot-clés et les mots facultatifs.

L'usage du mot-clé est obligatoire chaque fois que le format dans lequel il est inclus est utilisé. Les mot-clés, à l'intérieur d'un format, sont représentés par des majuscules soulignées.

Un mot facultatif est un mot que l'utilisateur peut, au choix, inclure ou non. Ils sont représentés dans chaque format par des majuscules non soulignées. Les fautes d'orthographe ou le remplacement d'un mot facultatif par un autre ne sont pas autorisés.

Le Tableau 1-1 donne la liste des mots réservés avec, entre parenthèses, leur forme abrégée.

Tableau 1-1. Mots Reservés du langage DDL

ACCESS-CONTROL	EXCLUSIVE (EXCL)	ORGANIZATION
ACTUAL	FIND	OWNER
AFTER	FIRST	PACKED
ALL	FIXED	PACKED-2
ALLOWED	FLOAT	PERMANENT
ALTER	FOR	PICTURE (PIC)
ALWAYS	GE	POSTPONED
AND	GET	PRIOR
ANY	GT	PROCEDURE (PROC)
APPLICATION	IDENTIFIED	PROCESSABLE
ARE	IN	PROTECTED (PROT)
AREA	INDEX	RANGE
AREA-ID	INDEXED	REAL
ASCENDING (ASC)	INSERT	RECORD
AUTOMATIC	INSERTION	RECORD-TYPE
BEFORE	IS	REMOVE
BINARY (BIN)	KEY	RESULT
BIT	KEYS	RETENTION
BY	LAST	RETRIEVAL (RETR)
CALC	LE	SCHEMA
CALC-KEY	LINKED	SEARCH
CALL	LOCATION (LOC)	SELECTION
CHARACTER (CHAR)	LOCK	SEQUENCE
CHECK	LOCKS	SEQUENTIAL
CLOSE	LT	SET
COMMENT	MANDATORY (MAND)	SIGNED
COMPLEX	MANUAL	SORTED
CONSTRAINT	MEMBER	SOURCE
COPY	MEMBERS	STORE
CURRENT	MEMBERSHIP	STRUCTURAL
DATA-BASE-KEY (DB-KEY)	MODE	SYSTEM
DECIMAL	MODIFY	SYSTEM-DEFAULT
DECODING	NAME	TEMPORARY (TEMP)
DEFINED	NE	THEN
DELETE	NEXT	THIS
DESCENDING (DESC)	NON EXCLUSIVE	THRU (THROUGH)
DIRECT	NONNULL	TIMES
DISPLAY	NOT	TO
DUPLICATES (DUP)	NULL	TYPE
DURING	OCCURS	UNPACKED
DYNAMIC	OF	UNSIGNED
ENCODING	ON	UPDATE
END	ONLY	USING
END-SCHEMA	OPEN	VALUE
EQ	OPTIONAL (OPT)	VIA
EQUAL	OR	VIRTUAL
ERROR	ORDER	WHERE
		WITHIN

1.6 NOMS

Un nom est un mot qui commence par une lettre. Les différents types de noms sont indiqués ci-dessous :

- nom-de-donnée-utilisateur
- nom-article
- nom-aire
- nom-ensemble
- nom-schéma
- paramètre-base-de-données.

Les noms peuvent se regrouper également en :

- noms-rubrique = noms-schéma + noms-aire + noms-article + noms-ensemble
- noms-des-données = noms-rubrique + noms-de-donnée-utilisateur.

L'exemple fourni plus loin illustre un schéma complet d'une base de données constituée de deux types d'articles. Pour plus de clarté, tous les noms définis par l'utilisateur ont été soulignés, ce qui ne constitue en aucun cas le format réel d'écriture comme il a été défini plus haut et tel qu'il sera utilisé tout au long de ce manuel.

```

SCHEMA NAME IS nom-schéma
AREA NAME IS nom-aire-A
AREA NAME IS nom-aire-B

RECORD NAME IS nom-article-A

    LOCATION MODE DIRECT paramètre-base-de-données-A

    WITHIN nom-aire-A

    02 nom-de-donnée-utilisateur-A TYPE CHARACTER 1

RECORD NAME IS nom-article-B
    LOCATION MODE VIA nom-ensemble-A

    WITHIN nom-aire-A nom-aire-B

    AREA-ID IS paramètre-base-de-données-B

    02 nom-de-donnée-utilisateur-B TYPE IS SIGNED BINARY 15

SET NAME IS nom-ensemble-A
    OWNER IS nom-article-A ORDER PERMANENT INSERTION LAST

    MEMBER IS nom-article-B

    INSERTION AUTOMATIC RETENTION MANDATORY

    SET SELECTION IS THRU nom-ensemble-A

```

Langage DDL

```
OWNER IDENTIFIED BY DATA-BASE-KEY  
EQUAL TO paramètre-base-de-données-C  
END-SCHEMA
```

1.6.1 Représentation synonymique

Un nom peut avoir un synonyme sous forme d'une chaîne, de 30 caractères maximum, choisis dans le jeu de caractères autorisés du DPS7, délimitée par des apostrophes. Le caractère apostrophe lui-même peut être inclus dans la chaîne à condition d'être répété.

Exemples :

```
'WORK.SCHEMA' , 'O''CLOCK'
```

Lorsqu'un nom défini par l'utilisateur est identique à un mot réservé, il doit toujours s'écrire sous sa forme synonyme.

Exemple :

```
SCHEMA NAME IS 'SCHEMA'
```

1.6.2 Représentation unique

Les règles suivantes s'appliquent pour respecter l'unicité des noms :

1. Il ne doit pas y avoir de duplication de noms-rubrique. Deux aires par exemple ne peuvent disposer du même nom, de même qu'une aire ne peut porter le même nom que le schéma ou qu'un article ou un ensemble.
2. Un nom-rubrique ne peut être identique à un nom-de-donnée-utilisateur.
3. Il ne doit pas exister de nom-de-donnée-utilisateur en double dans une rubrique article.
4. Deux éléments de donnée ou deux agrégats de données d'article peuvent porter le même nom à condition de figurer dans deux rubriques articles différentes.
5. Un paramètre de base de données ne peut être le même qu'un nom de donnée.

1.6.3 Maximum de représentations autorisées

Comme indiqué ci-dessous :

Manuel de référence IDS/II

Spécification	Nombre maximum
Noms-aire	2048
Noms-article	2048cc
Noms-ensemble	2048
Noms-de-donnée-utilisateur dans un type d'article	2048
Paramètres-base-de-données	2048

1.7 LITTERAUX

Un littéral est une chaîne de caractères qui exprime une valeur. Il existe trois types différents de littéraux en IDS/II :

- Littéraux alphanumériques.
- Littéraux numériques.
- Littéraux hexadécimaux.

1.7.1 Littéraux alphanumériques

Un littéral alphanumérique s'exprime sous forme d'une chaîne de 256 caractères maximum choisis dans le jeu de caractères DPS 7 et délimités par des guillemets. La valeur est exprimée par les caractères représentés. La longueur de la chaîne est indiquée par le nombre de positions de caractères à l'intérieur de la chaîne. Si le caractère guillemet doit apparaître dans la chaîne il doit être répété et ne comptera qu'une fois pour la détermination de la longueur de la chaîne.

Exemples :

```
"1ER JANVIER" valeur: 1ER JANVIER longueur: 11
"CODE ""4"" valeur: CODE "4" longueur: 8
```

1.7.2 Littéraux numériques

Un littéral numérique exprime un nombre. La seule forme de littéral numérique autorisée est le littéral décimal à signe décimal fixe. Il est composé :

- des chiffres 0 à 9,
- des signes plus ou moins,
- de la marque décimale (le point).

Sa longueur maximum est de 30 chiffres. Lorsque la marque décimale (le point) est représentée elle doit être suivie au moins d'un chiffre. La valeur représentée est celle de la valeur algébrique conventionnelle.

Exemples :

```
Valable      : 1 1.0 +322 +0.004 -5789.333 -.004
Non valable: 122.      +322.1A4
```

1.7.3 Littéraux hexadécimaux

Un littéral hexadécimal est une chaîne de 512 caractères maximum, choisis parmi :

- les chiffres 0 à 9,
- les lettres A à F.

Il est délimité par des guillemets et le guillemet droit doit être suivi du caractère X. Le nombre de caractères doit être toujours pair. La valeur représentée est celle de l'équivalent EBCDIC de chaque paire de caractères dans la chaîne. La longueur de la chaîne est le nombre de caractères divisé par 2. Une chaîne hexadécimale s'utilise pour représenter des caractères pour lesquels il n'existe pas de symbole externe ou qui ne sont pas disponibles sur un appareil d'entrée.

Exemple :

"9481992A" X valeur: mars longueur: 4

1.8 COMMENTAIRES

On peut introduire des commentaires dans le langage DDL pour renseigner la programmation du schéma. Ces commentaires peuvent être écrits dès qu'un espace apparaît comme caractère séparateur. Un commentaire peut s'écrire de deux manières :

- soit :
 - le mot réservé COMMENT,
 - un certain nombre d'espaces au aucun,
 - le texte du commentaire encadré par des guillemets,

- soit :
 - le texte du commentaire encadré par des caractères /*.

Si on utilise un caractère guillemet dans le corps du commentaire il doit être répété.

Exemples :

```
COMMENT SCHEMA INVENTAIRE  
  
SCHEMA NAME IS INVENTAIRE  
  
02 QTE TYPE DEC 6. /* QUANTITE COMMANDEE */  
  
02 SEX TYPE CHAR 1. COMMENT "VALEUR "M" si Masculin  
                               VALEUR "F" si Féminin"
```

1.9 NOM-DE-DONNEE (OU NOM-DE-ZONE) -UTILISATEUR

Un nom-de-donnée-utilisateur est un nom défini par l'utilisateur pour désigner une donnée élémentaire ou un agrégat de données.

1.10 IDENTIFICATEURS-DE-DONNEE (OU IDENTIFICATEURS-DE-ZONE)

Un identificateur-de-donnée permet de citer une donnée (élémentaire ou partie d'un agrégat) déclarée dans le schéma. Il consiste en un nom-de-donnée suivi, si nécessaire, par une combinaison syntaxiquement correcte d'indices et de qualificateurs nécessaires pour remplir la condition de représentation unique de la référence. La qualification, dans certains formats, n'est pas nécessaire si l'ambiguïté peut-être résolue par le contexte (les règles de syntaxe spécifiques).

Le format d'un identificateur-de-donnée se présente :

```
nom-de-donnée-utilisateur [(entier-1 [ ,entier-2 ]... )][{OF} nom-article]
                           [           ] [IN] ]
```

1.10.1 Indicage

Les indices peuvent s'utiliser lorsque la citation d'une donnée est nécessaire à l'intérieur d'un vecteur ou d'un groupe répétitif. L'indice doit toujours être un entier.

La plus petite valeur possible d'un indice est 1. Cette valeur se réfère à la première apparition du nom-de-donnée-utilisateur cité. La plus grande valeur d'un indice est 32767.

Lorsque plus d'un indice est nécessaire, on les écrit de gauche à droite dans l'ordre croissant des numéros de niveau des agrégats de données.

Par exemple :

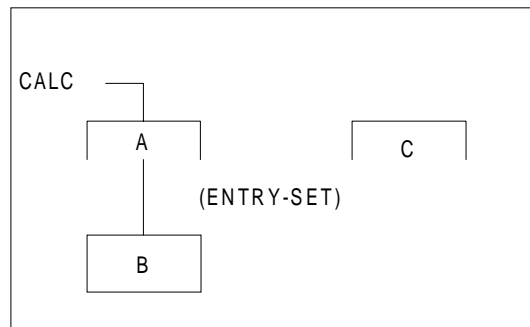
```
RECORD NAME IS R04
...
02 R04-T1 OCCURS 3.
03 R04-P1 TYPE IS CHAR 2.
03 R04-P2 OCCURS 2 TYPE IS SIGNED PACKED DEC 3.
...
SET NAME IS S17
...
INSERTION IS SORTED BY DEFINED KEYS...
MEMBER IS R04...
KEY IS ASCENDING R04-P2 (1,2)
```

Le nombre d'indices est limité à 3 car c'est la valeur admise par le COBOL.

1.10.2 Qualification

Lorsque le même nom-de-donnée-utilisateur doit être déclaré dans plus d'une rubrique article, son utilisation comme identificateur-de-donnée peut nécessiter une qualification de manière à remplir la règle d'unicité. Lorsque la qualification est nécessaire, des règles de syntaxe le précisent. Même si un nom ne nécessite pas de qualification il peut être qualifié.

Exemple :



```

RECORD NAME IS A
...
02 MY-DB-ID TYPE IS CHAR 5.
RECORD NAME IS C
...
02 MY-DB-ID TYPE IS CHAR 5.
SET ENTRY-SET OWNER A... MEMBER B...
  SET SELECTION FOR ENTRY-SET IS THRU ENTRY-SET
    OWNER IDENTIFIED BY CALC-KEY MY-DB-ID EQUAL
      TO MY-DB-ID IN C

```

Remarque : La donnée CALC est généralement localisée dans la zone de manoeuvre utilisateur ou zone UWA (USER WORK AREA) affectée à l'article A, mais dans cet exemple une zone UWA affectée à une donnée d'un autre article est utilisée comme forme origine homonyme.

1.11 PARAMETRES BASE DE DONNEES

Un paramètre-base-de-données est un nom défini par l'utilisateur et disponible au moment de l'exécution pour accroître les noms-des-données (de la base de données). Les paramètres-base-de-données ne sont pas contenus dans des articles de la base de données mais sont utilisés pour la localisation d'articles (DIRECT), la sélection d'aire (AREA-ID) ou la sélection d'ensemble (EQUAL TO). Ils s'adaptent aux caractéristiques de l'usage pour lequel ils ont été spécifiés. C'est-à-dire que le paramètre est une clé-base-de-données lorsqu'il est utilisé dans la clause de localisation DIRECT, un nom d'aire lorsqu'il est utilisé dans la clause de sélection d'aire (WITHIN), et le même que la zone de contrôle lorsqu'il est spécifié dans la phrase EQUAL TO.

1.12 CONDITIONS

Le mot condition s'utilise d'une manière générale pour spécifier une expression qui sera évaluée comme vraie ou fausse par IDS/II au moment de l'exécution.

- Une condition se présente généralement sous la forme d'une ou plusieurs alternatives séparées par l'opérateur logique OR :

```
alternative-1 [OR alternative-2]...
```

La condition est vraie si l'une quelconque des alternatives (et éventuellement plusieurs, l'opérateur OR n'étant pas exclusif) est vraie. Autrement elle est fausse.

- Une alternative est constituée d'une ou plusieurs conditions-simples séparées par l'opérateur logique AND sous la forme:

```
condition-simple-1 [AND condition-simple-2]...
```

L'alternative est vraie si toutes les conditions simples sont vraies, autrement elle est fausse.

- Une condition simple se présente sous la forme :

```
{ condition - de - comparaison - 1      }
{ NOT (condition - de - comparaison - 2) }
{ NOT (condition - 1)                  }
```

La condition simple est vraie si :

- La condition-de-comparaison-1 est vraie.
- La condition-de-comparaison-2 est fausse.
- NOT est omis et la condition-1 est vraie.
- NOT est spécifié et condition-1 est fausse.

Dans tous les autres cas la condition simple est fausse.

- Une condition de comparaison consiste en une paire d'opérandes qui peuvent être des identificateurs-de-donnée ou des littéraux séparés par un opérateur de comparaison. Condition de comparaison et opérateurs de comparaison sont représentés ci-dessous :

{	LT	}	{	<	}
{	LE	}	{	<=	}
{	EQ	}	{	=	}
{	GE	}	{	>=	}
{	GT	}	{	>	}
{	NE	}	{	#	}

La condition de comparaison est vraie si la comparaison entre les opérandes est vérifiée, autrement elle est fausse. Il est possible d'effectuer une comparaison de deux identificateurs-de-donnée pour les cas illustrés au Tableau 1-2 plus loin (OUI dans l'intersection des colonnes).

La comparaison entre un identificateur-de-donnée et un littéral est autorisée dans les cas illustrés au Tableau 1-3 plus loin. Les opérandes sont de type numérique et il ne doit y avoir aucune perte de chiffres non nul (à gauche de la partie entière et à droite de la partie fractionnée) lorsque le littéral est traduit sous forme d'identificateur-de-donnée.

Si les opérandes sont des chaînes de données élémentaires, la comparaison s'exécute suivant l'ordre de classement EBCDIC. Si les opérandes sont des données numériques, la comparaison s'exécute en utilisant la valeur algébrique des données. La signification symbolique des opérateurs de comparaison est indiquée ci-dessous :

LT ou <	inférieur à
LE ou <=	inférieur ou égal à
EQ ou =	égal à
GE ou >=	supérieur ou égal à
GT ou >	supérieur à
NE ou #	différent de

Les opérateurs < <= = >= > doivent toujours être encadrés par un espace. Les deux caractères qui composent l'opérateur >= et <= doivent figurer sur la même ligne et sans espace intercalé.

1.12.1 Utilisation des crochets

Chaque alternative, dans une condition sans crochets, est évaluée et la valeur vraie de l'alternative est utilisée pour déterminer la valeur vraie de la condition complète.

En ce qui concerne les conditions pourvues de crochets, les expressions encadrées par des crochets sont évaluées en premier et la valeur vraie de la totalité de l'expression entre crochets est utilisée pour vérifier la condition complète. Lorsque les crochets sont imbriqués, c'est l'expression interne qui est évaluée la première suivie de l'expression intermédiaire etc.

Les crochets doivent toujours s'apparier.

1.12.2 Exemples de conditions

```

TAUX-DE-REMISE = 0.10
AGE >= 25
NOT (AGE < 25)
DATE-DE-NAISSANCE LE DATE-DE-DECES
MOIS = "JANVIER" AND JOUR <= 31 OR MOIS = "FEVRIER" AND JOUR <= 29
SEXE = "M" AND NOM-DE-JEUNE-FILLE = " "
OR SEXE = "F" AND NOT (NOM-DE-JEUNE-FILLE = " ")
RTYPE = "1" AND (SUBTYPE = "1" OR SUBTYPE = "2") OR
RTYPE = "2" AND ((SUBTYPE = "1" AND QTE < 1000) OR
                  (SUBTYPE = "2" AND QTE >= 1000))
OR RTYPE = "3"

```

Tableau 1-2. Comparaison de deux identificateurs-de-donnée

TYPE identificateur-1 / TYPE identificateur-2	DECIMAL m	DECIMAL m,+p	BINARY	CHARACTER
DECIMAL m	OUI		OUI	
DECIMAL m,+p		OUI (si échelle décimal identique)		
BINARY	OUI		OUI	
CHARACTER				OUI

Tableau 1-3. Comparaison d'un identificateur-de-donnée et d'un littéral

TYPE identificateur / TYPE littéral	DECIMAL m,+p	BINARY	CHARACTER
décimal signé/non signé/entier/fractionné	OUI	OUI	
chaîne caractères			OUI
chaîne hexadécimale			OUI

1.13 SYNTAXE DU DDL DE SCHEMA

Un schéma se divise en plusieurs sections ou rubriques :

- Schéma désigne le schéma.
- Aire désigne chaque aire dans la base de données.
- Article désigne et définit chaque type d'article dans la base de données. Spécifie les données élémentaires et la méthode d'implantation en mémoire.
- Ensemble désigne et définit chaque relation entre articles.

La séquence rubrique-sous-rubrique se termine par un point suivi d'un espace ou d'une fin de ligne. Le texte origine DDL complet se termine par le mot-clé END-SCHEMA. La séquence de DDL complète pour un schéma peut être ainsi résumée :

```
{Rubrique Schéma}
{Rubrique aire}...
{Rubrique article}...
[Rubrique ensemble]...
```

Chacune des rubriques est décrite en détail dans les pages qui suivent.

Tableau 1-4. Echantillon DDL de schéma origine (1/2)

```
COMMENT "*****"
COMMENT " *          SCHEMA DDL FOR TEST-GROUP *TG19*          * "
COMMENT "*****"

COMMENT "*****"
COMMENT " *          SCHEMA          * "
COMMENT "*****"

SCHEMA TG19-SH.

COMMENT "*****"
COMMENT " **          AREAS          * "
COMMENT "*****"

AREA TG19-A00.
AREA TG19-A01.

COMMENT "*****"
COMMENT " *          RECORDS          * "
COMMENT "*****"
```

Langage DDL

Tableau 1-4. Echantillon DDL de schéma origine (2/2)

```
COMMENT "*****"
COMMENT "*"      R01 (PART)      "*"
COMMENT "*****"
RECORD R01
    LOCATION CALC USING R01-NUM DUP NOT
    WITHIN ANY AREA AREA-ID ARID1.
    02      R01-IDENT      TYPE CHAR 7.
    02      R01-NUMH      TYPE CHAR 5.
    02      R01-NUM      TYPE UNPACKED DEC 3.
    02      R01-TRAILER  TYPE CHAR 4.

COMMENT "*****"
COMMENT "*"      R02 (COMPONENT)  "*"
COMMENT "*****"
RECORD R02
    LOCATION VIA S01
    WITHIN AREA OF OWNER.
    02      R02-IDENT      TYPE CHAR 7.
    02      R02-R01-UPH    TYPE CHAR 4.
    02      R02-R01-UP     TYPE UNPACKED DEC 3.
    02      R02-R01-DOWNH  TYPE CHAR 6.
    02      R02-R01-DOWN   TYPE UNPACKED DEC 3.
    02      R02-QTYH      TYPE CHAR 5.
    02      R02-QTY       TYPE UNPACKED DEC 3.
    02      R02-TRAILER  TYPE CHAR 4.

COMMENT "*****"
COMMENT "*"      SETS      "*"
COMMENT "*****"

COMMENT "*****"
COMMENT "*"      S01 (CALL-OUT)  "*"
COMMENT "*****"
SET S01
    OWNER R01
    ORDER PERMANENT INSERTION SORTED DEFINED KEYS DUP NOT.
    MEMBER R02
    INSERTION AUTO RETENTION MAND
    KEY ASCENDING R02-R01-DOWN
    SELECTION THRU S01
        OWNER IDENTIFIED BY CALC-KEY R01-NUM EQUAL TO R02-R01-UP.

COMMENT "*****"
COMMENT "*"      S02 (WHERE-USED)  "*"
COMMENT "*****"
SET S02
    OWNER R01
    ORDER PERMANENT INSERTION SORTED DEFINED KEYS DUP NOT.
    MEMBER R02
    INSERTION AUTO RETENTION MAND
    KEY ASCENDING R02-R01-UP
    SELECTION THRU S02
        OWNER IDENTIFIED BY CALC-KEY R01-NUM EQUAL TO R02-R01-DOWN
        AREA-ID EQUAL TO OTHER-ARID1.

END-SCHEMA.
```


1.14 RUBRIQUE SCHEMA

Fonction

Permet de désigner le schéma. Ce nom est introduit dans le dictionnaire du fichier du schéma et s'utilise dans toutes les références ultérieures à ce schéma. Cette clause est obligatoire.

Format général

```
SCHEMA NAME IS nom-schéma._
```

Règles de syntaxe

1. Nom-schéma est un nom d'une longueur de 30 caractères maxi.
2. Cette clause doit apparaître en premier dans le texte de schéma.
3. Nom-schéma doit être unique dans les noms-de-donnée de schéma.

Règle générale

Le schéma désigné par nom-schéma est constitué de toutes les rubriques DDL qui apparaissent entre la clause SCHEMA et le mot-clé END-SCHEMA.

1.15 RUBRIQUE AREA

Fonction

Permet de désigner une aire dans la base de données.

Format général

```
AREA NAME IS nom-aire._
```

Règles de syntaxe

1. Nom-aire doit être unique parmi les noms-de-donnée du schéma.

1.16 RUBRIQUE RECORD

Fonction

Permet de désigner et d'attribuer certaines caractéristiques aux articles et aux données élémentaires qui s'y rapportent dans la base de données.

Format général de la rubrique article

```
sous-rubrique article
[sous-rubrique données]...
```

Format général de la sous-rubrique article

RECORD NAME IS nom-article

```

                                { DIRECT parametre-base-de-donnees-1 }
LOCATION MODE IS { CALC   USING identificateur-de-donnees-1 ... }
                                { DUPLICATES ARE [NOT] ALLOWED }
                                { VIA nom-ensemble-1 SET }

```

```

WITHIN { { ANY } AREA } AREA-ID IS parametre-base-de-donnees-2 }
          { { { nom-aire-1 } ... } }
          { AREA OF OWNER }

```

[CHECK IS condition-1]

Format général de la sous-rubrique donnée

```

[ no-niveau-1 ]      parametre-base-de-donnees-1

[ { { { { UNPACKED } } } } ] ]
[ { { [ UNSIGNED ] { [ PACKED ] } } } ] ]
[ { { [ ] { PACKED-2 } } } ] ]
[ TYPE IS { { SIGNED } { UNPACKED } } DECIMAL } entier-1 [ entier-2 ] ] ]
[ { { [ PACKED ] } } } ] ]
[ { { { 15 } } } ] ]
[ { [ SIGNED ] BINARY { 31 } } ] ]
[ { CHARACTER } entier-3 ] ]

[ OCCURS      entier-4      TIMES ]
[ CHECK IS VALUE [ NOT ] { littéral-1 [ THRU littéral-2 ] } ... ] .

```

1.17 CLAUSE RECORD

Fonction

Permet de désigner un type-article du schéma, c'est-à-dire, de spécifier un nom pour tous les types d'article dans la base de données.

Format général

```
RECORD NAME IS nom-article
```

Règles de syntaxe

1. Nom-article doit être unique parmi les noms-de-donnée attribués par un schéma.
2. Il doit exister au moins une spécification nom-article dans le schéma.

Règles générales

Aucune.

1.18 CLAUSE LOCATION

Fonction

Permet de gérer l'affectation des clés-base-de-données à l'article.

Format général

```

LOCATION MODE IS
    { DIRECT paramètre-base-de-données-1 }
    { CALC USING {identificateur-de-données-1} ... }
    { DUPLICATES ARE [NOT] ALLOWED }
    { VIA nom-ensemble-1 SET }

```

Règles de syntaxe

1. Paramètre-base-de-données-1 se rapporte à la donnée générée par le compilateur COBOL comme clé-base-de-données. L'apparition de cet élément dans la clause le définit implicitement.
2. Identificateur-de-donnée doivent se rapporter à des données incluses dans le type d'article décrit (CALC).
3. Nom-ensemble-1 est le nom du type d'ensemble dans lequel le type-article est défini comme article détail (VIA).

Règles générales

1. Paramètre-base-de-données-1, du fait qu'il apparaît dans une clause LOCATION est traité comme une clé-base-de-données et non comme un élément d'article.
2. Le système de gestion de base de données SGBD* affecte une clé-base-de-données à un article lorsque l'instruction STORE est exécutée. C'est le cas également lorsqu'une instruction MODIFY CALC-KEY provoque une migration d'article.

*DATA BASE CONTROL SYSTEM (abréviation anglaise DBCS).

3. L'affectation des clés-base-de-données est sujette aux règles imposées par la clause WITHIN pour la même sous-rubrique article.
4. Si DIRECT est spécifié, au moment de l'exécution l'élément introduit par DIRECT s'utilise pour affecter la clé-base-de-données. Il peut être constitué d'une clé-base-de-données complète, c'est-à-dire qui se rapporte à la base de données en général, ou d'une clé-d'aire qui ne se rapporte qu'à une aire. De toutes façons, le SGBD n'utilise que la partie clé-d'aire.
5. Si CALC est spécifié, au moment de l'exécution les données associées à CALC sont utilisées pour affecter une clé-base-de-données. L'élément clé-base-de-données est développé par un algorithme fourni par le système.

6. La phrase DUPLICATES ARE NOT ALLOWED renseigne le SGBD pour qu'il interdise le rangement dans une aire de la base de données de plus d'un article de ce type (données CALC identiques).
7. Si VIA nom-ensemble-1 SET est spécifié, le SGBD affecte à l'article une clé-base-de-données, comme s'il devait désormais constituer un article détail du type-ensemble désigné dans VIA. Lorsqu'il affecte une clé-base-de-données, le SGBD consulte les critères de sélection définis pour le type d'ensemble concerné.

1.19 CLAUSE WITHIN

Fonction

Permet de définir les aires dans lesquelles les types d'article peuvent être rangés et de reconnaître ces différentes aires.

Format général

```

      { { ANY AREA } [ AREA-ID IS parametre-base-de-donnees-1 ] }
      { { { nom-aire-1 } ... } }
WITHIN { }
      { AREA OF OWNER }

```

Règles de syntaxe

1. Les noms d'aire doivent être ceux qui sont spécifiés par le schéma dans la rubrique aire.
2. Si OWNER est spécifié, la localisation de l'article doit se faire par VIA nom-ensemble.

Règles générales

1. Paramètre-base-de-données-1, du fait qu'il est mentionné dans la clause WITHIN est implicitement défini comme des données de type "chaîne de caractères" qui se conforment aux règles de représentation des noms-aire (30 caractères alphanumériques).
2. Lorsqu'un seul nom-aire est spécifié, tous les articles du type indiqué sont rangés dans l'aire ainsi désignée (nom-aire-1).
3. Lorsque plus d'un nom-aire est mentionné, le contenu du paramètre-base-de-données-1 spécifie l'aire dans laquelle l'article est rangé.
4. Si la spécification est ANY AREA, toutes les aires déclarées peuvent valablement contenir ce type d'article.
5. L'utilisation de ANY AREA suppose l'existence de plusieurs aires dans une liste. En ce cas AREA-ID doit être spécifié.
6. Paramètre-base-de-données-1, qui doit être identique à l'un des noms-aire spécifiés dans la clause WITHIN ou contenir un nom-aire valable si ANY AREA a été spécifié, doit être mentionné avant que l'article puisse faire l'objet d'un rangement.

Exemple :

```
Schéma : RECORD A...  
        WITHIN AREA-1, AREA-2, AREA-3, AREA-ID IS PARM-1
```

```
Programme : MOVE "AREA-3" TO PARM-1.  
          STORE A.
```

Au moment de l'exécution, le programme choisit l'aire qui contiendra cet article. Avant son enregistrement, le SGBD vérifie que PARM-1 contient un nom-aire valable. Si le nom-aire n'est pas valable, une erreur est détectée et le message "INVALID REALM" est envoyé. Dès qu'une aire de la clause WITHIN est spécifiée, les règles concernant la plage de clés standard et le mode de localisation s'appliquent.

Lorsque l'on désire extraire un article CALC au moyen d'une instruction "FIND ANY", le nom-aire approprié doit être introduit dans la zone AREA-ID de PARM-1.

7. Si la spécification est AREA OF OWNER, l'article sera rangé dans la même aire que son article maître en fonction des critères de sélection d'ensemble pour l'ensemble spécifié par VIA.
8. L'affectation de l'aire à l'article reste constante quels que soient les changements intervenant dans ses relations avec l'ensemble.

Remarque : Il est conseillé de réserver à chaque paramètre de AREA-ID un nom unique. On évitera ainsi des conflits dans l'utilisation des identificateurs AREA-ID au moment de la sélection d'ensemble.

1.20 CLAUSE CHECK (SOUS-RUBRIQUE ARTICLE)

Fonction

Permet de spécifier un contrôle de validité des valeurs attribuées à des données dans un article.

Format général

```
CHECK IS condition-1
```

Règles de syntaxe

1. Tous les identificateurs-de-données utilisés comme arguments de condition-1 doivent être des données spécifiées dans la rubrique article.

Règles générales

1. La condition est vérifiée chaque fois qu'un article de ce type est enregistré ou que l'un quelconque des arguments de la condition est changé par une instruction MODIFY.
2. Si la condition est fausse, le SGBD notifie l'erreur et aucun changement ne sera apporté à la base de données.

1.21 CLAUSE NOM-DE-DONNEE-UTILISATEUR

Fonction

Permet de désigner une donnée élémentaire, un vecteur, un groupe répétitif ou non répétitif et d'indiquer son niveau de structure dans l'article.

Format général

<code>[n°-niveau] nom-de-donnée-utilisateur</code>
--

Règles de syntaxe

1. Nom-de-donnée-utilisateur doit être unique parmi les noms-de-donnée (de la base de données) déclarés pour cet article et ne peut non plus être un non de rubrique.
2. n°-niveau est un entier décimal non signé compris entre 1 et 99. Sa valeur implicite est 01.

Règles générales

1. Le contenu d'un article se définit par une série (ou aucune) de sous-rubriques données. Lorsqu'un article ne contient aucune sous-rubrique donnée il est utilisé dans la structure pour faciliter l'accès à d'autres articles. On l'appelle alors article "repère" (ROOT) ou "de relation".
2. Les données élémentaires sont décrites par une sous-rubrique donnée contenant une clause TYPE mais aucune clause OCCURS.
3. Un vecteur est décrit par une sous-rubrique donnée contenant une clause TYPE et une clause OCCURS.
4. Un groupe répétitif est décrit par une sous-rubrique donnée contenant une clause OCCURS mais aucune clause TYPE.
5. Un groupe non répétitif est décrit par une sous-rubrique donnée qui ne contient ni clause TYPE ni clause OCCURS.
6. Si une sous rubrique donnée définit un groupe répétitif ou un groupe non répétitif, elle doit être suivie d'une ou plusieurs sous-rubrique donnée portant des numéros de niveau de valeurs plus élevées.

1.22 CLAUSE TYPE

Fonction

Permet de décrire les caractéristiques d'une donnée élémentaire (isolée ou élément d'un vecteur, d'un groupe répétitif ou d'un groupe non répétitif).

Format général

```

TYPE IS { { { [UNSIGNED] { UNPACKED } } } }
        { { { [PACKED] } } } }
        { { { PACKED-2 } } } }
        { { { DECIMAL entier-1 [entier-2] } } } }
        { { { SIGNED { UNPACKED } } } } }
        { { { [PACKED] } } } } }
        { { { [SIGNED] BINARY { 15 } } } } }
        { { { CHARACTER entier-3 { 31 } } } } }

```

Règles de syntaxe

- Entier-1 et entier-3 doivent être des constantes décimales non signées de valeur plus élevée que 0. La valeur maximale varie en fonction du type de donnée et est définie par les règles générales.
- Entier-2 est une constante décimale signée.

Règles générales

- La clause TYPE définit une donnée numérique (décimale ou binaire) ou une chaîne de caractères.
- DECIMAL définit une donnée numérique décimale. Les données DECIMAL correspondent au COBOL PIC 9.
- Entier-1 spécifie le nombre de chiffres significatifs à conserver pour toutes les valeurs de données. La valeur maximale de entier-1 est 30.
- Entier-2, lorsqu'il est spécifié, indique l'échelle décimale de la donnée. Entier-2 peut prendre n'importe quelle valeur comprise entre -29 et 30. Lorsque entier-2 est négatif, la marque décimale implicite est positionnée à droite du chiffre réel le plus à droite. DECIMAL 4, -1 est l'équivalent du format COBOL-74 PICTURE 9999P. Lorsque entier-2 est positif, la marque décimale est positionnée à gauche du chiffre le plus à droite. DECIMAL 4, 1 est l'équivalent du format COBOL-74 PICTURE 9999V9. Lorsque entier-2 n'est pas spécifié ou qu'il est égal à 0 la marque décimale est positionnée immédiatement à droite du dernier chiffre de droite.
- Le total des chiffres fourni par entier-1 et entier-2 ne doit pas, pour des raisons d'affichage des données, dépasser 30.
- Les mots-clé SIGNED/UNSIGNED PACKED/PACKED-2/UNPACKED définissent le format interne de la donnée. Leurs équivalents COBOL sont indiqués dans le Tableau 1-4.

7. BINARY indique une donnée numérique à base binaire. Les spécifications BINARY correspondent aux formats COBOL COMP-1 et COMP-2. Ils contiennent les valeurs d'entiers signés sur 2 octets (15 bits plus le bit de signe) ou sur 4 octets (31 bits plus le signe de bit).
8. CHARACTER indique une chaîne de caractères. Les spécifications CHARACTER correspondent à la famille des formats COBOL PIC X. Une donnée caractère est supposée contenir des caractères du jeu EBCDIC et toutes les comparaisons s'effectuent en utilisant l'ordre de classement EBCDIC.
9. Entier-3 donne la longueur de la chaîne de caractères. Sa valeur théorique peut aller jusqu'à 65535 mais la taille d'article ne peut dépasser la taille de page de l'aire d'accès.
10. On trouvera au Tableau 1-4 un résumé des correspondances existant entre la clause TYPE et son équivalent en COBOL-74.
11. L'alignement de toutes les données s'effectue sur la limite d'octet lorsqu'elles apparaissent dans la rubrique de description d'articles UWA d'un programme DML.

Exemple :

```
RECORD NAME IS nom-article-utilisateur-1...
```

```
02 Donnée-affichage-1 TYPE IS CHARACTER 3.
```

```
02 Donnée-décimale-1 TYPE IS SIGNED PACKED DECIMAL 5, +2.
```

```
02 Donnée-décimale-1 TYPE IS SIGNED BINARY 31.
```

- Remarques :**
1. Donnée-affichage-1 occupe dans l'article 3 octets EBCDIC de 8 bits.
 2. Donnée-décimale-1 occupe 5 positions de décimales condensées plus une position pour le signe, soit un total de 6, ou 3 octets de 8 bits.
La marque décimale implicite est positionnée avant les deux derniers chiffres, soit l'équivalent COBOL-74 :

```
PICTURE S9(3)V99 USAGE COMP.
```

3. Donnée-binaire-1 s'étend sur 4 octets de 8 bits à partir de la limite de l'octet suivant.

Tableau 1-5. Correspondance entre types de données DDL et COBOL

Clause TYPE du DDL			Format COBOL produit	
			PICTURE	USAGE
SIGNED UNPACKED DECIMAL	m		S9(m)	
	m,p	m <p> p m = p m	S9(m-p)V9(p)	
	m,-p		S9(M)P(M)	
m SIGNED PACKED DECIMAL m,-p	m,p		comme SIGNED UNPACKED	COMP
	m		9(m)	
	m,p	m <p> p	9(m-p)V9(p)	
		m = p	V9(m)	
		m	P9(p-m)9(p)	
	m,-p		9(m)P(p)	
m UNSIGNED PACKED DECIMAL m,-p	m,p		comme UNSIGNED UNPACKED	COMP
m UNSIGNED PACKED-2 DECIMAL m,-p	m,p		comme UNSIGNED UNPACKED	COMP-8
SIGNED BINARY 15				COMP-1
SIGNED BINARY 31				COMP-2
CHARACTER n			X(n)	

1.23 CLAUSE OCCURS

Fonction

Permet de définir un vecteur ou un groupe répétitif en spécifiant le nombre de fois où la donnée ou le groupe de données apparaîtra dans un article ou un groupe répétitif.

Format général

<code><u>OCCURS</u> entier TIMES</code>

Règles générales

1. Le nombre de répétition du groupe est spécifié par la valeur (supérieure à 1) de l'entier.
2. Cette clause sert à définir un vecteur conjointement à la clause TYPE :

```
02 EDX TYPE SIGNED BINARY 31 OCCURS 10.
```

ou, peut s'utiliser pour définir un groupe répétitif, si la clause TYPE est omise :

```
02 EDAB OCCURS 18.
```

```
03 EDA TYPE CHARACTER 20.
```

```
03 EDB OCCURS 8.
```

```
04 EDBX...
```

```
04 EDBY...
```

```
04 EDBZ...
```

EDAB et EDB étant les identificateurs de groupe.

1.24 CLAUSE CHECK (SOUS-RUBRIQUE DONNEE)

Fonction

Permet de demander le contrôle de validité qui sera exécuté, soit lorsqu'une valeur est changée, soit lorsqu'une nouvelle donnée est enregistrée.

Format général

```
CHECK IS VALUE [NOT] {littéral-1 [THRU littéral-2]}
```

Règles de syntaxe

1. La sous-rubrique donnée qui contient cette clause CHECK doit également comporter une clause TYPE.
2. Le type de littéral-1 et littéral-2 doit concorder avec le type de la sous-rubrique donnée comme indiqué au Tableau 1-3.
3. En ce qui concerne les chaînes alphanumériques, les littéraux doivent être en ordre ascendant comme indiqué dans la séquence de classement EBCDIC. Pour les données numériques, les littéraux doivent être classés dans l'ordre algébrique ascendant.

Règles générales

1. La valeur de la donnée est comparée aux valeurs individuelles ou aux valeurs de plages indiquées dans cette clause. Une valeur individuelle est indiquée par un littéral. Une plage est indiquée par deux littéraux séparés par THRU.
2. Dans le cas d'un littéral simple, la valeur d'une donnée satisfait aux conditions de comparaison si elle est égale au littéral. Dans le cas d'une plage, la valeur de la donnée doit être supérieure ou égale au littéral qui indique la position inférieure de la plage ou inférieure ou égale au littéral spécifiant la position supérieure de la plage.
3. La valeur est correcte si elle remplit la condition de comparaison, sans NOT, ou si NOT étant spécifié, elle ne la remplit pas.
4. La fonction de vérification de validité est lancée dès qu'une nouvelle donnée est introduite dans la base de données (instruction STORE) ou si une valeur est modifiée (instruction MODIFY).
5. En cas de détection de valeur incorrecte, une erreur est signalée et la base de données n'est pas modifiée.

Exemples de clause CHECK

CHECK IS VALUE "1"

CHECK IS VALUE NOT 1000

CHECK IS VALUE "A" "B" "C"

CHECK IS VALUE 20 THRU 30 50 THRU 60

CHECK IS VALUE NOT THRU 200

1.25 RUBRIQUE SET

Fonction

Permet de désigner les ensembles d'une base de données et de leur attribuer certaines caractéristiques.

Format général de la rubrique ensemble (SET)

```
Sous-rubrique ensemble
{sous-rubrique détail}...
```

Format général de sous-rubrique ensemble

SET NAME IS nom-ensemble

OWNER IS nom-article-1

ORDER IS PERMANENT INSERTION IS

```
{FIRST
{LAST
{NEXT
{PRIOR
{
{ WITHIN RECORD-TYPE
{ BY DEFINED KEYS
{ RECORD-TYPE SEQUENCE IS {nom-article-2}...}
{SORTED
{
{ DUPLICATES ARE {FIRST
{ LAST
{ NOT ALLOWED
}
```


Format général de sous-rubrique détail (MEMBER)

```

MEMBER IS nom-article-3

INSERTION IS {AUTOMATIC RETENTION IS MANDATORY}
             {MANUAL RETENTION IS OPTIONAL}

[ DUPLICATES ARE NOT ALLOWED FOR {data-base-identif-1}... ]...

[ KEY IS {ASCENDING } {identificateur de donnée-2} ]
[ {DESCENDING} {RECORD-TYPE} ]
[ {DATA-BASE-KEY} ]
[ ]
[ [ASCENDING ] {identificateur de donnée-3} ] ]
[ [DESCENDING] {RECORD-TYPE} ] ... ]
[ [DATA-BASE-KEY} ] ] ... ]
[ ]
[ {FIRST } ] ]
[ DUPLICATES ARE {LAST } ] ]
[ {NOT ALLOWED} ] ]
SET SELECTION [FOR nom-ensemble-1] IS

THRU nom-ensemble-2 OWNER IDENTIFIED BY

{APPLICATION }
{DATA-BASE-KEY [EQUAL TO paramètre-base-de-données-1] }
{CALC-KEY [identificateur-de-données-4 EQUAL TO ] }
{ [ ] }
{ [ identificateur-de-données-5] }
{ [ ] ... }
{ [ paramètre-base-de-données-2] }
{ [AREA-ID EQUAL TO paramètre-base-de-données-3] }

THEN THRU nom-ensemble-3 WHERE OWNER IDENTIFIED BY
[ ]
[ [ {identificateur-de-données-7} ] ] ]
[ {identificateur-de-données-6 [EQUAL TO {paramètre-base-de-données-4}]...} ]...

```

1.26 CLAUSE SET

Fonction

Permet de désigner un type d'ensemble dans le schéma; c'est-à-dire de spécifier un nom pour chaque type-ensemble de la base de données.

Format général

```
SET NAME IS nom-ensemble-1
```

Règle de syntaxe

Nom-ensemble-1 doit être unique dans les noms-de-donnée connus du schéma.

Règles générales

Aucune.

1.27 CLAUSE OWNER

Fonction

Permet de spécifier le nom d'un type d'article, dont l'apparition détermine celle d'un type d'ensemble dans la rubrique ensemble.

Format général

<code><u>OWNER</u> IS nom-article-1</code>
--

Règle de syntaxe

1. Nom-article-1 doit avoir été préalablement déclaré dans une rubrique article.
2. Nom-article-1 doit être différent des types d'article détail contenus dans cet ensemble.
3. Un type d'article peut être déclaré maître dans plus d'une rubrique ensemble. De même il peut être déclaré article détail dans une ou plusieurs rubriques ensemble.

Règles générales

1. La clause OWNER spécifie le type d'article qui est maître dans le type d'ensemble décrit. La présence du type d'article maître implique l'existence du type d'ensemble décrit.
2. Il ne peut y avoir qu'un seul article maître dans une apparition d'un type d'ensemble donné.

1.28 CLAUSE ORDER

Fonction

Permet de désigner le point d'insertion d'un article détail dans un ensemble et de définir ainsi l'ordre de pénétration dans la séquence d'articles.

Format général

ORDER IS PERMANENT INSERTION IS

```

{ FIRST
{ LAST
{ NEXT
{ PRIOR
{
{ WITHIN RECORD-TYPE
{
{ BY DEFINED KEYS
{
{ SORTED { [RECORD-TYPE SEQUENCE IS {nom-article-2} ... ] }
{
{ DUPLICATES ARE { FIRST
{ LAST
{ NOT ALLOWED

```

Règles de syntaxe

1. Si la locution SORTED WITHIN RECORD-TYPE est utilisée, la clause KEY doit avoir été spécifiée dans au moins une sous-rubrique détail pour ce type d'ensemble.
2. Si la locution SORTED BY DEFINED KEYS est utilisée, la clause KEY doit avoir été spécifiée pour chaque sous-rubrique-détail de ce type d'ensemble.

Règles générales

1. La spécification ORDER FIRST indique l'insertion sur la position qui suit immédiatement l'article maître. L'ordre chronologique est inversé. C'est le dernier article introduit qui devient le premier article détail de l'ensemble.
2. La spécification ORDER LAST indique l'insertion sur la position qui précède immédiatement l'article maître. L'ordre chronologique est appliqué. C'est l'article détail le plus récent qui devient le dernier de l'ensemble.
3. Les spécifications ORDER PRIOR et ORDER NEXT déterminent le point d'insertion suivant par rapport à l'article détail du dernier ensemble sélectionné par l'unité d'exécution. ORDER PRIOR est l'équivalent de ORDER LAST si l'article concerné est l'article maître.
4. La locution SORTED permet d'introduire un ordre de classement basé sur les valeurs ou les clés des articles détails de l'ensemble, ou sur les valeurs spécifiées dans les clauses KEY pour les articles détails de l'ensemble.

5. La locution WITHIN RECORD-TYPE permet le tri des articles sans tenir compte de l'ordre de classement des autres types d'articles dans l'ensemble. Cela ne signifie pas qu'il existe un classement général par type d'article. C'est simplement l'indication que pour un type donné d'article le critère de classement adopté est celui de sa propre clé de tri. Les clés de tri sont fournies par la clause KEY pour chaque type article détail à trier.
6. La locution DEFINED spécifie que les articles détails d'un ensemble doivent être conservés dans une séquence unique sans tenir compte des différents types d'articles détails spécifiés dans la rubrique ensemble. Les clés de tri correspondantes sont spécifiées dans les clauses KEY pour chaque type d'article détail.
7. La locution facultative RECORD-TYPE SEQUENCE IS permet de déclarer le classement par type d'article de sorte que le type d'article puisse s'utiliser comme clé de tri dans la clause KEY pour les articles détails de l'ensemble.
8. Si la locution DUPLICATES ARE NOT ALLOWED est spécifiée, le SGBD refuse l'insertion dans le type d'ensemble concerné de tous les articles détails ayant la même valeur pour la même clé, l'article étant déjà un article détail de cet ensemble. Cette situation peut se produire lorsque l'on tente d'enregistrer un nouvel article détail dans la base de données, ou que l'on tente d'insérer un article déjà existant dans un ensemble, ou que l'on tente de modifier la valeur d'une clé.
9. Si la locution DUPLICATES ARE FIRST ou DUPLICATES ARE LAST est spécifiée, les articles détails sont insérés dans l'ensemble avant ou après, suivant l'option choisie, les articles détails de l'ensemble ayant des clés identiques.

1.29 CLAUSE MEMBER

Fonction

Permet de désigner un type d'article qui peut figurer comme détail de types d'ensemble spécifiés.

Format général

```
MEMBER IS nom-article-3
```

Règles de syntaxe

1. La clause MEMBER est obligatoire pour chaque type d'article qui peut être article détail du type d'ensemble spécifié.
2. Plusieurs types d'article peuvent figurer comme détails dans un type d'ensemble donné.
3. Un article peut être article détail de plus d'un type d'ensemble. Il peut également être défini comme maître dans un ou plusieurs type d'ensemble.
4. Nom-article-3 doit être différent du type d'article maître de l'ensemble comme indiqué dans la clause OWNER.

Règles générales

1. Chaque article ne peut figurer que comme unité dans chaque apparition du type ensemble pour lequel il a été déclaré détail. Ceci veut dire qu'il ne peut être rattaché à plus d'un article maître dans le type d'ensemble pour lequel il peut être détail. Un article ne peut apparaître qu'une fois dans un ensemble donné.

1.30 CLAUSE INSERTION RETENTION

Fonction

Permet de spécifier le type de la relation dans le type d'ensemble.

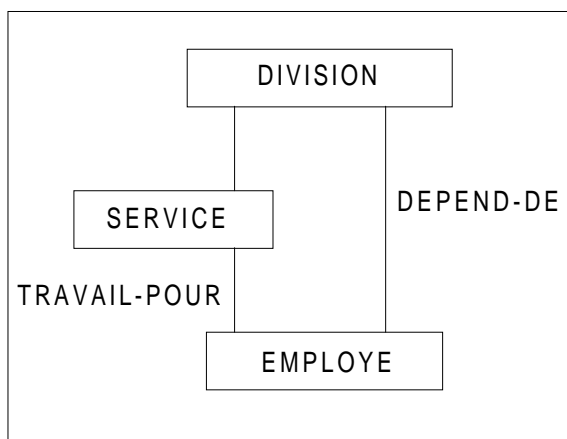
```

INSERTION IS AUTOMATIC RETENTION IS MANDATORY
      MANUAL RETENTION IS OPTIONAL
    
```

Règles générales

1. Si on utilise AUTOMATIC MANDATORY, l'article détail, au moment de son rangement dans la base de données, est introduit dans (et devient donc article détail de) l'ensemble désigné par la rubrique ensemble. Il devient donc article détail de ce type d'ensemble. Il ne pourra devenir article d'un type d'ensemble semblable que par une fonction MODIFY.
2. Si l'option MANUAL OPTIONAL est utilisée, le fait d'ajouter l'article détail article-1 à la base-de-données ne provoque pas l'insertion automatique de l'article comme article détail du type d'ensemble mentionné dans la rubrique ensemble. C'est la fonction DML CONNECT qui permet à l'unité d'exécution d'établir la relation à l'intérieur de l'ensemble. Cette relation peut être annulée au moyen de la fonction DML DISCONNECT. Lorsque la localisation d'article se fait par VIA, l'article est placé physiquement à côté de son article maître dans l'ensemble désigné dans la spécification VIA. Une fois que l'article a été rangé, il peut être inséré dans le ou les ensemble(s) approprié(s) au moyen de l'option MANUAL OPTIONAL.

Exemple :



```

RECORD-NAME IS EMPLOYE
LOCATION MODE IS VIA TRAVAIL-POUR
SET

SET NAME IS DEPEND-DE
OWNER IS DIVISION ...

MEMBER IS EMPLOYE
INSERTION MANUAL
      RETENTION OPTIONAL ...
...
SET-NAME IS TRAVAIL-POUR
OWNER IS SERVICE ...
MEMBER IS EMPLOYE
INSERTION AUTOMATIC RETENTION
      MANDATORY ...
    
```

Langage DDL

Dans l'exemple donné plus haut, lorsqu'un article EMPLOYE est enregistré, il est automatiquement inséré dans l'ensemble TRAVAIL-POUR et placé à côté de son article maître SERVICE. Si l'employé exerce une fonction au niveau d'une division du groupe, il peut être rattaché au moyen de la fonction CONNECT à l'ensemble facultatif DEPEND-DE.

1.31 CLAUSE DUPLICATES

Fonction

Permet la vérification et le refus d'insertion dans le même ensemble des articles détails ayant des valeurs de données identiques.

Format général

```
DUPLICATES ARE NOT ALLOWED FOR {identificateur-de-donnée-1}
```

Règles de syntaxe

1. L'utilisation de plus d'une clause DUPLICATES, correspondant à différents groupes de zones, est autorisée dans la sous-rubrique MEMBER (détail).
2. Les identificateurs-de-données doivent se rapporter à des données incluses dans le type d'article désigné dans la sous-rubrique MEMBER (détail).

Règles générales

1. Si la clause DUPLICATES ARE NOT ALLOWED est utilisée, le SGBD refuse l'insertion dans un ensemble de tout article détail dont les valeurs de données sont identiques aux valeurs contenues dans un article détail de l'ensemble. Cette situation peut se produire lorsqu'on essaye de ranger un nouvel article dans la base de données, ou d'insérer un article existant dans un ensemble, ou de modifier la valeur des données, ce qui aurait pour effet de créer des doubles. Une recherche est effectuée sur l'ensemble pour satisfaire à la règle de non-duplication avant que la clause ORDER INSERTION ne soit traitée. En particulier, si une zone est spécifiée avec INSERTION MODE IS NEXT, la totalité de l'ensemble est vérifiée avant que le nouvel article soit inséré à la suite de l'article courant de l'ensemble.

1.32 CLAUSE KEY

Fonction

Permet de spécifier la clé de tri d'un article détail dans un ensemble. Une clé peut être constituée d'une donnée de l'article, du type d'article ou de la clé base-de-données (DB-KEY) de l'article.

Format général

```

KEY IS { ASCENDING } { identificateur-de-donnée-2 }
      { DESCENDING } { RECORD-TYPE }
                          { DATA-BASE-KEY }

[ [ ASCENDING ] { identificateur-de-donnée-3 } ]
[ [ DESCENDING ] { RECORD-TYPE } ] ...
[ { DATA-BASE-KEY } ]

[ { FIRST } ]
[ DUPLICATES ARE { LAST } ]
[ { NOT ALLOWED } ]

```

Règles de syntaxe

1. La clause KEY doit figurer dans toutes les sous-rubriques détail d'une rubrique ensemble qui comprend la locution SORTED BY DEFINED KEYS dans la clause ORDER.

Elle doit figurer dans au moins une sous-rubrique détail (MEMBER) d'une rubrique ensemble qui comprend la locution SORTED WITHIN RECORD-TYPE dans la clause ORDER.

La clause KEY ne doit pas figurer dans la rubrique ensemble si celle-ci ne comprend pas la locution SORTED dans la clause ORDER.

2. Les identificateurs-de-donnée doivent se rapporter à des données incluses dans le type d'article désigné dans la clause MEMBER.
3. Si RECORD-TYPE est une clé, elle doit figurer une seule fois et RECORD-TYPE SEQUENCE IS... doit également figurer dans la clause ORDER de la rubrique SET. Tous les noms-articles spécifiés doivent appartenir à l'ensemble. Tous les articles détails de l'ensemble doivent avoir été déclarés dans RECORD-TYPE SEQUENCE.
4. Si DATA-BASE-KEY est une clé, elle doit être la dernière et ne doit figurer qu'une fois. La locution DUPLICATES figurant dans la clause ORDER ou la clause KEY doit indiquer NOT ALLOWED.

5. Si la définition d'ensemble comprend l'option SORTED BY DEFINED KEYS dans la clause ORDER, les données correspondantes doivent avoir été spécifiées dans les clauses KEY pour tous les types d'article détail. Les données correspondantes doivent avoir des caractéristiques identiques et être classées dans le même ordre (ASCENDING ou DESCENDING). Les caractéristiques de données sont identiques lorsqu'elles ont été définies par des clauses TYPE identiques. Si RECORD-TYPE est pris comme clé les principes de correspondance bi-univoque précédemment décrits ne s'appliquent qu'aux données situées à gauche de RECORD-TYPE.
6. La phrase DUPLICATES ne peut s'utiliser que si la clause ORDER de la rubrique ensemble contient la phrase SORTED WITHIN RECORD-TYPE.

Règles générales

1. Les identificateurs de donnée sont des clés qui constituent l'ensemble des clés de tri du type d'article détail désigné dans la clause MEMBER. Les clés sont classées, dans la clause KEY, par ordre d'importance décroissante; c'est-à-dire que identificateur-de-donnée-2 est plus significatif que identificateur-de-donnée-3 etc qui sont moins significatif.
2. L'ordre de classement des clés est croissant si ASCENDING figure dans la clause KEY immédiatement avant identificateur-de-donnée. Il est décroissant si DESCENDING est intercalé entre ASCENDING et l'id-de-donnée. Les articles détails d'un ensemble sont donc triés, soit de la plus petite valeur de clé à la plus élevée (ASCENDING), ou l'inverse (DESCENDING).
3. Si DUPLICATES ARE NOT ALLOWED est spécifié, le SGBD refuse l'insertion dans un ensemble de tous les articles détails de type identique et ayant les mêmes valeurs de clé car un tel article détail existe déjà dans l'ensemble. Cette situation peut se produire lorsque l'on tente d'introduire un nouvel article dans une base de données, ou que l'on tente d'insérer un article existant dans un ensemble ou encore lorsque l'on tente de modifier la valeur d'une clé.
4. Suivant l'option choisie, DUPLICATES ARE FIRST ou DUPLICATES ARE LAST, l'insertion des articles détails dans l'ensemble se fait avant ou après les articles détails de l'ensemble qui sont du même type et ont les mêmes valeurs de clés.

1.33 CLAUSE SELECTION

Fonction

Permet de définir les critères de sélection d'un type d'ensemble afin d'y insérer un article détail ou d'accéder à cet article.

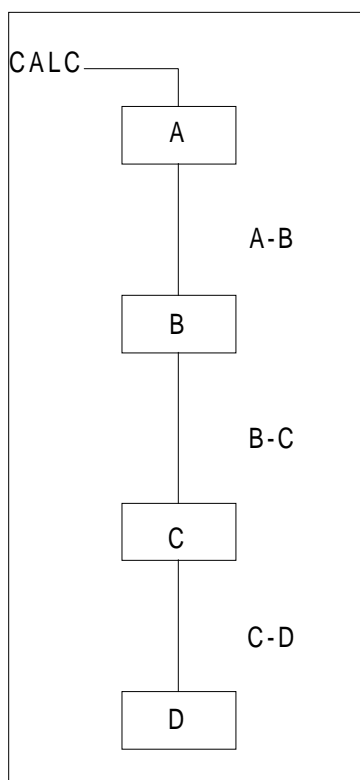
Format général

```
SET SELECTION [FOR nom-ensemble-1] IS
  THRU nom-ensemble-2 OWNER IDENTIFIED BY
  { APPLICATION }
  { DATA-BASE-KEY [EQUAL TO paramètre-base-de-données-1] }
  { CALC-KEY [identificateur-de-donnée-4 ]
    [ EQUAL TO {identificateur-de-donnée-5} ]
    [ {paramètre-base-de-données-2} ] ... }
  { [AREA-ID EQUAL TO paramètre-base-de-données-3] }
  [ THEN THRU nom-ensemble-3 WHERE OWNER IDENTIFIED BY ]
  [ {identificateur-de-donnée-6 [EQUAL TO {identificateur-de-donnée-7 }]} ... ] ...
  [ { {paramètre-base-de-données-4} } ] ]
```

Règles de syntaxe

1. Nom-ensemble-1 est le nom du type d'ensemble de la rubrique ensemble concernée.
2. Si l'option CALC-KEY est choisie, l'article maître désigné dans nom-ensemble-2 doit avoir un mode de localisation CALC avec DUPLICATES NOT. Identificateur-de-donnée-5 doit avoir des caractéristiques identiques à identificateur-de-donnée-4 de CALC-KEY comme indiqué dans la clause LOCATION. Si la locution EQUAL TO est utilisée pour désigner un élément d'une clé CALC multi-élément, elle doit figurer pour chacun d'entre eux.
3. Identificateur-de-donnée-6 est une donnée élémentaire de l'article maître de nom-ensemble-3.
4. L'identificateur-de-donnée-7 doit avoir des caractéristiques identiques à celles de identificateur-de-donnée-6.
5. Nom-ensemble-2, nom-ensemble-3,... doivent former une chaîne continue en ce sens que l'article maître de nom-ensemble-3 est un article détail de nom-ensemble-2..., nom-ensemble-2 étant considéré comme point d'entrée ou repère. Un même ensemble ne doit pas être répété dans cette chaîne. Le dernier type d'ensemble mentionné doit appartenir à la rubrique ensemble dont cette clause fait partie.

Exemple :



SET SELECTION FOR C-D IS THRU A-B OWNER IDENTIFIED BY CALC-KEY, THEN THRU B-C WHERE OWNER IS IDENTIFIED BY FIELD-IN-B, THEN THRU C-D WHERE OWNER IS IDENTIFIED BY FIELD-IN-C.

6. Les données élémentaires fournies par identificateur-de-donnée-6 doivent servir à identifier de façon unique l'article maître de nom-ensemble-3 parmi les articles détails de nom-ensemble-2. Ces données élémentaires, considérées comme un groupe, doivent être l'objet d'une locution DUPLICATES ARE NOT ALLOWED, soit dans la clause ORDER pour nom-ensemble-2, soit dans la clause KEY, soit dans une clause DUPLICATES pour l'article maître de nom-ensemble-3, article détail de nom-ensemble-2.
7. Le mode de localisation spécifié pour l'article maître de nom-ensemble-2 doit être DIRECT ou CALC en fonction de l'option DATA BASE KEY ou CALC KEY choisie respectivement.

Règles générales

1. Les caractéristiques de paramètre-base-de-données-1 doivent permettre de contenir une valeur clé-base-de-données.
2. Les caractéristiques de paramètre-base-de-données-2 sont identiques à celles de la clé CALC, comme indiqué dans la clause LOCATION de la rubrique article définissant l'article maître de nom-ensemble-2.
3. Les caractéristiques de paramètre-base-de-données-3 doivent permettre de contenir un nom d'aire.

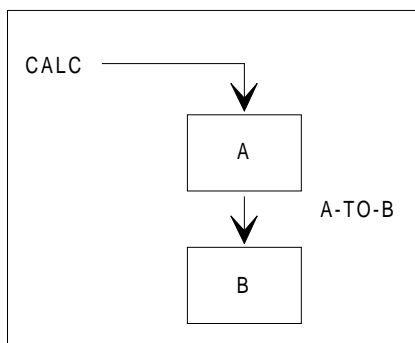
Langage DDL

4. Les caractéristiques du paramètre id-base-de-données-4 sont identiques à celles de id-base-de-données-6.
5. La clause SELECTION appliquée conjointement au type d'article détail et au type d'ensemble, détermine la sélection d'un ensemble donné afin d'y insérer un article détail ou d'accéder à cet article. C'est le choix d'un article maître spécifique qui détermine l'identification de l'ensemble.
6. Avant l'exécution de toute fonction DML impliquant une sélection, des valeurs doivent être attribuées aux paramètres et/ou aux données élémentaires de la zone UWA spécifiés dans la ou les locutions EQUAL TO si celle-ci est utilisée.
7. L'option APPLICATION provoque la sélection dans nom-ensemble-2 de l'article courant indiqué par "article-courant-de-nom-ensemble-2". L'article courant d'un ensemble peut être le maître ou un article détail.
8. L'option DATA-BASE-KEY provoque la sélection, dans nom-ensemble-2, de l'article dont le maître possède un DB-K de valeur égale à paramètre-base-de-données-1. En l'absence de la locution EQUAL TO, c'est l'article maître spécifié dans LOCATION MODE IS DIRECT qui est sélectionné.
9. L'option CALC KEY provoque la sélection dans nom-ensemble-2 de l'article maître :
 - Dont la clé CALC équivaut aux données élémentaires de la zone UWA, ou celui désigné par le paramètre de la locution EQUAL TO. En l'absence de EQUAL TO, c'est celui qui correspond aux données de la zone UWA définies dans la clause LOCATION MODE IS CALC qui est sélectionné.
 - Figurant, dans le cas d'un article multi-zone, dans la zone indiquée par AREA-ID EQUAL TO. En l'absence de cette locution, c'est celui qui est indiqué par le paramètre AREA-ID de la clause WITHIN qui est sélectionné.
10. Lorsque seule la locution THRU nom-ensemble-2 est spécifiée, le nom-ensemble-1 est sélectionné comme indiqué plus haut dans les règles générales, points 7 à 9. Par exemple, pour l'ensemble A-B, la clause s'écrit :

```
SET SELECTION THRU A-B OWNER IDENTIFIED BY CALC-KEY.
```
11. Lorsque la locution THRU nom-ensemble-2 est complétée par une ou plusieurs THEN THRU, la sélection s'opère selon les critères définis dans les règles générales, points 7 à 9. Pour chaque ensemble suivant, la sélection s'opère à l'aide de THEN THRU, à partir de son article maître, nom-ensemble-3, etc, lui-même détail de l'ensemble précédent. Cet article maître est représenté, soit par les valeurs de identificateur-de-donnée-6, elles-mêmes égales aux valeurs de données élémentaires de la zone UWA, soit par les paramètres de EQUAL TO. En l'absence de EQUAL TO, l'identificateur est fourni par les données UWA de identificateur-de-donnée-6.

Exemples :

1.



Si le mode de localisation de A est CALC, la clause SELECTION s'écrit :

```
SELECTION THRU A-TO-B OWNER IDENTIFIED BY APPLICATION
```

et le programme sélectionne un ensemble A-TO-B avant toute exécution d'une STORE (ou FIND), portant sur B. La position de l'article courant de l'ensemble (A-TO-B) s'utilise pour identifier l'ensemble. Cet article peut être le maître ou un détail. L'insertion du nouvel article avec NEXT et PRIOR se fera par rapport à l'article courant de l'ensemble (point courant), soit après (NEXT), soit avant (PRIOR). L'insertion du nouvel article avec FIRST et LAST se fera par rapport à l'ensemble dont A est article maître, soit en amont de l'ensemble A-TO-B (FIRST), soit en aval (LAST).

L'insertion déterminée par la locution SORTED dépend de l'article courant de l'ensemble :

- Si c'est le maître, les comparaisons de clés s'effectuent dans l'ensemble en prenant pour base celle de l'article maître.
- Si c'est un article détail, la clé de l'article détail à insérer est comparée à celle de l'article courant. Si l'article détail doit être insérée après l'article courant, la comparaison des clés se fait en prenant pour base celle de l'article courant. Si le nouvel article détail doit être inséré entre l'article maître et l'article courant, les comparaisons de clés s'effectuent à partir de la clé de l'article maître.
- Lorsque l'ordre de classement de l'ensemble est déterminé par SORTED WITHIN RECORD TYPE, si le nouvel article détail et l'article courant de l'ensemble ne sont pas du même type, la comparaison des clés s'effectue en prenant pour base la valeur de celle de l'article maître.

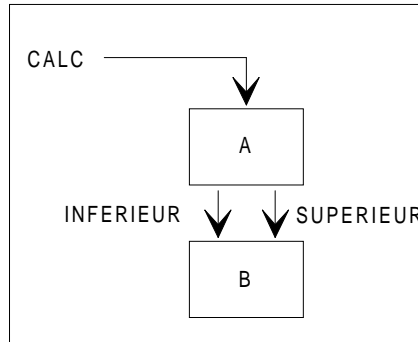
2. Si la clause de sélection est écrite :

```
SELECTION THRU A-TO-B OWNER IDENTIFIED BY CALC-KEY.
```

l'article maître A est toujours localisé et c'est l'ordre d'insertion spécifié qui est utilisé. Dans tous les cas l'opération de sélection commencera par l'article maître A.

3. Utilisation de EQUAL TO.

Dans le cas d'une structure de réseau où les différentes apparitions de l'article A sont en relation avec l'article B, A étant identifié par l'une de ses zones (DIRECT ou CALC) le problème existe d'identifier en réalité deux articles A qui serviront de base d'accès pour le rattachement de B à deux ensembles distincts. Le schéma se présente comme suit :



```
SET NAME IS INFERIEUR...
```

```
MEMBER IS B...
```

```
SELECTION THRU INFERIEUR OWNER IDENTIFIED BY CALC-KEY.
```

```
SET NAME IS SUPERIEUR...
```

```
MEMBER IS B...
```

```
SELECTION THRU SUPERIEUR
```

```
OWNER IDENTIFIED BY CALC-KEY MYKF EQUAL TO PARM-1.
```

De manière à localiser l'article A qui convient, la (ou les) zone(s) CALC spécifiée(s) dans la clause LOCATION MODE de A doivent être initialisées pour identifier l'article A. Dans l'ensemble d'articles INFERIEUR, ce processus est suffisant pour satisfaire les critères de sélection concernant cet ensemble. En ce qui concerne l'ensemble SUPERIEUR, si la clause de sélection est la même, il y aura conflit car la même zone CALC servirait à identifier la seconde apparition de A. La locution EQUAL TO permet d'utiliser un paramètre-base-de-données comme zone de rangement temporaire de la clé CALC. Dès qu'un nouvel article B est inséré dans l'ensemble INFERIEUR, l'ensemble SUPERIEUR peut identifier son propre article maître d'après la valeur de la zone sauvegardée dans PARM-1.

2. Langage DMCL

2.1 VUE DE LA MEMOIRE

Entre la vue du schéma, composée d'articles, d'ensembles et d'aires, et les fichiers disques qui représentent le rangement final de la base de données, il existe une vue intermédiaire, la vue de la mémoire, composée d'articles mémoire et d'aires mémoire (voir Figure 2-1).

Un article mémoire dispose des attributs habituels d'un article.

Une aire mémoire est un groupe de n pages qui forment un conteneur d'articles mémoire. Une aire correspond à un fichier UFAS.

Figure 2-1. Vue de la mémoire entre vue du schéma et supports

2.1.1 Langage DMCL

Le sigle DMCL signifie Device/Media Control Language; ou langage de gestion des supports, ou encore langage DMCL. Il a été rebaptisé DSDL (Data Storage Description Language) ou langage de description du rangement des données dans les propositions CODASYL car il ne se rapporte plus directement aux appareils et supports. Cependant, pour des raisons de compatibilité et aussi parce que certaines caractéristiques des appareils et supports peuvent influencer le choix de certains paramètres, le terme "Langage DMCL" a été conservé dans le présent manuel.

Le but du DMCL est double :

1. Sélectionner des options pour la projection du schéma dans la mémoire. Dans la version en cours, les règles de projection (mapping) sont imposées par IDS/II :
 - chaque aire du schéma correspond à une aire mémoire.
 - Chaque article du schéma correspond à un article mémoire.
 - Chaque ensemble du schéma est indiqué par des pointeurs disposés dans les articles mémoire de l'ensemble (assemblage chaîné).
2. Fournir une information quantitative sur la base de données, information que le DDL ne donne pas. Plutôt que de spécifier le nombre maximum d'apparitions de chaque type d'article, la base de données fournit les paramètres de taille globale des conteneurs, à savoir des aires mémoire.

D'autres paramètres peuvent être fournis, par exemple pour restreindre l'implantation des articles mémoire à certaines plages des aires mémoire, ou pour permettre la migration des articles, ou pour indiquer les besoins en espace tampon implicite pour l'exécution, etc.

2.1.2 Fonction PREALLOC

La fonction PREALLOC permet la projection (mapping) d'une aire mémoire, autrement dit d'un fichier UFAS sur un ou plusieurs espace(s) fichier(s) disque(s).

Ce sujet sera développé au chapitre 4 de ce manuel.

2.1.3 Indépendance Programme-Données

La Figure 2-2 permet d'illustrer ce que l'on entend par indépendance du programme vis-à-vis des données.

- Il n'existe aucune indépendance entre le programme et la vue du schéma car la logique du programme s'adapte aux objets décrits dans cette vue.
- Il existe une indépendance entre le programme et la vue de mémoire, car il est possible de spécifier plusieurs vues de la mémoire par rapport à la même vue du schéma, sans que le programme en soit informé. Par exemple, le même programme peut aussi bien traiter la base de données production ou d'une base de données test plus réduite.

Dans certaines circonstances telles que le calcul et l'interprétation des clés-base-de-données, le programme doit prendre connaissance des paramètres DMCL. Une extension de la fonction ACCEPT permet au programme de lire cette information et de s'adapter ainsi automatiquement à la version en cours.

En ce qui concerne la mise en oeuvre, vue du schéma et vue de la mémoire sont contenues dans le même schéma résultant. Si une seconde vue de la mémoire est nécessaire, il faut donc dupliquer la vue du schéma sous forme d'un second schéma résultant (commande MOVE de DDLPROC).

- Il existe une indépendance entre le programme et les appareils/supports, car il est possible de ranger plusieurs représentations de la base de données dans la même vue de la mémoire, sans que le programme en soit informé. C'est le cas notamment lorsque chaque programmeur utilise sa propre copie de la même base de données de test.
- La liaison entre programme et base de données se fait au moment de l'exécution en sélectionnant, par ordre JCL ASSIGN, le schéma résultant et les aires mémoire nécessaires.

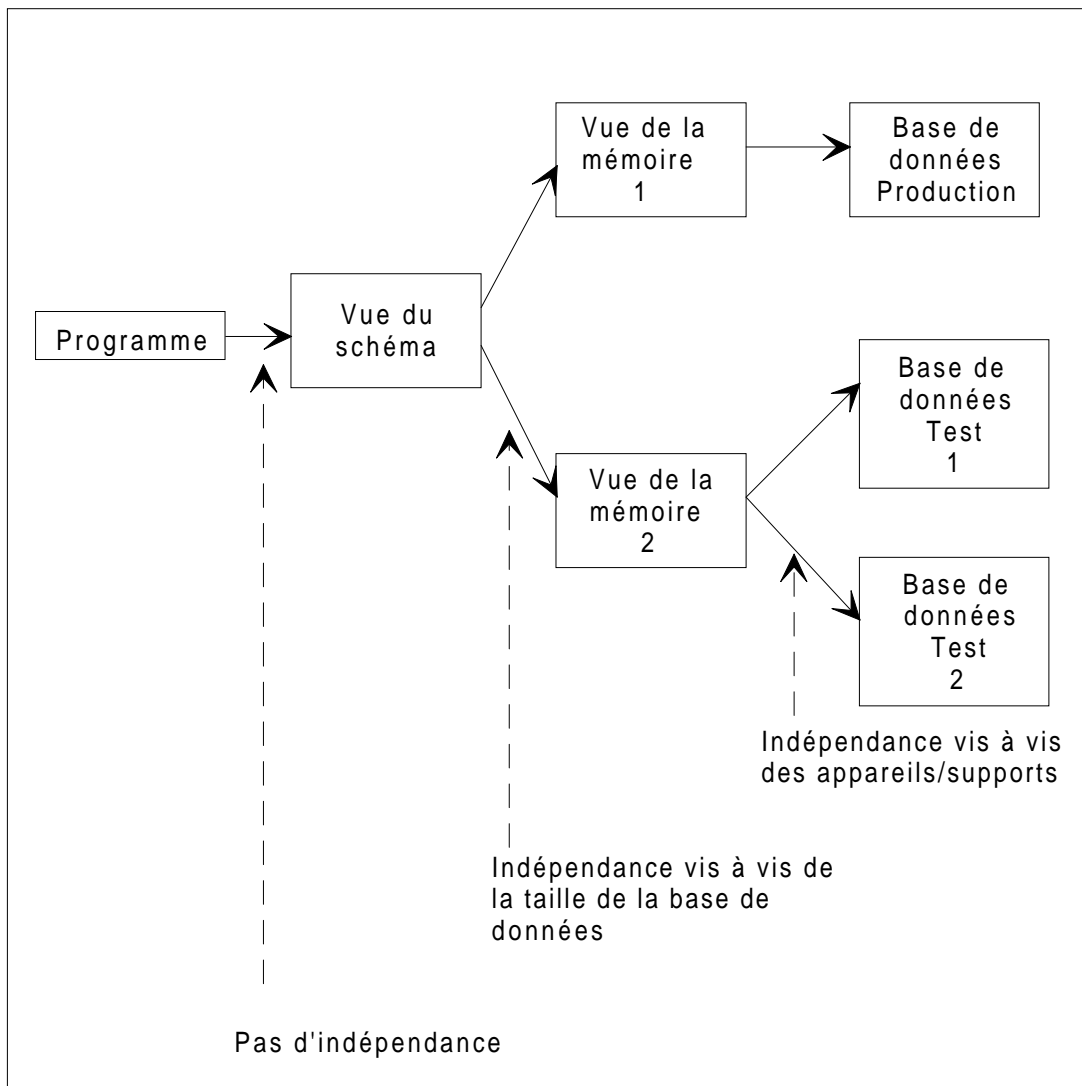


Figure 2-2. Indépendance Programme-Données

2.2 STRUCTURE DE L'AIRE MEMOIRE

2.2.1 Numéro d'aire

La base de données est composée d'un nombre NUMA d'aires mémoire. Chaque aire mémoire est identifiée par un numéro "a" d'aire mémoire, commençant à zéro.

$$0 \leq a \leq \text{NUMA} - 1$$

Les numéros d'aires sont affectés à la suite, dans l'ordre des rubriques aire du DDL de schéma.

2.2.2 Caractéristiques des aires

- Chaque aire est divisée en un certain nombre NUMP de pages d'égales dimensions (voir Figure 2-3). La page constitue l'unité de transfert entre le disque et les tampons du SGBD.

La limite supérieure de NUMP est $(2^{24}) - 1 = 16,777,215$.

L'identification de la page à l'intérieur de l'aire se fait par un numéro "p" de page, en commençant à zéro :

$$0 \leq p \leq \text{NUMP} - 1$$

Les pages sont disposées séquentiellement sur disque dans l'ordre de leurs numéros de page. Les programmes peuvent bénéficier de cette disposition pour optimiser le nombre de mouvements des têtes en traitant l'aire d'une façon séquentielle.

NUMP est spécifié dans la clause NUMBER-OF-PAGES de la rubrique aire du DMCL. La taille de page est spécifiée dans la clause PAGE-SIZE de la même rubrique.

- Chaque page est divisée en un certain nombre LPP de lignes (voir Figure 2-3). La ligne constitue le conteneur d'article.

Le nombre maximum de lignes exprimé par LPP est 255.

L'identification de chaque ligne dans la page se fait par un numéro "l" de lignes, en commençant par zéro :

$$0 \leq l \leq LPP - 1$$

La taille d'une ligne varie en fonction du type d'article qu'elle contient. Un article ne peut s'étendre sur plusieurs pages.

LPP (nombre de lignes par page) est spécifié dans la clause LINES-PER-PAGE de la rubrique aire DMCL.

- Une aire mémoire contenant des articles CALC est aussi caractérisée par la taille des compartiments qui ont déterminé l'algorithme de composition. Ces tailles s'expriment sous forme d'un nombre de pages (255 maxi) dans la clause CALC-INTERVAL de la rubrique aire.
- Le nombre de pages, la taille de page, le nombre de lignes par page et l'intervalle CALC peuvent varier d'une aire à l'autre.

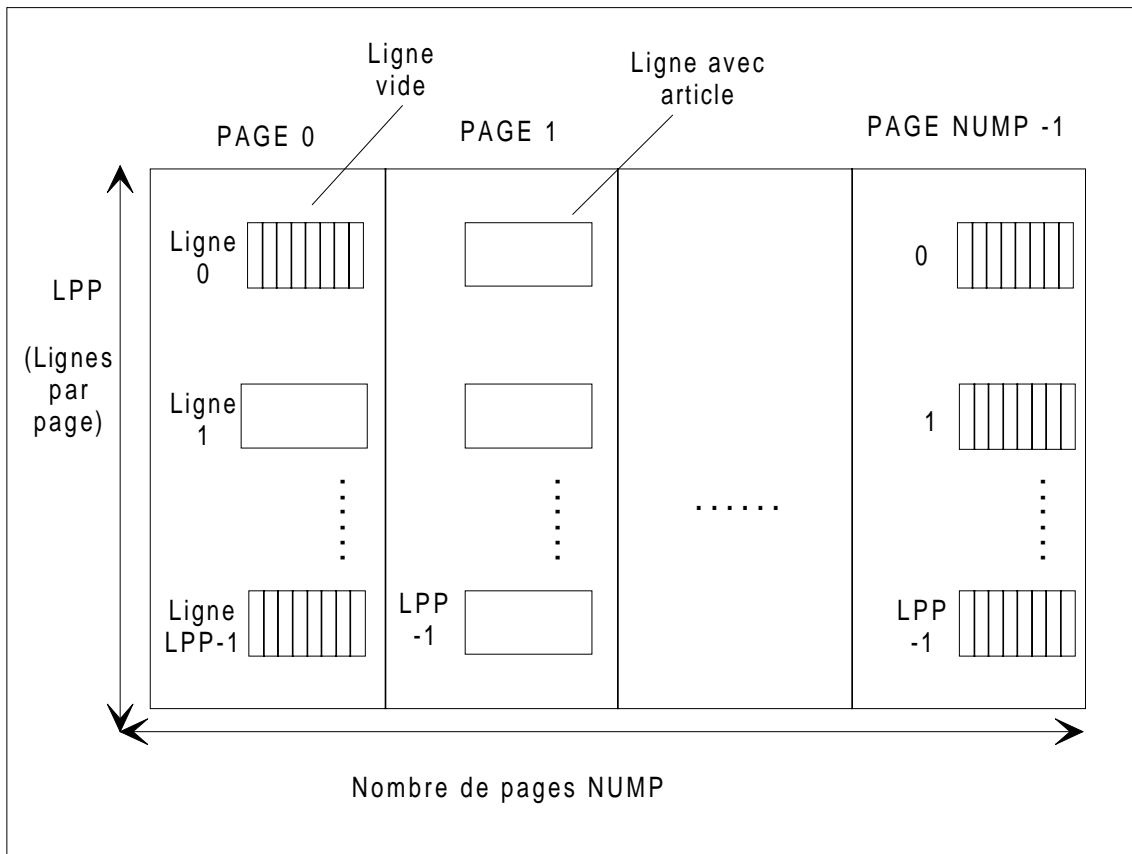


Figure 2-3. Découpage de l'aire mémoire

2.2.3 Structure de page

La structure d'une page est représentée dans la Figure 2-4. La page comprend :

- Un en-tête de page, d'une longueur de 12 octets, qui contient l'information sur l'espace utilisé couramment, l'espace occupé par les articles logiquement supprimés, le nombre le plus élevé de lignes utilisées, etc.
- Un tableau de longueur variable de localisateurs d'article mémoire, indexé par numéro de ligne à partir de la fin de page. Chaque localisateur contient le décalage à deux octets (par rapport à la fin d'en-tête de page) de l'article mémoire correspondant à ce numéro de ligne. La taille maximum du tableau de localisateur est $2 * LPP$.
- Des articles mémoire composés d'un en-tête, d'une zone pointeurs et d'une zone données. Ces articles sont classés par numéro de ligne. Ils peuvent être actifs ou logiquement supprimés. La suppression logique d'article permet au système de gestion SGBD de retarder la réorganisation de la page jusqu'à ce que le tassement devienne nécessaire pour enregistrer un nouvel article dans la page.
- Des articles en débordement (ou articles OWL) de chaîne CALC, occupant un emplacement de dix octets, utilisés dans la mise en oeuvre de la chaîne CALC. Ces articles ne sont pas référencés par numéro de ligne. Un tel article existe toujours. Il est disposé immédiatement après l'en-tête de page, même si l'aire mémoire ne contient aucun article CALC.

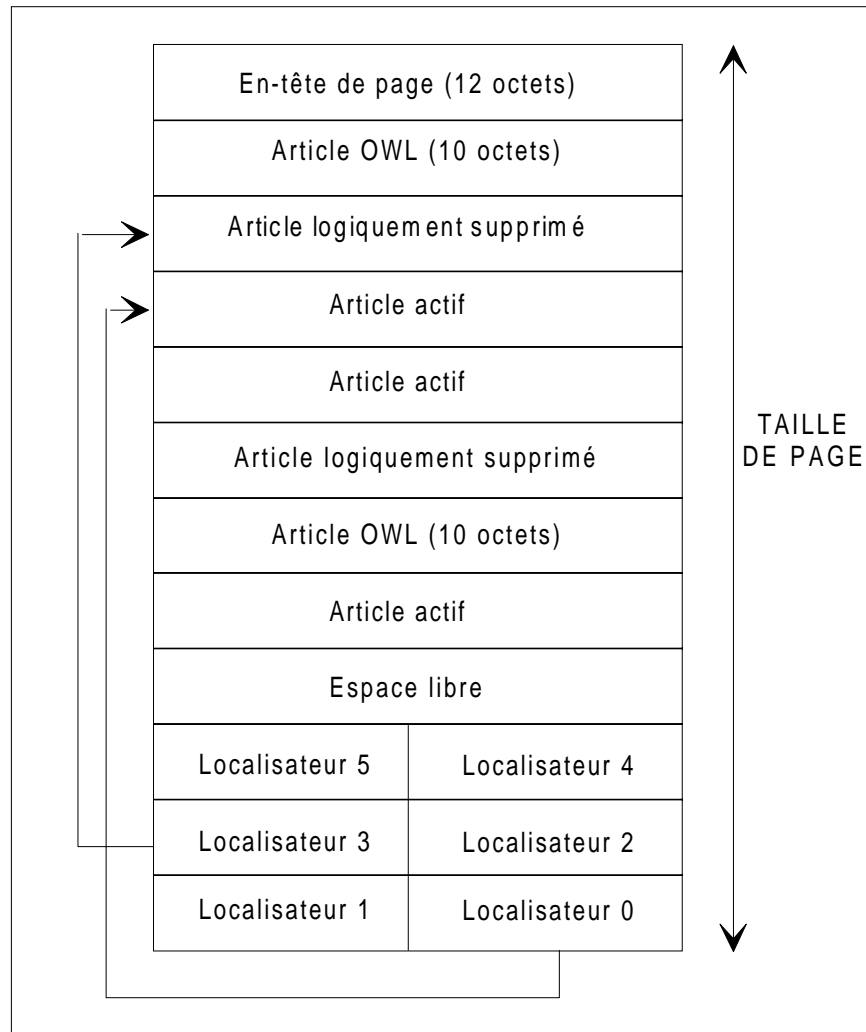


Figure 2-4. Structure de page

2.3 ADRESSAGE D'ARTICLE MEMOIRE

2.3.1 Définition

- Lorsqu'un article mémoire est rangé dans une aire, ou lorsqu'il est l'objet d'une migration à l'intérieur de cette aire à la suite d'une instruction MODIFY CALC-KEY, une adresse logique lui est affectée appelée "clé d'aire" qui permet de l'identifier à l'intérieur de cette aire. Les éléments composant la clé d'aire sont :

numéro-page p
éléments de la clé-aire ak: numéro-ligne l

- Cette adresse logique, purement locale à l'aire mémoire, donne naissance, par addition du numéro d'aire à une autre adresse logique appelée clé-base-de-données (parfois appelée clé IDS), qui identifie de façon unique l'article dans la base de données.

numéro-aire
éléments de la clé-base-de-données abk: numéro-page
numéro-ligne

- Lorsque la clé-aire est utilisée dans le "chaînage d'ensemble", elle est représentée sous la forme d'un pointeur local dont la valeur numérique est calculée par la formule suivante :

$ak = p * LPP + l$

dans laquelle LPP est le nombre de lignes par page de l'aire.

De façon à optimiser l'espace disque, un pointeur local est disposé sur un minimum d'octets: 2, 3 ou 4 (voir Figure 2-5). La taille d'un pointeur local peut varier d'une aire à l'autre.

- Lorsque la clé-base-de-données est utilisée dans le "chaînage d'ensemble", elle est représentée par un pointeur global obtenu par addition du numéro d'aire et de la valeur ak de la clé d'aire.

De façon à optimiser l'espace sur disque, un pointeur global doit être disposé sur un nombre d'octets minimum: 2, 3 ou 4 (voir Figure 2-5). La zone du numéro d'aire est alignée à gauche sauf pour le pointeur à 4 octets, dans lequel le bit le plus à gauche est réservé au signe positif d'un élément du langage COBOL appelé DB-KEY, comme indiqué plus loin. La taille du pointeur global est constante pour toutes les aires.

- Lorsqu'une clé-base-de-données est représentée dans les données COBOL déclarées par USAGE DB-KEY (équivalent au format COBOL USAGE COMP-2), elle est égale à la valeur du pointeur global aligné à droite, avec remplissage de 0 en cas d'utilisation de pointeur à 3 ou 2 octets (voir Figure 2-5).

Un élément DB-KEY peut être spécifié avec tout verbe COBOL (MOVE, COMPUTE ...) qui manipule des zones COMP-2.

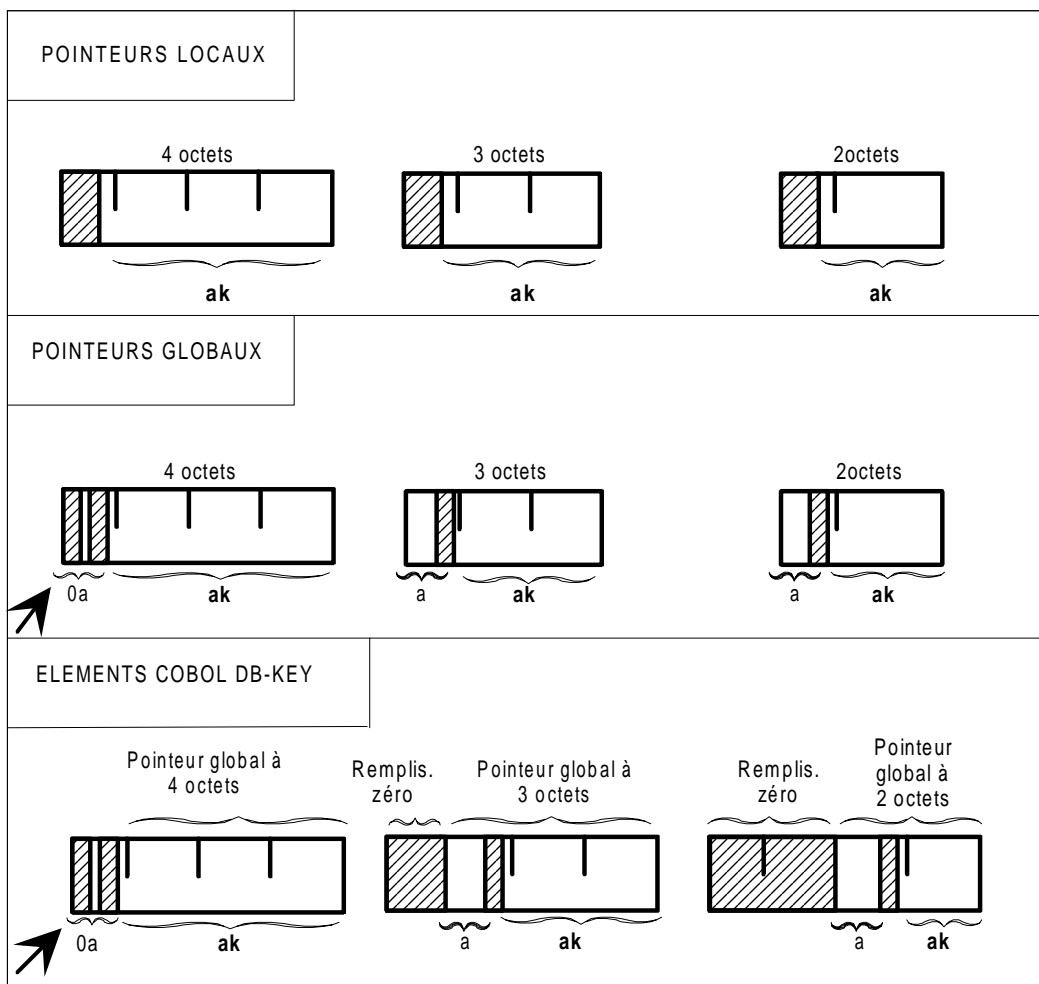


Figure 2-5. Structure des pointeurs locaux, globaux et des éléments COBOL DB-KEY

2.3.2 Taille de pointeur local

- LPP et NUMP étant respectivement le nombre de lignes par page et le nombre de pages d'aire, le nombre de clés d'aire disponibles dans cette aire est représenté par NUMAK suivant la formule :

$$\text{>NUMAK} = \text{LPP} * \text{NUMP}$$

BL représente le nombre de bits nécessaires pour exprimer la valeur la plus élevée des clés d'aire, NUMAK-1, qui est un nombre décimal non signé calculé en utilisant la formule :

$$2^{\text{BL}-1} < \text{NUMAK} \leq 2^{\text{BL}}$$

La taille du pointeur local, en octets, sera donc :

$$\begin{aligned} &2, \text{ si } \text{BL} \leq 16 \\ &3, \text{ si } 16 < \text{BL} \leq 24 \\ &4, \text{ si } 24 < \text{BL} \leq 30 \end{aligned}$$

L'ensemble des valeurs est indiqué ci-dessous :

Tableau 2-1. Taille de pointeur local

Plages de valeurs NUMAK	Nombre de bits (BL)	Octets utilisés	pour le pointeur local
9-16	4	1	2 octets sont effectivement utilisés
17-32	5	1	
33-64	6	1	
63-128	7	1	
129-256	8	1	
257-512	9	2	
513-1024	10	2	
1025-2048	11	2	
2049-4096	12	2	
4097-8192	13	2	
8193-16384	14	2	
16385-32768	15	2	
32769-65536	16	2	
65537-131072	17	3	
131073-262144	18	3	
262145-524288	19	3	
524289-1048576	20	3	
1048577-2097152	21	3	
2097153-4194304	22	3	
4194305-8388608	23	3	
8388609-16777216	24	3	
16777217-33554432	25	4	

- La taille de pointeur local peut varier d'une aire à l'autre car LPP et NUMP dépendent de l'aire elle-même.

- Une fois calculée, la taille de pointeur local sera choisie comme valeur implicite par DDLPROC. Toutefois, la clause EXTEND LOCAL POINTERS de la rubrique aire du DMCL permet à l'administrateur de base de données de sélectionner éventuellement une taille supérieure à celle qui est nécessaire. (Voir le paragraphe intitulé "Ensembles locaux").

2.3.3 Taille de pointeur global

- NUMA étant le nombre d'aires mémoire de la base de données, le nombre BLA de bits nécessaires pour exprimer le nombre le plus élevé d'aires NUMA-1 est un nombre binaire non signé calculé par la formule :

$$2^{BLA-1} < NUMA \leq 2^{BLA}$$

Les valeurs sont indiquées ci-dessous :

Tableau 2-2. Taille de pointeur global

Nombre d'octets	Nombre de bits (BLA)
1-2	1
3-4	2
5-8	3
9-16	4
17-32...	5...

- Le nombre BLM de bits représentant le pointeur local le plus grand, le nombre total BLG de bits nécessaires à un pointeur global est :

$$BLG = BLM + BLA$$

La taille du pointeur global en octets sera donc :

- 2, si $BLG \leq 16$
- 3, si $16 < BLG \leq 24$
- 4, si $24 < BLG \leq 31$

- Lorsque BLG n'est pas exactement égal à 16, 24 ou 31, la zone du numéro d'aire est alignée à gauche dans les limites d'octets demandées, laissant ainsi une zone de bits plus grande pour la zone clé-aire. Dans le cas des pointeurs à 4 octets, le bit de gauche de poids fort est toujours réservé et mis à zéro.

- La capacité maximale d'adressage des pointeurs globaux est indiquée ci-dessous :

Tableau 2-3. Adressage par pointeur global

Taille de pointeur	Nombre d'articles
4 octets (-1 bit)	2 147 483 648
3 octets	26 777 216
2 octets	65 536

- La taille de pointeur global est la même pour toutes les aires de la base de données.
- Une fois calculée, la taille de pointeur global constitue la valeur implicite choisie par DDLPROC. Toutefois, l'administrateur de la base de données, s'il envisage une extension de la base de données, peut choisir l'une des options DMCL suivante :
 - La clause `EXTEND GLOBAL POINTERS` de la rubrique schéma permet d'augmenter la taille de pointeur global. La zone de numéro d'aire est alignée à gauche entre les limites du nombre d'octets spécifié.
 - La clause `EXTEND NUMBER OF AREAS` de la rubrique schéma permet d'augmenter le nombre de bits réservés pour la zone de numéro d'aire des pointeurs globaux.
- La Figure 2-6 ci-dessus illustre le format et la taille des pointeurs locaux et globaux d'une base de données comportant trois tailles d'aire de valeurs différentes.

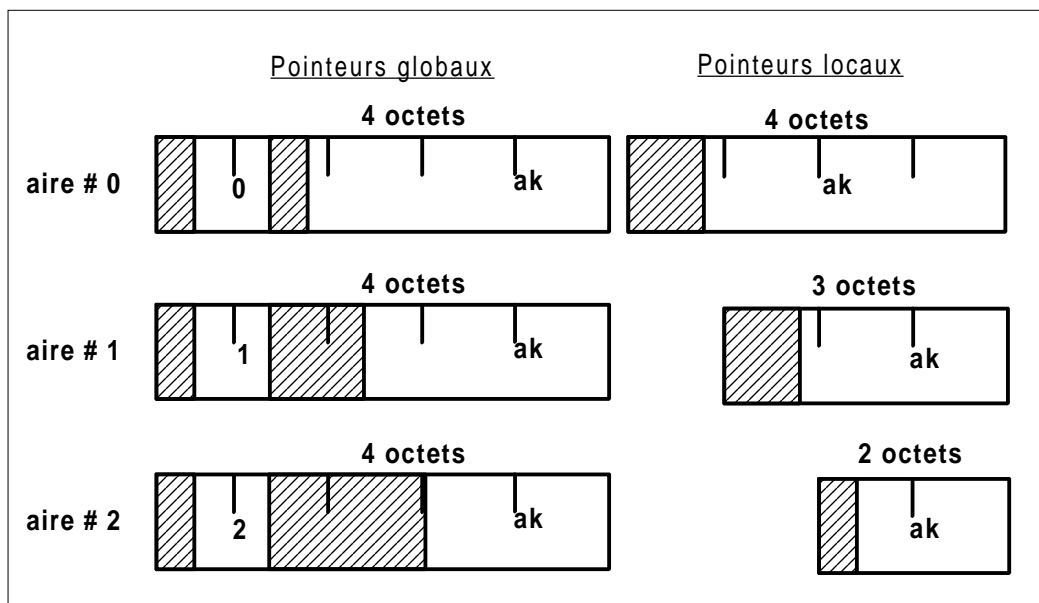


Figure 2-6. Pointeurs locaux et globaux dans des aires de tailles différentes

2.4 MISE EN OEUVRE D'UN ENSEMBLE PAR CHAINE DE POINTEURS

2.4.1 Numéro d'ensemble

Chaque type d'ensemble DDL est identifié d'une façon interne par un numéro "s" d'ensemble, commençant à 1 (s=0 est réservé au système).

Les numéros d'ensemble sont affectés séquentiellement dans l'ordre des rubriques ensemble du DDL de schéma.

Le numéro d'ensemble lui-même n'est pas enregistré dans des articles mémoire mais il influe sur la disposition des pointeurs d'ensemble imbriqués dans les articles mémoire (se reporter au paragraphe "Structure d'article mémoire").

2.4.2 Nombre de pointeurs

Lorsqu'un article mémoire est maître d'un ensemble il contient deux pointeurs, NEXT et PRIOR.

Lorsqu'un article mémoire est article détail de l'ensemble, il contient 3 pointeurs, NEXT, PRIOR et OWNER.

2.4.3 Ensembles locaux

Un ensemble est dit local si les déclarations DDL signifient que le maître et les détails d'un ensemble sont situés dans la même aire. Il existe deux variantes qui sont illustrées par les Figures 2-7 et 2-8.

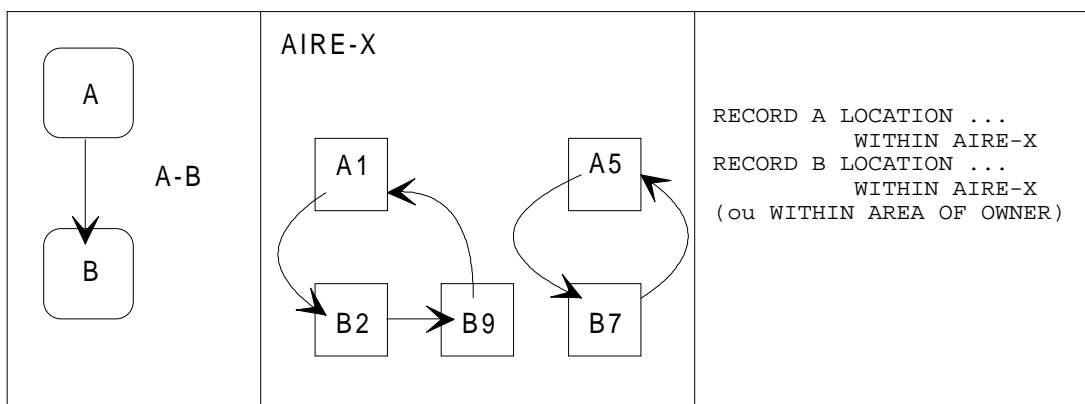


Figure 2-7. Ensemble local mono-aire

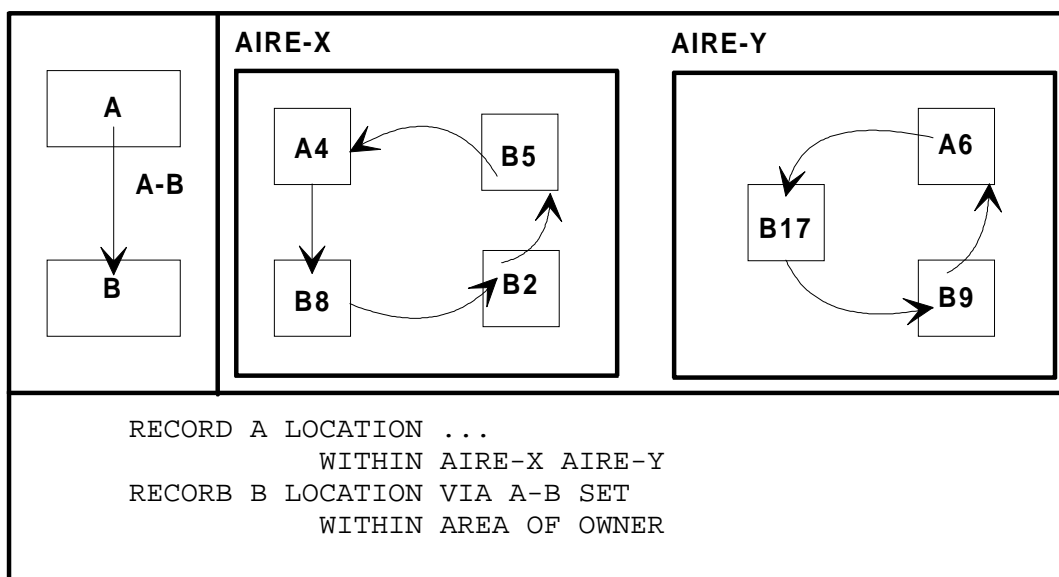


Figure 2-8. Ensemble local multi-aire

DDLPROC, si aucune option DMCL ne l'en empêche, disposera pour un ensemble local :

- des pointeurs locaux pour l'ensemble mono-aire,
- des pointeurs locaux pour l'ensemble multi-aire si la taille de pointeur local en octets est la même pour toutes les aires concernées. (La mise en oeuvre actuelle impose que la structure d'un article mémoire soit la même pour toutes les aires où il peut être enregistré).
- des pointeurs globaux pour l'ensemble multi-aire, si la taille de pointeur local en octets n'est pas la même pour toutes les aires concernées.

L'Administrateur de base de données, lorsqu'il sait que les ensembles locaux en cours sont appelés à devenir des ensembles globaux, peut utiliser :

- soit la clause NO LOCAL POINTERS de la rubrique schéma pour forcer la mise en oeuvre de tous les ensembles locaux avec des pointeurs globaux ;
- soit la clause NO LOCAL POINTERS de la rubrique ensemble pour forcer la mise en oeuvre de certains ensembles locaux avec des pointeurs globaux.

Dans le cas d'un ensemble multi-aire avec plusieurs tailles différentes de pointeurs locaux, l'Administrateur de base de données peut utiliser la clause EXTEND LOCAL POINTER de la rubrique aire pour forcer la mise en oeuvre d'une taille de pointeur local supérieure à celle qui est nécessaire et déterminer ainsi des tailles de pointeurs locaux qui seront identiques pour toutes les aires concernées.

Exemple :

Supposons que les rubriques DMCL correspondant à la Figure 2-8 sont définies comme suit :

```

AREA NAME IS AIRE-X
  NUMBER-OF-PAGES IS 1500
  NUMBER OF LINES-PER-PAGE IS 50
  ...
AREA NAME IS AIRE-Y
  NUMBER-OF-PAGES IS 1300
  NUMBER OF LINES-PER-PAGE IS 50
  EXTEND LOCAL POINTERS TO 3 BYTES
  ...
    
```

Le nombre d'articles dont l'adressage se fera par pointeur local dans AIRE-X est de $1500 * 50 = 75000$, soit un nombre équivalent de valeurs de clé. D'après la liste de tailles de pointeur de la Figure 2-1, 3 octets sont nécessaires. Le même calcul effectué pour AIRE-Y donne 2 octets ($1300 * 50 = 65000$). Puisque dans cet exemple, le type d'ensemble est représenté dans les deux aires, la taille de pointeur local doit être forcée à la même valeur pour les deux aires. Il sera donc nécessaire d'introduire une clause `EXTEND LOCAL POINTERS TO 3 BYTES` pour AIRE-Y dans la rubrique DMCL.

2.4.4 Ensembles globaux

Un ensemble est dit global lorsque les spécifications DDL indiquent que les éléments d'un ensemble peuvent être disposés dans différentes aires. La Figure 2-9 permet d'illustrer cette situation.

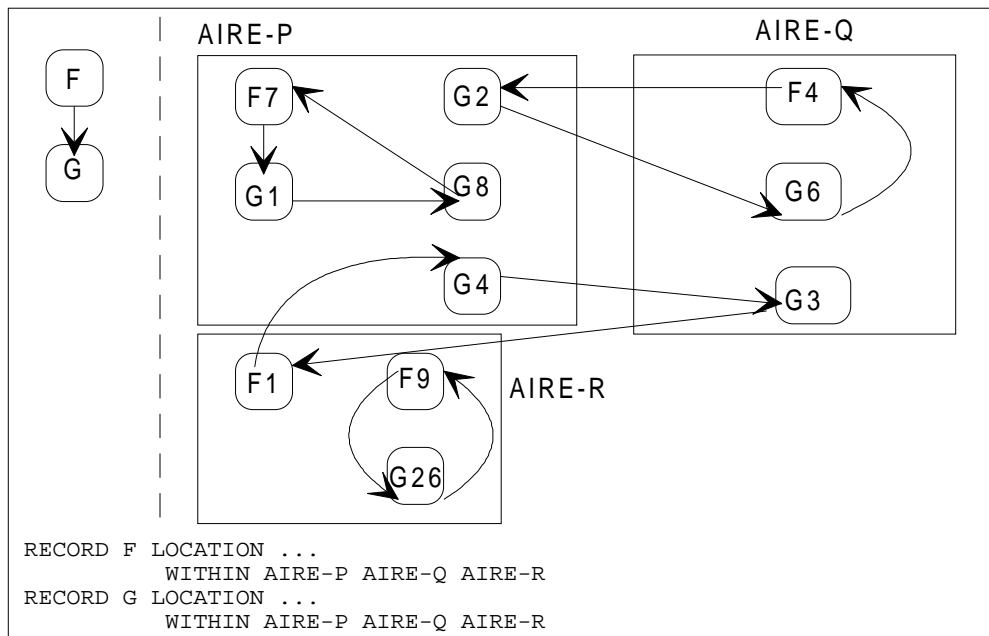


Figure 2-9. Ensemble global

Un ensemble global est toujours mis en oeuvre à l'aide de pointeurs globaux.

2.5 STRUCTURE D'ARTICLE MEMOIRE

2.5.1 Numéro d'article

Chaque article mémoire est identifié d'une manière interne par un numéro "r" d'article, commençant à 1 (r = 0 est réservé au système).

Les numéros d'articles sont rangés séquentiellement dans l'ordre des rubriques article du DDL.

Le numéro de l'article est rangé dans l'en-tête d'article de la base de données.

2.5.2 Structure d'article

Un article mémoire se compose de trois parties comme indiqué dans la Figure 2-10.

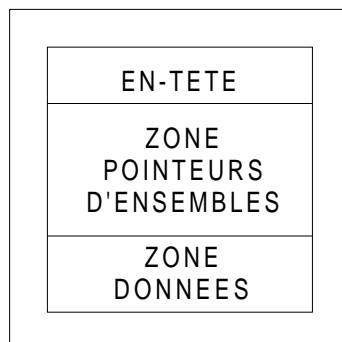


Figure 2-10. Structure de l'article mémoire

L'en-tête a une longueur de 5 octets pour les articles non-CALC et de 9 octets pour les articles CALC. Il contient le numéro de l'article, l'indication de sa longueur totale, ses indicateurs d'état et, pour ce qui concerne les articles CALC, des informations sur sa chaîne CALC.

La zone pointeurs d'ensembles, dont la longueur varie en fonction du type d'article, contient les pointeurs d'ensembles qui désignent les ensembles pour lesquels l'article a été déclaré maître ou détail.

Le nombre, le type et la taille des pointeurs sont déterminés pour chaque ensemble comme il a été expliqué dans les paragraphes précédents.

L'ordre de classement du groupe de pointeurs d'ensembles se présente comme suit :

- En premier, les ensembles dans lesquels l'article est déclaré comme maître. Ces ensembles apparaissent dans l'ordre croissant de leur numéro.
- Puis, les ensembles pour lesquels l'article a été déclaré comme détail. Si le mode de localisation de l'article est VIA nom-ensemble, ces ensembles apparaissent dans l'ordre croissant de leur numéro. Si le mode de localisation de l'article est VIA nom-ensemble-1, nom-ensemble-1 apparaît en premier puis viennent les autres noms d'ensembles dans l'ordre croissant de leurs numéros.

Lorsqu'un type d'article n'appartient à aucun type d'ensemble, la zone pointeurs n'existe pas.

La zone données, dont la longueur varie en fonction du type d'article, contient les données telles qu'elles ont été définies dans les différentes sous-rubriques donnée du DDL. Pour la mise en oeuvre courante, le format des données dans les articles mémoire est identique à celui des articles de la zone UWA tels que le programme les voit.

Pour certains articles appelés "articles repère" ou "articles de relation" la zone données peut ne pas exister car le rôle de ces articles est réduit à une participation à la structure des ensembles.

L'Administrateur de base de données peut obtenir l'affichage de l'agencement interne des articles au moyen de la commande PRINT STORAGE DESCRIPTION du DDLPROC.

2.5.3 Exemple

Considérons la base de données de la Figure 2-11. Supposons que :

- Le mode de localisation de l'article RC se fait par LOCATION VIA B-C et contient une partie donnée décrite par TYPE CHARACTER 12.
- Les ensembles sont classés dans l'ordre A-C B-C C-D C-E dans le programme DDL.
- Les ensembles locaux B-C C-D sont mis en oeuvre au moyen de pointeurs locaux à deux octets.
- Les ensembles globaux A-C C-E sont mis en oeuvre au moyen de pointeurs globaux à 3 octets.

L'article RC, non-CALC, dispose d'un en-tête de 5 octets. La longueur totale de l'article est 42 octets. (En-tête + pointeur + données).

L'ensemble C-D est le premier dont l'article RC est le maître. Le groupe de pointeurs est représenté par deux pointeurs locaux: NEXT, PRIOR.

L'ensemble C-E est le second ensemble dont RC est le maître. Le groupe de pointeurs est représenté par deux pointeurs globaux: NEXT, PRIOR.

L'ensemble B-C n'est pas le premier ensemble dont RC est article détail mais il est utilisé comme élément de localisation de l'article dans LOCATION mode VIA. Le groupe de pointeurs est représenté par trois pointeurs locaux: NEXT, PRIOR, OWNER.

Langage DMCL

L'ensemble A-C est le premier ensemble dont RC est un article détail. Le groupe de pointeurs contient trois pointeurs globaux: NEXT, PRIOR, OWNER.

La zone données a une longueur de 12 octets.

Table 2-11. Exemple d'agencement d'un article mémoire

RA	Ensemble	RB	
	A-C		
		Ensemble B-C	
		RC	Ensemble C-E
		Ensemble C-D	
		RD	RE
AIRE-1		AIRE-2	AIRE-3
Article m,moire RC			
En-tête			
Maître dans C-D NEXT PRIOR			
Maître dans C-B NEXT PRIOR Longueur			
D,tail dans B-C NEXT PRIOR OWNER			
D,tail dans A-C NEXT PRIOR OWNER			
Zone données			

2.6 SELECTION DU PARAMETRE PAGE-SIZE

Lorsque l'Administrateur de base de données choisit le paramètre DMCL PAGE-SIZE, il doit tenir compte d'un certain nombre de facteurs et décider en fonction du meilleur compromis possible.

1. PAGE-SIZE représente la taille totale de la page. Il comprend l'en-tête de page, les localisateurs d'articles, au moins un article OWL et pour chaque article mémoire, son en-tête, sa zone pointeurs et sa zone données.
2. La taille de page doit se conformer aux caractéristiques d'appareil et aux règles UFAS :
 - PAGE-SIZE doit être un multiple de 256 octets. De façon à faciliter la migration sur les disques à sectorisation fixe, il est conseillé de choisir des multiples de 512 octets.
 - Plus la taille de page est grande par rapport à la taille de piste, moins grand est le nombre d'octets réservé pour les espaces interpages.
 - Les pages ne peuvent chevaucher les pistes. Cette contrainte réduit le choix des valeurs si l'on veut utiliser efficacement l'espace disque disponible, en particulier si la taille de page se rapproche de la taille nominale de la piste.

Les valeurs conseillées sont indiquées dans le Tableau 2-4 pour les familles de disques MSU0300 et MSU0400 et pour les types MSU0500 et MSD300.

Les pourcentages indiquent les taux de rendement de l'espace piste par rapport à la capacité nominale de la piste.

3. Il est conseillé d'utiliser de grandes pages lorsque celles-ci contiennent des groupes d'articles qui sont traités en commun. En effet, dans cette situation, une seule opération d'E/S est nécessaire pour transférer le groupe en mémoire.

Les grandes pages sont par contre déconseillées si elles sont généralement chargées en mémoire pour le traitement d'un seul article à la fois. En effet, non seulement la taille de tampon nécessaire est aussi grande que celle de la page et peut dépasser largement la taille d'article, mais le temps de transfert est lui aussi proportionnel à la taille de page.

4. Dans un cadre d'exploitation TDS, qui implique de nombreuses transactions simultanées, de grandes tailles de pages génèrent un grand besoin en espace tampon si l'on veut éviter des conflits pour les ressources tampon (en supposant que le partage d'accès n'a pas été autorisé pour les mêmes pages).
5. La taille de page peut varier d'une aire à l'autre. Si l'on utilise plusieurs aires avec des tailles de pages différentes dans le même pool, il en résulte une contrainte supplémentaire de temps pour le processeur CPU du fait de l'adaptation dynamique des tailles de tampons aux tailles de pages. Cette contrainte peut être évitée dès lors que les aires disposent d'une taille de page identique.

Pour plus de détails sur la gestion du pool de tampons UFAS, se reporter au manuel UFAS User Guide (référence: 46 A2 AQ84). Le mot IDS "page" est synonyme de "CI" ou "Intervalle de contrôle".

Tableau 2-4. Valeurs conseillées pour la taille de page

TAILLE PAGE	PAGES PAR PISTE	PAGES PAR CYLINDRE	
			MSU 0350/400/401/402/452
12800	1 (98%)	19	
12288	1 (94%)	19	
6144	2 (94%)	38	
4096	3 (94%)	57	
3072	4 (94%)	76	
2048	6 (94%)	114	
1536	7 (83%)	133	
1024	11 (86%)	209	
512	20 (79%)	380	
			MSU 0310
6144	1 (84%)	20	
3072	2 (84%)	40	
2048	3 (84%)	60	
1536	4 (84%)	80	
1024	6 (84%)	120	
512	11 (77%)	220	
			MSU 0500
18944	1 (98%)	38	
9216	2 (96%)	76	
6144	3 (96%)	114	
4608	4 (96%)	152	
3584	5 (93%)	190	
2560	7 (93%)	266	
2048	8 (85%)	304	
1536	11 (88%)	418	
1024	15 (80%)	570	
512	27 (72%)	1026	
			SMD 300
18944	1 (97%)	19	
18432	1 (94%)	19	
9216	2 (94%)	38	
6144	3 (94%)	57	
4608	4 (94%)	76	
3584	5 (91%)	95	
3072	6 (94%)	114	
2560	7 (91%)	134	
2048	8 (83%)	152	
1536	11 (86%)	209	
1024	16 (83%)	304	
512	26 (73%)	532	

2.7 SELECTION DU PARAMETRE LINES-PER-PAGE

- Le paramètre LINES-PER-PAGE permet de définir le nombre maximum d'articles qui peuvent être adressés dans une page. Sa valeur qui peut varier d'une aire à l'autre, ne doit pas dépasser 255.

Si le paramètre PAGE-SIZE a été défini auparavant, la valeur de LINES-PER-PAGE en est déduite en considérant la taille des articles de l'aire. La valeur choisie doit être assez grande pour adresser tous les articles que l'espace page peut contenir.

- Si LINES-PER-PAGE est trop petit, il en résulte un épuisement des lignes qui se produit avant la saturation d'espace, provoquant ainsi une mauvaise utilisation de l'espace disque.
 - Si la valeur de LINES-PER-PAGE est trop grande, il en résulte une saturation d'espace avant l'épuisement des lignes. Un certain nombre de lignes sont perdues réduisant ainsi la capacité totale d'adressage de la clé de base de données.
 - Si les aires contiennent des types d'articles de tailles différentes il est conseillé de choisir la valeur de LINES-PER-PAGE en fonction des plus petits articles. Pour les grands articles, la saturation d'espace précèdera ainsi l'épuisement des lignes.
- Le paramètre LINES-PER-PAGE peut être choisi avant le paramètre PAGE-SIZE. C'est le cas notamment quand l'aire contient des groupes d'articles détails et qu'une seule page est supposée pouvoir contenir le nombre moyen d'articles du groupe. Ce nombre moyen est ensuite utilisé en l'attribuant comme valeur à LINES-PER-PAGE. PAGE-SIZE correspond alors à la taille de l'article.
 - Si, pour des raisons qui ont été vues au paragraphe Sélection du paramètre PAGE-SIZE, la valeur de PAGE-SIZE est trop élevée, il peut être utile d'enregistrer le groupe sur un nombre n fixe de pages (avec n différent de 1). On choisit ensuite LINES-PER-PAGE en lui donnant une valeur égale au nombre d'articles d'un groupe divisé par n . PAGE-SIZE est ensuite déduit de la taille d'article comme expliqué plus haut.

2.8 SELECTION DE L'INTERVALLE CALC

L'intervalle CALC d'un type d'article dans une plage peut être choisi, en supposant une distribution CALC uniforme, de la manière suivante :

- Si, dans la plage, le nombre d'articles CALC est plus petit que le nombre de pages, la valeur de l'intervalle CALC est calculée en divisant le nombre de pages par le nombre d'articles CALC. C'est le cas par exemple lorsque les groupes d'articles détails non CALC sont situés dans la même plage et s'étendent sur deux ou plusieurs pages. La situation idéale consisterait à avoir une chaîne CALC qui ne contiendrait qu'un seul article.
- Si, dans la plage, le nombre d'articles CALC est plus élevé que le nombre de pages, la valeur de l'intervalle CALC est d'une page car la valeur minimale rend le plus petit possible le nombre moyen d'articles par chaîne CALC et par conséquent diminue le débordement sur les pages qui suivent.

L'intervalle CALC est constant dans toute l'aire. Sa valeur peut donc résulter d'un compromis si plusieurs types d'articles CALC doivent être implantés sur la même ou sur plusieurs plages de l'aire.

Le nombre de pages d'une aire de plage doit être un multiple de l'intervalle CALC. Le nombre de compartiments CALC, c'est-à-dire le nombre de pages divisé par la valeur de l'intervalle CALC devrait être un nombre premier ou, sinon, qui ne comporte pas de sous-multiples trop élevés: 2, 3, 5, 7, 11, 13, 17, 19. L'option OPTIMIZE de l'aire DMCL et des sous-rubriques article permet à l'utilisateur de choisir le nombre premier le plus proche.

L'insertion dans une chaîne CALC de nouveaux articles ou leur extraction de cette chaîne s'ils existent déjà peut nécessiter une recherche dans la liste complète des articles du compartiment CALC concerné. Si la duplication d'articles n'est pas autorisée, la recherche porte sur la totalité de la chaîne CALC pour trouver les doubles de zones utilisées dans la randomisation.

(C'est-à-dire le calcul de l'adresse à partir du contenu de l'article).

Lorsque des articles détails non-CALC sont regroupés près de leur article CALC maître et lorsque l'accès au maître doit se faire plus souvent qu'au détail, il peut être utile de charger tous les articles maîtres en premier puis les articles détails. L'opération aura pour effet d'implanter les synonymes sur la première page d'un compartiment CALC et de ne nécessiter qu'une seule opération d'entrée-sortie pour accéder à ces articles. Dans le cas contraire, lorsque la répartition n'est pas uniforme ou lorsque le nombre d'articles d'un groupe est variable, les articles CALC détails peuvent être amenés à remplir la première page du compartiment CALC et de provoquer ainsi le déport du synonyme suivant sur une page de débordement.

Le taux de débordement CALC s'accroît avec le nombre d'articles enregistrés dans l'aire. Les aires doivent donc disposer d'un espace suffisant pour éviter la saturation. Un facteur de charge de 70% (d'espace ou pourcentage de lignes) se révèle généralement satisfaisant.

L'utilitaire H _ DBUTILITY permet de mettre en oeuvre une fonction de simulation qui permet de vérifier la validité de l'intervalle CALC choisi avant l'affectation d'espace aux aires.

2.9 SYNTAXE DMCL

Le langage DMCL utilise les mêmes règles de syntaxe que le DDL. Voir au chapitre 1.

Chaque rubrique DMCL fait référence à un schéma DDL résultant et décrit la façon dont il est projeté (MAPPED) en mémoire.

La structure générale du DMCL comprend:

- la rubrique schéma,
- une ou plusieurs rubriques aire,
- une ou plusieurs rubriques article,
- une ou plusieurs rubriques ensemble.

Les rubriques peuvent apparaître dans n'importe quel ordre après la rubrique schéma. Chaque rubrique doit se terminer par un point suivi d'un espace et d'une fin de ligne. Le texte DMCL origine complet se termine par le mot-clé END-DMCL.

Chacune des rubriques est détaillée dans les pages suivantes.

La Figure 2-12 donne un échantillon de DMCL origine.

Figure 2-12. Echantillon DMCL origine

COMMENT	
COMMENT	SCHEMA DMCL FOR TEST-GROUP *TG19*
COMMENT	
COMMENT	
COMMENT	SCHEMA
COMMENT	
SCHEMA	TG19-SH
COMMENT	
COMMENT	AREAS
COMMENT	
AREA	TG19-A00. NUMBER-OF-PAGES 53 LINES-PER-PAGE 45 PAGE-SIZE 2048.
AREA	TG19-A01 NUMBER-OF-PAGES 52 LINES-PER-PAGE 45 PAGE-SIZE 2048.
END-DMCL	

2.9.1 Rubrique schéma

Fonction

Permet d'identifier le schéma et de spécifier divers attributs généraux de mémoire et d'exécution qui n'appartiennent pas en propre à une aire, un article ou un ensemble donné.

Format général

```

SCHEMA NAME IS nom-schéma

[EXTEND NUMBER OR AREAS TO entier-1]

[EXTEND GLOBAL POINTERS TO entier-2 BYTES]

[NO LOCAL POINTERS]

[BUFFER POOL NAME IS nom-pool-tampons]

[NUMBER OF BUFFERS IS entier-3] .

```

2.9.1.1 Clause SCHEMA

Fonction

Permet de spécifier le nom du schéma.

Format

```

SCHEMA NAME IS nom-schéma

```

Règles

1. Nom-schéma est le nom du schéma qui a été spécifié dans la rubrique SCHEMA du DDL.

2.9.1.2 Clause EXTEND NUMBER OF AREAS

Fonction

Permet de forcer la taille des pointeurs globaux de manière à leur permettre de contenir un plus grand nombre d'aires que celui qui a été spécifié dans le schéma en cours.

Format

```
EXTEND NUMBER OF AREAS TO entier
```

Règles

1. Cette clause, si elle existe, doit spécifier une valeur plus élevée que le nombre d'aires qui a été défini.
2. Si cette clause est spécifiée, le SGBD construira des pointeurs globaux avec une zone de numéro d'aire suffisamment grande pour contenir le nombre d'aires spécifié.

Pour plus de détails se reporter au paragraphe "Adressage d'article mémoire" plus haut dans ce chapitre.

3. Si cette clause n'est pas utilisée, les pointeurs globaux devront contenir suffisamment d'espace pour traiter le nombre d'aires déclaré. Si, par exemple, trois aires ont été déclarées, il n'est pas nécessaire de spécifier EXTEND NUMBER OF AREAS TO 4 car il y aura assez de place dans les pointeurs globaux pour traiter quatre aires.

2.9.1.3 Clause EXTEND GLOBAL POINTERS

Fonction

Permet de forcer la taille des pointeurs globaux du schéma à une taille plus grande que celle qui est couramment nécessaire.

Format

```
EXTEND GLOBAL POINTERS TO entier BYTES
```

Règles

1. "entier" peut prendre une valeur de 3 ou 4.
2. Lorsqu'elle est spécifiée, la valeur doit être égale ou supérieure à la taille qui aurait été choisie automatiquement.
3. Lorsque cette clause est omise, le calcul de taille de pointeur global s'effectue en utilisant l'information qui a été fournie dans le DMCL pour chaque aire.
4. Si cette clause est utilisée, les pointeurs globaux sont mis en oeuvre avec la taille spécifiée.
5. Pour plus de détails, se reporter également au paragraphe "Adressage d'article mémoire" au début de ce chapitre.

2.9.1.4 Clause NO LOCAL POINTERS

Fonction

Permet de forcer tous les ensembles à posséder des pointeurs globaux.

Format

```
NO LOCAL POINTERS
```

Règles

1. Si cette clause est omise et si aucune spécification contraire n'est indiquée dans la rubrique ensemble elle-même, les types d'ensemble définis comme locaux dans le schéma seront maintenus avec leurs pointeurs locaux.
2. Si cette clause est spécifiée, tous les ensembles locaux sont mis en oeuvre avec pointeurs globaux.
3. Pour plus de détails se reporter au paragraphe "Mise en oeuvre par chaîne de pointeurs" au début de ce chapitre.

2.9.1.5 Clause BUFFER POOL

Fonction

Permet de désigner un pool de tampons que les aires du schéma vont utiliser au moment de l'exécution.

Format

```
BUFFER POOL NAME is nom-pool-tampons
```

Règles de syntaxe

1. Nom-pool-tampons ne doit pas dépasser quatre caractères.
2. Si un nom en blanc est utilisé il doit être encadré par deux apostrophes ' ', soit quatre espaces entre les apostrophes.
3. Les noms qui ne sont pas en blanc peuvent contenir des lettres, des chiffres, des signes moins et le caractère de soulignement. Tous ces caractères, s'ils ne se conforment pas aux règles de formation des noms DDL, doivent être encadrés par des apostrophes.

Règles générales

1. Si cette clause est omise, la valeur implicite choisie pour ce nom est DBPL (Data Base Pool) qui est le pool de tampons de la base de données.
2. Lorsque la valeur spécifiée est un nom en blanc (' ') la clause NUMBER OF BUFFERS indique le nombre de tampons réservé pour chaque aire, c'est-à-dire que chaque aire dispose de son propre groupe de tampons.
3. Lorsque la spécification tampon n'est pas un nom en blanc (ce qui constitue l'option implicite) la clause NUMBER OF BUFFERS donne le nombre total de tampons qui sera partagé par toutes les aires.

Langage DMCL

4. Les spécifications requises pour la gestion des tampons peuvent également être fournies par des options choisies au moment de l'exécution. Ainsi toute déclaration DMCL peut être remplacée.
5. L'utilisation d'un pool de tampons a pour but de réduire l'espace tampon global nécessaire lorsque plusieurs aires se trouvent à l'état prêt (READY). En supposant qu'un programme DML accède à une aire pour une période de temps indéterminée :
 - S'il n'existe pas de pool de tampons, elle ne peut utiliser plus d'espace tampon que celui qui lui a été réservé.
L'espace réservé aux autres aires est indisponible.
 - S'il existe un pool de tampons, l'espace tampon complet peut être utilisé pour l'aire en cours d'exécution. Lorsque le programme lance le traitement d'une autre aire, l'espace tampon complet est affecté au tampon de cette nouvelle aire.

2.9.1.6 Clause NUMBER OF BUFFERS

Fonction

Permet de spécifier le nombre de tampons qui seront utilisés au moment de l'exécution.

Format

`NUMBER OF BUFFERS IS entier`

Règles

1. Cette spécification est facultative. Si elle est omise la valeur choisie implicitement est 4.
2. La valeur minimum qui peut être spécifiée est 3.
3. Si le nom-pool-tampons n'est pas sous la forme à blanc, la valeur spécifiée est le nombre de tampons que doivent se partager toutes les aires. Dans la situation où les différentes aires ont des tailles de pages différentes, la valeur choisie sera la plus élevée.
4. Si le nom-pool-tampons est un nom à blanc, la valeur spécifiée (ou choisie implicitement) est le nombre de tampons à réserver pour chaque aire.
5. Les spécifications requises pour la gestion des tampons peuvent être également fournies par des options au moment de l'exécution. Ainsi, toute déclaration DMCL peut être remplacée.

6. Le nombre de tampons que l'on peut choisir au moment de l'exécution constitue un compromis entre l'espace total requis et les contraintes de performances.

Le SGBD, que la spécification en pool ait été indiquée ou non, fonctionne sur un minimum de trois tampons.

Si le programme est appelé à travailler un certain temps sur un groupe limité de pages, le nombre de tampons peut être choisi en fonction de ces pages à traiter. Si le programme fonctionne sur un mode d'accès sélectif aux pages, l'accroissement du nombre de tampons ne permet pas d'augmenter la rapidité de traitement global.

2.9.2 Rubrique AREA

Fonction

Permet de déclarer les attributs de l'aire mémoire qui correspond à une aire du schéma.

Format général

```

AREA NAME IS nom-aire

    [AREA INTERNAL FILE NAME IS nfi]
    [ NUMBER-OF-PAGES IS entier-1 [ OPTIMIZE { HIGH } ] ]
    [ NUMBER OF LINES-PER-PAGE IS entier-2 ]
    [PAGE-SIZE IS entier-3 BYTES ]
    [CALC-INTERVAL IS entier-4 PAGES]
    [EXTEND LOCAL POINTERS TO entier-5 BYTES] .
    
```

Règles

1. Il doit exister une rubrique aire pour chaque aire déclarée dans le schéma.

2.9.2.1 Clause AREA NAME

Fonction

Permet de nommer l'aire du schéma qui est projetée dans l'aire mémoire décrite dans les autres clauses de cette rubrique.

Format

```
AREA NAME IS nom-aire
```

Règles

1. "nom-aire" est le nom DDL d'une aire du schéma.
2. Le nom DMCL de l'aire mémoire est implicitement le même que le nom DDL de l'aire du schéma.

2.9.2.2 Clause AREA INTERNAL FILE NAME

Fonction

Permet de définir le nom du fichier interne (nfi) à utiliser au moment de l'exécution dans les instructions ASSIGN/DEFINE citant cette aire mémoire.

Format

```
AREA INTERNAL FILE NAME IS nfi
```

Règles

1. "nfi" doit se conformer aux règles de définition du nom-de-fichier-interne du manuel JCL Reference Manual (Réf: 47 A2 11UJ).

De plus, nfi doit être différent de :

```
IDSOPT  
IDSTRACE  
PRTFILE  
COMFILE  
SLLIBa  
DDLIBb  
SAVEc
```

Dans lesquels a, b, c, représentent des chiffres décimaux ou des espaces.

2. Si cette clause est omise, le nom-de-fichier-interne implicite est celui du nom d'aire mémoire (aligné à droite sur 8 caractères si nécessaire).

3. Il ne doit pas exister de double parmi les noms-de-fichier d'un schéma.

Si, au cours d'une même activité, l'accès doit se faire à plusieurs bases de données et si les noms-de-fichier-interne ne sont pas identiques parmi les différents schémas utilisés, l'utilisateur doit se préoccuper de l'unicité de nom de chaque fichier interne au moyen de la commande INTERNAL FILE NAME pour les schémas concernés au moment de l'exécution.

4. Dans la génération TDS et pour les commandes opérateur, les aires mémoire sont citées par leur nom-de-fichier-interne comme spécifié, explicitement ou implicitement, dans le DMCL. Différentes options permettent de remplacer ces noms au moment de l'exécution.

2.9.2.3 Clause NUMBER OF PAGES

Fonction

Permet de définir le nombre de pages d'une aire mémoire.

Format

```
NUMBER-OF-PAGES IS entier [ OPTIMIZE { HIGH } ]
                             [ { LOW } ]
```

Règles

1. "entier" représente le nombre de pages NUMP. Il doit être conforme à la formule :

$$1 \leq \text{NUMP} \leq 16,777,215$$
2. Si OPTIMIZE n'est pas spécifié et si l'aire mémoire ne contient pas d'article mémoire CALC, il n'existe aucune autre contrainte de programmation pour NUMP.
3. Si OPTIMIZE n'est pas spécifié et que l'aire mémoire contient des articles CALC, NUMP doit obligatoirement être un multiple de la valeur fournie par le paramètre CALC-INTERVAL. Il est conseillé de fournir un nombre de "compartiments" (soit le nombre de pages divisé par CALC-INTERVAL) qui soit un nombre premier ou, sinon, qui ne comporte pas des sous-multiples trop élevés: 2, 3, 5, 7, 11, 13, 17, 19.
4. Si OPTIMIZE est spécifié et que l'aire mémoire contient des articles CALC, DDLPROC procède à une incrémentation (HIGH) ou une décrémentation (LOW) de NUMP de manière à obtenir le nombre premier suivant pour les compartiments.
5. OPTIMIZE ne doit pas être spécifié si l'aire mémoire ne contient aucun article CALC.

2.9.2.4 Clause LINES-PER-PAGE

Fonction

Permet de définir le nombre de lignes pour chaque page de l'aire mémoire.

Format

```
NUMBER OF LINES-PER-PAGE IS entier
```

Règles

1. "entier" est le nombre LPP de lignes par page. Il doit respecter la formule:
$$1 \leq LPP \leq 255$$
2. Pour plus de détails sur la manière de sélectionner la valeur se reporter au paragraphe "Sélection du paramètre LINES-PER-PAGE" dans ce chapitre.

2.9.2.5 Clause PAGE-SIZE

Fonction

Permet de définir la taille en octets des pages de l'aire mémoire.

Format

```
PAGE-SIZE IS entier-BYTES
```

Règles

1. "entier" doit être un multiple de 256 et, si possible, un multiple de 512 pour des raisons de migration d'articles sur des disques à sectorisation fixe.
2. Pour plus de détails sur la sélection des valeurs, se reporter au paragraphe "Sélection du paramètre PAGE-SIZE" dans ce chapitre.

2.9.2.6 Clause CALC-INTERVAL

Fonction

Permet de définir le nombre de pages du "compartiment" impliqué dans l'algorithme de randomisation pour l'agencement des articles CALC dans cette aire mémoire.

Format

```
CALC-INTERVAL IS entier-PAGES
```

Règles

1. "entier" est le nombre B de pages du compartiment. Il doit respecter la formule:
$$1 \leq B \leq 255$$
2. Cette clause ne doit pas être spécifiée pour une aire mémoire qui ne contient aucun article CALC.
3. Lorsque cette clause n'est pas spécifiée et que l'aire mémoire contient des articles CALC, c'est la valeur implicite 1 qui est choisie.
4. Pour plus de détails sur la manière de sélectionner les valeurs, se reporter au paragraphe "Sélection de l'intervalle CALC" dans ce chapitre.

2.9.2.7 Clause EXTEND LOCAL POINTERS

Fonction

Permet de forcer la taille des pointeurs locaux de l'aire mémoire à une valeur supérieure à celle qui est théoriquement nécessaire.

Format

```
EXTEND LOCAL POINTERS TO entier-BYTES
```

Règles

1. "entier" peut prendre la valeur 3 ou 4.
2. On utilise cette clause lorsque des articles de même type sont contenus dans un type d'ensemble local et sont rangés sur plusieurs aires avec des tailles de pointeurs locaux différentes. Comme les pointeurs du même type d'article mémoire doivent être de même taille, l'utilisateur a la possibilité de forcer la taille du plus petit pointeur à celle du plus grand.
3. Pour plus de détails, se reporter au paragraphe "Mise en oeuvre d'ensembles par chaîne de pointeurs" dans ce chapitre.

2.9.3 Rubrique RECORD**Fonction**

Permet de déclarer les attributs de l'article mémoire qui correspond à celui du schéma.

Format général

RECORD NAME IS nom-article

```
[ RANGE [WITHIN nom-aire] IS {PAGE p-1 THRU P-2      } ]
[                               {m PAGES FROM PAGE p-1  } ]
[                               ] ...
[                               [ {HIGH} ] ]
[                               [OPTIMIZE { } ] ]
[                               [ LOW } ] ]

[MIGRATION IS ALLOWED] .
```

Règles

1. Cette rubrique est facultative. Si elle n'est pas spécifiée, ce sont les options implicites de chaque clause qui sont prises en compte.

2.9.3.1 Clause RECORD

Fonction

Permet de désigner le type d'article du schéma qui est projeté (mapped) sur le type d'article mémoire dont les attributs sont décrits dans les autres clauses de cette rubrique.

Format général

```

RECORD NAME IS nom-article
```

Règles

1. "nom-article" est le nom DDL du type d'article d'un schéma.
2. Le nom DMCL du type d'article mémoire est implicitement le même que le nom DDL du type d'article du schéma.

2.9.3.2 Clause RANGE

Fonction

Permet de spécifier, pour chacune des aires mémoire dans laquelle l'article mémoire peut être implanté, la plage qui le contiendra.

Format

```

RANGE [WITHIN nom-aire] IS {PAGE p-1 THRU p-2 }
                               {m PAGES FROM PAGE p-1 }

                               [ OPTIMIZE {HIGH} ]
                               [ {LOW} ]
```

Règles

1. Lorsque cette clause est omise, la plage est supposée correspondre à l'aire elle-même.
2. Si cette clause est mentionnée, la localisation de l'article mémoire se cantonne à la plage spécifiée. Il ne peut exister qu'une seule plage par type d'article dans une aire donnée.
3. Si l'article est du type multi-aire, la locution WITHIN est obligatoire. Pour le type d'article mono-aire cette locution est facultative.

4. La plage se définit comme étant :
 - Soit, une succession de pages spécifiée par un numéro indiquant la page début et la page fin (spécification PAGE THRU). Si NUMP est le nombre de pages de l'aire, les entiers p-1, p-2 doivent remplir la condition :

$$0 \leq p-1 \leq p-2 \leq \text{NUMP} - 1$$
 - Soit, un nombre de pages à partir d'un certain numéro de page (spécification PAGES FROM PAGE). Les entiers m et p-1 doivent remplir les conditions :

$$m > 0$$

$$0 \leq p-1 \leq \text{NUMP} - 1$$

$$p-1 + m \leq \text{NUMP}$$
5. Si le type d'article est CALC, la première page de la plage doit coïncider avec celle de la première page du compartiment CALC.
6. Si OPTIMIZE n'est pas spécifié pour un type d'article CALC, le nombre de pages de la plage doit être un multiple de la valeur de CALC INTERVAL.

Il est conseillé de fournir un nombre de compartiments (c'est-à dire le nombre de pages divisé par la valeur CALC INTERVAL) qui soit un nombre premier, ou sinon, qui ne comporte pas des sous-multiples trop élevés.
7. Si OPTIMIZE est spécifié pour un type d'article CALC, le processeur DDLPROC incrémente (HIGH) ou décrémente (LOW) le nombre de pages spécifié de manière à obtenir le nombre premier suivant pour déterminer le nombre de compartiments.

Si OPTIMIZE HIGH provoque le débordement de l'aire par la plage, aucune optimisation n'est possible. La valeur sélectionnée sera la valeur spécifiée, sous réserve qu'elle soit un multiple de CALC INTERVAL.
8. Si deux types d'article CALC ne sont pas implantés dans la même aire, sur deux plages contiguës, l'utilisation de la fonction OPTIMIZE peut provoquer un léger chevauchement d'une plage sur l'autre. Comme le chevauchement de plages n'est pas admis, les adresses début des plages peuvent être ajustées au cours d'un second passage du DDLPROC.
9. La locution OPTIMIZE ne peut s'utiliser pour des types d'article non-CALC.

Exemple :

Soit une base de données constituée de deux aires (AA, BB), deux types d'article (R01, R02) et un type d'ensemble (S01) dont l'article maître est R01 et l'article détail R02.

R01 est un article CALC dans AA et BB.

R02 est localisé par VIA S01 WITHIN AREA OF OWNER.

L'aire AA est de 500 pages. Son intervalle CALC est 1. R01 est localisé dans les pages 0 à 102 (103 est un nombre premier), R02 dans les pages 103 à 499.

L'aire BB a 1500 pages. Son intervalle CALC est 1. R01 est localisé dans les pages 0 à 306 (307 est un nombre premier), R02 dans les pages 307 à 1499.

Les rubriques DMCL article se présentent de la manière suivante :

RECORD R01

RANGE WITHIN AA IS PAGE 0 THRU 102.
RANGE WITHIN BB IS PAGE 0 THRU 306.

RECORD R02

RANGE WITHIN AA IS PAGE 103 THRU 499.
RANGE WITHIN BB IS PAGE 307 THRU 1499.

2.9.3.3 Clause MIGRATION

Fonction

Permet de spécifier si l'article mémoire CALC doit être transféré dans un nouveau compartiment CALC à la suite de l'utilisation de la fonction MODIFY CALC-KEY.

Format

MIGRATION IS ALLOWED

Règles

1. Cette clause ne peut être mentionnée pour les articles non-CALC.
2. Si cette clause est spécifiée pour un article CALC, cet article sera transféré dans le nouveau compartiment CALC à la suite d'une fonction MODIFY CALC-KEY. Tous les pointeurs d'ensembles qui sont dirigés sur cet article seront mis à jour. Si la clé-base-de-données de l'article est une clé de tri d'un ensemble, cet ensemble sera réorganisé.
3. Si, pour un article CALC, cette clause est omise, l'article ne sera pas transféré à la suite d'une fonction MODIFY CALC-KEY. Cependant, le résultat de la fonction elle-même peut être aléatoire si un article en débordement OWL correspondant à la nouvelle chaîne CALC doit être créé dans la page de l'article alors qu'il n'existe plus de place dans cette page.

2.9.4 Rubrique SET

Fonction

Permet de spécifier la mise en oeuvre d'un type d'ensemble du schéma.

Format Général

```
SET NAME IS nom-ensemble  
[NO LOCAL POINTERS].
```

Règles

1. Cette rubrique est facultative. Si elle n'est pas spécifiée, le système utilise les options implicites de chaque clause.

2.9.4.1 Clause SET

Fonction

Permet de désigner le type d'ensemble du schéma dont la mise en oeuvre est décrite dans l'autre clause de cette rubrique.

Format Général

```
SET NAME IS nom-ensemble
```

Règles

1. "nom-ensemble" est le nom DDL d'un type d'ensemble du schéma.

2.9.4.2 Clause NO LOCAL POINTERS

Fonction

Permet de mettre en oeuvre des pointeurs globaux pour un type d'ensemble.

Format

<code><u>NO</u> LOCAL POINTERS</code>

Règles

1. Cette clause ne doit pas être spécifiée pour un type d'ensemble global.
2. Si, le type-ensemble étant local, cette clause a été omise, l'ensemble sera mis en oeuvre avec pointeurs locaux. Cependant, il n'y aura pas de pointeurs locaux si l'ensemble est de type multi-aire, si les tailles de pointeurs sont inégales ou si la clause NO LOCAL POINTERS a été spécifiée.
3. Si cette clause est spécifiée pour un type d'ensemble local, celui-ci est mis en oeuvre avec des pointeurs globaux.
4. Pour plus de détails, se reporter au paragraphe intitulé "Mise en oeuvre d'ensemble par chaîne de pointeurs" plus haut dans ce chapitre.

3. Traitement du schéma

3.1 FONCTIONS DU PROCESSEUR DDLPROC

Le processeur IDS/II DDLPROC permet de traduire les langages origine DDL et DMCL sous forme d'un schéma résultant. Ce processeur exécute cinq fonctions différentes qui sont détaillées dans la Figure 3-1.

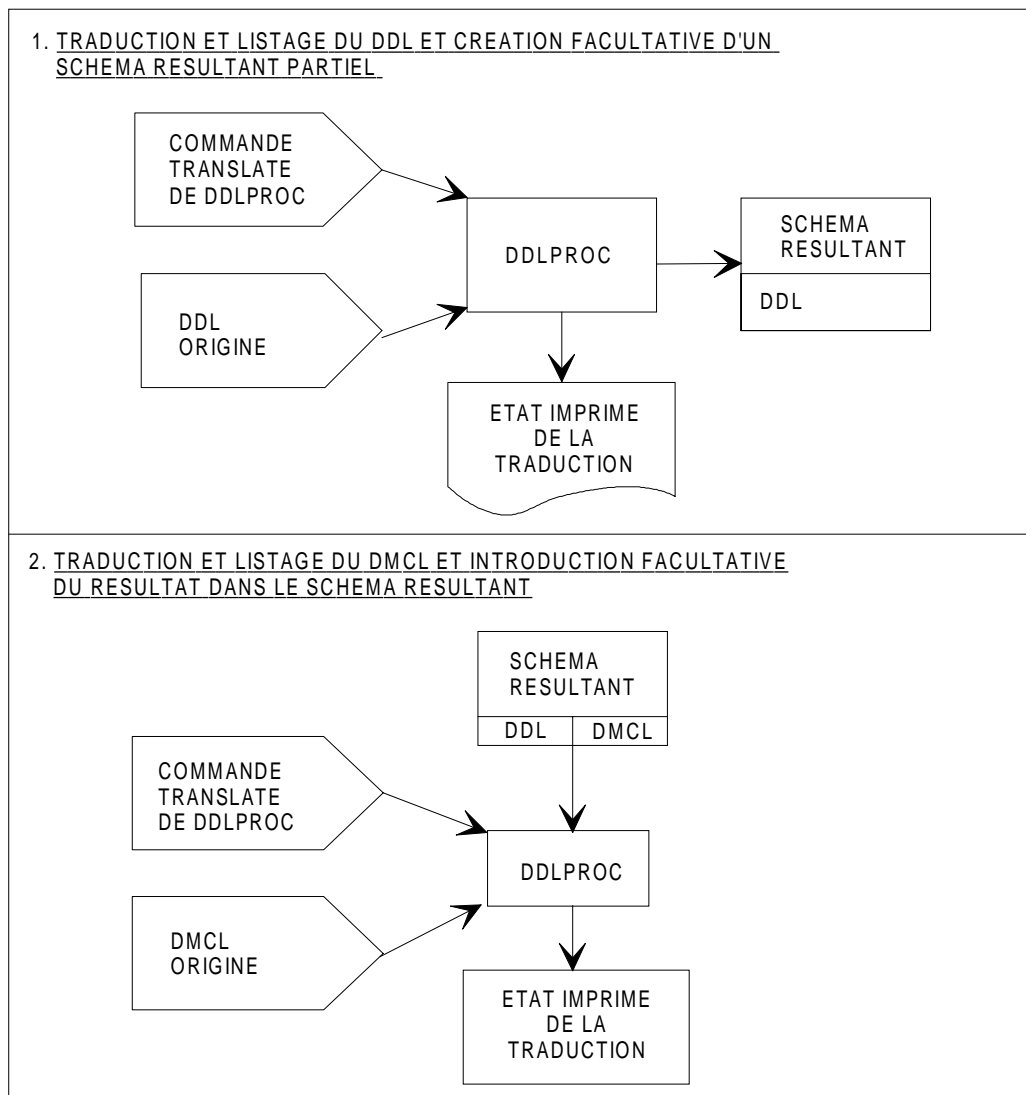


Figure 3-1. Fonctions du Processeur DDLPROC (1/2)

Traitement du schéma

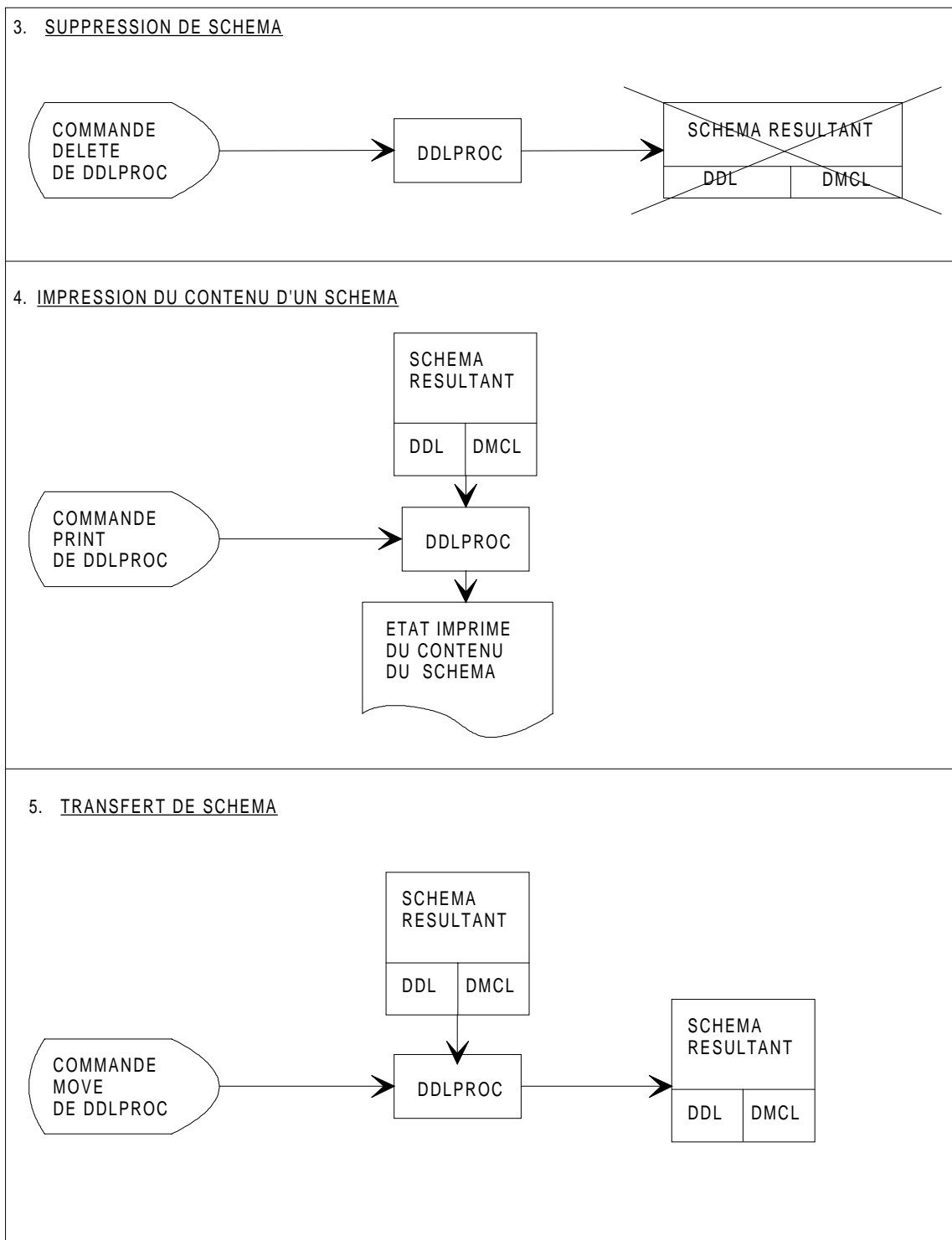


Figure 3-1. Fonctions du DDLPROC (2/2)

3.2 TRAITEMENT DES ELEMENTS SL ET DD

- DDLPROC manipule deux types d'éléments :
 - Les éléments du langage origine SL (Source Language) correspondant au DDL de schéma origine et au DMCL origine.
 - Les éléments résultants de description des données DD (Data Description).
- Les éléments SL peuvent être enregistrés dans :
 - Des rubriques de description d'entrées pourvues de l'option TYPE DATA ou TYPE DATASSF.
 - Des unités de bibliothèques SL. L'option TYPE de ces unités doit être DATASSF. On utilise la fonction LIBALLOC SL pour affecter de la place aux bibliothèques SL :

```
LIBALLOC SL (description-bibliothèque SIZE=entier)
            [MAXSIZE=entier]
            {DIRSIZE=entier}
            {MEMBERS=entier}
            [COMPACT] ;
```

- Les éléments DD sont enregistrés dans des unités de bibliothèque de type BIN. On leur affecte de la place en utilisant la fonction LIBALLOC BIN :

```
LIBALLOC BIN (description-bibliothèque SIZE=entier)
            [MAXSIZE=entier]
            {DIRSIZE=entier} ;
            {MEMBERS=entier}
```

- Il n'existe pas de relation directe entre le nom de schéma X, tel qu'il apparaît dans la clause DDL SCHEMA NAME et le nom de la rubrique de description d'entrées ou l'unité de bibliothèque qui contient le DDL origine de schéma ou le DMCL origine.

Les règles d'écriture ci-dessous peuvent être adoptées si le nom de schéma n'est pas trop long :

- Nom de l'entité contenant le DDL origine: X-DDL (rubrique de description d'entrées, unité de bibliothèque).
- Nom de l'entité contenant le DMCL origine: X-DMCL (rubrique de description d'entrées, unité de bibliothèque).

Le nombre de caractères utilisés ne doit pas dépasser :

- 30 pour un nom-schéma ;
- 16 pour une rubrique-de-description-d'entrées ;
- 31 pour une unité-de-bibliothèque.

Traitement du schéma

- Le nom de l'unité de bibliothèque contenant le schéma résultant doit être identique au nom X utilisé pour le schéma, tel qu'il a été déclaré dans la clause DDL SCHEMA NAME :
 - Le nom de l'entité contenant le schéma résultant: X (unité-de-bibliothèque).

Si plusieurs schémas correspondant au même nom-schéma du DDL sont nécessaires (changement initial, test, production) ils doivent être implantés dans différentes bibliothèques car les noms-d'unités-de-bibliothèque sont uniques dans une bibliothèque.
- Les schémas résultants peuvent se trouver dans deux états différents :
 - Etat "DDL". Cet état correspond à une exécution réussie de la traduction origine DDL. Dans cet état, au schéma résultant n'est accessible qu'aux programmes et demandes de compilation du COBOL et de QUERY.
 - L'état "DDL-DMCL". Cet état correspond à l'exécution réussie d'une traduction origine DMCL. Dans cet état un schéma résultant est accessible aux programmes de compilation du COBOL et de QUERY, aux activités utilisateurs, aux utilitaires IDS/II et à QUERY pour exécution des demandes, commandes et programmes du DML.
- Les rubriques de description d'entrées contenant les éléments SL ne sont pas l'objet d'une affectation statique car DDLPROC les affecte dynamiquement en utilisant le nom de fichier interne INPENC. Les commandes du langage DDLPROC citent les programmes SL en nommant la rubrique-de-description-d'entrées immédiatement précédée d'un astérisque (*).
- L'affectation statique des bibliothèques contenant des éléments SL se fait en utilisant, soit pour le JCL étendu les noms-de-bibliothèques LIB SL INLIB1, INLIB2, INLIB3, soit en JCL de base les noms-de-fichiers-internes (nfi) SLLIB1, SLLIB2, SLLIB3. Dans les deux cas le nom de fichier interne (nfi) utilisée est SLLIB1, SLLIB2, SLLIB3.

Les éléments SL sont cités dans le langage de commande de DDLPROC par :

- Le nom de l'unité de bibliothèque qui les contient. En ce cas, l'élément SL est recherché, dans l'ordre de la séquence SLLIB1, SLLIB2, SLLIB3, dans les bibliothèques affectées.
- Le nom de l'unité de bibliothèque qui les contient. En ce cas, l'élément SL est recherché, dans l'ordre de la séquence SLLIB1, SLLIB2, SLLIB3, dans les bibliothèques affectées.
- Le nom de l'unité de bibliothèque qui les contient représenté, soit par le nom de fichier externe (nfe) d'une bibliothèque, soit par l'un des mots-clés SLLIB1, SLLIB2, SLLIB3. En ce cas, l'élément SL est recherché dans la bibliothèque qui a été spécifiée. La désignation par un mot-clé SLLIBi est nécessaire lorsque les bibliothèques disposent de noms-de-fichier externes (nfe) identiques mais sont rangées sur différents supports.
- L'affectation statique des bibliothèques contenant des éléments DD se fait, soit en JCL étendu par utilisation des mots-clés DDLIB1, DDLIB2, DDLIB3, soit en JCL de base par les noms nfi réservés DDLIB1, DDLIB2, DDLIB3. Dans les deux cas, ce sont les noms de fichiers internes (nfi) DDLIB1, DDLIB2, DDLIB3 qui sont utilisés.

Les éléments résultants DD portent le même nom que l'entité qui les contient. Ils sont cités par le langage de commande de DDLPROC :

- Soit par le nom du schéma (explicitement ou implicitement dérivé du DDL et du DMCL origines). Si l'élément DD est traité en mode extraction ou mise à jour, il est recherché, dans l'ordre défini par la séquence DDLIB1, DDLIB2, DDLIB3, dans les bibliothèques affectées. Si l'élément DD est traité en mode création, il est créé dans l'ordre défini par la séquence DDLIB1, DDLIB2, DDLIB3, dans la première bibliothèque affectée.
- Soit par le nom de schéma (explicite ou implicite) qualifié par le nom de fichier externe (nfe) de la bibliothèque, ou, par l'un des mots-clés DDLIB1, DDLIB2, DDLIB3. En ce cas, l'élément DD est recherché ou créé dans la bibliothèque spécifiée. Si les bibliothèques disposent de noms (nfe) identiques mais sont enregistrées sur plusieurs supports, la qualification par le mot-clé DDLIBi est nécessaire.

La Figure 3-2 ci-après permet d'illustrer ce qui a été décrit dans le paragraphe précédent.

Traitement du schéma

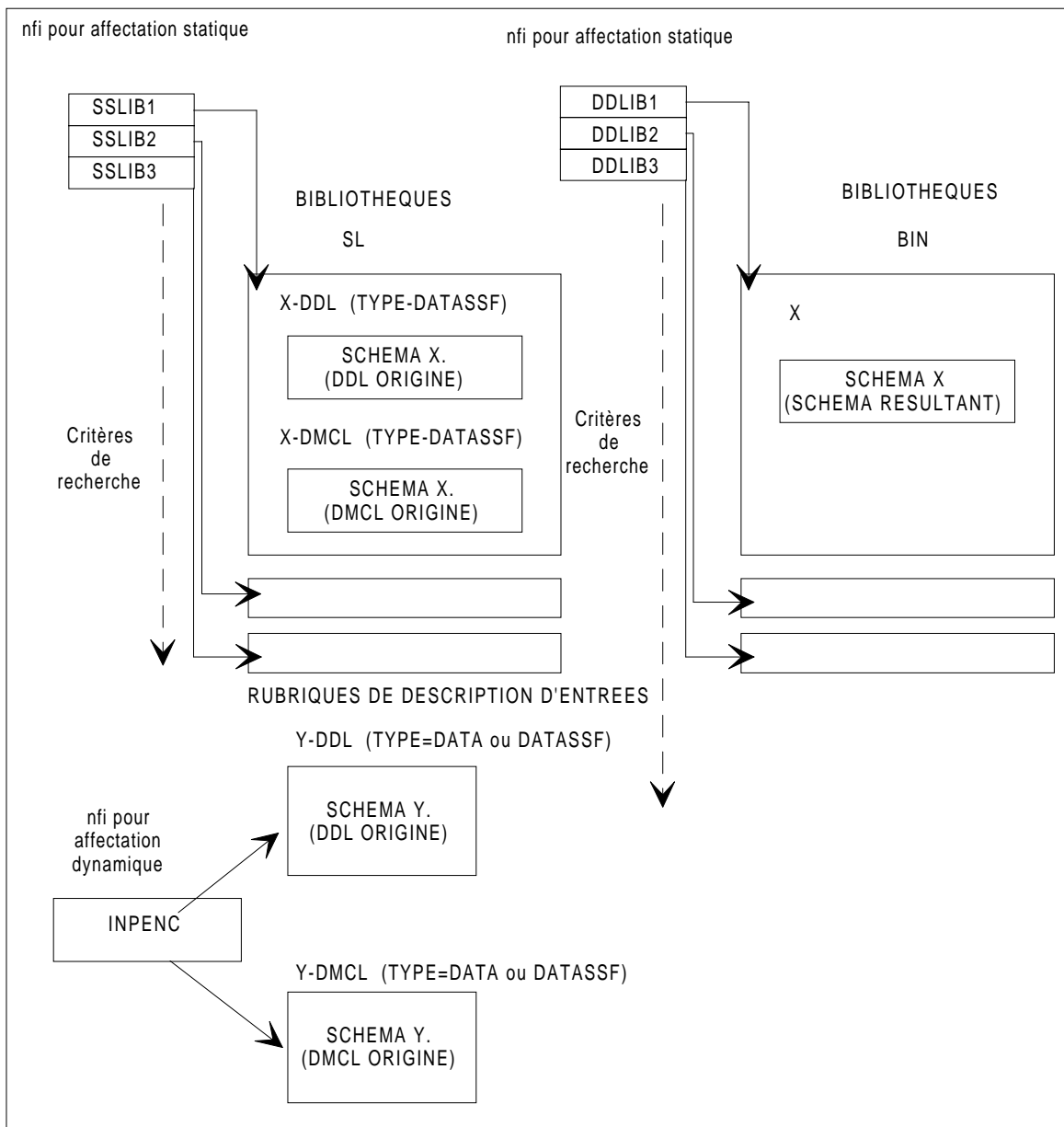


Figure 3-2. Accès du processeur DDLPROC aux bibliothèques SL et aux bibliothèques BIN

3.3 JCL ETENDU DU DDLPROC

Les spécifications JCL suivantes permettent d'appeler DDLPROC :

```
[LIB SL INLIB1=(description-bibliothèque)]
[  [INLIB2=(description-bibliothèque)]  ]
[  [INLIB3=(description-bibliothèque)]; ]

DDLPROC
{COMFILE={*nom-rubrique-description-dacutedonnées}}
{          {(description-fichier-séquentiel)          }}
{          }
{COMMAND=' commande ...'                          }

[DDLIB1=(description-bibliothèque)]
[DDLIB2=(description-bibliothèque)]
[DDLIB3=(description-bibliothèque)]
[PRTFILE=(description-fichier-impression)]
[PRTDEF=(paramètres DEFINE)]
[PRTOUT=(paramètres SYSOUT)]
```

3.3.1 Description des paramètres DDLPROC

LIB SL	Permet de définir les bibliothèques contenant les éléments SL et leur itinéraire de recherche.
COMFILE	Mot-clé suivi d'un groupe de paramètres définissant le fichier d'entrées contenant les commandes DDLPROC. Celles-ci peuvent provenir soit d'une rubrique de description d'entrées (TYPE=DATA ou TYPE=DATASSF), soit d'une unité de bibliothèque (TYPE=DATASSF), ou encore d'un fichier séquentiel.
COMMAND	Mot-clé utilisé à la place de COMFILE pour introduire directement des commandes DDLPROC.
DDLIBi	Ces mots-clés permettent de définir les bibliothèques contenant des éléments DD et leur itinéraire de recherche.
PRTFILE PRTDEF PRTOUT	Ces mots-clés permettent la redéfinition des options d'impression standard spécifiées pour l'état imprimé DDLPROC.

3.4 JCL DE BASE DE DDLPROC

Le JCL de base sert également à appeler DDLPROC :

```
STEP H-DDLPROC SYS [activité-option] ...

{OPTIONS=' {commande} ... ' ; }
{; ASSIGN COMFILE {*nom-rubrique-description-dacuteentrées} ; }
{ (description-fichier-séquentiel) } }

[SIZE 100;]
[ASSIGN SLLIB1 description-bibliothèque;]
[ASSIGN SLLIB2 description-bibliothèque;]
[ASSIGN SLLIB3 description-bibliothèque;]
[ASSIGN DDLIB1 description-bibliothèque;]
[ASSIGN DDLIB2 description-bibliothèque;]
[ASSIGN DDLIB3 description-bibliothèque;]
[ASSIGN PRTPFILE description-fichier-impression;]
[DEFINE PRTPFILE paramètres DEFINE;]
[SYSOUT PRTPFILE SYSOUT paramètres SYSOUT;]
ENDSTEP;
```

3.4.1 Description des paramètres

L'équivalence entre le JCL étendu et le JCL de base est pratiquement complète. Les quelques différences qui existent entre eux sont indiquées ci-dessous.

Le mot-clé COMMAND est remplacé dans le JCL de base par la chaîne OPTIONS.

La spécification LIB SL est remplacée, dans le JCL de base, par ASSIGN SLLIBi.

3.5 LANGAGE DE COMMANDE DDLPROC

Les règles de syntaxe et de représentation sont pratiquement identiques à celles décrites pour les spécifications DDL au chapitre 1.

Une commande est une rubrique composée de plusieurs clauses et se terminant par un point suivi d'un espace ou d'une fin de ligne. A la différence cependant des rubriques DDL, une commande DDL PROC peut aussi se terminer par un point virgule (;). Le point virgule, dans ces conditions, ne peut plus être utilisé comme caractère de ponctuation dans le texte de la commande. A l'exception de la première, qui identifie la rubrique, les clauses ne suivent aucun ordre de classement particulier.

Un nfe de bibliothèque qui ne peut se conformer aux règles d'écriture DDL :

- Peut être écrit sans protection à condition de n'utiliser dans le jeux de caractères illégaux que le point (exemple: PRINT SCHEMA INVENTORY OF MIDS.OD).
- Doit être écrit dans sa forme "échappement" (\$...\$) dans tous les autres cas.

3.5.1 Mots réservés

Les mots réservés à DDLPROC sont indiqués ci-dessous :

CHECK	QUERY
COBOL	RECORD (RECORDS)
COMPACT	REPLACE
CONVERSION	REPORT
DATA	RESET
DDL	RETRIEVAL
DDL1B1	SCHEMA
DDL1B2	SEVERITY
DDL1B3	SLL1B1
DEBUG	SSL1B2
DELETE	SSL1B3
DESCRIPTION	SOURCE
DMCL	STATEMENTS
DML	STATUS
EXCEPT	STORAGE
FOR	TIME
FORCED	TO
IN (OF)	TRANSLATE
IS	UPDATE
MOVE	USE
OBJECT	UWA
OF (IN)	XREF
PRINT	

3.5.2 Commandes

Il existe six types de commandes. Cinq d'entre elles correspondent aux cinq fonctions qui ont été indiquées dans la Figure 3-1. La dernière se rapporte à la gestion du flux des commandes lorsque plusieurs commandes sont introduites lors du même passage.

Les commandes décrites plus loin sont listées ci-dessous :

- TRANSLATE SCHEMA DDL

- TRANSLATE DMCL

- PRINT SCHEMA

- DELETE SCHEMA

- MOVE SCHEMA

- STATUS

3.6 COMMANDE TRANSLATE SCHEMA DDL

Fonction

Permet de traduire le DDL origine du schéma et de créer éventuellement un schéma résultant à l'état "DDL".

Format général

TRANSLATE SCHEMA DDL

```

                                {*nom-rubrique-description-d'entrées
                                {
SOURCE IS {nom-unité-bibliothèque [ {IN} {SLLIB1} ] } }
                                [ {OF} {SLLIB2} ] ] }
                                [ {SLLIB3} ] ] }

[ [ {nfe-de-bibliothèque} ] ]
[OBJECT [ {IN} {DDLIB1} ] ] [REPLACE]]
[ [ {OF} {DDLIB2} ] ] ]
[ [ {DDLIB3} ] ] ]

[REPORT SEVERITY entier ]
[ XREF ]

[FORCED DATE IS {aa mm jj} TIME IS hh mm]
[ {aa jjj } ]
[FOR CONVERSION]

[CHECK FOR COBOL USE]
[ QUERY ] .

```

Règles

1. La clause SOURCE permet de définir la rubrique de description d'entrées ou l'unité de bibliothèque contenant les spécifications DDL.
2. Si la clause OBJECT est spécifiée, il y a création d'un schéma résultant à l'état DDL, à condition que le DDL origine soit correct.

Si les critères de recherche se rapportant au mode création ne sont pas utilisables, on peut spécifier la bibliothèque en sortie.

Si le but de l'opération est de remplacer dans la bibliothèque en sortie une unité qui porte déjà un nom courant, le mot-clé REPLACE doit être spécifié dans la clause OBJECT. Dans le cas contraire il se produira une erreur.

3. Si la clause OBJECT n'est pas spécifiée, aucun schéma résultant n'est créé. Seul le listage du DDL est produit.

Traitement du schéma

4. Si la clause REPORT est spécifiée, elle peut être accompagnée de quelques options. L'option SEVERITY indique le niveau minimum de diagnostic à produire. Il existe cinq niveaux de gravité numérotés 0, 1, 2, 3, 4. Le niveau 0 correspond aux messages tel que "SYNTAX ANALYSIS RESUMED" (reprise de l'analyse de syntaxe)..., les niveaux 1 et 2 correspondent à des avertissements. Le niveau 3 est celui des messages signalant les erreurs de syntaxe bloquantes et le niveau 4 celui des messages d'erreurs bloquantes du cadre d'exploitation.

"Entier" peut prendre la valeur 0, 1, 2, 3. Par exemple, si on écrit SEVERITY 3, seuls les messages d'erreurs bloquantes seront indiqués.

Si l'option SEVERITY n'est pas spécifiée, le niveau 0 est pris par défaut.

L'option XREF demande qu'un listage des références croisées des noms définis par l'utilisateur soit produit.

Si l'option XREF n'est pas choisie il n'y a aucun listage des références croisées.

5. Si la clause REPORT n'est pas choisie, l'option de gravité SEVERITY 0 est prise par défaut et il n'y a pas de référence croisée XREF.
6. Si la clause FORCED DATE n'est pas spécifiée, DDLPROC attribue au schéma résultant créé le groupe date-heure de référence DDL qui correspond au groupe date-heure courant.

Remarque : Il est à noter que les contrôles de cohérence globale effectués au moment de l'exécution entre le groupe date-heure de référence DDL d'un schéma et le groupe date-heure de référence DDL des programmes qui ont été compilés ne seront effectives que si cette clause a été omise.

7. Si la clause FORCED DATE est spécifiée, c'est le groupe date-heure indiqué qui est attribué au schéma résultant créé.

La date peut être exprimée sous la forme année, mois et jour (aa mm jj) ou en quantième de jour de l'année (aa jjj). L'heure s'exprime en heure et minute (hh mm). L'omission des zéros de tête est admise.

Le fait que le groupe date-heure a été forcé est enregistré dans le schéma résultant.

Lorsque, pour des raisons de compatibilité, le schéma utilisé dans la version 1D est retraduit en version GCOS 7, on peut spécifier la locution FOR CONVERSION si le groupe date-heure de référence du schéma 1D n'a pas été imposé. En ce cas, l'imposition du nouveau groupe date-heure n'est pas enregistrée dans le schéma.

A part le cas de conversion de 1D à GCOS 7 décrit plus haut, l'utilisation de la clause FORCED DATE doit être restreinte aux deux circonstances suivantes :

- Cas où le schéma résultant est détruit et des programmes compilés vont être amenés à citer le schéma: de manière à permettre à ces programmes de continuer à travailler sans une nouvelle compilation, le DDL origine est traité avec l'ancien groupe date-heure de DDL.

 - Cas où les mêmes programmes DML doivent tourner sur une base de données production et une base de données test de taille plus petite. Le schéma de test doit avoir le même DDL que le schéma de production et doit donc porter le même groupe date-heure de référence DDL. Le DMCL lui agit différemment, car il tient compte des variations de taille des aires. Noter que l'on peut éviter l'utilisation de l'option FORCED DATE si l'on transfère dans un premier temps le schéma de production dans une autre bibliothèque (voir commande MOVE) et que l'on applique le DMCL à cette copie.
8. Si la clause CHECK est spécifiée, DDLPROC vérifiera que les noms définis par l'utilisateur ne sont pas des mots réservés au COBOL et à QUERY.
 9. Si la clause CHECK n'est pas spécifiée, aucune vérification n'est effectuée.
 10. La Figure 3-3 illustre un état imprimé, comportant quelques erreurs, de traduction du DDL de schéma.

Traitement du schéma

DDLPROC	30.00	X1167.2	DOCLIST	FAYOT	IDS2	IDS2__R5
TRANSLATE	SCHEMA	COMMAND	REPORT	SOURCE	LISTING	AND ERRORS
ILN	XLN	TEXT				
			18:15:33MAY 22, 1980	PAGE	4	
1	10	COMMENT "*****"				
2	20	COMMENT "*" SCHEMA DDL FOR TEST-GROUP *TG19* *				
3	30	COMMENT "***** **"				
4	40					
5	50	COMMENT "*****"				
6	60	COMMENT "*" SCHEMA *				
7	70	COMMENT "*****"				
8	80					
9	90	SCHEMA TG19-SH.				
10	100					
11	110	COMMENT "*****"				
12	120	COMMENT "*" AREAS *				
13	130	COMMENT "*****"				
14	140					
15	150	AREA TG19-A00.				
16	160	AREA TG19-A01.				
17	170					
18	180	COMMENT "*****"				
19	190	COMMENT "*" RECORDS *				
20	200	COMMENT "*****"				
21	210					
22	220	COMMENT "*****"				
23	230	COMMENT "*" R01 (PART) *				
24	240	COMMENT "*****"				
25	250	RECORD R01				
26	260	LOCATION CALC USING R01-NUM DUP NOT				
27	270	WITHIN ANY AREA .				
		1				
***	1	3-03-52	AREA	-ID	PHRASE	MISSING

Figure 3-3. Etat imprimé de traduction du DDL de schéma (1/3)

Manuel de référence IDS/II

```

28      280      02      R01-IDENT      TYPE CHAR 7.
29      290      02      R01-NUHN      TYPE CHAR 5.
30      300      02      R01-NUM      TYPE UNPACKED DEC 3.
31      310      02      R01-TRAILER  TYPE CHAR 4.
32      320
33      330      COMMENT "*****"
34      340      COMMENT "*      R02 (COMPONENT)  *"
35      350      COMMENT "*****"
36      360      RECORD R02
37      370          LOCATION VIA S01
38      380          WITHIN AREA OF OWNNER
39      390      02      R02-IDENT      TYPE CHAR 7.
          1
          2
*** 1 3-00-02 CLAUSE NOT RECONNIZED WITHIN THE CURRENT (SUB)ENTRY - DATA SKIPPED TILL
          NEXT RECORD SUBENTRY CLAUSE OR PERIOD
          2 0-00-50 SYNTAX ANALYSIS RESUMED AT THIS POINT

          40      400      02      R02-R01-UPH  TYPE CHAR 4.
          41      410      02      R02-R01-UP   TYPE UNPACKED DEC 3.
          42      420      02      R02-R01-DOWNH TYPE CHAR 6.
          43      430      02      R02-R01-DOWN  TYPE UNPACKED DEC 3.
          44      440      02      R02-QTYH   TYPE CHAR 5.
          45      450      02      R02-QTY    TYPE UNPACKED 3.
          12
*** 1 5-07-09 MANDATORY KEYWORD MISSING : DECIMAL - TYPE CLAUSE ANALYSIS DISCONTINUED

DDLPROC   30.00 X1167.2  DOCLIST  FAYOT      IDS2      IDS2__R5
          18:15:33MAY 22, 1980  PAGE      5
          TRANSLATE SCHEMA COMMAND  REPORT      SOURCE LISTING AND ERRORS
          ILN      XLN      TEXT

*** 2 0-00-50 SYNTAX ANALYSIS RESUMED AT THIS POINT

          46      460      02      R02-TRAILER  TYPE CHAR 4.
          47      470
          48      480      COMMENT "*****"
          49      490      COMMENT "*      SETS      *"
          50      500      COMMENT "*****"
          51      510
          52      520      COMMENT "*****"
          53      530      COMMENT "*      S01 (CALL-OUT)  *"
          54      540      COMMENT "*****"
          55      550      SET S01
          56      560          OWNER  R01
          57      570          ORDER  PERMANENT INSERTION SORTED DEFINED KEYS DUP NOT.
          58      580          MEMBER R02
          59      590          INSERTION AUTO RETENTION MAND
          60      600          KEY AS CENDING R02-R01-DOWN
          61      610          SELECTION THRU S01
          62      620          OWNER IDENTIFIED BY CALC-KET R01-NUM EQUAL TO R02-R01-
          UP.
          63      630
          64      640      COMMENT "*****"
          65      650      COMMENT "*      S02 (WHERE-USED)  *"
          66      660      COMMENT "*****"
          67      670      SET S02
          68      680          OWNER  R01
          69      690          ORDER  PERMANENT INSERTION SORTED DEFINED KEYS .
          1
*** 1 4-03-07 MANDATORY KEYWORD MISSING : DUPLICATES DISPOSITION - ORDER CLAUSE
          ANALYSISDISCONTINUED

```

Figure 3-3. Etat imprimé de traduction du DDL de schéma (2/3)

Traitement du schéma

```
1 0-00-50 SYNTAX ANALYSIS RESUMED AT THIS POINT
      70      700 MEMBER R02
      71      710 INSERTION AUTO RETENTION MAND
      72      720 KEY R02-R01-UP
                1
*** 1 6-04-02 MANDATORY KEYWORD MISSING : ASCENDING/DESCENDING - KEY CLAUSE ANALYSIS
      DISCONTINUED
      73      730 SELECTION THRU S02
                1
1 0-00-50 SYNTAX ANALYSIS RESUMED AT THIS POINT
                1
      74      740 OWNER IDENTIFIED BY CALC-KEY R01-NUM EQUAL TO R02-R01-
      DOWN
      75      750 AREA-ID EQUAL TO OTHER-ARID1.
      76      760
      77      770 END-SCHEMA.
```

Figure 3-3. Etat imprimé de traduction du DDL de schéma (3/3)

3.7 COMMANDE TRANSLATE DMCL

Fonction

Permet de traduire le DMCL origine du schéma et de le ranger éventuellement dans le schéma résultant.

Format général

TRANSLATE DMCL

```

          {*nom-rubrique-description-d'entrées
          {
SOURCE IS {nom-unité-bibliothèque
          [ {IN} {nfe-de-bibliothèque}]]
          [ {SLLIB1} ]]]
          [ {OF} {SLLIB2} ]]]
          [ {SLLIB3} ]]]

[ {nfe-de-bibliothèque} ]
[ SCHEMA {IN} {DDLIB1} ] ]
[ {OF} {DDLIB2} ] ]
[ {DDLIB3} ] ]

[ OBJECT [REPLACE]]
[ REPORT SEVERITY entier]

[ FORCED DATE IS {aa mm jj} TIME IS hh mm]
[ {aa jj} ] .
[ [FOR CONVERSION] ] ]
    
```

Règles

1. La clause SOURCE permet de définir la rubrique de description d'entrées ou l'unité de bibliothèque contenant les spécifications DMCL.
2. Si la clause SCHEMA n'est pas spécifiée, le nom du schéma résultant, dérivé de celui indiqué dans la clause SCHEMA NAME du DMCL est extrait en fonction des critères de recherche.
3. Si la clause SCHEMA est spécifiée, le schéma résultant est extrait de la bibliothèque mentionnée.
4. Avant tout traitement, le schéma résultant doit se trouver à l'état "DDL" ou à l'état "DDL-DMCL".
5. Si la clause OBJECT est spécifiée, le schéma résultant est mis à jour en fonction des informations fournies par le DMCL, sous réserve que le DMCL origine lui-même soit correct.

Le mot-clé REPLACE doit être spécifiée si le schéma résultant se trouve déjà à l'état "DDL-DMCL", c'est-à-dire s'il contient des informations DMCL.

Lorsque le traitement se déroule normalement, le schéma résultant est laissé à l'état "DDL-DMCL".

6. Lorsque la clause OBJECT n'est pas spécifiée, le schéma résultant est laissé tel quel. Seul l'état imprimé du DMCL est produit.

Traitement du schéma

7. Si la clause REPORT est spécifiée, le niveau de diagnostic est égal ou supérieur à celui indiqué par "entier". (Voir la commande TRANSLATE SCHEMA DDL pour plus de détails).
8. C'est le degré de gravité SEVERITY 0 qui est implicite lorsque la clause REPORT n'est pas spécifiée.
9. Si la clause FORCED DATE n'est pas spécifiée, le groupe date-heure de référence DMCL équivalent au groupe date-heure courant, est imposé au schéma résultant.

Noter que les contrôles de cohérence entre le groupe date-heure de référence DMCL du schéma et le groupe date-heure de référence DMCL des aires réservées pour le traitement ne sont effectués que si cette clause est omise.

10. Lorsque la clause FORCED DATE est spécifiée, c'est le groupe date-heure de référence DMCL indiqué qui est imposé au schéma résultant.

La date s'exprime, soit sous la forme année, mois et jour (aa mm jj), soit sous la forme du quantième de jour de l'année (aa jjj). L'heure est donnée en heure et minute (hh mm). Les zéros de remplissage peuvent être omis.

Le fait que le groupe date-heure a été imposé est enregistré dans le schéma résultant.

Lorsque pour des raisons de compatibilité, un schéma utilisé en version 1D est retraduit sous GCOS 7, la locution FOR CONVERSION doit être spécifiée si le groupe date-heure de référence du schéma 1D n'a pas été imposé. Dans ce cas particulier, le fait que le nouveau groupe date-heure a été imposé n'est pas enregistré dans le schéma.

Hormis les cas de conversion de la version 1D à GCOS 7, l'utilisation de la clause FORCED DATE n'est conseillée que pour les deux cas suivants :

- Lorsque le schéma a été détruit, et que des aires chargées en mémoire citent ce schéma. De manière à éviter un nouveau chargement de ces aires, la traduction de l'origine DMCL se fait avec l'ancien groupe date-heure de référence du DMCL.
 - Lorsque, dans un but d'accélération du chargement initial de la base de données, un schéma chargeable dispose d'un DDL différent de celui du schéma de production à cause de certains critères de sélection d'ensembles ou de classement d'ensembles. Le schéma en cours de chargement, comme il fait référence à des aires de la production, contient le même DMCL et doit donc se voir attribuer le groupe date-heure de référence DMCL du schéma de production.
11. La Figure 3-4 illustre un état imprimé, contenant quelques erreurs, de traduction DMCL.

Manuel de référence IDS/II

```

DDLPROC      30.00  X1167.2  DOCLIST  FAYOT      IDS2      IDS2__R5      18:15:33
              MAY 22, 1980      PAGE      8
TRANSLATE DMCL  COMMAND  REPORT      SOURCE LISTING AND ERRORS
  ILN      XLN      TEXT
  1         10  COMMENT "*****"
  2         20  COMMENT "*"          SCHEMA DMCL FOR TEST-GROUP *TG19*  *"
  3         30  COMMENT "*****"
  4         40
  5         50  COMMENT "*****"
  6         60  COMMENT "*"          SCHEMA          *"
  7         70  COMMENT "*****"
  8         80
  9         90  SCHEMA TG19-SH.
 10        100
 11        110 COMMENT "*****"
 12        120 COMMENT "*"          AREAS          *"
 13        130 COMMENT "*****"
 14        140
 15        150 AREA TG19-A00
 16        160          NUMBER-OF-PAGES 53.
 17        180          PAGE-SIZE 2040.

*** 2-00-24 MANDATORY LINES-PER-PAGE CLAUSE MISSING

 18        190
 19        200 AREA TG19-A01
 20        210          NUMBER-OF-PAGES 52.
 21         0          CALC-INTERVAL 3.
                               1

*** 1 2-06-60 CALC-INTERVAL VALUE MUST DIVIDE THE AREA NUMBER OF PAGES

 22        220          LINES-PER-PAGE 45
 23        230          PAGE-SIZE 2040.
                               1 2

*** 1 2-05-17 PAGE-SIZE VALUE MUST BE AN INTEGER NUMBER OF TIMES 256 BYTES - CLAUSE
          IGNORED
  2 0-00-50 SYNTAX ANALYSIS RESUMED AT THIS POINT

 24        240
 25        250 END-DMCL.

```

Figure 3-4. Etat imprimé de traduction DMCL

3.8 COMMANDE PRINT SCHEMA

Fonction

Permet d'obtenir un état imprimé des caractéristiques du rangement de la base de données et un état imprimé des règles de programmation DML.

Format général

```
PRINT SCHEMA nom-schéma [ {IN} {DDL1B1} ]
                        [ {OF} {DDL1B2} ]
                        [ {DDL1B3} ]
```

STORAGE DESCRIPTION [COMPACT]

```
[ [FOR RECORDS [EXCEPT] {nom-article} ... ] ]
[
[ DML [ UWA DESCRIPTION ] ]
[ [ [ UPDATE ] STATEMENTS ] ]
[ [ [ RETRIEVAL ] ] ]
[ [FOR RECORD [EXCEPT] {nom-article} ... ] ] ] .
```

Règles

1. Si la clause STORAGE DESCRIPTION est spécifiée, un état imprimé des caractéristiques du rangement de la base de données est produit.

Cet état contient :

- Des informations générales sur les dates de référence du schéma, le nombre d'aires, les types d'articles, les types d'ensembles, les zones, la taille et la segmentation du schéma en mémoire centrale, la taille et le format des pointeurs globaux de la base de données, etc.
 - Des informations particulières à un article mémoire: numéro de l'article, numéros des ensembles auxquels il est rattaché, numéros de ses zones, agencement de l'article (en-tête, zone pointeurs et zone données), les caractéristiques des aires mémoire contenant cet article.
2. Si l'on omet la locution FOR RECORDS, l'état STORAGE fournit des informations sur tous les articles du schéma.
 3. Si FOR RECORDS est spécifié, l'état STORAGE fournit des informations :
 - sans EXCEPT: sur tous les articles de la liste ;
 - avec EXCEPT: sur tous les articles du schéma sauf ceux de la liste.

4. Si l'on omet le mot-clé COMPACT, les informations sur les groupes répétitifs et les vecteurs sont fournies pour chaque apparition de leurs éléments.
5. Si COMPACT est spécifié, les informations portant sur les groupes répétitifs et les vecteurs ne sont pas détaillées par apparition de leurs éléments.
6. Que COMPACT soit spécifié ou non, la longueur indiquée pour un groupe est celle de l'apparition de ce groupe.

En ce qui concerne le décalage d'un groupe, celui-ci équivaut à l'apparition du groupe si COMPACT n'a pas été spécifié. Si COMPACT est spécifié, c'est le décalage de la première apparition de ce groupe (valeur 1 pour tous les index) qui est indiqué.

7. Si la clause DML est spécifiée, un état imprimé fournit les règles de programmation COBOL pour ce schéma.

Cet état contient, sur la base du type d'article, deux genres d'informations :

- La description COBOL dans l'article UWA et les ensembles auxquels celui-ci participe.
- Les verbes COBOL DML utilisables dans le traitement de cet article et les diverses zones qui doivent être initialisées avant que ces verbes ne soient utilisés.

8. Si la locution UWA est spécifiée, l'état contient la description COBOL de l'article et les diverses informations de contrôle.
9. Si la locution STATEMENT est spécifiée, l'état contient les verbes COBOL DML.

Avec UPDATE, la liste des verbes de mise à jour est fournie.

Avec RETRIEVAL, les verbes d'extraction sont listés.

Lorsque ni UPDATE ni RETRIEVAL ne sont fournis, tous les verbes sont listés.

10. Lorsque ni UWA, ni STATEMENTS est spécifié, l'état contient les descriptions COBOL et tous les verbes DML.
11. Si la locution FOR RECORDS n'est pas spécifiée, l'état DML fournit des informations sur tous les articles du schéma.
12. Si FOR RECORDS est spécifié, l'état DML fournit des informations :
 - sans EXCEPT: sur tous les articles de la liste;
 - avec EXCEPT: sur tous les articles du schéma sauf ceux de la liste.
13. Lorsque ni la clause STORAGE, ni la clause DML n'est spécifiée, toutes les options sont utilisées dans l'impression des deux états.
14. L'état STORAGE n'est imprimé que si le schéma résultant est à l'état "DDL-DMCL". L'état DML est imprimé que le schéma résultant soit à l'état "DDL" ou "DMCL".

Traitement du schéma

15. La Figure 3-5, plus loin, illustre un état imprimé STORAGE sans option COMPACT. La Figure 3-6 illustre un état STORAGE avec option COMPACT. Enfin la Figure 3-7 montre un listage DML.

Manuel de référence IDS/II

```

* SCHEMA TG05-SH * * PRINT STORAGE REPORT OF RECORD R17 *
RECORD CODE NUMBER : 17
          S T O R A G E   R E C O R D   L A Y   -   O U T
*****
* LEVEL   *          FIELD DESCRIPTION          * CODE *SIZE IN BYTES*
*         *          OFFSET(HEXA)* OFFSET(HEXA)*
*         *          OR FIELD NAME             * NUMBER * (DECIMAL) * /
* RECORD  *          /USER DATA              *
*****
* HEADER  * UFAS HEADER                          *         *         4 *
*         *          * 0000 *
* HEADER  * STATUS                              *         *         1 *
* 0004    *
* POINTER * MEMBERSHIP GLOBAL POINTERS FOR SET: *         *         *
*         * S07 * 7 * 6 *
* 0005    *
*****
* 1       * R17-IDENT                            * 0 * 17 *
* 000B    * 0000 *
* 2       * R17-IDENTH                          * 1 * 7 *
* 000B    * 0000 *
* 2       * R17-CH                              * 2 * 3 *
* 0012    * 0007 *
* 2       * R17-C                              * 3 * 2 *
* 0015    * 000A *
* 2       * R17-FH                              * 4 * 3 *
* 0017    * 000C *
* 2       * R17-F                              * 5 * 2 *
* 001A    * 000F *
* 1       * R17-S2H                             * 6 * 9 *
* 001C    * 0011 *
* 1       * R17-S2GROUP1(1)                     * 7 * 5 *
* 0025    * 001A *
* 2       * R17-ITEM1(1)                       * 8 * 3 *
* 0025    * 001A *
* 2       * R17-S2(1,1)                         * 9 * 1 *
* 0028    * 001D *
* 2       * R17-S2(1,2)                       * 9 * 1 *
* 0029    * 001E *
* 1       * R17-S2GROUP1(2)                    * 7 * 5 *
* 002A    * 001F *
* 2       * R17-ITEM1(2)                       * 8 * 3 *
* 002A    * 001F *
* 2       * R17-S2(2,1)                       * 9 * 1 *
* 002D    * 0022 *
* 2       * R17-S2(2,2)                       * 9 * 1 *
* 002E    * 0023 *
* 1       * R17-S1H                            * 10 * 4 *
* 002F    * 0024 *
* 1       * R17-S1H                            * 11 * 1 *
* 0033    * 0028 *
* 1       * R17-FILLER                         * 12 * 120 *
* 0034    * 0029 *
* 1       * R17-TRAILER                       * 13 * 4 *
* 00AC    * 00A1 *
*****
*          HEADER SIZE : 5 *
*          POINTER ZONE SIZE : 6 *
*          DATA ZONE SIZE : 165 *
*          RECORD TOTAL LENGTH : 175 *
*****
          C O N T A I N I N G   A R E A   (   S   )
*****
* AREA NAME * CODE *PAGE SIZE* RANGE IN * RANGE IN *
* LINES * LOCAL PTR SIZE* *NUMBER* IN BYTES* PAGES *
* AREA-KEYS *PERPAGE *IN BYTES* BITS*
*****
* TG05-A01 * 1 * 1024 * 0,120 * 0,1200 *
* 10 * 2 * 11 *
*****

```

Figure 3-5. Etat imprimé STORAGE sans option COMPACT

Traitement du schéma

```

* SCHEMA TG05-SH * * PRINT STORAGE REPORT OF RECORD
* R17 *
RECORD CODE NUMBER : 17
          S T O R A G E   R E C O R D   L A Y - O U T
*****
* LEVEL * FIELD DESCRIPTION * CODE *SIZE IN BYTES*
* OFFSET(HEXA)* OFFSET(HEXA)*
* * OR FIELD NAME * NUMBER * (DECIMAL) * /
* RECORD * /USER DATA *
*****
* HEADER * UFAS HEADER * * 4 *
* 0000 * * *
* HEADER * STATUS * * 1 *
* 0004 * * *
* POINTER * MEMBERSHIP GLOBAL POINTERS FOR SET: * * *
* * S07 * 7 * 6 *
*0005 * *
*****
* 1 * R17-IDENT * 0 * 17 *
* 000B * 0000 *
* 2 * R17-IDENTH * 1 * 7 *
*000B * 0000 *
* 2 * R17-CH * 2 * 3 *
*0012 * 0007 *
* 2 * R17-C * 3 * 2 *
*0015 * 000A *
* 2 * R17-FH * 4 * 3 *
*0017 * 000C *
* 2 * R17-F * 5 * 2 *
*001A * 000F *
* 1 * R17-S2H * 6 * 9 *
*001C * 0011 *
* 1 * R17-S2GROUP1 * 7 * 5 *
*0025 * 001A *
* 2 * R17-ITEM1 * 8 * 3 *
*0025 * 001A *
* 2 * R17-S2 * 9 * 1 *
*0028 * 001D *
* 1 * R17-S1H * 10 * 4 *
*002F * 0024 *
* 1 * R17-S1 * 11 * 1 *
*0033 * 0028 *
* 1 * R17-FILLER * 12 * 120 *
*0034 * 0029 *
* 1 * R17-TRAILER * 13 * 4 *
*00AC * 00A1 *
*****
* HEADER SIZE : 5 *
* POINTER ZONE SIZE : 6 *
* DATA ZONE SIZE : 165 *
* RECORD TOTAL LENGTH : 175 *
*****
          C O N T A I N I N G   A R E A   (   S   )
*****
* AREA NAME * CODE *PAGE SIZE* RANGE IN * RANGE IN *
* LINES *LOCAL PTR SIZE* *NUMBER* IN BYTES* PAGES * AREA-KEYS *
*PERPAGE *IN BYTES* BITS*
*****
* TG05-A01 * 1 * 1024 * 0,120 * 0,1200 *
*10 * 2 * 11 *
*****

```

Figure 3-6. Etat imprimé STORAGE avec option COMPACT

Manuel de référence IDS/II

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT    IDS2      IDS2__R5    08:02:18
AUG 20, 1980 PAGE      19
* SCHEMA TG24-SH *
* PRINT DML REPORT OF RECORD R16 *

*****
*          DML RECORD CHARACTERISTICS          *
*****

*****
*          UWA RECORD DESCRIPTION            *
*****

01 R16.
02 R16-IDENT PIC X(7).
02 R16-CREATE-IC .
03 R16-CREATE-CATEH PIC X(10).
03 R16-CREATE-CATE PIC 9(6).
03 R16-CREATE-TIMEH PIC X(10).
03 R16-CREATE-TIME PIC 9(8).
03 R16-CREATE-TESTH PIC X(10).
03 R16-CREATE-TEST PIC X(4).
03 R16-CREATE-SIMUH PIC X(10).
03 R16-CREATE-SIMU PIC X(12).
02 R16-MODIFY-IC .
03 R16-MODIFY-CATEH PIC X(10).
03 R16-MODIFY-CATE PIC 9(6).
03 R16-MODIFY-TIMEH PIC X(10).
03 R16-MODIFY-TIME PIC 9(8).
03 R16-MODIFY-TESTH PIC X(10).
03 R16-MODIFY-TEST PIC X(4).
03 R16-MODIFY-SIMUH PIC X(10).
03 R16-MODIFY-SIMU PIC X(12).
02 R16-CALCH PIC X(6).
02 R16-CALC.
03 R16-CALC-OCC PIC 9(2).
03 R16-CALC-NUM PIC 9(3).
02 R16-MODIFY-DATAH PIC X(10).
02 R16-MODIFY-DATA PIC 9(8).
02 R16-TRAILER PIC X(4).

*****
*          CONTROL FIELDS                    *
*****

CALC-KEYS ITEMS (CUP NOT)
  R16-CALC-OCC
  R16-CALC-NUM
SORT KEY ITEMS(DUF NOT) IN SET S10 :
  R16-CALC-OCC          ASCENDING
  R16-CALC-NUM          ASCENDING
SORT KEY ITEMS(DUP NOT) IN SET S11 :
  R16-CALC-OCC          DESCENDING
  R16-CALC-NUM          DESCENDING

*****
*          SET PARTICIPATION                *
*****

OWNER OF SETS :
```

Figure 3-7. Listage DML (1/14)

Traitement du schéma

```
DDLPROC      20.00 X1864.1 DOCLIST  FAYOT    IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      20
* SCHEMA    TG24-SH *                  * PRINT DML REPORT OF RECORD R16*

S13
MEMBER OF SETS :
S10
  ORDER SORTED BY DEFINED KEYS DUP NOT
  INSERTION AUTOMATIC      RETENTION MANDATORY
S11
  ORDER SORTED BY DEFINED KEYS DUP NOT
  INSERTION AUTOMATIC      RETENTION MANDATORY

*****
*   LOCATION   *
*****

LOCATION MODE  CALC

REALMS :
  TG24_A02
  TG24-A05
  TG24-A08
  TG24-A11
  TG24-A14
AREA-ID DB-PARAMETER : ARID16
```

Figure 3-7. Listage DML (2/14)

Manuel de référence IDS/II

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT    IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      21
* SCHEMA    TG24-SH *                    * PRINT DML REPORT OF RECORD R16*

*****
*   DML RETRIEVAL STATEMENTS (FORMAT AND PREREQUISITE ACTION OR CONDITION)   *
*****

*****
*   FIND (WITHIN DATA BASE) *
*****

FIND |R16|   DB-KEY IS IDENTIFIER   |RETAINING . . .|
* INITIALIZE IDENTIFIER WITH A DATA-BASE-KEY

FIND ANY R16 |RETAINING . . . |
* INITIALIZE AREA-ID DB-PARAMETER ARID16 WITH ANY OF :
  TG24-A02
  TG24-A05
  TG24-A08
  TD24-A11
  TG24-A14
* INITIALIZE CALC-KEY ITEMS :
  R16-CALC-OCC
  R16-CALC-NUM

FIND CURRENT |RETAINING . . .|
* CURRENT-OF-RUN-UNIT MUST REFERENCE A RECORD OF THE SAME TYPE

*****
*   FIND (WITHIN RECORD-TYPE) *
*****

FIND CURRENT R16 |RETAINING . . .|
* CURRENT-OF-RECORD-TYPE MUST REFERENCE A RECORD

*****
*   FIND (WITHIN REALM WHERE RECORD IS LOCATED) *
*****

FIND FIRST      |R16| WITHIN REALM-NAME |RETAINING . . .|
FIND LAST       |R16| WITHIN REALM-NAME |RETAINING . . .|
FIND INTEGER    |R16| WITHIN REALM-NAME |RETAINING . . .|
FIND IDENTIFIER |R16| WITHIN REALM-NAME |RETAINING . . .|
  WHERE REALM-NAME IS ANY OF :
    TG24-A02
    TG24-A05
    TG24-A08
    TG24-A11
    TG24-A14
* INITIALIZE IDENTIFIER IF FORMAT 4 IS USED

FIND NEXT |R16| WITHIN REALM-NAME |RETAINING . . .|
FIND PRIOR |R16| WITHIN REALM-NAME |RETAINING . . .|
  WHERE REALM-NAME IS ANY OF :
    TG24-A02
    TG24-A05
    TG24-A08
    TG24-A11

```

Figure 3-7. Listage DML (3/14)

Traitement du schéma

```

DDLPROC      20.00  X1864.1  DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      22
* SCHEMA    TG24-SH *                  * PRINT DML REPORT OF RECORD R16 *

      TG24-A14
* CURRENT-OF-REALM MUST REFERENCE A RECORD OR BE VIRTUAL

FIND CURRENT |R16| WITHIN REALM-NAME |RETAINING . . . |
  WHERE REALM-NAME IS ANY OF :
      TG24-A02
      TG24-A05
      TG24-A08
      TD24-A11
      TG24-A14
* CURRENT-OF-REALM MUST REFERENCE A RECORD OF THE SAME TYPE

*****
*   FIND (WITHIN SET OF WHICH RECORD IS OWNER)   *
*****

FIND OWNER WITHIN S13 |RETAINING . . . |
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL
FIND CURRENT |R16| WITHIN S13 |RETAINING . . . |
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD(OWNER)

*****
*   FIND WITHIN SET OF WHICH RECORD IS MEMBER)   *
*****

FIND FIRST      |R16| WITHIN SET-NAME |RETAINING . . . |
FIND LAST       |R16| WITHIN SET-NAME |RETAINING . . . |
FIND INTEGER    |R16| WITHIN SET-NAME |RETAINING . . . |
FIND IDENTIFIER |R16| WITHIN SET-NAME |RETAINING . . . |
FIND NEXT       |R16| WITHIN SET-NAME |RETAINING . . . |
FIND PRIOR      |R16| WITHIN SET-NAME |RETAINING . . . |
  WHERE SET-NAME IS ANY OF :
      S10
      S11
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL
* INITIALIZE IDENTIFIER IF FORMAT 4 IS USED

FIND CURRENT |R16| WITHIN SET-NAME |RETAINING . . . |
  WHERE SET-NAME IS ANY OF :
      S10
      S11
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (MEMBER) OF THE SAME TYPE

FIND R16 WITHIN SET-NAME CURRENT |USING FIELD-LIST| |RETAINING . . . |
  WHERE SET-NAME IS ANY OF :
      S10
      S11
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL
* INITIALIZE ELEMENTARY FIELDS OF THE LIST , IF USING IS SPECIFIED

FIND R16 WITHIN S 10 |USING FIELD-LIST| |RETAINING . . . |
* APPLY SET SELECTION FOR S10
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15

```

Figure 3-7. Listage DML (4/14)

Manuel de référence IDS/II

```

DDLPROC      30.00  X1159.2  DOCLIST  FAYOT      IDS2      IDS2__R5      08:06:49
                AUG 22, 1980      PAGE      21
                * SCHEMA  TG24-SH *                * PRINT DML REPORT OF RECORD R16*

                OF SET : S1*
INITIALIZE AREA-ID DB-PARAMETER  ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* INITIALIZE ELEMENTARY FIELDS OF THE LIST IF "USING" IS SPECIFIED

FIND R16 WITHIN S11 |USING FIELD-LIST | |RETAINING . . . |
* APPLY SET SELECTION FOR S11
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                OF SET : S11
INITIALIZE AREA-ID DB-PARAMETER  ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* INITIALIZE ELEMENTARY FIELDS OF THE LIST IF "USING" IS SPECIFIED

FIND DUPLICATE WITHIN SET-NAME USING FIELD-LIST |RETAINING . . . |
  WHERE SET-NAME IS ANY OF :
    S10
    S11
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (MEMBER) OF THE SAME TYPE
* INITIALIZE ELEMENTARY FIELDS OF THE LIST

*****
*   GET   *
*****

GET |R16|
GET FIELD-LIST
* ESTABLISH OBJECT RECORD AS :CURRENT-OF-RUN-UNIT

```

Figure 3-7. Listage DML (5/14)

Traitement du schéma

```

DDLPROC      20.00 X1864.1  DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      22
* SCHEMA    TG24-SH *                      * PRINT DML REPORT OF RECORD R16 *

*****
*   DML UPDATE STATEMENTS (FORMAT AND PREREQUISITE ACTION OR CONDITION)   *
*****

*****
*   STORE *
*****

STORE R16 |RETAINING . . .|
* INITIALIZE ALL FIELDS OF RECORD
* INITIALIZE AREA-ID DB-PARAMETER ARID16
  WITH ANY OF :
    TG24-A02
    TG24-A05
    TG24-A03
    TG24-A11
    TG24-A14
* APPLY SET SELECTION FOR S10
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                                OF SET : S10
  INITIALIZE AREA-ID DB-PARAMETER ARID15
  WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
  INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* APPLY SET SELECTION FOR S11
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                                OF SET : S11
  INITIALIZE AREA-ID DB-PARAMETER ARID15
  WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
  INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM

*****
*   MODIFY (CONTENTS) *
*****

MODIFY |R16| |RETAINING . . .|
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT (A GET IS NOT REQUIRED)
* INITIALIZE ALL FIELDS OF THE RECORD WITH THEIR NEW VALUES

MODIFY FIELD-LIST |RETAINING . . .|
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT (A GET IS NOT REQUIRED)
* INITIALIZE ELEMENTARY FIELDS OF THE LIST WITH THEIR NEW VALUES

```

Figure 3-7. Listage DML (6/14)

Manuel de référence IDS/II

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980      PAGE      25
* SCHEMA    TG24-SH *                      * PRINT DML REPORT OF RECORD R16 *

*****
*   MODIFY (MEMBERSHIP)   *
*****

MODIFY |R16| ONLY ALL MEMBERSHIP |RETAINING . . . |
      WHERE "ALL" REPRESENTS THE FOLLOWING SETS :
          S10
          S11
* APPLY SET SELECTION FOR S10
- LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                      OF SET : S10
INITIALIZE AREA-ID DB-PARAMETER ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* APPLY SET SELECTION FOR S11
- LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                      OF SET : S11
INITIALIZE AREA-ID DB-PARAMETER ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT

MODIFY |R16| ONLY SET-NAME-LIST MEMBERSHIP |RETAINING . . . |
      WHERE SET-NAME-LIST IS A SUBSET OF THE LIST OF THE PRECEDING FORMAT
* APPLY SET SELECTION FOR EACH SET OF SET-NAME-LIST (SEE PRECEDING FORMAT)
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT

*****
*   MODIFY (CONTENTS AND MEMBERSHIP)   *
*****

MODIFY |R16| INCLUDING ALL MEMBERSHIP |RETAINING . . . |
MODIFY |R16| INCLUDING SET-NAME LIST MEMBERSHIP |RETAINING . . . |
MODIFY FIELD-LIST INCLUDING ALL MEMBERSHIP |RETAINING . . . |
MODIFY FIELD-LIST INCLUDING SET-NAME-LIST MEMBERSHIP |RETAINING . . . |
* PERFORM THE ACTIONS REQUIRED BY THE MODIFY(CONTENTS) STATEMENT
  AND THE MODIFY (MEMBERSHIP) STATEMENT WHICH ARE COMBINED IN THE
  SELECTED MODIFY (CONTENTS AND MEMBERSHIP) STATEMENT

*****
*   ERASE   *

```

Figure 3-7. Listage DML (7/14)

Traitement du schéma

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980      PAGE      26
              * SCHEMA  TG24-SH *                * PRINT DML REPORT OF RECORD R16 *

*****

ERASE |R16|
* OBJECT RECORD MUST BE OWNER OF EMPTY SETS
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT
ERASE |R16| ALL MEMBERS
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT
  - WARNING - THIS STATEMENT MAY ERASE THE FOLLOWING RECORDS :
    R16
    R17
    
```

Figure 3-7. Listage DML (8/14)

```

bibliothéque BIBSL.UIDS          bibliothéque BIBBIN.UIDS
DDL-PERSONNEL                     PERSONNEL
DDL origine                       TRANSLATE
                                SCHEMA DDL      Schéma résultant
DMCL-PERSONNEL                   DDL  DMCL      PRINT
                                TRANSLATE DMCL    SCHEMA
DMCL origine

bibliothéque BIBSL.UIDS          bibliothéque BIBBIN.UIDS
DDL-PERSONNEL                     PERSONNEL
DDL origine                       Schéma résultant
  (production et test)            (production)
DMCL-PERSONNEL                   DDL  DMCL
DMCL origine                       MOVE SCHEMA

bibliothéque TEST.BIBSL.UIDS     bibliothéque TEST.BIBIN.UIDS
                                Schéma résultant
                                (test)
DMCL-PERSONNEL                   DDL  DMCL
DMCL origine                       TRANSLATE DMCL
  (test)                          (REPLACE)
    
```

Figure 3-7. Listage DML (9/14)

Manuel de référence IDS/II

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980 PAGE      22
              * SCHEMA  TG24-SH *                * PRINT DML REPORT OF RECORD R16 *

              TG24-A14
              * CURRENT-OF-REALM MUST REFERENCE A RECORD OR BE VIRTUAL

FIND CURRENT |R16| WITHIN REALM-NAME |RETAINING . . . |
              WHERE REALM-NAME IS ANY OF :
              TG24-A02
              TG24-A05
              TG24-A08
              TD24-A11
              TG24-A14
              * CURRENT-OF-REALM MUST REFERENCE A RECORD OF THE SAME TYPE

*****
* FIND (WITHIN SET OF WHICH RECORD IS OWNER) *
*****

FIND OWNER WITHIN S13 |RETAINING . . . |
              * CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL

FIND CURRENT |R16| WITHIN S13 |RETAINING . . . |
              * CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD(OWNER)

*****
* FIND WITHIN SET OF WHICH RECORD IS MEMBER) *
*****

FIND FIRST   |R16| WITHIN SET-NAME |RETAINING . . . |
FIND LAST    |R16| WITHIN SET-NAME |RETAINING . . . |
FIND INTEGER |R16| WITHIN SET-NAME |RETAINING . . . |
FIND IDENTIFIER |R16| WITHIN SET-NAME |RETAINING . . . |
FIND NEXT    |R16| WITHIN SET-NAME |RETAINING . . . |
FIND PRIOR   |R16| WITHIN SET-NAME |RETAINING . . . |
              WHERE SET-NAME IS ANY OF :
              S10
              S11
              * CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL
              * INITIALIZE IDENTIFIER IF FORMAT 4 IS USED

FIND CURRENT |R16| WITHIN SET-NAME |RETAINING . . . |
              WHERE SET-NAME IS ANY OF :
              S10
              S11
              * CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (MEMBER) OF THE SAME TYPE

FIND R16 WITHIN SET-NAME CURRENT |USING FIELD-LIST| |RETAINING . . . |
              WHERE SET-NAME IS ANY OF :
              S10
              S11
              * CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (OWNER OR MEMBER) OR BE VIRTUAL
              * INITIALIZE ELEMENTARY FIELDS OF THE LIST , IF USING IS SPECIFIED

FIND R16 WITHIN S 10 |USING FIELD-LIST| |RETAINING . . . |
              * APPLY SET SELECTION FOR S10
              - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
    
```

Figure 3-7. Listage DML (10/14)

Traitement du schéma

```
DDLPROC      30.00  X1159.2  DOCLIST  FAYOT      IDS2      IDS2__R5      08:06:49
              AUG 22, 1980    PAGE      21
* SCHEMA  TG24-SH *                * PRINT DML REPORT OF RECORD R16 *

              OF SET : S1*
INITIALIZE AREA-ID DB-PARAMETER  ARID15
WITH ANY OF :
  TG24-A01
  TG24-A04
  TG24-A07
  TG24-A10
  TG24-A13
INITIALIZE CALC-KEY ITEMS :
  R15-CALC-OCC
  R15-CALC-NUM
* INITIALIZE ELEMENTARY FIELDS OF THE LIST IF "USING" IS SPECIFIED

FIND R16 WITHIN S11 |USING FIELD-LIST | |RETAINING . . . |
* APPLY SET SELECTION FOR S11
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
              OF SET : S11
INITIALIZE AREA-ID DB-PARAMETER  ARID15
WITH ANY OF :
  TG24-A01
  TG24-A04
  TG24-A07
  TG24-A10
  TG24-A13
INITIALIZE CALC-KEY ITEMS :
  R15-CALC-OCC
  R15-CALC-NUM
* INITIALIZE ELEMENTARY FIELDS OF THE LIST IF "USING" IS SPECIFIED

FIND DUPLICATE WITHIN SET-NAME USING FIELD-LIST |RETAINING . . . |
WHERE SET-NAME IS ANY OF :
  S10
  S11
* CURRENT-OF-SET-TYPE MUST REFERENCE A RECORD (MEMBER) OF THE SAME TYPE
* INITIALIZE ELEMENTARY FIELDS OF THE LIST

*****
*   GET   *
*****

GET |R16|
GET FIELD-LIST
* ESTABLISH OBJECT RECORD AS :CURRENT-OF-RUN-UNIT
```

Figure 3-7. Listage DML (11/14)

Manuel de référence IDS/II

```

DDLPROC      20.00 X1864.1  DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      22
* SCHEMA TG24-SH *
* PRINT DML REPORT OF RECORD R16 *

*****
* DML UPDATE STATEMENTS (FORMAT AND PREREQUISITE ACTION OR CONDITION) *
*****

*****
* STORE *
*****

STORE R16 |RETAINING . . .|
* INITIALIZE ALL FIELDS OF RECORD
* INITIALIZE AREA-ID DB-PARAMETER ARID16
  WITH ANY OF :
    TG24-A02
    TG24-A05
    TG24-A03
    TG24-A11
    TG24-A14
* APPLY SET SELECTION FOR S10
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                                     OF SET : S10
INITIALIZE AREA-ID DB-PARAMETER ARID15
  WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
  INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* APPLY SET SELECTION FOR S11
  - LEVEL 1 - IN ORDER TO SELECT OWNER : R15
                                     OF SET : S11
INITIALIZE AREA-ID DB-PARAMETER ARID15
  WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
  INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM

*****
* MODIFY (CONTENTS) *
*****

MODIFY |R16| |RETAINING . . .|
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT (A SET IS NOT REQUIRED)
* INITIALIZE ALL FIELDS OF THE RECORD WITH THEIR NEW VALUES

MODIFY FIELD-LIST |RETAINING . . .|
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT (A SET IS NOT REQUIRED)
* INITIALIZE ELEMENTARY FIELDS OF THE LIST WITH THEIR NEW VALUES

```

Figure 3-7. Listage DML (12/14)

Traitement du schéma

```

DDLPROC      20.00 X1864.1 DOCLIST  FAYOT      IDS2      IDS2__R5
              08:02:1AUG 20, 1980  PAGE       25
* SCHEMA TG24-SH *                      * PRINT DML REPORT OF RECORD R16 *

*****
*   MODIFY (MEMBERSHIP)   *
*****

MODIFY |R16| ONLY ALL MEMBERSHIP |RETAINING . . . |
WHERE "ALL" REPRESENTS THE FOLLOWING SETS :
    S10
    S11
* APPLY SET SELECTION FOR S10
- LEVEL 1 - IN ORDER TO SELECT OWNER : R15
              OF SET : S10
INITIALIZE AREA-ID DB-PARAMETER ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* APPLY SET SELECTION FOR S11
- LEVEL 1 - IN ORDER TO SELECT OWNER : R15
              OF SET : S11
INITIALIZE AREA-ID DB-PARAMETER ARID15
WITH ANY OF :
    TG24-A01
    TG24-A04
    TG24-A07
    TG24-A10
    TG24-A13
INITIALIZE CALC-KEY ITEMS :
    R15-CALC-OCC
    R15-CALC-NUM
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT

MODIFY |R16| ONLY SET-NAME-LIST MEMBERSHIP |RETAINING . . . |
WHERE SET-NAME-LIST IS A SUBSET OF THE LIST OF THE PRECEDING FORMAT
* APPLY SET SELECTION FOR EACH SET OF SET-NAME-LIST (SEE PRECEDING FORMAT)
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT

*****
*   MODIFY (CONTENTS AND MEMBERSHIP)   *
*****

MODIFY |R16| INCLUDING ALL MEMBERSHIP |RETAINING . . . |
MODIFY |R16| INCLUDING SET-NAME LIST MEMBERSHIP |RETAINING . . . |
MODIFY FIELD-LIST INCLUDING ALL MEMBERSHIP |RETAINING . . . |
MODIFY FIELD-LIST INCLUDING SET-NAME-LIST MEMBERSHIP |RETAINING . . . |
* PERFORM THE ACTIONS REQUIRED BY THE MODIFY(CONTENTS) STATEMENT
  AND THE MODIFY (MEMBERSHIP) STATEMENT WHICH ARE COMBINED IN THE
  SELECTED MODIFY (CONTENTS AND MEMBERSHIP) STATEMENT

*****
*   ERASE   *

```

Figure 3-7. Listage DML (13/14)

```
DDLPROC      20.00 X1864.1 DOCLIST  FAYOT    IDS2      IDS2__R5
              08:02:1AUG 20, 1980    PAGE      26
              * SCHEMA TG24-SH *          * PRINT DML REPORT OF RECORD R16 *
```

```
ERASE |R16|
* OBJECT RECORD MUST BE OWNER OF EMPTY SETS
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT
ERASE |R16| ALL MEMBERS
* ESTABLISH OBJECT RECORD AS CURRENT-OF-RUN-UNIT
- WARNING - THIS STATEMENT MAY ERASE THE FOLLOWING RECORDS :
R16
R17
```

Figure 3-7. Listage DML (14/14)

3.9 COMMANDE DELETE SCHEMA

Fonction

Permet de supprimer un schéma résultant.

Format général

```
DELETE SCHEMA nom-schéma [ {nfe-de-bibliothèque} ]
[ {IN} {DDL1B1} ] .
[ {OF} {DDL1B2} ]
[ {DDL1B3} ]
```

Règles

1. La bibliothèque contenant le schéma résultant à supprimer est, soit celle qui est déterminée par l'itinéraire de recherche du traitement en mode mise à jour, soit par la qualification.
2. L'état du schéma résultant à supprimer peut être "DDL" ou "DDL-DMCL".
3. La commande DELETE de l'utilitaire LIBMAINT BIN peut servir également à supprimer un schéma résultant.

3.10 COMMANDE MOVE SCHEMA

Fonction

Permet de transférer un schéma résultant d'une bibliothèque à une autre.

Format général

```

MOVE SCHEMA nom-schéma [ {nfe-de-bibliothèque} ]
                        [ {IN} {DDL1B1} ]
                        [ {OF} {DDL1B2} ]
                        [ {DDL1B3} ]

                        [ {nfe-de-bibliothèque} ]
                        [ TO {DDL1B1} ] [REPLACE] .
                        [ {DDL1B2} ]
                        [ {DDL1B3} ]

```

Règles

1. La bibliothèque dans laquelle se trouve le schéma résultant à transférer est, soit déterminée par l'itinéraire de recherche implicite pour le traitement en mode extraction (RETRIEVAL), soit indiquée par la qualification.
2. L'état du schéma à transférer peut être "DDL" ou "DDL-DMCL".
3. La bibliothèque qui contient la copie du schéma est, soit déterminée par l'itinéraire de recherche du mode création, soit indiquée par la qualification.
4. Si le transfert a pour but de recouvrir une unité homonyme de la bibliothèque de sortie sélectionnée, le mot-clé REPLACE doit obligatoirement être spécifié sinon l'opération se traduit par une erreur.
5. Le transfert d'un schéma résultant peut se faire également par utilisation de la commande MOVE de l'utilitaire LIBMAINT BIN.

3.11 COMMANDE STATUS

Fonction

Permet de gérer les commandes DDLPROC.

Format général

<u>STATUS</u> <u>RESET</u> .

Règles

1. Après chaque commande autre que la commande STATUS, DDLPROC met un indicateur d'état dans l'une des deux positions suivantes :
 - Position NORMAL qui indique une exécution normale de la commande (pas d'erreur bloquante).
 - Position ABNORMAL qui indique une exécution erronée de la commande (erreur bloquante).

2. Lorsque l'indicateur d'état est positionné à NORMAL, DDLPROC recherche et exécute la commande suivante.
3. Lorsque l'indicateur d'état est positionné sur ABNORMAL, DDLPROC qui attend une commande STATUS RESET, recherche la commande suivante.

Si la commande suivante n'est pas STATUS RESET, DDLPROC provoque un arrêt prématuré et émet un code de gravité SEV3.

Si la commande est STATUS RESET, DDLPROC remet l'indicateur d'état à la position normale et poursuit le traitement comme indiqué dans la règle 2.

4. Une commande STATUS RESET est sans effet si l'indicateur d'état est déjà positionné à NORMAL.

3.12 EXEMPLES DE TRAITEMENT DU DDLPROC

1. Le DDL du schéma INVENTAIRE est introduit à partir d'une rubrique de description d'entrée (DDL-INVENTAIRE) et doit faire l'objet d'une traduction et d'un listage mais non d'un chargement. L'opération consiste à effectuer un premier passage pour détecter les erreurs de syntaxe du DDL.

```

$JOB TRADD, USER=PC, PROJECT=MCE;
DDLPROC COMMAND='
TRANSLATE SCHEMA DDL
SOURCE IS *DDL-INVENTAIRE
';
$INPUT DDL-INVENTAIRE;
SCHEMA NAME IS INVENTAIRE.
...
END-SCHEMA
$ENDINPUT;
$ENDJOB;

```

2. Le DDL du schéma PERSONNEL est introduit à partir d'une bibliothèque BIBSL.UIDS (unité de bibliothèque de DDL-PERSONNEL). Ce DDL doit être traduit et chargé dans une bibliothèque de schéma résultant appelée BIBBIN.UIDS (unité de bibliothèque de PERSONNEL). Le DMCL du schéma PERSONNEL est également introduit à partir d'une bibliothèque origine BIBSL.UIDS (unité de bibliothèque de DMCL-PERSONNEL). Ce DMCL doit être traduit et chargé dans le schéma résultant partiel créé par la commande précédente. Des informations sur le rangement de la base de données sont aussi demandées.

La Figure 3-8 illustre la séquence d'opérations.

```

$JOB TDDLDMCL, USER=PC, PROJECT=MCE;
BIB SL INLIB1=BIBSL.UIDS;
PROC COMFILE=*COMTRAD
    DDLIB1=MIDS.BIBBIN.UIDS;
$INPUT COMTRAD;
TRANSLATE SCHEMA DDL
    SOURCE IS DDL-PERSONNEL
    OBJECT
    CHECK FOR COBOL.
TRANSLATE DMCL
    SOURCE IS DMCL-PERSONNEL
    OBJECT
PRINT SCHEMA PERSONNEL
    STORAGE DESCRIPTION.
$ENDINPUT;
$ENDJOB;

```

En ce qui concerne les objets de la commande TRANSLATE SCHEMA DDL l'unité de bibliothèque origine DDL PERSONNEL est extraite de la bibliothèque BIBSL.UIDS (première et unique bibliothèque sur l'itinéraire de recherche des éléments SL) et le schéma résultant PERSONNEL, à l'état "DDL", est enregistré dans la bibliothèque BIBBIN.UIDS (première et seule bibliothèque sur l'itinéraire de recherche des éléments DD).

En ce qui concerne les objets sur lesquels porte la commande TRANSLATE DMCL, l'unité de bibliothèque DMCL origine PERSONNEL est extraite de la bibliothèque BIBSL.UIDS (première et unique bibliothèque sur l'itinéraire de recherche SL). Le schéma résultant PERSONNEL est extrait de la bibliothèque BIBBIN.UIDS

Traitement du schéma

(première et seule bibliothèque sur l'itinéraire de recherche DD). Le schéma est ainsi mis à jour pour contenir le DMCL et il devient schéma résultant, à l'état "DDL-DMCL", disponible pour le chargement au moment de l'exécution.

En ce qui concerne l'objet de la commande PRINT-SCHEMA, le schéma résultant PERSONNEL, il est à l'état "DDL-DMCL" et extrait de la bibliothèque BIBBIN. UIDS (première et seule bibliothèque sur l'itinéraire de recherche DD).

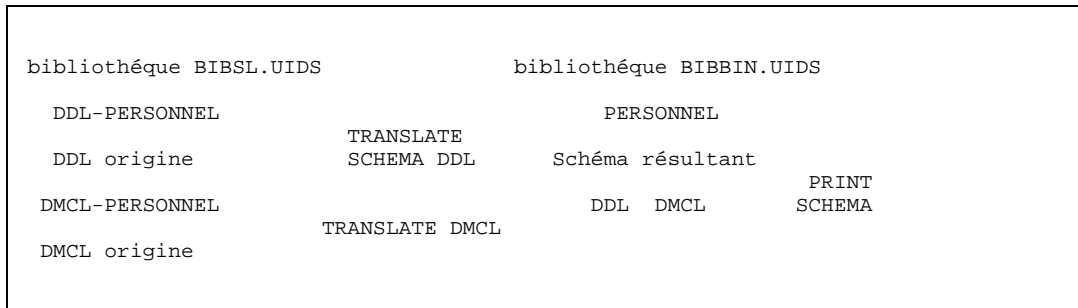


Figure 3-8. Traitement du schéma PERSONNEL

3. Poursuivons sur le même exemple. L'utilisateur décide de modifier les paramètres DMCL après avoir effectué plusieurs passages à l'aide de la commande SIMULOAD de l'utilitaire DBUTILITY. Il met à jour l'unité de bibliothèque DMCL-PERSONNEL à l'aide de l'utilitaire LIBMAINT SL puis il appelle DDLPROC pour traduire le nouveau DMCL et mettre à jour le schéma résultant.

```
$JOB TNOUVDMCL, USER=PC, PROJECT=MCS;  
BIB SL INLIB1=BIBSL.UIDS;  
DDLPROC COMFILE=*COMTRAD  
        DDLIB1=BIBBIN.UIDS;  
$INPUT COMTRAD;  
TRANSLATE DMCL  
        SOURCE IS DMCL-PERSONNEL  
        OBJECT REPLACE.  
$ENDINPUT;  
$ENDJOB;
```

Le mot-clé REPLACE est nécessaire pour recouvrir l'ancienne partie DMCL.

4. L'utilisateur décide maintenant de vérifier ses programmes DML à l'aide d'une base de données et test dont la taille est inférieure à celle de la base de données de production.

Le DDL du schéma est exactement le même pour la version test et la version production. Les variations du DMCL portent sur les tailles des aires.

L'utilisateur prépare le DMCL origine du test dans la bibliothèque de test BIBSL.UIDS (unité de bibliothèque DMCL-PERSONNEL). Il transfère ensuite le schéma résultant de production PERSONNEL de la bibliothèque BIBBIN.UIDS dans la bibliothèque TEST.BIBBIN.UIDS et traduit le DMCL de test pour remplacer le DMCL de production se trouvant dans la copie, créant ainsi le schéma résultant de test. La Figure 3-9 plus loin illustre la séquence des opérations.

```
$JOB TESTDMCL, USER=PC, PROJECT=MCE;  
BIB SL INLIB1=BIBSL.UIDS  
        INLIB2=TEST.BIBSL.UIDS;  
DDLPROC DDLIB1=BIBBIN.UIDS  
        DDLIB2=TEST.BIBBIN.UIDS
```

Manuel de référence IDS/II

```
COMMAND= '  
MOVE SCHEMA PERSONNEL TO TEST.BIBBIN.UIDS  
TRANSLATE DMCL  
SOURCE IS DMCL-PERSONNEL OF SLLIB2  
SCHEMA OF DDLIB2  
OBJECT REPLACE.  
' ;  
$ENDJOB;
```

A l'aide de la commande MOVE, le schéma résultant PERSONNEL est extrait de BIBBIN.UIDS (première bibliothèque sur l'itinéraire de recherche DD) et transféré dans la bibliothèque TEST.BIBBIN.UIDS comme l'indique le nfe accompagnant la locution TO. A l'aide de la commande TRANSLATE, l'unité de bibliothèque DMCL-PERSONNEL est extraite de TEST.BIBSL.UIDS comme spécifié par le qualificateur SLLIB2. Le schéma résultant est extrait de la bibliothèque TEST.BIBBIN.UIDS comme indiqué sur le qualificateur DDLIB2.

bibliothèque BIBSL.UIDS		bibliothèque BIBBIN.UIDS
DDL-PERSONNEL		PERSONNEL
DDL origine (production et test)		Schéma résultant (production)
DMCL-PERSONNEL		DDL DMCL
DMCL origine		MOVE SCHEMA
bibliothèque TEST.BIBSL.UIDS		bibliothèque TEST.BIBIN.UIDS
		Schéma résultant (test)
DMCL-PERSONNEL		DDL DMCL
DMCL origine (test)	TRANSLATE DMCL (REPLACE)	

Figure 3-9. Traitement du schéma de test PERSONNEL

3.13 MISE AU POINT DE DDLPROC

Si, pendant le traitement d'une commande, DDLPROC s'arrête prématurément à cause d'une condition d'exception ou s'il émet un message "INTERNAL BUG" (erreur interne), l'Inspecteur de maintenance (ou l'utilisateur) doit :

1. Repasser DDLPROC avec le paramètre DUMP=DATA du JCL étendu et introduire une commande composée du mot-clé DEBUG suivi de la commande qui a provoqué l'erreur, par exemple :

```
$INPUT DDLCOM;  
DEBUG.  
TRANSLATE SCHEMA DDL ...  
$ENDINPUT;
```

2. Envoyez le résultat imprimé de la sortie accompagné des états imprimés des DDL et DMCL origines au système de mise au point STAR.

4. Réserveation d'aire mémoire

4.1 UTILITAIRE PREALLOC

La fonction de PREALLOC est de projeter une aire mémoire, équivalente d'un fichier UFAS, sur un ou plusieurs espaces fichier d'un ou plusieurs volumes disque (voir Figure 4-1).

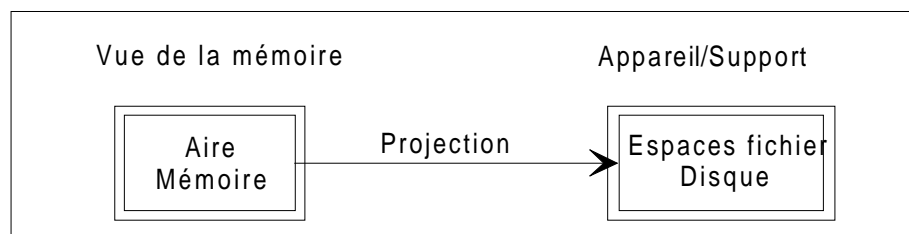


Figure 4-1. Fonction PREALLOC

Chaque aire mémoire de la base de données doit être réservée au moyen de PREALLOC.

Les caractéristiques de l'aire mémoire que constituent le nombre de pages, le nombre de lignes par page, la taille de page et l'intervalle CALC sont extraites du schéma résultant, à l'état "DDL-DMCL", par PREALLOC.

Les principaux paramètres à spécifier directement par PREALLOC sont :

- Le nom-de-fichier-externe (nfe), qui identifie l'aire mémoire.
- Le type de disque.
- Les noms des volumes sur lesquels l'espace est réservé.
- Facultativement, la quantité de place à réserver sur chaque volume et la position qui marquera le début de l'affectation de place.

4.2 RESERVATION D'ESPACE

1. A partir des informations du DMCL (nombre de pages NUMP, taille de page) PREALLOC calcule la taille totale du fichier en termes de pistes pour la classe d'appareil spécifié.
 - Ce calcul s'effectue automatiquement. Toutefois, le nombre de pistes réservées est une information qui présente un intérêt administratif pour l'utilisateur.
 - Le calcul s'effectue de la manière suivante :

Trouver le nombre (PPT) de pages par piste en utilisant le Tableau 2-4 du chapitre 2.

Calculer le nombre total (NUMT) de pistes à l'aide de la formule :

$$\text{NUMT} = (\text{NUMP} - 1) / \text{PPT} + 2$$

2. L'unité de réservation de place est la piste de disque. La place affectée est constituée d'une série (un ou plusieurs) espaces fichier disque. Un espace fichier disque est constitué d'une piste, ou d'un groupe de pistes contiguës. Un fichier, sur un volume quelconque, peut s'étendre sur 16 espaces. Cependant l'utilisateur peut, pour des raisons d'efficacité, ramener le nombre d'espaces fichier par volume en dessous du maximum de 16 autorisé.
3. Pour la réservation d'espace automatique, l'utilisateur ne spécifie que la liste des volumes sur lesquels les aires sont rangées.

PREALLOC affecte au premier volume de la liste autant d'espaces fichier nécessaires pour contenir le nombre maximum (NUMT) de pistes. Si ce premier volume peut contenir le fichier, l'opération s'arrête et les autres volumes ne sont pas pris en compte. Si la place disponible sur ce volume est plus petite que NUMT ou si le nombre maximum d'espaces fichier autorisé est atteint prématurément, PREALLOC tente de disposer le reste du fichier sur le volume suivant de la liste selon le même type de réservation et ainsi de suite jusqu'à ce que le fichier complet soit affecté.

Le processus de réservation des espaces fichier de chaque volume est décrit ci-dessous.

Une liste de tous les espaces fichier libres disponibles est examinée. Le plus petit espace qui convient, c'est-à-dire celui qui est égal ou immédiatement supérieur à la quantité de place demandée est choisi. Ainsi, si les espaces fichier libres représentaient 20, 23, 25, 60 cylindres et que la place demandée correspond à 24 cylindres, la réservation de l'espace fichier 25 cylindres s'effectuerait, laissant libres les autres espaces de 1, 20, 23 et 60 cylindres. Si la place demandée est plus grande que le plus grand des espaces fichier disponibles, il y a d'abord réservation du plus grand des espaces, puis, comme il y a un reliquat de place à trouver, une recherche va s'effectuer sur le plus petit des grands espaces fichier restant, ou le choix du plus grand espace fichier va s'opérer accompagné d'une nouvelle recherche pour la place restant à trouver. En reprenant la liste d'espaces disponibles indiqués plus haut et en supposant que le besoin de place soit de 86 cylindres, les espaces 60 et 25 seront réservés et un cylindre sera pris sur l'espace 20 cylindres.

Réservation d'aire mémoire

4. Il existe deux autres procédés de réservation non automatique d'espace. Dans la première, l'utilisateur spécifie pour chaque volume la quantité d'espace à réserver. Dans ce cas, la réservation ne se fait pas si la quantité de place demandée n'est pas disponible sur le volume ou si le nombre d'espace fichier requis dépasse celui qui est autorisé.

Dans le deuxième, l'utilisateur peut spécifier l'adresse du cylindre à partir duquel la réservation doit commencer. En ce cas, il doit exister suffisamment de place d'un seul tenant à partir de l'adresse indiquée pour réserver, dans un seul espace fichier disque, la quantité de place totale demandée.

5. PREALLOC effectue lui-même la mise en forme de l'espace affecté. Pour chaque page, il écrit l'en-tête de page et l'article OWL de chaîne CALC. Il remplit le reste de la page de zéros binaires.
6. PREALLOC crée également les tables UFAS standard et les labels spécifiques à IDS. Ces derniers contiennent: le nom et la date de référence DMCL du schéma utilisé pour la réservation d'espace, le nom de l'aire mémoire et ses caractéristiques DMCL.
Toutes ces informations devront être cohérentes avec celles qui sont fournies à l'exécution dans le schéma prêt.

4.3 JCL ETENDU DE PREALLOC

Le JCL étendu décrit ci-dessous permet de lancer PREALLOC.

```

PREALLOC nom-de-fichier-externe

{
  FILESTAT = { CAT          }
             { UNCAT       }
             { TEMPRY      }
  TEMPRY
}

[ CATALOG = entier-1 ] [CATNOW]

[ EXPDATE = { jjj          } ]
[           { aa/jjj      } ]
[           { aa/mm/jj    } ]

[ UNIT = { CYL          } ]
[         { TRACK       } ]

[ MAXEXT = { 5           } ]
[           { entier-2   } ]

{
  RESIDENT
  {
    DEVCLASS = classe-d'appareil
    {
      { GLOBAL = (MEDIA = (nom-volume [nom-volume] ... )) }
      {
        { SPLIT = ((nom-volume SIZE = entier-3
                   [CYL = entier-4 [TRACK = entier-5]]) }
        {
          { [(nom-volume SIZE = entier-6
              [CYL = entier-7 [TRACK = entier-8]])] ... ) }
        }
      }
    }
  }

  UFAS = (IDS = (AREA = nom-aire
                SCHEMA = nom-schéma
                DDLIB1 = (description-bibliothèque))) ;

```

4.4 DESCRIPTION DES PARAMETRES PREALLOC

Nom-de-fichier-externe (nfe)	Ce nom doit être unique parmi les noms de fichiers d'ensemble déjà rangés sur le ou les volumes à utiliser.
FILESTAT	Ce mot-clé indique si l'aire est cataloguée (CAT), non cataloguée (UNCAT) ou temporaire (TEMPRY).
TEMPRY	Ce mot-clé autodéfini est l'équivalent de FILESTAT=TEMPRY.
CATALOG	Entier-1 représente un nombre entier à un chiffre qui doit remplir la condition: 1 <= entier-1 <= 5 Il représente, dans l'instruction ATTACH, le rang du catalogue dans lequel le fichier est ou doit être répertorié. Si CATALOG n'est pas spécifié alors que FILESTAT=CAT l'est, l'itinéraire de recherche implicite de CATALOG s'applique.
CATNOW	Ce mot-clé indique qu'une description de fichier doit être introduite dans le catalogue spécifié. CATNOW n'est pas pris en compte si une description de fichier existe déjà.
EXPDATE	Ce mot-clé donne la date de péremption du fichier. Si aucune date n'est fournie, la date de péremption est implicitement la date courante.
UNIT	Ce mot-clé s'utilise conjointement au paramètre SPLIT. Il indique l'unité de mesure (cylindre ou piste) du paramètre SIZE (taille).
MAXEXT	Entier-2 comporte un ou deux chiffres décimaux qui doivent remplir la condition: 1 <= entier-2 <= 16 Il spécifie le nombre maximum d'espaces fichier choisi par volume.
RESIDENT	Si ce mot-clé est spécifié, la réservation de place s'effectuera sur les volumes résidants du système.
DEVCLASS	Lorsque ce mot-clé est spécifié, l'espace réservé est non-résidant. Ce mot-clé est suivi de l'identificateur de la classe d'appareil disque sélectionné.
GLOBAL	Ce mot-clé indique que la réservation d'espaces fichier doit se faire automatiquement.

MEDIA	<p>Ce paramètre est suivi d'une liste de un ou plusieurs volumes de support d'enregistrement. L'ordre de la liste est important. L'utilitaire affecte le nombre maximum d'espaces fichier (voir paramètre MAXEXT) sur le premier volume, puis sur le second et ainsi de suite.</p> <p>Même si la réservation est de type multivolume, il peut arriver que les besoins de place pour le fichier ne nécessitent pas la totalité des volumes. En ce cas les volumes restant ne sont pas utilisés et ne sont plus mentionnés dans les références ultérieures au fichier.</p> <p>Les noms de volumes sont classés dans l'ordre de leur utilisation. Les pages sont classées par numéros de page, les plus petits numéros sur le premier volume et les plus grands sur le dernier.</p>
SPLIT	<p>Ce mot-clé spécifie que l'affectation de place est effectuée volume par volume. Il est suivi d'un groupe de paramètres qui établit la liste des volumes et, pour chaque volume, la place qui y est nécessaire. L'utilisateur peut, facultativement, spécifier l'adresse début de la réservation pour chaque volume. Il peut également spécifier les adresses début pour certains volumes et ne pas les mentionner pour d'autres.</p> <p>Les noms de volumes doivent être classés dans l'ordre de leur utilisation. Les plus petits numéros de pages sont enregistrés sur le premier volume et les plus grands sur le dernier.</p>
SIZE	<p>Ce mot-clé fournit, pour chaque volume, la quantité de place requise sur le volume. La valeur numérique fournie, maximum 5 chiffres, s'exprime en unités de mesure indiquées dans UNIT.</p>
CIL et TRACK	<p>Ces mots-clés sont facultatifs. Ils servent à indiquer l'adresse début de la réservation de place dans le volume. L'utilisation d'un de ces mots-clés signifie qu'il n'y a pas de sélection automatique d'espaces fichier. La place, exprimée par SIZE pour le volume, est réservée en partant de l'adresse début spécifiée. Toutefois l'utilisateur doit s'assurer qu'il existe un espace d'un seul tenant suffisant, à partir de l'adresse indiquée. Dans le cas contraire la réservation de place ne réussit pas et s'arrête prématurément.</p> <p>La valeur numérique exprimant l'adresse du cylindre ne doit pas dépasser 3 chiffres. Une adresse début de réservation peut indiquer un cylindre partiellement utilisé.</p> <p>L'adresse de piste ne doit pas dépasser deux chiffres.</p> <p>La non-utilisation de CYL pour un volume signifie que la sélection d'espaces fichier se fait automatiquement.</p>
UFAS	<p>Ce mot-clé est suivi d'un groupe de paramètres qui donne les caractéristiques du fichier.</p>

Réservation d'aire mémoire

IDS	Ce mot-clé signifie que la réservation de place concerne une aire IDS.
AREA	Ce mot-clé sert à introduire le nom DMCL de l'aire mémoire (30 caractères maximum).
SCHEMA	Ce mot-clé indique le nom du schéma qui contient la définition de l'aire mémoire.
DDLIB1	Ce mot-clé est suivi d'un groupe de paramètres qui identifie la bibliothèque du schéma résultant.

4.5 EXEMPLES

1. La réservation de place concerne deux aires mémoire. Les rubriques DMCL qui les décrivent sont :

```
AREA NAME IS REVISION
NUMBER-OF-PAGES IS 10000
PAGE-SIZE IS 1024 BYTES
LINES-PER-PAGE IS 20.
```

```
AREA NAME IS ADDITION
NUMBER-OF-PAGES IS 50000
PAGE-SIZE IS 2048 BYTES
LINES-PER-PAGE IS 28.
```

Le nom de fichier externe de la première aire est UIDS.RE. Elle est affectée au volume BD1023 du MSU 0400.

Le nom de fichier externe de la seconde aire est UIDS.LB. Elle est affectée aux volumes BD1024, BD1025 du MSU 0400.

Les deux fichiers sont protégés par une période de validité de 300 jours.

Le schéma résultant est l'unité de bibliothèque TPL de la bibliothèque UIDS.BIBBIN déclarée RESIDENT.

La description de travail qui permettra cette réservation est indiquée ci-dessous :

```
$JOB AFFECBD USER=JEAN PROJECT=CMSMCETP;
  PREALLOC UIDS.RE
    FILESTAT=UNCAT
    EXPDATE=300
    DEVCLASS=MS/M400 GLOBAL=(MEDIA=BD1023)
    UFAS=(IDS(AREA=REVISION
              SCHEMA=TPL
              DDLIB1=UIDS.BIBBIN));
  PREALLOC UIDS.LB
    FILESTAT=UNCAT
    EXPDATE=300
    DEVCLASS=MS/M400 GLOBAL=(MEDIA=(BD1024 BD1025))
    UFAS=(IDS=(AREA=ADDITION
                SCHEMA=TPL
                DDLIB1=UIDS.BIBBIN));
$ENDJOB;
```

Les besoins en place peuvent être calculés en utilisant le Tableau 2-4.

- Pour le fichier UIDS.RE: NUMP = 10 000 PPT = 11
 Nombre de pistes = $(10\ 000 - 1)/11 + 2 = 911$ (48 cylindres)
- Pour le fichier UIDS.LB: NUMP = 50 000 PPT = 6
 Nombre de pistes = $(50\ 000 - 1)/6 + 2 = 8335$ (439 cylindres)

Réservation d'aire mémoire

2. Une aire mémoire est affectée sur trois volumes MSU 0400 VL0015, VL0016, VL0017. La rubrique DMCL est :

```
AREA NAME IS PRODUITS
NUMBER-OF-PAGES IS 100000
PAGE-SIZE IS 2048 BYTES
LINES-PER-PAGE IS 45.
```

Le calcul du nombre total de pistes découle des informations ci-dessus à savoir NUMP = 100 000 et PPT = 6 :

$$(100\ 000 - 1)/6 + 2 = 16\ 668$$

ce qui donne un total de 878 cylindres.

La répartition de la place réservée étant la suivante :

- 300 cylindres réservés sur le volume VL0015 avec comme adresse début le cylindre 65.
- 300 cylindres réservés quelque part sur le volume VL0016.
- Le reste étant réservé quelque part sur le cylindre VL0017.
- Le fichier est déjà répertorié dans le catalogue de site avec nfe PRODUITS.DEPT1.
- Le schéma est l'unité de bibliothèque INVENTAIRE de la bibliothèque cataloguée CONTROL.DEPT1.BIBBIN.

La description du travail à exécuter sera :

```
$JOB AFFECBD USER=JEAN PROJECT=CMS;
PREALLOC PRODUITS.DEPT1
FILESTAT=CAT
DEVCLASS=MS/M400
SPLIT=((VL0015 SIZE=300 CYL=65)
      (VL0016 SIZE=300)
      (VL0017 SIZE=300))
UFAS=(IDS=(AREA=PRODUITS
           SCHEMA=INVENTAIRE
           DDLIB1=CONTROL.DEPT1.BIBBIN));
$ENDJOB;
```

Les valeurs implicites pour UNIT et MAXEXT sont respectivement CYL et 5.

L'utilisation doit s'assurer que les cylindres 65 à 364 sont libres sur le volume VL0015 avant de lancer l'opération de réservation.

Sur le volume VL0016, un minimum de 300 cylindres doit être libre sur un peu moins de 6 espaces fichier.

La taille (SIZE) spécifiée pour le volume VL0017 étant plus grande que nécessaire, PREALLOC n'affecte que la place nécessaire, à savoir 277 cylindres.

5. Langage DML de manipulation des données

5.1 GENERALITES

IDS/II offre une fonction de manipulation des données qui permet d'accéder à une base de données décrite par un schéma. Cette fonction se compose :

1. De zones de communication entre l'utilisateur et le système SGBD
Celles-ci comprennent :
 - Les articles de la zone de manoeuvre utilisateur ou zone UWA, zone dans laquelle les articles sont écrits avant leur rangement ou dans laquelle ils sont transférés après extraction. Le format des articles UWA correspond à celui qui a été défini par le schéma.
 - Les registres spéciaux par l'intermédiaire desquels le SGBD informe l'utilisateur de la fin normale ou anormale d'une fonction.
 - Les indicateurs de points courant dans lesquels le SGBD entretient des pointeurs d'articles de la base de données. Ces indicateurs constituent des paramètres en entrée et en sortie pour la plupart des fonctions de manipulation.
2. Fonctions de manipulation. Elles permettent à l'utilisateur de ranger, extraire et supprimer des articles, de modifier le contenu des articles et leur relation entre eux et enfin de vérifier les différentes conditions de la base de données.

5.1.1 Langage Hôte

La fonction de la base de données ne constitue en aucune manière un ensemble isolé. Elle est considérée comme un service accessible à tout programme écrit en langage hôte.

Le programme utilise les structures de ce langage hôte pour exprimer la séquence des fonctions de manipulation à exécuter, pour travailler sur le contenu des articles UWA et pour interpréter le résultat des fonctions de manipulation.

La manière dont la fonction de la base de données est utilisée par le langage hôte lui-même dépend du type de langage.

5.1.2 Langages de mise en oeuvre

Pour ces langages, la fonction de base de données n'ajoute aucune nouvelle structure au langage hôte. Les fonctions de manipulation de données sont mises en oeuvre par des appels de procédure standard. Le programmeur doit définir les zones de communication et les paramètres de fonction en utilisant pour cela les instructions déclaratives du langage hôte.

La relation entre le schéma et les programmes ne dépend pas du compilateur. Ces langages sont donc particulièrement adaptés aux systèmes d'interprétation que sont QUERY ou les utilitaires de base de données. Ceux-ci doivent s'adapter d'eux-mêmes, au moment de l'exécution, à tous les schémas de la base de données.

5.1.3 Langages de haut niveau (COBOL)

Pour le COBOL, la fonction base de données est assimilée au langage hôte lui-même. Elle introduit de nouvelles structures reconnues par le compilateur. Les fonctions de manipulation prennent la forme de nouvelles instructions de procédure selon les règles de syntaxe du langage hôte. Les zones de communication ne sont pas définies par le programmeur. A partir du schéma, ces zones sont créées automatiquement par le compilateur et rendues disponibles au programme sous la forme d'instructions déclaratives spécifiques au langage hôte.

A la différence de ce qui se passe pour le langage de mise en oeuvre, le programme est consacré à un schéma donné.

5.2 LANGAGE DE MANIPULATION DES DONNEES COBOL

Les nouvelles structures introduites par la fonction base de données dans le langage COBOL concerne la DATA DIVISION et la PROCEDURE DIVISION.

1. Dans la DATA DIVISION, une nouvelle section, la SUB-SCHEMA SECTION, définit le schéma auquel il est fait référence dans le programme. Les zones de communication découlent automatiquement de ce schéma.

Une seule référence à un schéma peut être faite dans un programme compilé. Si une unité d'exécution doit utiliser plus d'un schéma, des compilations séparées sont autorisées.

2. Un nouveau format, USAGE: DB-KEY, est introduit dans les sections WORKING-STORAGE et LINKAGE. Il permet de définir les éléments de la clé base de données.

3. De nouvelles instructions sont introduites dans la PROCEDURE DIVISION :

- Les verbes ACCEPT, CONNECT, DISCONNECT, ERASE, FIND, FINISH, GET, MODIFY, READY et STORE deviennent des instructions impératives.
- L'instruction IF vérifie les conditions de la base de données.
- L'instruction USE de la partie DECLARATIVES permet la gestion centralisée des exceptions de la base de données.

4. Le paramètre DDLIST de l'instruction JCL COBOL fournit au programmeur un listage des articles-utilisateur (DATA BASE RECORDS), contenant tous les articles sélectionnés dans le schéma accompagnés de leurs éléments correspondants.

5. La commande PRINT DML du processeur DDLPROC fournit au programmeur l'information requise pour accéder à la base de données. Cette information, qui porte sur le type d'article, comprend :

- la description de l'article UWA, en termes COBOL,
- les zones de contrôle d'article (CALC-KEY), clés de tri, zones d'interdiction de duplication d'article,
- les ensembles auxquels l'article est rattaché,
- le mode de localisation de l'article et les aires qui le contiennent. Bien noter que le terme DDL "aire" correspond au terme COBOL "domaine",
- la liste des verbes du langage de manipulation autorisés pour le traitement de l'article, accompagnée des conditions préalables à remplir avant de les appeler. Les instructions ACCEPT, FINISH, READY, IF et USE ne sont pas mentionnées dans cette liste.

5.2.1 Représentation

La représentation des formats et les règles qui s'y appliquent sont les suivantes :

- Les éléments d'une clause sont constitués de mots majuscules, mots minuscules, symboles spéciaux et caractères spéciaux.
- L'usage des mots-clés du format, mots majuscules soulignés, est obligatoire lorsque le schéma est utilisé.
- Les mots majuscules non soulignés constituent des mots facultatifs.
- Les mots minuscules sont des termes génériques qui doivent être remplacés par des noms ou des valeurs appropriées.
- L'encadrement d'une partie d'un format général entre des crochets [] indique que cette partie peut être, au choix de l'utilisateur, incluse ou omise. Les accolades gauche et droite, encadrant une partie d'un format général, signifient un choix à opérer parmi les options encadrées par les accolades. L'apparition de barres verticales ||, encadrant une partie du format indique qu'une sélection d'une ou plusieurs des options encadrées, mais une seule à la fois, doit être opérée dans la rubrique ou l'instruction auxquelles ces options se rapportent. Dans tous les cas, les paramètres exprimant les choix sont listés verticalement. Les éléments du format encadrés par des crochets ou des accolades peuvent être suivis d'une éclipse (...) pour indiquer la répétition.

Exemples :

1.

```
FIND [nom-article] DB-KEY IS identificateur-1
```

L'élément nom-article doit être inclus ou omis suivant les besoins du programme. S'il est inclus, cet élément indique qu'une comparaison est effectuée pour s'assurer que l'article trouvé appartient bien au type d'article mentionné.

2.

```
FIND { ANY } nom-article
      { DUPLICATE }
```

Un choix doit être opéré entre ANY et DUPLICATE.

3.

```
FIND sélection-d'article
[
[
[ RETAINING CURRENCY FOR { MULTIPLE } ] ] ]
[ { REALM } ] ] ]
[ { { SETS } } ... ] ] ]
[ { { {nom-ensemble} } } ... ] ] ]
[ { RECORD } ] ] ]
```

Langage DML de manipulation des données

Cet exemple utilise les barres verticales pour indiquer qu'un choix doit être opéré, et au moins un, dans la liste des paramètres fournis. La partie encadrée par des barres verticales aurait pu aussi bien s'écrire :

```
RETAINING REALM RECORD
RETAINING RECORD
RETAINING REALM nom-ensemble-1 nom-ensemble-2 nom-ensemble-3
RETAINING REALM SETS
```

5.2.2 Règles de syntaxe

5.2.2.1 Syntaxe COBOL

Les règles du COBOL-74 concernant la syntaxe, la ponctuation et l'alignement des colonnes s'appliquent aux instructions DML.

5.2.2.2 Mots réservés

La liste des mots réservés du COBOL/DML se trouve dans le COBOL REFERENCE MANUAL (Réf: 65 A2 DE01).

5.2.2.3 Qualification de zone d'article

Dans le DDL de schéma :

- les noms de zone sont uniques dans un type d'article,
- des zones homonymes peuvent figurer dans plusieurs types d'articles différents.

Lorsque avec des verbes COBOL-74 on utilise des zones homonymes, elles doivent être qualifiées par le nom-article. Prenons l'exemple du DDL suivant :

```
RECORD CLIENT-1A ...
  02 NUMERO-CLIENT  TYPE UNPACKED DEC 6.
  02 NOM-CLIENT     TYPE CHAR 30.
  02 ADRESSE-CLIENT TYPE CHAR 40
RECORD CLIENT-2A ...
  02 NUMERO-CLIENT  TYPE UNPACKED DEC 6.
  02 NOM-CLIENT     TYPE CHAR 30.
  02 ADRESSE-CLIENT TYPE CHAR 40 OCCURS 2.
SET REGION ...
  MEMBER CLIENT-1A ...
  MEMBER CLIENT-2A ...
```

Dans l'instruction DML :

```
FIND CLIENT-1A WITHIN REGION USING NUMERO-CLIENT OF CLIENT-1A.
```

NUMERO-CLIENT doit être qualifié même si CLIENT-1A est déjà spécifié.

Dans l'instruction DML :
 GET ADRESSE-CLIENT OF CLIENT-1A NOM-CLIENT OF CLIENT-1A.

Toutes les zones ont été qualifiées de manière à identifier CLIENT-1A.

En COBOL, la qualification précède l'indiciage. En DDL, l'indiciage précède la qualification :

COBOL: ADRESSE-CLIENT OF CLIENT-2A (2)
 DDL : ADRESSE-CLIENT (2) OF CLIENT-2A

5.2.3 Section SUB-SCHEMA dans la DATA DIVISION

La SUB-SCHEMA SECTION, qui constitue la première section de la DATA DIVISION, ne contient qu'une rubrique, la rubrique de sous-schéma ou rubrique DB. Cette rubrique décrit le schéma de la base de données auquel le programme doit accéder. Il ne peut y avoir qu'une rubrique DB par programme.

Format général

DATA DIVISION .

SUB-SCHEMA SECTION .

DB nom-sous-schema .

```
[
[ DB-DESCRIPTIONS IN { WORKING-STORAGE } SECTION . ]
[ { LINKAGE } ] ]

[ {
[ { RECORDS ARE { ALL } } ] ]
[ { [NOT] {nom-article-utilisateur} ... } } ] ]
[ { } ] ]
[ {
[ { REALMS ARE { ALL } } ] ]
[ { [NOT] {nom-domaine} ... } } ] ] ]
```

Règles

1. Le schéma référencé par nom-sous-schéma doit être accessible au programme pendant l'exécution (voir chapitre 7).
2. La clause DB-DESCRIPTIONS spécifie dans quelle section se trouve les descriptions de la base de données et les structures de contrôle du SGBD au moment de la compilation du programme.
3. Un programme est dit principal lorsqu'il n'est appelé par aucun autre programme. Un tel programme peut être une activité en traitement par lots ou une routine TPR de TDS. En ce cas, la clause DB-DESCRIPTIONS doit spécifier WORKING STORAGE. Si cette clause est omise, WORKING STORAGE constitue la valeur par défaut.

Langage DML de manipulation des données

4. Un programme est dit secondaire lorsqu'il est appelé par un autre programme. En ce cas, la clause DB-DESCRIPTIONS doit spécifier LINKAGE SECTION. Un programme secondaire peut appeler un autre programme secondaire.
5. Pour un schéma donné, la phrase DB-DESCRIPTIONS IN WORKING STORAGE ne doit figurer que dans un seul programme parmi l'ensemble des programmes qui constitue une activité en traitement par lots ou une routine TPR de TDS.
6. Tout programme (principal ou secondaire) d'une activité en traitement par lots peut contenir des instructions READY et FINISH. Une routine TPR ne contient généralement aucune instruction READY ou FINISH. Si elle en contient, ces instructions ne sont pas prises en compte au moment de l'exécution.
7. La locution USING de l'instruction CALL qui appelle un programme secondaire doit contenir des identificateurs-de-donnée qui seront utilisés au moment de l'exécution du programme secondaire.

Les mêmes identificateurs placés dans le même ordre doivent figurer dans la locution USING de l'instruction PROCEDURE du programme secondaire lui-même.

C'est l'utilisateur qui doit fournir la liste des identificateurs. Le format de cette liste d'identificateurs-de-donnée est indiqué ci-dessous :

```
USING{id-donnée} ... DB-REGISTERS BD-CXT [DB-PARAMETERS { nom-art-utilisateur} ...
```

dans lequel:

- id-donnée ... sont des arguments de donnée utilisateur qui sont échangés entre les programmes.
 - Les paramètres DB-REGISTERS, DB-CXT et DB-PARAMETERS fournissent les structures de contrôle nécessaires à l'exécution des fonctions DML.
- DB-PARAMETERS ne doit pas être spécifié si le schéma ne contient pas de paramètres-base-de-données.
- Nom-art-utilisateur ... constitue une liste des noms d'article utilisateur référencés dans le programme appelé. Les noms d'article utilisateur doivent être les mêmes que ceux qui ont été enregistrés sans le schéma. Si un type d'article n'a pas de zones, cet article ne doit pas être spécifié dans la liste.
8. La clause RECORDS permet à l'utilisateur de restreindre les références du programme à des types d'articles particuliers. Seule l'information concernant ce type d'article sera conservée par le programme.
 9. Si le mot NOT est placé avant la liste des articles, tous les articles, sauf ceux indiqués par NOT sont disponibles.
 10. Si l'option RECORDS ARE ALL est choisie, la totalité de la base de données décrite par le schéma peut être citée.
 11. A la place de RECORDS ARE, l'utilisateur a la possibilité de spécifier REALMS ARE qui limite la portée des citations du programme aux types d'article des aires sélectionnées. Cette option constitue la manière la plus pratique de désigner une longue liste d'articles.
 12. Si le mot NOT précède la liste des noms-domaine, tous les types d'articles définis dans toutes les aires sont disponibles, à l'exception de ceux qui ont été spécifiés par NOT.

13. Les effets de REALMS ARE ALL sont les mêmes que RECORDS ARE ALL.
14. Si ni RECORDS, ni REALMS ne sont spécifiés tous les types d'articles de la base de données sont disponibles.
15. L'utilisation des clauses RECORDS/REALMS a pour effet de réduire la taille de la zone UWA ou de la description d'aire de LINKAGE qui sera requise par le programme.

Exemple d'Édition de liens entre programmes

Un programme principal, PP, met à l'état prêt (READY) une aire unique, AIRE-X, d'un schéma SBD-L et appelle deux programmes secondaires, P1 et P2, qui citent le même schéma PP.

Le programme P1 cite deux types d'articles, TA-A et TA-B, tandis que le programme P2 cite trois types d'articles, TA-A, TA-D et TA-G. La structure du programme est indiquée à la page suivante.

```
IDENTIFICATION-DIVISION.
PROGRAM-ID. PP.
...
DATA DIVISION.
SUB-SCHEMA SECTION.
DB SBD-L.
    RECORDS ARE TA-A TA-B TA-D TA-G.
...
    READY AIRE X ...
...
    CALL P1 USING MYARG DB-REGISTERS DB-CXT DB-
PARAMETERS
                TA-A TA-B.
    CALL P2 USING MAX MAY DB-REGISTERS DB-CXT DB-
PARAMETERS
                TA-A TA-D TA-G.
...
    FINISH AIRE-X.
...
    STOP RUN.
IDENTIFICATION-DIVISION.
PROGRAM-ID. P1.
...
DATA DIVISION.
SUB-SCHEMA SECTION.
DB SBD-L.
    DB-DESCRIPTIONS IN LINKAGE SECTION.
    RECORDS ARE TA-A TA-B.
...
PROCEDURE DIVISION USING FUNCX DB-REGISTERS DB-CXT
    DB-PARAMETERS TA-A TA-B.
...
    FIND TA-A ...
...
    EXIT PROGRAM.
IDENTIFICATION-DIVISION.
PROGRAM-ID. P2.
...

```

Langage DML de manipulation des données

```
DATA DIVISION.  
SUB-SCHEMA SECTION.  
DB SBD-L.  
    DB-DESCRIPTIONS IN LINKAGE SECTION.  
    RECORDS ARE TA-A TA-D TA-G.  
    ...  
PROCEDURE DIVISION USING ARA ARB DB-REGISTERS DB-CXT  
    DB-PARAMETERS TA-A TA-D TA-G.  
    ...  
    FIND TA-D ...  
    ...  
    ACCEPT MID FROM TA-A CURRENCY.  
    ...  
    EXIT PROGRAM.
```

5.3 ELEMENTS DB-KEY

Les données contenant les éléments des clés-base-de-données, figurant dans la WORKING STORAGE ou la LINKAGE SECTION, sont définies par USAGE IS DB-KEY :

Numéro-de-niveau identificateur USAGE IS DB-KEY

Ce type de description de donnée est nécessaire pour désigner l'"identificateur" dans les instructions DML suivantes :

```
ACCEPT identificateur FROM ... CURRENCY
ACCEPT ... FROM identificateur REALM-NAME

ACCEPT identificateur FROM ... {NEXT }
                                {PRIOR}
                                {OWNER}
ACCEPT identificateur FROM ... MINIMUM-DB-KEY
FIND ... DB-KEY IS identificateur
```

USAGE DB-KEY est l'équivalent du format USAGE COMP-2 lorsque ces données sont utilisées dans des instructions du COBOL-74.

5.4 REGISTRES SPECIAUX IDS

Il existe des registres spéciaux de la base de données qui sont utilisés pour échanger des informations de la base de données entre le programme et le SGBD. Le programme peut lire le contenu des registres mais il ne peut les modifier. Ces registres sont :

5.4.1 DB-STATUS

Le mot réservé DB-STATUS, permet de renvoyer l'état de chaque instruction du DML après son exécution. Cet état s'exprime sous la forme d'un code retour occupant 7 positions de caractères. Si aucune erreur n'est détectée pendant l'exécution de l'instruction, DB-STATUS est mis à zéro. Lorsqu'une erreur intervient, DB-STATUS est mis à la valeur correspondante. Voir plus loin le paragraphe intitulé "Indicateur d'état de la base de données".

5.4.2 DB-REALM-NAME

Le mot réservé DB-REALM-NAME renvoie le nom d'un domaine à la fin de l'exécution de certaines instructions DML. Le code retour implicite se présente sous la forme d'une chaîne de 30 caractères alphanumériques. Toute exécution d'instruction autre que FIND ou STORE n'a aucun effet sur ce registre. L'exécution réussit d'une instruction FIND et STORE met à jour ce registre en lui attribuant le nom-de-domaine approprié. Une exécution non réussie de la fonction peut fournir un nom-de-domaine dans le registre en plus du code d'erreur-base-de-données fourni dans le registre DB-STATUS. Si le nom-de-domaine n'a pas de signification pour la fonction non exécutée, le registre est mis à blancs.

5.4.3 DB-RECORD-NAME

Le mot réservé DB-RECORD-NAME, renvoie le nom d'un article à la fin de certaines instructions DML. Le code retour implicite est constitué d'une chaîne de 30 caractères alphanumériques. Toute instruction, à l'exception de FIND ou STORE ne modifie pas ce registre. Chaque exécution réussie de FIND et STORE met à jour ce registre avec le nom-article approprié. Une fonction non réussie peut fournir un nom d'article au registre, en plus du code d'erreur-base-de-données fourni dans le registre DB-SATATUS. Si un nom d'article n'a pas de signification pour la fonction non exécutée, ce registre est mis à blancs.

5.4.4 DB-SET-NAME

Le mot réservé DB-SET-NAME renvoie le nom d'un ensemble à la fin de certaines instructions DML. Le code retour implicite est constitué d'une chaîne de 30 caractères alphanumériques. L'exécution réussie d'une instruction n'influe pas sur ce registre. L'exécution non réussie d'une instruction peut fournir un nom-ensemble au registre, en plus du code d'erreur-base-de-données fourni dans le registre DB-STATUS. Lorsque le nom-ensemble n'est pas significatif pour la fonction non exécutée, le registre est mis à blancs.

5.4.5 DB-DETAILED-STATUS

Le mot réservé DB-DETAILED-STATUS, renvoie un message d'explication de la fin d'exécution anormale de certaines instructions DML.

Ce message a le format suivant :

```
01 DB-DETAILED-STATUS .  
   02 DB-MESSAGE-LENGTH USAGE COMP-1 .  
   02 DB- MESSAGE-TEXT PIC LX (256)
```

(suivant la longueur spécifiée dans DB-MESSAGE-LENGTH)

S'il n'y a pas de message, DB-MESSAGE-LENGTH est mis à zéro.

5.5 INDICATEUR D'ETAT DE LA BASE DE DONNEES

Toute exécution d'instruction de manipulation de la base de données fournit, dans le registre spécial DB-STATUS, une valeur. Celle-ci connue sous le nom d'"indicateur d'état de la base de données" est constituée de deux parties: les deux premiers caractères de gauche constituent le code d'instruction, et les cinq derniers caractères de droite, le code d'état. Si, lors de l'exécution d'une instruction de manipulation, une erreur se produit, le code d'instruction indique l'instruction qui a provoqué l'erreur. Si aucune erreur ne se produit pendant l'exécution de l'instruction, les deux codes sont mis à zéro.

5.5.1 Code d'instruction et code d'état

Les codes d'instruction, comme indiqué au Tableau 5-1, sont constitués des deux premiers caractères de gauche de l'indicateur d'état de la base de données.

Les cinq derniers caractères de droite de l'indicateur d'état contiennent le code d'état qui indique une erreur dans la base de données. Ces codes d'état sont listés dans le Tableau 5-2.

5.5.2 Combinaisons des codes d'instruction et d'état

Les combinaisons possibles de codes d'instruction et de codes d'état sont indiquées au Tableau 5-3. L'X, situé aux intersections, indique une combinaison autorisée. Ce tableau se rapporte aux fonctions DML originelles. Pour plus de détails sur les variations de ces fonctions il faut se rapporter à la description individuelle de chaque instruction DML.

5.5.3 Utilisation des registres spéciaux dans la détection d'erreur

Lorsque le SGDB reconnaît certaines erreurs de la base de données, il introduit des valeurs dans les registres spéciaux DB-REALM-NAME, DB-RECORD-NAME et si nécessaire dans les registres DB-DETAILED-STATUS et DB-SET-NAME.

5.5.3.1 DB-REALM-NAME

Ce registre, en cas d'insuccès de l'instruction, contient le nom-de-domaine du domaine concerné par l'erreur détectée dans la base de données. Par exemple: FIND NEXT WITHIN domaine-1 peut renvoyer une erreur "fin de domaine". Le nom du domaine incriminé sera disposé dans le registre DB-REALM-NAME.

5.5.3.2 DB-RECORD-NAME

Ce registre, en cas d'insuccès de l'instruction, contient le nom d'article du type d'article qui a provoqué l'erreur dans la base de données.

5.5.3.3 DB-SET-NAME

Ce registre, en cas d'insuccès de l'instruction, contient le nom d'ensemble du type d'ensemble qui a provoqué l'erreur dans la base de données. Par exemple, FIND NEXT WITHIN nom-ensemble-1 peut renvoyer l'erreur "fin d'ensemble". Dans ce cas, le nom de l'ensemble incriminé est placé dans le registre spécial DB-SET-NAME.

5.5.3.4 DB-DETAILED-STATUS

Ce registre, en cas d'insuccès de l'instruction, contient un message explicatif qui complète la valeur fournie dans DB-STATUS. Par exemple, une instruction STORE peut renvoyer un message qui indique, après exécution des contrôles de cohérence, la zone qui ne remplit pas les conditions de validité.

Tableau 5-1. Codes d'instruction

Instruction	Code d'instruction
ACCEPT	01
CONNECT	02
DISCONNECT	03
ERASE	04
FIND	05
FINISH	06
GET	08
DB Condition (IF)	09
MODIFY	11
READY	13
STORE	15

Tableau 5-2. Codes d'état de base de données

Code	Description de l'erreur
	(ERREUR DANS LA RECHERCHE)
02100	Fin de domaine ou domaine introuvable
02300	Aucun ensemble reconnu ne satisfait les critères de sélection
02400	Aucun article reconnu ne satisfait la locution de sélection d'article
	(ERREUR D'INDICATEUR DE POINT COURANT)
03100	Point courant du domaine, du type-ensemble ou du type-article NUL (ou VIRTUEL)
03200	Point courant de l'unité d'exécution NUL
03300	Article courant de l'unité d'exécution, du domaine ou du type d'ensemble n'est pas un type-article correct
	(ERREUR DE SPECIFICATION)
04100	Valeur de clé-base-de-données non concordante avec nom-domaine
04300	Nom-domaine incorrect (pas de concordance dans le schéma, ou type-article erroné)
	(ERREUR DANS LES DONNEES)
05100	Données en double dans la base-de-données
05200	Données ne répondant pas aux contrôles de cohérence
	(ERREUR DANS LA SUPPRESSION D'ARTICLE)
07200	Suppression d'un ensemble non-vide
	(ERREUR DE RELATION)
08100	Article déjà rattaché à l'ensemble (CONNECT)
08300	Article non encore rattaché à l'ensemble (DISCONNECT OR MODIFY)
	(ERREUR DANS LA MISE A L'ETAT PRET)
09100	Domaine ou base-de-données non à l'état prêt (READY)
09200	Domaine non passé en mode mise à jour (UPDATE)
09300	Domaine déjà à l'état prêt
	(SPECIFIQUES A GCOS 7)
73630	Nombre ordinal nul dans instruction FIND identificateur WITHIN nom-ensemble (ou-nom-domaine)
73640	Modification de CALC-KEY impossible sans migration.
73650	Instruction de mise en relation (MODIFY ou CONNECT) contradictoire à la clause AREA OF OWNER
73660	Absence de description d'article UWA
	(ERREUR DANS L'AFFECTATION DES RESSOURCES)
80200	Manque de place dans le domaine.

Tableau 5-3. Combinaisons des codes d'instruction et d'état

CODE D'ETAT	CODE D'INSTRUCTION											DESCRIPTION DE L'ERREUR
	01 ACC	02 CON	03 DISC	04 ERA	05 FIND	06 FINIS	08 GET	09 IF (Cond DB)	11 MODI	13 REA	15 STO	
02100					X							Fin de domaine ou domaine introuvable
02300		X			X				X		X	Aucun ensemble reconnu ne satisfait les critères de sélection
02400					X							Aucun article reconnu ne satisfait la locution de sélection d'article
03100	X				X			X				Point courant d'article du domaine, du type-ensemble ou du type-article NUL (ou VIRTUEL)
03200	X	X	X	X	X		X	X	X			Point courant d'article de l'unité l'exécution NUL
03300		X	X	X	X		X	X	X			Article courant de l'unité d'exécution du domaine ou du type d'ensemble n'est pas un type-article correct
04100	X				X						X	Valeur de cle-base-de-données non concordance avec nom-domaine
04300					X						X	Nom-domaine incorrect (pas de concordance dans le schema ou type-article erroné)
05100		X							X		X	Données en double dans la base de données
05200					X				X		X	Données ne répondant pas aux contrôles de cohérence
07200				X								Suppression d'un ensemble non-vide
08100		X										Article déjà rattaché à l'ensemble (CONNECT)
08300			X						X			Article non encore rattaché à l'ensemble (DISCONNECT ou MODIFY)
09100	X	X	X	X	X	X	X	X	X		X	Domaine ou base de données non à l'état prêt (READY)
09200		X	X	X					X		X	Domaine non passé en mode mise à jour (UPDATE)
09300									X			Domaine déjà à l'état prêt
73630					X							Nombre ordinal nul dans instruction "FIND identificateur WITHIN nom-ensemble (ou nom-domaine)"
73640									X			Modification de CALC-KEY impossible sans migration
73650		X							X			Instruction de mise en relation (MODIFY ou CONNECT) contradictoire à la clause AREA OF OWNER
73660							X		X			Absence de description d'article UWA
80200											X	Manque de place dans le domaine

5.6 INDICATEURS DE POINT COURANT

Dans un environnement base de données, l'utilisateur doit pouvoir accéder à des articles qui entretiennent entre eux plus d'une relation logique. A la différence des fichiers séquentiels, beaucoup d'articles peuvent suivre logiquement un article quelconque, en fonction de la structure de la base de données et des critères de recherche en usage. Ainsi, il peut y avoir simultanément plusieurs "articles courants" pendant le traitement d'une unité d'exécution et donc plusieurs "pointeurs d'article courant" simultanés. Ces pointeurs simultanés d'article courant sont appelés "indicateurs de point courant". Les indicateurs de point courant sont entièrement tenus à jour par le SGBD tout au long de l'exécution d'un programme.

- Point courant d'unité d'exécution.
- Point courant de type d'ensemble (un pour chaque type d'ensemble).
- Point courant de domaine (un pour chaque domaine).
- Point courant de type d'article (un pour chaque type d'article).

L'indicateur de point courant peut prendre une valeur NUL; ce faisant il indique qu'il n'existe pas d'article courant de ce type. Le point courant d'un type d'ensemble ou d'un domaine peut être VIRTUEL s'il indique, dans l'ensemble ou le domaine, une position qui se situe entre deux articles. C'est généralement le résultat d'une instruction DISCONNECT ou d'une instruction. ERASE. Un indicateur de point courant de ce type est valable pour toutes les instructions DML à l'exception de "FIND CURRENT WITHIN nom-ensemble ou nom-domaine", "FIND DUPLICATE WITHIN nom-ensemble ...", "ACCEPT ... FROM nom-ensemble ou nom-domaine CURRENCY ou REALM-NAME" ainsi que pour la "Condition-base de-données (IF)".

5.6.1 Point courant d'unité d'exécution

Le point courant de chaque unité d'exécution est tenu à jour de manière à identifier l'article paramètre dans certaines instructions de manipulation de données qui ne spécifient pas d'article particulier. Ceci amène le SGBD à mettre à jour, pour certains types d'instruction, son indicateur de point courant sur un autre article, alors que l'exécution de certains autres types d'instruction n'affectent pas le pointeur de l'article courant de l'unité d'exécution.

5.6.2 Point courant de type d'ensemble

Pour chaque rubrique de description d'ensemble, le SGBD tient à jour un indicateur séparé. L'indicateur de chaque type d'ensemble est tenu à jour par le SGBD de manière à identifier l'article de ce type d'ensemble qui vient d'être traité par la dernière instruction. Sauf indication fournie par la locution RETAINING, un article, lorsqu'il est l'objet d'un accès, positionne les indicateurs de point courant de tous les ensembles auxquels il appartient en tant que maître ou détail. Une instruction DISCONNECT ou une

instruction ERASE peut disposer les indicateurs de point courant sur la position, située entre deux articles, de l'ancien article courant.

5.6.3 Point courant de domaine

Pour chaque rubrique de description de domaine, le SGBD tient à jour un indicateur de point courant séparé. Sauf indication contraire fournie par la locution RETAINING, le SGBD tient à jour un indicateur de point courant pour chaque domaine de manière à identifier l'article du domaine auquel la dernière instruction vient de faire référence. Un instruction ERASE peut positionner l'indicateur de point courant, entre deux articles, sur l'emplacement de l'ancien article courant.

5.6.4 Point courant de type d'article

Pour chaque rubrique de description d'article, le SGBD entretient un indicateur de point courant séparé. Sauf indication contraire fournie par la locution RETAINING, l'indicateur de chaque type d'article est entretenu par le SGBD de manière à identifier le type d'article auquel la dernière instruction vient de faire référence.

5.6.5 Valeurs de point courant

En plus des indicateurs de point courant, l'utilisateur a la possibilité de conserver une valeur clé-base-de-données pour chaque article ou ensemble auquel il accède. Une instruction telle que :

```
ACCEPT SAVE-RECORD-03-DBK FROM RECORD-03 CURRENCY.
```

permettra de spécifier un mode de relation MANUAL pour l'article courant de ce type. Pour certaines structures complexes, il peut être utile de garder sous le contrôle du programme les valeurs de point courant utiles car beaucoup de fonctions DML peuvent modifier les indicateurs de point courant. Dans l'exemple ci-dessus, lorsque l'article est utilisé ultérieurement, une instruction telle que :

```
FIND RECORD-03 DB-KEY IS SAVE-RECORD-03-DBK.
```

permettra de rétablir l'article ainsi sauvegardé comme point courant de l'unité d'exécution et de remettre tous les indicateurs de point courant à jour pour cet article.

5.6.6 Répercussion des instructions du DML sur les indicateurs de point courant

Chaque instruction du DML a des incidences sur les quatre types d'indicateurs de point courant comme le montre le Tableau 5-4. Ce tableau indique les points courants utilisés comme paramètres implicites en entrée et les mises à jour successives de point courant lors de l'exécution réussie de ces instructions.

Une instruction du format "FIND FIRST WITHIN domaine-1" établit un point courant de l'unité d'exécution, du domaine, du type d'article, et de tout type d'ensemble pour lequel l'article est maître ou détail. L'article localisé le premier dans le domaine, met à jour l'indicateur de point courant des ensembles appropriés. Ainsi, si l'instruction "FIND NEXT WITHIN nom-ensemble-1" est utilisé, les indicateurs de point courant pour l'unité d'exécution, le domaine, le type d'article et les types d'ensemble sont tous mis à jour pour répercuter le premier article qui a été trouvé dans cet ensemble. Si l'ensemble est vide, une erreur "fin d'ensemble" est renvoyée et les indicateurs de point courant ne sont pas mis à jour pour l'unité d'exécution, le domaine, le type d'article et les types d'ensembles.

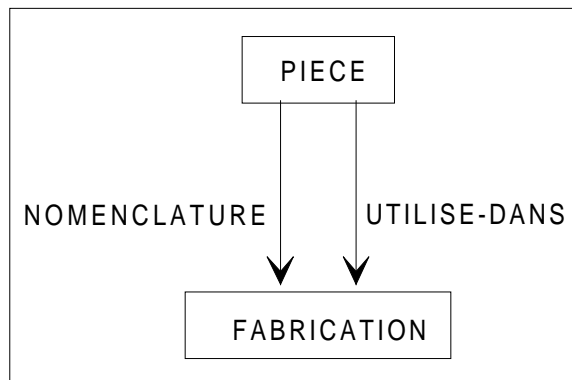
Une instruction STORE qui a pour but de ranger un nouvel article dans la base de données, a des répercussions sur les indicateurs de point courant de l'unité d'exécution, du domaine et du type d'article. Il y a également modification pour tous les types d'ensemble dans lesquels l'article est maître ou détail.

Une instruction ERASE, qui a pour but de supprimer un article de la base de données, a des répercussions sur les indicateurs de point courant. Le point courant de l'unité d'exécution est mis à NUL. Le point courant du type d'article est mis à NUL si l'article supprimé était l'article courant pour ce type d'article. Le point courant du type d'ensemble et du domaine sont positionnés à VIRTUEL si l'article supprimé était article courant du type d'ensemble et/ou du domaine respectivement. Même si l'article courant du type d'ensemble ou du domaine n'est plus disponible, l'indicateur de point courant donne comme position courante celle qui se situe entre les deux articles entre lesquels se trouvait l'article courant.

L'exécution non réussie d'une instruction DML ne modifie pas les indicateurs de point courant. Par exemple, si l'instruction "FIND NEXT WITHIN nom-ensemble-1" renvoie un état "fin d'ensemble", l'indicateur de point courant pour l'ensemble-1 continuera à indiquer le dernier article du type d'ensemble. Si l'ensemble est vide, l'indicateur pointerait sur l'article maître qu'il avait préalablement identifié avant l'exécution de l'instruction FIND. A noter que l'exécution réussie d'une instruction DML, met à jour les registres DB-REALM-NAME, DB-RECORD-NAME et DB-SET-NAME.

Il existe certaines circonstances dans lesquelles la mise à jour automatique des indicateurs de point courant n'est pas souhaitable. La locution RETAINING CURRENCY FOR peut s'utiliser avec une instruction FIND, STORE, MODIFY ou CONNECT pour supprimer la mise à jour automatique des indicateurs de point courant. Par exemple, dans une structure de réseau complexe, comme celle indiquée dans l'exemple qui suit, et comprenant un article PIECE et un article FABRICATION utilisé pour générer une décomposition des pièces, il peut être souhaitable de ne pas modifier le point courant du type NOMENCLATURE lorsqu'on procède au listage de tous les composants entrant dans la chaîne d'assemblage.

Exemple :



Une PIECE spécifique est localisée au moyen d'une instruction FIND ANY PIECE qui établit cet article comme étant l'article courant des ensemble NOMENCLATURE et UTILISE-DANS. Lorsque l'on parcourt l'ensemble NOMENCLATURE pour déterminer tous les éléments de PIECE, il est nécessaire de préserver le point courant lorsque l'on va de l'article FABRICATION à l'article maître PIECE. L'opération sera écrite comme indiqué ci-dessous :

```

    FIND ANY PIECE
  LOOP-ON-COMPOSANTS
    FIND NEXT FABRICATION WITHIN NOMENCLATURE
    IF DB-STATUS EQUAL "0502100" GO TO LAST-COMPOSANT.
    FIND OWNER PIECE WITHIN UTILISE-DANS
      RETAINING CURRENCY FOR NOMENCLATURE
    GET.
    .
    .
    .

  GO TO LOOP-ON-COMPOSANTS
  
```

Si la locution RETAINING CURRENCY FOR NOMENCLATURE avait été omise, tous les articles PIECE trouvés auraient mis à jour le point courant de cet ensemble et auraient probablement conduit le programme de listage sur un itinéraire erroné.

Tableau 5-4. Utilisation et mise à jour des points courants (1/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
ACCEPT...FROM CURRENCY	Point-courant-d'unité-d'exécution	
ACCEPT...FROM nom-domaine CURRENCY	Point-courant-de-domaine spécifié	
ACCEPT...FROM nom-article CURRENCY	Point-courant-de-type-d'article spécifié	
ACCEPT...FROM nom-ensemble CURRENCY	Point-courant-de-type-d'ensemble spécifié	
ACCEPT...FROM REALM-NAME	Point-courant-d'unité-d'exécution	
ACCEPT...FROM nom-article REALM-NAME	Point-courant-de-type-d'article spécifié	
ACCEPT...FROM nom-ensemble REALM-NAME	Point-courant-de-type-d'ensemble spécifié	
ACCEPT...FROM nom-ensemble NEXT/PRIOR/OWNER	Point-courant-de-type-d'ensemble spécifié	
CONNECT...TO nom-ensemble	<ul style="list-style-type: none"> - Point-courant d'unité d'exécution utilisé comme paramètre dans l'instruction. - Si APPLICATION identifie le premier niveau de l'itinéraire de recherche, le point-courant-du-type-d'ensemble correspondant est utilisé. De plus, si l'itinéraire de recherche ne comprend qu'un seul niveau et si l'ordre de classement indiqué est NEXT ou PRIOR, le point-courant-de-type-d'ensemble détermine le point d'insertion de l'article 	Mise à jour du point-courant-du-type-d'ensemble spécifié sauf indication contraire d'une clause RETAINING CURRENCY
DISCONNECT...FROM nom-ensemble	Point-courant-d'unité d'exécution utilisé comme paramètre dans l'instruction	Le point-courant-du-type-d'ensemble spécifié est passé à l'état VIRTUEL si l'article désigné était l'article courant de cet ensemble

Tableau 5-4. Utilisation et mise à jour des points courants (2/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
ERASE... [ALL MEMBERS]	Point-courant-de-l'unité-d'exécution utilisé comme paramètre	<ul style="list-style-type: none"> - Point-courant de l'unité d'exécution mis à l'état NUL, - Point-courant-de-domaine passé à l'état VIRTUEL si un article supprimé était article courant de son domaine, - Point-courant-de-type-d'article passé à l'état NUL si un article supprimé était article courant de son type d'article, - Point-courant-de-type-d'ensemble mis à l'état NUL -si l'article annulé était l'article courant de l'ensemble- pour tous les ensembles dans lesquels cet article figurait comme maître, - Point-courant-de-type-d'ensemble mis à l'état VIRTUEL -si l'article annulé était article courant de l'ensemble- pour tous les ensembles dans lesquels cet article figurait comme article détail réel
FIND...DB-KEY IS		- Point-courant-de-l'unité d'exécution
FIND ANY nom-article		- Point-courant-de-domaine, sauf RETAINING CURRENCY

Tableau 5-4. Utilisation et mise à jour des points courants (3/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
FIND DUPLICATE nom-article	Point-courant de l'unité d'exécution utilisé pour reconnaître un type d'article CALC et la position dans la chaîne de "synonymes"	<ul style="list-style-type: none"> - Point-courant-du-type-d'article, sauf clause RETAINING CURRENCY, - Point-courant-du-type-d'ensemble, sauf clause RETAINING CURRENCY, pour tous les ensembles dans lesquels il figurait comme maître ou comme article détail avec option AUTOMATIC-MANDATORY, ou encore comme article détail couramment rattaché à l'ensemble avec MANUAL-OPTIONAL
FIND FIRST/LAST ordinal... WITHIN nom-domaine		<ul style="list-style-type: none"> - Point-courant-de-l'unité-d'exécution - Point-courant-du-domaine, sauf clause RETAINING CURRENCY - Point courant du type d'article' sauf clause RETAINING CURRENCY, - Point courant du type d'ensemble, sauf clause RETAINING CURRENCY, pour tous les types d'ensembles dans lesquels l'article est déclaré maître, ou article détail avec option AUTOMATIC MANDATORY ou encore article détail couramment rattaché à l'ensemble avec option MANUAL OPTIONAL.
FIND NEXT/PRIOR... WITHIN nom-domaine	- Point-courant-de-domaine spécifié qui permet d'identifier la position de l'article dans le domaine	
FIND FIRST.LAST ordinal... WITHIN nom-ensemble	- Point-courant-de-type-d'article spécifié qui permet d'identifier l'ensemble	

Tableau 5-4. Utilisation et mise à jour des points courants (4/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
FIN NEXT/PRIOR... WITHIN nom-ensemble	- Point-courant-de-type-d'ensemble spécifié qui permet d'identifier l'ensemble et la position de l'article dans l'ensemble	
FIND OWNER WITHIN nom-ensemble	- Point-courant-de-type-d'ensemble spécifié, qui permet d'identifier l'ensemble	
FIND CURRENT	- Point-courant-de-l'unité-d'exécution	
FFIND CURRENT nom-article	- Point-courant-de-type-d'article spécifié	
FIND CURRENT... WITHIN nom-domaine	- Point-courant-de-domaine spécifié	
FIND CURRENT... WITHIN nom-ensemble	- Point-courant-de-type-ensemble spécifié	
FIND nom-article WITHIN nom-ensemble CURRENT [USING...]	- Point-courant-du-type-d'ensemble spécifié qui permet d'identifier l'ensemble	
FIND nom-article WITHIN nom-ensemble [USING...]	- Point-courant-du-type-d'ensemble correspondant, si le premier niveau de l'itinéraire de recherche est défini par APPLICATION	
FIND DUPLICATE WITHIN nom-ensemble [USING...]	- Point-courant-du-type-d'ensemble spécifiée qui permet d'identifier le type d'article, l'ensemble et la position de l'article dans l'ensemble	
FINISH... ordinal...WITHIN nom-domaine		Mise à l'état NUL de tous les points-courants qui ont permis de citer les articles dans les domaines.
GET	- Point-courant-de-l'unité-d'exécution, utilisé comme paramètre de l'instruction	
OWNER/MEMBER/TENANT CONDITION	- Point-courant-de-l'unité-d'exécution	
nom-ensemble EMPTY CONDITION	- Point-courant-du-type-d'ensemble spécifié, qui permet d'identifier l'ensemble	
MODIFY (pas de clause MEMBERSHIP)	- Point-courant-de-l'unité-d'exécution utilise comme paramètre de l'instruction	

Tableau 5-4. Utilisation et mise à jour des points courants (5/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
MODIFY...MEMBERSHIP	- Point-courant-de-l'unité-d'exécution utilisé comme paramètre de l'instruction	- Point-courant-de-domaine, sauf clause RETAINING CURRENCY
FIND CURRENT nom-article	- Si, pour un nouvel ensemble, le premier niveau de l'itinéraire de recherche est identifié par APPLICATION, utilisation du point-courant-de-type-d'ensemble. Egalement, si l'itinéraire de recherche ne dispose que d'un seul niveau et que l'ordre de classement est NEXT ou PRIOR, utilisation du point-courant-de-type-d'ensemble pour indiquer le point d'insertion.	-Point-courant-de-type-d'ensemble, sauf clause -RETAINING CURRENCY, pour tous les ensembles dans lesquels l'article est maître ou dans lesquels l'article est détail avec option AUTOMATIC MANDATORY, ou encore article détail couramment rattaché à l'ensemble avec option MANUAL-OPTIONAL.
FIND CURRENT...WITHIN nom-domaine		
FIND CURRENT...WITHIN nom-ensemble		
FIND nom-article WITHIN nom-ensemble		
READY...		

Tableau 5-4. Utilisation et mise à jour des points courants (6/6)

INSTRUCTION	POINT COURANT UTILISE EN ENTREE	MISE A JOUR DES POINTS COURANTS APRES EXECUTION REUSSIE
STORE nom-article	- Utilisation du point-courant-de-type-d'ensemble si le premier niveau de l'itinéraire de recherche est identifié par APPLICATION. De plus, si l'itinéraire de recherche ne porte que sur un seul niveau et si l'ordre de classement est NEXT ou PRIOR, utilisation du point-courant-de-type d'ensemble qui permet de désigner le point d'insertion.	- Point-courant-de-l'unité-d'exécution - Point-courant-du domaine, sauf clause RETAINING CURRENCY - Point-courant-de-type-d'article sauf clause RETAINING CURRENTY - Point-courant-de type-d'ensemble, sauf clause RETAINING CURRENTY, pour tous les ensembles dans lesquels l'article est déclaré maître ou article détail avec option AUTOMATIC MANDATORY, ou encore article détail rattaché a l'ensemble avec option MANUAL OPTIONAL
IDS remise a l'état initial		Tous les indicateurs de point courant sont mis a l'état NUL.

5.7 INSTRUCTIONS DML

Le langage DML consiste en une série d'instructions listée ci-dessous :

ACCEPT	Permet d'obtenir les clés de la base de données ou les noms de domaine désignés par les indicateurs de point-courant du SGBD, ou d'extraire de l'information DMCL du schéma résultant.
CONNECT	Permet de rattacher un nouvel article détail à un ensemble pour lequel l'article a été déclaré MANUAL OPTIONAL.
DISCONNECT	Permet d'éliminer un article détail, défini avec MANUAL OPTIONAL, d'un ensemble.
ERASE	Permet d'effacer un article détail de la base de données, sous réserve que cet article ne soit pas lui-même maître d'autres articles détails non effacés.
FIND	Permet, à l'aide de diverses options de sélection d'article, de localiser tout article de la base de données.
FINISH	Permet d'interdire l'accès à un domaine.
GET	Permet d'obtenir le contenu d'un article courant.
DB-CONDITION (IF)	Permet de vérifier les conditions de la base de données.
MODIFY	Permet de modifier le contenu des données de la base de données et/ou les relations d'un article dans un ensemble.
READY	Permet de rendre accessible au traitement le contenu d'un domaine.
STORE	Permet d'ajouter un nouvel article à la base de données.
USE	Permet de définir la procédure DB-EXCEPTION qui sera fournie automatiquement.

Chaque instruction DML est décrite en détail dans les pages qui suivent. Le tableau intitulé "valeurs de DB-STATUS" qui clot la description de chacune des instructions, doit être interprété de la manière suivante :

Un trait de soulignement "_" dans les colonnes REALM, RECORD ou SET indique que le SGBD ne modifie pas le contenu des états correspondants DB-REALM-NAME, DB-RECORD-NAME ou DB-SET-NAME.

UN blanc dans les colonnes REALM, RECORD ou SET indique que le SGBD met à blancs les registres correspondants.

Un "X" dans les colonnes REALM, RECORD ou SET indique que SGBD dispose dans le registre correspondant le nom de domaine, d'article ou d'ensemble. Si "X" est encadré par des parenthèses, le registre contiendra un nom ou des blancs, suivant l'erreur détectée.

L'ordre dans lequel les causes d'erreur sont indiquées pour une instruction DML, reflète, dans la plupart des cas, la séquence de vérification exécutée par le SGBD pendant l'exécution de la fonction DML.

5.7.1 ACCEPT

Fonction

Permet de rendre accessible au programme le contenu des indicateurs de point courant spécifiés, de déduire le nom de domaine qui correspond à une valeur clé-base-de-données, de fournir les clés-base-de-données aux pointeurs NEXT, PRIOR et OWNER, et enfin de lire les informations du DMCL à partir du schéma résultant.

Format général

Format 1

```
ACCEPT identificateur-1 FROM [nom-article ]
                                [nom-ensemble ] CURRENCY
                                [nom-domaine ]
```

Format 2

```
ACCEPT identificateur-2 FROM [nom-article ]
                                [nom-ensemble ] REALM-NAME
                                [identificateur-3]
```

Format 3

```
ACCEPT identificateur-4 FROM nom-ensemble {NEXT }
                                                {PRIOR }
                                                {OWNER }
```

Format 4

```
ACCEPT identificateur-5 FROM nom-domaine LINES-PER-PAGE
```

Format 5

```
ACCEPT identificateur-6 FROM nom-domaine MINIMUM-DB-KEY
                                                [OF nom-article]
```

Format 6

```
ACCEPT identificateur-7 FROM nom-domaine NUMBER-OF-PAGES
                                                [OF nom-article]
```

Règles de syntaxe

1. Identificateur-1, identificateur-3, identificateur-4 et identificateur-6 doivent être des éléments DB-KEY.
2. Identificateur-2 doit être une donnée élémentaire alphanumérique. Il doit fournir un chiffre suffisamment grand pour contenir tout nom-de-domaine (REALM-NAME). La taille minimale à fournir est 30 (PIC X(30)).
3. Identificateur-5 doit être un entier élémentaire définissant une plage au moins égale à celle de COMP-1
4. Identificateur-7 doit être un entier élémentaire définissant une plage au moins égale à celle de COMP-2.

Règles générales

Format 1

1. Identificateur-1 exprime la valeur de la clé-de-base-de-données de l'article courant du type d'article, type d'ensemble ou nom de domaine spécifié suivant le cas par nom-article, nom-ensemble ou nom-domaine.
2. Si un paramètre nom-article, nom-ensemble ou nom-domaine n'est pas spécifié, la valeur clé-base-de-données de l'article courant de l'unité d'exécution est exprimé par identificateur-1
3. L'indicateur de point courant désigné ne doit être ni NUL, ni à VIRTUEL.

Format 2

1. Identificateur-3 exprime la valeur de clé-base-de-données à partir de laquelle est déduit le nom de domaine. Celui-ci peut être également déduit de l'indicateur de point courant exprimé par identificateur-2 suivant les règles de transfert des éléments alphanumériques.
2. Si, ni identificateur-3, ni un paramètre nom-article ou nom-ensemble n'est spécifié, le nom de domaine de l'article courant de l'unité d'exécution est exprimé par identificateur-2 suivant les règles de transfert des éléments alphanumériques.
3. L'indicateur de point-courant désigné ne doit présenter une valeur, ni NUL, ni VIRTUEL.

Format 3

1. L'indicateur de point courant de nom-ensemble doit présenter un article (maître ou détail) ou désigner une position (VIRTUEL).
2. Identificateur-4 indique la clé-base-de-données de l'article NEXT, PRIOR ou OWNER de l'ensemble, ou une position.

L'article NEXT ou PRIOR peut être lui-même article maître, la "fin-d'ensemble" n'existant pas. L'ensemble peut être vide.

Format 4

1. Le nombre de lignes par page du nom-domaine est transféré dans la donnée utilisateur représentée par identificateur-5.

Format 5

1. identificateur-6 exprime la plus petite (première) valeur de clé-base-de-données du nom-domaine. Si l'option nom-article est utilisé dans la clause, la valeur est la plus petite, ou la première de clé-base-de-données du domaine pour cette plage de types d'article.

Format 6

1. Le nombre de pages de nom-domaine est exprimé par identificateur-7. Si l'option nom-article est utilisée, la taille, en pages, de la plage d'articles de nom-domaine est exprimée par identificateur-7.

FUNCTION	CAUSE D'ERREUR	REGISTRES SPECIAUX IDS			
		DB-STATUS	REALM	RECORD	SET
	Aucune	0000000	-	-	-
	Non-ouverture de la BD	0109100			
ACCEPT...CURRENCY	Indicateur P.C. de l'unité d'exécution à NUL	0103200			
	Indicateur P.C. de l'aire à NUL ou VIRTUEL	0103100	X		
	Indicateur P.C. du type d'ensemble à NUL ou VIRTUEL	0103100		X	
ACCEPT...REALM-NAME	Indicateur P.C. de l'unité d'exécution à NUL	0103200			X
	Indicateur P.C. du type d'article à NUL	0103100			
	Indicateur P.C. du type d'ensemble à NUL ou VIRTUEL	0103100		X	
	Incohérence de la clé-base-de-données	0104100			X
ACCEPR...NEXT/PRIOR/OWNER	Incohérence	0104100			
	Indicateur P.C. du type d'ensemble à NUL	0103100			X

5.7.2 CONNECT

Fonction

Permet de rattacher un article enregistré dans la base de données à un ensemble désigné. L'article doit avoir été déclaré `MANUAL OPTIONAL` pour cet ensemble. L'article courant de l'unité d'exécution constitue le paramètre en entrée de cette instruction.

Format général

```
CONNECT      [nom-article ] TO nom-ensemble
[ RETAINING CURRENCY FOR      {SETS      } ]
[                                     {nom-ensemble} ]
```

Règles de syntaxe

1. Nom-article n'est pas obligatoire. S'il est mentionné, il doit correspondre à l'article courant de l'unité d'exécution.
2. L'article courant de l'unité d'exécution doit avoir été défini comme article détail `MANUAL OPTIONAL` de l'ensemble spécifié dans l'instruction `CONNECT`.

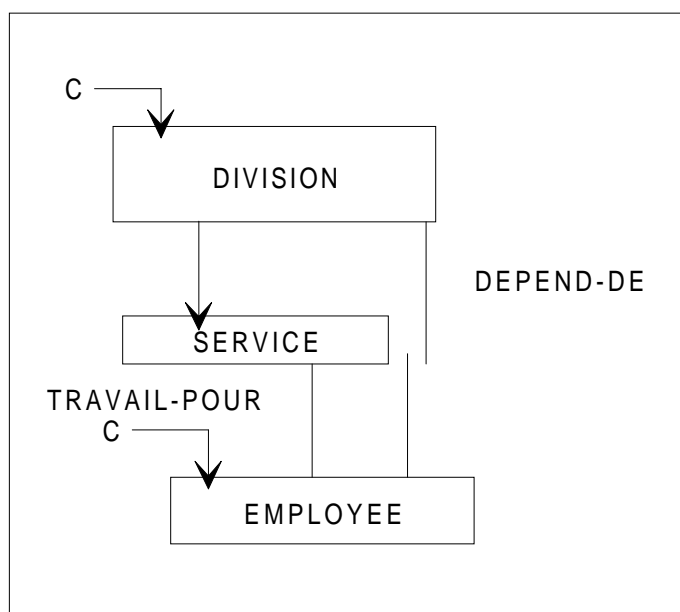
Règles générales

1. L'exécution de la fonction `CONNECT` rend l'article courant de l'unité d'exécution article détail du type d'ensemble spécifié sous réserve qu'il ne soit déjà rattaché à une apparition de ce type d'ensemble.
2. Ce sont les critères de sélection d'ensemble qui déterminent l'apparition d'ensemble à laquelle l'article courant de l'unité d'exécution doit être rattaché. La manière dont l'article est rattaché à cette apparition de l'ensemble est déterminée par les critères de classement de l'ensemble.
3. Les domaines dans lesquels l'article courant et les articles de l'ensemble concerné sont rangés doivent être ouverts en mode mise à jour (`UPDATE`).
4. Si la locution `RETAINING` n'est pas spécifiée, l'article courant de l'unité d'exécution devient article courant de l'ensemble auquel il a été rattaché. Les autres indicateurs de point courant ne sont pas concernés.
5. Si la locution `RETAINING` est mentionnée, aucun indicateur de point courant n'est modifié.

- Remarques :**
1. Le processus de sélection d'ensemble utilise les valeurs de zone ou de paramètre-base-de-données de la zone UWA. Le processus d'insertion d'ensemble utilise les valeurs de clé de tri ou de zone de contrôle de non duplication de l'article de la base de données.
 2. Si une erreur se produit pendant l'exécution d'une instruction CONNECT, la base de données est remise à l'état dans lequel elle se trouvait avant l'exécution de la commande. Cette opération s'effectue sans utilisation des journaux. Une fonction non réussie ne modifie pas les indicateurs de point courant.

Exemple :

L'article EMPLOYE est un article détail de type MANUAL OPTIONAL de l'ensemble DEPEND-DE, l'itinéraire de sélection n'ayant qu'un seul niveau identifié par CALC-KEY.



La séquence du DML rattachant un employé à DIVISION est la suivante :

```

MOVE "mmm" TO NOM-EMPLOYEE.
FIND ANY EMPLOYEE
    
```

```

MOVE "nnnn" TO DIVISION.
CONNECT EMPLOYEE TO
DEPEND-DE
    
```

Indique le point courant de l'unité d'exécution,

Détermine l'itinéraire de sélection d'ensemble,

Rattache l'article à son ensemble.

Langage DML de manipulation des données

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
	Aucune; article rattaché	0000000	-	-	-
	Non-ouverture de la BD	0203200			
CONNECT	Indicateur P.C. d'UE à NUL	0203300			
	P.C. d'UE incorrect (si spécifié, ou article détail non déclaré MANUAL OPTIONAL pour l'ensemble	0203300		X (PCUE)	X
	P.C. d'UE article OPTIONAL déjà rattaché	0208100		X	X
	Aire non à l'état prêt (READY)	0209200	X	X	(X)
	Aire non en mode mise à jour	0209200	X	X	(X)
	Violation de condition "AREA OF OWNER"	0273650	X (maitre)	X	X
	Sélection d'ensemble non réussie	0202300		X	X
	Clause DUPLICATE NOT ALLOWED non respectée (clé-de-tri ou autre zone de contrôle)	0205100		X	X

5.7.3 DISCONNECT

Fonction

Elimine logiquement un article d'un ensemble spécifié sous réserve que l'article ait été déclaré article détail MANUAL OPTIONAL de l'ensemble. Le point courant de l'unité d'exécution constitue le paramètre de l'instruction.

Format général

```
DISCONNECT [nom-article] FROM nom-ensemble
```

Règles de syntaxe

1. nom-article n'est pas obligatoire. S'il est spécifié il doit être le même que celui du point courant de l'unité d'exécution.
2. L'article représentant le point courant de l'unité d'exécution doit avoir été défini comme article détail MANUAL OPTIONAL de l'ensemble spécifié dans l'instruction DISCONNECT.

Règles générales

1. L'exécution de la fonction DISCONNECT provoque l'élimination de l'article courant de l'unité d'exécution de l'ensemble spécifié par nom-ensemble sous réserve qu'il ait été préalablement rattaché à ce type d'ensemble.
2. Les domaines dans lesquels l'article courant de l'unité d'exécution et les articles de l'ensemble concerné sont enregistrés doivent être ouverts en mode mise à jour (UPDATE).
3. Si l'article courant de l'unité d'exécution est en même temps l'article courant du type d'ensemble spécifié, l'indicateur de point courant pour ce type d'ensemble est mis à jour pour permettre de reconnaître la position, entre deux articles, qu'occupait l'article éliminé. Si l'article courant de l'unité d'exécution n'est pas l'article courant du type d'ensemble spécifié, l'indicateur de point-courant n'est pas modifié. Les autres indicateurs de point-courant ne le sont pas non plus.

Langage DML de manipulation des données

- Remarques :**
1. L'article courant du type d'ensemble spécifié peut, comme indiqué dans la règle générale 3, n'être plus disponible. Cependant, les articles indiqués par NEXT, PRIOR et OWNER le sont. Cela signifie que l'indicateur de point courant reste valable pour toutes les instructions DML suivantes qui utilisent une position plus qu'un article particulier. Par exemple, la séquence suivante reste valable :

```
DISCONNECT FROM nom-ensemble-1.  
FIND NEXT WITHIN nom-ensemble-1.
```

2. Si une erreur se produit dans la base de données au moment de l'exécution d'une instruction DISCONNECT, la base de données est remise à l'état dans lequel elle se trouvait, sans utilisation des journaux, avant l'exécution de l'instruction. De plus, les indicateurs de point courant ne sont pas modifiés par l'exécution non réussie d'une instruction.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
DISCONNECT	Aucune; l'article est éliminé	0000000	-	-	-
	Non-ouverture de la BD	0309100			
	Indicateur P.C. d'UE à NUL	0303200			
	P.C. d'UE incorrect (si spécifié ou article détail non déclaré MANUAL OPTIONAL pour l'ensemble)	0303300		X (P.C UE)	X
	P.C. d'UE portant sur article OPTIONAL non rattaché antérieurement	0308300		X	X
	Aire non à l'état prêt (READY)	0309100	X	X	
	Aire non en mode mise à jour	0309200	X	X	

5.7.4 ERASE

Fonction

Permet de supprimer un ou plusieurs articles de la base de données. L'article courant de l'unité d'exécution est le paramètre de l'instruction.

Format général

```
ERASE [nom-article] [ALL MEMBERS]
```

Règle de syntaxe

Nom-article n'est pas obligatoire. S'il est mentionné, il doit être le même que celui de l'article courant de l'unité d'exécution.

Règles générales

1. L'exécution de la fonction ERASE provoque la suppression d'un ou plusieurs articles de la base de données.
2. Si ALL n'est pas spécifié et que l'article courant de l'unité d'exécution n'est pas le maître d'un ensemble non vide, l'article est d'abord éliminé des ensembles auxquels il est éventuellement rattaché, puis éliminé de la base de données.
3. Si ALL n'est pas spécifié et que l'article courant de l'unité d'exécution est le maître d'un ensemble non vide, il en résulte une erreur.
4. Si ALL étant spécifié, l'article courant de l'unité d'exécution est le maître d'un ensemble non vide, la suppression de l'article courant de l'unité d'exécution et de tous ses articles détails est opérée suivant les modalités expliquées à la règle 2. La suppression d'un article dans ces conditions est équivalent à celle qui aurait été opérée par une instruction ERASE ... ALL.

Le processus se répète jusqu'à ce que tous les articles qui possèdent des liens hiérarchiques entre eux aient été supprimés de la base de données.

Remarque : Il ne faut donc utiliser l'expression ALL qu'avec prudence.

5. La spécification ALL n'est pas prise en compte si l'article courant de l'unité d'exécution n'est pas le maître d'un ensemble non vide.
6. Les domaines dans lesquels l'article courant de l'unité d'exécution et les articles des ensembles concernés sont rangés doivent être ouverts en mode mise à jour.
7. Les modifications d'indicateurs de point courant s'effectuent de la manière suivante lors de la suppression d'article de la base de données :
 - a) Le point courant de l'unité d'exécution est mis à NUL.
 - b) Le point courant de type d'article est mis à NUL si l'article supprimé était courant du type d'article.
 - c) Le point courant de type d'ensemble est mis à NUL si l'article supprimé était courant du type d'ensemble pour lequel il était maître.
 - d) L'indicateur de point courant du type d'ensemble est mis à jour de manière à identifier la position qu'occupait l'article supprimé entre deux articles, si cet article supprimé constituait le point courant du type d'ensemble auquel il était rattaché.
 - e) L'indicateur de point courant du domaine est mis à jour pour refléter la position qu'occupait l'article désormais supprimé.

Remarques :

1. Comme indiqué dans les règles générales 2 et 4, chaque article paramètre est éliminé de tous les ensembles auxquels il était précédemment rattaché. Cela signifie qu'il n'y a pas d'indications particulières à fournir pour les ensembles MANUAL OPTIONAL. Si un article spécifié est couramment rattaché à un ensemble MANUAL OPTIONAL, il sera éliminé de cet ensemble avant d'être supprimé de la base de données. Si un article est déjà éliminé d'un ensemble MANUAL OPTIONAL il n'en résulte aucune erreur et l'article est simplement supprimé de la base de données.

2. Pour des raisons décrites à la règle 7d, l'article courant du type d'ensemble peut n'être plus disponible. Cependant les articles indiqués par NEXT, PRIOR et OWNER le sont. Ceci signifie que l'indicateur de point courant reste valable pour les instructions DML suivantes qui citent une position à l'intérieur de l'ensemble plutôt qu'un article particulier de l'ensemble. Par exemple la séquence suivante reste valable

```
ERASE B.  
FIND NEXT WITHIN A-B.
```

dans lesquelles l'article B est article détail de l'ensemble A-B.

3. Pour des raisons expliquées à la règle générale 7e, l'article courant d'un domaine peut n'être plus disponible. Les articles indiqués par NEXT et PRIOR le sont. Ceci signifie que l'indicateur de point courant reste valable pour les instructions DML suivantes citant une position dans le domaine plutôt qu'un article particulier de ce domaine.

4. a) Si une erreur se produit dans la base de données pendant l'exécution d'une instruction ERASE, sans suppression d'article détail, la base de données est remise à l'état dans lequel elle se trouvait au moment de l'exécution de l'instruction, sans utilisation des journaux. Les indicateurs de point courant ne sont pas modifiés et le contrôle repasse au programme appelant. La valeur appropriée est affichée dans le registre DB-STATUS.

b) Si une erreur se produit dans la base de données pendant l'exécution d'une instruction ERASE, avec suppression d'article détail, mais avant que la mise à jour de la base de données ait réellement commencé, le SGBD procède de la manière indiquée ci-dessus.

c) Si une erreur se produit dans la base de données pendant l'exécution d'une instruction ERASE, avec suppression d'article détail, mais avant que la mise à jour de la base de données ait réellement commencé, (soit que l'aire ne soit pas à l'état prêt ou en mode mise à jour), la suite du traitement dépend du niveau de protection de la base de données :

- si la base de données bénéficie de la protection du journal AVANT, que ce soit en modes traitement par lots, IOF ainsi que TDS, la base de données est remise à l'état dans lequel elle se trouvait avant l'exécution de l'instruction et les indicateurs de point courant ne sont pas modifiés. Le contrôle repasse au programme appelant avec la valeur appropriée dans le registre DB-STATUS.

- si la base de données ne bénéficie pas de la protection du journal AVANT en traitement par lots ou IOF, la base de données est laissée dans un état instable et l'activité s'arrête prématurément. La base de données doit alors faire l'objet d'une restauration par utilitaire FILREST ou VOLREST.

- si la base de données est sous la protection du journal APRES et du dispositif de mise à jour différée de TDS, la base de données est remise à l'état dans lequel elle se trouvait au moment du dernier point de consolidation. La routine TPR s'arrête prématurément et n'est pas relancée.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
Générale	Aucune les articles sont effacés	0000000	-	-	-
	Non-ouverture de la BD	0409100			
	Indicateur P.C. d'UE à NUL	0403200			
	P.C. d'UE incorrect (si spécifié)	0403300		X (PCUE)	
	Aire non à l'état prêt	0409100	X	X	
	Aire non sous mode mise à jour	0409200	X	X	
ERASE [nom-article]	L'article est maître d'un ensemble non-vide	0407200	X	X	

5.7.5 FIND

Fonction

Permet de désigner un article de la base de données comme paramètre pour les instructions déclarées par la suite.

Format général

FIND expression-de-sélection-d'article

```
[
  [
    [ RETAINING CURRENCY FOR { MULTIPLE } ]
    [ { REALM } ]
    [ { SETS } ]
    [ { {nom-ensemble}... } ]
    [ { RECORD } ]
  ]
]
```

Règles générales

1. L'instruction FIND rend l'article désigné par expression-de-sélection-d'article article courant de l'unité d'exécution. Le nom du domaine auquel appartient cet article est affiché dans le registre DB-REALM-NAME et le nom de l'article lui-même dans le registre DB-RECORD-NAME.
2. Si la locution RETAINING n'est pas spécifiée, l'article référencé par l'expression devient point courant de son domaine, de son type d'article et de tous les types d'ensemble auxquels il est rattaché.
3. Si la locution RETAINING est accompagnée de l'option REALM, l'indicateur de point courant du domaine n'est pas changé.
4. Si la locution RETAINING est accompagnée de l'option RECORD, l'indicateur de point courant du type d'article n'est pas changé.
5. Si la locution RETAINING est accompagnée du mot SETS, il n'y a pas de modification des indicateurs de point courant du type d'ensemble.
6. Si la locution RETAINING est accompagnée de l'option nom-ensemble, ... les indicateurs de point courant d'ensemble pour tous les types d'ensemble spécifiés ne sont pas modifiés.
7. Si la locution RETAINING est accompagnée de l'option MULTIPLE, les indicateurs de point courant du domaine, du type d'article et du type d'ensemble ne sont pas modifiés.

- Remarques :**
1. L'exécution d'une instruction FIND ne rend pas l'article désigné accessible au programme. Elle identifie simplement l'article pour lui permettre de servir de référence dans les futures instructions. Pour avoir accès au contenu des données de l'article, il faut utiliser une instruction GET.
 2. Si une erreur se produit, les indicateurs de point courant ne sont pas modifiés. Cependant, les registres spéciaux DB-REALM-NAME, DB-RECORD-NAME et DB-SET-NAME sont mis à jour.
 3. Les indicateurs de point courant pour un ensemble MANUAL OPTIONAL ne seront pas mis à jour si l'article trouvé n'est pas couramment rattaché à cet ensemble. S'il l'est, les indicateurs de point courant seront mis à jour.

Format 1

```
FIND [nom-article-1] DB-KEY IS identificateur-1
```

Règles

1. Identificateur-1 doit être un élément DB-KEY.
2. L'article identifié est celui dont la valeur de clé-base-de-données est égale à celle de la donnée fournie par identificateur-1.
3. Nom-article-1 n'est pas obligatoire. S'il est mentionné, l'article trouvé doit correspondre au type indiqué.
4. La valeur de identificateur-1 doit être celle d'une clé de-base-de-données et non celle d'une clé d'aire. Des instructions ACCEPT permettent d'obtenir les valeurs des clés de la base de données.

Format 2

FIND { FIND } nom-article-2
{ DUPLICATE }

Règles

1. Le mode de localisation de nom-article-2 doit être CALC.
2. S'il existe un choix, c'est un domaine qui est choisi dans FIND ANY. Ce format utilise le paramètre d'identification d'aire fourni par le schéma (clause WITHIN du DDL).
3. Avec FIND ANY, c'est la valeur CALC-KEY de nom-article-2 qui est utilisé pour calculer la valeur clé-base-de-données de l'article spécifié. Cette valeur doit être transférée dans les zones CALC-KEY de la mémoire UWA avant émission de la commande FIND.
4. En ce qui concerne FIND DUPLICATE, le SGBD recherche la chaîne CALC qui contient le point courant de l'unité d'exécution, puis il recherche l'article dont la clé CALC est égale à celle de l'article courant de l'unité d'exécution. L'instruction FIND DUPLICATE suppose que l'article courant de l'unité d'exécution est lui-même un des articles en double. La clause DUPLICATES ARE ALLOWD doit figurer dans la spécification de localisation LOCATION MODE CALC du schéma.

Remarque : La valeur de clé CALC utilisée pendant l'exécution FIND DUPLICATE est celle de l'article de la base de données. La valeur représentée dans la zone UWA n'est pas utilisée pendant l'exécution de cette instruction.

Format 3

```
FIND DUPLICATE WITHIN nom-ensemble-1 USING {identificateur-2}
```

Règles

1. Identificateur-2,... doit avoir été défini, dans la rubrique article, pour l'article courant du type d'ensemble désigné par nom-ensemble-1.
2. L'article identifié :
 - a) est un article détail de l'ensemble identifié par l'indicateur de point courant de nom-ensemble-1.
 - b) est du même type d'article que l'article courant du type d'ensemble référencé.
 - c) dispose d'un contenu donnée, identificateur-2,... équivalent à celui de l'article courant du type d'ensemble référencé.
3. Le SGBD recherche le prochain article en double, à partir de l'article courant, en progressant vers l'avant.

- Remarques :**
1. Les valeurs représentées par identificateur-2,... utilisées dans la recherche sont celles des articles de la base de données. Les valeurs de la zone UWA ne sont pas utilisées pendant l'exécution d'une instruction FIND DUPLICATE.
 2. Une erreur est signalée par le message "article non trouvé" (02400), si l'article maître est trouvé avant un article en double. De même, un article en double ne sera pas trouvé s'il se situe entre le premier article et l'article courant de l'ensemble.

Format 4

```

      {NEXT
      {PRIOR
  FIND {FIRST
      {LAST
      {entier-1
      {identificateur-4}
      [nom-article-3] WITHIN {nom-ensemble-2}
                              {nom-domaine-1}
  
```

Règles

1. Entier-1 peut être signé, ce qui autorise la citation relative amont ou aval.
2. La donnée désignée dans identificateur-4 doit être un entier élémentaire signé de valeur non nulle.
3. Si nom-article-3 n'est pas spécifié, la recherche porte sur tous les types d'article du schéma.
4. Si nom-article-3 est spécifié, la recherche ne porte que sur le type d'article exprimé dans nom-article-3. Tous les autres types d'article ne sont pas pris en compte.
5. Si NEXT, accompagné de nom-domaine, est spécifié, l'article trouvé est celui dont la valeur clé-base-de-données est immédiatement supérieure à celle de la clé de l'article courant du domaine cité.
6. Si PRIOR, accompagné de nom-domaine, est spécifié, l'article trouvé est celui dont la valeur clé-base-de-données est immédiatement inférieure à celle de la clé de l'article courant du domaine cité.
7. Si FIRST, accompagné de nom-domaine, est spécifié, l'article trouvé est celui dont la valeur clé-base-de-données est la plus petite par rapport à tous les articles enregistrés dans le domaine cité.
8. Si LAST, accompagné de nom-domaine, est spécifié, l'article identifié est celui dont la valeur clé-base-de-données est la plus élevée par rapport à tous les articles du domaine cité.
9. Si entier-1 ou identificateur-4, accompagné de nom-domaine, est spécifié, l'article trouvé est celui dont la position ordinale dans le domaine est égale à entier-1, ou à identificateur-4, suivant le cas. Si la valeur spécifiée est positive, la position ordinale est relative à l'article dont la valeur clé-base-de-données est la plus petite dans le domaine cité. Si cette valeur est négative, la position ordinale est relative à l'article dont la valeur clé-base-de-données est la plus élevée dans le domaine cité.
10. Si NEXT, accompagné de nom-ensemble, est spécifié, c'est l'article suivant l'article courant de l'ensemble qui est trouvé, en fonction de l'ordre de classement propre à l'ensemble.
11. Si PRIOR, accompagné de nom-ensemble, est spécifié, c'est l'article précédent l'article courant de l'ensemble qui est trouvé, suivant l'ordre de classement propre à l'ensemble.
12. Si FIRST, accompagné de nom-ensemble, est spécifié, c'est le premier article détail de l'ensemble dont l'article courant est lui-même article détail qui est trouvé, suivant l'ordre de classement propre à l'ensemble.

13. Si LAST, accompagné de nom-ensemble, est spécifié, c'est le dernier article détail de l'ensemble dont l'article courant est lui-même article détail qui est trouvé, en fonction de l'ordre de classement propre à l'ensemble.
14. Si entier-1 ou identificateur-4, accompagné de nom-ensemble, est spécifié, l'article trouvé est celui dont la position ordinale dans l'ensemble est exprimée par entier-1 ou identificateur-4, suivant le cas. Si la valeur spécifiée est positive, la position ordinale est calculée par rapport au premier article détail de l'ensemble, suivant l'ordre de classement propre à l'ensemble. Si cette valeur est négative, la position ordinale est calculée par rapport au dernier article détail de l'ensemble, suivant l'ordre de classement propre à l'ensemble.

Format 5

```
FIND CURRENT [nom-article-4] [ { WITHIN nom-domaine-2 } ]  
[ { nom-ensemble-3 } ]
```

Règles

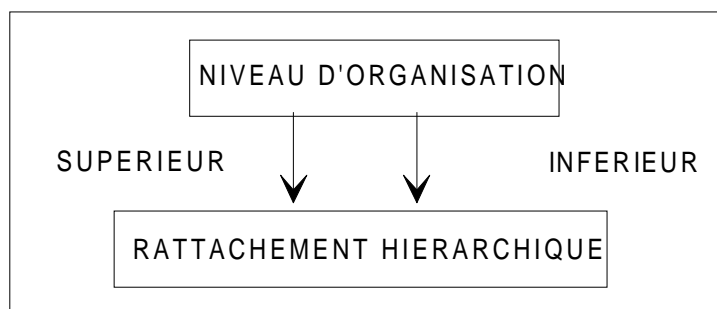
1. Si nom-ensemble-3 est spécifié, l'article trouvé est l'article courant du type d'ensemble indiqué.
2. Si nom-ensemble-2 est spécifié, l'article trouvé est l'article courant du domaine indiqué.
3. Si nom-article-4 et nom-ensemble-3 sont spécifiés, nom-article-4 doit correspondre à l'article courant du type d'ensemble indiqué.
4. Si nom-article-4 et nom-domaine-3 sont spécifiés, nom-article-4 doit correspondre à l'article courant du domaine indiqué.
5. Si nom-article-4 est spécifié et que la clause WITHIN est omise, l'article trouvé est l'article courant du type d'article indiqué.
6. Si, ni nom-domaine-2, ni nom-ensemble-3, ni nom-article-4 ne sont spécifiés, l'article trouvé par l'instruction est l'article courant de l'unité d'exécution.

Format 6

```
FIND OWNER WITHIN nom-ensemble-4
```

Règles

1. L'article est le maître de l'ensemble identifié par l'indicateur de point courant de nom-ensemble-4.
2. Dans certaines structures de réseau complexe il est conseillé d'utiliser la clause facultative `RETAINING CURRENCY FOR...` de manière à s'assurer que les autres indicateurs de point courant ne sont pas modifiés par l'exécution de `FIND`. Lorsque l'article maître est trouvé, il devient l'article courant de tous les types d'ensembles auxquels il est rattaché. Par exemple, dans une structure de réseau d'application d'organigramme :



Lorsque le programme parcourt l'ensemble SUPERIEUR pour localiser tous les autres NIVEAUX D'ORGANISATION qui sont subordonnés à l'article NIVEAU D'ORGANISATION en cours, l'instruction `FIND OWNER WITHIN INFERIEUR` est émise chaque fois qu'un article RATTACHEMENT HIERARCHIQUE est trouvé. Le nouvel article NIVEAU D'ORGANISATION repositionnera l'indicateur de point courant du type d'ensemble sur l'ensemble SUPERIEUR, mettant fin ainsi à la recherche initiale. Si, au contraire, l'instruction est écrite: `FIND OWNER WITHIN INFERIEUR RETAINING CURRENCY FOR SETS.`, lorsque le nouvel article maître NIVEAU D'ORGANISATION est trouvé, le point courant de SUPERIEUR n'est pas modifié, permettant ainsi de trouver l'article RATTACHEMENT HIERARCHIQUE suivant.

Format 7

```
FIND nom-article-5 WITHIN nom-ensemble-5 [CURRENT]  
  
[USING {identificateur-5}...]
```

Règles

1. Identificateur-5,... doit être défini dans la rubrique article de nom-article-5.
2. Nom-article-5 doit être défini dans le DDL de schéma comme article détail (MEMBER) de nom-ensemble-5.
3. L'article trouvé est du même type que celui indiqué par nom-article-5.
4. Si la locution USING est spécifiée, l'article trouvé contient les données de identificateur-5,... qui sont équivalentes à celles de la zone UWA. Si USING n'est pas spécifié, l'article identifié est le premier article de l'ensemble.
5. Le SGBD entreprend sa recherche de la manière suivante :
 - a) Si CURRENT n'est pas spécifié, la sélection du type d'ensemble adéquat se fait en fonction des critères de sélection d'ensemble spécifié dans le schéma.
 1. Le point d'entrée est sélectionné selon l'une des trois méthodes suivantes : par APPLICATION (indicateur de point courant de type d'ensemble); par CALC-KEY (randomisation d'une zone de contrôle spécifiée); ou par DATABASE-KEY (clé-base-de-données) fourni par programme.
 2. La recherche s'effectue dans chaque ensemble à l'aide des clés définies dans le DDL de schéma. Chacun de ces ensembles correspond à une clause THEN THRU dans la rubrique ensemble du DDL de schéma.
 3. Le dernier ensemble recherché identifie le type d'ensemble dans lequel l'article trouvé est article détail. S'il n'existe pas de clause THEN THRU dans le DDL de schéma, l'article servant de point d'entrée permet d'identifier l'ensemble dans lequel l'article trouvé est article détail.
 - b) Si CURRENT est spécifié, l'ensemble est identifié par le point courant du type d'ensemble.
 - c) La recherche à l'intérieur de l'ensemble, une fois identifié comme indiqué ci-dessus, se poursuit en aval pour, USING étant spécifié, localiser l'article qui contient identificateur-5..., lui-même correspondant aux données de la zone UWA. Si USING n'est pas spécifié, c'est le premier article qui est sélectionné. Seuls les articles qui correspondent à nom-article-5 sont pris en compte dans cette recherche. Les autres types d'article sont ignorés.

- Remarques :**
1. Si l'article de point d'entrée tel qu'il est décrit dans la règle 5a.(1) est identifié par CALC-KEY et qu'il existe une éventualité que l'article soit rangé dans plus d'un domaine, la sélection de domaine est appelée. Elle utilise pour cela le paramètre d'identification d'aire fourni par le schéma. La clé CALC de cet article doit être préalablement initialisée dans la zone UWA avant d'exécuter la fonction FIND.
 2. Si l'article de point d'entrée tel qu'il est décrit dans la règle 5a.(1) est identifié par APPLICATION, l'article courant du type d'ensemble spécifié dans le DDL de schéma doit être déterminé avant l'exécution de la fonction FIND. Ce type d'ensemble est l'un de ceux qui ont été référencés par la locution THRU de la clause SELECTION de la sous-rubrique NUMBER dans le DDL de schéma. L'article identifié peut être maître ou détail du type d'ensemble.
 3. Si l'article de point d'entrée tel qu'il est décrit dans la règle 5a.(1), est identifié par DATA-BASE-KEY, le paramètre-base-de-données spécifié dans le DDL de schéma doit être mis à sa valeur correcte avant toute exécution de la fonction FIND. L'article identifié doit être maître de l'ensemble.
 4. Les clés de sélection d'ensemble, s'il en existe, qui sont mentionnées dans la règle 5a.(2) doivent être mises à leur valeur correcte dans la zone UWA avant toute exécution de la fonction FIND
 5. Si aucun ensemble n'est trouvé qui satisfait les critères de sélection d'ensemble décrit dans la règle 5a., un code de retour (023000) "Sélection d'ensemble non réussie" est renvoyé pour signaler une erreur. Cette circonstance peut se produire dans n'importe quel point de la règle 5a.
 6. Si l'ensemble est trouvé mais qu'aucun article ne peut être identifié pour satisfaire la condition de sélection d'article de la règle 5b., un code de retour (024000) "Sélection d'article non réussie" est renvoyé pour signaler l'erreur.
 7. La recherche dans l'ensemble de la règle 5b. se poursuivra sur la totalité de l'ensemble, en commençant par le premier article de ce type dans l'ensemble.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
Générale	Aucune; article rattaché	0000000	X	X	-
	Non-ouverture de la BD	0509100			
	Aire non à l'état êt	0509100	X	X	
FIND DB-KEY IS	Clé-base-de-données erronée ou n'entrant pas dans la plage type-article	0504100			
	Clé-BD correcte mais article introuvable ou article spécifié incorrect	0502400	X	(X)	
FIND ANY... (CALC)	Nom-aire de AREA-ID inconnu, ou pas concordant avec type-article	0504300			
	Données décimales incorrectes dans la zone UWA (CALC)	0505200		X	
	Pas d'article correspondant à cette clé CALC	0502400	X	X	
	Absence de description d'article UWA	0573660		X	
FIND DUPLICATE... (CALC)	Indicateur P.C. d'UE à NUL	0503200			
	P.C. d'UE non conforme	0503300		X PCUE	
	Absence d'article en double	0502400	X	(X)	
FIND relatif dans l'aire	Indicateur P.C. d'aire à NUL (NEXT, PRIOR)	0503100	X		
	Ordinal = 0	0573630			
	Article introuvable (fin de domaine)	0502100	X	(X)	
Find relatif dans l'ensemble	Indicateur P.C. de type ensemble à NUL	0503100			X
	Ordinal = 0	0573630			
	Article introuvable (fin d'ensemble)	0502100		(X)	X
FIND OWNER dans l'ensemble	Indicateur P.C. de type ensemble à NUL	0503100			X
FIND CURRENT	Indicateur P.C. d'UE à NUL	0503200			
FIND CURRENT nom-article	Indicateur P.C. de type-article à NUL	0503100		X	
FIND CURRENT [nom-article] WITHIN aire	Indicateur P.C. d'aire à NUL ou VIRTUEL	0503100		X	
	Point-courant d'aire non valable si spécifié	0503300		X	
FIND CURRENT [nom-article] WITHIN ensemble	Indicateur P.C. de type-ensemble à NUL ou VIRTUEL	0503100			X
	Point courant d'ensemble incorrect (spécifié)	0503300		X	

Langage DML de manipulation des données

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
FIND nom-article WITHIN ensemble [CURRENT] [USING..]	Indicateur PC de type ensemble à NUL	0503100			X
	Sélection d'ensemble non réussie	0502300		X	X
	Données décimales interdites en zone UWA	0505200		X	
	Article non trouvé dans l'ensemble sélectionné	0502400		X	X
	Absence de description d'article UWA	0573660		X	
FIND DUPLICATE WITHIN ensemble USING... FIND CURRENT nom_article	Indicateur P.C. de type ensemble selectione à NUL ou VIRTUEL	0503100			X
	P.C. de type-ensemble est article maître ou article incorrect	0503300		X	
	Absence d'article en double	0502400		X	X

5.7.6 FINISH

Fonction

Permet de mettre fin à l'accessibilité d'un ou de plusieurs domaines par le programme.

Format général

FINISH [nom-domaine-1] ...

Règles générales

1. L'exécution de la fonction FINISH met fin à la disponibilité d'un ou de plusieurs domaines.
2. Si nom-domaine-1,... est spécifié, le ou les domaine(s) indiqué(s) n'est (ne sont) plus à l'état prêt. Nom-domaine-1 ne peut apparaître qu'une seule fois dans une instruction FINISH déterminée.
3. Si nom-ensemble-1,... n'est pas spécifié, tous les domaines cessent d'être à l'état prêt.
4. Lorsque la fonction FINISH est exécutée, tous les indicateurs de point courant des articles rangés dans les domaines concernés affichent NUL.
5. Lorsqu'une instruction FINISH est lancée, tous les domaines sur lesquels elle porte doivent être à l'état prêt.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAU X IDS		
		DB-STATUS	REALM	RECORD	SET
FINISH	Aucune	0000000	-	-	-
	Non-ouverture de la BD	0609100			
	Aire non à l'état prêt	0609100		X	

5.7.7 GET

Fonction

Permet de mettre à la disposition du programme certaines ou toutes les zones d'un article de la base de données. Le paramètre de l'instruction est l'article courant de l'unité d'exécution.

Format général

Format 1

```
ET [nom-article]
```

Format 2

```
GET identificateur-1 ...
```

Règles de syntaxe

1. Dans le format 1, nom-article doit être le nom de l'article courant de l'unité d'exécution.
2. Dans format 2, identificateur-1,... doit citer les données élémentaires définies dans la rubrique article pour l'article courant de l'unité d'exécution.

Règles générales

1. L'exécution d'une fonction GET transfère tout ou partie de l'article courant de l'unité d'exécution dans l'article UWA correspondant.
2. Dans format 1, c'est tout le contenu de l'article courant qui est transféré dans la zone UWA.
3. Dans format 2, seules les données dont la liste est indiquée sont transférées dans l'article UWA correspondant à l'article courant de l'unité d'exécution. Toutes les autres zones de l'article UWA ne sont pas modifiées.
4. L'exécution de la fonction GET ne modifie pas les indicateurs de point courant.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX IDS		
		DB-STATUS	REALM	RECORD	SET
GET	Aucune	0000000	-	-	-
	Non-ouverture de la BD	0809100			
	Indicateur P.C. d'UE à NUL	0803200			
	P.C. d'UE incorrect (si spécifié)	0803300		X PCUE	
	Absence de description article UWA courant	0873660		X	

5.7.8 Instruction IF de test de condition

Fonction

Permet au programme d'orienter le traitement en fonction de la valeur d'un test portant sur un article courant et/ou ses relations avec les articles détail de l'ensemble. Il existe deux conditions de base de données.

1. Appartenance d'un article à un ou plusieurs types d'ensemble.
2. Présence d'articles détails dans un ensemble: état d'un type d'ensemble.

Remarque : Le terme "condition-base-de-données" sera exprimé, dans le reste du présent manuel et dans les options de statistiques, de jalonnement, et d'exécution, sous le vocable "fonction IF DML".

Format général

Appartenance

```
IF [nom-ensemble] {OWNER }
                   {MEMBER }
                   {TENANT }
```

Présence d'article détail

```
IF [nom-ensemble] IS [NOT] EMPTY
```

Règles de syntaxe

La condition de base de données est, du point de vue COBOL, une condition simple. Comme telle, elle peut apparaître partout où l'usage d'une condition simple est autorisée.

Une condition de base de données a une valeur "vraie" ou "fausse" (condition vérifiée ou non). Une condition ne peut faire l'objet d'une vérification lorsqu'une erreur apparaît dans la base de données.

Règles générales concernant la condition d'appartenance

1. La condition d'appartenance indique si l'article courant de l'unité d'exécution est actuellement :
 - L'article maître d'un type d'ensemble spécifié (ou au moins d'un type d'ensemble quelconque).
 - Ou un article détail réel d'un type d'ensemble spécifié (ou au moins d'un type d'ensemble quelconque).
 - Ou les deux à la fois.
2. Nom-ensemble OWNER est vérifié lorsque l'article qui correspond au point courant d'unité d'exécution est l'article maître d'un type d'ensemble désigné par "nom-ensemble". Dans le schéma, le type d'article courant d'unité d'exécution doit avoir été déclaré comme le type d'article maître de ce type d'ensemble. Dans le cas contraire, le SGBD signale une condition d'erreur de base de données.
3. OWNER est vérifié lorsque le point courant d'unité d'exécution est l'article maître d'un ensemble non-vidé d'au moins un type d'ensemble. Le type d'article courant d'unité d'exécution doit figurer dans le schéma comme type d'article maître d'au moins l'un des types d'ensemble. Dans le cas contraire, le SGBD signale une condition d'erreur de base de données.
4. nom-ensemble MEMBER est vérifié lorsque le point courant d'unité d'exécution correspond réellement à un article détail d'un ensemble du type désigné par "nom-ensemble". Le type d'article courant d'unité d'exécution doit avoir été déclaré dans le schéma comme article détail avec MANUAL OPTIONAL pour ce type d'ensemble. Dans le cas contraire, le SGBD signale une condition d'erreur de base de données.
5. MEMBER est vérifié lorsque le point courant d'unité d'exécution correspond effectivement à un article détail d'un ensemble d'au moins un type d'ensemble. Les types d'ensembles pris en compte sont ceux pour lesquels le type d'article courant de l'unité d'exécution a été déclaré dans le schéma comme article détail avec MANUAL OPTIONAL. Dans le cas contraire, le SGBD signale une condition d'erreur de base de données.
6. nom-ensemble TENANT revient aux conditions suivantes :
 - "nom-ensemble OWNER" lorsque le type d'article courant d'unité d'exécution a été déclaré dans le schéma comme type d'article maître pour ce type d'ensemble.
 - "non-ensemble MEMBER" lorsque le type d'article courant d'unité d'exécution a été déclaré dans le schéma comme type d'article détail avec MANUAL OPTIONAL pour ce type d'ensemble. Le SGBD signale une condition d'erreur de base de données si le type d'article courant d'unité d'exécution n'a pas été déclaré comme type d'article maître ou comme type d'article détail avec MANUAL OPTIONAL pour ce type d'ensemble.
7. TENANT est l'union (OU inclusif) des conditions IF OWNER et IF MEMBER. Le schéma doit prévoir des types d'ensembles pour lesquels le type d'article correspondant au point courant d'unité d'exécution est indifféremment article maître, ou article détail défini avec MANUAL OPTIONAL. Dans le cas contraire, le SGBD signale une condition d'erreur de base de données.

Remarque : On ne peut pas se servir des conditions d'appartenance pour connaître la structure des données du type décrit dans le schéma. Cette condition ne porte que sur les "apparitions" des éléments de la structure des données. Par exemple, la fonction de IF OWNER n'est pas de tester la présence du type d'article du point courant d'unité d'exécution dans le schéma en tant que type d'article maître de certains types d'ensembles; sa fonction est de vérifier si le point courant d'unité d'exécution est effectivement l'article maître d'au moins un ensemble non-vide.

Règles générales concernant la condition "présence d'article détail"

1. Cette condition s'utilise pour demander au programme de vérifier si oui ou non l'ensemble du type spécifié, identifié par le point courant du type d'ensemble, ne contient pas d'article détail.
2. L'indicateur de point courant de type d'ensemble identifiant l'ensemble, ne doit être ni à NUL, ni à VIRTUEL à la suite d'une instruction ERASE. Si tel est le cas, le SGBD signale une condition d'erreur de base de données.
3. nom-ensemble IS EMPTY est vérifié lorsque l'ensemble ne contient pas d'article détail.
4. Avec la mention NOT, le résultat du test est inverse.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX IDS		
		DB-STATUS	REALM	RECORD	SET
	Aucune; condition vérifiée	0000000	-	-	-
	Aucune; condition fausse	0000000	-	-	-
	Non-ouverture de la BD	0909100			
Condition d'appartenance	Indicateur P.C. d'UE à NUL	0903300			
	P.C. non conforme à la condition à vérifier (n'est ni maître, ni détail OPTIONAL des ensembles)	0903200		X	
Condition de présence détail	Indicateur P.C. de type-ensemble à NUL	0903100			X

5.7.9 MODIFY

Fonction

Permet de modifier le contenu d'une ou des données d'un article et/ou de changer sa relation d'appartenance à l'ensemble. L'article paramètre de l'instruction est l'article courant de l'unité d'exécution.

Format général

Format 1

```

MODIFY { [nom-article] }
        { {identificateur} ... } [RETAINING-spécification]

```

Format 2

```

MODIFY [nom-article] ONLY { ALL } MEMBERSHIP
                          { {nom-ensemble} ... }
                          [RETAINING-spécification]

```

Format 3

```

MODIFY { [nom-article] } INCLUDING { ALL } MEMBERSHIP
        { {identificateur} ... }        { {nom-ensemble} ... }
                                         [RETAINING-spécification]

```

Format de la locution RETAINING

```

RETAINING CURRENCY FOR { MULTIPLE }
                       { REALM }
                       { RECORD }
                       { { SETS } }
                       { {nom-ensemble} ... }

```

Règles de syntaxe

1. Si le nom d'article est spécifié il doit être le même que celui de l'article courant de l'unité d'exécution.
2. Identificateur-1,... doit citer des données qui existent dans l'article courant de l'unité d'exécution.
3. L'article courant de l'unité d'exécution doit avoir été défini dans le DDL de schéma comme article détail des types d'ensembles, s'ils existent, désignés par nom-ensemble,...

Règles générales

Règles communes à tous les formats :

1. Si RETAINING n'est pas spécifié, l'article paramètre devient, après exécution réussie de l'instruction, l'article courant de son domaine, de son type d'article et de tous les types d'ensembles auxquels il appartient.
2. Si RETAINING est spécifié, avec option REALM, l'indicateur de point courant de domaine n'est pas modifié.
3. Si RETAINING est spécifié, avec option RECORD l'indicateur de point courant du type d'article n'est pas modifié.
4. Si RETAINING est spécifié, avec option SETS aucun indicateur de point-courant de type d'ensemble n'est modifié.
5. Si RETAINING est suivie de nom-ensemble,... les indicateurs de point courant des types d'ensemble désignés ne sont pas modifiés.

Format 1

1. Ce format permet de changer le contenu d'une ou des données de l'article courant de l'unité d'exécution.
2. Si nom-article est omis, le SGBD remplace les données de l'article courant de l'unité d'exécution par celles de l'article correspondant de la zone UWA.
3. Si nom-article est spécifié, le SGBD vérifie préalablement que l'article courant de l'unité d'exécution est du type d'article désigné et exécute ensuite la règle 2.
4. Si identificateur-1,... est spécifié, le SGBD remplace les données spécifiées par celles de l'article correspondant de la zone UWA.
5. Si les données nouvellement introduites ne satisfont pas les contrôles de validité indiqués dans le DDL de schéma pour ce type d'article, la modification est refusée et il se produit une erreur de base de données.
6. Si les éléments CALC-KEY sont modifiés et que la clause MIGRATION IS ALLOWED n'est pas spécifiée dans le DMCL pour ce type d'article, l'article conserve sa clé-base-de-données et est inséré logiquement dans la chaîne CALC correspondant à la nouvelle valeur CALC-KEY.

Si un article en débordement ou article OWL, correspondant à la nouvelle chaîne CALC doit être créé dans la page de l'article et que cette page ne dispose pas d'un espace suffisant, la modification est refusée et il se produit une erreur de base de données.

7. Si les éléments CALC-KEY sont modifiés et que la clause MIGRATION IS ALLOWED est spécifiée, l'article est transféré dans le compartiment correspondant à la nouvelle valeur CALC-KEY. L'article ne peut modifier les aires à la suite de sa migration.

Les pointeurs d'ensemble et les indicateurs de point courant se rapportant à l'article transféré sont mis à jour pour indiquer la nouvelle position.

Langage DML de manipulation des données

8. Si les données modifiées sont des clés de tri d'un ensemble (y compris les éléments DATA-BASE-KEY en cas de migration) l'ensemble est réorganisé en fonction des critères de classement qui ont été spécifiés pour cet ensemble.
9. Si les données modifiées ne satisfont pas au contrôle de non duplication (CALC-KEY, clé de tri, zone de contrôle de non duplication), la modification est refusée et il se produit une erreur de base de données.

Format 2

1. Format 2 permet de changer la relation à l'ensemble de l'article courant de l'unité d'exécution pour les types d'ensemble spécifiés.
2. Si nom-article est spécifié, le SGBD vérifie que l'article courant de l'unité d'exécution appartient bien à ce type d'article.
3. Si All est spécifié, les types d'ensembles concernés sont ceux dans lesquels l'article est détail, avec AUTOMATIC MANDATORY ou avec MANUAL OPTIONAL et couramment rattaché à ces ensembles.
4. Si nom-ensemble,... est spécifié, les types d'ensembles concernés sont ceux de la liste. Si l'article a été défini comme détail MANUAL OPTIONAL pour ces types d'ensembles, il doit avoir fait l'objet d'un rattachement à cet ensemble.
5. Pour chaque type d'ensemble concerné, la relation à l'ensemble est modifiée de la façon suivante :
 - a) L'article courant de l'unité d'exécution est éliminé du type d'ensemble auquel il appartient.
 - b) Un nouveau type d'ensemble est sélectionné en fonction des critères de sélection d'ensemble spécifiés dans le DDL de schéma.
 - c) L'article courant de l'unité d'exécution est rattaché au type d'ensemble préalablement sélectionné. L'insertion se fait selon les critères de classement spécifiés dans le DDL de schéma.
6. Si le mode de localisation de l'article courant de l'unité d'exécution est "LOCATION IS VIA nom-ensemble WITHIN AREA OF OWNER", toute modification de la relation de cet article à l'ensemble sera refusée si le nouvel article maître ne se trouve pas dans la même aire que l'article courant de l'unité d'exécution.
7. Si les données de l'article courant de l'unité d'exécution ne satisfont pas au contrôle de non-duplication (clés de tri, zone de contrôle de non duplication) dans les nouveaux types d'ensembles, la modification est refusée et il se produit une erreur base de données.

Format 3

1. Format 3 permet de remplacer le contenu d'une ou des données de l'article courant de l'unité d'exécution et de modifier la relation à l'ensemble avec un ou plusieurs ensembles.
2. Les interventions du SGBD constituent une combinaison de celles qui ont été décrites pour les formats 1 et 2.

- Remarques :**
1. La relation logique MANUAL OPTIONAL d'un article à un ensemble peut être changée par instruction MODIFY. Si l'article courant de l'unité d'exécution est déjà rattaché à l'ensemble, l'article est traité comme AUTOMATIC MANDATORY. Aucune possibilité n'est offerte d'obtenir les effets d'une instruction CONNECT avec une instruction MODIFY. Si l'article est déjà rattaché à un ensemble avant MODIFY, l'effet de cette instruction est de le rattacher à un autre ensemble. Si l'article est déjà éliminé d'un ensemble avant MODIFY, l'effet de cette instruction est de l'éliminer d'un autre ensemble.
 2. Si une des zones de contrôle de l'article doit être changée, cette zone sera extraite de la zone UWA. Les autres zones de contrôle, nécessaires pour exécuter la fonction, seront prises dans l'article courant de l'unité d'exécution de la base de données. Ceci signifie que toutes les combinaisons de données sont possibles dans un article.
 3. a) Si une erreur se produit dans la base de données au moment d'une exécution de MODIFY sans qu'il y ait migration d'article, la base de données est remise à l'état dans lequel elle se trouvait avant l'exécution de l'instruction, sans utilisation des journaux. Les indicateurs de point courant ne sont pas modifiés et le contrôle revient au programme appelant avec affichage de la valeur appropriée dans le registre DB-STATUS.

b) Si une erreur se produit dans la base de données pendant l'exécution d'une instruction MODIFY avec migration d'article, mais avant que la mise à jour de la base de données ait réellement commencé, le SGBD procède de la même façon que ci-dessus.

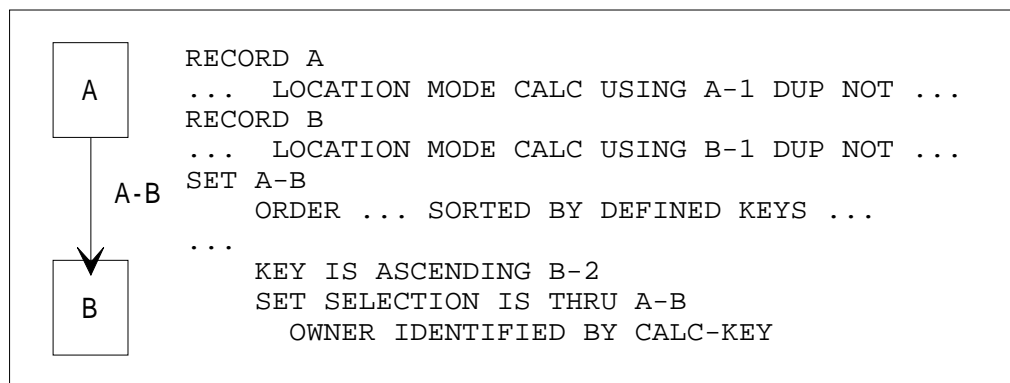
c) Si une erreur se produit dans la base de données pendant l'exécution d'une instruction MODIFY avec migration d'article et après que la mise à jour de la base de données ait commencé, la suite du traitement dépend du niveau de protection spécifié pour la base de données:

- En traitement par lots, ou dans un cadre d'exploitation IOF ou TDS, si la base de données est protégée par journal AVANT, elle est restaurée à l'état où elle se trouvait avant exécution de l'instruction et le contrôle revient au programme appelant avec affichage de la valeur appropriée dans le registre DB-STATUS.

- En traitement par lots, ou dans un cadre d'exploitation IOF, si la base de données n'est pas protégée par journal AVANT, elle est laissée dans un état instable de l'activité est arrêtée prématurément. Il faut faire appel à l'utilitaire FILREST ou VOLREST pour restaurer la base de données.

- Si la base de données est sous la protection du journal APRES et des mécanismes de mise à jour différée de TDS, elle est restaurée à l'état dans lequel elle se trouvait au moment du dernier point de consolidation. La routine TPR en cours est arrêtée prématurément et n'est pas relancée.

Exemples :



1. Modification de clé CALC dans l'article A.

```

MOVE clé-1 TO A-1.
FIND ANY A.
MOVE nouvelle-donnée-identificateur TO A-1.
MODIFY A-1.
  
```

2. Modification de clé de tri dans l'article B sans changement des types d'article.

```

MOVE clé-2 TO B-1.
FIND ANY B.
MOVE nouvelle-clé-de-tri TO B-2.
MODIFY B-2.
  
```

3. Insérer article B sous article A sans modification de zone de B.

```

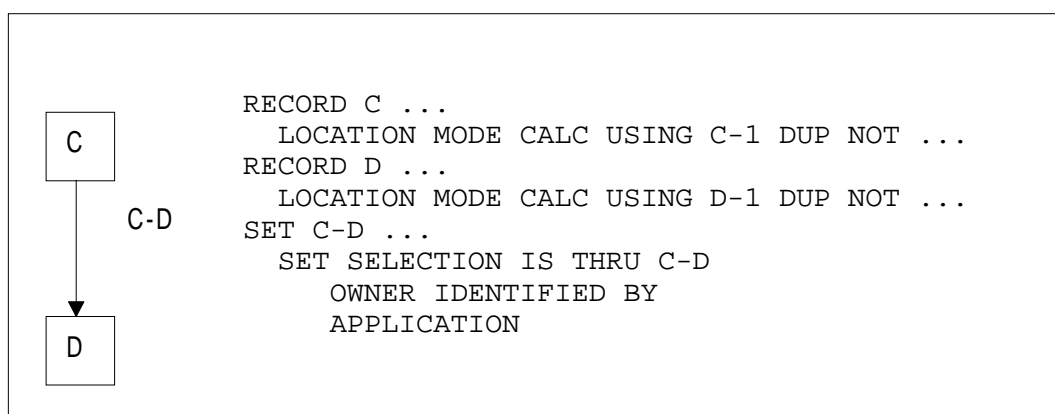
MOVE clé-1 TO B-1.
FIND ANY B.
MOVE clé-nouveau-maître TO A-1 (itinéraire de
                                sélection pour
                                nouveau type
                                d'article).
MODIFY ONLY A-B.
  
```

4. Insérer article B sous article A et modifier clé-de-tri de B.

```

MOVE clé-1 TO B-1.
FIND ANY B.
MOVE clé-nouveau-maître (itinéraire de sélection
                          pour nouveau type
                          d'article).
MOVE nouvelle-clé-de-tri TO B-2.
MODIFY B-2 INCLUDING A-B.
  
```

Langage DML de manipulation des données



5. Déplacer article D sous nouvel article C.

MOVE clé-nouveau-maître TO C-1. (Itinéraire de sélection pour nouveau type ensemble).

FIND ANY C.
 MOVE clé-1 TO D-1.
 FIND ANY D RETAINING CURRENCY C-D.
 MODIFY D ONLY C-D.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
Générale	Aucune; modification exécutée	0000000	-	-	-
	Non-ouverture de la BD	1109100			
	Indicateur P.C. d'UE à NUL	1103200			
	P.C. d'UE incorrect (si spécifié) ou non détail de l'ensemble spécifié	1103300		X PCUE	
	Aire non à l'état prêt (READY)	1109100	X	X	(X)
	Aire non en mode mise à jour	1109200	X	X	(X)
	Données en double interdites (clé-CALC ou clé-tri ou autre zone de contrôle)	1105100	(X)	X	(X)
Modification de données	Absence de description d'article UWA courant	1173660		X	
	Données ne satisfaisant pas les contrôles de validité	1105200		X	
	Modification de clé-CALC interdite sans migration	1173640		X	
	Espace d'aire épuisé (migration)	1180200	X	X	
Modification de relation	Tentative erronée de modifier relation d'un détail OPTIONAL non rattaché	1108300		X	X
	Sélection d'ensemble infructueuse	1102300		X	X
	Violation par le nouveau maître du type-ensemble de la condition AREA OF OWNER	1173650	(X) (maître)	X	X

5.7.10 READY

Fonction

Permet de mettre un ou plusieurs domaines à l'état prêt pour le traitement.

Format général

```

READY {[nom-domaine-1] ... USAGE-MODE IS
      { EXCLUSIVE      { RETRIEVAL } }
      { SHARED        { UPDATE   } }
      { [MONITORED]   { RETRIEVAL } } ...
      {                { RETRIEVAL } }
      {                { UPDATE   } }
    
```

Règles générales

1. Quel que soit son format, l'exécution d'une instruction READY porte sur la totalité du domaine.
2. Au moment de l'exécution d'une instruction READY, aucun domaine concerné par cette instruction ne doit se trouver à l'état prêt.
3. Si nom-domaine-1,... est spécifié, chacun des domaines cités est rendu disponible aux accès et mis à l'état prêt. Nom-domaine-1 ne peut figurer qu'une seule fois dans une instruction READY particulière. Si nom-domaine-1,... n'est pas spécifié, tous les domaines de la base de données sont rendus accessibles et placés à l'état prêt.
5. La locution USAGE-MODE établit le mode d'utilisation des domaines concernés et indique donc le type d'opération autorisé sur ces domaines. Le mode d'utilisation reste valable pour les domaines concernés jusqu'à exécution d'une instruction FINISH, ou jusqu'à ce que l'unité d'exécution de l'instruction READY prenne fin.
6. La locution RETRIEVAL autorise l'accès aux domaines concernés aux seules fins de les inspecter et d'en lire le contenu.
7. La locution UPDATE autorise aussi bien l'accès que la modification du contenu des domaines concernés.
8. La locution EXCLUSIVE indique que les domaines désignés ne peuvent faire l'objet d'accès simultanés de la part d'unités d'exécution concurrentes.
9. SHARED spécifie que les domaines désignés peuvent faire l'objet d'une lecture de la part d'unités d'exécution concurrentes sous réserve de le faire sous le même mode d'utilisation (USAGE-MODE identique). Le partage au niveau fichier se fait en mode RETRIEVAL seulement.
10. La locution MONITORED indique que les domaines désignés peuvent faire l'objet de lecture ou de modification pour des unités d'exécution concurrentes sous réserve d'utiliser le mode géré (MONITORED, c'est-à-dire sous GAC). Le partage s'effectue au niveau page.

Langage DML de manipulation des données

11. Les instructions JCL ASSIGN/DEFINE doivent être cohérentes avec le mode d'utilisation défini par le programme. Des règles de substitution permettent à l'utilisateur de spécifier dans le JCL un mode de partage plus restrictif que celui qui a été spécifié dans le programme. Pour plus de détails se reporter au chapitre 7.
12. Lorsque USAGE MODE IS EXCLUSIVE UPDATE est spécifié et qu'il n'existe pas de protection par journal AVANT le domaine est mis à l'état transitoire. Il le demeure jusqu'à l'exécution réussie d'une instruction FINISH.

Le domaine reste à l'état transitoire si l'activité s'arrête prématurément avant la fin d'exécution de l'instruction FINISH. Sauf si l'utilisateur demande d'ignorer cet état (voir les commandes émises au moment de l'exécution au chapitre 7), l'activité s'arrêtera prématurément chaque fois que le domaine fera l'objet d'une instruction READY.

Le SGBD enregistre dans les labels IDS la date et l'heure à laquelle un utilisateur a, pour la dernière fois, décidé de ne pas prendre en compte l'état transitoire d'un domaine.

L'intervention normale, après un arrêt prématuré d'activité ayant entraîné la mise à l'état transitoire d'un domaine, est de restaurer ce domaine à son état initial au moyen de FILREST.

Le drapeau indicateur d'état transitoire peut lui-aussi être remis à zéro au moyen de la commande PATCH LABEL de DBUTILITY.

13. Les instructions READY écrites dans une routine TPR ne sont pas prises en compte au moment de l'exécution.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
	Aucune	0000000	-	-	-
READY	Aire déjà mise à l'état prêt	1309300	X		

5.7.11 STORE

Fonction

Permet de ranger un article dans la base de données. Cette instruction détermine l'article courant de l'unité d'exécution.

Format général

```

STORE nom-article [ RETAINING CURRENCY FOR { MULTIPLE } ]
[ { REALM } ]
[ { SETS } ]
[ { {nom-ensemble} ... } ]
[ { RECORD } ]
    
```

Règles générales

1. Les zones de l'article UWA référencées par nom-article sont transférées dans l'article de la base de données. Les contrôles de validité définis dans le schéma sont exécutés.
2. L'article référencé par nom-article devient détail de chaque type d'ensemble pour lequel il a été défini comme AUTOMATIC. Les ensembles auxquels l'article est rattaché sont choisis selon les critères de sélection d'ensemble. La manière par laquelle l'article est rattaché se fait selon les critères de classement propres à chaque type d'ensemble.
3. L'article référencé devient maître d'ensemble vide pour chaque type d'ensemble dans lequel il est maître.
4. L'instruction STORE établit l'article référencé comme article courant de l'unité d'exécution. Les nom-domaine et nom-article de l'article sont rangés dans les registres spéciaux, respectivement DB-REALM-NAME et DB-RECORD-NAME.
5. Si RETAINING n'est pas spécifié, l'article référencé devient article courant de son domaine, de son type d'article et de tous les types d'ensembles pour lesquels il a été déclaré maître ou détail de type AUTOMATIC.
6. Si RETAINING est spécifié avec REALM, l'indicateur de point courant de domaine n'est pas modifié.
7. Si RETAINING est spécifié avec RECORD, l'indicateur de point courant de type d'article n'est pas modifié.
8. Si RETAINING est spécifié avec SETS, aucun indicateur de point courant de type d'ensemble n'est modifié.
9. Si RETAINING, suivie de non-ensemble,... est spécifié, les indicateurs de point-courant du (ou des) nom-ensemble,... ne sont pas modifiés.
10. Si RETAINING est spécifié avec MULTIPLE, les indicateurs de point courant de domaine, de type d'article et de type d'ensemble ne sont pas modifiés.

11. Le domaine dans lequel l'article est rangé doit être ouvert en mode mise à jour (UPDATE). Il en est de même pour tous les domaines dans lesquels des articles sont détaillés AUTOMATIC d'ensembles.
12. Les articles des ensembles qui ont été cités dans les critères de sélection d'ensemble doivent être rangés dans des domaines à l'état prêt (READY).
13. L'article référencé par nom-article reste accessible au programme dans la zone UWA.

Implantation d'article

Le SGBD, à la suite d'une instruction STORE, affecte une clé-base-de-données unique à l'article utilisateur. La définition du schéma et le programmeur COBOL peuvent déterminer un certain nombre de contraintes dans l'affectation d'une clé-base-de-données. Ces contraintes dépendent du mode de localisation spécifié pour l'article et sont indiquées ci-dessous en fonction des trois modes de localisation possibles :

DIRECT

Si le mode de localisation est DIRECT, les données spécifiées sont utilisées pour affecter une clé-base-de-données au nouvel article. Ces données doivent contenir une clé-aire ou une clé-base-de-données. Dans le dernier cas, le numéro d'aire n'est pas pris en compte dans la sélection de l'aire.

1. La sélection d'aire, se fait, s'il existe un choix, au moyen du paramètre d'identification d'aire spécifié dans le schéma.
2. Si la clé d'aire fournie se trouve dans la plage d'articles et n'est pas déjà affectée à un article existant, le SGBD l'affecte au nouvel article.
3. Si la clé d'aire fournie se trouve à l'intérieur de la plage d'articles mais est déjà affectée à un article existant, le SGBD poursuit sa recherche en aval et affecte la première clé rencontrée au nouvel article. S'il arrive en fin de la plage, le SGBD recommence sa recherche en début de plage.
4. Si la clé-base-de-données fournie ne se trouve pas à l'intérieur de la plage d'articles, le SGBD signale une erreur.

Remarque : Il est conseillé d'éviter autant que possible le mode de localisation DIRECT car celui-ci ne se prête pas à une réorganisation automatique au moyen d'un utilitaire standard.

CALC

Si le mode de localisation de l'article est CALC, le contenu des données UWA spécifiées est utilisé pour affecter une clé-base-de-données.

1. En cas de choix c'est un domaine unique qui est sélectionné au moyen du paramètre d'identification d'aire spécifié dans le schéma.
2. Les données spécifiées dans le mode de localisation sont utilisées pour déterminer le "compartiment" CALC dans lequel l'article doit être rangé. L'article est implanté sur la première page du compartiment ou sur une page suivante.

VIA SET

Le SGBD, lorsque l'article est localisé par VIA SET affecte une clé-base-de-données en fonction de la clé-base-de-données de l'article maître du type d'ensemble spécifié dans le mode de localisation. La procédure suivie pour affecter une clé-base-de-données et une implantation physique à un article localisé par VIA SET, est la suivante :

1. Les critères de sélection propres à l'ensemble désigné sont utilisés pour localiser l'article maître de l'ensemble. Cette procédure peut impliquer l'exploration de plusieurs niveaux d'itinéraire avant que cet article ne soit trouvé.
2. S'il existe un choix, c'est un domaine qui est sélectionné au moyen du paramètre d'identification d'aire spécifié dans le schéma.
3. Si le type d'article maître de l'ensemble spécifié et le nouveau types d'article se trouvent sur des plages différentes de la même aire, ou si ce type d'article se trouve dans des aires différentes, le nouvel article est implanté relativement à la position de son article maître dans la plage. Si les deux articles sont dans la même plage, le nouvel article est implanté le plus près possible de la position du maître.

Si l'itinéraire de recherche n'est constitué que d'un seul niveau identifié par APPLICATION et si le classement dans l'ensemble est NEXT ou PRIOR, la recherche d'une clé-base-de-données libre se poursuit à partir de la position de l'article courant d'ensemble et non à partir de la position qui a été indiquée ci-dessus. Ceci permet d'optimiser le temps de traitement lorsque le type d'ensemble représente un fichier séquentiel volatile s'étendant sur de nombreuses pages.

- Remarques :**
1. Comme l'article enregistré n'est rattaché à aucun ensemble MANUAL, les indicateurs de point courant de tous les ensembles dont la caractéristique est MANUAL, ne seront pas mis à jour. Cette règle s'applique, que RETAINING ait été spécifié ou non.
 2. En cas de mode de localisation VIA SET et pour un article destiné à être rangé sous la spécification détail MANUAL de l'ensemble désigné, le nouvel article est implanté physiquement comme s'il était destiné à devenir un article détail de l'ensemble. La sélection d'ensemble a lieu pour localiser la position du maître adéquat mais le nouvel article n'est pas rattaché à l'ensemble.
 3. C'est le paramètre défini par le schéma qui est utilisé lorsqu'il y a sélection d'un domaine en vue d'y ranger un article. Il revient au programme exécutant l'instruction STORE de disposer le nom du domaine adéquat dans le paramètre avant d'exécuter la fonction STORE.
 4. Il est possible qu'il faille sélectionner une aire pour localiser l'article de point d'entrée lorsque les règles de sélection des ensembles AUTOMATIC ou de l'ensemble VIA l'exigent. C'est le cas notamment lorsque le point d'entrée est un article CALC enregistré dans plusieurs domaines. Le programme COBOL doit transférer le nom-domaine adéquat dans le paramètre AREA-ID avant exécution de la fonction STORE.

Langage DML de manipulation des données

5. Comme aucune garantie n'existe qu'un article localisé par DIRECT ait reçu la clé fournie par le programme, il est conseillé de vérifier ce point lorsque les opérations sont d'une importance capitale. L'instruction ACCEPT permet de vérifier la réalité de l'affectation de la clé que l'on compare ensuite à la clé-base-de-données fournie. La fonction STORE n'altère pas la donnée contenant la clé de base de données, même si l'affectation de la clé à l'article n'a pas lieu.
6. En cas d'erreur de STORE, la base de données est restaurée à l'état où elle se trouvait avant que l'instruction ne soit exécutée. Il n'y a pas utilisation des journaux. De plus la non exécution d'une instruction n'affecte pas les indicateurs de point courant.

FUNCTION	CAUSE D'ERREUR	REGISTRES	SPECIAUX	IDS	
		DB-STATUS	REALM	RECORD	SET
STORE	Aucune; article rangé	0000000	X	X	-
	Non-ouverture de la BD	1509100			
	Données ne satisfaisant pas les contrôles de validité	1505200		X	
	Nom-aire erroné dans AREA-ID ou ne correspondant pas au type article	1504300		X	
	Aire non à l'état prêt	1509100	X	X	(X)
	Aire non sous mode mise à jour	1509200	X	X	(X)
	Clé-aire erronée	1504100	X	X	
	Épuisement d'espace aire	1580200	X	X	
	Données en double non autorisées (clé CALC ou cle-de-tri ou autre zone)	1505100	(X)	X	(X)
	Sélection d'ensemble impossible	1502300		X	X
	Absence de description article UWA	1573660		X	

5.7.12 USE

Fonction

Permet de spécifier les procédures à exécuter lorsqu'une instruction DML provoque une erreur dans la base de données.

Format général

```
USE FOR DB-EXCEPTION.
```

Règles de syntaxe

1. L'instruction USE, si elle est utilisée, doit suivre immédiatement l'en-tête de section de la partie DECLARATIVES de la PROCEDURE DIVISION et doit être suivie d'un point. Le reste de la section doit être occupé par un (ou plusieurs) paragraphe(s) de procédure à utiliser.
2. Une, et une seule, instruction USE FOR DB-EXCEPTION peut figurer dans la partie DECLARATIVES.
3. Aucune référence à une procédure non déclarative ne peut figurer dans USE.

Règles générales

1. Toute exécution d'une section de traitement d'erreur de la base de données (ERROR-PROCESSING SECTION) doit être précédée de la mise aux valeurs requises des registres spéciaux DB-STATUS, DB-REALM-NAME, DB-RECORD-NAME, DB-SET-NAME et DB-DETAILED-STATUS.
2. Une section de traitement d'erreur de base de données entre en action dès que l'exécution d'une instruction quelconque provoque une erreur dans la base de données. A la fin de l'exécution de cette section, le contrôle revient à l'instruction qui suivait celle qui a provoqué l'erreur, sauf si une instruction d'interruption STOP RUN doit être exécutée dans la partie DECLARATIVES.

Exemples :

1. En l'absence de toute instruction USE FOR DB-EXCEPTION le programme doit vérifier le contenu du registre DB-STATUS pour s'assurer, après exécution de chaque fonction DML, qu'il ne contient pas de valeur différente de zéro. Les fonctions exécutées normalement renvoient un état zéro, alors que les fonctions qui ne le sont pas renvoient une valeur qui dépend à la fois de l'action corrective qui a été tentée et de la condition d'erreur elle-même. A titre d'exemple, une séquence d'instruction pourrait s'écrire :

```

MOVE "valeur" TO DONNEES-CONTROLE-CALC.
FIND ANY ART-CALC RECORD.
IF BD-STATUS IS EQUAL TO "0502400" GO TO ART-ABSENT.
ELSE IF DB-STATUS IS NOT EQUAL TO ZERO GO TO AUTRE-ERREUR.

```

2. Une seule instruction USE FOR DB-EXCEPTION peut être utilisée pour traiter toutes les conditions d'erreur. Toutefois, cette procédure doit distinguer entre les événements "normaux" tels que la détection "fin d'ensemble" ou "article introuvable" et ceux qui constituent à proprement parler des erreurs. La procédure dispose pour cela de toutes les données des registres spéciaux, dont la plupart doivent être mises à jour en cas d'erreur et de toutes les données de la DATA DIVISION du programme. Des drapeaux indicateurs d'état peuvent être disposés en fonction du type d'erreur à signaler et de l'identité des articles, ensembles ou domaines impliqués. Par exemple :

```

77 DRAPEAU-ETAT          PIC 9          VALUE 0.
88 FIN-ITINERAIRE       VALUE 1.
88 ART-ABSENT           VALUE 2.
88 ERREUR-BLOQ         VALUE 3.
PROCEDURE DIVISION.
DECLARATIVES.
ERROR-PROCESSING SECTION.
    USE FOR DB-EXCEPTION.
ERROR-PROCESSING SECTION.
    IF DB-STATUS IS EQUAL TO "0502100" GO TO DRAP-FIN-ENSEMBLE.
    IF DB-STATUS IS EQUAL TO "0502400" GO TO ART-ABS-ENSEMBLE.
    MOVE 3 TO DRAPEAU-ETAT.
    GO TO EXIT-EXIT.
DRAP-FIN-ENSEMBLE.
    MOVE 2 TO DRAPEAU-ETAT.
    GO TO EXIT-EXIT.
.
EXIT-EXIT. EXIT.
END DECLARATIVES.
.
FIND NEXT ART-RECHERCHE WITHIN CET-ENSEMBLE.
IF FIN-ITINERAIRE GO TO FIN-D'EXECUTION.
.

```

Dans cet exemple, chaque condition d'erreur relevée dans la base de données provoque le déclenchement d'une instruction USE FOR DB-EXCEPTION. Cette procédure ne fait que vérifier les valeurs de certains codes d'état et les indicateurs de condition des ensembles à l'intention du programme. C'est le programme qui doit par la suite tester les conditions pour savoir si elles sont vérifiées ou non. La procédure USE elle-même, lorsqu'elle détecte une erreur, peut provoquer l'arrêt prématuré du programme ou une fin d'exécution anormale.

5.8 FONCTIONS D'IMPLANTATION DE POINTS DE REPRISE ET DE POINTS DE CONSOLIDATION

La totalité de l'activité ou de la transaction, et pas seulement la base de données, est impliquée par ces fonctions. Bien qu'elles soient fournies par le système et non par le SGBD, elles sont mentionnées ici à cause de leurs implications avec la spécification USAGE-MODE des instructions READY.

5.8.1 Comparaison entre points de reprise et points de consolidation

- Les notions de points de reprise et de points de consolidation sont similaires en ce sens qu'elles rendent compte d'un état du traitement de la base de données et de l'unité d'exécution auxquels il est possible de revenir lorsque le point de reprise ou le point de consolidation suivant ne peut être atteint normalement.
- Les deux notions diffèrent cependant en ce qui concerne le partage de la base de données.
- Le point de reprise s'applique au cas d'une réservation exclusive de la base de données à l'unité d'exécution (USAGE-MODE EXCLUSIVE), depuis le début jusqu'à la fin. Chaque point de reprise ne constitue pas une occasion d'accès pour les autres unités d'exécution à la base de données. Celle-ci ne devient accessible qu'à la fin de l'unité d'exécution. Les incidents qui peuvent empêcher l'UE de passer d'un point de reprise à l'autre ne sont pas des conflits de ressource, mais des erreurs de programme ou des défaillances système. C'est dire que dans un cadre d'exploitation soigneusement testé, ce genre d'incident est hautement improbable.
- Le point de consolidation s'applique au cas d'une réservation partagée de la base de données entre plusieurs unités d'exécution (USAGE-MODE MONITORED). Les pages de base de données sont réservées pendant l'exécution de l'unité de consolidation et libérées au point de consolidation suivant. Contrairement à ce qui a été dit pour les erreurs de programme ou les défaillances système, ce genre de conflit pour les pages de base de données sont probables et peuvent provoquer, soit une attente temporaire, soit un blocage, du genre de ceux qui peuvent se produire avec des programmes travaillant en mode recherche-extraction (RETRIEVAL). Les blocages se résolvent par un retour au dernier point de consolidation avec libération des pages déjà réservées.
- L'intervalle entre deux points de reprise peut durer jusqu'à une heure, tandis que l'intervalle entre deux points de consolidation peut être de l'ordre de quelques secondes.
- La notion de point de consolidation a été d'abord introduite dans le système TDS. Elle a été ensuite étendue aux programmes en traitement par lots ou IOF qui ont des accès concurrents à la base de données, soit entre eux, soit avec TDS.
- Ces programmes doivent être découpés en unité de consolidation, en disposant à intervalles réguliers des appels à une fonction système appelée (H_GAC_UCOMIT) qui enregistre régulièrement l'état du programme et libère les pages de base de données antérieurement réservées en modes exclusif ou partagé.

- Lorsque TDS opère dans un environnement concurrent, la fréquence des points de consolidation des programmes en traitement par lots doit être suffisamment grande pour ne pas perturber le temps de réponse des transactions TDS. Si seuls les programmes en traitement par lots opèrent dans un environnement concurrent, la fréquence peut être moins élevée mais le risque d'attente ou de blocage s'accroît du fait du plus grand nombre de pages verrouillées par une unité de consolidation.
- Que le mode d'utilisation soit MONITORED ou EXCLUSIVE n'affecte pas la logique de fonctionnement de l'unité de consolidation, car celle-ci réserve toutes les pages nécessaires pour assurer la cohérence des interventions qui sont prises tout au long de son exécution.
- A l'inverse, la logique du programme en traitement par lots doit tenir compte du fait qu'une unité de consolidation ne peut tenir pour définitives les modifications qui ont été apportées par l'unité de consolidation précédente, puisque des unités d'exécution concurrentes peuvent avoir, entre-temps, apportées d'autres modifications.
- Il est parfois nécessaire de traiter sous mode EXCLUSIVE les programmes en traitement par lots qui ont été écrits en vue d'accès concurrents. Les instructions ASSIGN/DEFINE permettent à l'utilisateur d'annuler les options de READY et de supprimer l'accès concurrent au niveau page. En ce cas, la fonction système H_GAC_UCOMIT cesse automatiquement son activité, sauf si un paramètre indique la nécessité d'établir un point de reprise. Ce dernier dispositif évite de faire appel à la fonction système H_CK_UCHKPT. Il peut s'utiliser toutes les n unités de consolidation de manière à obtenir la même fréquence de points de consolidation que si H_CK_UCHKPT avait été programmé.
- Les appels de programme à la fonction H_CK_UCHKPT sont inopérants lorsqu'ils se produisent au cours d'une activité qui accède à la base de données sous mode MONITORED.

5.8.2 Fonction système de mise en oeuvre d'un point de reprise

Dans le traitement par lots et sous IOF, un programme peut appeler cette fonction en utilisant:

```
CALL "H_CK_UCHKPT" USING CKMODE CKINF
```

Les paramètres, qui doivent être définis par l'utilisateur, sont:

```
01 CKMODE USAGE COMP-2. (sortie)  
01 CKINF PIC X(32). (sortie)
```

Pour plus de détails, se reporter au System Administrator's Manual (Réf: 47 A2 01US).

Il n'existe pas de points de reprise sous TDS.

5.8.3 Fonction système de mise en oeuvre d'un point de consolidation

Dans le traitement par lots et sous IOF, un programme peut appeler cette fonction en utilisant :

```
CALL "H_CK_UCOMIT" USING CKMODE CKINF NUMLOCK NOCHKPT
```

Les paramètres sont :

```
01 CKMODE USAGE COMP-2. (sortie)
01 CKINF PIC X (32). (sortie)
01 NUMLOCK COMP-1. (entrée)
01 NOCHKPT PIC X. (entrée)
```

CKMODE et CKINF ont la même signification et les mêmes valeurs que H_CK_UCHKPT.

NUMLOCK définit le nombre maximum d'entrées dans la table de verrouillage associée à l'unité de consolidation de départ.

NUMLOCK = -1 demande au système de choisir la valeur implicite (50).

NOCHKPT spécifie s'il doit y avoir établissement d'un point de reprise dans le cas où il n'y aurait pas de fichier en mode MONITORED.

NOCHKPT = " ": établissement d'un point de reprise.

NOCHKPT = "Y": pas de point de reprise.

Dans un cadre d'exploitation TDS, la fonction correspondante est appelée, si elle n'est pas implicite, par l'instruction suivante:

```
CALL "H_DFCCMIT"
```

Pour plus de détails, se reporter au TDS Programmer's Manual (Réf: 47 A2 03UT)

5.9 ACCES AU FICHIER DE JALONNEMENT IDSTRACE

Le programmeur peut disposer des appels à la fonction du SGBD H_IA_UCTRACE pour exécuter des opérations sur le fichier de jalonnement IDSTRACE :

- Ecriture d'information du programme sur le fichier IDSTRACE,
 - Ouverture, fermeture de IDSTRACE,
 - Invalider ou valider les opérations d'écriture du programme et/ou du SGBD sur IDSTRACE.
1. La fonction d'écriture de H_IA_UCTRACE permet à l'utilisateur d'interclasser sur le même fichier des données de jalonnement du programme et du SGBD.
 2. Le fichier IDSTRACE est ouvert et fermé par le SGBD en début et en fin de session respectivement.

Les fonctions d'ouverture et de fermeture de H_IA_UCTRACE permettent à l'utilisateur d'ouvrir IDSTRACE avant la première instruction READY d'une session IDS et de le fermer après la dernière FINISH. Si tel est le cas, les demandes d'ouverture et de fermeture du fichier IDSTRACE par le SGBD ne sont pas prises en compte.

La Figure 5-1 illustre un programme DML qui doit ouvrir ou fermer IDSTRACE et interclasser des informations fournies par le programme avec les données de jalonnement du SGBD.

Lorsqu'une même unité d'exécution accède à deux bases de données, elle peut enregistrer des informations de jalonnement du programme et du SGBD des deux bases de données sur le même fichier IDSTRACE. Le nom de fichier interne (nfi) de jalonnement doit être le même pour le programme et les deux bases de données. L'ouverture du fichier IDSTRACE se fait dès la première demande, qu'elle soit soumise par le programme ou par l'une des deux sessions IDS, et la fermeture par la dernière demande.

La Figure 5-2 permet d'illustrer un programme COBOL PO qui commande deux programmes DML P1 et P2, qui accèdent chacun à une base de données différente. Il revient à P0 d'ouvrir et de fermer le fichier IDSTRACE. P0, P1 et P2 interclassent de l'information de programme dans les données de jalonnement du SGBD.

3. Les fonctions validation et invalidation de H_IA_UCTRACE permettent au programmeur de concentrer le jalonnement sur une portion quelconque du programme et d'y appliquer des critères de sélection plus affinés que ceux de la commande TRACE du langage de commande de l'unité d'exécution.

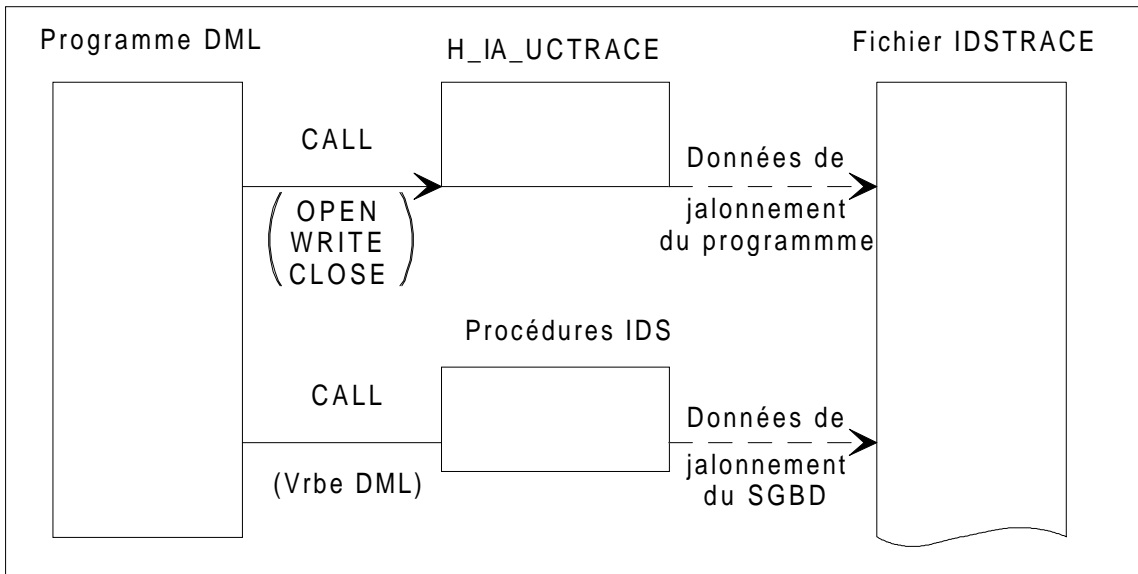


Figure 5-1. Accès utilisateur à IDSTRACE (un programme)

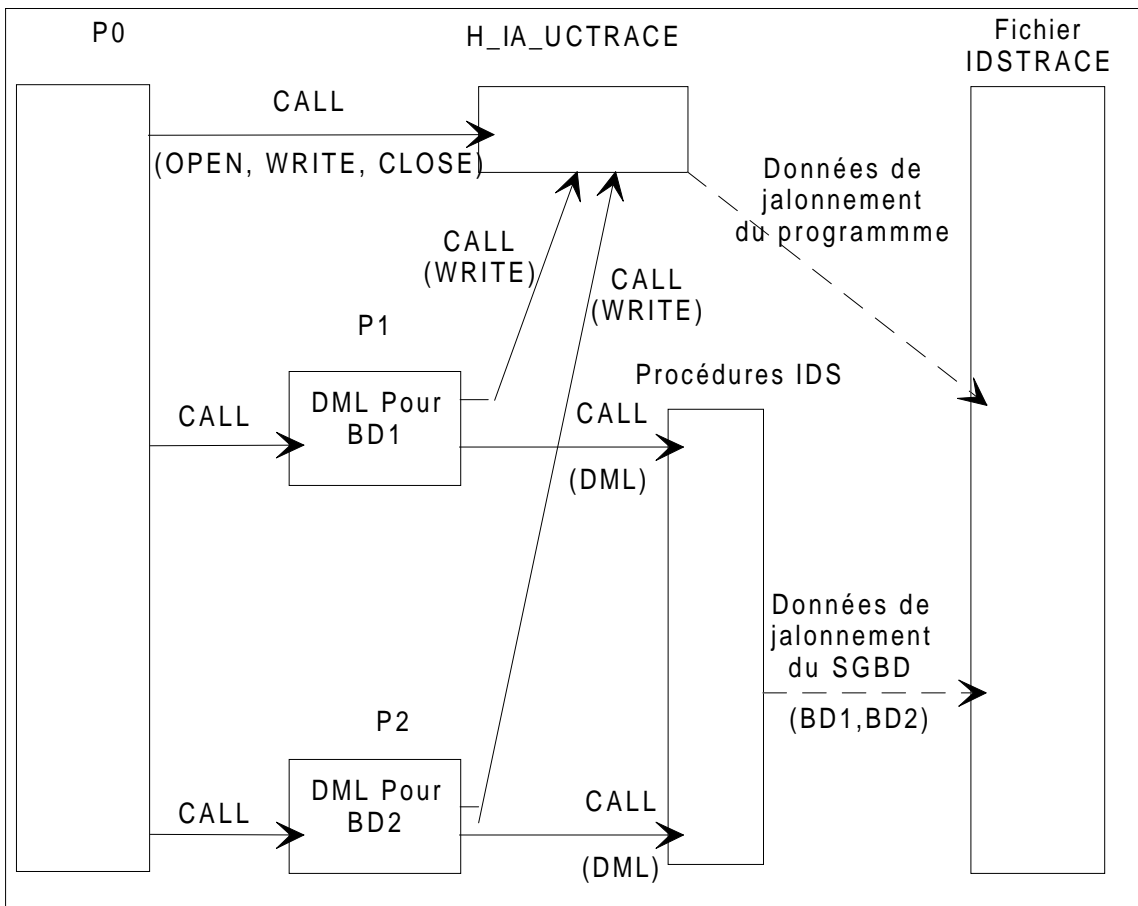


Figure 5-2. Accès utilisateur à IDSTRACE (plusieurs programmes)

Format de l'instruction CALL

```
CALL "H_IA_UCTRACE" USING TRACEPARAM TRACEIFN
```

Paramètres

```
01 TRACEPARAM.

    02 TRACEFUNCTION COMP-1. (entrée)
    02 TRACERETCODE  COMP-1. (entrée)
    02 TRACELENGTH   COMP-1. (entrée)
    02 TRACEDATA PIC X(n).  (entrée)

01 TRACEIFN PIC X(8).      (entrée)
```

Description des paramètres

TRACEFUNCTION code de fonction.

Les valeurs exprimées en décimal sont indiquées ci-dessous.

VALEUR	FUNCTION
1	Invalidation des opérations d'écriture sur IDSTRACE
2	Invalidation des opérations d'écriture du SGBD sur IDSTRACE
3	Invalidation des opérations d'écriture du programme et du SGBD sur IDSTRACE
4	Validation des opérations d'écriture du programme sur IDSTRACE
5	Validation des opérations d'écriture du SGBD sur IDSTRACE
6	Validation des opérations d'écriture du programme et du SGBD sur IDSTRACE
16	Ecriture sur IDSTRACE
17	Ouverture du fichier IDSTRACE en mode sortie (OUTPUT)
18	Ouverture du fichier IDSTRACE en mode extension (EXTEND)
19	Fermeture de IDSTRACE
VALEUR	FUNCTION
0	Exécution correcte de la fonction ou fonction rendue caduque par une "invalidation" antérieure (WRITE)
1	La fonction n'a pas été exécutée pour cause de paramètres incorrects ou d'anomalies quelconques

TRACELENGTH Nombre de caractères (n) de TRACEDATA.

TRACEDATA Données de programmes à écrire sur le fichier IDSTRACE.
Remarque : TRACELENGTH et TRACEDATA ne sont obligatoires que lorsque TRACEFUNCTION = 16. Autrement ils sont sans objet.

TRACEIFN Nom de fichier interne (nfi) de IDSTRACE.

Règles

1. Lorsque le fichier IDSTRACE n'est pas couramment ouvert, les fonctions validation/invalidation, écriture et fermeture ne sont pas exécutées.
2. Lorsque IDSTRACE n'est par couramment fermé, la fonction d'ouverture n'est pas exécutée.
3. Lorsque IDSTRACE est ouvert les opérations d'écriture sont implicitement validées.
4. Dans un cadre d'exploitation TDS, seule la fonction d'écriture est disponible. Les autres fonctions ne sont pas exécutées.

5.10 SIMULATION D'ALGORITHME CALC (HASHING)

Cette fonction détermine le numéro relatif de page d'une plage où un article CALC est soumis à randomisation, sous réserve qu'il n'y ait pas de débordement de page.

Format de l'instruction CALL

```
CALL "H_IA_UHASH" USING HASHFUNC HASHRANGEDESC
                                HASHKEYDESC HASHDATA
```

Paramètres

```
01 HASHFUNC.
    02 HASHRETCODE COMP-1. (entrée)
    02 HASHPAGENUMBER COMP-2.

01 HASHRANGEDESC.          (sortie)
    02 HASHNUMPAGE COMP-2.
    02 HASHCALCINTERVAL COMP-1.
    02 HASHINTERVALUNIT COMP-1. VALUE 0

01 HASHKEYDESC.           (entrée)
    02 HASHUMITTEM COMP-1 VALUE n.
    02 HASHITEMDESC OCCURS n.
        03 HASHITEMTYPE COMP-1.
        03 HASHITEMLENGTH COMP-2.
        03 HASHITEMSCALE COMP-1.
        03 HASHITEMPTR COMP-2.

01 HASHDATA                (entrée)
```

Données de clé dans l'article

Description des paramètres

Les valeurs numériques, en décimal, sont indiquées ci-dessous :

HASHRETCODE code de retour

VALEUR	SIGNIFICATION
0	Simulation exécutée
1	Paramètre erroné; qui peut être : -HASHNUMPAGE =0 -HASHCALCINTERVAL = 0 or < 0••-HASHNUMPAGE non-multiple de HASHCALCINTERVAL••HASHN255 -HASHINTERVALUNIT < 0••-HASHNUMPAGE non-multiple de HASHCALCINTERVAL••HASHN256 -Longueur totale de la clé <0••HASHNUMPAGE non-multiple de HASHCALCINTERVAL••HASHN256 -HASHITEMTYPE or HASHITEMLENGTH incorrect -HASHITEMPTR = 0
2	Données décimales interdites dans clé CALC

HASHPAGENUMBER Numéro relatif de page dans la plage à la suite de la randomisation (0 < HASHPAGENUMBER <= HASHNUMPAGE - 1).

Le calcul s'effectue de la manière suivante: le nombre de pages de la plage HASHNUMPAGE est divisé par HASHCALCINTERVAL fournissant le nombre de compartiments. Les données de clé CALC décrites par HASHKEYDESC sont extraites de HASHDATA, concaténées et recalculées par l'instruction HASH, donnant une valeur binaire de 32 bits. Cette valeur est ensuite étendue à 64 bits avec remplissage de zéro et divisée par le nombre de compartiments. Le reste est multiplié par HASHCALCINTERVAL qui produit HASHPAGENUMBER.

HASHNUMPAGE Nombre de pages de la plage. Il doit être un multiple de HASHCALCINTERVAL.

HASHCALCINTERVAL Nombre de pages de l'intervalle CALC.

HASHINTERVAL UNIT Doit être à zéro.

HASHNUMITEM Nombre de données de clé CALC.

HASHITEMDESC Groupe répétitif apparaissant un nombre HASHNUMITEM de fois et décrivant chaque donnée de clé.

HASHITEMTYPE,
HASHITEMLENGTH Type et longueur de la donnée de clé.

TYPES DE DONNEE	HASHITEMTYPE	HASHITEMLENGTH
UNSIGNED UNPACKED DECIMAL	9	Nombre de chiffres

Langage DML de manipulation des données

UNSIGNED PACKED DECIMAL	57	
UNSIGNED PACKED-2 DECIMAL	25	décimaux
SIGNED UNPACKED DECIMAL	1	(pas de signe)
SIGNED PACKED DECIMAL	17	1 <= m <= 30
SIGNED BINARY	3	Nombre de chiffres binaires (signes non compris 15 ou 31)
CHARACTER	0	Nombre de caractères 1 <= m <= 256

HASHITEMSCALE	<p>Echelle de la donnée de clé.</p> <p>0 pour une donnée binaire ou un caractère,</p> <p>$\pm p$ pour une donnée décimale.</p> <p>L'échelle n'est pas couramment utilisée dans le calcul d'algorithme.</p>
ASHITEMPTR	<p>Position de caractère, (débutant à 1), dans HASHDATA où commence la donnée de clé.</p>
HASHDATA	<p>Article contenant les données de clé CALC et éventuellement d'autres données.</p>

Exemple :

Considérons une plage de 101 pages, un intervalle CALC de 1 page et une clé CALC composée de 3 éléments PIC XX, PIC 999, PIC 10 situés sur les positions 8, 10, 13 de l'article CALC.

Manuel de référence IDS/II

Les trois derniers paramètres de H_IS_UCHASH se décrivent comme suit :

01 HASHRANGEDESC.

02 HASHNUMPAGE COMP-2. VALUE 101
02 HASHCALCINTERVAL COMP-1. VALUE 1
02 HASHINTERVALUNIT COMP-1. VALUE 0

01 HASHKEYDESC.

02 HASHUMITTEM COMP-1 VALUE 3.
02 HASHITEMDESC1.

03 HASHITEMTYPE1 COMP-1. VALUE 0.
03 HASHITEMLENGTH1 COMP-2 VALUE 2.
03 HASHITEMSCALE1 COMP-1 VALUE 0.
03 HASHITEMPTR1 COMP-2 VALUE 8.

02 HASHITEMDESC2.

03 HASHITEMTYPE2 COMP-1. VALUE 9.
03 HASHITEMLENGTH2 COMP-2 VALUE 3.
03 HASHITEMSCALE1 COMP-1 VALUE 0.
03 HASHITEMPTR1 COMP-2 VALUE 10.

02 HASHITEMDESC3.

03 HASHITEMTYPE3 COMP-1. VALUE 0.
03 HASHITEMLENGTH1 COMP-2 VALUE 1.
03 HASHITEMSCALE1 COMP-1 VALUE 0.
03 HASHITEMPTR1 COMP-2 VALUE 13.

01 HASHDATE

02 R01-IDENT PIC X(7).
02 R01-CALC.
03 R01-CALC1 PIC XX.
03 R01-CALC2 PIC 999.
03 R01-CALC3 PIC X.
02 R01-NUMBER PIC 9(4).

6. Compilation et édition de liens du COBOL ET DU DML

6.1 ELEMENTS SL ET DD

1. Le COBOL manipule deux types d'éléments :
 - Eléments SL (Source Language) composant les COBOL et DML origines d'entrée et le COBOL origine de sortie ;
 - Eléments DD (Data Description) composant le schéma résultant utilisé dans le traitement DML.
2. Le COBOL/DML origine d'entrée peut être enregistré dans :
 - Une rubrique de description d'entrées comportant l'option TYPE COBOL ou DATASSF ;
 - Une unité de bibliothèque SL. L'option TYPE de cette unité doit être COBOL, COBOLX ou DATASSF.
3. Le schéma résultant est une unité d'une bibliothèque BIN.
4. Les bibliothèques contenant le schéma résultant doivent faire l'objet d'une affectation statique au moyen des mots réservés de description de nfi, DDLIB1, DDLIB2, DDLIB3.

L'unité contenant le schéma dont le nom est référencé dans la SUB-SCHEMA SECTION, est recherchée dans les bibliothèques affectées suivant la séquence DDLIB1, DDLIB2, DDLIB3.
5. Le schéma résultant utilisé par le COBOL peut être à l'état "DDL" ou "DDL-DMCL".
6. Le programme compilé contient le nom et la date de référence DDL pour le schéma utilisé dans le traitement.

Au moment de l'exécution, ce groupe nom et date devra correspondre à celui qui a été indiqué pour le schéma chargé pour exécution.

6.2 JCL DU COBOL ETENDU

L'appel au COBOL se fait en utilisant les options spécifiques de son JCL étendu pour permettre la compilation d'un programme COBOL avec des instructions DML.

COBOL

```
DDLIB1=(description-de-bibliothèque)
[DDLIB2=(description-de-bibliothèque)]
[DDLIB3=(description-de-bibliothèque)]
```

```
[ {NDDLIST} ]
[ {DDLIST} ] ;
```

Description des paramètres spécifiques de COBOL

DDLIBi Ces mots-clés permettent de définir les bibliothèques contenant le schéma résultant ainsi que leur itinéraire de recherche.

DDLIST
ou
NDDLIST

Ce paramètre permet à l'utilisateur de demander une description de tous les articles utilisés. La valeur implicite est NDDLIST, qui ne demande aucun listage de description.

6.3 COMPILATION COBOL

La compilation du COBOL origine s'exécute par le compilateur COBOL. A l'exception du mot-clé LEVEL=L64 qui doit être spécifié dans l'instruction JCL COBOL, il n'existe aucune contrainte particulière IDS/II.

L'instruction JCL COBOL, sauf pour les options définies plus haut, reste inchangée. Pour plus de détails, se reporter au manuel COBOL USER GUIDE (réf. 47 A2 02UL).

6.4 EDITION DE LIENS

Après la compilation du programme COBOL/DML, l'utilisateur doit exécuter LINKER pour produire un module chargeable. Cette procédure d'Édition de liens se conforme au cas général décrit dans le manuel LINKER, mais l'édition de liens de programmes contenant du DML peut nécessiter la fourniture à l'Éditeur de liens d'une suite de commandes supplémentaires.

L'instruction LINKER contient un groupe de paramètres:

```
{ COMFILE = { *nom-rubrique-description-entrées          } }
  {          { (description-fichier-séquentiel-entrées) } }
{ COMMAND = '{commande}...' }
```

qui permettent de fournir ces commandes d'édition de liens. [Les commandes LINKER qui peuvent se révéler nécessaires sont :

<pre>SEMPOOL=(NUMSN = +p) STACK1=(INITSIZE = 3K, MAXSIZE = 16K)</pre>

La première commande est nécessaire si le nombre de sémaphores requis par UFAS est plus grand que celui qui a été réservé implicitement par l'instruction LINKER. En ce cas, l'utilisateur en est informé au moment de l'exécution d'une instruction READY par un message (TASKM, ENTRYOV) au JOR. La valeur p doit être établie à 7 + 4 * nombre d'aires. Le signe + est obligatoire et n'a pas la même signification qu'un espace. De manière à éviter un calcul, on peut formuler la commande SEMPOOL de la manière suivante :

```
SEMPOOL=(NUMSN=255)
```

qui fournit le nombre maximum possible de sémaphores.

La deuxième commande se rapporte au nombre de débordements de pile se produisant au moment de l'exécution. Ce nombre doit être vérifié dans le JOR, particulièrement lorsque la base de données est ouverte en mode mise à jour. Si sa valeur est plus grande que 200, la commande STACK1 doit être fournie pour augmenter la taille initiale de pile de l'anneau 1 (valeur implicite 28k). Si 3K ne sont pas suffisants, des valeurs plus élevées peuvent être choisies.

6.5 EXEMPLE

Un schéma résultant appelé INVENTAIRE est situé dans une unité INVENTAIRE de la bibliothèque UIDS.BIBBIN.

L'utilisateur désire compiler, à l'aide de ce schéma, un programme COBOL/DML appelé CONTROLE17. Le programme est écrit de la manière suivante :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. CONTROLE17
      ...
DATA DIVISION.
SUB-SCHEMA SECTION.
DB INVENTAIRE.
      ...
PROCEDURE DIVISION.
      ...
      READY
      ...
      FINISH
      ...
      END COBOL
```

L'origine en entrée est constituée d'une unité CONTROLE17 de la bibliothèque UIDS.BIBSL. L'unité compilée sera rangée dans la bibliothèque UIDS.BIBCOMP. On suppose que toutes les bibliothèques sont cataloguées.

```
$JOB DMLCOB USER=MURIEL PROJECT = LONGEOT;
LIB SL INLIB1=UIDS.BIBSL ;
```

```
COBOL SOURCE=CONTROLE17
      DDLIB1=UIDS.BIBBIN
      CULIB=UIDS.BIBCOMP
      LEVEL=L64 MAP XREF DDLIST ;
```

```
$ENDJOB
```

L'utilisateur prépare alors un module chargeable, dépourvu d'options particulières, dans la bibliothèque UIDS.CHARG.

```
$JOB LINK USER=MURIEL PROJECT=LONGEOT ;
LIB CU INLIB1=UIDS.BIBCOMP ;
LINKER CONTROLE17
      OUTLIB=UIDS.CHARG ;
$ENDJOB;
```

Après plusieurs passages du module chargeable, le JOR indique que le nombre de débordements de pile est trop élevé. L'utilisateur corrige cette anomalie en procédant à une nouvelle édition de liens, avec utilisation de la commande STACK1.

```
$JOB RELINK USER=MURIEL PROJECT=LONGEOT ;
LIB CU INLIB1=UIDS.BIBCOMP ;
LINKER CONTROLE17
      OUTLIB=UIDS.CHARG.
      COMMAND= 'STACK1=(INITSIZE=3K, MAXSIZE=16K) ' ;
$ENDJOB ;
```

7. Exécution du programme utilisateur

7.1 CADRE D'EXPLOITATION AU MOMENT DE L'EXECUTION

7.1.1 Conditions préalables

Les opérations suivantes constituent des préalables à l'exécution des programmes DML :

- Traduction des schémas DML et DMCL en schéma résultant.
- Réservation des aires pour ce schéma.
- Compilation, à l'aide de ce schéma, des programmes DML et Edition de liens.

Au moment de l'exécution du schéma, un certain nombre de contrôles de cohérence sont effectués de manière à s'assurer que le schéma cité est bien celui qui a été référencé au moment de la réservation et de la compilation :

- Dès la première instruction READY qui déclenche la session IDS, le nom de schéma et le groupe date heure de référence du DDL enregistrés dans chaque programme par le COBOL, sont comparés au nom et au groupe date heure de référence du DDL de schéma résultant utilisé au moment de l'exécution.
- A chaque instruction READY, le nom de schéma et le groupe date heure de référence du DMCL enregistrés dans le label de chaque aire par commande PREALLOC, sont comparés avec les mêmes du schéma résultant utilisé à l'exécution.

7.1.2 Modes de traitement

Les programmes DML peuvent s'exécuter en traitement par lots, IOF et TDS :

- Le traitement par lots comprend les travaux qui, soit n'établissent aucune ligne de communication, ou, si elles les utilisent, le font indirectement par l'intermédiaire des files d'attente en entrée et en sortie du système de gestion de messages MCS.
- Le cadre d'exploitation IOF correspond aux travaux associés à un terminal sous le contrôle du système interactif IOF.
- Le cadre d'exploitation TDS correspond aux activités qui se déroulent sous le contrôle du sous-système TDS.

Remarque : L'utilitaire base de données DBUTILITY est considéré comme une activité utilisateur lancée par le même type de JCL de base et capable de s'exécuter en traitement par lots ou en interactif (IOF).

7.2 OBJETS DE REFERENCE AU MOMENT DE L'EXECUTION

Pour ce qui concerne IDS/II, une activité utilisateur fait référence à quatre types d'objets :

- Schéma résultant de base de données.
- Aire mémoire de base de données.
- Fichier facultatif IDSOPT contenant les commandes utilisables à l'exécution.
- Fichiers facultatifs IDSTRACE recueillant les informations de jalonnement du programme et du SGBD.

Les schémas résultants de base de données sont rangés dans des bibliothèques de type BIN. Ces bibliothèques doivent faire l'objet d'affectation statique à l'aide des mots réservés de description de fichiers (nfi) DDLIB1, DDLIB2, DDLIB3.

Si un nom de schéma spécifié dans un programme de l'activité n'est pas qualifié dans les commandes utilisables à l'exécution, le schéma résultant à charger est recherché dans les bibliothèques affectées, dans l'ordre de la séquence DDLIB1, DDLIB2, DDLIB3.

Si un nom de schéma est qualifié par une commande SCHEMA utilisable à l'exécution, soit au moyen du nom de fichier externe (nfe) de bibliothèque, soit par l'un des mots réservés DDLIB1, DDLIB2, DDLIB3, le schéma résultant est recherché dans la bibliothèque spécifiée.

Si deux noms de schéma homonymes sont utilisés dans la même activité, ils doivent être enregistrés dans deux bibliothèques différentes et l'un au moins d'entre eux doit être qualifié par les commandes utilisables à l'exécution.

Les schémas résultants nécessaires au moment de l'exécution doivent être à l'état "DDL-DMCL".

Les aires mémoire de base de données doivent faire l'objet d'une affectation statique au moyen des noms de fichiers internes (nfi) définis dans le DMCL ou spécifiés dans les commandes utilisables à l'exécution. Seules les aires qui doivent être mises à l'état READY doivent faire l'objet d'une affectation. Si l'activité fait référence à plusieurs bases de données, tous les nfi d'aires doivent être différents.

Le fichier des commandes utilisables à l'exécution peut être :

- une rubrique de description d'entrées avec option TYPE=DATA ou DATASSF.
- une unité de bibliothèque SL, dont l'option TYPE doit être DATASSF.
- un fichier séquentiel.

Le fichier des commandes utilisables à l'exécution, lorsqu'il existe, doit faire l'objet d'une affectation statique à l'aide du nom de fichier interne réservé IDSOPT.

L'absence de l'instruction ASSIGN IDSOPT indique, dans le JCL, la non existence du fichier des commandes utilisables à l'exécution. En ce cas, ce sont les options implicites qui s'appliquent.

Si plusieurs bases de données entrent dans le traitement, le groupe des commandes utilisables à l'exécution se rapportant à une base de données déterminée est identifié par la commande SCHEMA, première commande de ce groupe.

Les fichiers de jalonnement IDS, sont des fichiers de type standard ou SYSOUT permanent. Ils ne s'utilisent que lorsque des commandes TRACE sont spécifiées dans le langage de commande d'exécution ou lorsque les programmes DML appellent la fonction H_IA_UCTRACE du SGBD.

Le nfi utilisé pour spécifier un fichier de jalonnement peut être :

- Le nom implicite IDSTRACE si nfi n'a pas été redéfini dans une commande TRACE.
- Le nfi spécifié dans la clause IDSTRACE de la commande TRACE.
- Le nfi spécifié dans le second paramètre de H_IA_UCTRACE.

Une opération d'affectation statique par ASSIGN n'est nécessaire que lorsque la sortie du jalonnement n'est pas dirigée sur un fichier standard SYSOUT.

- Lorsque plusieurs bases de données sont impliquées, la sortie du jalonnement (TRACE) peut être dirigée sur différents fichiers SYSOUT, à condition que les nfi soient différents, ou sur le même fichier SYSOUT si les nfi sont identiques.

Dans un cadre d'exploitation TDS :

- L'activité TDS n'a accès qu'à une seule base de données.
- Toute sortie dirigée sur un fichier de jalonnement IDS qualifié par nfi, est automatiquement redirigée sur une unité de la bibliothèque DBUGFILE dont le nfi est réservé à TDS.
- Au moment de la génération TDS, seuls les programmes qui contiennent la locution DB-DESCRIPTIONS IN LINKAGE peuvent faire l'objet d'une procédure facultative "USE".

Toutes ces explications sont illustrées par la Figure 7-1.

Exécution du programme utilisateur

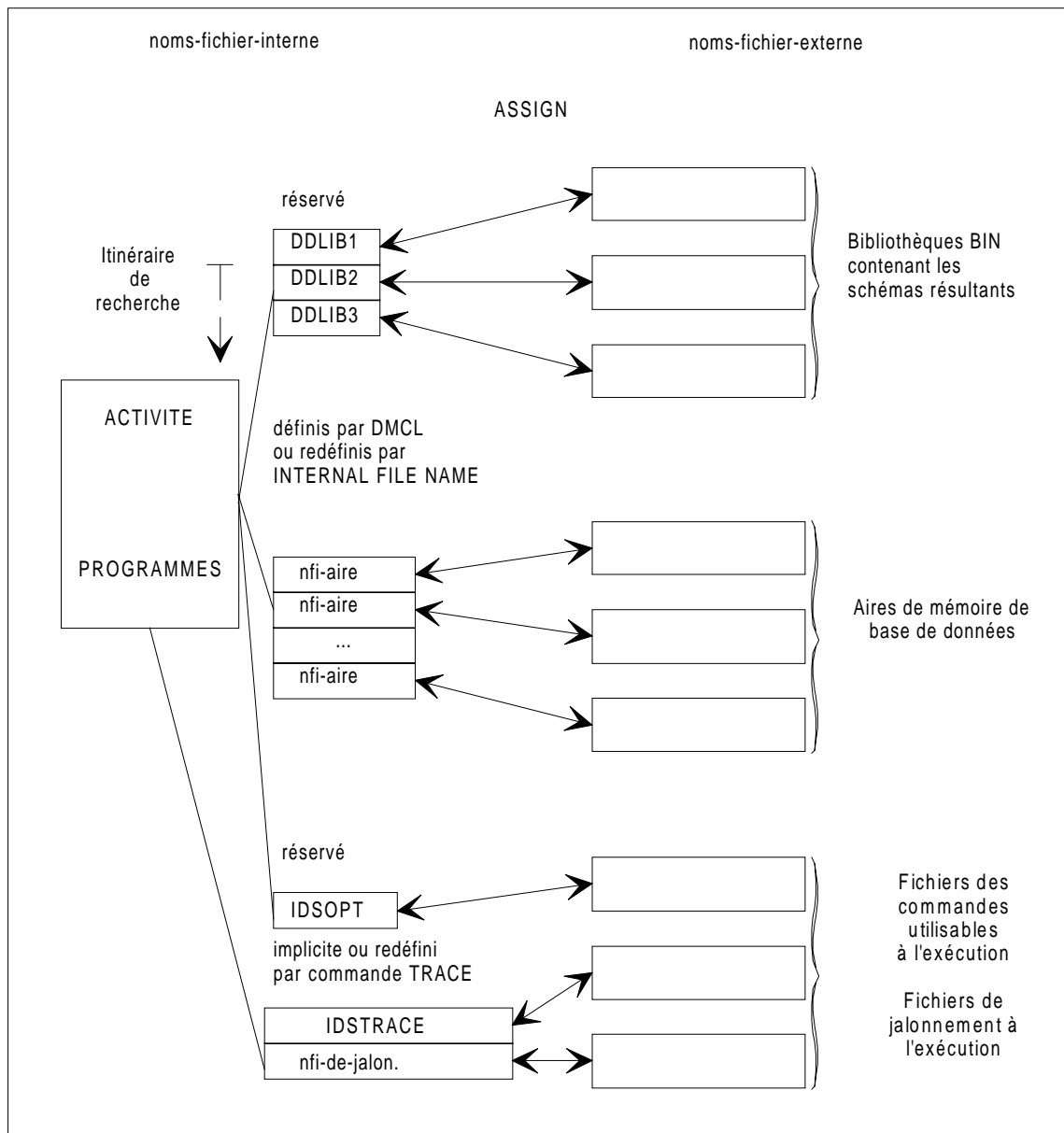


Figure 7-1. Objets référencés à l'exécution

7.3 JCL DE BASE DE L'UTILISATION

Une activité qui lance des programmes IDS doit comprendre le JCL de base suivant:

```
STEP module-chargeable-utilisateur{paramètre-activité} ... [REPEAT] ;
  [SIZE ... ;]
  ASSIGN DDLIB1 description-de-bibliothèque ;
  [ASSIGN DDLIB2 description-de-bibliothèque ;]
  [ASSIGN DDLIB3 description-de-bibliothèque ;]

  {ASSIGN nfi-aire nfi-aire paramètres-de-ASSIGN ; }
  { [DEFINE nfi-aire paramètres-de-DEFINE ;] } ...

  [ ASSIGN IDSOPT {nom-rubrique-description-entrées} ;]
  [ {description-fichier-entrées } ]

  [ASSIGN nfi-jalon description-fichier-simple ;] ...
  [autre instruction pour fichier non-IDS ;] ...

ENDSTEP ;
```

Description des paramètres

REPEAT	Ce paramètre est obligatoire si l'activité se déroule dans un environnement concurrent (sous GAC) ou si elle lance la fonction point de reprise.
SIZE	Le nombre total de tampons (NBBUF) et la taille totale (POOLSIZE) du pool UFAS doivent être spécifiés si les valeurs implicites ne sont pas suffisantes.
DDLIB1	Ces noms réservés de fichiers internes (nfi) indiquent les bibliothèques contenant les schémas résultants ainsi que leur itinéraire de recherche.
nfi-aire	Nom-de-fichier-interne d'une aire mémoire tel qu'il est défini dans le DMCL. Chaque aire mémoire impliquée nécessite une instruction ASSIGN. Celle-ci spécifie le mode d'utilisation de l'aire (paramètres SHARE et ACCESS) qui confirment ou annulent le mode d'utilisation spécifié dans l'instruction READY. Une instruction DEFINE peut être nécessaire pour spécifier les paramètres JOURNAL et READLOCK. Sous TDS, le paramètre READLOCK ne doit pas être spécifié.
IDSOPT	Ce nom réservé de nfi cite le fichier qui contient les commandes utilisables à l'exécution, s'il en existe.

Exécution du programme utilisateur

nfi-jalon. Ce nfi désigne un fichier SYSOUT sur lequel est enregistré l'information de jalonnement.

Le nfi implicite est IDSTRACE. Ce nom peut être redéfini dans une commande TRACE utilisable à l'exécution.

Sous TDS, le nfi de jalonnement ne doit pas être DEBUGFILE.

L'instruction ASSIGN n'est nécessaire que si le fichier de jalonnement n'est pas le type SYSOUT standard.

Exemple :

Un schéma possède les rubriques DMCL suivantes :

```
SCHEMA NAME IS PBL5-TECH-MARKT
AREA NAME IS MANUELS-LOGICIEL
  AREA INTERNAL FILE NAME IS PBL5LOGI ...
AREA NAME IS MANUELS-MATERIEL
  AREA INTERNAL FILE NAME IS PBL5MAT ...
```

La bibliothèque du schéma et les deux aires sont cataloguées au moyen des noms-de-fichiers-externes BD.BIBBIN, BD.LOGI, BD.MATE respectivement.

Des statistiques d'exécution sont demandées et le SGBD doit jalonner sur le fichier SYSOUT standard les instructions STORE qui portent sur au moins dix opérations de transfert entrée-sortie entre les tampons et la base de données.

L'écriture de la rubrique de description d'activité, pour des aires utilisées sous mode MONITORED et protégées par les journaux AVANT et APRES (BOTH), est décrite ci-dessous :

```
STEP ...REPEAT;
ASSIGN DDLIB1   BD.BIBBIN
ASSIGN PBL5LOGI BD.LOGI SHARE = MONITOR ACCESS = WRITE;
DEFINE PBL5LOGI JOURNAL = BOTH;
ASSIGN PBL5MAT  BD.MATE SHARE = MONITOR ACCESS = WRITE;
DEFINE PBL5MAT  JOURNAL = BOTH;
ASSIGN IDSOPT  *COMEXEC;
ENDSTEP;

$INPUT COMEXEC
SCHEMA NAME IS PBL5-TECH-MARKT
STATISTIC WITH TIMING.
TRACE STORE
  THRESHOLD IS 10 I-O
  PRINT CURRENT FIELDS STATISTICS TIMING.
$ENDINPUT;
```

7.4 ACCES CONCURENTS ET PROTECTION DES AIRES

7.4.1 Partage de la base de données

- Le partage de la base de données concerne principalement trois sujets :
 - Le partage de la ressource : partage de l'aire au niveau global ou partage de la page d'aire à un niveau inférieur.
 - L'utilisateur d'une ressource : activité en traitement par lots (BATCH) ou IOF, ou transaction TDS.
 - La durée de l'accès utilisateur à une ressource : période d'accès délimitée par les fonctions ASSIGN-DEASSIGN au niveau fichier, ou patrie de l'unité de consolidation au niveau page.
- Une base de données se compose de plusieurs aires. Lorsque plusieurs utilisateurs accèdent en même temps à des aires différentes on ne peut parler de partage de la base de données dans le sens où nous l'entendrons dans les paragraphes qui suivent.
- Il existe trois modes de partage au niveau fichier
 - Exclusif (EXCLUSIVE) : l'aire est réservée à un utilisateur qui peut y effectuer des opérations d'extraction (RETRIEVAL) ou de mise à jour (UPDATE).
 - Partagé (SHARED) : L'accès à l'aire est partagé par plusieurs utilisateurs qui peuvent y effectuer des opérations d'extraction.
 - Géré (MONITORED) : l'accès à l'aire est partagé par plusieurs utilisateurs qui peuvent y effectuer des opérations d'extraction et de mise à jour. La gestion des accès se fait au niveau page.

Lorsqu'un utilisateur a obtenu l'accès à une aire, il demeure responsable de cette ressource jusqu'à l'exécution d'une fonction DEASSIGN.

- Il existe trois modes de partage au niveau page (lorsque le mode d'accès au niveau fichier est MONITORED) :
 - Exclusif : la page est réservée à un seul utilisateur qui peut y effectuer des opérations d'extraction ou de mise à jour.
 - Partagé : l'accès à la page est partagé par plusieurs utilisateurs qui peuvent y effectuer des opérations d'extraction.
 - Statistiques (STATISTICAL) : la gestion des accès est temporairement invalidée. Tout utilisateur, dès qu'il en fait la demande, peut obtenir une copie d'une page disque.

Lorsqu'un utilisateur obtient l'accès à une page, il demeure responsable de cette ressource jusqu'à la fin de l'unité de consolidation.

7.4.2 Mode d'utilisation des aires

- Le paragraphe précédent nous a permis de considérer le partage du point de vue de la base de données. Ce paragraphe le décrit du point de vue de l'utilisateur.

Lorsqu'un utilisateur demande l'accès à une aire, il indique sous quel mode il compte l'utiliser. Le mode d'utilisation comprend le partage au niveau fichier (exclusif, partagé, géré) et le mode d'accès (extraction ou mise à jour). Lorsque le partage au niveau fichier se fait sous mode géré (MONITORED), partage et accès se font au niveau page.

Le Tableau 7-2 permet d'illustrer les différents modes d'utilisation et les cadres d'exploitation système pour lesquels ils peuvent être spécifiés.

- La première colonne indique les modes d'utilisation génériques tels qu'ils ont été définis ci-dessus.
- La seconde colonne rend compte des spécifications utilisées dans le programme DML. On peut voir que partage et accès au niveau fichier ne peuvent être spécifiés que dans une instruction READY.

La logique du programme doit être adaptée au mode d'utilisation sélectionné (découpage en unités de consolidation sous mode MONITORED).

- La troisième colonne indique les spécifications JCL dans une activité en traitement par lots et IOF.

Les paramètres SHARE et ACCESS de l'instruction ASSIGN définissent le partage et l'accès au niveau fichier. Le paramètre READLOCK de l'instruction DEFINE permet de spécifier le mode de partage des pages auxquelles l'utilisateur accède pour des opérations d'extraction. La réservation des pages auxquelles l'utilisateur a accès doit toujours s'effectuer sous mode exclusif.

Lorsqu'elle vérifie les autorisations d'accès à l'aire, la Gestion de fichiers ne prend en compte que l'instruction ASSIGN et non READY. C'est pourquoi l'instruction ASSIGN doit préciser le mode d'utilisation, soit le même que READY, soit un mode d'utilisation plus restrictif: exclusif (EXCLUSIVE) au lieu de partagé (SHARED) ou géré (MONITORED), partagé au lieu de géré, extraction (RETRIEVAL) au lieu de mise à jour (UPDATE). Les règles concernant les restrictions d'accès sont développées au paragraphes suivant.

Tableau 7-1. Contexte du mode d'utilisation des aires

MODE (USAGE)	D'UTILISATION GNERIQUE	TERMINOLOGIE FICHIER PRET	DML (READY)	TERMINOLOGIE DU T JCL ASSIGN JCL DEFINE	. PAR LOTS ET DE IOF (FICHIER) (PAGE)	TERMINOLOGIE JCL ASSIGN GENERATION	(FICHIER) TDS (PAGE)
Niveau fichier	Niveau page	Niveau fichier	Niveau page	Niveau fichier	Niveau page	Niveau fichier	Niveau page
EXCLUSIVE RETRIEVAL		EXCLUSIVE RETRIEVAL		SHARE = NORMAL ACCESS = SPREAD			
EXCLUSIVE UPDATE		EXCLUSIVE UPDATE		SHARE = NORMAL ACCES = WRITE			
SHARED RETRIEVAL		SHARED RETRIEVAL		SHARE = NORMAL ACCESS = READ			
MONITORED RETRIEVAL	EXCLUSIVE RETRIEVAL	MONITORED RETRIEVAL	ne peut être spécifié	SHARE = MONITOR ACCESS = READ	READLOCK = EXCL	SHARE = MONITOR ACCESS = {READ} {SPREAD} {ALLREAD}	implicitement EXCLUSIVE UPDATE
	SHARED RETRIEVAL		ne peut être spécifié		READLOCK = NORMAL		TRANSACT. SHARED READ
	STATISTIC RETRIEVAL		ne peut être spécifié		READLOCK = STAT		TRANSACT. SUPPRESS CONCURRE NT
MONITORED UPDATE	EXCLUSIVE RETRIEVAL	MONITORED UPDATE	ne peut être spécifié	SHARE = MONITOR ACCESS = WRITE	READLOCK = EXCL	SHARE = MONITOR ACCESS = {WRITE} {SPWRITE}	implicitement EXCLUSIVE RETRIEVAL
	EXCLUSIVE UPDATE		toujours EXCLUSIVE UPDATE		toujours EXCLUSIVE UPDATE		toujours EXCLUSIVE UPDATE
	SHARED RETRIEVAL		ne peut être spécifié		READLOCK = NORMAL		TRANSACT. SHARED READ

La nécessité de spécifier le mode d'utilisation par JCL présente des avantages :

- La gestion de travaux peut éviter les blocages au niveau fichier.
- L'information concernant le mode d'utilisation au niveau page peut être fournie.
- Tout programme dont la logique correspond au mode géré peut, par JCL, se voir attribuer le mode exclusif au niveau fichier. Ceci permet d'éviter la surcharge de la Gestion des accès au niveau page chaque fois que le programme peut se dérouler sous le mode exclusif.

La nécessité de spécifier le mode d'utilisation au niveau JCL présente des inconvénients :

- Obligation pour l'utilisateur de fournir deux fois la même information lorsque le mode d'utilisation est identique dans le programme et dans le JCL.
- La période d'exécution comprise entre ASSIGN et DEASSIGN peut elle-même se découper en plusieurs sessions comprises entre READY et FINISH, chacune ayant un mode d'utilisation différent. Il y a alors obligation pour les instructions ASSIGN et DEFINE de fournir le plus restrictif de ces modes d'utilisation.

Exécution du programme utilisateur

La quatrième colonne fournit les spécifications du mode d'utilisation dans un cadre d'exploitation TDS.

Une complexité supplémentaire résulte du fait que :

- L'activité TDS est considérée en tant que telle comme un utilisateur par rapport aux activités en traitement par lots, IOF ou aux autres activités TDS.
- L'activité TDS correspond en fait à plusieurs utilisateurs, autant qu'il y a de transactions en cours.

Le mode d'utilisation au niveau fichier est spécifié dans le JCL de l'activité TDS et s'applique à toutes les transactions.

Le mode exclusif est sans objet puisque les transactions sont normalement concurrentes.

Le mode partagé n'est pas pris en compte pour les fichiers contrôlés par TDS.

Sous mode géré (MONITORED), l'aire peut faire l'objet :

- d'une réservation en mode exclusif par rapport aux autres activités mais d'un partage au niveau des transactions. Ce niveau de finesse est obtenu dans le partage par les valeurs affectées aux paramètres ACCESS (SPRED, SPWRITE) ;
- d'un partage par toutes les autres activités aussi bien que par les transactions (READ, WRITE).

Le mode d'utilisation au niveau page se définit au moment de la génération de TDS et peut être différent entre toutes les transactions. Le paramètre READLOCK de l'instruction DEFINE ne doit pas avoir été spécifié.

7.5 REGLES CONCERNANT LES RESTRICTIONS DE READY PAR ASSIGN

L'instruction JCL ASSIGN peut restreindre la portée de l'instruction READY du programme. Les règles de restriction sont renforcées aussi bien par la Gestion de fichiers que par le SGBD.

Le Tableau 7-2 indique, par rapport au paramètres USAGE-MODE de READY, quelles spécifications SHARE et ACCESS de ASSIGN peuvent être fournies dans le JCL.

Lorsque la combinaison de paramètres est interdite, le point d'intersection entre ligne et colonne porte un tiret.

Lorsque la combinaison de paramètres est autorisée, le point d'intersection indique le mode d'utilisation qui est finalement retenu.

Pour plus de clarté dans le tableau, les valeurs SPREAD, ALLREAD et SPWRITE n'ont pas été mentionnées.

Lorsque l'utilisateur désire appliquer réellement le mode de partage EXCLUSIVE RETRIEVAL spécifié dans READY, il doit indiquer SHARE=NORMAL et ACCESS=WRITE/SPREAD dans ASSIGN.

Tableau 7-2. Mode d'utilisation final (au niveau fichier) après annulation de READY par ASSIGN

Instruction READY du programme		USAGE MODE				
Instruction JCL	ASSIGN	EXCLUSIVE		SHARED	MONITORED	
		RETRIEVAL	UPDATE	RETRIEVAL	RETRIEVAL	UPDATE
NORMAL	READ	-	-	SHARED RETRIEVAL	SHARED RETRIEVAL	-
	WRITE	EXCLUSIVE RETRIEVAL	EXCLUSIVE UPDATE	EXCLUSIVE RETRIEVAL	EXCLUSIVE RETRIEVAL	EXCLUSIVE UPDATE
MONITOR	READ	-	-	-	MONITORED RETRIEVAL	-
	WRITE	-	-	-	MONITORED RETRIEVAL	MONITORED UPDATE
ONEWRITE	READ	-	-	-	SHARED RETRIEVAL	-
	WRITE	EXCLUSIVE RETRIEVAL	EXCLUSIVE UPDATE	SHARED RETRIEVAL	SHARED RETRIEVAL	EXCLUSIVE UPDATE
FREE	READ	-	-	-	-	-
	WRITE	-	-	-	-	-

7.5.1 Protection des aires par les journaux

- La restauration à un état stable de tout ou partie d'une aire est garantie par deux modes de protection. Cet état correspond à un point de cohérence: début d'activité, point de reprise, point de consolidation et fin d'activité.

Le point de cohérence est dit local s'il se rapporte à la situation d'un utilisateur déterminé. D'autres utilisateurs peuvent être en train de travailler sur d'autres parties de l'aire.

Le point de cohérence est dit global s'il reflète la situation de tous les utilisateurs : chaque utilisateur a atteint un niveau de cohérence local et la base de données est inactive.

La protection d'aire présuppose que les pages modifiées par un utilisateur entre deux points de cohérence sont sous mode exclusif.

- Le journal AVANT (BEFORE), en enregistrant l'état des pages avant leur modification, permet au système de restaurer ces pages à l'état dans lequel elles se trouvaient au point de cohérence précédent.

Cette méthode s'utilise pour la reprise sur erreur du programme ou à la suite d'une défaillance du système ou encore pour résoudre les situations de blocage.

- Le journal APRES (AFTER), en enregistrant l'état des pages après leur modification, permet au système de relancer l'activité à partir d'une copie sauvegardée de l'aire, à un point de cohérence global et de réintroduire toutes les modifications qui ont été apportées jusqu'à l'apparition du point de cohérence suivant.

Cette méthode s'utilise pour la reprise sur défaillance du matériel ayant causé la destruction physique de la base de données. Il est donc conseillé de ne pas faire résider le journal APRES sur les mêmes volumes que ceux de la base de données.

Dans un cadre d'exploitation TDS, le journal APRES peut aussi s'utiliser conjointement au mécanisme de mise à jour différée pour remplacer la protection fournie par le journal AVANT, tout autant qu'il existe suffisamment d'espace tampon pour recueillir toutes les pages modifiées au cours de l'unité de consolidation.

- L'utilisation des journaux AVANT et APRES est facultative ou obligatoire suivant le mode d'utilisation de l'aire et le cadre d'exploitation.

Si le mode d'accès est extraction (RETRIEVAL) il n'y a pas utilisation des journaux.

Si le mode d'utilisation pour les activités en traitement par lots et IOF est défini par EXCLUSIVE UPDATE, l'utilisation des deux journaux est facultatives.

Si le mode d'utilisation est défini par MONITORED UPDATE pour les activités en traitement par lots et IOF, le journal AVANT est obligatoire et le journal APRES est facultatif.

Si le mode d'utilisation d'une activité TDS est défini par MONITORED UPDATE, il y a obligation de spécifier, soit le journal AVANT, soit le journal APRES accompagné du mécanisme de mise à jour différée (DEFERRED UPDATE). Lorsque l'option journal AVANT a été choisie, le journal APRES est facultatif.

Si plusieurs activités mettent à jour la même base de données, soit séquentiellement, soit concurremment, il est conseillé aux utilisateurs de prévoir une protection d'aire parfaitement cohérente. Cette remarque concerne particulièrement l'utilisation du journal APRES lorsqu'il sert à la protection contre les défaillances des supports.

L'Administration de la base de données a la possibilité d'imposer un niveau de protection uniforme et permanent en insérant les options JOURNAL dans le catalogue.

7.5.2 Cohérence des niveaux de protection d'aire pour le même utilisateur en traitement par lots et IOF

S'il est important que l'accès à une aire donnée par plusieurs utilisateurs bénéficie d'un niveau de protection uniforme, il est aussi important d'assurer une cohérence des niveaux pour un même utilisateur qui accède à différentes aires de la base de données.

Le système ne procède à aucune vérification de la cohérence des niveaux pour des fichiers différents. Comme sous IDS/II les aires font partie du même ensemble logique, le SGBD peut au moins vérifier, au cas où le retour à un point de cohérence serait nécessaire, s'il est possible d'effectuer une restauration non actualisée (ROLLBACK) de toutes les aires couramment ouvertes.

Le Tableau 7-3 illustre les contraintes qui pèsent sur le mode d'utilisation des différentes aires ouvertes pour mise à jour par le même utilisateur dans le cadre du traitement par lots ou sous IOF. Dans un cadre d'exploitation TDS, les aires doivent se trouver obligatoirement sous mode MONITORED.

Dans tous les cas de figures, l'Administrateur de la base de données a la possibilité de renforcer le niveau de protection entre les aires en insérant les options JOURNAL dans le catalogue.

Tableau 7-3. Cohérence des niveaux de protection des aires pour un même utilisateur en modes par lots et IOF

Aires à l'état prêt (READY)	EXCLUSIVE UPDATE		MONITORED UPDATE
	Demande READY pour une nouvelle aire	Pas BEFORE	BEFORE
EXCLUSIVE UPDATE	Pas BEFORE	Oui (Restauration non actualisée impossible)	-
	BEFORE	-	-
MONITORED UPDATE	Toujours BEFORE	-	Oui (Restauration non actualisée possible)

7.6 COMMANDES UTILISABLES A L'EXECUTION

Les règles de syntaxe et de représentation sont indiquées au chapitre 3, comme pour le langage de commande du processeur DDLPROC.

7.6.1 Mots réservés

ABNORMAL	MODLAB
ACCEPT	NAME (NAMES)
ALL	NO
ANY	NON-NULL
ARE (IS)	NOT
AREA (REALM)	NUMBER
AT	OCCURRENCE
BUFFER (BUFFERS)	OF
CANCEL	OPEN
CHECK	OUTPUT
CLEANPOINT	PAGE
COMMITMENT	POINTERS
CONNECT	POOL
CURRENCIES	PRINT
CURRENT	PROGRAM
DB-REGISTERS	PUTPAGE
DB-STATUS	READY
DDL1B1	REALM (AREA)
DDL1B2	RECORD
DDL1B3	RELPAGE
DISCONNECT	RESTART
DUMP	ROLLBACK
ERASE	SAVE
EXCEPT	SCHEMA
EXTEND	SEGMENT (SEGMENTS)
FIELDS	SET
FILE	SOURCE
FIND	STARTDBS
FINISH	STATE
FOR	STATEMENT (STATEMENTS)
GET	STATISTICS
GETLAB	STORE
GETPAGE	STRUCTURE
GETPTR	SWITCH
HEXADECIMAL (HEXA)	SYNCHRO
IAC	SYSTEM
IDSTRACE	TERMDBS
IF	THRESHOLD
IGNORE	THROUGH (THRU)
IN (OF)	TIMING
INCONSISTENT	TRACE
INPUT	TRANSIENT
INTERNAL	UNLOAD
IS (ARE)	WARNING
I-O	WHEN
LINE	WITH
MODIFY	

7.6.2 Commandes

Il existe onze commandes. Le groupe de commandes qui appartient à un schéma déterminé est identifié par la première commande qui doit obligatoirement être SCHEMA.

Les commandes utilisables à l'exécution sont listées ci-dessous :

```
[
[ Commande SCHEMA ]
[
[ [Commande IGNORE TRANSIENT] ]
[
[ [Commande POOL] ]
[
[ [Commande BUFFERS] ]
[
[ [Commande SAVE CURRENCIES] ]
[
[ [Commande SYNCHRO] ] ...
[
[ [Commande NO WARNING] ]
[
[ [Commande STATISTICS] ]
[
[ [Commande TRACE] ]
[
[ [Commande CHECK PAGE] ]
[
[ [Commande INTERNAL FILE NAME] ]
[
[ ]
```

Toutes ces commandes sont décrites individuellement dans les pages qui suivent.

7.6.2.1 Commande SCHEMA

Fonction

Permet d'identifier le schéma auquel s'applique les commandes suivantes et d'indiquer la manière dont ce schéma sera chargé.

Format

```

SCHEMA NAME IS nom-schéma [ [ {IN} {nfe-bibliothèque} ]
                             [ {ON} {DDLIB1} ] ]
                             [ {DDLIB2} ] ]
                             [ {DDLIB3} ] ]
    [FOR PROGRAM nom-programme]
    [NUMBER OF SEGMENTS IS entier] .
    
```

Règles

1. Chaque fois qu'un programme principal lance une session IDS le SGBD appelle le fichier des commandes utilisables à l'exécution pour trouver une commande SCHEMA dont le nom-schéma correspond à celui qui est indiqué pour le programme.
2. Si la clause PROGRAM n'est pas spécifiée, la recherche de la commande SCHEMA s'arrête dès que le nom de schéma correct a été identifié.
3. Si la clause PROGRAM est spécifiée, le SGBD vérifie en outre que le nom-programme fourni correspond à celui qui doit lancer la session IDS. Si les noms ne concordent pas, le SGBD poursuit sa recherche.

Cette option est nécessaire pour différencier deux schémas homonymes utilisés par deux programmes de l'activité.

4. Si la recherche de la commande SCHEMA ne réussit pas, toutes les options implicites sont prises en compte.
5. Si la qualification de bibliothèque n'a pas été fournie, le schéma à charger est recherché dans les bibliothèques affectées, dans l'ordre: DDLIB1, DDLIB2, DDLIB3.
6. S'il existe une qualification de bibliothèque, le schéma à charger est extrait de la bibliothèque spécifiée.
7. Si la clause SEGMENTS n'est pas spécifiée, le schéma est chargé dans le nombre de segments qui est indiqué dans l'état PRINT STORAGE DESCRIPTION du processeur DDLPROC.
8. Si la clause SEGMENTS est spécifiée, le schéma est chargé dans le nombre de segments qui a été indiqué, sous réserve qu'il soit inférieur à celui qui a été fourni implicitement par DDLPROC.

Cette option se révèle particulièrement utile lorsque le nombre de segments libres constitue une ressource limitée de l'activité.

Dans un cadre d'exploitation TDS, il est recommandé de spécifier un segment.

7.6.2.2 Commande IGNORE TRANSIENT

Fonction

Permet d'imposer au système la non prise en compte de l'état "transitoire" ou "incohérent" d'un ou plusieurs domaines au moment de leur mise à l'état prêt.

Format

```
IGNORE      TRANSIENT          STATE OF {nom-domaine}..._
            INCONSISTENT
```

Règles

1. Si TRANSIENT est spécifié, les domaines de la liste sont rendus accessibles au programme même si au moment de passer à l'état prêt ils se trouvent à l'état transitoire (TRANSIENT).

Cette situation apparaît lorsqu'une activité antérieure met à jour, en mode exclusif et sans protection du journal AVANT, des domaines qui, pour des raisons normales ou anormales, n'ont pu être traités entièrement.

L'état transitoire indique qu'une partie de la base de données est présumée incohérente. La vérification de cet état représente une sécurité au moment où les domaines passent à l'état prêt (READY), de manière à en interdire éventuellement l'utilisation.

La reprise normale à partir d'une telle situation est la restauration des domaines à partir de copies sauvegardées. Une autre manière consiste à lancer l'utilitaire DB-UTILITY de manière à s'assurer que la base de données est dans un état cohérent, de redresser d'éventuelles erreurs et d'utiliser la clause RESET TRANSIENT de la commande PATCH LABEL.

Toutefois, si l'utilisateur est certain que l'activité incriminée a laissé les domaines dans un état cohérent, aussi bien en ce qui concerne les pointeurs IDS que les données utilisateurs, il peut émettre la commande IGNORE TRANSIENT. Le fait d'imposer la non prise en compte de l'état transitoire est enregistré dans le label IDS.

2. Si INCONSISTENT est spécifié, les domaines de la liste sont rendus disponibles au programme même si leur état "incohérent" est reconnu au moment de la mise à l'état prêt.

L'état d'incohérence (INCONSISTENT) apparaît lorsqu'une session IDS antérieure a détecté et enregistré dans le label IDS une incohérence de la base de données. Cet état d'incohérence est enregistré même si les domaines ont été ouverts en mode extraction (RETRIEVAL), sauf si le mode de partage est SHARE=MONITOR avec READLOCK=STAT qui ferait apparaître une incohérence fictive. La vérification de l'état cohérent des domaines constitue une mesure de sécurité au moment du passage à l'état prêt.

Exécution du programme utilisateur

La manière normale de sortir d'une telle situation est de restaurer les domaines à leur état cohérent au moyen de copies de sauvegarde. On peut également corriger les erreurs relevées sur la partie incohérente de la base de données en utilisant la commande PATCH de l'utilitaire DBUTILITY (après avoir procédé à une analyse du message d'erreur du SGBD qui rend compte de cette incohérence) et valider la correction d'erreur. Après quoi on utilise la clause RESET INCONSISTENT de la commande PATCH LABEL.

Toutefois, si l'utilisateur est certain que le programme à exécuter n'utilisera pas cette partie "incohérente" de la base de données il peut utiliser une commande IGNORE INCONSISTENT. La non prise en compte de l'état incohérent est enregistré dans le label IDS.

3. Si la commande IGNORE est omise, toute tentative de passer à l'état prêt des domaines qui se trouvent dans un état transitoire ou incohérent provoque l'arrêt prématuré des programmes.
4. Si la commande IGNORE est utilisée, l'état transitoire et/ou incohérent des aires concernées n'est pas pris en compte mais on ne reviendra pas pour autant à l'état initial.

7.6.2.3 Commande BUFFER POOL

Fonction

Permet d'indiquer si les aires de la base de données partagent ou non un pool de tampons.

Format

```
BUFFER POOL NAME IS nom-pool-tampons_.
```

Règles de syntaxe

1. Nom-pool-tampons ne peut dépasser quatre caractères.
2. Si on utilise un nom à blancs, il doit s'écrire ' ', soit des espaces encadrés par des apostrophes.
3. Les noms qui ne sont pas à blancs peuvent contenir des lettres, des chiffres, le signe moins et le caractère de soulignement. Ils doivent être encadrés par des apostrophes s'ils ne se conforment pas aux règles de représentation des noms DDL.

Règles générales

1. Si la commande POOL est omise c'est le nom DMCL qui est pris comme valeur implicite. Si la commande POOL est utilisée elle remplace la spécification DMCL.
2. Lorsque on utilise l'option "nom à blancs", il n'existe pas de pool de tampons et c'est la commande NUMBER OF BUFFERS qui indique le nombre de tampons réservé pour chaque aire. Autrement dit chaque aire utilise son propre jeu de tampons.
3. La spécification d'un nom sans blancs indique l'existence d'un pool de tampons et la commande NUMBER OF BUFFERS donne le nombre total de tampons que se partagent toutes les aires.
4. Le but du travail en pool est de réduire la quantité d'espace tampon nécessaire lorsque plusieurs aires se trouvent à l'état prêt. Lorsqu'un programme DML accède à une aire pour un certain laps de temps :
 - Il ne peut, s'il n'existe pas de pool, utiliser plus d'espace tampon que celui qui a été réservé pour cette aire. L'espace réservé pour les autres aires ne lui est pas accessible.
 - Il peut, s'il existe un pool, utiliser la totalité de l'espace tampon disponible pour cette aire. Lorsque le programme commence à travailler avec une autre aire, la totalité de l'espace est affectée aux tampons de cette nouvelle aire.
5. Cette commande n'est pas prise en compte par TDS.

7.6.2.4 Commande NUMBER OF BUFFERS

Fonction

Permet de spécifier le nombre de tampons réservé pour chaque aire ou pour le pool.

Format

```
NUMBER OF BUFFERS IS entier .
```

Règles

1. Si la commande BUFFERS est omise, c'est le nombre de tampons fourni par le DMCL qui est pris comme valeur implicite. Lorsque la commande BUFFER est spécifiée, elle annule les spécifications DMCL.
2. La valeur minimum d'entier est fixée à 3.
3. Si le nom de pool de tampons n'est pas un nom à blancs, la valeur indiquée dans entier représente le nombre de tampons à partager entre toutes les aires. Lorsque plusieurs aires disposent de tailles de pages différentes, c'est la valeur la plus élevée qui est choisie comme taille de tampon.

Exécution du programme utilisateur

4. Si le nom de pool de tampons est un nom à blancs, la valeur spécifiée dans entier est le nombre de tampons à réserver pour chaque aire.
5. La détermination du nombre de tampons à choisir au moment de l'exécution constitue un compromis entre les besoins en espace tampon et les performances requises.

Lorsqu'un programme travaille pour une période limitée de temps sur un ensemble de pages groupées, le nombre de tampons peut être ajusté pour convenir à toutes ces pages.

Si le mode d'accès d'un programme aux pages est entièrement sélectif, le fait d'augmenter le nombre de tampons ne permet pas d'améliorer la performance globale.

6. Cette commande n'est pas prise en compte pas TDS.

7.6.2.5 Commande SAVE CURRENCIES

Fonction

Permet, dans les cadres du traitement par lots et IOF, de demander que les indicateurs de point courant ne soient pas mis à NUL lors des passages successifs aux points de consolidation.

Format

```
SAVE CURRENCIES AT COMMITMENT .
```

Règles

1. Si la commande SAVE CURRENCIES est omise, les indicateurs de point courant sont mis à NUL à chaque passage à un point de consolidation, ce qui provoque le déverrouillage par l'unité d'exécution des pages de base de données qu'englobe le point de consolidation.
2. Si la commande SAVE CURRENCIES est spécifiée, les indicateurs de point courant conservent leur valeur au passage des points de consolidation. Toutefois, les pages de la base de données correspondant à ces unités de consolidation sont déverrouillées par l'unité d'exécution au moment du passage au point de consolidation.

Il est conseillé de n'utiliser cette option que si des protocoles d'exploitation existent entre les applications de manière à s'assurer que les articles référencés par les indicateurs de point courant d'une unité d'exécution ne seront pas effacés ou remplacés par des articles d'unités d'exécution concurrentes.

Si une telle précaution n'a pas été prise, le SGBD peut provoquer l'arrêt prématuré de l'unité d'exécution dès qu'il relève une incohérence entre la base de données et les indicateurs de point courant.

3. Cette commande n'est pas prise en compte par TDS. En effet, dans ce cadre d'exploitation, les indicateurs de point courant sont toujours mis à NUL et les pages sont déverrouillées au point de consolidation.

Si la clé-base-de-données d'un article doit être intégralement conservée tout au long du déroulement des unités de consolidation, elle doit faire l'objet d'une sauvegarde dans la TRANSACTION STORAGE. **Error! Reference source not found.** au moyen d'une instruction ACCEPT avant la fin d'exécution de l'unité de consolidation, puis d'une restauration au moyen de l'instruction FIND DB-KEY dans l'unité de consolidation suivante.

4. Cette commande ne concerne pas les points de reprise. Dans un environnement non concurrent, les indicateurs de point courant ne subissent aucun changement au passage des points de reprise.

7.6.2.6 Commande SYNCHRO

Fonction

Permet d'invalider la lecture asynchrone de page au cours de la recherche séquentielle dans une aire.

Format

<u>SYNCHRO</u> .

Règles

1. Si cette commande est omise, chaque instruction FIND... WITHIN nom-domaine déclenche le mécanisme de recherche anticipée qui provoque le transfert en tampon de la page qui suit (ou précède) la page en cours. Ce transfert se produit d'une manière non synchrone pendant que le SGBD ou le programme traite la page en cours. Ce mécanisme permet donc de réduire le temps de traitement.
2. Si cette commande est spécifiée, le mécanisme de recherche anticipée est invalidé.

Exécution du programme utilisateur

3. L'usage de cette commande est utile dans le cas où le tampon supplémentaire que nécessite la lecture non synchronisée doit être libérée avant que le contenu de la page concernée ait été réellement utilisé.

Prenons l'exemple d'un programme, utilisant un pool de quatre tampons, qui poursuit une recherche séquentielle dans une aire et, pour chaque article trouvé, recherche un type d'ensemble étalé sur trois pages dans une autre aire. Le nombre total de pages englobées dans le processus de recherche du type d'ensemble est cinq, soit quatre pour l'article maître et ses détails et un pour la page qui doit faire l'objet d'une lecture anticipée. Comme il n'existe que quatre tampons, la page qui a fait l'objet d'une lecture anticipée et qui contient l'article maître sera libérée au cours du traitement des articles détails et devra donc faire l'objet d'une nouvelle lecture à la prochaine apparition, provoquant ainsi une perte de performance. Lorsque cette situation se produit il est conseillé d'utiliser la commande SYNCHRO ou d'augmenter le nombre de tampons, ou encore de supprimer le pool de manière à isoler les tampons des deux aires.

7.6.2.7 Commande NO WARNING

Fonction

Permet d'éviter que le SGBD envoie les messages d'avertissement à l'historique JOR.

Format

<code><u>NO WARNING</u> .</code>

Règles

1. Si la commande NO WARNING est omise, les messages d'avertissement enregistrés par le SGBD dans le registre DB-DETAILED-STATUS sont envoyés à l'historique des travaux JOR.
2. Si la commande NO WARNING n'est pas spécifiée, les messages d'avertissement ne sont pas envoyés au JOR.

7.6.2.8 Commande STATISTICS

Fonction

Permet de demander au SGBD d'enregistrer des statistiques et de chronométrer les opérations au cours d'une session IDS/II.

Format

```
STATISTICS [WITH TIMING] .
```

Règles

1. Si la commande STATISTICS est spécifiée sans option TIMING, les informations suivantes sont transmises au JOR à la fin de la session IDS :
 - Nombre de tampons utilisés pour le pool ou pour chaque aire.
 - Nombre d'appels du programme à chaque catégorie de fonction du SGBD : ACCEP, CONNECT, ...
 - Nombre moyen de fonctions SYSTEM appelées par chaque catégorie de fonction SGBD. Les fonctions SYSTEM en question sont limitées au système UFAS et à la gestion des accès GAC.
 - Nombre moyen de transferts de page effectués par chaque catégorie de fonction du SGBD, à l'exception des entrées-sorties portant sur les journaux.
 - Nombre d'appels à chaque catégorie de sous-fonction du SGBD. On entend par sous-fonction du SGBD, DIRECT PLACEMENT (pour le rangement d'un article à accès DIRECT ou VIA), CALC PLACEMENT, SET SELECTION ET SET INSERTION.
 - Nombre moyen de fonctions SYSTEM appelées pour chaque catégorie de sous-fonction du SGBD.
 - Nombre moyen de transferts de page effectués pour chaque catégorie de sous-fonction du SGBD.
 - Nombre d'appels à chaque catégorie de la fonction SYSTEM.
 - Nombre moyen de transferts de page effectués pour chaque catégorie de la fonction SYSTEM.
 - Nombre total d'appels aux fonctions du SGBD.
 - Nombre total d'appels aux fonctions SYSTEM.
 - Nombre total de transferts de page.
2. Si la commande STATISTICS est accompagnée de l'option TIMING, les informations suivantes sont ajoutées :
 - Temps moyen écoulé pour chaque catégorie de fonction du SGBD.
 - Temps moyen processeur pour chaque catégorie de fonction du SGBD. Ce temps comprend le temps processeur écoulé dans les appels aux fonctions du système d'exploitation.
 - Total du temps consommé par le SGBD.
 - Total du temps de traitement par le SGBD.
3. Si la commande STATISTICS est omise, aucune statistique n'est enregistrée pendant la session IDS.

Exécution du programme utilisateur

4. La Figure 7-2 fournit un exemple de statistiques d'exécution d'une session avec option TIMING.

- Remarques :**
1. Les valeurs non entières sont arrondies à la valeur imprimable la plus proche. Par exemple, un nombre moyen d'E-S indiqué par "0.02" signifie que la valeur v réelle s'exprime selon la formule $0.015 < v \leq 0.025$.
 2. Le nombre total d'E-S rapporté par le SGBD peut présenter une légère différence avec le nombre d'E-S de l'instruction CONNECT affiché dans le JOR. Cette différence s'explique par le fait que le SGBD ne saisit pas les E-S des instructions OPEN et CLOSE.
 3. Les statistiques SYSTEM servent au Service de Maintenance lorsque des problèmes de performance nécessitent une analyse.
 4. Pour l'interprétation des statistiques, se reporter au dernier paragraphe du présent chapitre.

Manuel de référence IDS/II

```

**** STATISTICS * V40.0 * PROGRAM: TG02 * SCHEMA: TG02-SH ****
BUFFER POOL: YES * NUMBER OF BUFFERS: 4
DBCS STATISTICS:
*****
* DBCS FUNCTION      * NUMBER * AVERAGE NB OF * AVERAGE NB * AV. ELAPSED * AV.
* PROCESS           *        *                 *             * TIME (MS)   *
* CATEGORY          * OF CALLS * SYSTEM CALLS * OF I/O      *
* TIME (MS)         *         *              *            *
*-----*-----*-----*-----*-----*
* ACCEPT            *      431 *      0.00 *      0.00 *      0.1 *
* 0.1 *            *         *         *         *         *
* FIND DIRECT       *      184 *      0.82 *      0.16 *      21.3 *
* 1.5 *            *         *         *         *         *
* FIND CURRENT      *       26 *      0.31 *      0.00 *      10.9 *
* 0.8 *            *         *         *         *         *
* FIND F/L/I AREA  *        2 *     58.00 *     20.00 *     533.5 *
* 94.6 *           *         *         *         *         *
* FIND N/P SET      *      396 *      0.76 *      0.14 *      20.9 *
* 1.7 *            *         *         *         *         *
* FIND SELECT USING *       33 *      2.18 *      0.64 *      71.2 *
* 5.3 *            *         *         *         *         *
* FINISH            *        1 *      1.00 *      -.- *     948.0 *
* 47.1 *           *         *         *         *         *
* GET               *      264 *      0.00 *      0.00 *       0.2 *
* 0.2 *            *         *         *         *         *
* MODIFY MEMBERSHIP *       33 *      8.30 *      1.00 *     143.1 *
* 11.1 *           *         *         *         *         *
* READY             *        1 *      0.00 *      -.- *    2165.3 *
* 234.9 *          *         *         *         *         *
* STORE             *      155 *      6.34 *      1.23 *     128.2 *
* 10.3 *           *         *         *         *         *
*****
* DBCS SUBFUNCTION  * NUMBER * AVERAGE NB OF * AVERAGE NB *
* CATEGORY          * OF CALLS * SYSTEM CALLS * OF I/O      *
*-----*-----*-----*-----*
* SET SELECTION     *      200 *      1.53 *      0.29 *
* SET INSERTION     *      167 *      2.29 *      0.20 *
* DIRECT PLACEMENT  *      125 *      2.00 *      0.64 *
* CALC PLACEMENT    *       30 *      2.70 *      1.63 *
*****
SYSTEM STATISTICS:
*****
* SYSTEM FUNCTION   * NUMBER * AVERAGE NB *
* CATEGORY          * OF CALLS * OF I/O      *
*-----*-----*-----*
* CHECKCI          *       36 *      0.00 *
* FREECI           *      734 *      0.00 *
* GETCI            *      734 *      0.51 *
* SETCI            *      401 *      0.00 *
*****
TOTAL NUMBER OF DBCS CALLS : 1526
TOTAL NUMBER OF SYSTEM CALLS : 1905
TOTAL NUMBER OF I-C : 371
TOTAL DBCS ELAPSED TIME (MN) : 0.729
TOTAL DBCS PROCESS TIME (MN) : 0.062

```

Figure 7-2. Statistiques de session

7.6.2.9 Commande TRACE

Permet de lancer le jalonnement des fonctions DML.

La fonction de jalonnement comprend :

- "Jalonnement externe", destiné à aider l'utilisateur à suivre l'exécution d'un programme DML. Il peut choisir les types de fonction DML à jalonner, limiter leur jalonnement effectif à certaines conditions et choisir la quantité d'information à imprimer.
- "Jalonnement interne", destiné à aider le Service de Maintenance pour vérifier l'interface IDS-UFAS-GAC ou pour analyser les problèmes de performance. L'utilisateur n'est pas concerné par le jalonnement interne.

Format général

```
clause TRACE  
  
    [clause WHEN RECORD]  
  
    [clause WHEN AREA]  
  
    [clause WHEN DB-STATUS]  
  
    [clause WHEN THRESHOLD]  
  
    [clause WHEN PROGRAM] ...  
  
    [clause PRINT]  
  
    [clause IDSTRACE]  
  
    [clause OPEN]  
  
    [clause IAC] .
```

Règles

1. Chaque clause est décrite individuelle dans les pages qui suivent.
2. Lorsque le "jalonnement interne" n'est pas nécessaire, la condition de jalonnement s'écrit :

```
TRACE fonction liste-de-conditions  
  
AND RECORD-condition  
  
AND REALM-condition  
  
AND DB-STATUS-condition  
  
AND THRESHOLD-condition  
  
AND (PROGRAM-1-condition [OR PROGRAM-2-condition]...)
```


3. Lorsque le "jalonnement interne" est nécessaire, la condition de jalonnement s'écrit :

```
TRACE fonction liste-de-conditions
```

```
AND (PROGRAM-1-condition [OR PROGRAM-2-condition] ...)
```

cette représentation différente est nécessaire car les autres conditions ne peuvent être vérifiées avant la fin de la fonction DML elle-même.

7.6.2.10 Clause TRACE

Fonction

Permet de spécifier les fonctions DML à jalonner.

Format

```
TRACE { ALL }
      { ACCEPT }
      { CONNECT }
      { DISCONNECT }
      { ERASE }
      { FIND }
      { [ { ANY EXCEPT } ] GET }
      { [ { NOT } ] IF }
      { MODIFY }
      { READY }
      { STARTDBS }
      { STORE }
      { TERMDBS }
      } STATEMENTS
```

Règles

- Si ALL est spécifié, toutes les fonctions DML font l'objet d'un jalonnement. Ces fonctions comprennent celles qui ont été mentionnées dans les instructions DML proprement dites ainsi que deux autres fonctions :
 - STARTDBS (Lancement de Session de Base de données) constitue la fonction d'initialisation déclenchée par la première instruction READY d'une session IDS.
 - TERMDBS (Fin de Session de Base de données) est la fonction de fin d'exploitation de la base de données déclenchée par la dernière instruction FINISH d'une session IDS.
- La fonction IF DML est le nom générique IDS/II pour la fonction "condition-base-de-données".
- Si une liste des fonctions DML est spécifiée et que NOT (ou ANY EXCEPT) n'est pas spécifiée, seules les fonctions de la liste font l'objet d'un jalonnement.

Exécution du programme utilisateur

4. Si une liste des fonctions DML est demandée et que NOT (ou ANY EXCEPT) est mentionné, toutes les fonctions autres que celles de la liste font l'objet d'un jalonnement.

7.6.2.11 Clause WHEN RECORD

Fonction

Permet de limiter le jalonnement aux seules opérations qui ont pour résultat d'introduire l'article courant de l'unité d'exécution dans une liste de types d'articles.

Format

```
WHEN RECORD NAME IS [NOT] {nom-article} ...
```

Règles

1. Si NOT est omis, la liste est constituée des types d'article spécifiés dans cette clause.
2. Si NOT est utilisé, la liste est constituée de tous les types d'article du schéma à l'exception de ceux qui sont spécifiés dans cette clause.
3. Lorsque cette clause est omise, toutes les opérations font l'objet d'un jalonnement quel que soit le type de l'article courant de l'unité d'exécution.

7.6.2.12 Clause WHEN AREA

Fonction

Permet de limiter le jalonnement aux seules opérations qui ont pour résultat de rattacher l'aire à laquelle appartient l'article courant de l'unité d'exécution à une liste d'aires.

Format

```
WHEN {AREA}  
     {REALM} NAME IS [NOT] {nom-aire} ...
```

Règles

1. Si NOT est omis, la liste est constituée de toutes les aires spécifiées dans cette clause.
2. Si NOT est utilisé, la liste est constituée de toutes les aires du schéma à l'exception de celles qui sont spécifiées dans cette clause.
3. Si cette clause est omise, toutes les opérations font l'objet d'un jalonnement quelle que soit l'aire à laquelle appartient l'article courant de l'unité d'exécution.

7.6.2.13 Clause WHEN DB-STATUS

Fonction

Permet de limiter le jalonnement aux opérations qui ont pour résultat l'affichage de certaines valeurs non nulles dans le registre DB-STATUS.

```
WHEN DB-STATUS IS {NON-NULL}  
                  {ABNORMAL}
```

Règles

1. Si NON-NULL est spécifié, les valeurs enregistrées dans DB-STATUS sont toutes celles qui sont différentes de "0000000".
2. Si ABNORMAL est spécifié, les valeurs enregistrées dans DB-STATUS sont celles qui sont différentes de "0000000", de "0502100" (fin d'ensemble ou fin de domaine) et de "0502400" (article introuvable).
3. Lorsque cette clause est omise, les opérations font l'objet d'un jalonnement indépendamment des valeurs enregistrées dans le registre DB-STATUS.

7.6.2.14 Clause WHEN THRESHOLD

Fonction

Permet de limiter le jalonnement aux seules opérations pour lesquelles un nombre minimum d'entrées-sorties physiques a été défini par l'utilisateur (transferts de page).

Format

```
WHEN THRESHOLD IS entier I-O
```

Règles

1. "Entier" spécifie le seuil minimum d'E-S.
2. Si cette clause est omise, le jalonnement de la fonction s'effectue quel que soit le nombre des entrées-sorties physiques.

7.6.2.15 Clause WHEN PROGRAM

Fonction

Permet de limiter le jalonnement aux seules apparitions des instructions DML référencées par numéros de ligne origine de programmes spécifiés.

Format

```
WHEN PROGRAM NAME IS nom-programme
```

```
[ SOURCE LINE NUMBER IS entier-1 [THRU entier-2] ]  
[ [ OCCURRENCE IS entier-3 [THRU entier-4]] ]
```

Règles

1. Lorsqu'ils sont spécifiés, "entier-1" doit être plus petit que "entier-2" et "entier-3" doit être plus petit que "entier-4".
2. Les numéros de lignes spécifiées ("entier-1", "entier-2") indiquent les numéros internes de ligne affectés par le compilateur COBOL à la compilation du programme mentionné dans cette clause.
3. Si la locution SOURCE LINE n'est pas spécifiée, le jalonnement s'effectue au moment où la fonction DML apparaît dans le programme spécifié.

4. Si la locution SOURCE LINE est spécifiée sans option OCCURRENCE, Le jalonnement s'effectue au moment où la fonction DML apparaît dans le programme, au(x) numéro(s) de ligne :
 - Indiqué par entier-1, si THRU n'est pas spécifié.
 - De la plage indiquée par entier-1 et entier-2, si THRU est utilisé.
5. Si la locution SOURCE LINE est spécifiée avec option OCCURRENCE, le jalonnement se fait :
 - Si THRU est omis, à l'apparition n (n = entier-3) d'une fonction DML dans la plage de lignes indiquée.
 - Si THRU est spécifié, des apparitions n à p (n = entier-3, p = entier-4) d'une fonction DML dans la plage de lignes indiquée.

Le compteur d'apparitions concernant la plage de lignes est mis à zéro au début de la session IDS pour les modes par lots ou IOF, ou au début d'une TPR sous TDS. Ce compteur est incrémenté de 1 chaque fois que les autres conditions de jalonnement (TRACE-liste-de-fonctions, RECORD, ..., PROGRAM, SOURCE LINE) sont satisfaites. Il est à remarquer en particulier que si différentes fonctions DML satisfaisant les conditions sont exécutées séquentiellement dans la même plage de lignes, le compteur d'apparitions est incrémenté pour chacune d'entre elles.

6. Lorsque plusieurs locutions SOURCE LINE sont utilisées, les plages de lignes ne doivent pas se chevaucher.
7. Si plusieurs clauses PROGRAM sont spécifiées, elles doivent citer des noms de programmes différents.
8. Si la clause PROGRAM est omise, les opérations font l'objet d'un jalonnement quel que soit le programme dans lequel elles apparaissent.

7.6.2.16 Clause PRINT

Fonction

Permet de spécifier la nature de l'information à imprimer pour chaque fonction sur laquelle porte le jalonnement.

Format

```

PRINT [ [ FIELDS [ IN HEXADECIMAL ] ] ]
      [ CURRENT [ ] ]
      [ [ SYSTEM POINTERS ] ]
      [ ]
      [ [ { AREA } ] ]
      [ [ { REALM } ] ]
      [ [ ] CURRENCIES ]
      [ [ RECORD ] ]
      [ [ SET ] ]
      [ ]
      [ DB-REGISTERS ]
      [ STATISTICS ]
      [ TIMING ]
    
```

Règles

1. Si CURRENT est spécifié, dès que la fonction jalonnée se termine l'information sur l'article courant de l'unité d'exécution est affichée :
 - Si FIELDS est spécifié, les zones de données de l'article sont affichées sous la forme définie dans leur schéma, ou en hexadécimal si HEXADECIMAL est spécifié.
 - Si POINTERS est spécifié, les pointeurs d'ensembles de l'article sont affichés.
 - Lorsque ni FIELDS, ni POINTERS n'est spécifié, données et pointeurs sont affichés.
2. Si CURRENCIES est spécifié, le contenu des indicateurs de point courant est affiché dès la fin d'exécution de la fonction jalonnée :
 - Avec AREA/REALM, les indicateurs de point courant d'aire sont affichés;
 - Avec RECORD les indicateurs de point courant de type d'article sont affichés;
 - Avec SET, les indicateurs de point courant de type d'ensemble sont affichés;
 - Si ni AREA, ni RECORD, ni SET n'est spécifié, tous les indicateurs de point courant sont affichés.
3. Si DB-REGISTERS est spécifié, le contenu des registres spéciaux IDS est affiché dès la fin d'exécution de la fonction jalonnée.
4. Si STATISTICS est spécifié, l'information sur les fonctions SYSTEM à l'exécution de la fonction jalonnée est affichée.
5. Si TIMING est spécifié, le temps de traitement et le temps total consommés par la fonction jalonnée sont affichés.
6. Le maximum d'information sera imprimé lorsque PRINT est spécifié seul.
7. Lorsque la clause PRINT est omise, seul le nom de la fonction et la clé-base-de-données de l'article courant de l'unité d'exécution sont affichés.

Interprétation du listage de jalonnement

Un échantillon de listage TRACE est fourni dans la Figure 7-3. Les symboles dont la signification n'est pas évidente nécessitent quelques commentaires.

Explication de la première ligne :

La première ligne indique le nom du programme et le numéro interne de ligne origine de l'instruction qui a fait l'objet d'un jalonnement, la catégorie de la fonction SGBD, l'heure courante à l'exécution de la fonction, le nom du schéma couramment utilisé ainsi que l'article courant de l'unité d'exécution.

Si au moment de l'exécution le point courant de l'unité d'exécution (CRU) n'est pas NUL, CRU est suivi de la lettre P (indiquant présent) et de la clé-base-de-données de l'article courant de l'unité d'exécution. Cette clé est fournie en hexadécimal avec traduction sous forme décimale (A: numéro d'aire, P: numéro de page, L: numéro de ligne). Les sigles AN et RN fournissent respectueusement le nom d'aire et le nom d'article du point courant de l'unité d'exécution.

Si le point courant d'unité d'exécution est à NUL, CRU est suivi de la lettre N (indiquant nul).

Pointeurs

Les pointeurs de point courant d'unité d'exécution, s'ils existent, sont fournis dans l'ordre où ils apparaissent dans la zone pointeurs de l'article. Ils sont affichés :

- Sous forme hexadécimale qui indique le nombre d'octets utilisés,
- Sous forme décimale donnant le numéro de zone (pointeur global seulement) ainsi que le numéro de page et le numéro de ligne.

Les lettres indiquées ci-dessous fournissent une information supplémentaire.

L => pointeur local

G => pointeur global

E => article maître d'un type d'ensemble vide

D => article détail OPTIONAL couramment non-rattaché

N => pointeur NEXT (suivant)

P => pointeur PRIOR (précédent)

O => pointeur OWNER (maître)

Zones

- Les décalages de zones dans la partie données de l'article sont fournis en décimal et hexadécimal.
- Le type des zones est indiqué symboliquement par :

CHAR	pour caractère
FB15, FB31	pour binaire signé 15 et 31
UPK	pour décimal éclaté non signé
PK	pour décimal condensé non signé
PK2	pour décimal non signé-2
UPKS	pour décimal signé éclaté
PKS	pour décimal signé condensé.

- Lorsque les zones sont affichées en hexadécimal, une traduction EBCDIC est également fournie.

Points courants

Un indicateur de point courant peut se trouver dans l'un des états suivants :

- PRESENT => lettre P
- NUL => l'indicateur correspondant n'est pas affiché.
- VIRTUEL => lettre V. L'élément DBK représente :
 - La clé-base-de-données de l'article effacé, pour le point courant de domaine,
 - La clé-base-de-données de l'article suivant l'article effacé ou supprimé de l'ensemble, pour le point courant de type d'ensemble.
- AN et RN donnent respectivement le nom-aire et le nom-article de l'article.

Exécution du programme utilisateur

Noms

Les noms d'aires, d'articles et d'ensembles sont représentés sur 17 caractères avec troncation à droite.

```

*****PGID: TG11      LINE: 9080  **FIND N/P SET      **AT 08:55:24.917**SCH: TG11-SH      **CRU N
**CURRENT OF REALM  TG11-A00      V DBK:000001A1 A:0      P:26      L:1      AN: TG11-A00      RN: R02
**CURRENT OF RECORD R01          P DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      RN: R01
**CURRENT OF SET      S01          V DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      RN: R01
                        S02          P DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      AN: R01
**DBSTATUS: 0502100 AN:          RN:          SN: S01
**TIMING                PROCESS: 2.3                ELAPSED: 2.3                (MILLISECS)
**STATISTICS  SYSTEM FUNCTION:  NUMBER : AVERAGE      DBCS SUBFUNCTION:  NUMBER : AVERAGE
                CATEGORY      : OF CALLS : I-O COUNT      CATEGORY      : OF CALLS : I-O COUNT
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
                FREECI      :      1 :      0.00
                GETCI      :      1 :      0.00

*****PGID: TG11      LINE: 9I30  **FIND N/P SET      **AT
08:55:24.949**SCH: TG11-SH      **CRU P
                        CRU DBK:00000030 A:0      P:3      L:0      AN: TG11-A00      RN: R02
**HEADER STATE: ACTIVE  CODE: 2      SIZE: 46      STATUS: 0      NEXT CALC-RECORD L:-      HASH
CHECK: E448 OWNER OWL RANK: 0
**POINTERS MEMBER: S01          L  N:      0180 A:-      P: 24      L: 0      AN: TG11-A00
                                P:      0180 A:-      P: 24      L: 0      AN: TG11-A00
                                O:      0180 A:-      P: 24      L: 0      AN: TG11-A00

**FIELDS      0 0000: R02-IDENT          CHAR  === R02
                7 0007: R02-CH          CHAR  C:
                10 000A: R02-C          UPKS  +3
                12 000C: R02-CALCH      CHAR  CALC:
                18 0012: R02-CALC      UPKS  +5327
                22 0016: R02-SH        CHAR  S:
                25 0019: R02-S          UPKS  +3
                27 0018: R02-TRAILER    CHAR  +++

**CURRENT OF REALM  TG11-A00      P DBK:00000030 A:0      P:3      L:0      AN: TG11-A00      RN: R02
**CURRENT OF RECORD R01          P DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      RN: R01
                        R02          P DBK:00000030 A:0      P:3      L:0      AN: TG11-A00      RN: R02
**CURRENT OF SET      S01          V DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      RN: R01
                        S02          P DBK:00000180 A:0      P:24      L:0      AN: TG11-A00      AN: R01
**DBSTATUS: 0000000 AN: TG11-A00  RN: R02      SN: S01
**TIMING                PROCESS: 2.3                ELAPSED: 2.3                (MILLISECS)
**STATISTICS  SYSTEM FUNCTION:  NUMBER : AVERAGE      DBCS SUBFUNCTION:  NUMBER : AVERAGE
                CATEGORY      : OF CALLS : I-O COUNT      CATEGORY      : OF CALLS : I-O COUNT
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
                FREECI      :      1 :      0.00
                GETCI      :      1 :      0.00

```

Figure 7-3. Listage du jalonnement externe

7.6.2.17 Clause IDSTRACE

Fonction

Permet de redéfinir le nom de fichier interne (nfi) du fichier de jalonnement IDS.

Format

```

IDSTRACE INTERNAL FILE NAME IS nfi
    
```

Règles

1. "nfi" doit se conformer aux règles de formation des noms de fichier interne telles qu'elles ont été définies dans le manuel de référence JCL (réf: 47 A2 11UJ).

A noter également que nfi doit être différent de :

```

IDSOPT
PRTFILE
COMFILE
SLLIBa
DDLIBb
SAVEc
    
```

dans lesquels a, b, c représentent des chiffres décimaux ou des espaces.

2. Si cette clause est omise, c'est implicitement IDSTRACE qui est pris comme fichier de jalonnement.

7.6.2.18 Clause OPEN

Fonction

Permet de spécifier le mode d'ouverture (OPEN) du fichier de jalonnement IDS.

Format

```

OPEN { OUTPUT }
        { EXTEND }
    
```

Règles

1. Si OUTPUT est spécifié, les informations de jalonnement sont écrites à partir du début du fichier. Toute information écrite antérieurement est recouverte.
2. Si EXTEND est spécifié, l'information de jalonnement est écrite à partir de la fin de fichier.

Cette option n'est pas prise en compte si le fichier de jalonnement est de type SYS. OUT.

3. Si cette clause est omise, c'est la spécification OPEN OUTPUT qui est prise en compte.

7.6.2.19 Clause IAC

Fonction

La clause IAC (Integrated Access Component) permet de lancer la fonction de "jalonnement interne" qui fournit de l'information sur certains appels au système lancés par le SGBD pendant l'exécution d'une fonction DML.

Format

```

      [          FIELDS [ IN HEXADECIMAL ] ]
      [
      [          SYSTEM POINTERS          ]
IAC [ WITH
      [          [ INPUT ]                  ]
      [          [ ] PAGE                   ]
      [          [ OUTPUT ]                 ]
  
```

Règles

1. Si IAC est utilisé seul, le jalonnement se limite aux appels internes au système avec leurs paramètres, leurs codes de retour, leurs temps de traitement et la totalité du temps écoulé.
2. Si IAC est accompagné de la locution FIELDS, pour chaque article mémoire impliqué dans un appel interne, les valeurs de la zone données sont rendues accessibles. Elles sont affichées en fonction des attributs fixés par le schéma, ou en hexadécimal si HEXADECIMAL est spécifié.
3. Si IAC est accompagné de la locution POINTERS, pour chaque article mémoire impliqué dans un appel interne, les pointeurs d'ensembles sont rendus disponibles.

4. Si IAC est accompagné de la locution PAGE, il y a impression en hexadécimal des pages auxquelles le SGBD accède et/ou qu'il modifie.
 - Si INPUT est spécifié, les pages demandées à la Gestion de Tampons sont imprimées avant leur utilisation (fonction interne GETCI, CHECKCI).
 - Si OUTPUT est spécifié, les pages modifiées sont imprimées avant leur remise à disposition de la Gestion de Tampons (fonction interne FREECI).
 - Si, ni INPUT, ni OUTPUT est spécifié, les pages demandées à la Gestion de Tampons et les pages modifiées remises à la disposition de la Gestion de Tampons sont imprimées.
5. Lorsque la clause IAC est omise, le "jalonnement interne" n'a pas lieu.

Remarque : Lorsque la clause IAC est spécifiée, les clauses RECORD, AREA, DB-STATUS et THRESHOLD ne sont pas prises en compte dans l'évaluation de la condition de jalonnement.

Exemple de commande TRACE

Reprenons l'exemple, exposé dans la Figure 5-2, d'une unité d'exécution composée de trois programmes: le programme principal PO est un module COBOL (pas de DML). Le second programme P1 accède à la base de données décrite par le schéma BD1. Le troisième programme, P2, accède à la base de données décrite par le schéma BD2.

Les informations de jalonnement de l'utilisateur et de IDS sont recueillies sur le même fichier de jalonnement IDS.

Supposons que le jalonnement IDS soit limité aux instructions FIND portant sur les articles EMPLOYE dans le domaine REGION-3 de la première base de données et aux instructions STORE de l'article EMPLOYE dans la seconde base de données. L'information à imprimer comprend les zones données de l'article en hexadécimal et les contenus des registres de la base de données. Les commandes utilisables à l'exécution sont spécifiées de la manière suivante :

```
SCHEMA NAME IS BD1
TRACE FIND
  RECORD IS EMPLOYE
  REALM IS REGION-3
  PRINT CURRENT FIELDS IN HEXADECIMAL DB-REGISTERS
```

```
SCHEMA NAME IS BD2
TRACE STORE
  RECORD IS EMPLOYE
  PRINT CURRENT FIELD IN HEXADECIMAL DB-REGISTERS
```

7.6.2.20 Commande CHECK PAGE

Fonction

Permet de valider la structure des pages extraites ou modifiées par le SGBD.

Format

```
CHECK [ INPUT ]  
[ OUTPUT ] PAGE STRUCTURE .
```

Règles

1. Si cette commande est spécifiée avec le mot-clé INPUT, la structure complète des pages demandées à la Gestion de Tampons est validée avant utilisation.
2. Si cette commande est spécifiée avec le mot-clé OUTPUT, la structure complète des pages modifiées est validée avant leur remise à disposition de la Gestion de Tampons.
3. Si cette commande est spécifiée avec INPUT et OUTPUT ou aucun des deux, la structure complète des pages extraites ou modifiées est validée.
4. Il n'y a pas de validation si cette commande n'est pas spécifiée.
5. La validation de structure de page est la même que celle qui est exécutée par la commande VALIDATED PAGE de l'utilitaire DB-UTILITY.
6. Lorsqu'une incohérence est relevée dans la structure de page, un message d'erreur accompagné du contenu de la page est imprimé sur le JOR et sur le fichier de jalonnement IDS, s'il existe, et l'activité (ou la transaction) est arrêtée prématurément.
7. Cette commande permet de rechercher la cause d'éventuelles incohérences de la base de données. Comme elle consomme du temps processeur, il est conseillé de ne pas l'utiliser dans des conditions normales d'exploitation.

7.6.2.21 Commande INTERNAL FILE NAME

Fonction

Permet de redéfinir les noms de fichiers internes (nfi) à utiliser au moment de l'exécution dans les instructions ASSIGN/DEFINE qui permettent de citer les aires mémoire.

Format

```
INTERNAL FILE NAME { FOR { AREA } nom-domaine IS nfi ....
                   { REALM }
```

Règles

1. "nfi" doit se conformer aux règles définies pour les noms de fichier interne dans le manuel de référence JCL, déjà cité.

De même, "nfi" doit être différent de:

```
DISOPT
IDSTRACE
PRTFILE
COMFILE
SLLIBa
DDLIBb
SAVEc
```

dans lesquelles a, b, c représentent des chiffres décimaux ou des espaces.

2. Si aucun nfi n'est fourni pour un domaine de la base de données, c'est la valeur définie implicitement ou explicitement dans le DMCL qui est utilisée.
3. Il ne doit pas exister de nom en double parmi les noms de fichiers internes d'un schéma.

Si plusieurs base de données font l'objet d'un accès par la même activité et si les noms de fichiers internes ne sont pas uniques parmi les différents schémas, l'utilisateur doit assurer lui-même la règle d'unicité de chaque nom de fichier interne pour les schémas concernés au moyen de la commande INTERNAL FILE NAME.

7.7 INTERPRETATION DES MESSAGES DU JOR

7.7.1 Format des messages

Le SGBD, sauf en ce qui concerne les drapeaux, les statistiques et les messages d'erreur de syntaxe, met en forme ses messages au JOR de la manière suivante :

({WNG}
DDM	indique un message de description et de manipulation de données. "x..." est un nombre décimal à 5 chiffres. Il sert de clé d'identification au texte de message du SGBD.
PGID	"y..." est une chaîne de 30 caractères, contenant l'identification du programme telle que spécifiée dans la clause PROGRAM-ID. Lorsque cette information n'est pas significative ou qu'elle n'est pas disponible, elle est remplacée par des astérisques.
SLN	"z..." est un nombre décimal de 7 chiffres, identifiant le numéro interne de ligne origine dans laquelle l'instruction correspondante est écrite. Lorsque cette information n'est pas significative ou non disponible, elle est remplacée par zéro.
SCH	"w..." est une chaîne de 30 caractères, contenant le nom de schéma qui caractérise la base de données sur laquelle opère la fonction courante du SGBD.
ORG	Cette zone fournit de l'information sur le module SGBD qui est à l'origine du message. "aa" est un nombre décimal de deux chiffres qui identifie la fonction SGBD. Cette fonction peut être de nature externe (telle qu'une instruction STORE) ou interne (telle qu'une instruction SELECT). Les codes de fonction du SGBD sont présentés plus loin dans ce chapitre. "bbbb" est un nombre décimal de quatre chiffres qui identifie un point de la procédure de la fonction SGBD. Cette identification dépend de la fonction et du type d'erreur. Elle ne peut être utilisée qu'en relation avec le listage. Elle est donc réservée au Service de Maintenance. Le groupe aa bbbb peut se répéter. Il indique alors le niveau de pile des fonctions internes du SGBD appelées au moment de la détection et du compte-rendu d'erreur. Le niveau d'empilage large maximum est 6.

ERR/WNG ERR indique une erreur bloquante, WNG indique un avertissement, "u..." est un nombre à quatre chiffres décimaux qui représentent le code de l'erreur. Il est égal à la clé du message lorsqu'un message correspond à une erreur. Egal ou plus grand que la clé du message, il indique que le message couvre une catégorie complète d'erreurs.

Les codes d'erreurs sont indiqués plus loin dans ce chapitre.

Texte Constitue le texte en clair de l'erreur. Bien que ce texte soit généralement très explicite on pourra trouver plus de détails dans le manuel Error Messages and Return Code (réf: 47 A2 10UJ).

Remarque : Les messages du SGBD décrits ci-dessus pour le JOR peuvent être également transmis sous le même format au fichier de jalonnement IDS s'il existe, et sur les états imprimés de SYSOUT et de TERMINAL des utilitaires de base de données. Ils peuvent être également disposés dans le registre DB-DETAIL-STATUS des programmes DML (messages d'avertissement seulement).

7.7.2 Codes de fonction du SGBD

Le Tableau 7-4 plus loin indique les codes de fonctions du SGBD qui apparaissent au JOR. Les valeurs sont indiquées en décimal. On peut distinguer plusieurs catégories :

- Codes 00 -> 24: correspondent aux instructions DML externes,
- Codes 25 -> 29: correspondent aux fonctions d'introduction générées dans les programmes COBOL
- Codes 30 -> 34: correspondent aux fonctions externes ou internes qui se rapportent au début et à la fin d'une session ou d'une unité de consolidation
- Codes 35 -> 44: correspondent aux fonctions externes lancées par les utilitaires de base de données
- Codes 45 -> 49: correspondent aux fonctions internes représentées sous forme de "sous-fonction" dans les statistiques d'une session IDS
- Codes 50 -> 79: correspondent aux autres fonctions internes telles que les fonctions d'appel à UFAS GAC (60 64) ou le JOURNAL (75 79)
- Codes 80 -> 99: correspondent au niveau de compatibilité qui traduit les instructions DML écrites en version 1C/1D en programme de la version 1E.

Tableau 7-4. Codes des fonctions du SGBD dans le JOR

Function	Code	Function	Code	Function	Code	Function	Code
ACCEPT	00	UCINIT0	25	CHECK	50	WAJN	75
CONNECT	01	UCINIT2	26	REPORT	51	OPAJN	76
DISCONNECT	02	UCINIT	27	ABTERM	52	CLAJN	77
ERASE	03		28	TRACE	53	NOTBJN	78
	04		29	PAGCHK	54	ROLLBJN	79
FIND	05	STARTDBS	30	GETRCALC	55		80
FINISH	06	TERMDBS	31	EOPSEM	56	IACCEPT	81
	07	ROLLBACK	32		57	ICONNEC	82
GET	08	RESTART	33		58	IDISCON	83
IF (DB Condition)	09	CLEANPOINT	34		59	IERASE	84
	10	GETPTR	35	GETCI	60	IFIND	85
MODIFY	11	SWITCH	36	FRECCI	61	IFINISH	86
	12	PUTPAGE	37	SETCI	62		87
READY	13	GETPAGE	38	LOKCI	63	IGET	88
	14	RELPAGE	39	CHCKCI	64	IIF	89
STORE	15	GETLAB	40	UNLCI	65		90
	16	MODLAB	41	INICMT	66	IMODIFY	91
	17	READCMT	42		67	IREADY	92
	18	MODOPT	43		68		93
	19	FREELock	44		69		94
	20	SELECT	45		70	ISTORE	95
	21	INSERT	46	PGREORG	71	IINIT1	96
	22	PLACEDIR	47		72	IINIT2	97
	23	PLACECALC	48		73		98
	24		49		74		99

7.7.3 Catégories des codes d'erreurs du SGBD

Le Tableau 7-5 ci-après indique les catégories de code d'erreur. Les valeurs sont exprimées en décimal.

La première catégorie correspond aux problèmes d'interface avec les fonctions systèmes UFAS/GAC/JOURNAL/TDS. Elle rend compte des codes retour système que le SGBD ne peut pas interpréter ou qui ne permettent pas une poursuite du traitement.

La seconde catégorie correspond à l'analyse du langage des commandes utilisables à l'exécution. Elle comprend les erreurs de syntaxe proprement dites et les erreurs causées par l'environnement du fichier IDSOPT.

La troisième catégorie correspond aux erreurs dans le changement du schéma (instruction ASSIGN incorrecte, non concordance entre les noms de schémas ou entre les dates, etc.) ou au moment du passage à l'état prêt des aires (instruction ASSIGN incorrecte, état transitoire, etc.).

La quatrième catégorie correspond aux contrôles de cohérence d'étiquette et de format que le SGBD effectue en permanence dès qu'il accède à ces structures de contrôle d'exécution.

Les cinq catégories suivantes correspondent aux contrôles de cohérence effectués en permanence par le SGBD lorsqu'il compare et interprète les structures de gestion préparées avant l'exécution: schéma, programme, base de données. Comme il est souvent difficile de déterminer lequel de ces trois ensembles est défectueux, la classification d'une erreur dans l'une de ces cinq catégories peut se révéler aléatoire et ne pas refléter la cause réelle de l'erreur.

La dernière catégorie correspond à des codes d'avertissement.

Tableau 7-5. Catégories des codes d'erreur du SGBD dans le JOR

Catégorie	Sous-catégorie	Code d'erreur
Interface UFAS/GAC/JOURNAL/TDS	UFAS/GAC	0 -<=256> 31
	JOURNAL	32 -<=256> 39
	TDS	40 -<=256> 47
	Autres	48 -<=256> 63
Analyse du langage de commande	Analyse syntactique	64 -<=256> 351
	Autres	352 -<=256> 383
Initialisation de session ou d'aire	Chargement du schéma	384 -<=256> 415
	Ouverture d'aire	416 -<=256> 447
	Autres	448 -<=256> 511
Contrôles de cohérence système	Étiquettes de structure de contrôle	512 -<=256> 535
	Formats de structure de contrôle de compatibilité (compatibilité de version)	536 -<=256> 559
	TDS	560 -<=256> 575
	Autres	576 -<=256> 767
Incohérences dans la base de données		768 -<=256> 895
Incohérences dans le DML		896 -<=256> 1023
Incohérences dans le schéma		1024 -<=256> 1151
Incohérences entre schéma et base de données		1536 -<=256> 1791
Incohérences entre schéma et DML		1792 -<=256> 1919
Avertissements		3584 -<=256> 3839

7.8 INTERPRETATION DES INFORMATIONS STATISTIQUES ET CHRONOLOGIQUES

1. Pour un même programme DML et le même état initial de la base de données, il convient d'interpréter les résultats d'une session IDS en fonction du cadre d'exploitation.
 - a) Le nombre d'appels aux fonctions SGBD est constant.
 - b) Le nombre d'appels internes aux fonctions UFAS GAC est plus grand sous mode MONITORED que sous les modes EXCLUSIVE et SHARED.
 - c) Le nombre d'entrées/sorties peut varier en fonction des options de pool de tampons et du nombre de tampons utilisé.

- Supposons que le programme effectue la recherche dans un type d'ensemble trié qui occupe quatre pages. Si le nombre de tampons est d'au moins quatre, les quatre pages sont chargées en mémoire principale dès la première citation et les transferts d'information entre les pages de la zone UWA deviennent des opérations de transfert (MOVE) de mémoire principale, et ce aussi longtemps que le programme travaille sur ce type d'ensemble. Si le nombre de tampons est inférieur à quatre et que l'ordre de classement fourni par le tri ne correspond pas à l'ordre de classement des clés base-de-données, le programme effectue de fréquents changements de page, ce qui nécessite à chaque fois une opération d'entrée/sortie pour remplacer une page par une autre dans l'un des tampons.

- Il est à noter également, qu'à cause de la technique de Gestion de Tampons qui retarde la réécriture des pages modifiées, des transferts de pages peuvent être attribués à une instruction qui ne les a pas provoqué directement. Par exemple, une instruction "FIND DB-KEY IS...", qui nécessite tout au plus un transfert, peut se voir attribuer une opération de transfert supplémentaire s'il se révèle d'abord nécessaire de libérer un tampon en réécrivant sur disque la page modifiée qu'il contenait.

C'est particulièrement le cas, dans un cadre d'exploitation TDS utilisant le mécanisme de mise à jour différée, lorsque toutes les opérations d'entrée/sortie se rapportant à la réécriture des pages modifiées sont exécutées par la fonction COMMIT.

- Dans les cadres d'exploitation par lots et IOF, les chiffres qui figurent sur les états ne comprennent pas les E-S opérées par les unités de consolidation qui ont échoué, c'est-à-dire celles qui ont été exécutées entre le dernier point de consolidation (ou le dernier point de reprise) précédant un point d'arrêt prématuré et ce point d'arrêt lui-même.

Dans le cas d'une relance d'activité après une situation de blocage ou tout autre incident, les statistiques JOR UFAS et E-S CONNECT correspondent aux opérations de la dernière partie de l'activité relancée, alors que les statistiques du SGBD reflètent la totalité de l'activité elle-même, à l'exclusion toutefois des phases d'arrêt prématuré comme indiqué ci-dessus.

- Dans un cadre d'exploitation TDS, les chiffres des états comprennent les opérations E/S des unités de consolidation qui ont échouées.

d) Dans un environnement de mono-programmation sans édition des sorties, les temps écoulés sont significatifs. Ils peuvent toutefois être influencés par le nombre de débordements de pile si ces phénomènes se produisent trop fréquemment. Voir au chapitre 6 la commande LINKER adéquate.

Les temps de traitement ne sont pas affectés, en théorie, par l'environnement de multi-programmation, sauf si des conflits d'accès à la mémoire principale ont pour résultat un grand nombre de segments manquants.

Comme pour le comptage des E-S, les décomptes des temps écoulés et des temps de traitement sont influencés par le choix d'un pool pour les tampons et le nombre de tampons utilisés. Ils sont également influencés par le mode de partage et l'utilisation ou non des journaux.

Les chiffres qui figurent dans les états ne comprennent pas le temps utilisé par la fonction de jalonnement externe en fin d'exécution d'instruction. Si le jalonnement interne est déclenché, le temps nécessaire à l'impression des appels système est inclus dans le temps écoulé et dans le temps de traitement, ce qui leur ôte toute signification.

Les chiffres figurant dans les états comprennent le temps consommé par la fonction CHECK PAGE lorsque cette fonction est lancée par les options d'exécution.

Dans un cadre d'exploitation TDS, les différentes chronologies affichées pour la session TDS englobent toutes les transactions de la session:

- Le temps total de traitement représente la somme des temps de traitements individuels des transactions. Aucun chevauchement n'est possible entre ces différentes chronologies de traitements.

- Le temps total écoulé représente également la somme de tous les temps écoulés de toutes les transactions. Comme ces transactions, dans le cadre de la session TDS, se déroulent en parallèle, ce total peut largement dépasser le temps utilisé par la session TDS elle-même.

2. De manière à obtenir des moyennes aussi significatives que possible, les instructions DML ont été divisées en un nombre de catégories plus grand que le nombre de verbes DML. Par exemple, une instruction "FIND DB-KEY IS...", qui nécessite tout au plus un seul transfert n'est pas classée dans la même catégorie qu'une instruction "FIND nom-article WITHIN nom-ensemble USING..." qui nécessite le parcours de l'itinéraire de sélection d'ensemble et la recherche dans le type d'ensemble sélectionné.

Malgré cela, cette catégorisation sur la base du format externe ne réussit pas à apporter une précision absolue. Par exemple, une instruction "FIND FIRST WITHIN nom-domaine" prendra 40 ms pour extraire un article de la première page d'une aire complète et peut-être 40 secondes pour réaliser qu'une autre aire est vide. De la même manière, une instruction STORE prendra 40 ms pour ranger un article CALC, maître exclusif d'un ensemble, et peut-être 15 secondes pour ranger un article détail de type AUTOMATIC dans 10 ensembles triés. D'après ce qui vient d'être dit, il apparaît clairement que les chiffres moyens d'E-S, des temps écoulés et des temps de traitements sont dépourvus de signification dans cette catégorie.

3. L'ensemble des informations qui ont été fournies dans les pages précédentes s'appliquent aussi bien aux statistiques et à la chronologie affichées dans le JOR à la fin d'une session IDS, qu'aux statistiques et à la chronologie enregistrées sur le fichier de jalonnement IDS à la fin d'une instruction DML jalonnée.

A. Aide-mémoire des syntaxes DDL, DMCL et DML

A.1 STRUCTURE DU DDL DE SCHEMA

Rubriques		Sous-rubriques		Clauses de rubrique ou sous-rubrique			
Nombre	Nom	Nombre	Nom	Nombre	Nom		
1	SCHEMA	-	-	1	SCHEMA		
1 à 2048	AREA	-	-	1	AREA		
1 à 2048	RECORD		RECORD	1	RECORD		
				1	LOCATION		
				1	WITHIN		
				0 à 240	CHECK		
			DATA	1	Nom-de-donnée-utilisateur		
				0 ou 1	TYPE		
				0 ou 1	OCCURS		
				0 ou 1	CHECK		
		0 à 2048	SET	1	SET	1	SET
						1	OWNER
1	ORDER						
1 à 2048	MEMBER			1	MEMBER		
				1	INSERTION		
				0 à 240	DUPLICATES		
				0 à 1	KEY		
				1	SELECTION		

A.1.1 DDL de schéma - Rubrique schéma

SCHEMA NAME IS nom-schéma .

A.1.2 DDL de schéma - Rubrique aire

AREA NAME IS nom-aire .

A.1.3 DDL de schéma - Sous-rubrique article

```

RECORD NAME IS nom-article-1

LOCATION MODE IS { DIRECT paramètre-base-de-données-1 }
                { CALC USING {identificateur-de-donnée-1} ... }
                { DUPLICATES ARE [NOT] ALLOWED }
                { VIA nom-ensemble-1 SET }

WITHIN { {ANY AREA } AREA-ID IS paramètre-base-de-données-2 }
        { {nom-aire-1} ... }
        { AREA OF OWNER }
[CHECK IS condition] ... .
    
```

A.1.4 DDL de schéma - Sous rubrique donnée

```

[num-niveau-1] nom-donnée-utilisateur-1

[ { { { UNPACKED } } } ]
[ { { [UNSIGNED] { PACKED } } } ]
[ { { { PACKED-2 } } } ]
[ { { SIGNED { UNPACKED } } } ]
[ TYPE IS { { { PACKED } } } ]
[ { { { } } } ]
[ { { [SIGNED] BINARY {15} } } ]
[ { { { } } } ]
[ { { { } } } ]
[ { { CHARACTER entier-3 } } ]

[OCCURS entier-4 TIMES]

[CHECK IS VALUE [NOT] {littéral-1 [THRU littéral-2]} ... ] .
    
```

A.1.5 DDL de schéma - Sous-rubrique ensemble

SET NAME IS nom-ensemble-1

OWNER IS nom-article-1

ORDER IS PERMANENT INSERTION IS

```
{ FIRST }
{ LAST }
{ NEXT }
{ PRIOR }
{
  { WITHIN RECORD-TYPE }
  { BY DEFINED KEYS }
  { SORTED [ RECORD-TYPE SEQUENCE IS {nom-article-2} ... ] }
  {
    DUPLICATES ARE { FIRST }
                   { LAST }
                   { NOT ALLOWED }
  }
}
```

A.1.6 DDL de schéma - Sous-rubrique détail

```

MEMBER IS nom-article-3

{INSERTION IS AUTOMATIC RETENTION IS MANDATORY}
{INSERTION IS MANUAL RETENTION IS OPTIONAL }

[DUPLICATES ARE NOT ALLOWED FOR {identificateur-de-donnée} ...] ...

[
[ KEY IS {ASCENDING } {identificateur-de-donnée-2} ]
[ {DESCENDING } {RECORD-TYPE } ]
[ {DATA-BASE-KEY } ] ]
[
[ [[ASCENDING ] {identificateur-de-donnée-3}] ]
[ [[DESCENDING] {RECORD-TYPE }]] ... ]
[ [ ] {DATA-BASE-KEY }]] ]
[
[ [ {FIRST } ] ] ]
[ [DUPLICATES ARE {LAST } ] ] ]
[ [ {NOT ALLOWED} ] ] ]

SET SELECTION [FOR nom-ensemble-1] IS

THRU nom-ensemble-2 OWNER IDENTIFIED BY

{ APPLICATION }
{ DATA-BASE-KEY [EQUAL TO paramètre-base-de-données-1] }
{ CALC-KEY [ identificateur-de-donnée-4 EQUAL TO }
{ { identificateur-de-donnée-5 } ] }
{ { paramètre-base-de-données-2 } ] ... }
{ [AREA-ID EQUAL TO paramètre-base-de-données-3] }

[ THEN THRU nom-ensemble-3 WHERE OWNER IDENTIFIE BY ]
[ {identificateur-de-donnée-6 [EQUAL TO ] ]
[ {identificateur-de-donnée-7 }]] ]
[ {paramètre-base-de-données-4 }]] ... ] ...

```

A.2 STRUCTURE DU DMCL

Rubriques		Clauses de la rubrique	
Nombre	Nom	Nombre	Nom
1	SCHEMA	1	SCHEMA
		0 ou 1	AREAS
		0 ou 1	GLOBAL
		0 ou 1	NO LOCAL
		0 ou 1	POOL
		0 ou 1	BUFFERS
1 à 2048	AREA	1	AREA
		0 ou 1	AREA FILE
		1	NUMBER-OF-PAGES
		1	LINES-PER-PAGES
		1	PAGE-SIZE
		0 ou 1	CALC-INTERVAL
		0 ou 1	LOCAL
0 à 2048	RECORD	1	RECORD
		0 à 2048	RANGE
		0 ou 1	MIGRATION
0 à 2048	SET	1	SET
		0 ou 1	NO LOCAL

A.2.1 DMCL Rubrique schéma

```

SCHEMA NAME IS nom-schéma
    [EXTEND NUMBER OF AREAS TO entier]
    [EXTEND GLOBAL POINTERS TO entier BYTES]
    [NO LOCAL POINTERS]
    [BUFFER POOL NAME IS nom-pool-tampons]
    [NUMBER OF BUFFERS IS entier] .
    
```

```

AREA NAME IS nom-aire
    [AREA INTERNAL FILE NAME IS nfi]
    NUMBER-OF-PAGES IS entier [ OPTIMIZE { HIGH } ]
    [ OPTIMIZE { LOW } ]
    NUMBER OF LINES-PER-PAGE IS entier
    PAGE-SIZE IS entier BYTES
    [CALC-INTERVAL IS entier PAGES]
    [EXTEND LOCAL POINTERS TO entier BYTES] .
    
```

A.2.2 DMCL - Rubrique article

```

RECORD NAME IS nom-article
    [
    [ RANGE [WITHIN nom-aire] IS {entier PAGES FROM PAGE entier} ]
    [ {entier PAGES FROM PAGE entier} ]
    [ {PAGE entier THRU entier } ]
    [ ] ...
    [ [ OPTIMIZE { HIGH } ] ]
    [ [ OPTIMIZE { } ] ]
    [ [ OPTIMIZE { LOW } ] ]
    ]
    [MIGRATION IS ALLOWED] .
    
```

A.2.3 DMCL - Rubrique ensemble

```

SET NAME IS nom-ensemble
    [NO LOCAL POINTERS] .
    
```

A.2.4 DML dans la DATA DIVISION

```

DATA DIVISION.
SUB-SCHEMA SECTION.
DB nom-sous-schéma.
[
[ {WORKING-STORAGE}
[ DB-DESCRIPTIONS IN { } SECTION. ]
[ {LINKAGE} ] ]
[ { ALL } ] ]
[ {RECORDS ARE { [NOT] {nom-article} ... } } ] ]
[ { } ] ]
[ { ALL } ] ]
[ {REALMS ARE { [NOT] {nom-domaine} ... } } ] ]
[ { } ] ]
...
WORKING-STORAGE SECTION.
...
numéro-niveau identificateur USAGE IS DB-KEY.
...
LINKAGE SECTION.
...
numéro-niveau identificateur USAGE IS DB-KEY.

```

A.2.5 DML dans la PROCEDURE DIVISION (Partie Déclarative)

```

USE FOR DB-EXCEPTION.(première section USE dans la partie DECLARATIVES)

```

A.2.6 DML dans la PROCEDURE DIVISION (Partie non déclarative)

```

ACCEPT identificateur-1 FROM [nom-domaine ]
      (DB-KEY)                [nom-article ] CURRENCY .
                              [nom-ensemble]

ACCEPT identificateur-2 FROM [nom-article ]
      (PIC X(30))             [nom-ensemble ] REALM-NAME .
                              [identificateur-3]
                              [(DB-KEY)      ]

ACCEPT identificateur-4 FROM nom-ensemble {NEXT }
      (DB-KEY)                {PRIOR}
                              {OWNER}

ACCEPT identificateur-5 FROM nom-domaine LINES-PER-PAGE .
      (COMP-1)

ACCEPT identificateur-6 FROM nom-domaine MINIMUM-DB-KEY
      [ OF nom-article ] .

ACCEPT identificateur-7 FROM nom-domaine NUMBER-OF-PAGES
      [ OF nom-article ] .

-----
CONNECT [nom-article] TO nom-ensemble [RETAINING-spécification] .
-----
DISCONNECT [nom-article] FROM nom-ensemble .
-----
ERASE [nom-article] [ALL MEMBERS] .

FIND [nom-article] DB-KEY IS identificateur [RETAINING-spécification].
      (DB-KEY)

FIND { ANY }
     { DUPLICATE } nom-article [RETAINING-spécification].

FIND { NEXT }
     { PRIOR }
     { FIRST }
     { LAST } [nom-article] WITHIN {nom-domaine }
     {integer } {nom-domaine }
     { identificateur } [RETAINING-spécification].
     { (COMP-2) }

FIND CURRENT [nom-article] [ WITHIN {nom-domaine } ]
      [ {set name } ]
      [RETAINING-spécification] .

FIND OWNER WITHIN nom-ensemble [RETAINING-spécification].

FIND nom-article WITHIN nom-ensemble [CURRENT] [USING {identificateur} ... ]
      [RETAINING-spécification ].

FIND DUPLICATE WITHIN nom-ensemble USING identificateur ...
      [RETAINING-spécification].

```

Aide-mémoire des syntaxes DDL, DMCL et DML

```

FINISH [nom-ensemble] ... .
-----
GET [nom-article] .
GET {identificateur} ... .
-----
[NOT] [nom-ensemble] {OWNER }
                        {MEMBER}
                        {TENANT}
nom-ensemble IS [NOT] EMPTY
-----
MODIFY { [nom-article ] } [RETAINING-spécification] .
      { {identificateur} ... }
-----
MODIFY [nom-article] ONLY {ALL } MEMBERSHIP
                        {nom-ensemble} ... }
                        [RETAINING-spécification] .
-----
MODIFY { [nom-article] } INCLUDING { ALL } MEMBERSHIP
      { {identificateur} ... } {nom-ensemble} ... }
                        [RETAINING-spécification] .
-----
READY { [nom-domaine-1] ... USAGE-MODE IS }
      { {EXCLUSIVE } {RETRIEVAL} } }
      { {UPDATE } } }
      { {SHARED } RETRIEVAL } ... }
      { {RETRIEVAL} } }
      { { [MONITORED] } {UPDATE} } }
-----
STORE nom-article [RETAINING-spécification] .
-----
Locution RETAINING :
                    { MULTIPLE }
                    { REALM }
RETAINING CURRENCY FOR { RECORD }
                    { SETS }
                    { {nom-ensemble} ... } }
-----
[num-niveau-1] nom-donnée-utilisateur-1
[ { { {UNPACKED} } } ]
[ { { [UNSIGNED] } { [PACKED] } } ]
[ { { {PACKED-2} } } } } DECIMAL entier-1 [ , entier-2] ]
[ { { SIGNED } {UNPACKED} } } ]
[ TYPE IS { { [PACKED] } } ]
[ { { { } } } ]
[ { { [SIGNED] } BINARY {15} } ]
[ { { {31} } } ]
[ { CHARACTER entier-3 } ]
-----
[OCCURS entier-4 TIMES]
-----
[CHECK IS VALUE [NOT] {littéral-1 [THRU littéral-2]} ... ] .

```


B. Différences entre GCOS64 1E et GCOS 7

Cette annexe résume les différences qui existent entre les deux versions de logiciel. Les différences entre les versions 1D de GCOS64 et V1 de GCOS 7 peuvent être déduites en comparant les différences entre 1D et 1E et le contenu de la présente annexe. Celle-ci reprend l'ordre de classement des chapitres du manuel pour examiner les différences spécifiques à chaque composant de IDS/II.

B.1 NOUVELLES SPECIFICATIONS DANS PREALLOC

L'option TRACK du paramètre SPLIT permet à l'utilisateur de préciser, en termes de numéros de cylindre et de piste, l'adresse début de chaque espace fichier.

B.1.1 Compatibilité

1. Aucune modification n'a été apportée au format interne de la base de données. Les aires 1E sont acceptées sous GCOS 7 sans conversion.
2. Les aires réservées sous GCOS 7 peuvent s'utiliser dans la version 1E.

B.2 NOUVELLES SPECIFICATIONS COBOL/DML POUR LA COMPILATION ET L'EDITION DE LIENS

1. Désormais, seul le compilateur COBOL procède à la compilation des programmes COBOL contenant des instructions de DML IDS/II. Le processeur DMLPROC devient ainsi inutile.

De nouveaux mots-clés, DDLIB1, DDLIB2, DDLIB3 ont été introduits dans le JCL étendu du COBOL de manière à permettre au compilateur COBOL de rechercher le schéma de la base de données.

2. L'utilisation de la nouvelle option DDLIST dans la compilation d'un programme COBOL/DML permet à l'utilisateur d'obtenir un état imprimé de description de tous les articles.
3. Les instructions DML ne sont plus limitées au texte principal du programme. Elles peuvent être sujettes à des instructions COPY.
4. Les instructions READY et FINISH peuvent apparaître dans tous programmes COBOL/DML, c'est-à-dire aussi bien dans les programmes secondaires (DB-DESCRIPTIONS IN LINKAGE) que dans les programmes principaux (DB-DESCRIPTIONS IN WORKING STORAGE). Il n'existe toujours aucune restriction sur le type de programme COBOL/DML (principal ou secondaire) qui exécute la première fonction READY ou la dernière fonction FINISH.
5. L'instruction "IF" DML devint une "condition-base-de-données" et peut s'utiliser comme n'importe quelle condition simple.
6. Autant que possible, des opérations de conversion automatique, portant sur le type et la longueur des données ont été introduites. Par exemple, dans l'instruction

```
ACCEPT identificateur FROM nom-domaine LINES-PER-  
PAGE
```

l'identificateur n'a plus besoin d'être déclaré comme COMP-1.

7. Les références croisées produites par le compilateur COBOL fournissent une information plus détaillée et plus précise sur l'utilisation du DML.
8. La commande VACSEG n'est plus nécessaire lors de l'Édition de liens contenant des instructions DML, car l'Éditeur de liens considère tous les segments sans affectation comme libres.

L'utilisateur, pour éviter un arrêt prématuré accompagné du message du gestionnaire de mémoire virtuelle VMM ENTRYOV (dépassement du nombre de rubriques), peut, lorsqu'il exécute un module édité sous la version 1E, procéder à une nouvelle édition de liens de ce module sous GCOS 7 sans utilisation de la commande VACSEG.

B.2.1 Compatibilité

1. Comme le pré-processeur DMLPROC est désormais obsolète, le JCL de compilation des programmes COBOL/DML doit être modifié :
 - Pour supprimer l'étape DMLPROC.
 - Pour inclure les nouveaux mots-clés du COBOL.
2. Les modules chargeables ayant fait l'objet d'une édition de liens sous la version 1E sont exécutables sous GCOS 7 sans modification.
3. Les programmes compilés par DMLPROC et COBOL sous la version 1E peuvent faire l'objet d'éditions de liens avec des programmes COBOL/DML compilés sous GCOS 7 par le compilateur COBOL.
4. Le compilateur COBOL, lorsqu'il procède à la compilation d'un programme COBOL/DML, arrête sa recherche dès qu'il trouve sur l'itinéraire DDLIB1, DDLIB2, DDLIB3, une unité de bibliothèque contenant le nom indiqué (c'est-à-dire le nom du schéma).

Il est donc conseillé de ne pas inclure dans un itinéraire de recherche deux éléments binaires de même nom que le nom de schéma et qui ne serait pas le schéma lui-même.

5. Quelques règles de syntaxe et de sémantique ont été renforcées :
 - Les instructions IDS/II ne doivent pas être écrites dans l'Aire A.
 - Toutes les fois où l'utilisation de l'élément DB-KEY est obligatoire elle sera vérifiée, comme par exemple dans l'instruction "ACCEPT identificateur FROM... CURRENCY".
 - Les attributs de donnée autorisés seront vérifiés en cas d'utilisation de l'élément DB-KEY.
 - Aucune possibilité n'est offerte d'écrire sur les tables et les registres IDS/II.
 - Les spécifications DB-CXT, DB-PARAMETERS et DB-REGISTERS ne peuvent être explicitement citées que dans :

```
. PROCEDURE DIVISION USING...  
. CALL... USING...
```
6. La qualification implicite des noms de zone par rapport au contexte n'est pas autorisée.
7. La SUB-SCHEMA SECTION doit contenir, implicitement ou explicitement, tous les articles auxquels un programme COBOL/DML fait référence.
8. Si, ni RECORDS, ni REALMS n'est spécifié dans la SUB-SCHEMA SECTION, tous les types d'articles de la base de données sont rendus disponibles au programme.

B.3 NOUVELLES SPECIFICATIONS DANS L'EXECUTION DU PROGRAMME UTILISATEUR

1. Des programmes COBOL/DML secondaires (c'est-à-dire qui contiennent DB-DESCRIPTIONS IN LINKAGE) qui sont compilés sous GCOS 7 peuvent être référencés dans les clauses "USE nom-de-procédure" de la génération TDS. Si tel est le cas, ils n'ont pas besoin de faire l'objet d'une édition de liens avec les routines TPR qui les appellent.
2. L'identificateur de programme, utilisé dans les spécifications IDSOPT et IDSTRACE, n'est plus limité à 12 caractères.
3. Afin de fournir une information plus précise lorsque l'on exécute un programme dans un contexte de base de données multiple, les messages du JOR contiennent l'identification du schéma utilisé.
4. De manière à éviter une troncation sous TDS, la longueur de ligne de sortie de IDSTRACE a été limitée à 120 caractères.
5. Lorsque la spécification STARTDBS (voir la clause TRACE au chapitre 7) est déclenchée par un programme secondaire, le jalonnement correspondant se rapporte au programme principal PGID et à une ligne non déterminée (LINE: ***).
6. Une nouvelle commande, INTERNAL FILE NAME... a été introduite comme option utilisable à l'exécution. Ceci permet à l'utilisateur de redéfinir les noms de fichiers internes (nfi) qui sont utilisés dans les instructions ASSIGN/DEFINE pour citer des aires mémoire spécifiques. On évite ainsi l'obligation de dupliquer les noms de fichier interne dans un contexte de base de données multiple.

B.3.1 Compatibilité

Les numéros de ligne utilisés dans les options d'exécution et dans les états produits par la commande TRACE ne sont plus des numéros externes de ligne origine mais des numéros internes de ligne générés par le COBOL.