HAL
open science

Réalisation d'un SGBD au CRIOP

Jean-Paul Bedouin

To cite this version:

Jean-Paul Bedouin. Réalisation d'un SGBD au CRIOP. Base de données [cs.DB]. 1981. dumas-00298594

HAL Id: dumas-00298594

https://dumas.ccsd.cnrs.fr/dumas-00298594v1

Submitted on 16 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE AGREE
DE GRENOBLE (C.U.E.F.A)

MEMOIRE

présenté en vue d'obtenir

LE DIPLOME D'INGENIEUR C.N.A.M.

en

INFORMATIQUE

par

JEAN PAUL BEDOUIN

REALISATION D'UN SGBD AU CRIOP

Les travaux relatifs au présent mémoire ont été effectués au CRIOP.
sous la direction de Monsieur BOUCHET.

1981

Que soient remerciés ici Monsieur BOLLIET qui a bien voulu accepter de présider le jury, Monsieur NAMIAN Professeur au Conservatoire National des Arts et Métiers et Monsieur ADIBA.

Au sein du CRIOP je tiens à exprimer ma reconnaissance à Monsieur BOUCHET pour sa confiance et son aide au cours de la réalisation du système présenté dans ce mémoire ainsi qu'aux membres de l'équipe qui ont contribué à l'aboutissement de ce projet.

T A B L E DES M A T I E R E S

SOMMAIRE 3

I INTRODUCTION 4

II PRESENTATION 5

- le CRIOP 6
- les choix 7

III LA METHODE DANIEL MARTIN 10

IV REALISATION D'UN SGBD AU CRIOP 15

. Le cahier des charges - Structures des fichiers 17
. Les fichiers annexes et les tables 25
. La zone de communication 40
. Structure et fonctionnement du noyau TP 49
. Application au traitement par lots 101
. Aspect conversationnel du SGBD 104
. Réorganisation des bases 109
. Aspect sécurité 114
. Evolution du système 119

V BILAN 131

VI ANNEXE 134

VII BIBLIOGRAPHIE 142

S O M M A I R E

Ce mémoire traite de la réalisation d'un système de bases de données hiérarchique au CRIOP.
Il tente d'expliquer comment le système a été généralisé à partir des concepts énoncés par Monsieur Daniel MARTIN.

I - INTRODUCTION

J'ai été amené, au cours de l'année 1978 à participer à la réalisation d'un système de base de données dans le cadre du démarrage d'un projet de télétraitement au sein du CRIOP.

De façon à mieux comprendre les choix qui ont été réalisés ainsi que les raisons qui les ont motivés, il semble intéressant de présenter assez brièvement et de façon simplifiée le contexte dans lequel le projet a vu le jour.

II - PRESENTATION

1- LE CRIOP : STRUCTURE ET MISSION

Le CRIOP est un GIE qui a été constitué en 1968 par quatre Caisses de Retraite : la CIPRA (Grenoble), la CIRCO (Lyon et Blois), la CIRR (Lyon) et l'IGIREL (Lyon) dans le but de résoudre d'une façon commune les problèmes d'exploitation informatique, chacune d'entre elles possédant alors son propre service d'analyse-programmation.

Pour ce faire le CRIOP louait un ordinateur et employait le personnel exclusivement nécessaire à la bonne marche de celui-ci.

Par la suite, il est apparu plus intéressant de mettre également en commun la partie étude et réalisations, vu la similitude des traitements informatiques effectués indépendamment par chaque Caisse.

Ce désir était confirmé et complété par la section Hôtellerie de la CIRCO (Blois) qui, vu l'éloignement ne pouvait envisager de travailler longtemps avec un système de traitement par lots.

Une étude réalisée au cours des années 1976 et début 1977 a permis d'établir un cahier des charges et au cours de la réunion du 8 Juillet 1977 les Directeurs de Caisse ont donné leur accord pour la réalisation d'une première partie d'un système de gestion en télétraitement.

La réalisation du projet a été confiée au CRIOP qui a vu sa structure modifiée par la constitution d'une équipe d'analyse-programmation.

2- LES CHOIX

Après la détermination du matériel (IBM 370/138) est intervenu le choix d'un système de télétraitement. Trois produits ont alors été examinés : CICS (IBM), WESTI (WESTINGHOUSE) et ENVIRON/1 (CINCOM).

Le choix s'est porté sur CICS pour les raisons suivantes :

- Evolution assurée (dans le cas d'évolution de matériel)
- Souplesse (possibilités très étendues)
- Bonnes performances pour de gros volumes
- Mise en oeuvre plus facile depuis l'arrivée de la nouvelle version (HLPI = High level Interface programming)
- Support technique assuré

Parallèlement, la question de l'opportunité d'un système de bases de données s'est posée. Un tel système était en fait nécessaire car :

- Le cahier des charges laissait présager d'une structure de données relativement complexe donc difficile à gérer au niveau des transactions.
- Dans un ensemble de données très vaste, il fallait éviter la redondance qui risquait d'être importante.
- Une gestion des fichiers au niveau de la transaction ne pouvait pas présenter le même niveau de fiabilité et de sécurité.
- L'évolution est beaucoup moins simple donc plus onéreuse si ce sont les transactions qui effectuent les accès au fichier.
- L'aspect conversationnel de certains systèmes, c'est-à-dire la possibilité de poser dans l'instant qui suit n'importe quelle question présentait un attrait non négligeable.

Trois SGBD ont été étudiés : il s'agit de :

- DL/1 (IBM)
- TOTAL (CINCOM)
- Produit "sur mesure" selon la méthode de Mr Daniel MARTIN.

L'examen de ces différents produits a conduit au tableau suivant (compte tenu du matériel et des applications futures).

	AVANTAGES	INCONVENIENTS
DL/1	- très fiable - installé dans beaucoup de sites donc évolution assurée	- consommation élevée en temps CPU - temps de réponse importants
TOTAL	- Bonnes performances - Simplicité - Sécurité	- Recherches sur index non aisées - Coût élevé - Traitement par lots s'effectuent dans l'ordre physique de la base
Produit sur mesure par la méthode DANIEL MARTIN	- Spécifité et souplesse - Bonnes performances - Outil puissant - Possibilité de questions de synthèse - Coût de location nul	- Conception difficile - Maintenance lourde - Investissement important au départ - Effort de réalisation - Difficultés rencontrées lors de la première réalisation

Bien que la troisième solution pouvait être qualifiée d'audacieuse, c'est son principe qui fut retenu. (Il faut dire que le temps de réalisation annoncée par Daniel MARTIN (2 à 3 hommes x mois) est intervenu de façon non négligeable dans la décision).

Cependant avant de s'engager définitivement, le CRIOP s'est fixé un certain délai de réalisation (fixé à un homme-mois) permettant de faire le point sur l'état d'avancement des travaux, et éventuellement de décider de l'abandon de ce produit au profit d'un "package" si les difficultés rencontrées étaient trop importantes (compte tenu de l'environnement CICS) ou si la mise en pratique de la méthode ne semblait pas pouvoir être réalisée dans une période de temps déterminée.

Au bout de ce délai il est apparu que la méthode Daniel MARTIN semblait parfaitement viable ; par contre, il fallait prévoir un temps de réalisation bien supérieur à celui annoncé et ceci principalement pour deux raisons :

- L'utilisation de CICS était nouvelle dans l'Entreprise
- La base était relativement complexe (nous verrons ce point plus en détail ultérieurement).

Il fut alors décidé que le choix était maintenu.

Nous allons donc étudier les grandes lignes de la méthode Daniel MARTIN.

III - LA METHODE 'DANIEL MARTIN'

Cette méthode est exposée dans l'ouvrage écrit par Daniel MARTIN "BASES DE DONNEES : METHODES PRATIQUES" édité chez DUNOD.

Elle s'adresse, selon l'auteur, aux utilisateurs [les plus nombreux] qui exploitent des petits ou moyens systèmes et qui de ce fait même n'osent pas envisager l'installation d'un SGBD sur leur configuration.

Le livre, dont le but est d'essayer de démythifier les bases de données, renonce à l'exposé de théories générales, difficiles à appliquer ; il fournit un ensemble de méthodes concrètes d'analyse, de programmation et d'exploitation de base de données.

Enfin l'ouvrage ne prétend pas être exhaustif. Il présente certaines méthodes simples qui ont fait leurs preuves et entend rester suffisamment général afin d'avoir la plus large écoute possible.

Nous allons essayer de dégager les grandes lignes de la conception qu'a Daniel MARTIN des bases de données par l'approche qu'il en fait, dans son livre ou au cours des séminaires qu'il organise.

Une base de données est un ensemble de renseignements qui présente les trois critères suivants :

- exhaustivité
- non redondance
- structure

Une base de données est subdivisée en fichiers logiques qui contiennent des renseignements fonctionnellement dépendants et qui sont découpés en enregistrements logiques.

Les fichiers logiques sont des groupements abstraits d'informations, mais le stockage de ces informations implique l'utilisation de fichiers "physiques" composés d'un certain nombre de sous-ensembles : les enregistrements physiques. En effet, un fichier logique donne souvent naissance à plusieurs fichiers physiques complémentaires pour des raisons d'efficacité de traitement ou de stockage. Il arrive aussi que l'on stocke dans un même fichier physique plusieurs types différents d'enregistrements.
Des fichiers appelés "index secondaires" et des tables de bits ou tables de présence facilitent certains types de sélection privilégiés.

Tous les renseignements ou zones constituant une base de données sont répertoriés dans un dictionnaire de données qui contient pour chaque information un certain nombre de caractéristiques telles que : fichier logique d'appartenance, fichier physique d'appartenance, nature de la zone, longueur, déplacement dans l'enregistrement physique, libellé de la zone, libellé réduit etc...

La gestion des fichiers de la base est assurée principalement par un programme appelé "noyau de base de données" ou tout simplement "noyau".
Il existe habituellement deux versions de ce programme ; l'une est destinées à fonctionner dans un environnement en temps réel, l'autre est utilisée dans un environnement batch.

Un noyau est en principe composé de quatre opérations fondamentales qui sont : sélection, addition, modification, suppression. Ces diverses opérations correspondent aux divers traitements qu'une base de données peut subir, indépendamment de son emploi, de son volume et de sa structure.

Les opérations d'addition et de suppression portent sur un enregistrement logique.

La modification s'applique au niveau le plus fin d'une base de données : la zone.

La sélection est une opération sur une base de données qui consiste à extraire un sous-ensemble d'enregistrements logiques répondant à un certain nombre de critères en fonction d'un parcours précisé par l'utilisateur.
Dans le cas où l'enregistrement logique comprend un grand nombre de zones la sélection permet d'extraire à l'intérieur du sous-ensemble d'enregistrements logiques répondant aux critères donnés, un certain nombre de ces zones. Ce deuxième type de sélection est appelé : "sélection avec extraction partielle".

D'autres modules doivent permettre de réaliser les opérations de mise à zéro et de nettoyage.

Par mise à zéro on entend :

- allocation de place disque des divers fichiers
- création des divers pointeurs ou compteurs
- écriture d'une date de création
- chargement d'une table décrivant les règles ou nombres-guides du traitement

Par nettoyage on entend :

- suppression des enregistrements "morts"
- regroupement des enregistrements qui se suivent logiquement
- tri des ensembles ordonnés (ex : index)

Plus généralement, l'opération de nettoyage laisse la base de données dans l'état où elle serait à la suite d'une création sans suppression. Cette opération permet également de récupérer de la place disque.

Ces opérations fondamentales sont également définies par leur mode de traitement qui porte sur :

- le temps dont on dispose pour effectuer l'opération
- le nombre d'enregistrements affecté par chaque opération

Ceci peut conduire éventuellement à ne rendre disponible que certaines opérations en télétraitement.

La structure des fichiers fait l'objet d'un choix dont les principaux paramètres sont :

- le type de fichier (fichier maitre, historique, en cours, etc...)
- le volume à stocker
- les traitements à effectuer
- la structure logique des informations.

L'aspect conversationnel des bases de données est assuré par deux modules, le compilateur et l'éditeur, qui permettent de questionner à chaque instant la base sur le contenu de chacune de ses zones.

Les étapes de la réalisation d'un système de bases de données selon la méthode Daniel MARTIN sont les suivantes:

- inventaire des informations à traiter et regroupement par fichier logique
- définition des sélections privilégiées et des index secondaires associés
- définition des liens entre les différents fichiers logiques et des traitements particuliers éventuels
- constitution du dictionnaire de données permettant de décrire chaque zone de la base
- description des enregistrements de la base
- analyse et programmation des modules permettant de réaliser les opérations énumérées ci-dessus.

Notons enfin que chaque noyau est spécifique de la sous-base qu'il gère. En effet, le dictionnaire n'est en principe pas accédé lors de la manipulation d'une zone. Chaque information est décrite dans l'enregistrement auquel elle appartient. D'autre part, les liens qui existent entre les fichiers logiques sont implicitement décrits dans la structure des programmes.

Cela signifie que la moindre modification concernant une zone de la base ou un lien entraine une intervention dans les programmes qui la gèrent. Cependant, dans le cas des bases de données dites "ouvertes", l'évolution peut se faire sans recompilation des programmes qui y accèdent, mais la méthode de réalisation qu'en donne Daniel MARTIN est présentée à la fois comme complexe et onéreuse.

IV - REALISATION D'UN SGBD AU CRIOP

AVERTISSEMENT :

La démarche pour aboutir à la réalisation d'un système de bases de données telle qu'elle a été effectuée au CRIOP ne correspond pas forcément aux idées reçues sur le sujet. Celle-ci a en effet évolué du particulier vers le général.

La première étape (définition de la base et de son environnement) s'est déroulée en collaboration avec Daniel MARTIN.

La deuxième étape (modification de la conception, analyse et programmation du système) a été réalisée sans son aide et dans une optique différente de ce qu'il préconisait.

Cette étude qui se limite à une description du système sans aborder l'aspect utilisateur laisse apparaître l'évolution d'idées qui a conduit à la réalisation du SGBD existant.

0- LE CAHIER DES CHARGES - STRUCTURES DE FICHIERS

La première phase du projet consiste en un suivi administratif des entreprises adhérentes, c'est-à-dire principalement à la création et mise à jour en temps réel des informations les concernant.

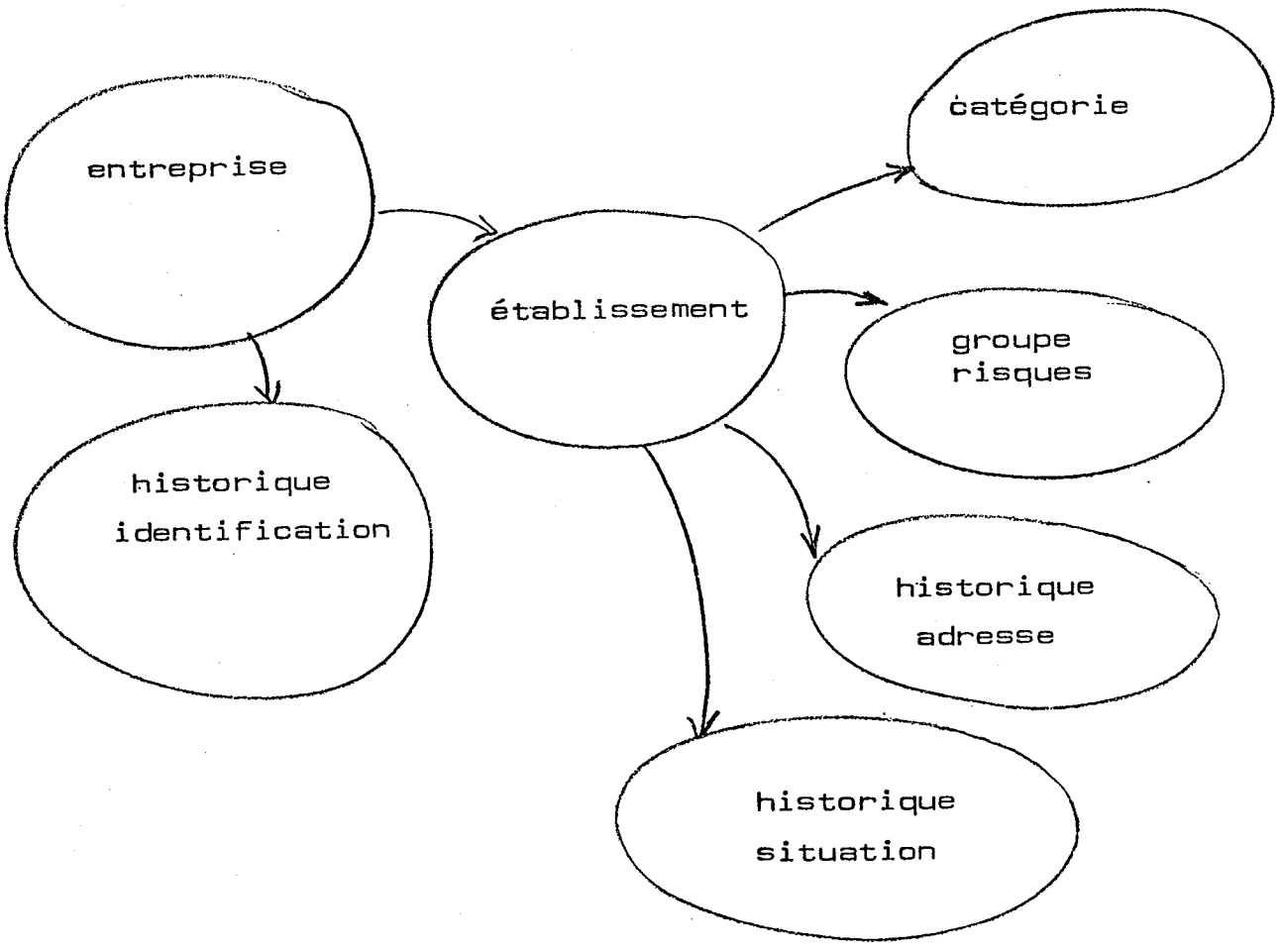
L'étude préalable avait permis de mettre en évidence les entités suivantes :

- l'entreprise est l'entité signataire d'un contrat de retraite avec l'institution.
- chaque entreprise peut posséder plusieurs établissements secondaires que l'institution doit connaître.
- pour chaque établissement d'une entreprise il est nécessaire de préciser les termes du contrat, c'est-à-dire les catégories du personnel concernées.
- Si l'entreprise est également signataire d'un contrat de prévoyance avec l'institution, à chaque établissement sont rattachés les risques couverts qui explicitent les termes de ce contrat par groupes de salariés.
- D'autre part, les utilisateurs souhaitent pouvoir connaître l'évolution de l'identification de l'entreprise ce qui implique la gestion d'un historique identification (au niveau entreprise).
- Ils souhaitent également pouvoir connaître l'évolution de l'adresse de chaque établissement ce qui implique la gestion d'un historique adresse (au niveau établissement).
- Les changements de situation d'un établissement doivent pouvoir être consultés, ce qui conduit à la gestion d'un historique situation (au niveau établissement).

Enfin il doit être possible de retrouver rapidement les informations concernant une entreprise lorsqu'on connaît soit :

- l'identification (raison sociale ou sigle) de l'entreprise
- l'adresse d'un établissement
- le nom du gérant de l'entreprise.

Nous sommes donc en présence d'un certain nombre d'entités et on peut schématiser ainsi les liens qui existent entre celles-ci.



Ce schéma met en évidence une structure hiérarchique des entités qui correspondent dans la méthode de Daniel MARTIN à la notion de fichier logique.

Une fois définie la structure logique de la base de données, il convient d'étudier sa structure physique. Cela nécessite des informations supplémentaires en particulier sur la fréquence d'apparition des entités les unes par rapport aux autres.

1ère constatation :

Environ 98 % des entreprises n'ont pas d'établissements secondaires. Ceci peut conduire pour des raisons de performances à assimiler la notion "d'entreprise" et "d'établissement principal" qui représente l'entreprise elle-même.

2ème constatation :

Pour environ 75 % des établissements, le contrat retraite est matérialisé par trois catégories /tranches ce qui tend à conduire, également pour des raisons de performances à englober trois catégories/tranches dans l'établissement (ou l'entreprise pour un établissement principal).

3ème constatation :

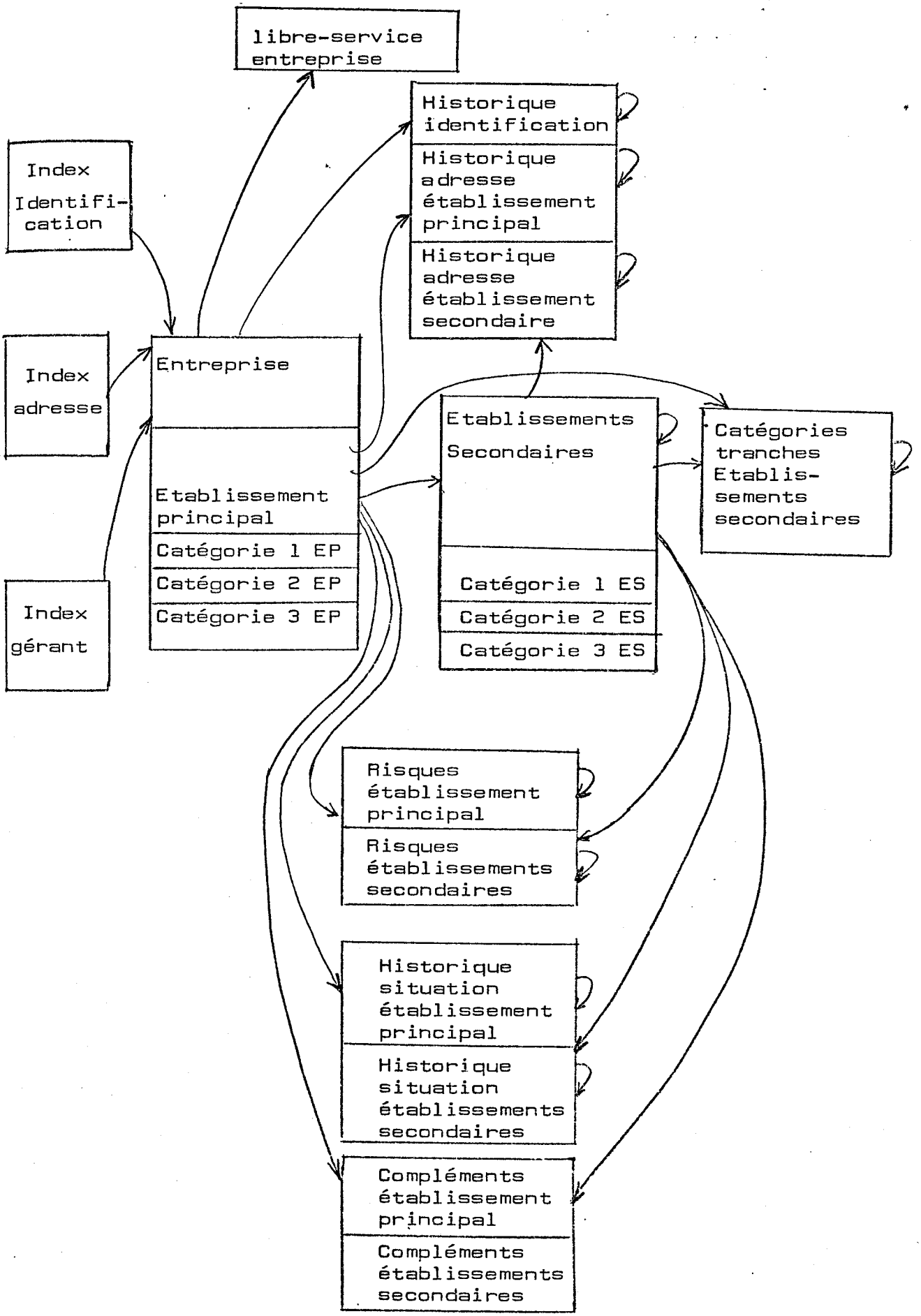
Au niveau de l'entreprise les utilisateurs souhaitent pouvoir signaler un certain nombre d'informations non codées donc non gérées. La fréquence d'apparition de ces "commentaires" est peu élevée (de l'ordre de 15 % des entreprises) c'est pourquoi il semble bon de ne pas réserver systématiquement la place dans le fichier entreprise mais de prévoir plutôt un fichier annexe ou "fichier de zones supplémentaires" que nous appelerons "libre service entreprise".

De même au niveau établissement certaines informations telles que la domiciliation bancaire apparaissent peu fréquemment (environ 10 % des cas) et pourraient aussi être mémorisées dans un fichier de zones supplémentaires que nous appelerons "compléments établissement".

4ème constatation :

Il est prévisible que les historiques identification et adresse auront des tailles d'enregistrement sensiblement les mêmes. Pourquoi ne pas en faire un seul fichier physique ? En effet il est tout à fait convenable de placer sur un même support physique des enregistrements différents par leur contenu mais de même taille physique.

La structure physique qui en découle est la suivante :



Après réflexion cette structure ne pouvait-être retenue principalement pour les trois raisons suivantes :

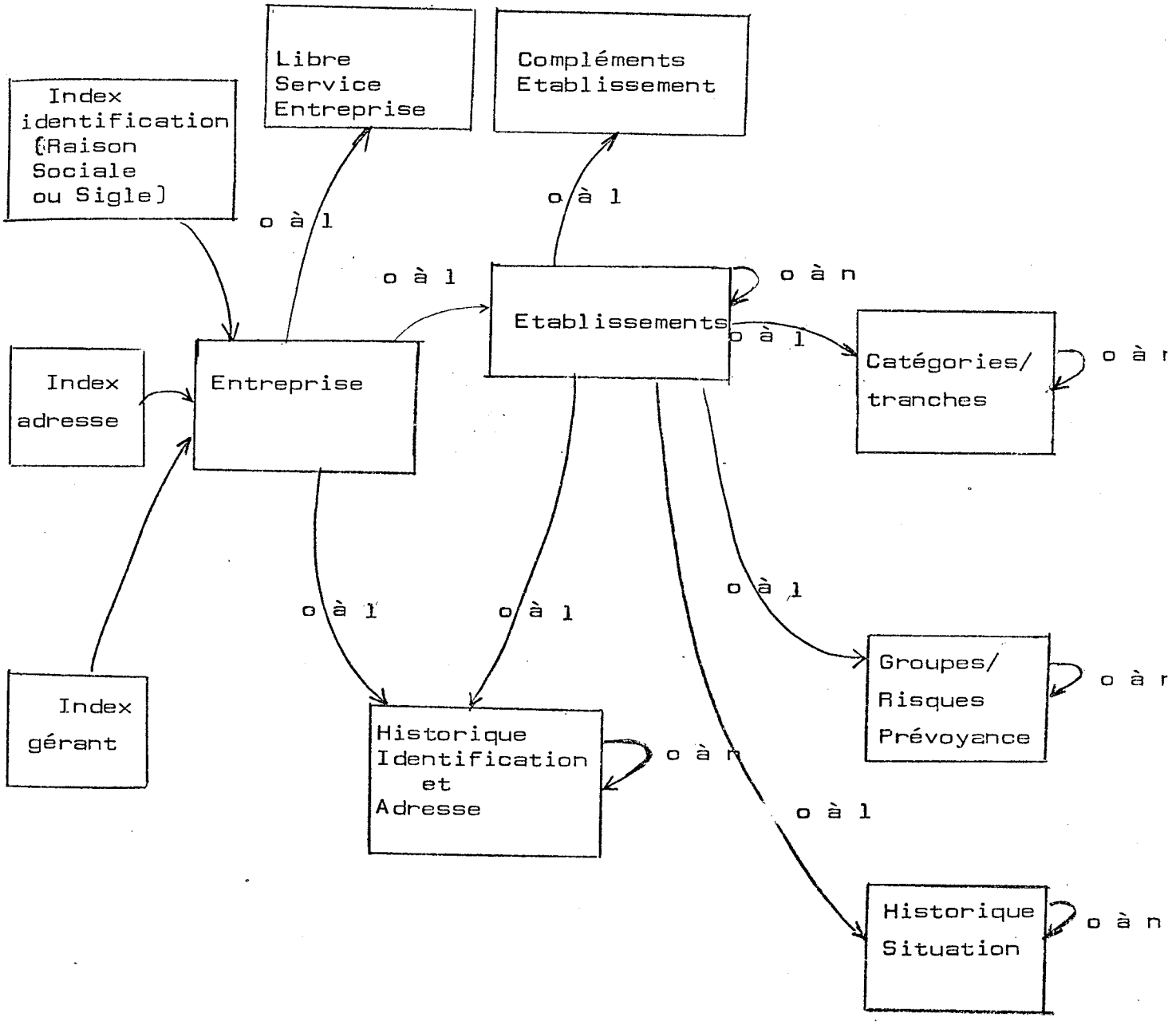
- La prise en compte de cas particuliers entraîne une gestion plus compliquée et plus délicate à mettre au point.

- D'autre part, ces particularités sont à l'encontre d'une conception plus générale des traitements réalisés sur des bases de données et otent tout espoir de réutiliser avec un minimum de modifications l'existant pour des applications futures.

- Enfin Daniel MARTIN n'y était pas très favorable.

Une structure physique plus simple était alors adoptée : elle consiste à ranger sur un même support physique les enregistrements logiques fonctionnellement identiques.

Elle est schématisée comme suit :



La définition de la structure physique comprend également l'étude de l'organisation de fichiers.

1- Le fichier entreprise :

La numérotation des entreprises par les institutions se prête bien à une organisation relative, cependant celle-ci ne peut-être retenue à cause de l'incorporation dans cette numérotation du code institution et du code gérant (alphabétique). C'est ce qui a conduit à s'orienter vers une organisation indexée.

2- Les index secondaires :

Il ne peut-être envisagé pour ce type de fichier autre chose qu'une organisation indexée.

Pour ces deux cas il fut décidé que cette organisation indexée serait au moins dans l'immédiat celle du constructeur (VSAM/KSDS) et non une gestion personnalisée.

3- Les autres fichiers :

Ceux-ci sont accessibles par pointeur et peuvent donc sans problème être prévus selon une organisation relative (nous avons choisi DAM relatif puis VSAM/RRDS).

La définition du dictionnaire de données (qui sera étudié ultérieurement) et quelques compléments d'information marquent la fin de l'aide technique apportée par Daniel MARTIN. Celui-ci présentant sa méthode pour réaliser des bases de données ouvertes, comme coûteuse en réalisation et en exploitation il ne pouvait en être question. La méthode permettant de réaliser des bases de données non ouvertes ne présentant, d'autre part pas un caractère suffisamment évolutif, nous nous sommes orientés vers une troisième solution que nous décrivons ci-après.

Sans avoir l'ambition de réaliser des bases de données totalement ouvertes il nous a cependant semblé indispensable de concevoir un système suffisamment souple pour permettre des évolutions de la base telles que ;

- variation des caractéristiques d'une zone (nature, longueur, emplacement).
- ajout d'informations
- ajout de fichiers logiques ou d'index secondaires.

Cette souplesse est réalisée par l'utilisation de plusieurs tables permettant le paramètrage du plus grand nombre possible d'éléments caractérisant la base de données.

Nous allons dans un premier temps effectuer une étude de la base dans un environnement de télétraitement. Cela suppose l'accès aux tables pendant toute la durée de la cession. Voyons comment cet accès est réalisé.

1- LES FICHIERS ANNEXES ET LES TABLES

1.1- CHARGEMENT DES TABLES SOUS CICS :

Au cours de sa phase d'initialisation CICS passe par une étape appelée PLT (program loading table) qui autorise l'exécution de taches utilisateurs. Ce moment se prête particulièrement au chargement des tables nécessaires au système de bases de données car elles peuvent ainsi être accessibles jusqu'à la fin de la cession de télétraitement.
Une contrainte est cependant imposée par CICS : ces tables doivent se trouver dans la bibliothèque des modules exécutables (CIL).
De façon pratique le chargement est effectué au moyen de la macroinstruction LOAD qui fournit en retour l'adresse d'implantation.

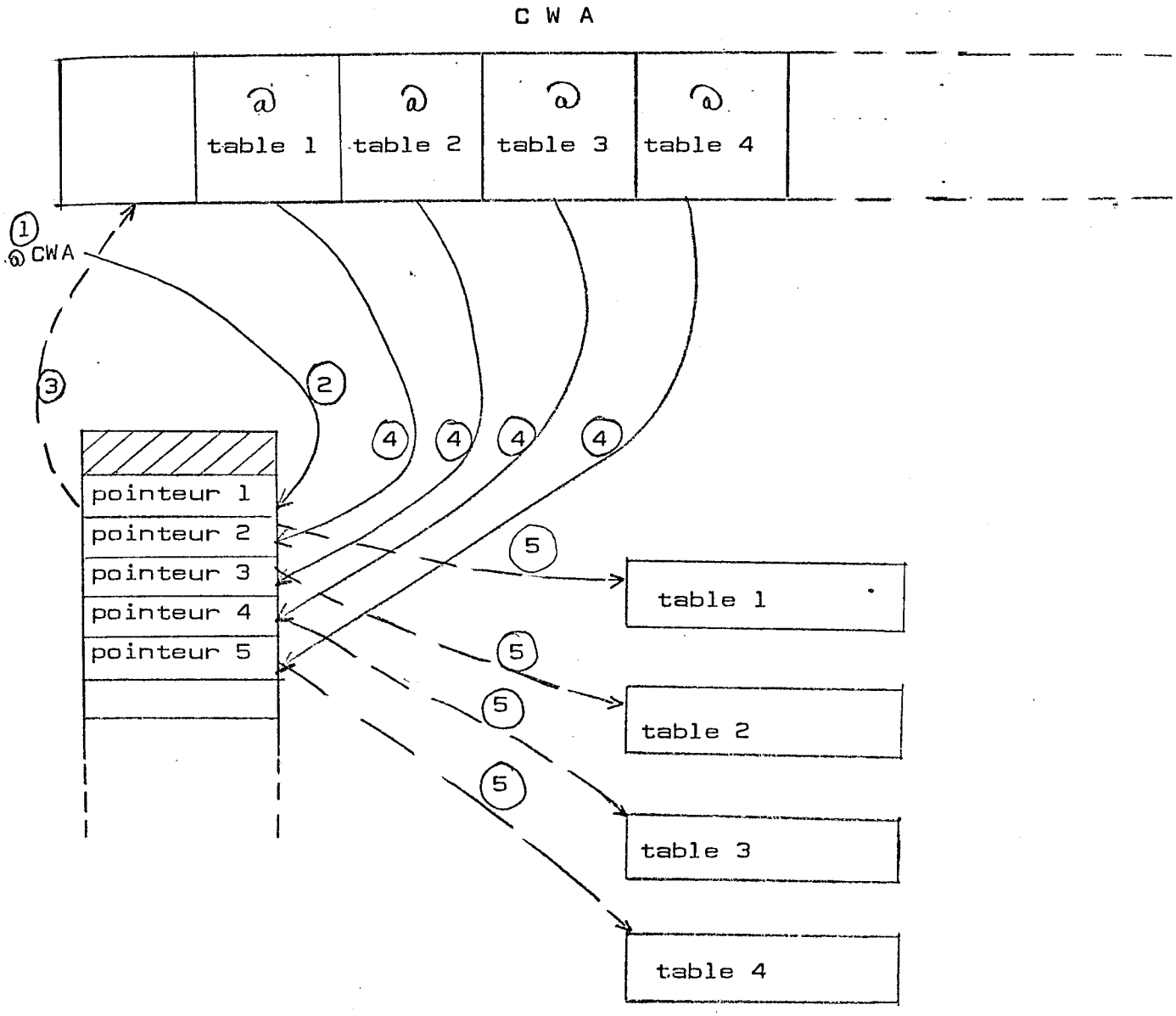
De plus parmi les différentes zones mémoire destinées à l'utilisateur du système TP il en existe une qui est commune à tous appelée CWA (Common work Area) et dont la taille est en général fixée aux alentours de 2000 à 3000 octets. Cette zone pouvant être adressée pendant la phase de PLT, elle permet de mémoriser les adresses d'implantation des différentes tables ainsi que l'occurence du prochain enregistrement libre sur les fichiers d'organisation directe relative.
Cela implique que les utilisateurs de la base de données ne puissent avoir accès à la CWA qu'en lecture au moins pour la partie de mémoire utilisée par le noyau.

1.2- ADRESSAGE DES TABLES :

Le langage de commandes HLPI possède un mécanisme particulier permettant d'accéder à des zones de mémoire externes au programme.
Ce mécanisme consiste à décrire en début de LINKAGE SECTION (partie du programme permettant de référencer les zones externes) une série de pointeurs puis les zones correspondant à ces pointeurs.

Une commande particulière permet d'obtenir l'adresse de la CWA. Cette adresse est transférée dans le pointeur correspondant et autorise à travailler avec le masque décrivant la CWA.
Il est maintenant possible de référencer les adresses d'implantation des différentes tables, adresses qui sont transférées dans les pointeurs correspondants.
Chaque transfert autorise à travailler avec les masques descripteurs des différentes tables.

Ce mécanisme d'adressage peut-être schématisé comme suit :



1 obtention adresse de la CWA

2 transfert adresse CWA dans pointeur 1

3 possibilité de référencer en CWA les adresses des tables

4 transfert des adresses de table dans les pointeurs correspondants

5 possibilité de référencer les tables

<u>REMARQUE</u> : Les transferts sont matérialisés en traits pleins, les possibilités d'adressage en traits discontinus.

1.3- LES TABLES :

1.3.1- Table des enregistrements vides :

Une base de données peut être considérée comme une collection d'objets dont certains seulement ont une valeur déterminée à un instant donné et de ce fait l'ajout d'un enregistrement logique à la base n'oblige pas à en renseigner chaque information.
Il convient donc de déterminer une valeur "nulle" ou "par défaut" pour toutes les zones de la base et c'est cette valeur qui est prise en compte chaque fois que l'information n'est pas connue.
Ces valeurs par défaut sont regroupées par enregistrements que nous appelerons "enregistrement vide".

A la création d'une entité il y a transfert préalable de l'enregistrement vide dans le buffer puis chaque information significative remplace la valeur initiale.

Ces enregistrements vides correspondant à des enregistrements physiques sont de longueur variable. De façon à ne pas perdre inutilement de la place mémoire en prévoyant une table dont la longueur de chaque poste correspondrait à une longueur maximale d'enregistrement vide ; chacun d'entre eux est placé dans la table à la suite de l'autre, de façon continue. Ceci oblige à mémoriser la position de chaque enregistrement vide par rapport au début de la table, mais permet un gain de place appréciable.

La description des enregistrements vides est faite en langage d'assemblage.
La table est assemblée au même titre qu'un programme puis après édition des liens, cataloguée dans la bibliothèque des modules exécutables (CIL).

Un schéma descriptif de la création de cette table se trouve en annexe à titre d'exemple.

REMARQUE : Dans le cas d'enregistrements index, on utilise deux postes descriptifs contigus, le premier contient les valeurs par défaut et les valeurs caractéristiques de l'index entraînant ou non la génération de celui-ci ; le deuxième poste contient des caractères spéciaux indiquant en regard de chaque information les traitements spécifiques à réaliser pour chacune d'elle :

X'00'	pas de traitement particulier
X'1n'	test par rapport à la valeur vide (opérateur n)
X'20'	compactage (suppression automatique de caractères)
X'3n'	test par rapport à la valeur vide et compactage

Exemple d'utilisation :

- pour les index raison sociale et adresse, possibilité de compactage des mots clé.
- possibilité de ne pas générer un index si les valeurs d'une ou plusieurs de ces informations sont égales à celles précisées dans le premier poste.

1.3.2- Fichier des bases - Table des chemins :

Daniel MARTIN suggère de codifier dans un fichier les principales caractéristiques de chaque base telles que :

- liste des programmes autorisés à travailler avec la base
- liste des parcours autorisés sur la base.

Cet ensemble d'informations peut-être subdivisé en deux catégories :

- La première catégorie permet de :
  - réaliser une partie de l'aspect sécurité en filtrant l'accès à la base
  - mettre en place la notion de schéma et sous-schéma (au sens CODASYL du terme)
- La deuxième catégorie d'informations permet d'effectuer des contrôles formels sur le chemin demandé par l'utilisateur. En effet ce chemin suit des règles bien précises :
  - Tout chemin doit commencer par la racine (ou sur le fichier logique pointé si on veut optimiser l'accès)
  - Un chemin ne peut être établi d'un fichier logique à un autre s'il n'existe pas de lien entre ces deux fichiers logiques.
  - Un chemin ne peut-être de structure arborescente.
  - Sur les fichiers de données un chemin est matérialisé par la liste des fichiers logiques concernés. Pour les index secondaires une lettre est attribuée pour chaque index ; elle représente l'accès à la racine, c'est-à-dire à l'index puis au fichier de niveau hiérarchique 1.

Vu l'environnement dans lequel allait être exploité le système de bases de données, il ne nous a pas paru nécessaire de mettre en place un dispositif permettant une sélection des accès à la base de données.

Par contre l'aspect "contrôle" est à retenir car le moyen le plus simple de contrôler qu'un chemin satisfait aux règles énoncées ci-dessus est de vérifier qu'il est l'un des chemins prédéfinis.

Nous avons donc conservé l'idée d'un fichier des bases (en organisation relative) où chaque enregistrement correspond à une base et contient la liste des chemins autorisés pour cette base (classement par ordre alphabétique).

Mais pour des raisons de performance ce fichier ne peut-être qu'un moyen de stockage intermédiaire.

Un programme permet de lire chaque enregistrement base renseigné et de générer un module en langage d'assemblage dont chaque instruction est une déclaration de constante correspondant à un chemin d'une base. Ce module est ensuite assemblé puis après édition des liens catalogué en CIL.

On obtient ainsi une table dont chaque élément est constitué du numéro de base, suivi du chemin. Le premier élément de la table indique le nombre d'éléments qui suivent. La lecture du fichier des bases ayant été effectuée sur ordre croissant des clés, la table reste tirée ce qui autorise une recherche dichotomique.

1.3.3- Table des enregistrements :

C'est la table maitresse du système.

La méthode Daniel MARTIN repose sur la notion d'enregistrement logique et ceci rend très difficile toute généralisation. En effet à enregistrement logique peuvent correspondre plusieurs enregistrements physiques. Lequel faut-il lire ? Lequel faut-il mettre à jour ?

La notion d'enregistrement logique est donc trop floue.

De plus dans un même fichier physique peuvent être mémorisés des enregistrements logiques différents. Lors de l'écriture d'un de ces enregistrements quelle structure logique faut-il lui associer ?

La notion d'enregistrement physique n'est donc pas suffisamment précise. Le concept de fichier et d'enregistrement logique doit cependant être conservé pour l'utilisateur qui ne connaît que des entités.

Nous avons crée une nouvelle notion : celle "d'enregistrement de base de données" ou simplement "d'enregistrement".

On appelle enregistrement un ensemble d'informations ayant des caractéristiques logiques ou physiques spécifiques.

Les caractéristiques de chaque enregistrement nécessaires à la gestion de la base de données sont énumérées ci-dessous. L'étude des différentes fonctions du noyau permettra de mettre en évidence l'utilisation de chacune d'entre elles.

Les informations décrivant un enregistrement sont :

- Numéro de l'enregistrement physique associé :
  (ou numéro de buffer)

  Il permet de sélecter le fichier physique correspondant.

- Numéro de zone pointeur d'accès :

  Les liens étant matérialisés par des pointeurs, il est nécessaire de connaître le numéro de zone qui, dans l'enregistrement "père", pointe sur l'enregistrement. On convient de la valeur zéro pour la racine.

- Numéro de zone pointeur de chainage :

  Dans le cas de fichiers chainés il faut connaître le numéro de zone qui pointe sur l'enregistrement suivant. On convient de la valeur zéro si le fichier n'est pas chainé.

- Facteur de groupage :

  Cette information est uniquement nécessaire si on désire regrouper les enregistrements par blocs.

- Nombre de blocs/piste :

  Même remarque que ci-dessus.

- Numéro enregistrement père :

  Indique le numéro de poste dans cette table qui contient les coordonnées de l'enregistrement "père" s'il s'agit de la racine on convient de la valeur zéro.

- Longueur du buffer :

  Il s'agit de la longueur unitaire si le facteur de groupage est différent de 1.

- Pour un enregistrement de données :

  . Numéro de zone clé complète :

  La redondance peut être admise si c'est dans un but de sécurité.
  Bien que les enregistrements soient en organisation relative et donc identifiés par leur emplacement, il est souhaitable de reproduire dans l'enregistrement une clé constituée par la concaténation des clés des enregistrements de niveau hiérarchique inférieur sur le chemin et de l'enregistrement lui-même. Cette clé est appelée clé complète et un numéro, de zone lui est associé.

. Nombre de zones constituant la clé complète :

Ce nombre n'est pas forcément égal au niveau hiérarchique de l'enregistrement (si on attribue le niveau hiérarchique 1 à la racine).

En effet afin de différencier des enregistrements logiques sur un même support physique il est possible d'insérer dans la clé des constantes qui, nous le verrons plus loin, sont aussi identifiées par un numéro de zone.

. Liste des numéros de zones constituant la clé complète :

Cette liste est limitée à 9 éléments. Elle est constituée par les numéros d'information clé (spécifique) des enregistrements de niveau hiérarchique inférieur sur le chemin et par le numéro d'information permettant d'identifier l'enregistrement lui-même.

- Pour un enregistrement index :

. Numéro de zone clé d'interrogation :

Un numéro de zone est attribué pour qualifier le groupe d'information que l'on doit fournir afin d'accéder à la base lorsqu'on ne connaît pas la clé de l'index principal. Bien entendu les informations pourront être fournies de façon partielle, nous le verrons ultérieurement.

. Nombre de zones constituant l'enregistrement index

. Liste des numéros de zone constituant l'enregistrement index :

Un enregistrement index peut-être constitué d'au plus 9 zones qui appartiennent à un même enregistrement logique.

- Pour tout type d'enregistrement :

. Numéro d'enregistrement extension :

Un enregistrement logique peut être constitué de plusieurs enregistrements physiques de façon à optimiser l'espace disque. Nous limitons à 2 ce nombre d'enregistrements physiques.

Un enregistrement peut donc avoir une extension. Le numéro d'enregistrement extension est le numéro de poste dans la table qui contient les caractéristiques de cette extension. L'absence d'information est codée zéro.

. Numéro de zone pointeur sur extension :

Il indique le numéro de zone qui permet d'accéder à l'enregistrement.

. Déplacement dans la table des enregistrements vides :

Indique l'adresse relative par rapport au début de la table des enregistrements vides de la valeur par défaut de l'enregistrement.

. Indicateur extension :

Il permet de savoir si l'enregistrement est lui même une extension.

- Pour un enregistrement de données :

. Numéro de poste table index de travail :

Cette information n'est pas significative (valeur zéro).

. Numéro de poste table des enregistrements index :

Cette information permet d'indiquer dans la table des enregistrements index la liste des numéros d'enregistrements des index impactés par l'enregistrement.

- Pour un enregistrement index :

. Numéro de poste table index de travail :

Cette information permet d'attribuer un numéro de poste spécifique dans une table de travail servant à la gestion des index. Le nombre de postes de cette table est limité à 3.

. Numéro de poste table des enregistrements index :

Cette information permet d'indiquer dans la table des enregistrements index la liste des numéros de zone constituant la clé d'accès au fichier de données associé.

- Pour tout type d'enregistrement :

. Fichier logique :

Par convention, cette information n'est pas renseignée pour les enregistrements index ce qui permet d'interdire facilement l'entrée par index secondaire pour les autres fonctions que la sélection.

Pour les fichiers de données cette information indique le fichier logique auquel est rattaché l'enregistrement.

. Séquence :

Pour les fichiers chainés on autorise un chainage selon la séquence ascendante ou descendante.

. Organisation :

Pour les fichiers de données il est nécessaire de connaître le type d'organisation (indexée ou directe relative).

. Fichier de données pointé :

Pour les enregistrements index cette information indique quel est l'enregistrement pointé par l'index. De ce fait l'entrée par index secondaire peut se faire à n'importe quel niveau hiérarchique.

. Autorisation de clé en double :

On admet ou non le principe d'avoir à générer un index alors qu'il en existe déjà un de clé identique dans le fichier.

Un module en langage d'assemblage est généré à partir de ces données.
Ce module est ensuite assemblé puis après édition des liens catalogué en CIL.

REMARQUE : Un poste spécial est prévu dans la table ; il concerne les constantes qui, nous le verrons plus loin, sont traitées comme des informations de la base.

1.3.4- Table des fichiers logiques :

Nous avons introduit la notion d'enregistrement qui est utilisée de manière interne au système. Il faut avoir le moyen de retrouver cette notion à partir de celles de fichier logique et de chemin, seules connues de l'utilisateur.

Ce passage s'effectue à l'aide d'une table appelée table des fichiers logiques. Son fonctionnement est le suivant :

A une lettre du chemin elle associe deux valeurs :

S'il s'agit d'un fichier de données ces deux valeurs sont identiques et indiquent le numéro d'enregistrement correspondant à l'enregistrement physique principal du fichier logique concerné.

S'il s'agit d'un fichier index la première valeur indique le numéro d'enregistrement de l'index principal ; la deuxième valeur indique le numéro d'enregistrement de l'index ou l'un quelquonque d'entre eux s'il en existe plusieurs.

En fait cette table est constituée pour une base de deux sous-tables de 256 octets et la fourniture des résultats est basée sur le principe de l'instruction assembleur TR.

Cette table des fichiers logiques est assemblée puis après édition des liens cataloguée en CIL.

1.3.5- Dictionnaire de données - Elixir :

Le dictionnaire de données est le répertoire de toutes les informations de la base qu'elles puissent ou non être mises à jour par les utilisateurs.

Pour chaque information, Daniel MARTIN préconise d'indiquer principalement :

- le numéro de l'information
- la base d'appartenance
- le fichier logique d'appartenance
- le numéro de séquence de l'information qui permet d'obtenir un ordre déterminé lorsqu'on trie le dictionnaire pour édition sur fichier logique
- le fichier physique d'appartenance
- le fichier physique pointé
- le fichier logique pointé
- la zone est-elle une clé ?
- la zone est-elle réelle ou fictive ?
  (On appelle zone fictive une zone qui n'existe pas physiquement mais qui est le résultat d'une opération sur des informations existantes)
  Nous ne traitons que les zones réelles.
- le type (alphanumérique, numérique, pointeur, date)
- la représentation de l'information si celle-ci est numérique (condensé ou non)
- la zone est-elle signée ?
- la zone est-elle sélectable, montrable, totalisable ?
- la position de l'information dans l'enregistrement
- la longueur de l'information
- le nombre de décimale
- le libellé de l'information (50 caractères)
- le libellé réduit (12 caractères)

Il s'avère que les notions de fichier physique pointé, fichier logique pointé ne nous sont d'aucune utilité. Nous les avons cependant conservé dans le dictionnaire car elles permettent de représenter de manière linéaire la structure de la base.

Mais nous avons incorporé deux nouvelles informations à ce dictionnaire :

1- Le numéro d'enregistrement (au sens du terme défini précédemment) auquel l'information est rattachée.

2- Un numéro de poste dans la table des enregistrements index où il est indiqué les numéros d'enregistrements des index impactés par la zone. Ceci permet lorsque la zone est modifiée de répercuter cette modification au niveau de chacun d'entre eux.

De plus, la différenciation que Daniel MARTIN fait concernant l'utilisation des informations (sélectable, montrable, totalisable) ne nous a pas paru justifiée, vu l'environnement dans lequel le système est exploité. En effet, toute information présente dans la base peut être sélectée et montrée aux utilisateurs.

Cependant les informations des index autres que la clé d'interrogation ne peuvent être l'objet de contraintes.

Par contre, nous avons introduit la notion de zone modifiable.

Les pointeurs, par exemple, ne sont pas modifiables.

De ce fait une zone est ainsi caractérisée :

S : sélectable

M : modifiable

T : totalisable (l'information est considérée comme étant également modifiable)

Enfin, nous avons incorporé au dictionnaire des numéros de zones se rapportant à des constantes pour lesquelles il est indiqué un numéro d'enregistrement particulier descriptif des constantes.

Le dictionnaire est stocké dans un fichier en accès direct, chaque enregistrement correspondant à une information de la base.

Nous avons jugé absolument impensable pour des raisons de performances que le noyau accède à ce fichier pour chaque information à traiter. Nous avons donc constitué un dictionnaire de données réduit ou élixir suffisamment concis pour être chargé en mémoire centrale.

Pour chaque information de la base l'élixir indique :

- un pointeur sur le fichier des libellés (2 octets) (ceci n'entraîne pas d'accès disque supplémentaire puisque le noyau n'utilise pas les libellés)

- le numéro d'enregistrement (1 octet)
- les caractéristiques de la zone (1 octet)

  On code sur un bit les notions suivantes :

  constante, réelle-fictive, signe, totalisable, modifiable, sélectable.
- la nature de la zone (1 octet)

  Les 4 bits de gauche permettent de codifier les notions suivantes :

  alphanumérique, numérique, pointeur, date

  Les 4 bits de droite permettent de codifier avec plus de précision la nature de la zone si celle-ci est numérique, à savoir :

  numérique étendu, condensé, binaire 1/2 mot, binaire 1 mot.
- le nombre de décimales (1 octet)
- le déplacement dans l'enregistrement (2 octets)
- la longueur de la zone (2 octets)
- le numéro de poste dans la table des enregistrements index.

Un programme permet de lire les enregistrements du dictionnaire et de générer un module en langage d'assemblage dont chaque instruction est une déclaration de constante correspondant à une information de la base.

Le module généré est ensuite assemblé puis après édition des liens catalogué en CIL. On obtient ainsi une table dont le nombre de poste est égal au nombre d'informations de la base plus un. En effet le premier poste indique le nombre d'éléments de la table qui suivent.

BASE = 01 ********** NUMERO LOGIQUE = 1 NOM = E N T R E P R I S E **********

C.R.I.O.P = L.O.G.I.C DICTIONNAIRE DES INFORMATIONS SEQUENCE = BASE-F.LOGIQUE-INFO 13/01/81 PAGE 1

BA SE	F L	SEQ	INFO	RENS	PH	PH PT	P L I T EN	C R I F	T Y P	I N U	S G	SMT	POS	LNG	D I E N C D	INFO	LIBELLE			TITRE REDUIT
01	1	001	1	0001	01		01		X		S		2	2		1	CODE CAISSE-SOUS-CAISSE			CC
01	1	002	2	0002	01		01		X		S		4	8		2	CODE ENTREPRISE			CO ENTR
01	1	003	3	0003	01		01	C	X		S		2	10		3	CLE ENTREPRISE			CL ENTR
01	1	004	4	0004	01		01		X		S		2	9		4	CLE NO. ENTREPRISE			CL.NO.ENT
01	1	005	5	0005	01		01		X		S		4	7		5	NO.ENTREPRISE			NO.ENT
01	1	006	6	0006	01		01		X		S		11	1		6	CODE GERANT			G
01	1	020	242	0242	01		01		X		SM		179	1		242	CELLULE DE GESTION			C
01	1	095	9	0009	01		01		D		SM		12	2	27	9	DATE D EFFET IDENT.		IDENT	D. IDENT
01	1	100	8	0008	01		01		X		SM		14	75		8	ZONE R.SOC.	R.SOC.	IDENT	Z. R. SOC.
01	1	103	10	0010	01		01		X		SM		14	60		10	R.SOC.	R.SOC.	IDENT	R. SOC
01	1	104	288	0288	01		01		X		SM		14	30		288	R.SOC. LIGNE 1	R.SOC.	IDENT	R.SOC.L1
01	1	105	289	0289	01		01		X		SM		44	30		289	R.SOC. LIGNE 2	R.SOC.	IDENT	R.SOC.L2
01	1	110	11	0011	01		01		X		SM		74	15	1	11	MOT CLE	R.SOC.	IDENT	MOT CLE RSOC
01	1	151	12	0012	01		01		X		SM		89	16		12	ZONE SIGLE	SIGLE	IDENT	Z. SIGLE
01	1	153	14	0014	01		01		X		SM		89	1		14	CODE EDITION	SIGLE	IDENT	E
01	1	154	15	0015	01		01		X		SM		90	15	3	15	SIGLE	SIGLE	IDENT	SIGLE
01	1	174	487	0487	01		01		X		SM		105	24		487	ZONE GERANT			Z.GERANT
01	1	175	292	0292	01		01		X		SM		105	4		292	TITRE (MR. MME. ...)			TIT.
01	1	180	16	0016	01		01		X		SM		109	20	5	16	NOM (GERANT/PROPRIETAIRE)			NOM
01	1	184	301	0301	01		01		X		S		109	15		301	DEBUT NOM POUR INDEX			DEBUT NOM
01	1	202	17	0017	01		01		X		SM		129	9		17	NO. SIREN (9 POS.)			N. SIREN
01	1	210	20	0020	01		01		X		SM		138	19		20	ZONE REGISTRE	REGISTRE		Z. REGISTRE
01	1	211	21	0021	01		01		X		SM		138	1		21	TYPE DU REGISTRE (C/M)	REGISTRE		T
01	1	212	22	0022	01		01		X		SM		139	13		22	NO. DU REGISTRE	REGISTRE		NO REGISTRE
01	1	213	23	0023	01		01		X		SM		152	5		23	NO. VILLE DU REGISTRE			T.C.
01	1	220	24	0024	01		01		N	P	SMT		157	3	0	24	NBR. D ENTREPRISES ABSORBEES			NEA
01	1	230	25	0025	01		01		N	P	SMT		159	3	0	25	NBR D ETABLISSEMENTS SECONDAIRES			NES
01	1	558	70	0070	01		01		X		SM		173	6		70	ZONE CODES CONTRATS/TRAITEMENTS			Z.C/TR.
01	1	560	41	0041	01		01		X		SM		173	3		41	ZONE CODES RET.	CODES RET.		ZCR
01	1	561	42	0042	01		01		X		SM		173	1		42	CODE CONTRAT	CODES RET.		C
01	1	562	43	0043	01		01		X		SM		174	1		43	CODE TRAITEMENT COTIS	CODES RET.		T
01	1	563	44	0044	01		01		X		SM		175	1		44	CODE TRAITEMENT E.N./POINTS	CODES RET.		E
01	1	570	45	0045	01		01		X		SM		176	3		45	ZONE CODES PRV.	CODES PRV.		ZP1
01	1	571	46	0046	01		01		X		SM		176	1		46	CODE CONTRAT	CODES PRV.		C
01	1	572	47	0047	01		01		X		SM		177	1		47	CODE TRAITEMENT PRIMES	CODES PRV.		M
01	1	573	48	0048	01		01		X		SM		178	1		48	CODE TRAITEMENT E.N. PRV.	CODES PRV.		E
01	1	840	50	0050	01	02	2	01	P		S		161	4		50	POINTEUR ETABLISSEMENT PRINCIPAL			P.ET.PR.
01	1	850	51	0051	01	03	5	01	P		S		165	4		51	POINTEUR HIST. IDENTIFICATION			P.H.ID
01	1	860	53	0053	01	07		01	P		S		169	4		53	EXISTANCE LIBRE SERVICE (PTR. L.SERV.)			EXLS
01	1	865	49	0049	01			01	D		SM		200	2		49	DATE DERNIERE MAJ			D.D.MAJ
01	1	870	28	0028	01			01	D		SM		198	2		28	DATE CREATION			D.CREA
01	1	903	54	0054	07			09	X		S		2	10		54	CLE ENTREPRISE LIBRE SERVICE			CL.EN.LSER
01	1	904	27	0027	07			09	X		SM		12	150		27	LIBRE SERVICE	L.SERV.		LIBRE SERV.
01	1	905	285	0285	07			09	X		SM		12	50		285	LIBRE SERVICE ZONE 1	L.SERV.		L.SERV.Z1
01	1	906	286	0286	07			09	X		SM		62	50		286	LIBRE SERVICE ZONE 2	L.SERV.		L.SERV.Z2
01	1	907	287	0287	07			09	X		SM		112	50		287	LIBRE SERVICE ZONE 3	L.SERV.		L.SERV.Z3
01	1	908	52	0052	07			09	X		SM		112	49		52	LIBRE SERVICE PREVOYANCE	L.SERV.		L.SERV.PRV
01	1	909	481	0481	07			09	X		SM		161	1		481	CODE EXISTENCE L. SERV. PRV.	L.SERV.		P.
01	1	931	55	0055	11			11	X		S		2	2		55	CODE CAISSE SOUS CAISSE	INDEX IDENT.		CI
01	1	933	56	0056	11			11	X		S		4	15		56	SIGLE/MOT CLE R.SOC	INDEX IDENT.		S/MCR.S.ID
01	1	935	58	0058	11			11	X		S		19	8		58	CODE ENTREPRISE	INDEX IDENT.		ENT.I.ID
01	1	938	61	0061	11			11	D		S		27	2		61	DATE DE EFFET SIGLE/R.SOC	INDEX IDENT.		D.I.ID

1.3.6- Table des enregistrements index :

Cette table, nous l'avons vu, a plusieurs fonctions.

Elle permet :

(a) - d'indiquer dans l'élixir la liste des numéros d'enregistrements index sur lesquels la zone a un impact.

Elle permet de référencer dans la table des enregistrements.

(b) - la liste des numéros d'enregistrements index sur lesquels l'enregistrement concerné a un impact s'il s'agit d'un fichier de données.

(c) - la liste des numéros de zone constituant l'enregistrement s'il s'agit d'un enregistrement index.

REMARQUE :

Si pour un enregistrement toutes les zones qui ont un impact l'ont sur le ou les mêmes enregistrements index les postes générés par a et c ont un contenu identique. On convient dans ce cas de décrire et référencer un seul et unique poste.

Comme pour la table des enregistrements un module en langage d'assemblage est généré à partir de données. Ce module est ensuite assemblé puis après édition de liens catalogué en CIL.

1.3.7- Table des constantes :

La constitution des enregistrements index et des clés des enregistrements de données peut nécéssiter l'utilisation de constantes, par exemple pour permettre une différenciation d'enregistrement.

Leur présence dans le dictionnaire (fichier logique zéro) permet de les considérer comme des zones de la base de données et de les traiter comme telles.

La table des constantes contient donc l'ensemble des constantes nécessaires. Celles-ci sont rangées de façon contigües et constituent après assemblage et édition de liens un module en CIL.

1.3.8- Le fichier des erreurs :

Un certain nombre de compte-rendus sont fournis par le noyau. Ils se rapportent à des causes diverses telles que :

- requête mal formulée
- requête incohérente (exemple : modification d'enregistrement inexistant)
- requête non terminée pour cause imprévisible (exemple : erreur d'entrée/sortie)
- requête terminée normalement

Le compte-rendu du noyau est codé sur deux positions et la signification de chaque code figure dans un fichier en organisation relative (ce qui autorise une correspondance numéro de code ⟶ emplacement).

2- LA ZONE DE COMMUNICATION

Toute requête au noyau s'effectue en remplissant une zone de communication qui lui est communiquée en paramètre. Cette zone de communication est de longueur variable. Elle est constituée par les informations suivantes :

- NOM DU MODULE EMETTEUR

  Cette information est facultative mais sa présence est souhaitable car elle permet d'identifier plus facilement une zone de communication dans un dump ou un fichier de sauvegarde.

- NUMERO DE BASE

  Nous avons par exemple attribué à notre base entreprise le numéro 01.

- FONCTION

  Les 4 fonctions habituellement disponibles en télétraitement sont : la sélection, l'addition, la modification et la suppression.

  Nous en avons prévu une cinquième : la restauration (remise en service d'un enregistrement supprimé).

  Les fonctions du noyau sont ainsi codifiées :

  S : Sélection

  A : Suppression

  M : Modification

  C : Addition

  R : Restauration

- SOUS-FONCTION

  Pour chaque fonction diverses possibilités sont offertes.

X Les différentes sous-fonctions autorisées pour une sélection sont :

. C : Sélection avec contraintes. L'entrée dans la base se fait à partir de la racine.

  Il peut ne pas y avoir de contraintes. Dans ce cas tous les enregistrements sont sélectés. C'est ce qui correspond à la sous-fonction SELTOUT de Daniel MARTIN que nous n'avons pas jugé utile d'écrire.

. P : Sélection avec pointeur. L'entrée dans la base s'effectue à partir du fichier logique pointé donc à n'importe quel niveau hiérarchique. Il est autorisé d'indiquer des contraintes mais elles doivent porter exclusivement sur les informations présentes dans les fichiers lus.

. S : Sélection du suivant. Lorsque le processus de recherche a déjà été initialisé par une sélection avec contraintes ou une sélection pointeur l'entité suivante est obtenue en appelant le noyau avec la sous-fonction 'S'. Cet appel, peut-être renouvelé jusqu'à ce que le noyau réponde qu'il n'y a plus d'entités correspondant à ce qui a été demandé.

. T : Sélection pour totalisation. Contrairement aux autres sous-fonctions, dans le cas de la totalisation le noyau ne fournit une réponse que lorsqu'il a examiné l'ensemble des entités satisfaisant à la demande et cumulé pour chacune de celles-ci les zones qui lui ont été précisées.

Vu la faible utilité qu'elle présente à ce stade du développement de nos applications, cette sous-fonction n'a pas été mise en oeuvre pour l'instant.

* Les différentes sous-fonctions autorisées pour la modification et la suppression sont :

. K : Sous fonction "clé". La modification et la suppression d'un enregistrement logique s'effectue dans ce cas en fournissant au noyau l'ensemble des clés des enregistrements logiques situés sur le chemin d'accès à l'enregistrement concerné (y compris la clé de l'enregistrement logique à traiter).

. P : Sous-fonction "pointeur". La modification et la suppression d'un enregistrement logique est obtenue en précisant au noyau le pointeur sur l'enregistrement principal de l'enregistrement logique.

* La restauration n'étant autorisée que sur les fichiers à organisation indexée (pour lesquels la création d'un deuxième enregistrement de clé identique ne peut-être envisagée) seule la sous-fonction "clé" est disponible.

* Enfin les sous-fonctions prévues pour le module d'addition sont :

. H : Insertion en tête

. I : Insertion en place

. E : Insertion en fin

En fait seule l'insertion en place est réalisée à ce jour ; les sous- fonctions "en-tête" et "en queue" ne nous ont pas paru indispensables pour l'instant vu qu'elles n'assurent pas un classement des entités chainées.

De plus l'insertion "en queue" pour être réalisée de manière optimisée nécessite la gestion d'un pointeur de fin de liste, ce qui n'est pas prévu.

REMARQUES :

. Bien qu'obligatoire dans tous les cas, le type d'insertion ne pourrait être utile que pour l'addition d'enregistrements dans des fichiers chainés.

. L'addition d'un enregistrement logique s'effectue toujours en fournissant au noyau l'ensemble des clés des enregistrements logiques situés sur le chemin d'accès (y compris la clé de l'enregistrement logique à ajouter)

- CODE REPONSE DU NOYAU

Le compte-rendu du travail effectué par le noyau est codé, on l'a vu, sur deux positions.

La liste de ces codes se trouve en annexe.

Notons que la valeur 00 indique que la requête s'est bien terminée.

- PARCOURS

La notion de parcours est subdivisée en chemin et action.

Le chemin indique quels sont les fichiers logiques de la base qu'on désire lire et l'ordre d'accès. Par convention un chemin doit être complété avec des blancs.

L'action indique le traitement à effectuer sur chaque fichier logique indiqué dans le parcours. (Elle ne concerne que la fonction de sélection)

Deux options sont possibles :

'␣' la recherche est effectuée normalement

' / ' Pour chaque fichier logique pour lequel ce type d'action est précisé, il y a saut sur le fichier logique père après lecture du premier enregistrement logique s'il s'agit de fichiers chainés ou arrêt de la recherche s'il s'agit de l'index principal.

- POINTEUR

Si la sous-fonction est 'P' cette zone contient le pointeur sur l'enregistrement logique concerné.

- POINTEUR-REPONSE

Cette zone ne concerne que la fonction de sélection.

S'il s'agit de la sous-fonction de totalisation le noyau indique à cet emplacement le nombre d'enregistrements logiques pour lesquels un cumul a été effectué. S'il s'agit d'une autre sous-fonction, le noyau indique dans le cas d'un fichier chainé le pointeur sur le prochain élément de la liste (même si celui-ci est logiquement supprimé).

- CODE ABSENCE DE ZONE

Ce code, aussi, est particulier à la sélection. Il peut prendre deux valeurs qui sont 1 et 2.

1 : Au cours de la recherche, si un pointeur vers un enregistrement logique se trouve être à zéro, le dernier enregistrement logique lu n'est pas retenu et la recherche continue.

2 : Au cours de la recherche, si un pointeur vers un enregistrement logique est à zéro, les zones concernant les enregistrements logiques précédemment lus sur le chemin sont délivrées à l'utilisateur (à condition que les contraintes éventuelles sur ces enregistrements logiques soient satisfaites). Les contraintes sur les enregistrements logiques absents ne sont pas examinées et il est fourni la valeur par défaut pour les zones les concernant.

- CONTRAINTES

Elles ne concernent, que la fonction de sélection. Leur nombre peut varier de 0 à 9 ; il est indiqué par l'utilisateur ce qui autorise n'importe quelle valeur par défaut pour les contraintes non renseignées. Elles permettent d'indiquer les caractéristiques des enregistrements logiques recherchés.

Elles sont codifiées sous la forme :

n° de zone, opérateur, code réponse, valeur

Tous les numéros de zones du dictionnaire peuvent être l'objet d'une (ou plusieurs) contrainte(s), sauf les ceux appartenant aux index autres que la clé d'interrogation.

Il y a six opérateurs possibles qui sont :

< ou 1 : $<$

I ou 2 : $\leqslant$

= ou 3 : $=$

S ou 4 : $\geqslant$

> ou 5 : $>$

D ou 6 : $\neq$

La notion de code réponse au niveau d'une contrainte n'est pas préconisée par Daniel MARTIN. Nous l'avons introduite de façon à localiser sans autre appel noyau le ou les fichier(s) concerné(s) dans le cas d'un code réponse "non trouvé".

Il est convenu que les contraintes sont implicitement reliées par l'opérateur booleen 'ET'. L'opérateur 'OU' est matérialisé par un numéro d'information à zéro. De ce fait un enregistrement logique est éliminé si une seule contrainte portant sur celui-ci n'est pas satisfaite. (opérateur ET) ou si l'ensemble des contraintes reliées par l'opérateur 'OU' ne sont pas satisfaites (opérateur OU).

- ZONE CLES

Elle doit être renseignée pour l'addition ou lorsque la sous-fonction clé est précisée. Nous nous sommes limités à 6 niveaux hiérarchiques, de ce fait le nombre de clés à renseigner varie de 1 à 6 selon le chemin. Dans le cas de la sous-base entreprise, le nombre de clés varie de 1 à 3. Les clés non renseignées doivent être à blanc. Pour des facilités d'écriture, il doit être fourni la clé spécifique de l'enregistrement logique et non la clé concaténée. Toute clé doit être complétée par des blancs à droite.

- LISTE DES ZONES

Cas de la sélection :

Il s'agit de la liste des zones à totaliser si la sous-fonction est T et de la liste des zones à montrer dans les autres cas. Les numéros de zone peuvent concerner l'ensemble des fichiers logiques situés sur le chemin.

Cas de l'addition :

Il s'agit de la liste des zones se rapportant à l'enregistrement logique à ajouter. De ce fait aucun numéro de zone ne peut se rapporter à un autre enregistrement logique.

Cas de la modification :

Pour des raisons de sécurité qui ne nous ont pas paru justifiées, Daniel MARTIN préconise la modification d'une seule zone par appel noyeu.

Nous avons donc permis la modification d'un nombre quelquonque de zones concernant un même enregistrement logique.

Cas de la suppression et de la restauration :

L'opération porte sur un enregistrement logique. Il n'y a donc pas lieu de préciser des numéros de zone et s'il en est fourni le noyau les ignore.

REMARQUES :

- Pour les fonctions autres que la sélection le traitement porte toujours sur un seul fichier logique.

- Ce fichier logique est le dernier précisé dans le chemin.

La structure de la liste des informations est la suivante :

- nombre de zones à traiter
- liste des numéros de zones
- liste des indicateurs (un par numéro de zone)
- liste banalisée des informations

La taille de la liste des informations est théoriquement illimitée mais pour des contraintes techniques (sauvegarde à des fins de sécurité) la longueur totale de la zone de communication ne peut excéder 1000 octets s'il ne s'agit pas d'une sélection. Le nombre de zones à traiter est au maximum de 999.

La liste banalisée des informations doit être structurée en accord avec la liste des numéros de zone.

Nous avons introduit la notion d'indicateur pour la raison suivante :

Lorsqu'un utilisateur de transaction modifie des informations sur un écran ceci se traduit par une requête de modification au noyau avec dans la liste des zones les numéros modifiés sur l'écran.

Etant donné que l'intervention de l'opérateur est aléatoire, c'est-à-dire peut concerner n'importe quelle combinaison d'informations, il faudrait prévoir dans la transaction autant de zones de communication avec le noyau que de combinaisons d'informations susceptibles d'être modifiées. Ceci n'est pas pensable et la solution retenue consiste à indiquer dans la liste des zones l'ensemble des numéros.
L'indicateur est mis à la valeur '*' si la zone correspondante est effectivement à traiter. Dans le cas contraire l'indicateur a la valeur "blanc".

Nous avons vu que dans la zone de communication certaines informations sont spécifiques d'une fonction ou d'une sous-fonction. De façon à optimiser la taille de celles-ci il est possible que certaines informations se recouvrent.

Les pages suivantes indiquent les informations à renseigner selon le type de requête et donnent le schéma de la zone de communication.

Fonction	Sélection		Modification		Suppression		Addition	Restauration
Sous-fonction	P	autres sous-fonctions	P	K	P	K		
Module émetteur	X	X	X	X	X	X	X	X
Numéro de base	X	X	X	X	X	X	X	X
Fonction	X	X	X	X	X	X	X	X
Sous-fonction	X	X	X	X	X	X	X	X
Code réponse	O	O	O	O	O	O	O	O
Parcours								
chemin	X	X	X	X	X	X	X	X
action	X	X						
Pointeur	X		X		X			
Pointeur frère ou Nombre d'enregistrements sélectés	O	O						
Absence de zone	X	X						
Contraintes	[X]	[X]						
Clés		X		X		X	X	X
Liste d'informations	X	X	X	X			X	

X : indique que l'information doit être fournie au noyau

O : indique que l'information est restituée par le noyau

[ ] : indique que l'information est facultative

ZONE DE COMMUNICATION NOYAU BASES LOGIC

GRIUP

APPLICATION ..... IDENT ION .....
UNITE DE TRAITEMENT ..... CODE UT .....
AUTEUR ..... DATE 01-08-78

PAGE 1/2

9/ ENREGISTREMENT

	Adresse secteur	MODULE EMETTEUR	NO BASE DES F.C.	S/FCT	REPONSE	PARCOURS CHEMIN 1 2 3 4 5 6	PARCOURS ACTIONS 1 2 3 4 5 6	PTEUR	PTEUR DERN. F. LOG.	9 ZONES DE CONTRAINTES DE 34 CAR. = 306 CAR	ZONE DE TRAVAIL
SELECTIONS	1ER SUR CONTRAINTES			S C							
	1ER SUR POINTEUR			S P				PTEUR			
	SUIVANT			S S							
	POUR TOTALISATION			S T					NBR ENREG. TRAITE SO(S) COMP		
CRÉATION	EN PLACE (CLES)			C K		CHEMIN 1 2 3 4 5 6		6 ZONES CLES DE 30 CAR. = 180 CAR.		FILLER = 138 CAR.	
MODIFICATIONS	SUR CLES			M K		CHEMIN 1 2 3 4 5 6		6 ZONES CLES DE 30 CAR. = 180 CAR.		FILLER = 138 CAR.	
	SUR POINTEUR			M P		F. LOG		PTEUR		FILLER = 314 CAR.	
ANNULATIONS RETAURATION	ANNULATION OU RESTAURA. SUR CLES			A K / R K		CHEMIN 1 2 3 4 5 6		6 ZONES CLES DE 30 CAR. = 180 CAR.			
	ANNULATION SUR POINTEUR			A P		F. LOG		PTEUR			

I-28

ZONE DE COMMUNICATION NOYAU BASES LOGIC

APPLICATION ..... IDENTI ON ..... PAGE 2/2

UNITE DE TRAITEMENT ..... CODE UT .....

AUTEUR ..... DATE 01-08-78

9/ ENREGISTREMENT

Adresse secteur				
DETAIL D'UNE ZONE CONTRAINTE	NO DE ZONE	CONTRAINTE (CADREE A GAUCHE)		
		PIC. IDENTIQUE A LA ZONE DE LA BASE		
DETAIL ZONE DE TRAVAIL	NBR ZONES	NO DES ZONES	CODES VALIDITE	ZONES
	N	LISTE DE N Nº DE ZONES	LISTE DE N CODES	DESCRIPTION DES N ZONES
			VALIDITE CORRESPONDANT A LA LISTE DES NO.	CORRESPONDANT A LA LISTE DES NO.
	NO. 3999	(PIC 3999 COMP)	(PIC X)	
	2	2N	N	

(DANS SELECTION POUR TOTALISATION, N DOIT ETRE EGAL A 30.

(DANS LES AUTRES FONCTIONS, $N \geq 0$

I-29

3- STRUCTURE ET FONCTIONNEMENT DU NOYAU TP

Nous allons étudier maintenant la structure interne et le fonctionnement du noyau. Ce module, nous l'avons vu, est écrit en Cobol-HLPI.
De façon à éviter le temps de chargement lors de son appel, il est déclaré "résident" dans la partition c'est-à-dire toujours présent en mémoire pendant la cession CICS.

Le noyau est composé de 8 parties qui sont :

- adressage des tables et contrôle de la zone de communication
- aiguillage selon le type de requête
- traitement de la requête sélection
- traitement de la requête addition
- traitement de la requête modification
- traitement de la requête suppression
- traitement de la requête restauration
- traitement des Entrées/Sorties

Nous allons examiner chacune d'elle en essayant de mettre en évidence le rôle des tables précédemment étudiées.

3.1- ADRESSAGE DES TABLES ET CONTROLE DE LA ZONE DE COMMUNICATION

Le mécanisme permettant de référencer des zones externes au programme a été étudié dans le chapitre 1.

La liste des tables ainsi adressées est la suivante :

- élixir de données
- table des chemins
- table des enregistrements
- table des enregistrements vides
- table des constantes
- table des fichiers logiques
- table des enregistrements index

D'autre part, le noyau a besoin de conserver des informations d'un appel à l'autre effectuées au cours de la même tâche. Ceci ne peut être fait en utilisant ses propres zones de travail puisqu'à chaque nouvel appel celles-ci sont réinitialisées. L'architecture de CICS offre la possibilité de disposer d'une zone de travail spécifique de la tâche appelée TWA (Task Work Area).

Une commande permet d'acquérir l'adresse de cette zone dont nous verrons l'emploi ultérieurement.

Après la phase d'adressage le noyau effectue un certain nombre de contrôles sur la zone de communication qui lui est soumise dont voici la liste :

1- La longueur de la zone de communication ne peut être inférieure à 400 octets car après sauvegarde le noyau l'utilise comme zone de travail sur une longueur de 400 octets (sauf pour la sélection).

2- La longueur de la zone de communication ne peut excéder 1000 octets (sauf en sélection) car elle est sauvegardée sur un fichier de taille d'enregistrement fixe (voir aspect sécurité).

3- La première position du module émetteur ne peut-être # ou Ø.

4- En sélection le code absence de zone ne peut prendre que les valeurs 1 ou 2.

5- Le numéro de base doit être numérique.

6- Le chemin doit appartenir à la table des chemins.

7- En sélection pour chaque lettre du chemin renseignée, le code action correspondant doit être / ou X.

8- Pour chaque lettre du chemin la table des fichiers logiques doit donner un numéro d'enregistrement.

9- Si ce n'est pas une sélection, on ne peut accéder à la base par un index secondaire donc le nom du fichier logique dans la table des enregistrements ne peut-être à blanc.

10- Pour une sous-fonction pointeur le niveau hiérarchique du fichier logique concerné est au moins 2 puisque les index ne sont pas en organisation relative.

11- En addition, modification et sélection, le nombre de zones à traiter doit être numérique.

12- En addition et modification, le nombre de zones à traiter doit être supérieur à zéro.

13- En sélection, pour chaque contrainte renseignée, l'opérateur doit être compris entre 1 et 6 (au sens large).

14- En sélection, un numéro de zone ne peut être inférieur à zéro ou supérieur au plus grand numéro présent dans l'élixir (les numéros de zones sont consécutifs et leur nombre est situé dans le premier poste de la table).

15- Un numéro de zone doit être défini comme "sélectable" dans l'élixir.

16- En sélection, les contraintes portent au maximum sur un fichier index.

17- Pour les fonctions d'addition et de modification :

- au moins un des indicateurs de zone est renseigné
- les numéros de zones ne peuvent être égaux à zéro ou supérieur au plus grand numéro présents dans le dictionnaire.
- les numéros de zone ne peuvent se rapporter qu'aux fichiers logiques indiqués dans le chemin (également vrai pour la sélection)
- la zone doit être définie comme "modifiable"

18- En sélection, la zone doit pouvoir être totalisée s'il s'agit d'une sous-fonction de totalisation.

19- En sélection si les numéros de zone portent sur un fichier index ce ne peut être que celui éventuellement référencé dans les contraintes et également celui indiqué comme première lettre du parcours.

20- En addition et modification tous les numéros de zone doivent se rapporter à un même fichier logique qui est celui correspondant à la dernière lettre du parcours puisque ces deux fonctions ne traitent qu'un seul fichier logique à la fois.

21- Le code fonction doit être un des cinq codes autorisés.

22- Le code sous-fonction doit être valide pour la fonction concernée.

23- Le pointeur s'il est renseigné doit avoir une valeur plausible c'est-à-dire compris entre 1 et la limite de remplissage du fichier.

24- Pour toutes les fonctions autres que : sélection, modification pointeur et suppression pointeur

- la liste des clés doit être renseignée
- le nombre de clés doit être égal au nombre de lettres du chemin

25- Pour la restauration, le fichier doit être en organisation indexée.

Parallèlement au déroulement des contrôles il y a initialisation d'un certain nombre de variables ou de tables utilisées dans la suite de traitement et qui sont :

- initialisation d'un indicateur I à la valeur # si pour les fonctions autres que sélection le fichier principal sur lequel porte le traitement est en organisation indexée.

- initialisation d'une table des fichiers comprenant un poste par fichiers aux valeurs suivantes :

  0 : le fichier n'est pas traité

  1 : le fichier est concerné par des contraintes

  2 : le fichier est concerné par des zones à montrer ou totaliser

  3 : le fichier est concerné par des contraintes et des zones à montrer ou totaliser

  4 : le fichier doit être mis à jour

- Initialisation d'une table de travail pour la gestion des index.

  Dans le cas de l'addition et de la modification, pour tous les numéros de zone ayant dans l'élixir un numéro de poste dans la table des enregistrements index renseigné, celui-ci est indiqué dans la table de travail. On connaît de cette façon la liste des enregistrements index à créer ou modifier.

  Dans le cas de la suppression ou de la restauration, on renseigne dans cette table le numéro de poste dans la table des enregistrements index (qui se trouve dans la table des enregistrements), pour le numéro d'enregistrement concerné par le traitement.

  Ceci permet d'avoir connaissance de l'ensemble des enregistrements index sur lesquels l'enregistrement à traiter a un impact.

3.2- AIGUILLAGE SELON LE TYPE DE REQUETE

Pour toutes les fonctions autres que la sélection il est nécessaire de localiser l'enregistrement (avec un cas particulier pour l'addition) pour pouvoir le traiter. Plutôt que de multiplier les taches ayant des finalités identiques, il est préférable d'utiliser un outil de localisation existant :
la sélection. Ainsi pour les opérations de mise à jour, le module d'aiguillage effectue préalablement un appel à la sélection.

Celui-ci permet de sélecter l'enregistrement logique s'il s'agit d'une modification, suppression ou restauration et l'enregistrement logique père (ou la racine s'il n'existe pas) dans le cas de l'addition.

En effet, dans ce dernier cas, la recherche de l'enregistrement logique à traiter ne présente aucun intérêt puisqu'il est en principe absent dans le fichier.

Le module d'aiguillage a besoin d'une zone de communication de sélection ; il utilise celle fournie par l'utilisateur après en avoir sauvegardé le contenu.

L'élaboration de la requête de sélection est très simple.

Le chemin est déduit de celui indiqué par l'utilisateur (un niveau hiérarchique de moins pour l'addition).

On ne demande pas à montrer de zones sauf s'il s'agit d'une addition pour un enregistrement logique de niveau hiérarchique supérieur à 1 auquel cas on demande le pointeur d'accès à cet enregistrement.

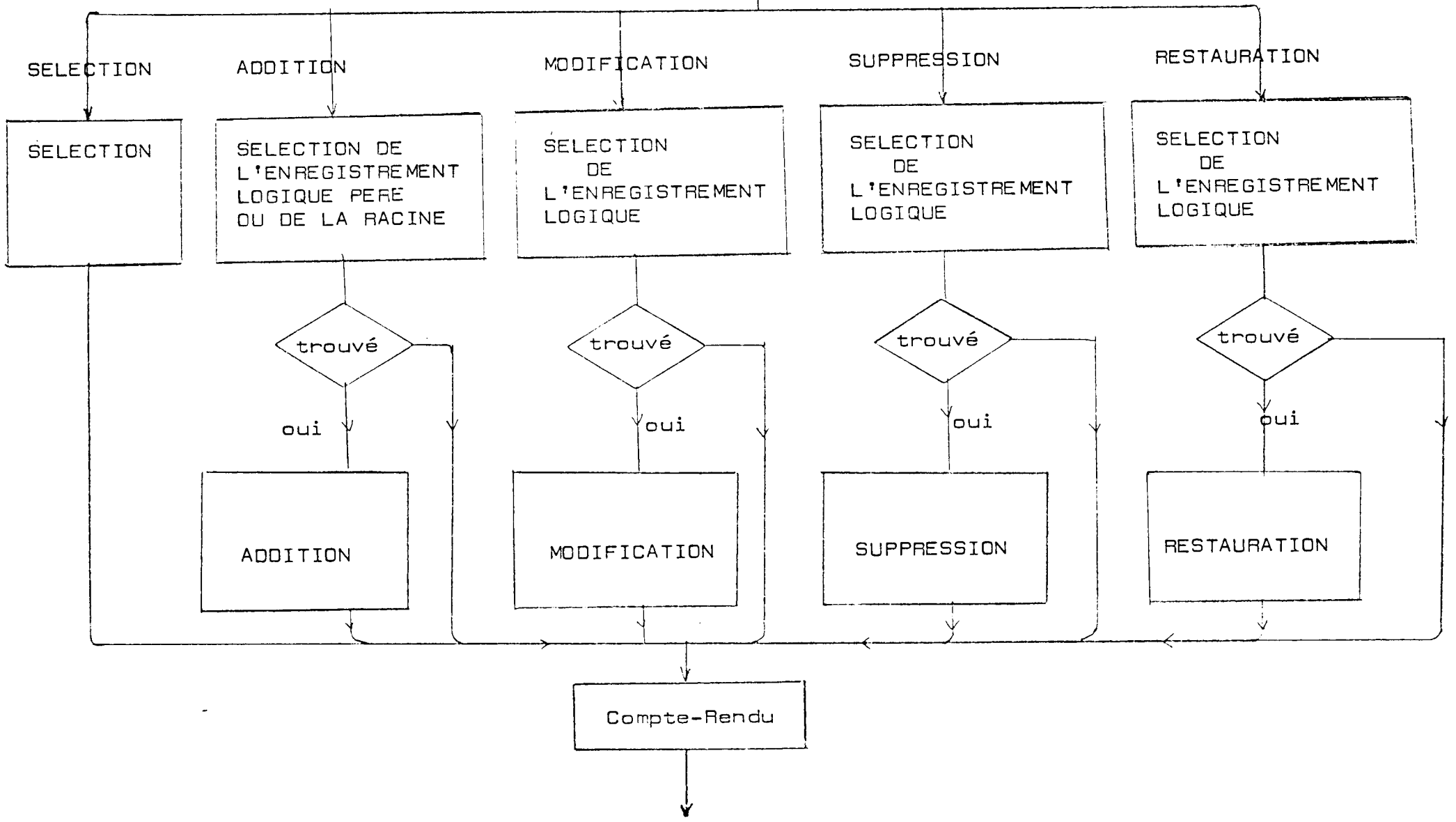
S'il s'agit d'une sous-fonction clé ou d'une addition les contraintes sont construites à partir de la liste des clés de l'enregistrement à traiter de la façon suivante :

- le numéro de zone sur lequel porte la contrainte est le numéro de zone de la clé complète présent dans la table des enregistrements.
- l'opérateur à la valeur 3 (égalité)
- les valeurs de comparaison sont obtenues en expansant les clés fournies par l'utilisateur de façon à obtenir les clés complètes des enregistrements (un niveau hiérarchique de moins pour l'addition).

S'il s'agit d'une sous-fonction pointeur, c'est cette donnée que la sélection utilise pour effectuer la recherche de l'enregistrement logique.

- Enfin la valeur de l'indicateur I est placé dans la première position du module émetteur. Ceci a pour conséquence la lecture avec intention de mise à jour de tous les enregistrements des fichiers en organisation indexée.

La partie aiguillage du noyau peut être schématisée comme suit :



3.4- MODIFICATION D'UN ENREGISTREMENT LOGIQUE

Remarque préliminaire :

Dans les traitements de modification et d'addition on procède à l'acquisition de buffers. Dans le cas de fichiers en organisation directe relative où les enregistrements sont toujours ajoutés en fin, il est nécessaire de connaître le numéro d'enregistrement que l'on crée. Celui-ci s'obtient en ajoutant 1 à la valeur présente en CWA. (Nous rappelons qu'il y a présence, dans cette zone, d'un poste par fichier).

La modification d'un enregistrement logique nécessite, nous l'avons vu, sa localisation préalable par le module de sélection. Celui-ci a effectué des lectures avec intention de mise à jour sur l'enregistrement principal à modifier si celui-ci est en organisation indexée et des lectures simples dans tous les autres cas.

Les différentes phases composant le module de modification sont les suivantes :

3.4.1- Lecture avec intention de mise à jour de l'enregistrement principal si les deux conditions suivantes sont réunies :

- le module de sélection a réalisé pour l'enregistrement une lecture simple.
- la table des fichiers indique que l'enregistrement doit être effectivement mis à jour (déduit de la liste des numéros de zone dans le module d'aiguillage).

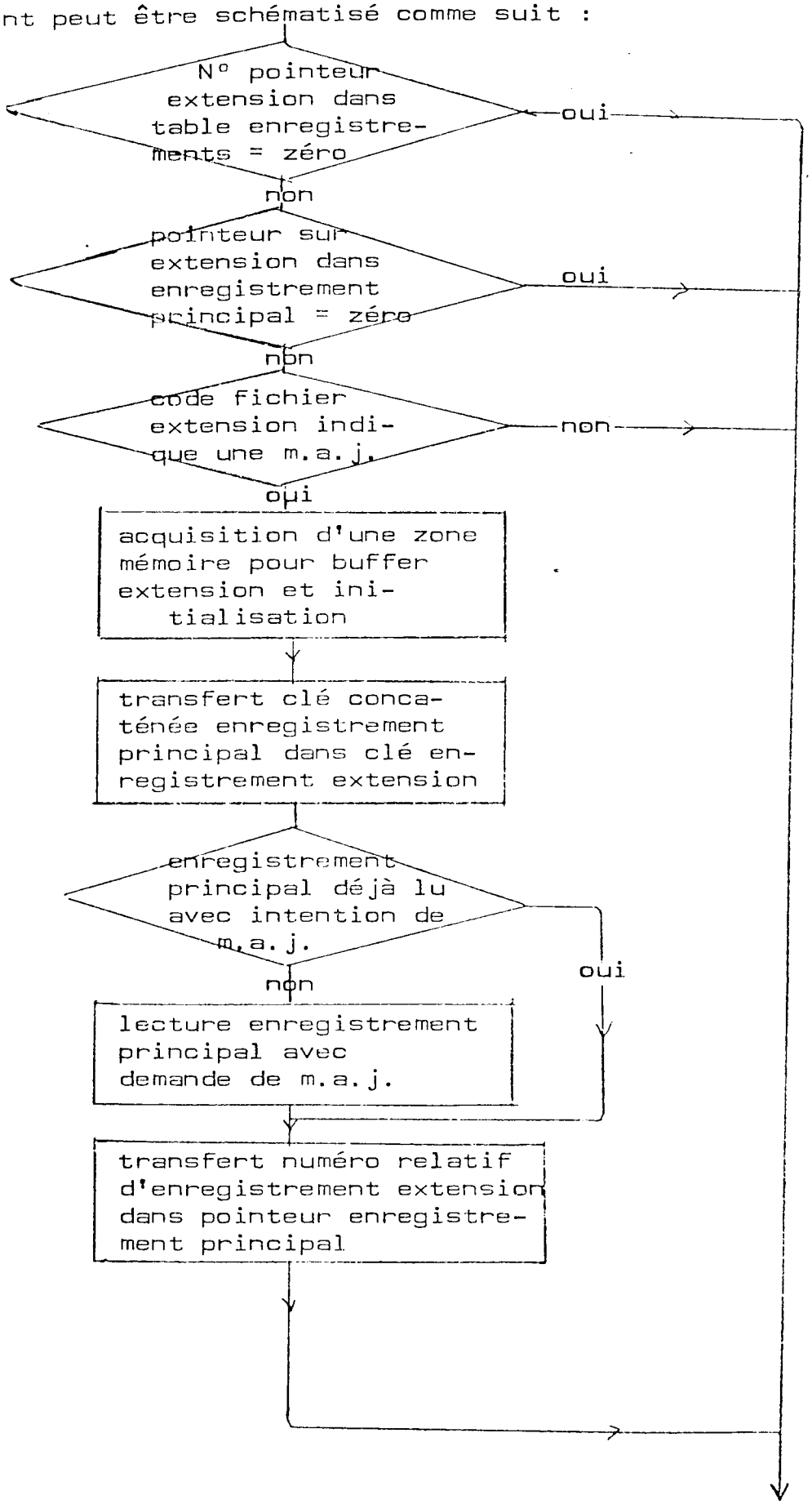
3.4.2- Lecture avec intention de mise à jour de l'extension si les deux conditions suivantes sont réunies :

- elle existe ; c'est-à-dire que le pointeur du fichier principal sur l'extension a une valeur significative.
- la table des fichiers indique que l'enregistrement doit être effectivement mis à jour.

3.4.3- Traitement d'ajout d'extension :

Si l'enregistrement principal est susceptible d'avoir une extension, mais n'en a pas alors que certains numéros de zone s'y rapportent, il faut initialiser un buffer de ce type et mettre à jour dans l'enregistrement principal le pointeur sur cette extension.

Ce traitement peut être schématisé comme suit :



3.4.4- Suppression des index secondaires existants :

Une modification de zone peut entraîner, si celle-ci a un impact sur des index secondaires, la modification de ces index. Plutôt que de procéder à une modification d'enregistrement index, il est réalisé une suppression suivie d'une récréation.

La première étape consiste donc à reconstituer les clés des enregistrements index à partir des anciennes valeurs c'est-à-dire avant toute modification, de façon à pouvoir donner des ordres de suppression de ces enregistrements. Ce travail de reconstitution s'effectue à l'aide de la table des enregistrements index, de la table des enregistrements et de l'élixir de la façon suivante :

- L'élixir donne pour chaque numéro de zone ayant un impact sur un (ou des) enregistrement(s) index, un numéro de poste dans la table des enregistrements index.

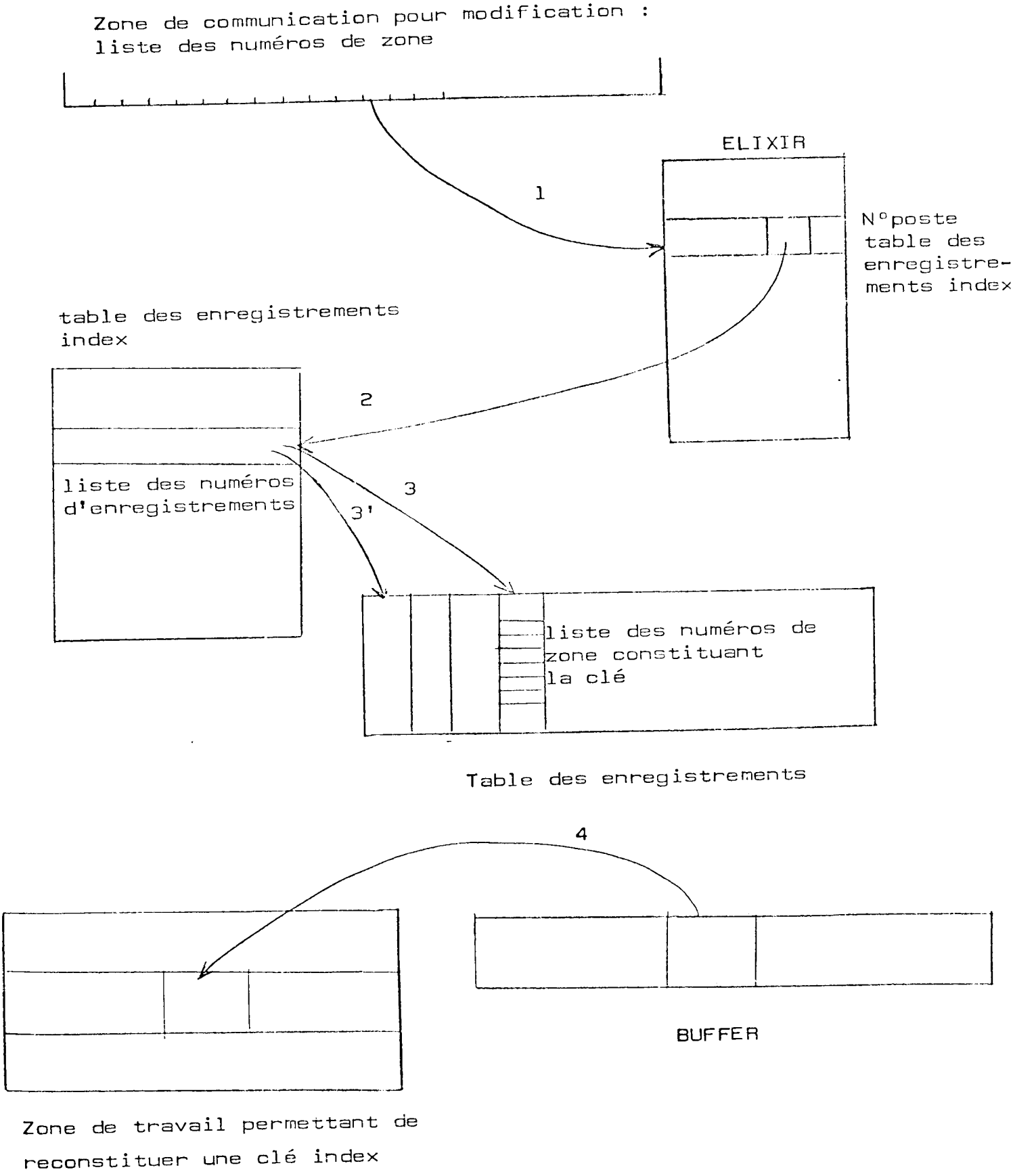
- Ceci permet d'obtenir les numéros d'enregistrement des index à supprimer.

- Or dans la table des enregistrements, il est précisé pour les enregistrements index la liste des numéros de zone constituant l'enregistrement donc la clé.

- A l'aide de cette liste les valeurs présentes dans le buffer de données sont transférées dans une zone de travail de façon à recomposer la clé de l'index.
  Il suffit ensuite, connaissant le numéro de l'enregistrement index et la clé d'effectuer une suppression physique de cet enregistrement.

Actuellement il est prévu que l'ensemble des zones modifiées au cours d'un appel noyau puisse avoir un impact sur trois enregistrements index au maximum, mais cette limite est arbitraire et pourrait être augmentée très facilement

Le cheminement dans les tables que nous avons décrit ci-dessus peut être schématisé comme suit :

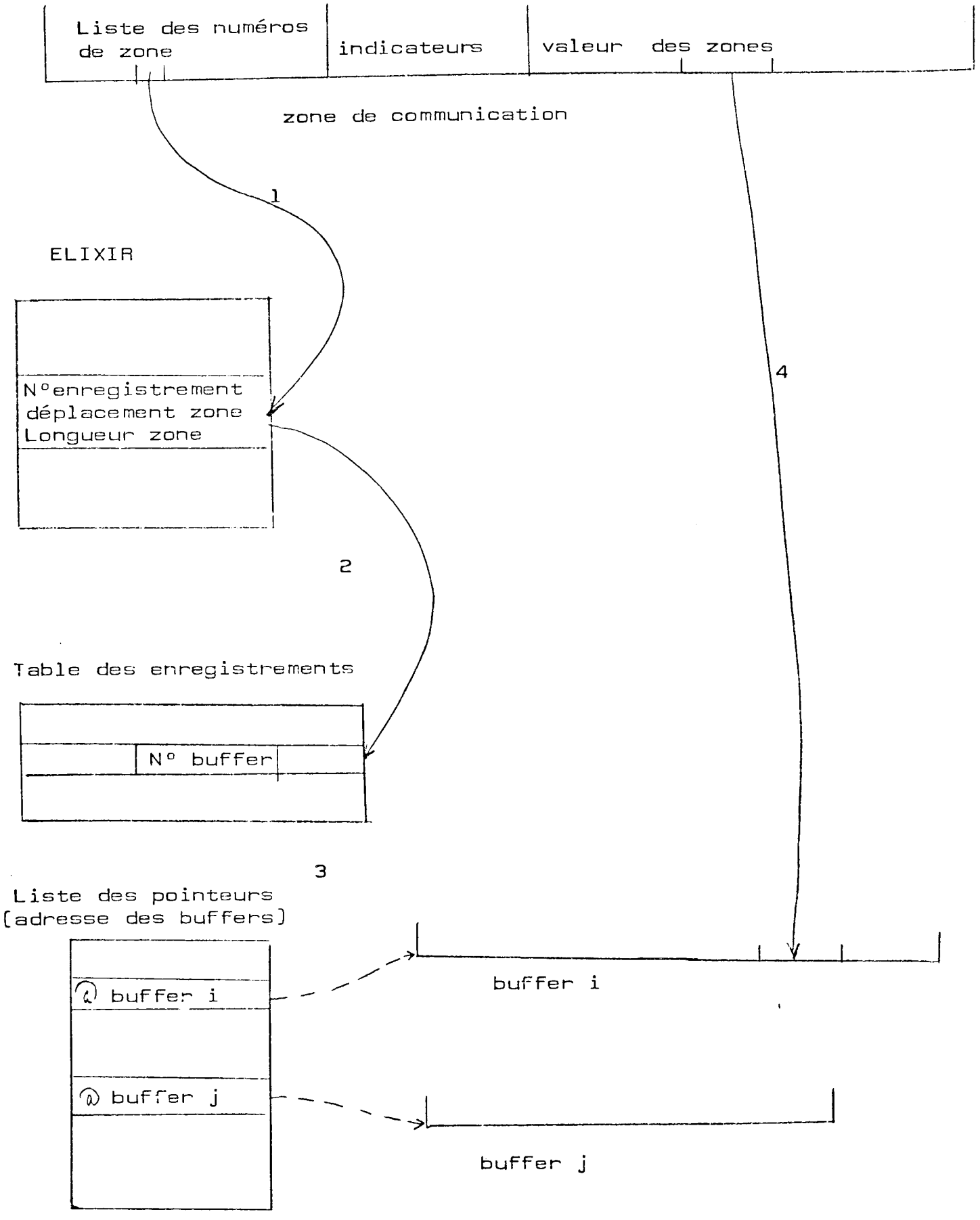

Zone de communication pour modification :
liste des numéros de zone
ELIXIR
1
N°poste
table des
enregistre-
ments index
table des enregistrements
index
2
liste des numéros
d'enregistrements
3
3'
liste des numéros de
zone constituant
la clé
Table des enregistrements
4
BUFFER
Zone de travail permettant de
reconstituer une clé index

3.4.5- Mise à jour de l'enregistrement logique :

Les buffers ont été lus ou acquis et initialisés (cas de création de l'extension). Ils sont prêts à recevoir les nouvelles valeurs précisées par l'utilisateur. Ces informations sont transférées de la zone de communication dans les buffers selon le processus suivant (qui est réalisé par un sous-programme) :

- Pour chaque numéro de zone présent dans la zone de communication on consulte l'élixir pour connaître :
  - le numéro d'enregistrement auquel appartient la zone
  - le déplacement de la zone par rapport au début de l'enregistrement
  - la longueur de la zone
- Le numéro d'enregistrement permet d'obtenir, par l'intermédiaire de la table des enregistrements le numéro de buffer.
- Ce dernier permet de connaître l'adresse du buffer dans la table des pointeurs (les numéros de pointeur correspondant aux numéros de buffer).
- En ajoutant l'adresse du buffer et le déplacement de la zone par rapport au début du buffer, on obtient l'adresse de la zone et il est alors possible d'en transférer sa valeur dans la base.

Ce processus de mise à jour peut être schématisé comme suit :


Liste des numéros de zone
indicateurs
valeur des zones
zone de communication
1
ELIXIR
N°enregistrement
déplacement zone
Longueur zone
4
2
Table des enregistrements
N° buffer
3
Liste des pointeurs
(adresse des buffers)
buffer i
buffer j
buffer i
buffer j

3.4.6- Traitement d'annulation d'extension :

Il se peut que la modification d'informations de l'extension conduise à générer un enregistrement de valeur nulle c'est-à-dire égal à sa valeur par défaut. Ceci peut se produire dans le cas d'annulations d'informations existantes ou dans le cas moins probable de modifications d'informations ayant pour valeur leur valeur par défaut et conduisant à la création d'un enregistrement extension.

Dans ce cas il est parfaitement inutile de conserver des informations nulles puisqu'elles sont présentes par ailleurs dans la table des enregistrements vides.

Ceci présente deux inconvénients majeurs :

1- Occupation d'emplacement disque superflue

2- Temps de réponse accru lors de la sélection de l'enregistrement logique (puisqu'il y a une lecture supplémentaire).

Ce problème est résolu de la façon suivante :

Si la valeur du pointeur vers l'extension est significative et si le code fichier correspondant est positionné il y a comparaison entre l'enregistrement et sa valeur par défaut.

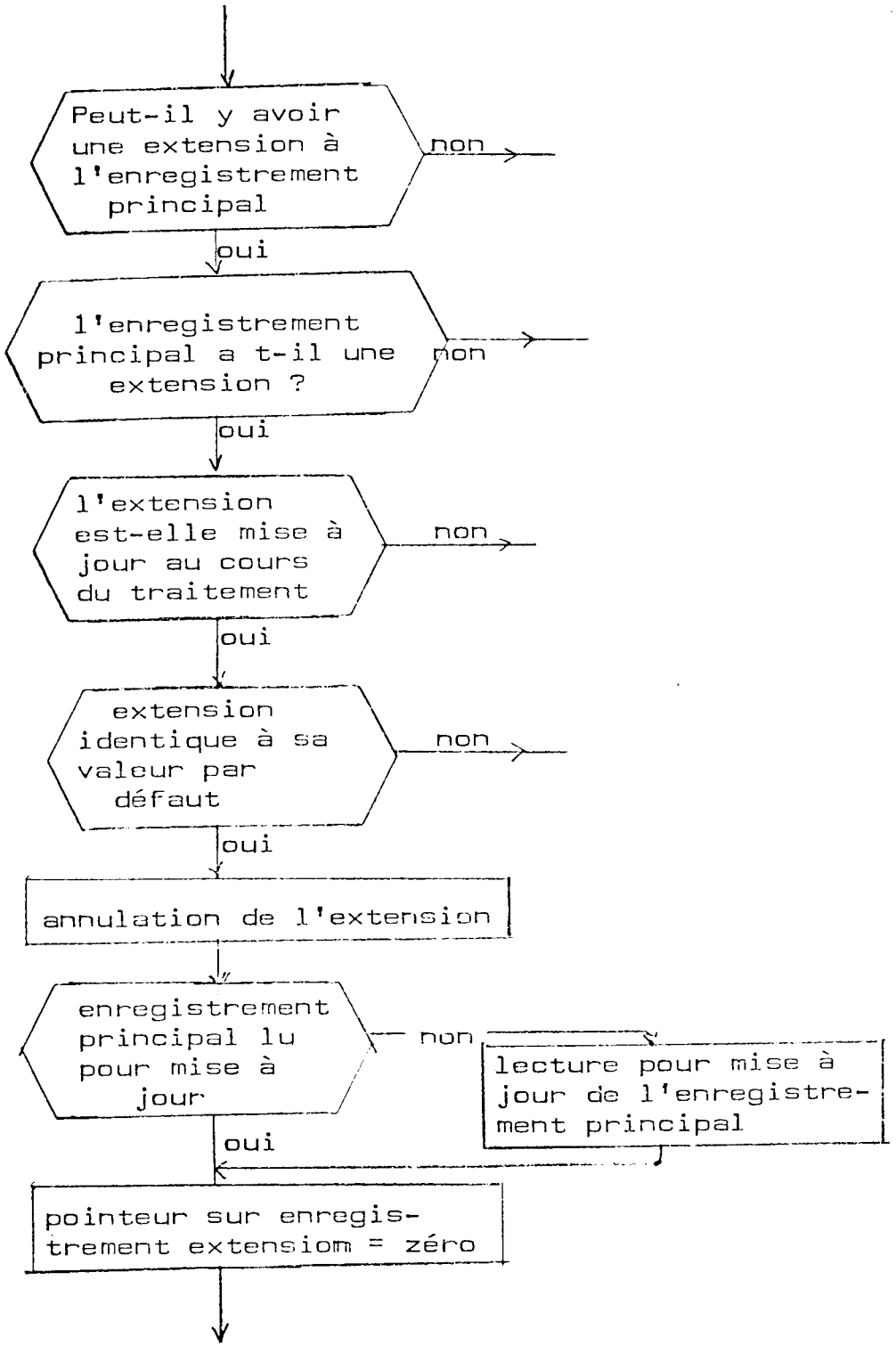
Dans le cas où il y a identité, le traitement suivant est effectué :

- La première position de l'enregistrement extension reçoit la valeur '*' ce qui signifie que l'enregistrement est logiquement supprimé (voir module de suppression) et de ce fait sera éliminé au cours de la prochaine réorganisation de la base (résolution du point N° 1 ci-dessus)

- Si le code fichier concernant le fichier principal indique que celui-ci n'est pas traité (et de ce fait n'a pas été lu avec intention de mise à jour) il est effectué une lecture pour mise à jour de l'enregistrement principal et le code fichier est positionné.

- Le pointeur de l'enregistrement principal vers l'extension est mis à la valeur zéro (résolution du point N° 2 ci-dessus).

Ce traitement d'annulation d'extension peut être schématisé comme suit :


Peut-il y avoir une extension à l'enregistrement principal
non
oui
l'enregistrement principal a t-il une extension ?
non
oui
l'extension est-elle mise à jour au cours du traitement
non
oui
extension identique à sa valeur par défaut
non
oui
annulation de l'extension
enregistrement principal lu pour mise à jour
non
lecture pour mise à jour de l'enregistrement principal
oui
pointeur sur enregistrement extensiom = zéro

3.4.7- Récréation des index secondaires :

Une fois les buffers mis à jour avec les buffers fournis par l'utilisateur il est nécessaire de régénérer les nouveaux enregistrements index.

Le processus de fabrication de ceux-ci est rigoureusement identique à ce qui est décrit dans le paragraphe 3.4.4.

On obtient dans une zone de travail un ensemble de couples (3 au maximum) constitués chacun d'un numéro d'enregistrement et de son contenu.

Par convention la clé est identique au début de l'enregistrement (sur une longueur indéterminée).

Ces éléments permettent d'effectuer une écriture des nouveaux enregistrements index.

3.4.8- Ecriture des enregistrements de données :

Tous les enregistrements de données dont le code fichier correspondant est positionné sont traités.

Ils sont respectivement écrits ou réécrits selon qu'ils ont été lus ou acquis au cours du traitement.

3.5- SUPPRESSION D'UN ENREGISTREMENT LOGIQUE

Daniel MARTIN préconise de ne pas effectuer de suppressions physiques mais seulement des suppressions logiques qui consistent à conserver l'enregistrement en le marquant comme invalide. Ceci permet d'en garder la trace quelques jours ou quelques mois jusqu'à la prochaine réorganisation où sa marque d'invalidité entraîne alors son effacement effectif.

Cette méthode peut, selon Daniel MARTIN, présenter un intérêt si un utilisateur effectue une requête de suppression par erreur et s'en aperçoit suffisamment rapidement.

Nous avons pris cette solution bien qu'elle présente une contrainte au niveau des fichiers d'organisation indexée.

En effet, la suppression d'un enregistrement sur un fichier en organisation relative ne présente aucun problème puisque la clé spécifique de l'enregistrement n'est pas la clé d'accès.

Ceci n'est pas le cas pour les fichiers d'organisation indexée et, étant donné que la marque d'invalidité ne fait pas partie de la clé, il n'est pas possible de recréer un enregistrement de même clé qu'un enregistrement logiquement (mais non encore physiquement) supprimé. Nous verrons comment ce problème est résolu dans l'étude du module de restauration d'un enregistrement logique.

Etudions maintenant le fonctionnement du module de suppression.

Comme pour les autres modules, l'enregistrement logique à traiter a été au préalable localisé par la sélection.

Ensuite, le traitement suivant est réalisé :

3.5.1- Lecture avec intention de mise à jour de l'enregistrement principal si la sélection a réalisé pour celui-ci une lecture simple

3.5.2- Suppression de l'enregistrement principal, c'est-à-dire transfert du caractère '*' en première position du buffer (marque d'invalidité)

Le mécanisme permettant d'effectuer cette opération représente une partie de celui décrit dans la phase de transfert des informations du module de modification (cheminement numéro d'enregistrement ⟶ numéro de buffer ⟶ table des pointeurs ⟶ adresse du buffer).

3.5.3- Traitement de l'extension :

Si une extension existe il est nécessaire de la supprimer également.

La démarche suivante est alors effectuée :

- On recherche si l'enregistrement est susceptible d'avoir une extension.
- Si c'est le cas, on teste la valeur du pointeur sur cette extension par rapport à sa valeur par défaut.
- On effectue une lecture de l'enregistrement extension avec demande de mise à jour (si le pointeur a une valeur significative).
- On transfère dans le buffer extension la marque d'invalidation par le même mécanisme que pour l'enregistrement principal.
- On indique dans la table des codes fichiers que l'enregistrement a été traité.

3.5.4- Traitement des enregistrements index :

Nous avons vu que dans la table des enregistrements figure un numéro de poste dans la table des enregistrements index.

Si ce numéro de poste est renseigné pour l'enregistrement principal on recherche dans le poste concerné de la table des enregistrements index, la liste des numéros d'enregistrements index sur lesquels l'enregistrement à supprimer a un impact.

Pour chaque numéro d'enregistrement index, on reconstitue dans une table de travail son contenu (et donc sa clé) à l'aide de la liste des numéros de zone présente dans la table des enregistrements.

Cas d'un enregistrement de données en organisation indexée :

Les enregistrements index sont ensuite lus avec intention de mise à jour. On transfère dans la première position du buffer la marque d'invalidation, puis ils sont réécrits.

Cas d'un enregistrement de données en organisation relative :

Les enregistrements index sont supprimés physiquement.

Ce mécanisme est calqué sur celui décrit dans le traitement d'annulation des enregistrements index du module de modification sauf le premier point (détermination du numéro de poste dans la table des enregistrements index).

3.5.5- Ecriture des enregistrements de données :

Tous les enregistrements dont le code fichier est positionné, c'est-à-dire l'enregistrement principal et son extension si elle existe, sont réécrits.

REMARQUES :

1- Le traitement de suppression d'un enregistrement logique s'effectue sans vérification de l'existence d'enregistrements fils éventuels. Ceci est donc à la charge de l'utilisateur.

2- Des enregistrements non supprimés dépendant d'un enregistrement logique supprimé sont inaccessibles et au cours du prochain traitement de réorganisation de la base, ils seront listés et supprimés.

3- D'après la description des traitements de modification et de suppression, il s'en suit qu'un enregistrement index ne peut être généré qu'à partir d'un seul enregistrement logique et de façon plus stricte de l'enregistrement principal qui le constitue.

3.6- RESTAURATION D'UN ENREGISTREMENT LOGIQUE

Nous avons vu dans la description du module de suppression qu'un enregistrement logique supprimé dont l'enregistrement principal est en organisation indexée, ne pouvait être recréé.

Dans le cas où la suppression a été demandée par erreur, ceci peut être contraignant pour l'utilisateur. Le module de restauration permet justement d'effacer la marque d'invalidité d'un enregistrement logique supprimé et des index qui en dépendent, c'est-à-dire de le "remettre en fonction".

Il est conçu de la façon suivante :

Dans le module d'aiguillage il est fait appel à la sélection de façon à localiser l'enregistrement logique à traiter.

Puis l'opération de restauration proprement dite se décompose comme suit :

3.6.1- Restauration de l'enregistrement principal c'est-à-dire transfert du caractère 'blanc' en première position du buffer pour supprimer la marque d'invalidité (Cet enregistrement principal a été lu avec intention de mise à jour par la sélection puisqu'il appartient forcément à un fichier d'organisation indexée)

3.6.2- Traitement de l'extension :

S'il existe une extension, il est nécessaire de la restaurer également.

La démarche effectuée est du même type que pour le module de suppression, à savoir :

- Peut-il y avoir une extension ?
- Si oui le pointeur sur l'extension a-t-il une valeur significative ?
- Si le pointeur est renseigné, lecture de l'enregistrement extension avec intention de mise à jour.
- Remise à blanc de la première position du buffer extension.
- Indication de traitement de l'enregistrement dans la table des codes fichiers.

3.6.3- Traitement des enregistrements index :

Nous avons vu dans l'étude du module de suppression que les enregistrements index sont réécrits avec le caractère d'invalidation.

La fonction de restauration consiste donc au niveau des index en une réécriture avec un blanc en première position.

Le mécanisme permettant d'effectuer ce traitement est identique à celui utilisé dans le module de suppression.

3.6.4- Ecriture des enregistrements de données :

Tous les enregistrements dont le code fichier est positionné, c'est-à-dire l'enregistrement principal et son extension éventuelle sont réécrits.

3.7- ADDITION D'UN ENREGISTREMENT LOGIQUE

On pourrait envisager, comme pour les autres fonctions, d'effectuer, dans le module d'aiguillage, une requête de sélection afin de localiser l'enregistrement à traiter. Ceci n'est guère réaliste puisque dans la grande majorité des cas l'enregistrement logique à créer sera déclaré "non trouvé" et il faudrait faire un deuxième appel avec un numéro hiérarchique de moins. (Le code absence d'enregistrement ne permet pas de faire l'économie de ce deuxième appel dans le cas de fichiers chainés et de liste non vide).

Il faut donc demander à la sélection de localiser l'enregistrement logique père, qui lui a toutes les chances d'exister, et de mettre en évidence le lien vers le fichier logique sur lequel doit porter l'opération d'addition.

Il est à noter que dans le cas de la racine, on effectue la recherche de l'enregistrement lui-même qui, sauf cas d'erreur, n'existe pas. Ceci permet de déterminer, à priori, la cohérence de la requête d'addition.

En partant de ce principe voyons comment est conçu le module d'addition.

3.7.1 - Cas de la racine :

Nous avons vu que le module d'aiguillage a fait appel au module de sélection afin de vérifier la non-existence de l'enregistrement logique.

Nous savons donc avec certitude qu'à ce niveau du traitement la création de l'enregistrement peut être effectuée. Les différentes phases de ce traitement sont :

3.7.1.1- Acquisition de buffers concernant l'enregistrement logique d'après les codes fichiers (un buffer pour l'enregistrement principal et éventuellement un buffer pour l'extension).

3.7.1.2- Transfert des données dans les buffers acquis (même mécanisme que pour la modification)

3.7.1.3- Génération d'une clé et transfert dans le buffer principal :

Cette tache est ainsi réalisée.

On se crée une zone de travail de structure identique à celle de la zone de communication mais concernant une seule information à traiter.

Le numéro d'information reçoit le numéro de zone clé complète de l'enregistrement. L'indicateur de validité indique que l'information est à traiter.

La valeur de l'information est rendue égale à la clé du niveau hiérarchique correspondant présente dans la zone de communication.

Cette zone de travail est traitée de façon standard afin d'effectuer les transferts désirés.

3.7.1.4- Traitement particulier à l'extension :

Dans le cas où l'enregistrement est susceptible d'avoir une extension et que les informations à créer s'y rapportent, un buffer a été acquis à cet effet et le transfert de ces informations a été réalisé.

x Il convient alors d'établir un lien entre les deux buffers et de placer dans l'enregistrement extension la clé de l'enregistrement principal.

Le premier point est réalisé de la manière suivante :

. Obtention dans la table des enregistrements du numéro de zone "pointeur d'accès" de l'enregistrement extension.

. Recherche de l'occurence attribuée à cet enregistrement dans le fichier et transfert de cette valeur dans la zone pointeur d'accès à l'extension.

Le deuxième point est réalisé comme suit :

. Obtention du numéro de zone clé complète de l'enregistrement principal (table des enregistrements).

. Obtention de la valeur de cette information.

. Obtention du numéro de zone clé complète de l'enregistrement extension (table des enregistrements)

. Transfert dans cette zone de la valeur obtenue précedemment.

x Cas particulier où le buffer de l'extension est égal à sa valeur par défaut (à l'exception de la clé).

Si la comparaison de la valeur du buffer avec sa valeur par défaut dans la table des enregistrements vides indique qu'il y a égalité, il y a :

. Transfert du caractère d'invalidation dans la première position du buffer extension.

. Mise à zéro du pointeur d'accès à l'enregistrement extension.

3.7.1.5- Traitement des enregistrements index :

Ce traitement consiste à générer dans une table de travail des enregistrements index à partir du fichier de données en cours de création ; il n'est effectué que s'il est indiqué dans l'élixir que les informations créées ont un impact sur des enregistrements index.

Le processus de remplissage de la table de travail est identique à celui décrit dans le paragraphe 3.4.4- du module de modification.
Puis tous les enregistrements index dont le contenu a été constitué dans la table sont écrits.

3.7.1.6- Ecriture des enregistrements de données :

Les enregistrements de données dont le code fichier correspondant est positionné sont écrits. Il s'agit de l'enregistrement principal et éventuellement de son extension.

3.7.2- Cas du premier enregistrement sur un fichier logique autre que le premier niveau

Il s'agit uniquement de fichiers non chainés ou de création d'un premier "maillon" sur un fichier chainé. Le cas de l'insertion en tête sera étudié ultérieurement.

Ici le module d'aiguillage fait appel au module de sélection afin de sélecter le père de l'enregistrement logique à traiter.

La sélection a permis de connaître la valeur du pointeur de l'enregistrement père vers un enregistrement fils éventuel.

Plaçons nous dans le cas où cette valeur est zéro, c'est-à-dire absence de lien.

La procédure d'ajout d'un tel enregistrement logique varie légèrement par rapport à celle d'une racine.

On peut la résumer comme suit :

3.7.2.1- Acquisition de buffers concernant l'enregistrement logique d'après les codes fichiers

3.7.2.2- Transfert des informations dans les buffers acquis

3.7.2.3- Génération d'une clé et transfert dans le buffer principal

3.7.2.4- Relecture avec intention de mise à jour de l'enregistrement père

(Au cours de la première lecture de l'enregistrement la clé a été sauvegardée)

3.7.2.5- Création du lien père-fils

Cette opération consiste à transférer dans l'information pointeur de l'enregistrement "père" la valeur de l'occurence dans le fichier de l'enregistrement principal en cours d'addition.

3.7.2.6- Traitement particulier à l'extension

3.7.2.7- Traitement des enregistrements index

3.7.2.8- Ecriture des enregistrements de données

3.7.3- Cas d'insertion dans un fichier chainé :

Ce mécanisme est basé sur la conservation de clés triées en parcourant le chaînage.

Nous nous plaçons dans le cas où la sélection de l'enregistrement père a permis de déterminer l'existence d'un lien vers un enregistrement logique "fils" (pointeur renseigné).
Le travail consiste dans ce cas à déterminer à quel endroit du chaînage doit se faire l'insertion.

(Bien entendu il faut vérifier que le fichier sur lequel porte le traitement est bien chaîné, car le cas contraire peut être assimilé à une tentative de création d'un enregistrement déjà existant).

La progression sur le chaînage s'effectue au moyen d'appels successifs à la sélection à laquelle on fournit le pointeur de l'enregistrement à sélecter et on demande la clé complète de l'enregistrement sélecté et la valeur du pointeur vers l'enregistrement suivant.

Dans le cas où la clé complète obtenue est égale à la clé de l'enregistrement que l'on veut ajouter il s'agit d'une anomalie.

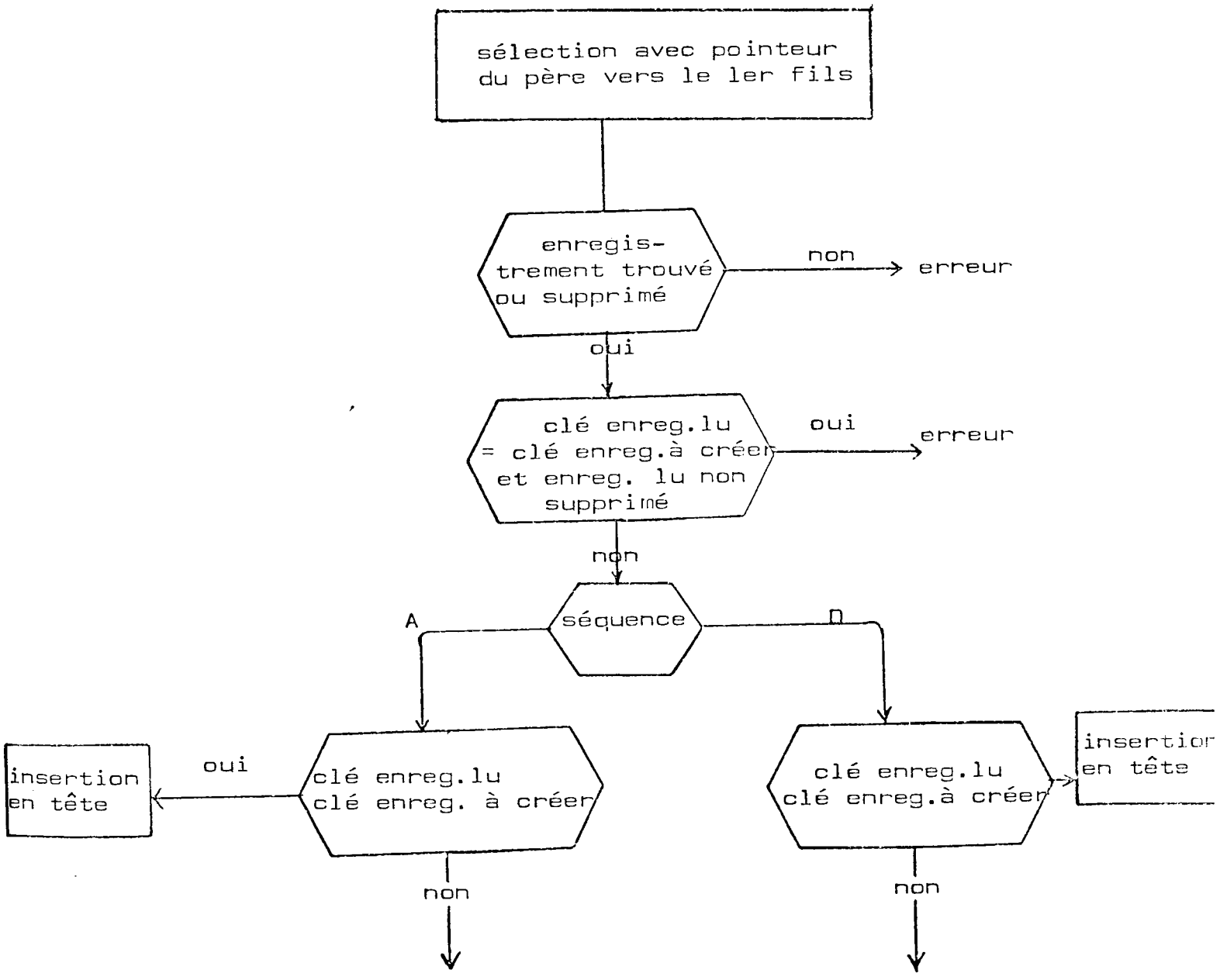
Sinon la comparaison des deux clés doit se faire en tenant compte de la séquence du fichier précisée dans la table des enregistrements.

Suite à la première sélection deux cas (un pour chaque type de séquence) permet de s'orienter vers une insertion en tête.

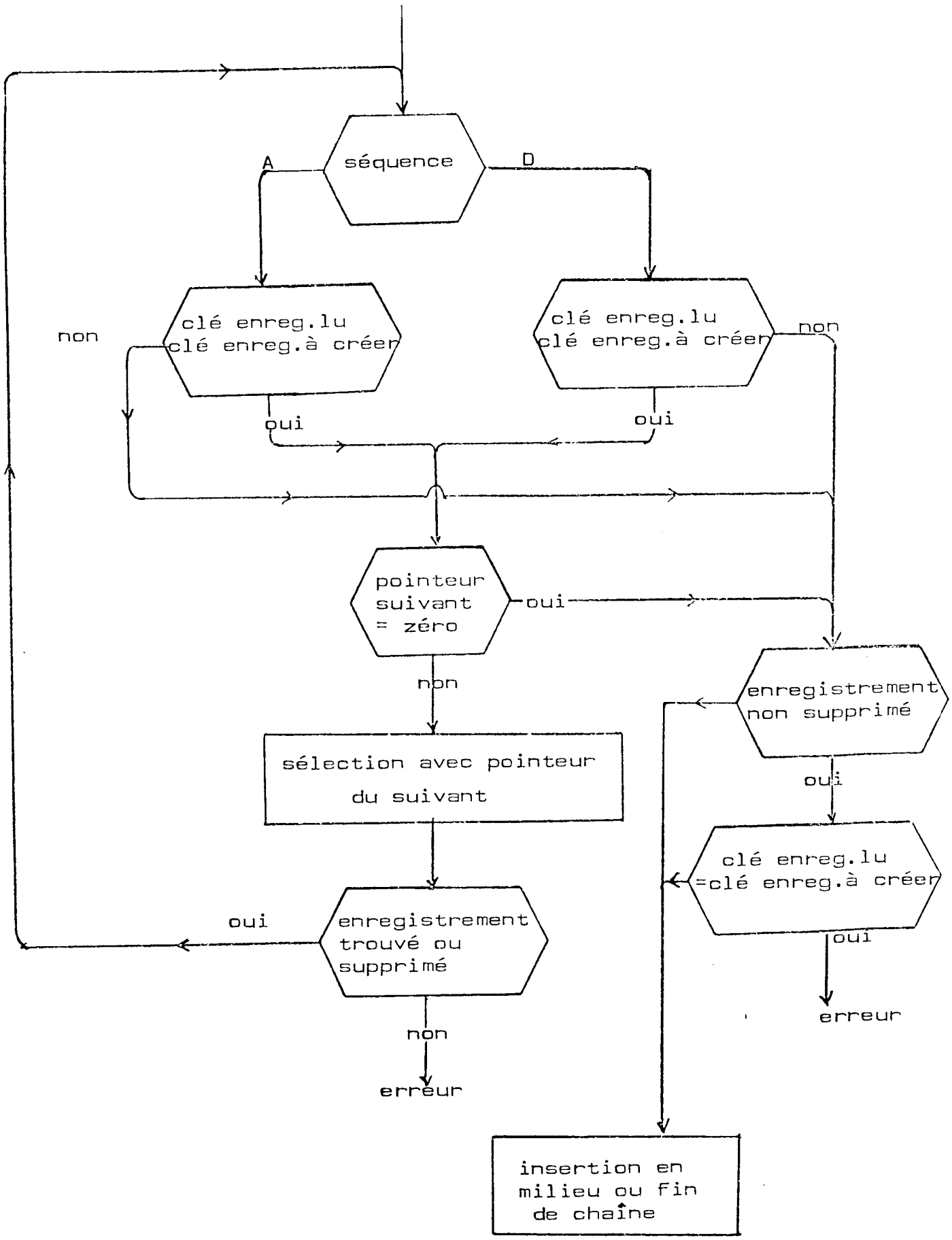
Les autres cas entraînent toujours une insertion en milieu ou fin de chaîne.

La page suivante matérialise l'algorithme de localisation de l'endroit du chaînage où doit s'effectuer l'insertion.

CAS DE LA PREMIERE SELECTION



CAS DES SELECTIONS SUIVANTES



3.7.3.1- Insertion d'un enregistrement "en tête " :

Ce processus comprend toutes les phases de celui décrit précédemment.

Il permet de plus d'effectuer le lien entre l'enregistrement en cours de création et l'enregistrement qui était en tête initialement.

Ce point est réalisé de la façon suivante :

. Obtention du numéro d'information du pointeur de chaînage dans la table des enregistrements.

. Transfert dans cette zone du numéro de l'enregistrement qui était en tête initialement (dernier enregistrement sélecté).

3.7.3.2- Insertion d'un enregistrement en milieu ou fin de chaîne :

L'insertion a lieu en fin de chaîne si la dernière sélection a mis en évidence une absence de lien vers une entité "frère" suivante (pointeur à zéro) et en milieu de chaîne dans le cas contraire.

Le processus d'insertion peut être décrit comme suit :

3.7.3.2.1- Acquisition de buffers concernant l'enregistrement logique d'après les codes fichiers

3.7.3.2.2- Transfert des données dans les buffers acquis

3.7.3.2.3- Génération d'une clé et transfert dans le buffer principal

3.7.3.2.4- Chaînage aval :

Dans le cas d'insertion en milieu de chaîne, il faut établir un lien entre l'enregistrement en cours de création et le suivant. Pour cela il faut :

. Obtenir le numéro d'information "pointeur de chainage" dans la table des enregistrements.

. Transférer, dans cette information l'occurence dans le fichier du numéro d'enregistrement suivant, c'est-à-dire la valeur du pointeur avec laquelle on a effectué la dernière sélection.

3.7.3.2.5- Traitement particulier à l'extension

3.7.3.2.6- Traitement des enregistrements index

3.7.3.2.7- Ecriture des enregistrements de données

3.7.3.2.8- Chaînage amont :

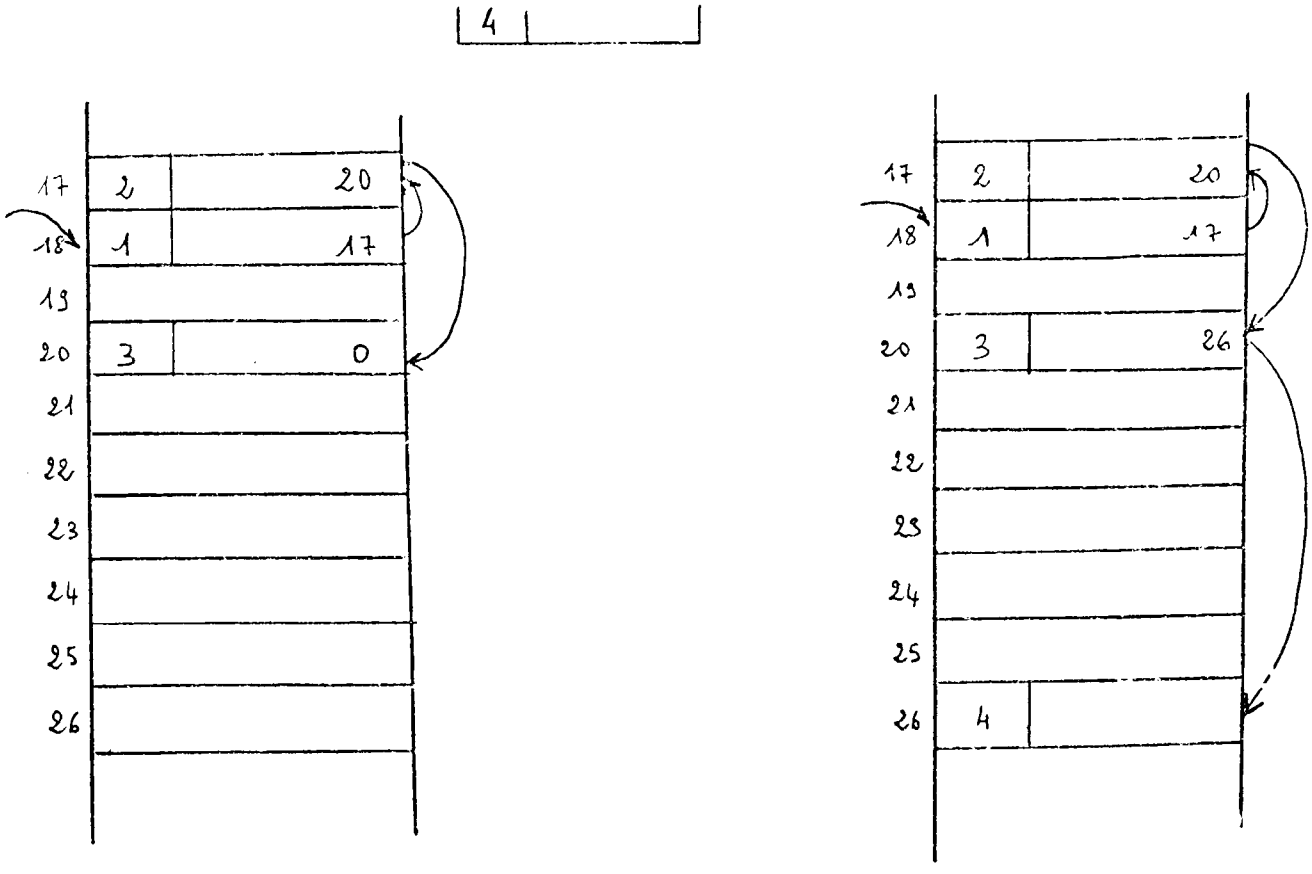
Il faut relire l'enregistrement "en amont" de l'enregistrement en cours de création au sens du chaînage.

Dans le cas d'insertion en fin de chaîne, il s'agit de l'enregistrement obtenu au cours de la dernière sélection, dans le cas contraire, il s'agit de l'enregistrement obtenu au cours de l'avant-dernière sélection.

Cet enregistrement est lu avec intention de mise à jour puis on y transfère dans l'information "pointeur de chaînage" l'occurence dans le fichier de l'enregistrement principal en cours de création.

Enfin on réécrit l'enregistrement amont.

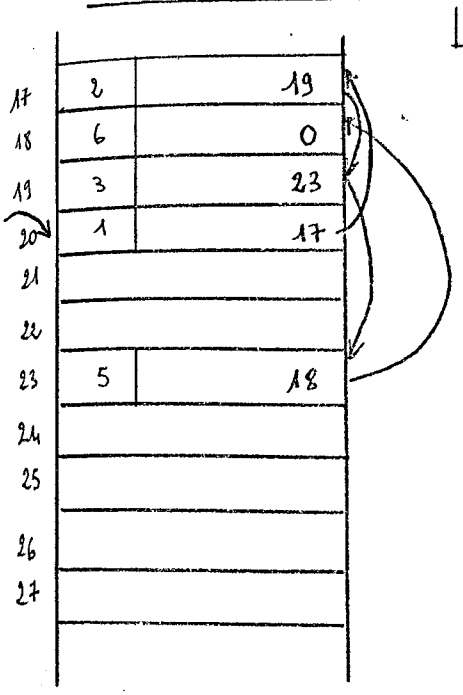
Exemple d'insertion en fin de chaîne :



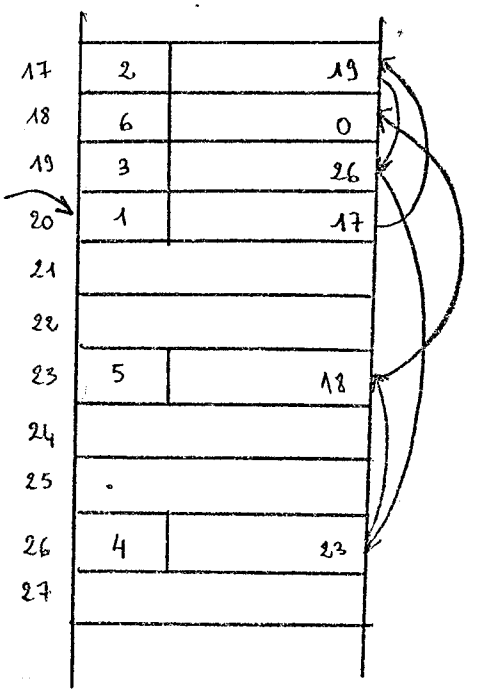
AVANT CHAINAGE

APRES CHAINAGE

Exemple d'insertion en milieu de chaîne :

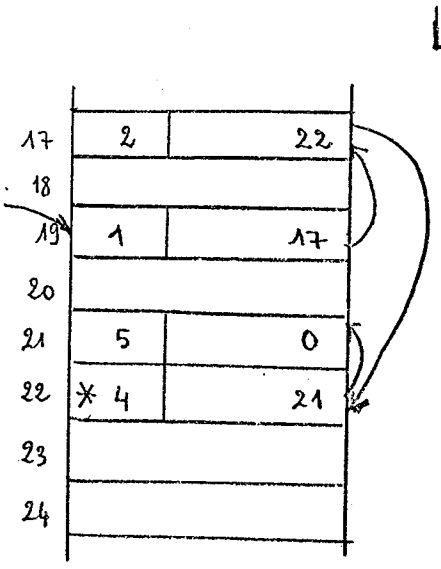


AVANT CHAINAGE

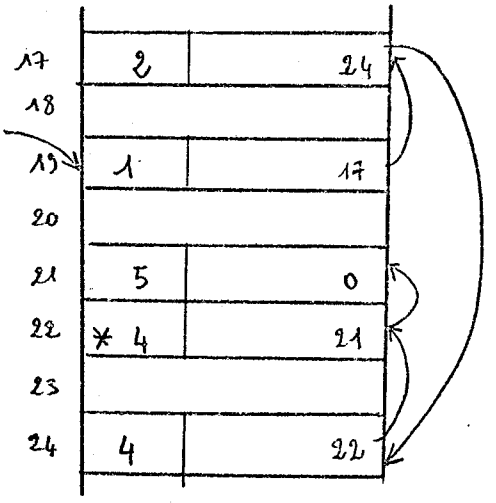


APRES CHAINAGE

Exemple d'insertion en milieu de chaîne d'un enregistrement déjà supprimé :



AVANT CHAINAGE



APRES CHAINAGE

REMARQUE : Le chiffre de gauche représente la clé de l'enregistrement, le chiffre de droite le pointeur et '*' le caractère d'invalidation.

3.8- SELECTION D'INFORMATIONS

C'est la requête la plus sollicitée du noyau puisqu'elle est utilisée à la fois par les utilisateurs lorsqu'ils effectuent une interrogation sur des informations de la base et par les autres fonctions pour localiser un enregistrement logique.

Contrairement aux autres opérations, elle n'opère pas spécifiquement sur un fichier logique mais sur un ensemble de fichiers reliés par des liens de type père-fils. Le résultat de l'opération de sélection ne s'exprime pas en terme de fichiers logiques mais en terme d'informations réparties sur un ou plusieurs fichiers logiques.

Notre étude sur le module de sélection se découpe en quatre parties qui sont :

- Définition de la fonction de sélection, étude des concepts qui y sont rattachés et de la structure sur laquelle elle opère.
- Détermination du point d'entrée dans la base.
- Parcours de la structure arborescente.
- Restitution des résultats.

3.8.1- Etude générale du module de sélection :

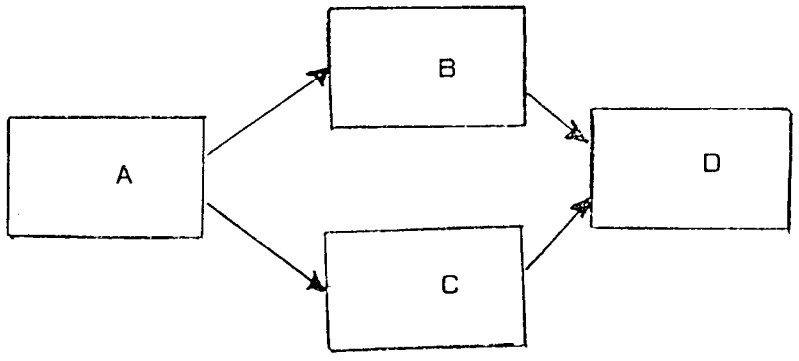
Le shéma de fonctionnement de ce module peut être résumé de façon très succinte comme suit :

- Etude du point d'entrée dans la base, celui-ci pouvant être un fichier index ou non selon le type de sélection.
- Examen des fichiers logiques et progression sur la structure arborescente selon les contraintes fixées et le chemin choisi par l'utilisateur.
- Fourniture des résultats ou d'un code indiquant une impossibilité technique à réaliser la requête formulée.

Etudions d'abord de façon plus précise la notion de chemin et d'action.

3.8.1.1- Chemin - Action :

Nous avons défini un chemin comme une suite de fichiers logiques que l'on doit parcourir. Celui-ci est fourni par l'utilisateur d'après la méthode Daniel MARTIN car il permet de lever le doute sur le parcours de structure telles que :



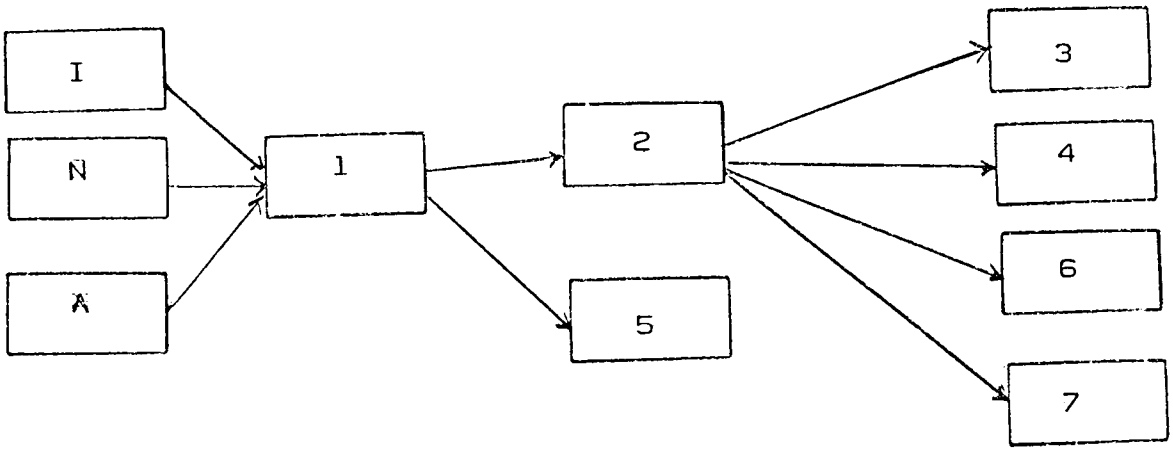
et où les informations à sélecter et les contraintes ne portent que sur A et D.

Une telle structure n'est pas autorisée dans le système présenté, c'est pourquoi il serait envisageable de déterminer automatiquement le parcours d'après les informations à sélecter et les informations sur lesquelles portent des contraintes.

Pour Daniel MARTIN, les fichiers index secondaires, le fichier index principal et le fichier de données de niveau 1 ne forment qu'un seul et même fichier logique. De ce fait une seule lettre du parcours suffit, nous l'avons vu, à indiquer l'accès au fichier de niveau 1 par index secondaire. Mais aussi un index secondaire ne peut être consulté indépendamment du fichier de données auquel il est associé. Et ceci présente dans certains cas un inconvénient qui se traduit par une baisse de performances.

Exemples de parcours :

Soit la base de données :



1, I 2 7, N, I 5, 2 3, 5 sont des parcours autorisés

N 1, 2 3 4, 5 7 sont des parcours interdits

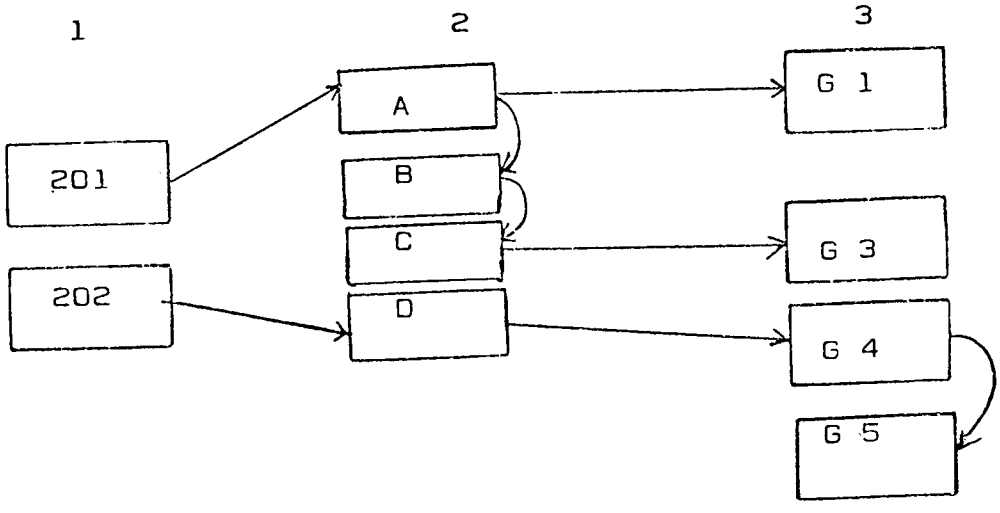
REMARQUE :

2 3 et 5 ne sont des parcours autorisés qu'avec la sous-fonction sel-pointeur.

Notion d'action :

Il ne nous semble pas que ce concept puisse présenter un intérêt important. Son utilité ne paraît évidente que dans le cas où le premier enregistrement d'une chaîne possède un rôle privilégié.

Exemple :



Les actions X/X associées au parcours 123 permettent de sélecter :

201	A	G 1
202	D	G 4
et 202	D	G 5

L'opération de sélection peut-être effectuée selon différents modes, chacun d'eux correspondant à un type d'accès à la base.

3.8.1.2- Types de sélection :

3.8.1.2.1- Sélection avec contraintes (appelé également sel premier sur contraintes) :

C'est le type de sélection le plus courant et le plus complet.

Dans ce cas, les informations d'une entité sont recherchées en fonction de contraintes portant sur des clés d'accès et/ou des informations quelquonques de la base, puisque toute information répertoriée dans la base peut être l'objet d'une contrainte.

3.8.1.2.2- Sélection avec pointeur :

Ce type de sélection est en général demandé par un autre module du noyau. En effet, cette opération fait référence à un pointeur, objet inconnu d'un utilisateur qui n'a, en principe, qu'une vue logique de la base. Cependant, une telle possibilité reste intéressante dans des cas particuliers où le problème de performances est aigu.

Cette façon particulière d'accéder à la base de données n'est pas incompatible avec la fourniture de contraintes dans la mesure où celles-ci ne sont pas incohérentes avec le parcours choisi.

3.8.1.2.3- Sélection du suivant :

Lorsque l'opération "sel premier sur contraintes" ou sel pointeur a fourni les informations concernant une ou plusieurs entités données, on peut être tenté d'obtenir les informations de l'entité suivante qui satisfait aux contraintes et ceci jusqu'à ce qu'il n'y en ait plus. C'est ce que permet de faire le "sel-suivant" à condition toutefois de conserver le même chemin d'accès d'un appel à l'autre. Ceci implique que le système mémorise un positionnement sur la base.

Le fait que cette sous-fonction fournisse la prochaine entité satisfaisant aux contraintes autorise à modifier celles-ci d'un appel à l'autre, à condition de ne pas altérer les combinaisons de contraintes sur les clés des index. Une telle possibilité est bien entendu à utiliser avec précautions car elle permet de référencer des entités qui peuvent n'avoir aucun point commun.

Ces différentes opérations de sélection traitent un certain nombre de structures de fichiers que nous allons étudier.

3.8.1.3- Types de fichiers :

3.8.1.3.1- Le fichier index principal :

A chaque enregistrement du fichier principal de la base (appelé également fichier maître ou racine) correspond un poste dans l'index principal. Ce poste contient la clé de l'enregistrement et la valeur du lien vers le fichier maître. Index principal et fichier principal peuvent être confondus grâce à une méthode d'accès indexée. C'est le cas dans notre application et de ce fait le fichier racine exerce deux fonctions simultanément : celle d'un fichier index et celle d'un fichier de données.

La clé de l'enregistrement fournie par l'utilisateur est toujours une clé complète. Une partie seulement des informations de ce fichier peut être modifiée.

3.8.1.3.2- Le fichier index secondaire :

Il n'y a pas bijection, comme dans le cas précédent, entre l'ensemble des enregistrements de l'index secondaire et l'ensemble des enregistrements du fichier principal.
En effet, plusieurs enregistrements index peuvent être associés au même enregistrement du fichier principal et un enregistrement du fichier principal peut ne pas avoir de correspondance dans l'index.

La structure d'un index secondaire est déterminée d'après l'utilisation qui en est faite ultérieurement, de façon à à répondre le mieux possible aux besoins des applications.

Les contraintes ne peuvent porter que sur la clé d'interrogation.

La recherche d'informations peut se faire sur clé tronquée ce qui permet de connaître tous les enregistrements du fichier principal correspondant à un même préfixe.

Exemples :

- Rechercher toutes les entreprises dont la raison sociale est ou commence par NET PRESSING

- Rechercher toutes les entreprises dont la rue commence par BER à SAINT-ETIENNE ou toutes les entreprises de toutes les rues des villes dont le nom commence par SAINT-MARTIN.

3.8.1.3.3- Les fichiers de données :

Les fichiers autres que ceux ayant une fonction d'index sont dans notre application sous organisation relative, c'est-à-dire que l'accès à un enregistrement s'effectue en connaissant son occurence dans le fichier.

Comme nous l'avons déjà vu, toutes les informations sont accessibles à l'utilisateur bien que certaines soient sans rapport avec le schéma de la base qu'il connaît. (Ceci est également vrai pour les fichiers index). La clé concaténée de l'enregistrement et sa clé spécifique jouent un rôle privilégié (optimisation des recherches).

Une partie seulement des informations présentes dans ces fichiers, peut être modifiée.

3.8.1.4- Les contraintes de sélection :

L'utilisateur indique le sous-ensemble des informations qu'il désire sélecter au moyen de contraintes qui permettent un affinement successif de l'espace des informations de la base.

Ces contraintes sont formulées au moyen d'un numéro de zone qui est la codification de chaque information dans le dictionnaire.

La formulation d'une contrainte s'effectue en constituant une relation arithmétique ainsi structurée :

< numéro d'information > < opérateur > < code réponse > < valeur de référence >

Les différents opérateurs autorisés sont, rappelons le :

< ≤ = ≥ > ≠ que l'on codifie respectivement par un chiffre de 1 à 6.

Les contraintes sont reliées entre elles par les opérateurs logiques ET et OU. L'opérateur OU est toujours considéré comme le plus prioritaire dans l'analyse des critères de choix.

Il est matérialisé par un numéro d'information à zéro indiquant que la valeur de référence suivante doit être considérée comme une autre possibilité autorisée.
Cet opérateur subit cependant une restriction d'utilisation : il est interdit sur les numéros d'information qui sont des clés d'accès aux index.

D'autre part, pour les clés des enregistrements en organisation relative, les valeurs de référence doivent être présentées dans la même séquence que le fichier concerné.

Le code réponse est une information rendue par le système et qui donne un compte-rendu sur la contrainte.

Trois valeurs possibles peuvent être fournies :

Y : la contrainte est satisfaite

N : la contrainte n'est pas satisfaite

blanc : la contrainte n'a pas été examinée

Ce code présente une utilité certaine car il permet en particulier de connaître lorsque le système répond "non trouvé" de déterminer à quel niveau hiérarchique il y a eu absence d'entité.

REMARQUE : Dans le cas où on veut rendre le OU moins prioritaire que le ET il est prévu de chaîner des zones de communication, le chaînage étant équivalent à un OU implicite (non réalisé pour l'instant).

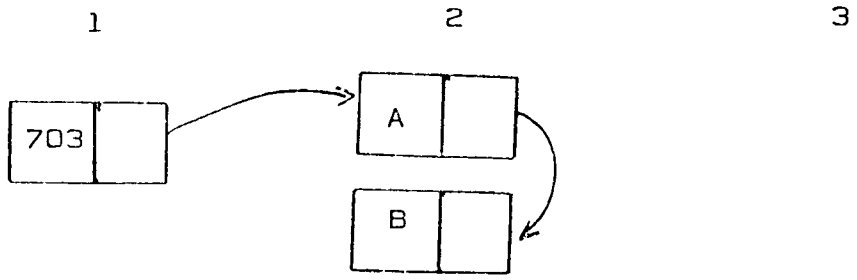
3.8.1.3.4- Le code absence d'informations ou absence d'enregistrement :

Ce code, nous l'avons vu, signifie que les contraintes portant sur un enregistrement logique absent sont considérées comme satisfaites.

De ce fait, les informations appartenant aux fichiers logiques en amont sur le parcours peuvent être délivrées et pour l'enregistrement logique absent, il est fourni les valeurs par défaut des informations demandées.

Cette possibilité doit être considérée comme une optimisation car elle peut permettre d'éviter une requête supplémentaire. C'est le cas lorsqu'on veut savoir si une entité existe et où on veut connaître des informations appartenant aux entités "père".

Exemples :



parcours 123

code absence enregistrement = 1

contraintes sur niveau 1 et 2 satisfaites

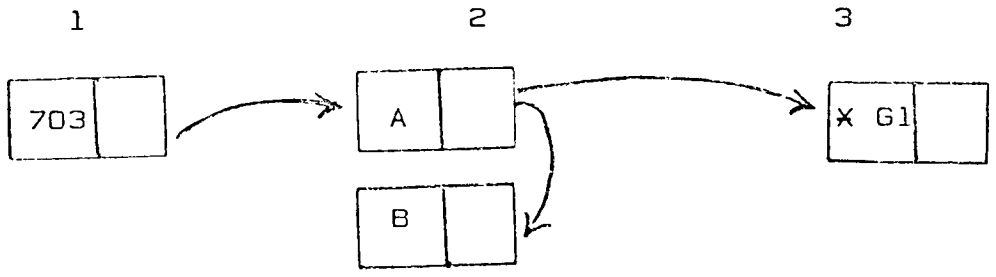
réponse : non trouvé 10

---

si code absence enregistrement = 2

réponse = trouvé 00

On obtient les informations de la base sur les niveaux 1 et 2 et les informations par défaut sur le niveau 3

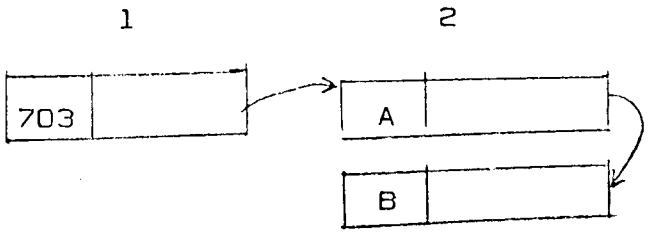


parcours 123

code absence enregistrement = 2

contraintes sur niveau 1 et 2 satisfaites

réponse = non trouvé 10 car en cas de suppression les pointeurs sont laissés dans leur état initial



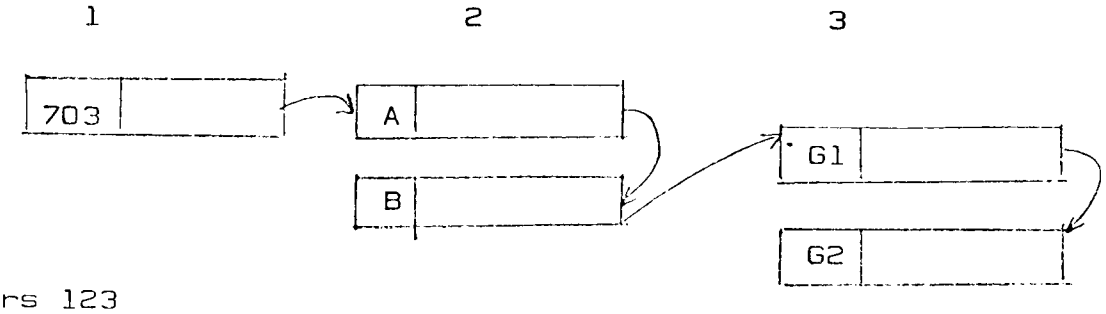
parcours 123

code absence enregistrement = 2

contrainte sur niveau 1 satisfaite

contrainte sur niveau 2 non satisfaite (ex : n° = C)

réponse : non trouvé 10



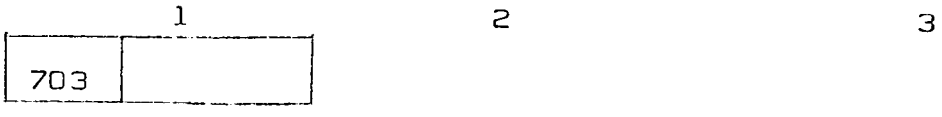
parcours 123

code absence enregistrement = 2

contrainte sur niveau 1 et 2 satisfaite

contrainte sur niveau 3 non satisfaite (ex : n° G4)

réponse : non trouvé 10



parcours 123

code absence enregistrement = 2

contrainte sur niveau 1 satisfaite

réponse : trouvé 00

On obtient les informations de la base sur le niveau 1 et les informations par défaut sur les niveaux 2 et 3.

3.8.2- Détermination du point d'entrée dans la base :

Nous allons pour cette partie de l'étude différencier trois cas correspondant aux trois sous-fonctions de sélection.

3.8.2.1- Cas de la sélection avec contraintes (ou sélection du premier avec contraintes)

Le parcours est indiqué de façon complète, de l'index jusqu'au dernier niveau hiérarchique atteint. L'utilisateur a en principe fourni parmi les contraintes un certain nombre de critères qui sont des informations priviligiées. Le traitement se déroule en plusieurs phases qui sont :

contrôle de validité des contraintes, détermination de la fourchette index et cycle de recherche dans la base.

3.8.2.1.1- Contrôle de validité des contraintes :

Les contraintes portant sur les clés de lecture sont repérées, marquées et analysées. Il est vérifié que :

- Il n'y a pour ce type de contrainte aucun opérateur "différent"($\neq$)
- Il n'y a pas d'opérateur logique OU
- Si l'utilisateur précise une fourchette de lecture (au moyen de deux contraintes) la valeur minimum doit être inférieure à la valeur maximum.
- Il n'y a qu'un opérateur de chaque classe ($<$, $=$, $>$)
- L'ensemble de ces contraintes donne une recherche de type :
  - . égal
  - . inférieur
  - . supérieur
  - . inférieur - supérieur
  - . non précisé, c'est-à-dire sélection de tous les enregistrements de la racine.

L'ensemble de ces contrôles est destiné à vérifier qu'il est possible de déterminer une fourchette unique de recherche sur le fichier index (principal ou secondaire). En conséquence les recherches telles que :

. clé $<$ 10 et clé $>$ 30005
. clé $<$ 20 ou clé $>$ 280
. clé $\neq$ 11002

sont interdites.

3.8.2.1.2- Détermination de la fourchette index :

Cela consiste à déterminer d'après les contraintes sur les clés index des bornes MINI et MAXI de lecture dans ce type de fichier.

Celles-ci sont déterminées, selon le code opérateur, de la façon suivante : (où les bornes MINI et MAXI ont été respectivement initialisées aux valeurs binaires minimales et maximales).

<	Si MINI = (valeur minimale) ou (valeur maximale) MINI = (valeur minimale) Si MAXI = (valeur minimale) ou (valeur maximale) MAXI = (valeur minimale) MAXI = SPMOVE (valeur référence)
≤	Si MINI = (valeur minimale) ou (valeur maximale) MINI = (valeur minimale) Si MAXI = (valeur minimale) ou (valeur maximale) MAXI = (valeur maximale) MAXI = SPMOVE (valeur référence)
=	MINI = (valeur référence) MAXI = (valeur référence) Si entrée par index secondaire MINI = CMPLMT (MINI, (valeur minimale)) MAXI = CMPLMT (MAXI, (valeur maximale))
≥	Si MINI = (valeur minimale) ou (valeur maximale) MINI = (valeur minimale) MINI = SPMOVE (valeur référence) Si MAXI = (valeur minimale) ou (valeur maximale) MAXI = (valeur maximale)
>	Si MINI = (valeur minimale) ou (valeur maximale) MINI = (valeur maximale) MINI = SPMOVE (valeur référence) Si MAXI = (valeur minimale) ou (valeur maximale) MAXI = (valeur maximale)

- SPMOVE est une fonction qui permet de mouvementer une information sur sa longueur définie dans le dictionnaire.

- CMPLMT est une fonction qui complète dans le contenu d'une variable tous les blancs par une valeur précisée [ici(valeur minimale)ou(valeur maximale)].

<u>Exemple illustrant le traitement de l'opérateur = sur une clé d'accès à un index secondaire</u> :

Nous avons précédemment cité le cas de la recherche des entités dont on connaissait partiellement l'adresse : par exemple :

- le nom de la rue commence par BER dans la localité SAINT-ETIENNE

ou - le nom de la localité commence par SAINT-MARTIN

Dans le premier exemple et

dans le cas d'un index adresse constitué des informations : localité et rue et de contraintes du type:

n° info localité-rue = 'SAINT-ETIENNEb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BER'

La constitution des bornes de recherche évolue comme suit :

1.

MINI : SAINT-ETIENNEb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BERb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸

MAXI : SAINT-ETIENNEb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BERb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸

2. MINI :

SAINT-ETIENNEb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BER 00 00 00 00 00 00 00 00 00 00 00 00

MAXI :

SAINT-ETIENNEb̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BER ff ff ff ff ff ff ff ff ff ff ff ff

<u>REMARQUE</u> : Cet exemple est théorique. Lors de la création et de la consultation des index un certain nombre de traitements sont effectués pour filtrer les informations (suppression des tirets, apostrophes, etc...)

Dans ce cas, l'enregistrement index délivre par exemple les informations :

SAINT-ETIENNE	Cours BERRIAT	entreprise 1724
	rue Paul BERT	entreprise 3
	Avenue du BERRY	entreprise 8886

Dans le cas d'un index principal la génération de bornes de lecture autorise un accès immédiat au fichier du niveau 1.

Dans le cas d'un index secondaire on effectue les traitements suivants :

. positionnement sur le premier enregistrement ayant un préfixe supérieur ou égal à MINI. (Cette opération est autorisée par certaines méthodes d'accès)

. lecture d'un enregistrement (le premier puis le suivant)

. vérification de la clé lue par rapport à la valeur de MAXI.

. refus de prise en compte de l'enregistrement s'il est supprimé (présence d'un caractère d'invalidation en première position)

. génération de la clé de l'enregistrement principal associé.

- la lettre du parcours permet d'obtenir le numéro d'enregistrement de l'index traité
- la table des enregistrements donne un numéro de poste dans la table des enregistrements index
- Dans cette table le poste indique la liste des numéros d'informations de l'enregistrement index qui concaténées donne la valeur de la clé d'accès au niveau 1.

Ce principe autorise à morceller les informations constituant la clé du fichier pointé. Il n'est pas une vue de l'esprit mais permet de répondre à des besoins pratiques réels.

On est alors ramené au cas de l'accès par l'index principal. A ce niveau peut commencer le parcours de la structure arborescente qui est décrit plus loin.

3.8.2.2- Cas de la sélection par pointeur :

C'est rappelons-le une facilité pouvant être utilisée à des fins d'optimisation.

Dans ce cas, le parcours commence au fichier logique pointé mais peut également faire référence aux fichiers logiques dépendant de celui-ci

En effet, une sélection par pointeur permet plus que le simple parcours du fichier logique pointé.

Elle représente une façon particulière de se positionner sur la base mais ensuite l'algorithme de parcours est identique dans tous les cas.

Pour ce type de sélection, il ne peut y avoir de contraintes portant sur les clés d'accès aux index et plus généralement sur les niveaux hiérarchiques en amont du niveau pointé. Il en est de même pour les informations sélectées.

3.8.2.3- Cas de la sélection du suivant :

La procédure utilisée est semblable à celle de la sélection du premier avec contraintes.

Un contrôle de validité des contraintes sur clé index est effectué. La modification de la valeur des fourchettes est autorisée mais pas les combinaisons d'opérateurs. Les contraintes sur clé index peuvent être omises.

Dans ce cas, ce sont les fourchettes déterminées au cours du précédent appel qui seront prises en compte, sinon le contrôle est suivi par la détermination de la fourchette index.

Ce type de sélection est basé sur la mémorisation de l'état d'exploration de la base à chaque instant. Ceci est réalisé en conservant pour chaque niveau hiérarchique la valeur du pointeur vers le prochain enregistrement logique à lire. Cette valeur est zéro en fin de chaîne ou s'il s'agit d'un fichier non chaîné.

En télétraitement une restriction est faite à l'utilisation du SEL-SUIVANT.

Lorsqu'une requête fait référence à la précédente (cas du sel-suivant) elle doit être effectuée dans la même tâche que la requête référencée. Ceci est lié à un problème de stockage des informations.

En effet, il ne paraît pas pensable, en télétraitement, de sauvegarder à priori des états d'exploration pour le cas où ils seraient utilisés ultérieurement.

Par contre CICS offre, pour chaque tâche, une zone de travail appelée TWA (Task Work Area) de longueur paramétrable (dans notre cas 200 octets) de durée de vie égale à celle de la tâche et qui permet de conserver à moindre coût les valeurs des pointeurs correspondant aux six niveaux hiérarchiques théoriques. Cette zone permet également de sauvegarder les fourchettes index et les clés en cours sur le fichier index et le fichier principal.

Nous allons maintenant nous intéresser à l'étude du parcours dans la base qui est fonction du chemin et des résultats successifs des contraintes.

3.8.3- Parcours de la structure arborescente :

Le cycle de recherche dans la base peut se résumer ainsi :

- lecture d'un enregistrement
- examen des contraintes
- progression dans la base

Nous allons détailler chacun de ces points.

3.8.3.1- Lecture d'un enregistrement :

Dans le cas où l'enregistrement à lire n'a pas été trouvé (ce qui ne peut pas se produire pour des fichiers en organisation directe relative) la recherche s'arrête ou se poursuit en prenant en compte les valeurs par défaut pour ce niveau hiérarchique compte tenu du code absence d'enregistrement.

La recherche s'arrête également avec un code réponse approprié dans les cas de fin de fichier et d'erreur d'entrée sortie.

Si une racine est supprimée on mémorise ce fait de façon à répondre non-trouvé ou à fournir les valeurs réelles avec un code réponse particulier selon le résultat des contraintes.

Tout enregistrement supprimé autre que la racine n'est pas pris en compte, sauf si la requête provient du module d'addition car dans ce cas, les informations d'un tel enregistrement sont nécessaires afin d'établir des chaînages corrects.

3.8.3.2- Examen des contraintes :

Au cours des contrôles effectués par le module d'aiguillage une table a été renseignée ; elle indique pour chaque contrainte le numéro de buffer concerné. Ainsi, il est aisé d'effectuer, par fichier physique lu, l'examen des contraintes s'y rapportant.

Ce traitement n'est donc pas effectué par fichier logique. Cela signifie que si des contraintes ne sont pas satisfaites sur l'enregistrement principal il n'y a pas lecture de l'extension, d'où économie de temps.

Etude de l'examen d'une contrainte :

Les différents résultats qui peuvent être issus de cet examen sont :

- contrainte satisfaite
- contrainte non satisfaite
  - . L'information n'est pas une clé et la valeur de référence ne correspond pas (au sens de l'opérateur) à la valeur lue dans la base.
  - . L'information est une clé et d'après la valeur lue et la séquence du fichier, on ne pourra plus trouver d'enregistrement appartenant au chaînage qui satisfasse la relation arithmétique fournie. Nous avons appelé ce cas EOR (END OF RESEARCH).
    Il est détecté grâce à un traitement particulier effectué seulement si le numéro d'information est une clé complète ou spécifique de l'enregistrement.
- erreur de formulation d'une contrainte
  - . Le numéro de zone n'existe pas
  - . L'opérateur est invalide
  - . La valeur de référence est contradictoire avec le type de la zone à comparer (présent dans l'élixir).

Le troisième point que nous étudions est la progression dans la base.

3.8.3.3- Progression dans la base :

Nous avons vu que les liens entre entités sont réalisés à l'aide de pointeurs. Ceux-ci permettent un accès rapide de la racine vers les extrémités de l'arbre. Mais le chaînage inverse n'existe pas, ce qui pose le problème du retour au noeud précédent.

La solution nous l'avons déjà vu consiste à se passer du chaînage inverse en conservant à chaque noeud l'adresse du sous-arbre que l'on n'explore pas.
La structure arborescente est donc parcourue de façon préfixée.

REMARQUE : Ce chaînage n'est pas utile sur les fichiers index puisque la méthode d'accès permet de les obtenir de façon classée.

Cette simulation du chaînage inverse permet un cheminement, qui dans le cas normal, c'est-à-dire résultat de contraintes positif ou absence d'enregistrement accepté, est décrit comme suit :

. Si la donnée traitée est une extension, on se positionne sur le fichier principal et on exploite le résultat.

. Si la donnée traitée est un fichier principal :

.. On sauvegarde l'adresse du demi sous-arbre non traité (valeur du pointeur frère)

.. S'il existe une extension, on se positionne sur celle-ci et on effectue le test des contraintes.

Les causes d'arrêt du cheminement peuvent être explicitées comme suit :

3.8.3.3.1- Fin du parcours :

On se situe au plus haut niveau hiérarchique et il y a eu un résultat positif aux contraintes à tous les niveaux. La recherche se termine de façon correcte ; le résultat peut-être délivré.

3.8.3.3.2- Contraintes non satisfaites :

A un niveau hiérarchique donné, une contrainte (ou un ensemble de contraintes reliées par l'opérateur logique OU) n'est pas satisfaite.

. Il existe une autre occurence :

Celle-ci est examinée.

. Il n'existe pas d'autre occurence :

Il peut s'agir d'une inexistence physique ou logique.

.. Inexistence physique :

L'occurence est la dernière du chaînage.

.. Inexistence logique :

- L'action correspondant au niveau hiérarchique traité est / qui n'autorise le traitement que du premier enregistrement d'un chaînage.
- Vu la séquence de tri du fichier logique une valeur ne peut-être trouvée (cas EOR).

Dans ces deux cas, il faut remonter au dernier noeud rencontré de façon à voir si le sous-arbre dont l'adresse a été conservée n'est pas vide, c'est-à-dire s'il existe d'autres occurences.

3.8.3.3.3- Fin de fourchette :

La remontée vers les noeuds précédents s'est effectuée jusqu'à la racine. La lecture d'un enregistrement index laisse apparaître que la valeur maximale de la fourchette a été atteinte. Le résultat de la recherche est négatif.

REMARQUE : Dans le cas de critères d'égalité sur la clé de l'index principal, on fait l'économie de la lecture de l'enregistrement index principal suivant.

3.8.3.3.4- Un fils n'existe pas :

C'est le cas où le pointeur vers le niveau aval est à zéro. La décision est fonction du choix de l'utilisateur (code absence d'enregistrement).

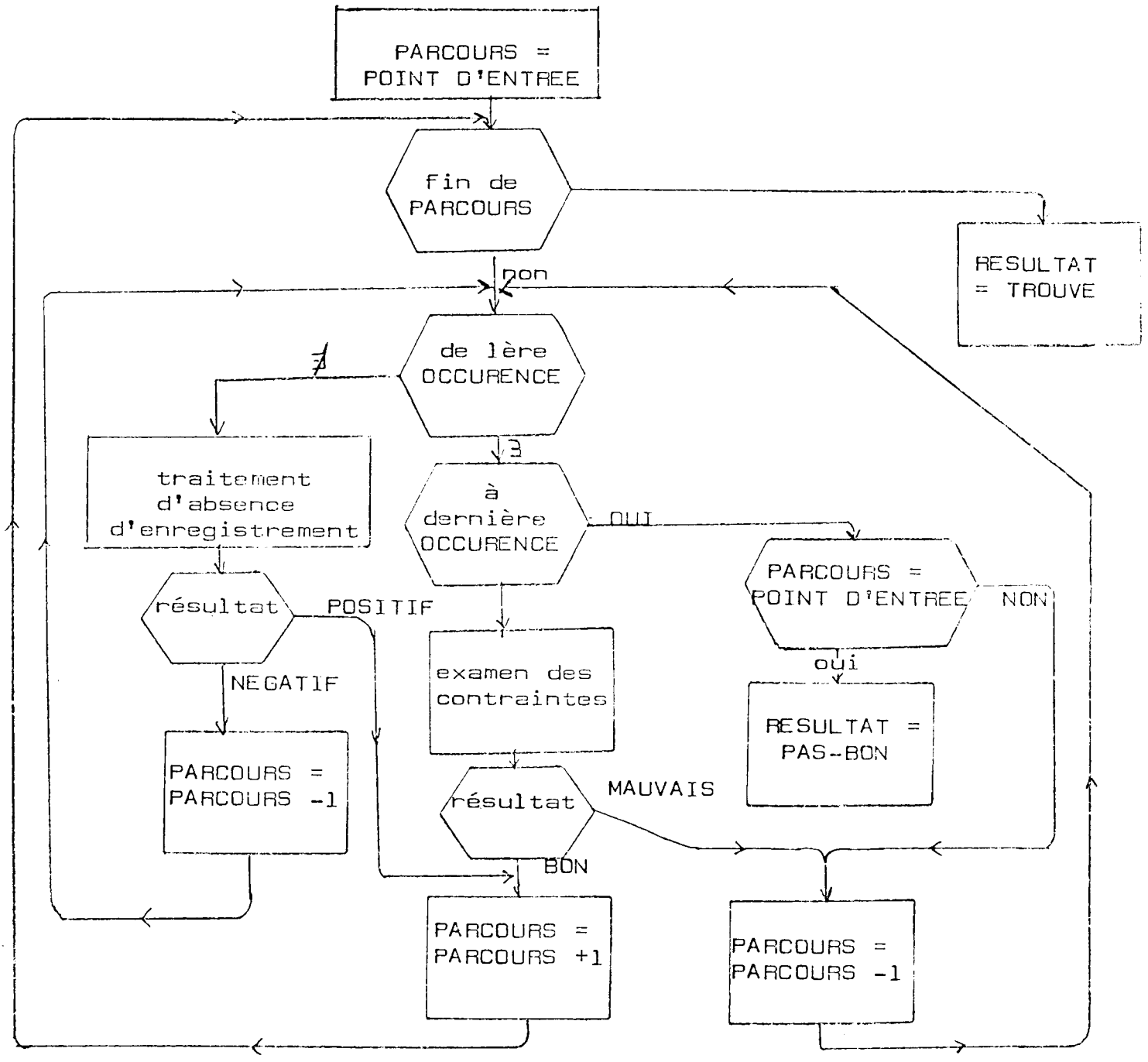
. L'utilisateur accepte l'inexistence :

On est ramené au cas qui traite de l'absence d'une autre occurence.

. L'utilisateur désire des valeurs par défaut :

Les valeurs par défaut de l'enregistrement sont prises en compte et l'exploration se poursuit.

L'algorithme de cheminement sur la structure peut-être schématisé comme suit :



La quatrième partie de notre étude sur la sélection concerne la restitution des résultats.

3.8.4- Restitution des résultats

3.8.4.1- Cas d'erreur :

Pour chaque situation anormale un code approprié est placé dans la zone de communication, ce qui marque la fin de traitement de la requête.

3.8.4.2- Cas de fin normale :

Un ensemble d'enregistrements logiques correspondant au parcours indiqué par l'utilisateur et satisfaisant aux contraintes a été lu. Il suffit d'effectuer le transfert des informations contenues dans les buffers vers la zone de communication. Ce processus n'est pas décrit car il est identique à celui présenté dans le module de modification à deux différences près, qui sont :

- Le transfert d'informations a lieu en sens inverse.
- Le transfert s'opère à partir de plusieurs buffers (ce qui est absolument transparent pour le processus).

Conjointement au transfert d'informations un code réponse de fin normale est placé dans la zone de communication. Le traitement de la requête est terminé.

3.9- LES ENTREES SORTIES

Chaque type de fichier nécessite un jeu particulier d'Entrées/ Sorties. Celui-ci est lié à la méthode d'organisation utilisée.

Nous énumérons ci-après par type de fichier les jeux d'instructions nécessaires :

3.9.1- Fichier index secondaire :

. Positionnement sur clé tronquée

. Lecture de l'enregistrement suivant (accès séquentiel)

. Fin de positionnement

. Lecture pour mise à jour (accès direct)

. Réécriture

. Ecriture

. Suppression

3.9.2- Fichier index principal :

. Positionnement sur clé tronquée

. Lecture de l'enregistrement suivant (accès séquentiel)

. Fin de positionnement

et tous les ordres utilisés pour un fichier de données.

3.9.3- Fichier de données :

. Lecture directe

. Lecture directe pour mise à jour

. Réécriture

. Ecriture

4- APPLICATION AU TRAITEMENT PAR LOTS

Il est peu réaliste d'envisager pour une base de données des traitements exclusivement conversationnels. En effet de façon périodique, il est nécessaire d'effectuer une réorganisation de la base. Ce traitement implique la lecture de tous les enregistrements et ne peut donc s'envisager que sous un aspect "batch".

Mis à part ce traitement très spécifique, d'autres explorations sont en général nécessaires, soit pour la recherche d'enregistrements répondant favorablement à certains critères de sélection afin de réaliser des traitements périodiques, soit pour la mise à jour systématique d'informations.
Par exemple, les institutions de retraite éditent chaque trimestre à partir des informations de leurs entités "entreprise" des bordereaux d'appel de cotisations qui s'adressent aux adhérents présentant les conditions requises.

La façon normale de réaliser de tels traitements consiste à utiliser une version "batch" du noyau. Dans le cas où on ne dispose pas d'une telle ressource, une autre solution consiste à utiliser le 'noyau TP' sous un système de télétraitement dans des conditions particulières.

4.1- TRAITEMENT A L'AIDE DU NOYAU TP

Le système CICS permet un paramètrage important des tâches qu'il est apte à réaliser et donc de son niveau de complexité. De façon à perdre le moins possible de ressources consommées par le système, on peut rechercher un paramétrage qui engendre un système peu gourmand.

Ce dernier n'a à gérer de façon simultanée qu'une seule tâche.

L'initialisation de cette tâche s'effectue normalement par la saisie d'un code transaction, mais la durée de vie de la tâche augmente dans des proportions très importantes en fonction des volumes traités. Ce système de télétraitement spécialisé n'est en service qu'en dehors des cessions normales. On peut estimer que le temps consommé par le système TP, et donc perdu, n'excède pas 40 % du temps total du traitement.

4.2- TRAITEMENT A L'AIDE D'UN NOYAU BATCH :

Ce module doit être conçu de la même façon que le noyau TP existant mais avec un certain nombre de particularités. Nous énumérons ci-après les points de ressemblance et de divergences.

4.2.1- Point commun avec le système TP :

La conception des traitements est identique. Les algorithmes de chaque opération du noyau sont à conserver et de ce fait l'utilisation de toutes les tables précédemment étudiées.

4.2.2- Points de divergence avec le système TP :

- La hiérarchisation des ressources du système TP, due à la multiplicité de déroulement des tâches concurrentes n'existe pas en batch. Elle n'a d'ailleurs pas lieu d'être puisqu'il n'y a qu'un seul utilisateur dans la partition. De ce fait il existe une structure unique de ressource mémoire permettant la manipulation et la sauvegarde d'informations.

- Le chargement des tables n'a pas été effectué au préalable (réalisé au cours de l'initialisation du système en TP). Il convient donc de le faire au début du traitement de la première requête.

- Tous les traitements du noyau TP utilisent une structure particulière de sauvegarde des adresses des tables et des buffers. Il est nécessaire de s'y ramener si l'on veut utiliser le code existant, en particulier, en simulant le contenu de la zone commune appelée CWA.

- En télétraitement les appels étant ponctuels et les chaînages contenant peu d'éléments on ne tient pas compte du regroupement des enregistrements logiques en blocs physiques. Ainsi deux lectures successives d'enregistrements présents dans un même bloc entraîne théoriquement l'exécution de deux lectures physiques. Ceci est atténué par le fait que l'organisation VSAM gère des CI c'est-à-dire des groupements d'enregistrements qui sont conservés en mémoire centrale pendant un certain délai dépendant de l'activité du système. De ce fait deux entrées/sorties successives sur un même CI n'entraînent pas toujours deux lectures physiques.

  Pour les traitements par lots où l'exploration de tous les enregistrements logiques d'un même niveau est en général nécessaire, il convient d'optimiser les entrées/sorties et n'effectuant pas une nouvelle lecture si le bloc a déjà été consulté ou en attendant le début de traitement du bloc suivant pour réécrire le bloc précédent.

. Il n'est pas, à priori, évident que toutes les fonctions qui sont offertes en télétraitement soient utiles en batch. Cela dépend des besoins des utilisateurs. Il ne peut pas y avoir, nous semble-t-il une règle générale, mais on peut faire le choix de fournir les mêmes possibilités en batch et en télétraitement.

On peut également décider de ne rendre disponible que les fonctions de modification et de sélection si ce sont les seules susceptibles d'être utilisées.

. Dans notre cas les sécurités au niveau du SGBD sont pour la plupart offertes par CICS. En batch il n'existe pas de procédure particulière équivalente. Une partie des problèmes de sécurité peut être résolue par les procédures classiques de points de reprise mais celles-ci ne permettent pas de résoudre le cas d'un incident survenu au cours d'un chaînage par exemple.

Une première solution consiste, à ne pas gérer de pointeurs sauf en lecture, c'est-à-dire à interdire la fonction d'addition et la modification d'extension (qui peut éventuellement entraîner l'ajout d'un enregistrement).

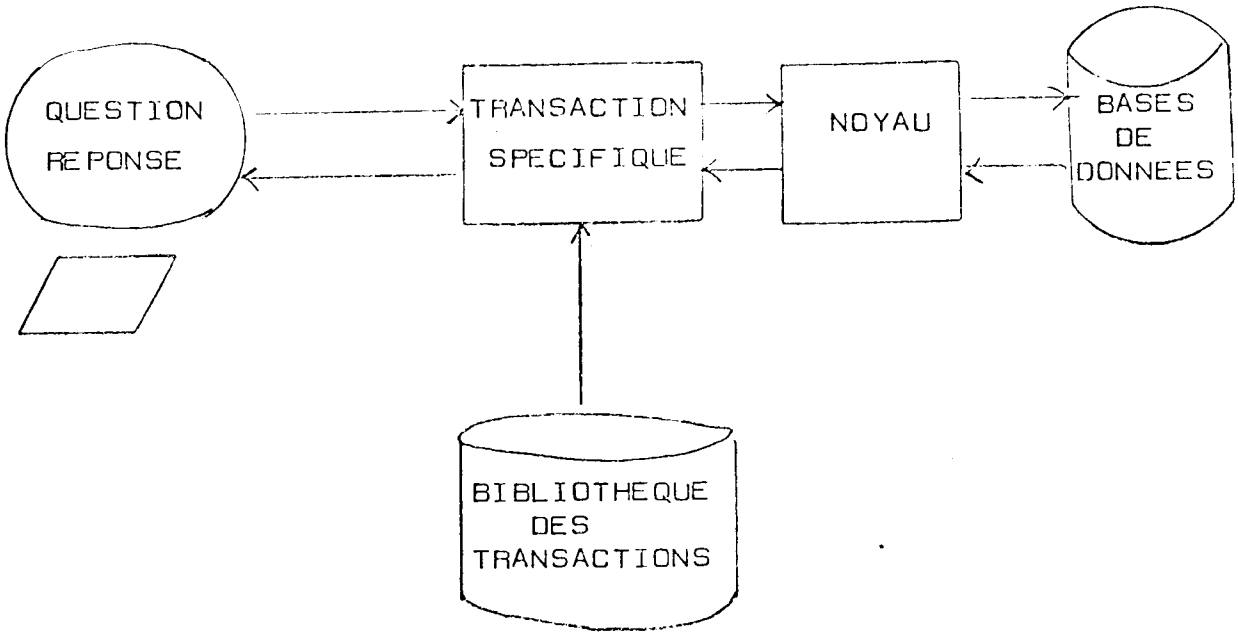
Une deuxième solution consiste à développer son propre système de sécurité, dans le même esprit que celui offert par CICS.

5- ASPECT CONVERSATIONNEL DU SGBD

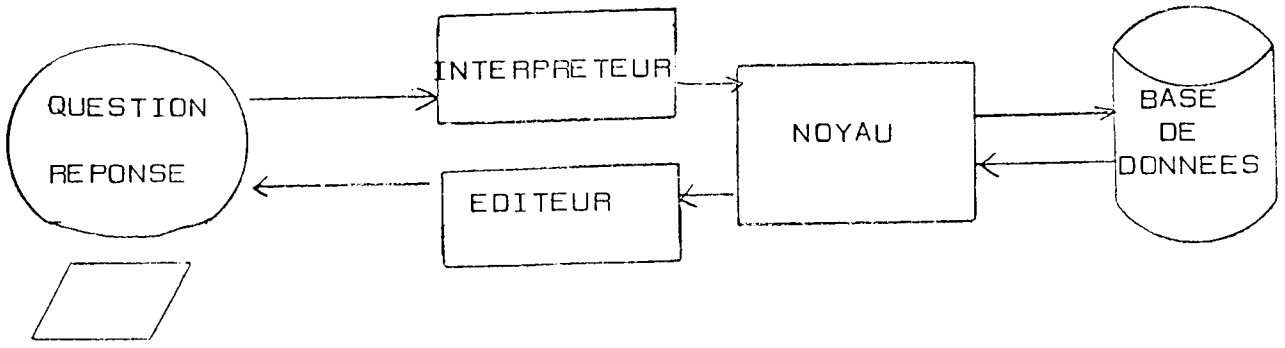
La plupart des systèmes de bases de données permettent une interrogation transactionnelle des informations. C'est le cas pour les systèmes basés sur le modèle CODASYL. Cela signifie que la base de données peut être à chaque instant interrogée en télétraitement à l'aide de transactions existantes, mais ceci implique ainsi que toute question nouvelle doit au préalable être codée à l'intérieur d'un programme avant de pouvoir être utilisée couramment.

L'aspect conversationnel offert par certains systèmes permet d'éliminer cette contrainte, qui peut être gênante dans des cas où le résultat de l'interrogation présente un caractère d'urgence. L'utilisateur dispose alors d'un langage permettant de poser dans l'instant qui suit n'importe quelle question nouvelle ou non à la base et d'obtenir une réponse immédiate.

Ce service est offert par le système de bases de données mis en place au CRIOP.



SCHEMA SIMPLIFIE DE L'ASPECT TRANSACTIONNEL



SCHEMA SIMPLIFIE DE L'ASPECT CONVERSATIONNEL

Nous décrivons ci-après le langage de communication avec la base disponible au CRIOP. Celui-ci est sensiblement le même que celui que préconise Daniel MARTIN. Il s'agit exclusivement d'un langage d'interrogation. En effet si les questions des utilisateurs qui portent sur des combinaisons d'informations sont loin d'être toutes prévisibles, les mises à jour sont par contre bien cernées et peuvent faire l'objet de transactions spécifiques. C'est du reste le seul moyen d'assurer la cohérence des informations.

Le faible pourcentage des cas où l'utilisation d'un traducteur pour un usage autre que l'interrogation, sans risque d'introduire des informations incohérentes dans la base, ne nous a pas paru justifier la mise en place d'un tel dispositif.

La syntaxe de l'interpréteur telle qu'elle est préconisée par Daniel MARTIN est la suivante :

blancs	numéro de base (2 chiffres de 01 à 99)	NIV 1 caractère niveau d'entrée qui désigne le fichier d'entrée	PTR pointeur
PAR	parcours ensemble de lettres-chiffres	NPA non-parcours ensemble de lettres-chiffres	
MON TØT	liste des numéros de zones à montrer ou totaliser (4 chiffres séparés par 1 blanc)	SEL TOUT SEL numéro de zone opérateur valeur de référence (*)	

(*) la deuxième possibilité peut être répétée plusieurs fois

REMARQUES :

1 signifie que la séquence à l'intérieur des crochets est facultative

2 signifie qu'un blanc est obligatoire mais il peut y en avoir plusieurs

3 les opérateurs possibles sont : $< \leq = \geq > \neq$

que nous avons respectivement codés : $<$ I = S $>$ D

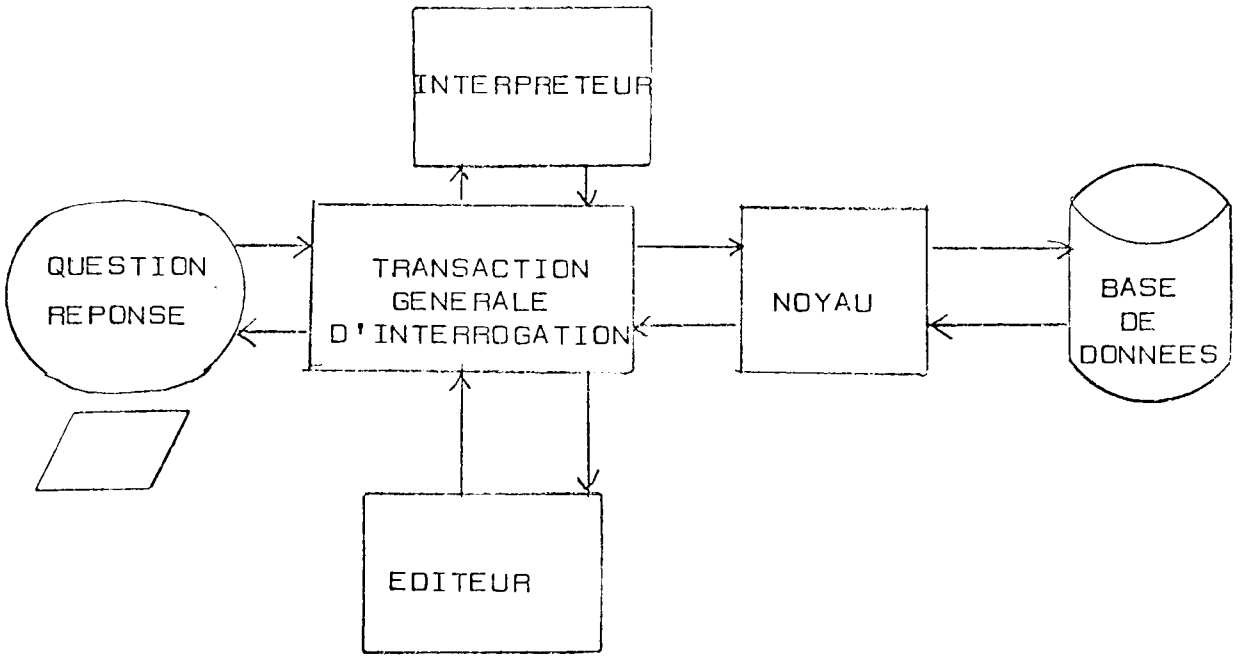
4 le niveau d'entrée est facultatif ; sa valeur par défaut est 1

(Nous avons abandonné cette notion qui nous a semblé faire double emploi avec le chemin).

Mis à part le concept de niveau d'entrée, toutes les autres informations sont effectivement à fournir par l'utilisateur si l'on veut pouvoir constituer de façon complète une zone de communication. Notons que nous avons appelé 'action' ce qui dans la syntaxe présentée est noté 'non-parcours'.

Une telle syntaxe nous a cependant paru compliquée. De plus, le format libre est source d'erreurs plus fréquentes d'où, l'idée de figer sur un écran l'emplacement des informations à fournir.

L'interpréteur n'étant pas prévu pour gérer des informations sur un écran, nous avons fait évoluer le schéma du parcours d'interrogation comme suit :



Ce système permet de ne renseigner sur l'écran que les informations variables de la question et minimise considérablement le risque d'erreurs de syntaxe.

Ces informations sont automatiquement transformées en format interne par l'interpréteur.

De façon à conserver les conventions d'utilisation de l'interpréteur, c'est la transaction générale d'interrogation qui, à partir du format fixe saïsi sur l'écran, génére une question en format "libre" compatible.

L'interpréteur pour résoudre le message qui lui est fourni, utilise des prémices de façon identique à l'analyseur syntaxique d'un compilateur classique.

L'affichage des résultats ne peut avoir lieu directement à partir de la zone de communication sans un traitement particulier. En effet beaucoup d'informations de la base ne sont pas en format étendu et de ce fait non affichable. De plus la formulation d'une requête entraîne souvent plusieurs réponses et il peut paraître intéressant d'en afficher plus d'une sur un même écran.

Ces deux raisons amènent à utiliser un module de "mise en forme" appelé éditeur dont les fonctions sont les suivantes :

- Rappel des numéros de zone demandés avec leur libellé correspondant (pris dans le fichier des libellés).
- Fourniture d'un titre (d'écran ou de liste) composé des libellés des informations demandées intercalés de séparateurs, ces libellés étant fournis sur la même longueur que l'information. Ce titre peut éventuellement tenir sur plusieurs lignes.
- Restitution des informations en format étendu, structurées de la même façon que le titre.

La transaction ou le programme qui réceptionne les informations en provenance de l'éditeur peut les afficher ou éditer sans autre traitement.

L'éditeur travaille à partir de la zone de communication issue du noyau.

Les informations qui lui sont nécessaires pour fournir les résultats sont recherchées sur le dictionnaire de données et le fichier des libellés.

Nous avons décrit l'aspect conversationnel qui permet à l'aide d'un langage d'interrogation, d'obtenir une réponse immédiate à toute question.

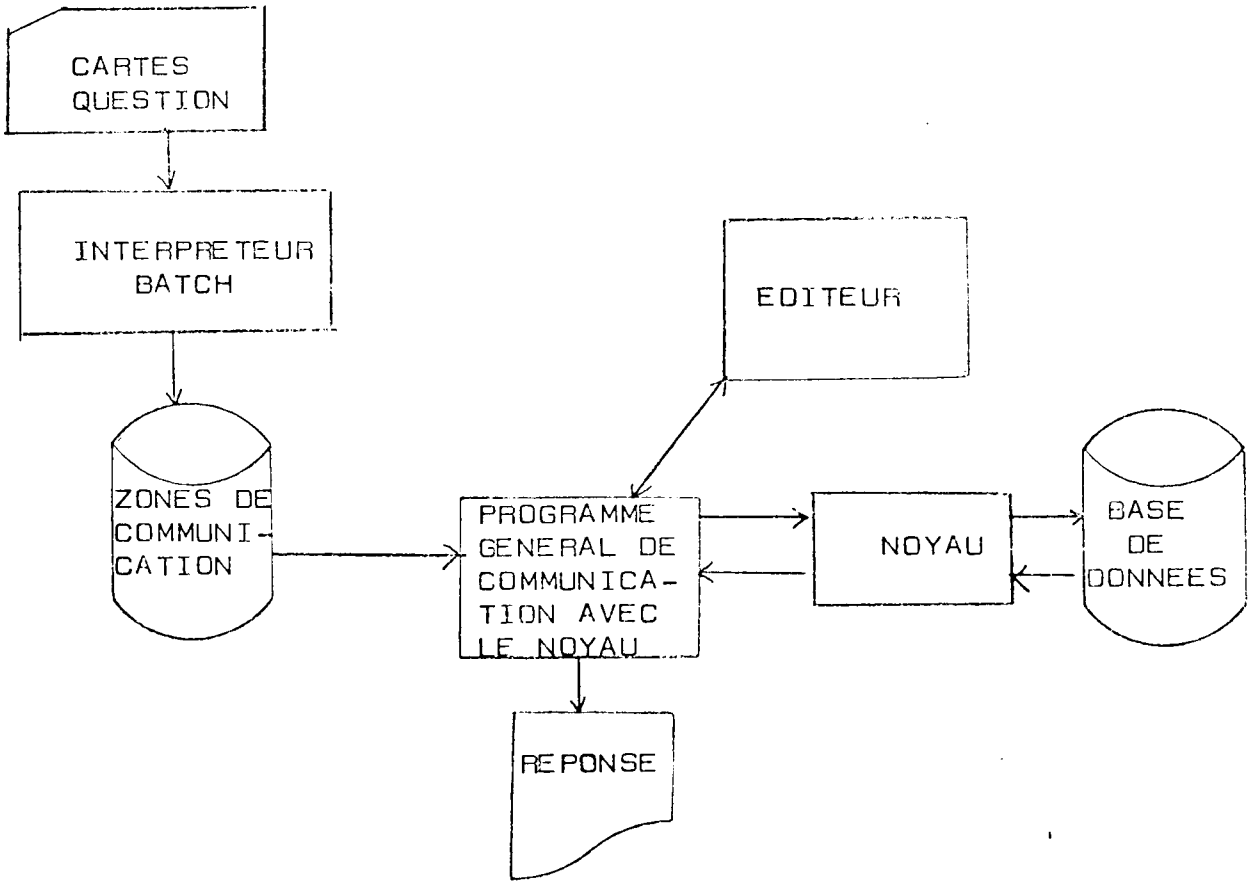
Le point important est en fait la possibilité offerte de ne pas avoir à codifier au préalable la question dans un programme.

Ceci présente également un intérêt non négligeable dans le cas de traitements par lots. En effet les questions portant sur des gros volumes ne peuvent pas toujours être envisagées dans un environnement de télétraitement.

De plus les évolutions systématiques et l'éventualité d'un besoin de correction d'erreurs amènent à offrir la possibilité de coder directement des requêtes de modification.

Un outil semblable à l'interpréteur mais plus complet car traitant toutes les fonctions du noyau permet donc la codification de requêtes destinées à être exécutées dans un environnement batch.

Le schéma du processus d'accès à la base est le suivant :



6- TRAITEMENT DE REORGANISATION DES BASES

Il est nécessaire de réorganiser les bases de données de façon périodique pour des raisons d'efficacité :

- Suppression physique des enregistrements logiquement annulés, ce qui entraîne un gain de place sur les emplacements disques.

- Restructuration des fichiers chaînés pour que tous les enregistrements logiques dépendant d'un même père soient physiquement consécutifs, ceci afin de réaliser un gain de temps lors des traitements par lots.

- Restructuration par la méthode d'accès des fichiers en organisation indexée.

Une réorganisation de tout ou partie d'une base peut également être nécessaire de manière ponctuelle dans les cas suivants :

- Changement de domaines
- Modification de la structure des clés
- Modification des liens entre fichiers logiques
- Modification de la séquence de tri d'un fichier.

Enfin on peut éventuellement utiliser le traitement de réorganisation pour résoudre des problèmes spécifiques liés à certaines évolutions d'un ou plusieurs fichiers de la base, par exemple :

- Ajout de la valeur par défaut d'une nouvelle information
- Modification des caractéristiques d'une ou plusieurs informations (nature, longueur, etc...)
- Evolution de la longueur des enregistrements.

Ce traitement consiste à recréer les liens entre les différents fichiers de données réorganisés en se basant sur la valeur de la clé concaténée (ou clé complète) présente, nous l'avons vu, au début de chaque enregistrement de données.

Le fait que l'on puisse ne réorganiser qu'une partie de la base ou même qu'un fichier logique offre une souplesse non négligeable car permet de minimiser des traitements relativement longs et coûteux.

Le traitement de réorganisation est toujours cohérent par rapport à ce que réalise le noyau car il utilise les deux principales tables mises en place qui sont : tables des enregistrements et élixir.

Le principe de réorganisation retenu pour les fichiers en organisation directe relative est le suivant :

Le fichier à réorganiser est trié puis il y a mise à jour éventuelle des pointeurs de chaînage et mise à jour des pointeurs du fichier amont vers le fichier réorganisé. Les pointeurs vers les fichiers aval ne sont pas altérés.

Selon les fichiers de la base que l'on désire réorganiser un certain nombre de traitements sont nécessaires afin de conserver l'intégrité des liens de celle-ci.

On distingue pour un fichier le traitement de réorganisation de celui de mise à jour pointeur qui consiste uniquement à renseigner la nouvelle valeur des pointeurs vers le fichier aval réorganisé.

Dans le cas de la base entreprise, par exemple, l'ensemble des traitements à réaliser en fonction des fichiers à réorganiser est déterminé à l'aide du tableau suivant :

Fichiers à réorganiser								
- établissement	X							
- Catégorie-tranches		X						
- Groupes-risques			X					
- Historiques ID-AD				X				
- Historique Situation					X			
- Extension entreprise						X		
- Compléments établissement							X	
Traitements à effectuer								
- Réorganisation établissement	X							
- " catégories-tranches		X						
- " groupes-risques			X					
- " historique ID-AD				X				
- " historique situation					X			
- " extension entreprise						X		
- "complément établissement							X	
- mise à jour pointeurs établissement		X	X	X	X		X	
- mise à jour pointeurs entreprise	X			X		X		

La somme des traitements à réaliser est obtenue par projection horizontale du demi-tableau inférieur.

Une mise à jour des pointeurs et une réorganisation sur un même fichier se traduisent uniquement par une réorganisation de ce fichier.

Les traitements sont réalisés dans l'ordre où ils apparaissent dans le tableau c'est-à-dire de haut en bas.

Dans le cas d'une réorganisation de toute la base, l'ensemble des traitements est à réaliser compte-tenu des remarques ci-dessus.

Pour l'instant les traitements de réorganisation effectués sur des fichiers en organisation indexée se limitent à l'exploitation de deux utilitaires de la méthode d'accès qui permettent de :

- Sauvegarder la version de fichier sur une mémoire secondaire (ici bande magnétique)

- Restaurer une nouvelle version logiquement identique à la précédente à partir de la sauvegarde.

Si un traitement spécifique est nécessaire sur un fichier indexé de la base on utilise alors un module d'accès généralisé à un fichier qui, pour chaque enregistrement, donne le contrôle à un programme contenant les spécificités en question et qui peut demander au programme principal les trois possibilités suivantes :

- Prise en compte de la version initiale de l'enregistrement
- Prise en compte de la version modifiée de l'enregistrement
- Suppression de l'enregistrement

Nous étudions ci-après de façon plus détaillée le traitement de réorganisation d'un fichier en direct-relatif. Celui-ci peut se décomposer en plusieurs étapes qui sont :

1- <u>Tri du fichier à réorganiser</u>

Vu les volumes traités, l'espace disque nécessaire au tri est très important (il y a pour certains fichiers plusieurs centaines de milliers d'enregistrements).

On utilise donc un moyen détourné pour obtenir les enregistrements dans un ordre trié qui consiste à extraire du fichier de la base la clé complète de chaque enregistrement que l'on accouple à l'emplacement de ce dernier (fourni par la méthode d'accès) puis à trier ces couples en ordre ascendant sur la clé complète.

2- Réorganisation proprement dite :

. Il y a alignement du fichier des couples (ou extrait) trié avec les N fichiers clés-amont des fils déjà réorganisés.

Pour chaque enregistrement extrait lu on accède directement à l'enregistrement concerné sur la base.

Les N fichiers clés-amont permettent de mettre à jour les N pointeurs sur les premiers enregistrements des N fichiers fils.

Parallèlement on établit le chaînage entre enregistrements du fichier réorganisé par attribution d'un numéro séquentiel croissant. Le dernier enregistrement de la chaîne reçoit pour valeur de pointeur de chaînage la valeur nulle. (ceci correspond à une rupture sur la clé de l'enregistrement père).

. Il y a également création d'un fichier clé-amont pour le père.

Une fois le fichier réorganisé, le fichier père doit subir une mise à jour des pointeurs ou une réorganisation. Pour ce faire il est nécessaire de conserver les informations permettant d'établir les nouveaux liens c'est-à-dire pour chaque clé du père différente, l'occurence du premier enregistrement du chaînage sur le fichier réorganisé.

. Puis chaque enregistrement nouvelle situation est écrit sur une bande magnétique.

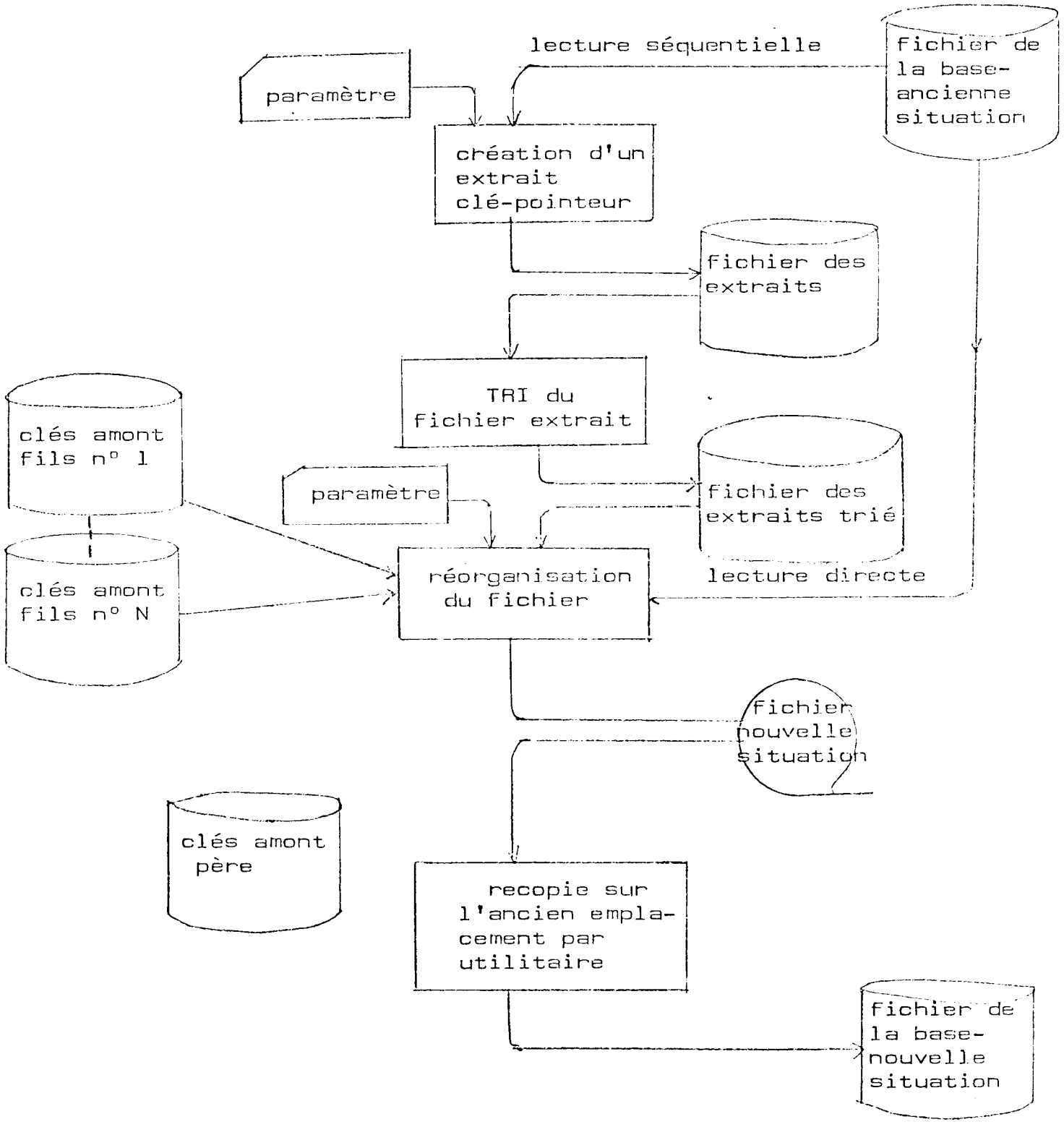
3- Recopie sur le fichier de la base

REMARQUES :

. Le fichier à traiter est connu grâce à son numéro d'enregistrement fourni en paramètre.

. Les informations clé et les caractéristiques de l'enregistrement sont déduites de la table des enregistrements et de l'élixir.

Le schéma correspondant à ce traitement est décrit ci-après.


lecture séquentielle
fichier de la base-ancienne situation
paramètre
création d'un extrait clé-pointeur
fichier des extraits
TRI du fichier extrait
clés amont fils n° 1
paramètre
fichier des extraits trié
clés amont fils n° N
réorganisation du fichier
lecture directe
fichier nouvelle situation
clés amont père
recopie sur l'ancien emplacement par utilitaire
fichier de la base-nouvelle situation

7- ASPECT SECURITE

Divers incidents ou accidents peuvent se produire pendant l'exploitation d'un système de bases de données : arrêt anormal d'une transaction, panne d'électricité, support magnétique détruit, etc... Des dispositifs doivent être mis en oeuvre pour assurer la cohérence des informations d'une part et un niveau d'information le plus proche possible de celui existant au moment de l'incident d'autre part.

Nous allons étudier les différentes classes d'incidents et les remèdes qui permettent de résoudre chacun d'eux.

7.1- MARCHE ARRIERE :

Il arrive qu'une transaction s'arrête au cours de son exécution pour les raisons suivantes :

- Evenement anormal pour le programme (problème de logiciel)
- Arrêt de la transaction par le système de télétraitement
- Arrêt de la transaction par l'utilisateur.

Dans tous les cas certains fichiers ont pu être mis à jour et d'autres non. Par exemple si une requête d'ajout d'enregistrement sur un fichier chaîné avait été demandée au noyau, il se peut que l'arrêt intervienne au cours de l'établissement du chaînage ce qui rend incohérentes les informations en place au moment de l'arrêt.

Un dispositif de CICS appelé DTB (Dynamic Transaction Backout) permet d'effacer tous les ajouts, modifications ou suppressions effectués sur les fichiers depuis le début de la transaction lorsque celle-ci s'est mal terminée. Ceci permet donc de remettre les fichiers dans l'état où ils se trouvaient avant que la transaction ne commence à travailler.

Il est possible de paramètrer dans les tables de CICS les transactions et les fichiers sur lesquels ce processus doit s'appliquer. Ainsi tous les fichiers des bases sont dits "protégés" par le DTB et celui-ci s'applique à toutes les transactions qui effectuent des requêtes autres que des sélections sur ces bases.

Très schématiquement ce processus fonctionne de la manière suivante :

Pour les transactions concernées par le DTB

- Pour les fichiers dits "protégés", s'il s'agit d'une lecture avec intention de mise à jour ou d'une suppression, on conserve l'image de l'enregistrement initial, et s'il s'agit d'une création on mémorise le fait que l'enregistrement est nouveau.

Si la transaction se termine normalement, tout ce qui a été mémorisé est détruit.

Sinon on applique dans le sens chronologique toutes les "images avant" des enregistrements modifiés ou supprimés et on efface les enregistrements créés (Ceci est possible car la mémorisation des informations concernant les entrées-sorties au cours de la transaction est effectuée séquentiellement selon l'ordre chronologique).

7.2- ARRET INCONTROLE DU RESEAU TP :

Ceci peut arriver dans les cas suivants :

- Evenement anormal pour CICS
- Panne de courant
- Arrêt du réseau TP d'une autre façon que par la procédure habituelle.

Dans ces cas là une ou plusieurs transactions étaient éventuellement en cours d'exécution et il se pose pour chacune d'elles les mêmes problèmes de cohérence d'informations.

Un autre dispositif de CICS appelé Emergency Restart permet de retrouver les transactions qui n'étaient pas terminées ou transactions "en vol" et d'appliquer sur les fichiers "protégés" les images avant des enregistrements traités par la transaction. Ce processus est de conception très proche de celle du DTB. Son fonctionnement est le suivant :

. Pour chaque tâche un message de début est inscrit sur un fichier journal.

. Parallèlement à leur mémorisation pour le DTB, les informations concernant les entrées/sorties de la transaction sont également écrites sur le fichier journal dans leur ordre chronologique.

. Pour toutes les transactions qui se terminent normalement, un message de "fin normale" est également inscrit sur le journal.

Dans le cas d'arrêt imprévu de CICS le processus Emergency Restart permet de se positionner à la fin logique du fichier journal, d'effectuer une lecture de ce fichier en marche arrière et de prendre en compte toutes les transactions qui n'ont pas une marque de "fin normale".

Toutes les images avant concernant ces transactions sont mémorisées séquentiellement sur un fichier appelé "Restart Data Set". Elles s'y trouvent donc dans leur ordre chronologique et il ne reste plus qu'à les appliquer sur les fichiers concernés.

REMARQUE : Lorsqu'une transaction va se terminer normalement tous les buffers modifiés au cours de celle-ci sont obligatoirement transférés sur les fichiers (pour des raisons de performances, le transfert n'est pas systématique). A ce moment là seulement la transaction est considérée comme terminée.

7.3- PROBLEMES LIES A L'ORGANISATION DES FICHIERS DE LA BASE :

Au cours de la PLT initiale les numéros des derniers enregistrements écrits sur chaque fichier en organisation directe relative (présents sur un fichier appelé "fichier maître") sont transférés en CWA. Au cours de la cession de télétraitement seules les valeurs présentes en CWA progressent au fur et à mesure de l'ajout d'enregistrements (pour des raisons de performances).

Dans le cas d'un arrêt normal du réseau une phase exécutée au cours de la PLT finale permet de transférer de la mémoire centrale vers le fichier maître la valeur des occurences des derniers enregistrements écrits. Dans le cas d'un arrêt incontrôlé du réseau ce traitement ne peut être effectué.

Voyons comment est résolu ce problème qui ne se pose pas pour les fichiers d'organisation indexée.

Une procédure appelée VERIFY est exécutée pour tous les fichiers et de façon systématique (fin normale ou anormale de la cession). Elle est, pour des raisons de sécurité incluse dans la procédure d'EMERGENCY RESTART.

Son exécution après un arrêt normal de CICS n'est pas superflue car elle permet de détecter une éventuelle altération des valeurs présentes en mémoire (incohérence entre l'état de remplissage du fichier et les valeurs écrites sur le fichier maître).

La procédure VERIFY (fournie avec la méthode d'accès VSAM) est basée sur le principe suivant :

Pour chaque fichier VSAM la méthode d'accès indique, dans un catalogue qui lui est propre, une information appelée High Used RBA (gérée automatiquement) et qui correspond au nombre total d'octets contenus par rapport au début du fichier.

La procédure VERIFY compare donc la valeur du High Used RBA présente dans le catalogue à celle calculée d'après l'occurence du dernier enregistrement créé.

S'il n'y a pas égalité, le High Used RBA du catalogue est mis à jour avec la valeur calculée puis il y a répercussion dans le fichier maître.

Ainsi les pointeurs sur de nouveaux enregistrements créés dans la base restent cohérents puisqu'ils correspondent à des enregistrements "récupérés" par la procédure VERIFY.

La mise à jour des nouvelles valeurs dans le fichier maître permet d'effectuer les prochaines cessions de télétraitement sans perte d'informations et en conservant la cohérence de celles-ci.

7.4- MARCHE AVANT :

Suite à un accident tel que : effacement d'un domaine, chute d'un disque, il est nécessaire de reconstituer les bases de données dans l'état le plus proche possible de l'état où elles étaient au moment de l'accident.

Ce problème est résolu par l'application sur une version sauvegardée de toutes les images "après" des enregistrements qui ont été créés, modifiés ou supprimés.

Nous avons mis en place la procédure suivante :

- Il y a sauvegarde hebdomadaire de tous les fichiers constituant les bases de données.

- Au lieu de sauvegarder les images après, on mémorise les informations permettant de les reconstituer.

Ceci est effectué par le noyau qui écrit sur son propre fichier journal ou fichier "Recovery", lui même sauvegardé tous les soirs, toutes les requêtes autres que les sélections avec d'autres informations telles que le jour et l'heure pour conserver la chronologie. Cette solution a été retenue car elle présente pour nous l'avantage de résoudre en même temps des problèmes liés à certaines de nos applications.

En cas de panne les opérations suivantes sont effectuées :

- Restauration sur la base de la dernière version sauvegardée.
- Application chronologique (par le noyau TP) de toutes les requêtes appliquées à la base suite à la sauvegarde de celle-ci et jusqu'à la dernière sauvegarde du fichier Recovery.

Si l'incident survient en dehors d'une cession CICS on est en mesure de restaurer une situation exacte.

Si l'incident se produit pendant une cession CICS et si le fichier Recovery n'est pas altéré, il y a également application des requêtes de la cession, ce qui permet dans ce cas d'avoir également une situation exacte.

Dans le plus mauvais cas (fichier Recovery inutilisable) la version disponible est celle du début de la cession ce qui oblige à recommencer le travail effectué au cours de celle-ci.

Ce principe nécessite une excellente gestion des bandes de sauvegarde ainsi qu'un stockage étudié.

7.5- LIENS INCORRECTS :

Si, à la suite de résultats anormaux, on découvre que des liens sont incorrects (problème de logiciel) il y a deux possibilités d'intervention.

- Dans le cas où on est en mesure de déterminer de façon précise depuis quand ce problème existe on peut utiliser le processus décrit dans le paragraphe précédent, à condition bien sûr de posséder toutes les sauvegardes nécessaires.

- Mais s'il s'agit uniquement d'un problème de liens il est plus simple d'utiliser les modules de réorganisation de la base dont nous avons déjà parlé. En effet ceux-ci sont conçus pour recréer tous les liens d'une base.

Quelle que soit la méthode choisie, une correction du logiciel s'impose (avant l'exploitation du processus de sécurité si celui-ci fait appel au logiciel incriminé).

8- EVOLUTIVITE DU SYSTEME

La mise en place de diverses tables est destinée à assurer une évolution plus aisée des bases existantes et une création simple de nouvelles bases de même structure.

Le système que nous étudions offre, sous certaines conditions que nous précisons, les possibilités suivantes :

1- Modification des caractéristiques d'une zone (nature, longueur, emplacement, etc...)

   Ceci est réalisé par une modification du dictionnaire.

2- Création de nouvelles zones dans un enregistrement logique.

   Il y a très vraisemblablement modification de la valeur d'initialisation dans la table des enregistrements vides et si cela se traduit par une variation de la taille de l'enregistrement physique correspondant il y a modification de la table des enregistrements (longueur de l'enregistrement et déplacement dans la table des enregistrements vides pour les numéros d'enregistrement suivant).

3- Création de nouveaux fichiers logiques (que cela se traduise ou non par la matérialisation d'un nouveau niveau hiérarchique dans une limite de six niveaux).

   Ceci a un impact dans la table des parcours, le dictionnaire, la table des enregistrements, la table des enregistrements vides et éventuellement dans la table des index et la table des constantes.

4- S'il n'était pas prévu d'extension pour un fichier possibilité d'en ajouter une.

   Ceci a un impact dans le dictionnaire, la table des enregistrements et la table des enregistrements vides.

5- Pour les fichiers chaînés possibilité de choisir la séquence de tri (modification de l'information "séquence" dans la table des enregistrements).

6- Modification des liens entre fichiers logiques existants à condition de conserver la racine en organisation indexée et des index secondaires compatibles avec celle-ci.

   Pour cela il faut intervenir dans la table des enregistrements au niveau des informations

   - numéro d'enregistrement père
   - liste des zones constituant la clé complète de l'enregistrement

7- Ajout de nouveaux index secondaires

Les tables concernées sont :

- dictionnaire (pour les informations ayant un impact sur ces index)
- table des enregistrements (pour décrire l'index)
- table des enregistrements vides (pour indiquer les traitements relatifs à la génération de ces nouveaux index)
- table des index.

8- Modification de la structure des index existants.

Ceci est réalisé en modifiant la table des enregistrements ou on indique la liste des informations composant l'index et éventuellement la table des index (s'il y a évolution de la liste des informations composant la clé du fichier principal) et le dictionnaire (si ce ne sont plus les mêmes zones qui impactent les index).

Il est à noter que tous ces points ne peuvent être réalisés que parallèlement à une réorganisation de la base de données (sauf le point 1 si la valeur par défaut en place est satisfaisante et le point 7).

En effet, de façon à garder une structure uniforme des informations, il est indispensable d'étendre les nouvelles caractéristiques à ce qui existe déjà.

Le rôle de réorganisation est très diversifié. Dans le point 4 par exemple il s'agit d'initialiser la valeur du pointeur sur l'extension à la valeur nulle.

<u>REMARQUE</u> : Chaque fois qu'il y a apparition d'un nouveau fichier le jeu d'ordres d'E/S correspondant est à inclure dans le noyau. C'est le cas pour les points 3, 4 et 7.

Enfin la souplesse du système est également caractérisée par trois propriétés se rapportant aux index secondaires et qui sont :

1- Gestion entièrement automatisée des index secondaires pour toutes les fonctions du noyau.

2- Possibilité prévue dans la sélection d'entrée par index secondaire à tous les niveaux hiérarchiques.

3- Constitution d'index secondaires à partir de n'importe quel fichier de données principal de la base.

Nous allons illustrer cette évolutivité sur deux exemples qui sont des cas réels et qui se sont produits lors du développement d'une nouvelle application concernant la prévoyance : la gestion des contrats prévoyance et des assurés.

Le premier exemple concerne la création d'un index prévoyance et le deuxième la création d'une base des assurés.

1- CREATION D'UN INDEX PREVOYANCE :

Les informations concernant le contrat prévoyance se situent dans le fichier groupes-risques pour la partie détail et dans le fichier établissement pour la partie générale.

Cependant bien peu d'entreprises ont un contrat prévoyance (moins de 10 %).

Il paraît donc très onéreux de parcourir toutes les entreprises et les établissements de la base pour n'en prendre en compte qu'un faible volume, dans le cas de traitements spécifiques à la prévoyance.

De ce fait la création d'un index prévoyance permettant d'accèder uniquement aux entreprises ayant des contrats prévoyance était nécessaire. C'est le quatrième index de la base entreprise.

Chaque fois qu'il y a pour un établissement, création ou modification des informations générales du contrat prévoyance, il y a répercussion dans celui-ci.

Nous décrivons ci-après sa procédure de mise en place :

1- Ajout de postes dans la table des chemins [choix d'une lettre matérialisant l'entrée par le nouvel index].

```
 1        B001033501 01BASE ENTREP.BASE        **   ENTREPRISE   **
 2        B101ATT0000 NOYAU        NOYAU
 3        B2010002CLE ENTREP. CLE  ENTREPRISE            ** PRIV **
 4        B301111     ENT/NUM.ENT ENTREPRISE / NUMERO ENTREPRISE
 5        B301111A    ENT/ADRESSE ENTREPRISE / MOT CLE ADRESSE
 6        B301111I    ENT/IDENT   ENTREPRISE / MOT CLE IDENTIF
 7        B301111N    ENT/MOTC GERENTREPRISE / MOT CLE GERANT
 8     -> B301111P    ENT/PREV    ENTREPRISE / CODE PREVOYANCE
 9        B3012212    ETAB/N. ENTREETABLISSEMENT / NUMERO ENTREP.
10        B3015215    HRQ ID/N.ENTHISTORIQUE IDENTIFICATION / NUM. ENTREP.
11        B3012242    ETAB/MC ADE ETABLISSEMENT / MOT CLE ADRESSE
12        B3015245    HRQ ID/MC AD HISTORIQUE IDENTIFICATION / MOT CLE ADRESSE
13        B3012212    ETAB/MC ID  ETABLISSEMENT / MOT CLE IDENTIFICATION
14        B3015215    HRQ ID MC IDHISTORIQUE IDENTIFICATION / MOT CLE IDENTIFICATION
15        B3012242    ETAB-MC GER ETABLISSEMENT / MOT CLE GERANT
16        B3015245    HRQ ID/MCG  HISTORIQUE IDENTIFICATION / MOT CLE GERANT
17     -> B3012212    ETAB/C.PRV  ETABLISSEMENT / CODE PREVOYANCE
18     -> B3015215    HRQ ID/C.PRVHISTORIQUE IDENTIFICATION / CODE PREVOYANCE
19        B3013323    CAT T  /NENTCATEGORIES TRANCHES  / NUM.  ENTREPRISE
20        B3014324    RISQUE / ENTRISQUE / NUMERO D'ENTREPRISE
21        B3016326    ADR ETB/NENTADRESSE ETABLISSEMENT / NUM. ENTREPRISE
22        B3017327    HRQ SG /NENTHISTORIQUE SITUATION GENERALE ETABL./ NUM. ENTREPRISE
23        B3013323    CATEG T/ADR CATEGORIES TRANCHES / MOT CLE ADRESSE
24        B3014324    RISQUES/ADR RISQUES / MOT CLE ADRESSE
25        B3016326    HRQ ADR/ADR HISTORIQUE ADRESSE ETABLISSEMENT / MOT CLE ADRESSE
26        B3017327    HRQ SGE/ADR HISTORIQUE SITUATION GENERALE ETABL/MOT CLE ADRESSE
27        B3013223    CAT T  /MCIDCATEGORIES TRANCHES  / MOT CLE IDENTIFICATION
28        B3014324    RISQUE /MCIDRISQUES / MOT CLE IDENTIFICATION
29        B3016326    ADR ETB/MCIDADRESSE ETABLISSEMENT / MOT CLE IDENTIFICATION
30        B3017327    HRQ SIT/MCIDHISTORIQUE SITUATION ETABLISSEMENT  / MOT CLE IDENTIF
31        B3013323    CAT T  /MCG CATEGORIES TRANCHES  / MOT CLE GERANT
32        B3014324    RISQUE /MCG RISQUES / MOT CLE GERANT
33        B3016326    ADR ETB/MCG ADRESSE ETABLISSEMENT / MOT CLE GERANT
34        B3017327    HRQ SG /MCG HISTORIQUE SITUATION GENERALE ETABL./MOT CLE GERANT
35     -> B3013323    CAT T /C.PRVCATEGORIES-TRANCHES / CODE PREVOYANCE
36     -> B3014324    RISQUE/C.PRVRISQUES / CODE PREVOYANCE
37     -> B3016326    HRQ.AD/C.PRVHISTORIQUE ADRESSE / CODE PREVOYANCE
38     -> B3017327    HRQ.SIT/C.PRHISTORIQUE SITUATION / CODE PREVOYANCE
```

2- Choix des informations ayant un impact sur cet index ici informations "situation prévoyance" dans le fichier établissement.

3- Choix des informations constituant l'index de façon à assurer l'unicité de l'enregistrement et détermination d'une clé d'interrogation.

4- Ajout de ces informations dans le dictionnaire en les associant au fichier logique pointé (ici entreprise).

BASE = 01 ********** NUMERO LOGIQUE = 1 NOM = E N T R E P R I S E ***

C.R.I.O.P - L.O.G.I.C DICTIONNAIRE DES INFORMATIONS SEQUENCE = BASE-F.LOGIQUE-INFO

BASE	F L	SEQ	INFO	RENS	PH	PH	P L T	EN	C L E	F	R	T Y P	T N U	S I G	SMT	POS	LNG	D É C	I N D	INFO	LIBELLE
01	1	940	62	0062	11			11				X			S	29	1			62	TYPE D' IDENT. (SIGLE/R.SOC.)
01	1	941	63	0063	11			11				X			S	2	28			63	CLE IDENT.
01	1	966	250	0250	11			11	C			X			S	2	17			250	CLE D' INTERRO. IDENTIFICATION
01	1	971	73	0073	10			15				X			S	2	2			73	CODE CAISSE SOUS CAISSE
01	1	973	74	0074	10			15				X			S	4	15			74	NOM
01	1	975	76	0076	10			15				X			S	19	8			76	CODE ENTREPRISE
01	1	978	79	0079	10			15				X			S	2	25			79	CLE NOM
01	1	979	251	0251	10			15	C			X			S	2	17			251	CLE D' INTERRO. NOM
01	1	981	80	0080	09			17				X			S	2	2			80	CODE CAISSE SOUS CAISSE
01	1	983	81	0081	09			17				X			S	4	15			81	MOT CLE LOCALITE
01	1	984	82	0082	09			17				X			S	19	15			82	MOT CLE RUE
01	1	985	83	0083	09			17				X			S	34	8			83	CODE ENTREPRISE
01	1	986	84	0084	09			17				X			S	42	3			84	NO. ETABLISSEMENT
01	1	987	85	0085	09			17				X			S	34	11			85	CODE ETABLISSEMENT
01	1	988	86	0086	09			17				D			S	45	2			86	DATE D EFFET ADRESSE
01	1	989	87	0087	09			17				X			S	2	45			87	CLE ADRESSE
01	1	990	68	0068	09			17	C			X			S	2	32			68	CLE D' INTERRO. ADRESSE
01	1	991	196	0196	12			08				X			S	2	2			196	CODE CAISSE SOUS-CAISSE
01	1	992	480	0480	12			08				X			S	4	8			480	CODE ENTREPRISE
01	1	993	482	0482	12			08				X			S	12	3			482	NO. ETABLISSEMENT
01	1	994	483	0483	12			08				X			S	4	11			483	CODE ETABLISSEMENT
01	1	995	484	0484	12			08				X			S	2	13			484	CLE INTERROGATION PREVOYANCE
01	1	996	408	0408	12			08				X			S	15	2			408	CODE SITUATION PRV
01	1	997	425	0425	12			08				D			S	17	2			425	DATE EFFET SITUATION PRV
01	1	999	489	0489	12			08	C			X			S	2	17			489	CLE INDEX PREVOYANCE

5- Description de cet index dans la table des enregistrements (autant de postes que de façons différentes de générer l'index (ici un seul poste) où on indique le numéro de poste index où est rangée la liste des numéros de zones constituant la clé d'accès au fichier de données associé.

```
107 DB ENTREPRISES - CATEG/TR.
    3   3  000 000  002 070 0175   3  0003 0091 0172 0000 0000  0   005 0161 0191   000   0000  0942  00   00
                                      0000 0000 0000 0000
108 DB ENTREPRISES - INDEX PREVOYANCE (ETABL.)
    P      000 000  000 018 0484   6  0282 0088 0089 0091 0129  0   012 0000 0000   000   0000  1012  02   2
                                      0128 0000 0000 0000
109 DB ENTREPRISES - LIBRE SERVICE
        1  000 000  001 161 0054   1  0003 0000 0000 0000 0000  1   007 0053 0000   000   0000  1030  00   00
                                      0000 0000 0000 0000
10 DB ENTREPRISES - COMPLEMENT ETABL.
        2  000 000  002 256 0166   2  0003 0091 0000 0000 0000  1   008 0165 0000   000   0000  1191  00   00
                                      0000 0000 0000 0000
11 DB ENTREPRISES - INDEX R.SOC.  (ENTR.)
    I      000 000  000 029 0250   6  0282 0001 0011 0002 0009  0   011 0000 0000   000   0000  1447  01   10
                                      0279 0000 0000 0000
12 DB ENTREPRISES - INDEX R.SOC.  (HIST. IDENT.)
    I      000 000  000 029 0250   6  0282 0200 0211 0201 0205  0   011 0000 0000   000   0000  1447  01   10
                                      0279 0000 0000 0000
```

6- Ajout dans le dictionnaire de numéros de poste dans la table index en regard des numéros de zone impactant l'index.

```
01 2  422  119 0119 02     02    X    SM  195   8          119 CODE ENTREPRISE ABSORBANTE
01 2  506  120 0120 02     02    X    SM  127  12          120 ZONE SITUATIONS ETABLISSEMENT
01 2  508  121 0121 02     02    X    SM  127   4          121 ZONE SITUAT. GEN.
01 2  509  122 0122 02     02    D    SM  127   2          122 DATE D'EFFET SITUAT. GEN.
01 2  512  123 0123 02     02    X    SM  129   2          123 CODE SITUAT. GEN.
01 2  513  124 0124 02     02    X    SM  131   4          124 ZONE SITUAT. RET.
01 2  529  125 0125 02     02    D    SM  131   2          125 DATE D'EFFET SITUAT. RET.
01 2  532  126 0126 02     02    X    SM  133   2          126 CODE SITUAT. RET.
01 2  543  127 0127 02     02    X    SM  135   4    26    127 ZONE SITUAT. PRV.
01 2  549  128 0128 02     02    D    SM  135   2    26    128 DATE D'EFFET SITUAT. PRV.
01 2  552  129 0129 02     02    X    SM  137   2    26    129 CODE SITUAT. PRV.
01 2  600  130 0130 02     02    D    SM  141   2          130 DATE SIGNATURE CONTRAT RET
01 2  602  149 0149 02     02    D    SM  139   2          149 DATE D' ADHESION RET.
01 2  603  347 0347 02     02    X    SM  176   6          347 NO. CONVENTION COLLECTIVE RET
```

7- Ajout dans la table des index d'un numéro de poste indiquant le numéro d'enregistrement de l'index (pointé par l'élixir) et d'un poste indiquant le détail des informations constituant la clé d'accès au fichier de données associé (pointé par la table des enregistrements).

```
024                                  Z. INDEX ENTR. CLE F.ASSOCIE
        02    0384 0387 0000 0000 0000 0000 0000 0000 0000
025                                  Z. INDEX PRV. CLE F. ASSOCIE
        02    0196 0480 0000 0000 0000 0000 0000 0000 0000
026                                  SITUATION PREVOYANCE ETAB.
        01    0008 0000 0000 0000 0000 0000 0000 0000 0000
027                                  DATE EFFET IDENTIFICATION
        02    0011 0013 0000 0000 0000 0000 0000 0000 0000
028                                  ENREG. ETAB.
        02    0017 0008 0000 0000 0000 0000 0000 0000 0000
```

8- Définition dans la table des enregistrements vides d'un poste permettant d'indiquer des valeurs pour lesquelles l'index ne doit pas être généré. (Dans notre cas, on désire ne pas générer d'index pour un code "situation" prévoyance à zéro).

```
LOC   OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER

                                  210 *
                                  211 *                          AEB120K - INDEX PREVOYANCE
                                  212 *                           DEPL. = 1012   LGR. 018
0003F4 50                         213          DC    01X'50'   C.ENREG.
0003F5 5050                       214          DC    02X'50'   C.CAISSE
0003F7 5050505050505050           215          DC    08X'50'   C.ENTR.
0003FF 505050                     216          DC    03X'50'   NO. ETABL.
000402 F0F0                       217          DC    CL02'00'  C.SIT.PRV.
000404 5050                       218          DC    02X'50'   DATE EFFET PRV.
                                  219 *
```

9- Ajout dans la table des fichiers logiques du numéro d'enregistrement correspondant à la lettre du chemin choisie.

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS

C00000                                1          CSECT
C00000 0000000000000000               2 TRENT1   DC    63X'00'
C0003F                      00000     3          CRG   TRENT1+C'A'-C'A'
C00000 0100000000000000               4          DC    XL9'0100000000000000001'
C00009                      00010     5          CRG   TRENT1+C'J'-C'A'
C00010 0000000001000100               6          DC    XL9'0000000001000000000'
C00019                      00021     7          CRG   TRENT1+C'S'-C'A'
C00021 0000000000000000               8          DC    XL8'0000000000000000'
C00029                      0002F     9          CRG   TRENT1+C'O'-C'A'
C0002F 0001020706040305              10          DC    XL10'0001020706040305000 0'
C00039                      0003F    11          CRG
C0003F 0000000000000000              12 TRENT2   DC    63X'00'
C0007E                      0003F    13          CRG   TRENT2+C'A'-C'A'
C0003F 1100000000000000              14          DC    XL9'1100000000000000008'
C00048                      0004F    15          CRG   TRENT2+C'J'-C'A'
C0004F 000000000F000800              16          DC    XL9'000000000F00080000'
C00058                      00060    17          CRG   TRENT2+C'S'-C'A'
C00060 0000000000000000              18          DC    XL8'0000000000000000'
C00068                      0006F    19          CRG   TRENT2+C'O'-C'A'
C0006E 0001020706040305              20          DC    XL10'0001020706040305000 0'
C00078                      0007F    21          CRG
```

10- Si des constantes non présentes dans la table des constantes entrent dans la composition de l'index ajout de celles-ci dans cette table. (ce qui n'est pas le cas ici)

De plus, vis à vis du système de télétraitement, il est nécessaire de définir le nouveau fichier et de l'inclure dans les utilitaires de sécurité associés à la méthode d'accès VSAM.

Enfin le jeu d'ordre d'Entrées/Sorties associé à un enregistrement index secondaire doit être inclus pour cet index au noyau.

Même si l'ajout d'un index entraîne pas mal de répercussions, nous constatons que sauf pour la dernière, il ne s'agit que de modifications de tables qui sont très rapides et beaucoup moins dangereuses que l'intervention dans des programmes.

2- CREATION D'UNE BASE DES ASSURES :

Nous étudions maintenant l'extension du noyau à une autre base.

Dans la méthode de Daniel MARTIN lorsque le besoin d'une nouvelle base se fait sentir, il y a écriture d'un nouveau noyau spécifique.

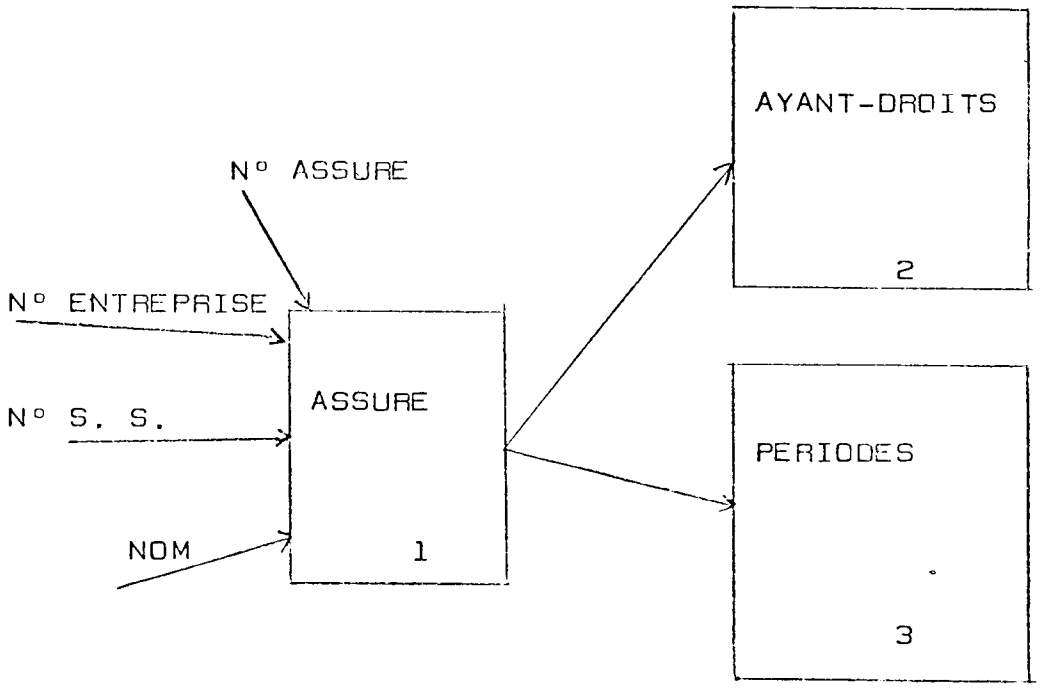
Nous nous sommes suffisamment attachés, au cours de la réalisation de notre système, à l'aspect généralisation pour que toutes les procédures écrites puissent être utilisées telles quelles pour les réalisations futures.

Le noyau que nous avons décrit est en fait apte à gérer n'importe quelle base de même structure que la base "entreprises" à condition d'y rajouter les ordres d'Entrées/Sorties nécessaires.

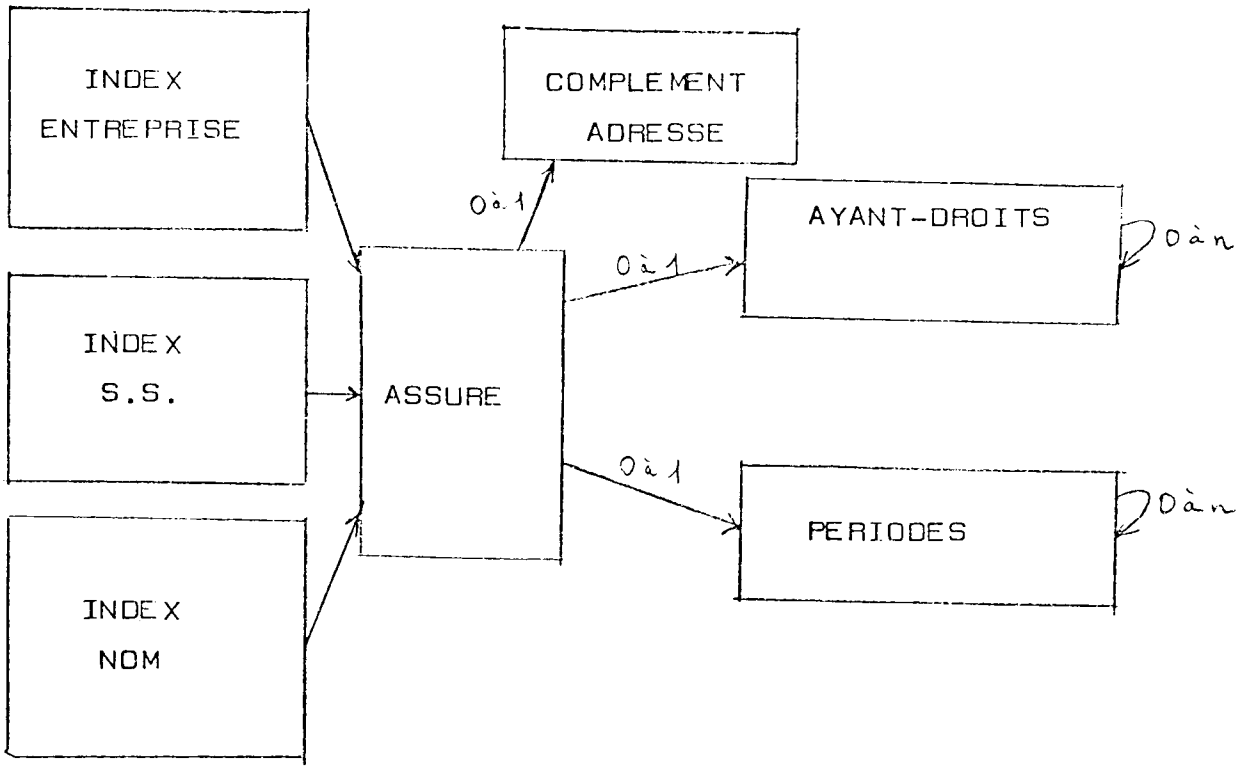
La création de la base des assurés, a bénéficié de cet esprit là.

Nous décrivons ci-après la procédure qui a permis sa mise en place.

L'analyse de l'application laisse apparaître trois entités qui sont : les assurés (dont on connaît l'adresse pour une faible partie), les ayants-droits qui dépendent de ces assurés et les périodes d'activités des assurés. De plus des accès sont nécessaires par numéro entreprise de l'assuré, numéro de sécurité sociale et nom de l'assuré. La structure logique retenue est la suivante :



La structure physique associée est la suivante :



Les étapes permettant la mise en place de cette base sont les suivantes : (Le degré de détail est moins fin que pour l'exemple précédent)

1- Attribution d'un numéro à la base.

2- Ajout de postes dans la table des chemins.

```
39      B0020215010 1BASE PREVOY.BASE        **   PREVOYANCE   **
40      B102ATT0000 NOYAU        NOYAU
41      B2020334CLE ASSURE. CLE  ASSURE                  ** PRIV **
42      B302111     IND/NUM.IND.INDIVIDU / NUMERO D'INDIVIDU
43      B30211N     IND/NOM     INDIVIDU / NOM
44      B30211S     IND/NUM.S.S.INDIVIDU / NUMERO DE SECURITE SOCIALE
45      B30211E     IND/NUM.ENT.INDIVIDU / NUMERO D'ENTREPRISE
46      B30211A     IND/NUM.ALL.INDIVIDU / NUMERO D'ALLOCATAIRE
47      B3022212    A.D/NUM.IND.AYANT DROITS / NUMERO D'INDIVIDU
48      B3023213    ACT/NUM.IND.ACTIVITES / NUMERO D'INDIVIDU
49      B30222N2    A.D/NOM     AYANT DROITS / NOM
50      B30232N3    ACT/NOM     ACTIVITES / NOM
51      B30222S2    A.D/NUM.S. S.AYANT DROITS / NUMERO DE SECURITE SOCIALE
52      B30232S3    ACT/NUM.S.S.ACTIVITES / NUMERO DE SECURITE SOCIALE
53      B30222E2    A.D/NUM.ENT.AYANT DROITS / NUMERO D'ENTREPRISE
54      B30232E3    ACT/NUM.ENT.ACTIVITES / NUMERO D'ENTREPRISE
55      B30222A2    A.D/NUM.ALL.AYANT DROITS / NUMERO D'ALLOCATAIRE
56      B30232A3    ACT/NUM.ALL.ACTIVIES / NUMERO D'ALLOCATAIRE
```

3- Définition et classement par fichiers logiques des informations devant figurer dans cette base y compris les informations figurant dans les fichiers index.

4- Choix des informations ayant des impacts sur les index.

5- Création du dictionnaire de données.

6- Description des enregistrements dans la table des enregistrements.

C.R.I.O.P. TABLE DES ENREGISTREMENTS AU 15 OCTOBRE 1980

NO. ENR	CHE FIC LOG	GRP	NBP	ENR. PERE	LGR	NO.Z CLE	NB.Z CLE	NO.Z CLE1	NO.Z CLE2	NO.Z CLE3	NO.Z CLE4	NO.Z CLE5	EXT	NO BUFF	PTR PERE	PTR CHAIN	NO.ENR EXT.	PTR EXT.	DEPL. ENR.V	NOPC INDX	NOPT INDX
019	DB INDIVIDUS - ASSURES																				
1	1	000	000	000	128	0334	1	0334	0000	0000	0000	0000	0	013	0000	0000	020	0354	1676	00	20
								0000	0000	0000	0000										
020	DB INDIVIDUS - ADRESSE ASSURES																				
	1	000	000	019	256	0357	1	0334	0000	0000	0000	0000	1	014	0354	0000	000	0000	1804	00	00
								0000	0000	0000	0000										
021	DB INDIVIDUS - AYANT DROITS																				
2	2	000	000	019	128	0400	2	0334	0399	0000	0000	0000	0	015	0353	0416	000	0000	2060	00	21
								0000	0000	0000	0000										
022	DB INDIVIDUS - PERIODE D'ACTIVITE																				
3	3	000	000	019	128	0423	2	0334	0422	0000	0000	0000	0	016	0352	0467	000	0000	2188	00	19
								0000	0000	0000	0000										
023	DB INDIVIDUS - INDEX S.S (ASSURE)																				
S		000	000	000	031	0473	5	0282	0332	0341	0333	0476	0	019	0000	0000	000	0000	2396	01	22
								0000	0000	0000	0000										
024	DB INDIVIDUS - INDEX S S (AYANTS-DROITS)																				
S		000	000	000	031	0473	5	0282	0396	0407	0397	0399	0	019	0000	0000	000	0000	2396	01	22
								0000	0000	0000	0000										
025	DB INDIVIDUS - INDEX NOM PATRONYMIQUE (ASSURE)																				
N		000	000	000	049	0468	7	0282	0332	0336	0478	0333	0	017	0000	0000	000	0000	2316	02	23
								0278	0476	0000	0000										
026	DB INDIVIDUS - INDEX NOM MARITAL (ASSURE)																				
N		000	000	000	049	0468	7	0282	0332	0337	0478	0333	0	017	0000	0000	000	0000	2316	03	23
								0474	0476	0000	0000										
027	DB INDIVIDUS - INDEX NOM PATRONYMICLE (AY.DROIT)																				
N		000	000	000	049	0468	7	0282	0396	0402	0479	0397	0	017	0000	0000	000	0000	2316	02	23
								0278	0399	0000	0000										
028	DB INDIVIDUS - INDEX NOM MARITAL (AY.DROIT)																				
N		000	000	000	049	0468	7	0282	0396	0403	0479	0397	0	017	0000	0000	000	0000	2316	03	23
								0474	0399	0000	0000										
029	DB INDIVIDUS - INDEX ENTR. (ACTIVITES)																				
E		000	000	000	031	0472	6	0282	0417	0477	0418	0422	0	018	0000	0000	000	0000	2365	01	24
								0424	0000	0000	0000										

7- Indication dans le dictionnaire de numéro de postes index en regard des numéros de zones impactant un ou des index.

<u>Exemple</u> :

BASE = 02 ********** NUMERO LOGIQUE = 1 NOM = A S S U R E S

C.R.I.O.P - L.O.G.I.C DICTIONNAIRE DES INFORMATIONS SEQUENCE = BASE-F.LOGIQUE-II

EA SE	F L	SEQ	INFO	RENS	PH	PH PT	PL T	EN	CLE F	RTYPU	TNIG	SMT	POS	LNG	DEC	IND	INFO	LIBELLE
02	1	001	332	0332	13			19		X		S	2	2			332	CODE CAISSE/SOUS CAISSE
02	1	002	333	0333	13			19		X		S	4	13			333	NO. ASSURE
02	1	003	334	0334	13			19	C	X		S	2	15			334	CLE ASSURE
02	1	100	335	0335	13			19		X		SM	17	80			335	ZONE IDENTIFICATION
02	1	105	492	0492	13			19		X		SM	17	4			492	TITRE ASSURE (MR,MME,MLLE)
02	1	110	336	0336	13			19		X		SM	21	20		14	336	NOM PATRONYMIQUE
02	1	120	337	0337	13			19		X		SM	41	20		15	337	NOM MARITAL
02	1	130	338	0338	13			19		X		SM	61	20		29	338	PRENOM
02	1	131	478	0478	13			19		X		S	61	10			478	PRENOM POUR INDEX
02	1	140	339	0339	13			19		D		SM	81	2			339	DATE DE NAISSANCE
02	1	150	340	0340	13			19		X		SM	83	1			340	CODE SEXE
02	1	160	341	0341	13			19		X		SM	84	13		13	341	NO. SECURITE SOCIALE
02	1	170	342	0342	13			19		X		SM	97	1			342	CODE SITUATION DE FAMILLE
02	1	200	343	0343	13			19		X		SM	98	4			343	ZONE INVALIDITE

8- Création de postes dans la table des index.

```
012                                          Z. INDEX ADRESSE CLE F.ASSOCIE
        02      0080 0083 0000 0000 0000 0000 0000 0000 0000
013                                          NO.S.S. ASSURE
        01      0023 0000 0000 0000 0000 0000 0000 0000 0000
014                                          NOM PATRONYMIQUE ASSURE
        01      0025 0000 0000 0000 0000 0000 0000 0000 0000
015                                          NOM MARITAL ASSURE
        01      0026 0000 0000 0000 0000 0000 0000 0000 0000
016                                          NO.S.S. AYANT-DROIT
        01      0024 0000 0000 0000 0000 0000 0000 0000 0000
```

9- Définition de nouveaux postes dans la table des enregistrements vides.

10- Ajout de numéros d'enregistrements dans la table de fichiers logiques (fonction du numéro de base et de la lettre de parcours)

```
00007E 0000000000000000                22 TRPRV1   DC    63X'00'
00008D                    0007E        23          CRG   TRPRV1+C'A'-C'A'
00007E 0000000013000000                24          DC    XL9'0000000013000000'
000087                    0008E        25          CRG   TRPRV1+C'J'-C'A'
00008E 0000000013000000                26          DC    XL9'0000000013000000'
000097                    0009F        27          CRG   TRPRV1+C'S'-C'A'
00009F 1300000000000000                28          DC    XL8'1300000000000000'
0000A7                    000AD        29          CRG   TRPRV1+C'O'-C'A'
0000AD 0013151600000000                30          DC    XL10'0013151600000000000'
000087                    000BD        31          CRG
0000BD 0000000000000000                32 TRPRV2   DC    63X'00'
0000FC                    000BD        33          CRG   TRPRV2+C'A'-C'A'
0000BD 0000000010000000                34          DC    XL9'0000000010000000'
0000C6                    000CD        35          CRG   TRPRV2+C'J'-C'A'
0000CD 0000000019000000                36          DC    XL9'0000000019000000'
0000D6                    000DE        37          CRG   TRPRV2+C'S'-C'A'
0000DE 1700000000000000                38          DC    XL8'1700000000000000'
0000E6                    000EC        39          CRG   TRPRV2+C'O'-C'A'
0000EC 0013151600000000                40          DC    XL10'0013151600000000000'
0000F6                    000FC        41          CRG
```

11- Eventuellement ajout d'informations dans la table des constantes.

Suite aux modifications de tables du SGBD les interventiona suivantes sont nécessaires :

- Définition des fichiers physiques pour le système de télétraitement.

- Ajout des nouveaux fichiers dans la liste traitée, par les utilitaires de sécurité (VERIFY) de la méthode d'accès VSAM.

- Ajout dans le noyau des jeux d'Entrées/Sorties pour chaque fichier physique.

Ces deux exemples marquent la fin de notre étude sur le SGBD réalisé au CRIOP.

V - B I L A N

L'ensemble des méthodes pratiques exposées par Daniel MARTIN, pour installer soi-même des bases de données, regroupe suffisamment d'idées et possibilités et d'une façon assez générale pour permettre de réaliser la plupart des aspects offerts par les SGBD présents sur le marché en 1978.

Cependant, par définition, ces méthodes s'appliquent à la résolution de problèmes concrets et amènent à un résultat trop spécifique. Il s'en suit un inconvénient important : le manque d'évolutivité. Nous avons été très sensibles à ce point, ce qui nous a conduit, tout en conservant les grands concepts de Daniel MARTIN, à penser un système plus général. Nous n'avons bien sûr pris en compte qu'une partie des idées énoncées par l'auteur, celles qui permettaient de répondre, à moyen terme, à nos propres besoins.

Nous n'avons, en particulier, pas prévu de liaisons logiques inter-base pour l'instant, ce qui nous classe dans la catégorie de modèles hiérarchiques non généralisés.

Nous n'avons également pas permis l'utilisation de tables de présence ou fichiers inverses, dont les besoins se feront peut être sentir ultérieurement.

Contrairement à ce que préconise Daniel MARTIN, les fichiers index devraient pouvoir être traités comme des fichiers logiques, c'est-à-dire lus indépendamment du fichier de données associé et susceptibles d'être sélectés à l'aide de n'importe quelle zone les constituant.

Enfin nous ne sommes pas en mesure de traiter avec efficacité toutes données pour lesquelles un modèle en réseau est nettement plus adapté. L'idéal serait de mettre en oeuvre un système conforme aux normes CODASYL en prenant en compte les deux aspects (hiérarchique et réseau).

Le fait que le noyau soit écrit en COBOL nous limite au point de vue généralisation des Entrées/Sorties. En effet toute nouvelle gestion de fichier se traduit par une recompilation du noyau. En outre le langage d'assemblage permettrait un gain de temps non négligeable dans l'exécution des requêtes.

Un certain nombre de points positifs se dégagent, néanmoins de cette première partie de réalisation (la version batch du noyau n'existe pas encore).

Le premier est l'aspect sécurité qui est réalisé pour une partie grâce à l'aide du système de télétraitement et sur lequel Daniel MARTIN ne nous semble pas insister suffisamment.

Le deuxième concerne l'aspect conversationnel du langage d'interrogation.

Il s'agit bien sûr d'un langage navigationnel car tout langage procédural ou assertionnel est très coûteux en réalisation surtout lorsqu'il est inclus dans un compilateur classique.

Enfin ce système nous a permis de réaliser des applications en termes de bases de données ce qui éventuellement aurait pu ne pas être le cas vu les raisons qui ont motivé ce choix.

V I - A N N E X E

C.R.I.O.P. TRACES D'ENREGISTREMENTS — CWA

CLIENT : _____ APPLICATION : _____ DEVIS : _____ DATE : _____

Bande ☐ LABEL
Disque ☐
Bande perfo. ☐ CONFIGURATION _____

Logique : Séquentiel ☐
Groupage : ISFMS ☐
DAM ☐

RESERVE SYSTEME				RECOVERY		
AFFECT (SSCAIS — TERMINEL)		P/MS BASE	N° CESSION	PTR DERNIER	PTR MAXI	tables systèmes (10 postes)

ELIXIR	PARC	ENREG	ENRGV	CSTES	FIC.LOG	INDEX						tables noyau (15 postes)

MAJOR ARRCO				- - - - - - -								tables applications (50 postes)

Fich 1 PTR DERNIER	Fich 2 PTR DERNIER	Fich 3 PTR DERNIER	- - - - - - -	Fich 25 PTR DERNIER		pointeurs fichiers utilisateurs (25 postes)

Fich 1 PTR DERNIER	Fich 2 PTR DERNIER	Fich 3 PTR DERNIER	- - - - - - -	Fich 50 PTR DERNIER		pointeurs fichiers bases de données (50 postes)

T. W.A. (200 octets)

CRIOP

APPLICATION ..... IDENTIFICATION .....

UNITE DE TRAITEMENT ..... CODE UT .....

AUTEUR ..... DATE .....

PAGE 1/1

9/ ENREGISTREMENT

9	8	7	6	5	4	3	2	1	0
BUFFER INDEX EN COURS							POINTEURS COURANTS		
CLE INDEX EN-COURS					CLE EN COURS INDEX PRINCIPAL		N° séquence de l'appel / N° buffer index	NIV 6 / NIV 5	NIV 4 / NIV 3 / NIV 2 / NIV 1

9	8	7	6	5	4	3	2	1	0
TYPE DE CONTRAINTE / TYPE D'ENTREE			CLE MAXI					CLE MINI	

TABLE DES B. L. L. (BASE LOCATOR FOR LINKAGE)

CRIOP

APPLICATION ..... IDENTIFICATION .....

UNITE DE TRAITEMENT ..... CODE UT .....

AUTEUR ..... DATE .....

9/ ENREGISTREMENT

0	1	2	3	4	5	6
@ CWA	@ ELIXIR + 12 K	@ TABLE ENREGISTREMENTS	@ TABLE ENR.VIDE + 4 K	@ SAVE Z. COMM.	@ TABLE EXPENSION CLES	@ TABLE SAVE R.R.N.
@ ELIXIR	@ ELIXIR + 8 K	@ TABLE PARCOURS	@ TABLE ENREG. VIDES	@ TWA	@ TABLE TRAVAIL/INDEX	@ TABLE ENREG. INDEX
@ ELIXIR + 4 K					@ TABLE FICH. LOGIQUES	

0	1	2	3	4	5	6	7
@ TABLE CONSTANTES	@ BUFFER 3					@ DES BUFFERS	@ BUFFER 50
@ BUFFER 1							
@ BUFFER 2							

TABLE DES ENREGISTREMENTS

CRIDP

APPLICATION ..... IDENTIFICATION .....

UNITE DE TRAITEMENT ..... CODE UT .....

AUTEUR ..... DATE .....

PAGE 1/1

9/ ENREGISTREMENT Longueur d'un poste 54 octets

Position	Rubrique
0	N° buffer
0	N° zone pointeur d'accès
0	N° zone pointeur de chainage
0	Facteur de groupage
0	Nombre de blocs/piste
0	N° enregistrement père
1	longueur du buffer
1	N° zone complète
1	Nbre zones constituant la clé
1	Zone 1
2	Zone 2
2	Zone 3
2	Zone 4
2	Zone 5
3	Zone 6
3	Zone 7
3	Zone 8
3	Zone 9
3	N° enregistrement extension
3	liste n° de zones constituant la clé complète
4	N° zone pointeur sur extension
4	Déplacement table enreg. vides
4	Indicateur extension
4	N° poste table clé index
5	N° poste table enreg. index
5	fichier logique
5	séquence organisation
5	fichier de données pointé si enreg. index
5	Autorisation duprec index

autorisation DUPREC index

Position	Rubrique
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	

ELIXIR DU DICTIONNAIRE DE DONNEES

CRIDP

APPLICATION ..... IDENTIFICATION .....

UNITE DE TRAITEMENT ..... CODE UT .....

AUTEUR ..... DATE .....

PAGE 1 / 1

9/ ENREGISTREMENT

1er poste

nbr	de postes qui suivent

autres postes

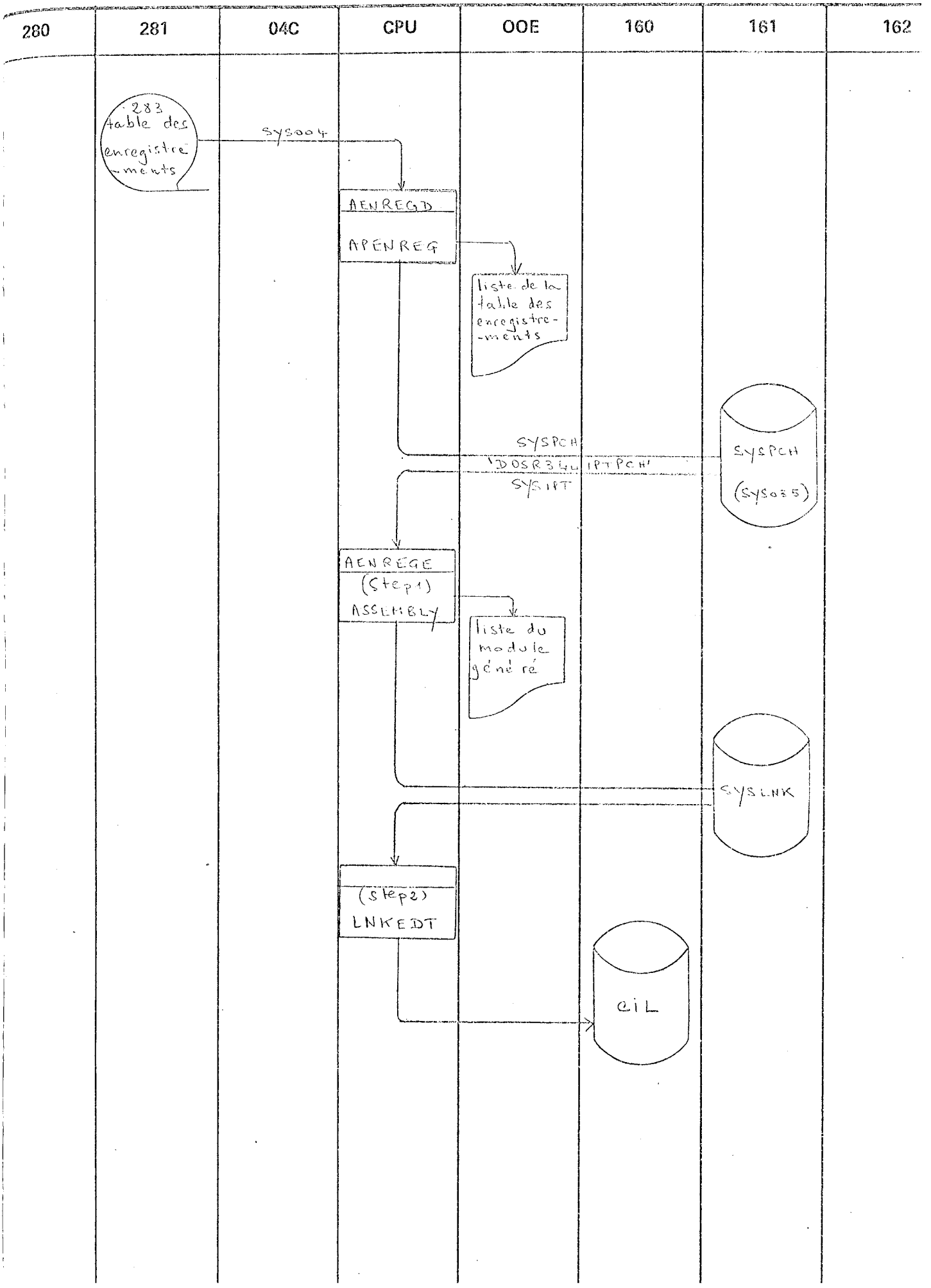
pointeur sur fichier libellés	
n° enregistrement	
caractéristiques zone	[1]
nature zone	[2]
nombre de décimales	
déplacement dans buffer	
longueur zone	
n° poste table enreg.index	

[1] bits 8 : constante
7 : réelle-fictive
6 : signe
5 :
4 :
3 : totalisable
2 : modifiable
1 : sélective

[2] bits 8 : alphanumérique
7 : numérique
6 : pointeur
5 : date
4 : numérique étendu
3 : condensé
2 : binaire 1/2mot
1 : binaire mot

NUMERO	ERREUR
1	MANQUE UN BLANC SEPARATEUR
2	QUESTION INCOMPLETE
3	ENTIER NON NUMERIQUE
4	OPERATEUR INCORRECT
5	PARCOURS ABSENT (MANQUE 'PAR')
6	CHAINE DU PARCOURS TROP LONGUE
7	NO. DE ZONE NON NUMERIQUE
8	CHAINE DE NON-PARCOURS TROP LONG
9	CHAINE NON PAR NON INCLUS DANS PAR
10	NON TROUVE
11	ENREGISTREMENT VSAM SUPPRIME
12	NUMERO DE ZONE N APPARTIENT PAS A LA BASE
13	NUM DE ZONE INCORRECT
14	PARTIE DECIMALE A PLUS DE 4 CHIF.
15	NOMBRE NON NUMERIQUE
16	NOMBRE A PLUS DE 13 CHIFFRES
17	VALEUR ALPHA-NUM INCORRECTE OU MANQUE CUCTE
18	QUESTION TROP LONGUE
19	PAS DE NUM DE ZONE APRES MON OU TOT
20	OPERATEUR LOGIQUE OU ET CONTRAINTES SUR CLES NON TRIEES
21	NUMERO DE SOUS-BASE INCORRECT
22	SOUS-BASE INCONNUE
23	INSTRUCTION SEL ABSENTE
24	TROP DE SEL
25	MANQUE MON OU TOT
26	NUMERO LOGIQUE A PLUS D'UN CARACTERE
27	ENREGISTREMENT SUPPRIME (CODE INTERNE)
28	******* MESSAGE NON GERE **********
29	******* MESSAGE NON GERE **********
30	******* MESSAGE NON GERE **********
31	******* MESSAGE NON GERE **********
32	******* MESSAGE NON GERE **********
33	******* MESSAGE NON GERE **********
34	******* MESSAGE NON GERE **********
35	******* MESSAGE NON GERE **********
36	******* MESSAGE NON GERE **********
37	******* MESSAGE NON GERE **********
38	******* MESSAGE NON GERE **********
39	******* MESSAGE NON GERE **********
40	******* MESSAGE NON GERE **********
41	******* MESSAGE NON GERE **********
42	******* MESSAGE NON GERE **********
43	******* MESSAGE NON GERE **********
44	******* MESSAGE NON GERE **********
45	******* MESSAGE NON GERE **********
46	******* MESSAGE NON GERE **********
47	******* MESSAGE NON GERE **********
48	******* MESSAGE NON GERE **********
49	VALEUR CONTRAINTE INCORRECTE
50	SOUS-BASE NON INTERROGEABLE
51	VALEUR ZONE INCORRECTE
52	TOTAL SUR UNE ZONE NON TOTALISABLE
53	SEL-MEME DEMANDE AVEC TOTAL
54	CODE ABSENCE D'ENREGISTREMENT ERRONNE
55	LONGUEUR ZONE COMMUNICATION < 400
56	ABSENCE ENREGISTREMENT
57	ESPACE DISQUE INSUFFISANT
58	CODE FONCTION ERRONNE
59	ZONE NON SELECTABLE
60	CODE PARCOURS NON AUTORISE
61	POINTEUR INVALIDE
62	ZONE POINTEUR NON RENSEIGNEE
63	NUMERO DE ZONE INCORRECT EN MODIFICATION OU ADDITION
64	MODIFICATION POUR ENREGISTREMENT INEXISTANT
65	ADDITION POUR ENREGISTREMENT EXISTANT
66	ERREUR ENTREE/SORTIE
67	NUMERO DE ZONE INCONNU DANS FICHIER
68	DEPASSEMENT DE CAPACITE SUR ENREG.
69	ZONE NON MODIFIABLE ET/OU NON TOTALISABLE
70	MODULE EMETTEUR INCORRECT
71	CONTRAINTES CONTRADICTOIRES
72	SEL-POINTEUR SUR ENTREPRISE INTERDIT
73	OPERATION INTERDITE POUR CE PARCOURS
74	CODE SOUS FONCTION ERRONNE
75	ZONES DE TRAVAIL INVALIDES POUR LA TRANSACTION
76	INCOHERENCE ENTRE FONCTION ET NOMBRE DE ZONES
77	NOMBRE DE CLES EN DESACCORD AVEC LE PARCOURS
78	PARCOURS INEXISTANT
79	ACTION INVALIDE
80	CODE PARCOURS AUTORISE EN SEL SEULEMENT
81	UN CODE PARCOURS AUTORISE SEULEMENT
82	PARCOURS EN DESACCORD AVEC LES NUMEROS DE ZONE
83	ADD/MOD PORTE SUR PLUS D'UN FICHIER LOGIQUE
84	DERNIERE LETTRE DU PARCOURS EN DESACCORD AVEC NUM DE ZONE
85	SE POINTEUR NON AUTORISE SUR ORGANISATION VSAM
86	AUCUN FLAG N'EST RENSEIGNE
87	ERREUR SUR FICHIER INDEX
88	MODIF ERRONEE SUR OPERATEUR CONTRAINTE CLE INDEX
89	RESTAURATION INTERDITE SUR FICHIER NON VSAM
90	ENREGISTREMENT NON SUPPRIME
91	MANQUE PLACE SUR FICHIER RECOVERY
92	ERREUR E/S SUR FICHIER RECOVERY
93	ZONE DE COMMUNICATION TROP LONGUE
94	******* MESSAGE NON GERE **********
95	******* MESSAGE NON GERE **********
96	******* MESSAGE NON GERE **********
97	******* MESSAGE NON GERE **********
98	******* MESSAGE NON GERE **********
99	******* MESSAGE NON GERE **********

CRIOP CREATION DE LA TABLE DES ENREGISTREMENTS



V I I - B I B L I O G R A P H I E

"BASES DE DONNEES : METHODES PRATIQUES" de DANIEL MARTIN

édité chez DUNOD