République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

Mémoire pour l'obtention
D'un diplôme d'ingénieur d'état en informatique.
Option : Intelligence Artificielle

Sujet :

Développement et Implémentation d'une Architecture Data Warehouse

Présenté par : Amaouz Amara
Ihaddadene Khaled

Promoteur : Mr Bala Mahfoud
Co-Promoteur : Mme Oukid Khouas

Soutenu le 02/10/2005, devant les jury suivant :

-Mme Ben Stitti Souad — **Présidente de jury**

-Mlle FAREH — **Examinatrice**

Numéro -2004/2005-

SOMMAIRE

Liste des Figures

Liste des Tableaux

Remerciements

Quelques lignes ne pourront jamais exprimés la reconnaissance que nous éprouvons envers tous ceux qui, de prés ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.

Nous tenons à exprimer toute notre reconnaissance à Mr. Bala & Mme Oukid, pour nous avoir proposer ce sujet, pour les conseils qu'ils n'ont cessé de nous prodiguer. Nous les remercions et tenons à leurs assurer notre profonde gratitude et grand respect.

Nous remercions S.Layla, A.Cherif et M.Nassia du groupe Oasys-PcCompo pour leurs aides, conseils et leurs disponibilités. Leurs précieuses remarques ont grandement contribué à notre travail. Qu'ils trouvent donc ici l'assurance de notre profonde gratitude.

Un grand merci envers, se qu'on a de plus cher au monde, nos parents et notre famille. Qui nous ont toujours soutenu et permis de mener à bien nos études.

Finalement, nos respects s'adressent aux membres de jury qui nous ferons honneur d'apporter des critiques et appréciation sur notre travail.

1. Introduction

Face à la mondialisation et à la concurrence grandissante, la prise de décision est devenue cruciale pour les dirigeants d'entreprises. Mais décider à partir de quoi ?
L'intuition, la réflexion et la prise de risques sont parfois payant mais ne suffisent pas car elles sont également à l'origine des décisions désastreuses pour les entreprises.

L'efficacité de cette prise de décision s'appuie sur la mise à disposition d'informations complètes, fiables et pertinentes avec des outils adaptés. Le problème des entreprises est d'exploiter efficacement d'importants volumes d'informations, qui constituent la principale matière première. Mais tout comme pour le charbon ou le pétrole, il faut l'extraire, le nettoyer, le raffiner et enfin le distribuer.

Les systèmes opérationnels s'avèrent inadaptés à une telle activité **[Kim, 97]** pour au moins deux raisons : les bases de données opérationnelles sont trop complexes pour pouvoir être appréhendées facilement par tout utilisateur et le système opérationnel ne peut être interrompu pour répondre à des questions nécessitant des calculs importants. Afin de pallier cet inconvénient, des systèmes décisionnels ont été développés.

Les systèmes de décisions s'intéressent au passé, au présent et au futur. Il faut donc garder un historique et restructurer les données de production, éventuellement récupérer des informations de différentes sources. Les entrepôts de données - **Data Warehouse** - peuvent stocker tous les événements qui surviennent dans la vie de l'entreprise, ils présentent les nouvelles architectures qui servent de fondations à de telles applications décisionnelles.
Un véritable défi est donc en train d'être posé aux entreprises étant données les contraintes qui pèsent sur elles. En amont, elles doivent consolider en quasi-temps réel les données provenant de sources diverses, internes ou externes, pour les rendre homogènes et exploitables. En aval, donner les bons outils, qui tirent le meilleur profit de l'architecture du data warehouse, aux utilisateurs finaux pour mieux comprendre et décider.

Ce projet de fin d'étude s'inscrit dans le cadre de Développement et D'implémentation d'une architecture Data Warehouse.

2. Problématique

Ayant pour objectif de préserver une part de marché, l'entreprise moderne se trouve face à des contraintes qui peuvent entraver l'aboutissement de ces préoccupations majeures. A noter que l'entreprise dispose des différents systèmes d'information. Les données contenues dans ces systèmes sont :

- Eparpillées
- Peu structurées pour l'analyse
- Focalisées pour améliorer le quotidien

De plus, un volume informationnel très important qui provient des différents flux entrants et sortants de ces systèmes a sérieusement contraint les décideurs à fournir beaucoup d'effort et perdre un temps considérable à collecter et analyser les données pour prendre des décisions, or ces dernières ne sont pas efficaces si elles ne sont pas prises au bon moment, ce qui entraînera les problèmes suivants :

- Manque d'une vision courante sur les clients et sur les personnels.
- Une mauvaise gestion des processus clés (stock,...).
- Indisponibilité d'un outil d'aide à la décision permettant de présenter des indicateurs pertinents.
- Retard dans l'établissement des rapports d'activité, cela peut prendre quelques jours voir quelques semaines. Ce qui fait que l'information n'est pas disponible au moment voulu.

Pour cela, le **Développement et l'Implémentation d'une architecture Data Warehouse**, pouvant regrouper les indicateurs pertinents de pilotage, permettra à l'entreprise de mieux contrôler son activité et maintenir son titre de leader dans un environnement où se développe de plus en plus l'aspect d'incertitude.

3. Objectifs

Notre objectif est le Développement et l'Implémentation d'une architecture Data Warehouse et cela à travers :

1) La conception et la réalisation d'un Data Warehouse qui a pour objectifs :

- Fournir des présentations claires et adaptées des données servant pour l'analyse.
- Analyser dynamiquement les résultats de requêtes
- Historiser l'information à des fins de statistiques et d'études

2) Le développement d'un outil pour la manipulation des données d'un data warehouse. Cet outil est développé dans un environnement DOT.NET. Il permet l'exploration et la publication des données. De plus :

- Il devra faciliter l'exploitation et l'analyse des données du data warehouse en produisant rapidement de nombreux graphiques et tableaux.
- Il devra simplifier la publication et la sauvegarde des rapports d'analyses et des consolidations pour une destination interne des différents utilisateurs.
- Offrir une meilleure accessibilité aux données.

4. Organisation du mémoire

Pour présenter notre travail nous avons retenu pour ce mémoire une organisation de 7 Chapitres.

- **Chapitre 1 :** afin de mieux comprendre pour quoi la solution Data Warehouise est-elle indispensable nous commencerons par une présentation générale du décisionnel au sein du système transactionnel comme problématique.
- **Chapitre 2 :** Explication et définition des objectifs et de l'architecture Data Warehouse
- **Chapitre 3 :** techniques de Modélisation des données du Data Warehouse
- **Chapitre 4 :** Nous parlerons des outils d'exploitation d'un data warehouse et les techniques de navigation, en mettant en évidence le concept OLAP. La Technologie OLAP
- **Chapitre 5 :** Explication des étapes de construction d'un data warehouse, d'après la méthodologie adoptée par Ralph Kimball, avec éclaircissement des différentes sous étapes de la modélisation dimensionnelle.
- **Chapitre 6 :** nous présenterons la construction de notre système en suivant la démarche de R.Kimball proposée ci-dessous et expliquée dans le chapitre 5 et cela comme suivant :

Conception de l'entrepôt de données et qui est repartie en trois grandes phases :

- Identification des besoins,
- Conception du modèle dimensionnel associé,
- Mise en œuvre de l'architecture en passant par trois étapes :
  - Construction de la base dimensionnelle.
  - Construction des cubes OLAP.
  - Construction de la zone d'alimentation.

- **Chapitre 7 :** Conception de l'application ainsi que ses différentes caractéristiques.

Pour finir par une conclusion générale.

A la fin, dans L'annexe, nous donnerons les différents outils et logiciels utilisés.

5. Methodologie suivie

Bien que les Data warehouse datent depuis, déjà des années, ils présentent toujours un thème de recherche à part entière. Les architectures du data warehousing et les systèmes décisionnels font l'objet de nombreuses études et travaux.

Pour mener à bien un projet quelle que soit sa nature, il faut suivre une méthodologie de travail bien appropriée telle que Merise pour les systèmes transactionnels.
C'est dans ce but, avant de démarrer dans notre projet il faut chercher une méthode ou une approche à suivre.
D'après nos résultats de recherche la méthodologie basée sur le cycle de vie décisionnel, proposé par Ralph Kimball (l'un des pionniers dans le domaine), s'avère la plus adaptée pour un tel système, les autres approches s'inscrive dans le génie logiciel sont orientées beaucoup plus vers le développeurs que vers les concepteurs.

Cette approche se résume en trois grandes étapes :

1. Etude préalable et définition des besoins :

Cette partie de l'étude ressemble à toute étape préliminaire à l'implantation d'un nouveau système d'information automatisé. Une bonne conception de l'entrepôt de données repose sur une bonne compréhension des besoins des utilisateurs finaux.
L'étude des besoins doit déterminer le contenu de l'entrepôt et son organisation

2. Modélisation multidimensionnelle C'est présenter un modèle idéal, en tenant compte des besoins recensés et des réalités des données disponibles. Ralph KIMBALL propose une démarche qui se résume à neuf points, dans un ordre bien défini.

3. Mise en oeuvre de l'entrepôt : en trois sous étapes successives :

- **Construction de l'entrepôt de données** : C'est transformer notre conception logique de données en une base physique.
- **Construction des cubes OLAP** : C'est le moteur qui permet de construire notre base multidimensionnelle en agrégeant les données stockées dans le data warehouse, l'objectif de ces cubes et de supporter efficacement les processus d'analyses de type OLAP.
- **Construction de la zone d'alimentation** : C'est la tâche la plus complexe, elle se réalise en trois étapes, l'extraction, la transformation et le chargement avec le choix des méthodes et des dates auxquelles les données entreront dans l'entrepôt.

CHAPITRE I :

LE DECISIONNEL AU SEIN DU SYSTEME TRANSACTIONNEL

1. Introduction

Avec la généralisation de l'informatique dans tous les secteurs d'activité, les entreprises produisent et manipulent de très importants volumes de données. Ces données sont stockées dans les systèmes opérationnels de l'entreprise au sein de bases de données, de fichiers... L'exploitation de ces données dans un but d'analyse et de support à la prise de décision s'avère difficile et fastidieuse ; elle est réalisée le plus souvent de manière imparfaite par les décideurs grâce à des moyens classiques (requêtes SQL) **[Tes00]**

Ces systèmes paraissent peu adaptés pour servir de support à la prise de décision. Ces bases opérationnelles utilisent le modèle relationnel, celui-ci convient bien aux applications gérant l'activité quotidienne de l'entreprise, mais s'avère inadapté au décisionnel **[Tes00]**

Face à cette inadéquation, il est fondamental de mettre en place une nouvelle informatique décisionnelle pour obtenir une meilleure compréhension de la valeur des informations disponibles, de définir des indicateurs pertinents pour faciliter la prise de décision. Cette nouvelle technologie est basée sur les entrepôts de données

2. Concepts de base [Nie98]

2.1. Les systèmes transactionnels

Egalement appelés systèmes opérationnels, ce sont les outils utilisés quotidiennement. Ils assurent le bon fonctionnement de l'ensemble de l'entreprise.

De la gestion des achats à celle des ventes, ils sont aujourd'hui indispensables au fonctionnement de toute entreprise. On trouve dans cette catégorie les progiciels horizontaux couvrant les grands métiers de l'entreprise : comptabilité, gestion commerciale, gestion des achats, gestion des stocks, paie et gestion des ressources humaines...

Les applications spécifiques, propres à une entreprise, développées pour répondre à une problématique métier ou à une particularité de gestion, figurent également dans cette catégorie : une application de réservation dans le secteur ferroviaire ou du transport aérien, une application de production bancaire…

Toutes ces applications répondent à la même attente : permettre la saisie d'informations, leur traitement, et la production en sortie de résultats, sous forme de documents papier, de consultations à l'écran ou d'autres informations.

Les principales caractéristiques des applications transactionnelles

Tout d'abord, elles brassent de grands volumes de données. Lorsqu'un système de réservation aérienne doit conserver la liste des places disponibles sur l'ensemble des vols d'une compagnie pendant les six mois à venir, en fonction des différents tarifs, c'est un volume de plusieurs gigaoctets de données (plusieurs milliards de caractères) qui doit être en permanence accessible.

Par ailleurs, cette masse d'information doit être accessible très rapidement. Lorsque un client se présente au guichet d'une agence de voyage, et souhaite réserver une place pour Montréal le 14 juillet prochain, il souhaite avoir la réponse en quelques secondes. Imaginant que l'agent de voyage lui demande de revenir le lendemain pour savoir si une place est disponible ! Le client changera vite de compagnie et d'agence.

Mais ces applications transactionnelles sont finalement relativement simples à programmer. Dans leur grande majorité, les requêtes générées par une application de production restent simples, du point de vue informatique. Même si la question de la disponibilité d'une place sur le vol Paris Montréal du 14 juillet semble compliquée, elle ne présente aucune difficulté du point de vue informatique.

Une simple requête sera envoyée, sur la base des vols programmés, pour connaître la disponibilité du vol demandé, et la réponse sera du même tenant : il s'agira de répondre si oui ou non une place est disponible, et éventuellement de fournir un numéro de siège.

Les trois principales caractéristiques d'un système transactionnel sont donc :

- La capacité à gérer de grands volumes de données.
- Des temps de réponse très réduits.
- Et des requêtes relativement simples du point de vue informatique.

2.2. Les systèmes décisionnels

A l'intérieur de l'entreprise, lorsque est venu le temps de l'analyse et de la réflexion, on doit se pencher sur les transactions enregistrées dans les systèmes opérationnels. Cette phase est un préalable à toute prise de décision. Ainsi, une compagnie aérienne souhaitera connaître le taux de remplissage moyen de ses vols par destination et par mois ; un grand magasin souhaitera connaître ses ventes par jour, puis par heure, pour chaque rayon ; et un financier établira le bénéfice de son entreprise détaillé par client, par produit, par secteur géographique.

Le principe même de la prise de décision est de s'appuyer sur des informations précises pour en déduire des comportements et passer à l'action.

Un système décisionnel est un ensemble de données organisées de façon spécifique, facilement accessibles et appropriées à la prise de décision .La finalité d'un système décisionnel est le pilotage de l'entreprise. **[Gog98]**

Les caractéristiques des applications décisionnelles

Tout d'abord, elles traitent également de gros volumes de données. Ainsi, lorsqu'un distributeur conserve l'ensemble du détail des tickets de caisse émis pendant plusieurs années, cela peut concerner des milliards d'informations unitaires, qui doivent être accessibles pour les analyses.

En revanche, les délais acceptables de ces analyses sont bien différents de ceux des requêtes opérationnelles. Autant il est primordial qu'un système répond en quasi temps réel aux demandes des clients, autant qu'il est certain d'attendre une minute, une heure, voir même parfois une journée, avant de disposer des informations analytiques sur des ventes depuis trois ans. Par opposition aux systèmes opérationnels, la notion de temps réel n'apparaît pas comme une contrainte à satisfaire. Mais des requêtes beaucoup plus complexes du point de vue informatique, car elles contiennent de nombreuses opérations de jointure et de regroupement et induisent des temps de réponse très élevés.

En conclusion donc, **les systèmes décisionnels** travaillent comme **les systèmes opérationnels**, sur de gros volumes de données, mais leur appliquent des requêtes beaucoup plus complexes. Mais ils disposent de plus de temps pour les exécuter.

2.3. Historique des systèmes décisionnels

Lorsque les entreprises ont commencé à comprendre la valeur ajoutée apportée par les outils d'aide à la décision, elles ont immédiatement cherché à en bénéficier. Pour cela elles ont commencé à lancer des requêtes, c'est- à dire à interroger leurs bases de données opérationnelles. Tout en poursuivant bien entendu leurs opérations quotidiennes, elles ont donc sollicité leurs applications opérationnelles, bien souvent au-delà de ce qu'elles étaient capables de supporter.

Parfois une requête envoyée sur les serveurs de production, bloquait les transactions et donc la vie de l'entreprise pendant plusieurs minutes, voir plusieurs heures. Les outils d'administration informatiques n'étaient à l'époque pas aussi perfectionnés que maintenant, et il était parfois impossible de stopper une telle requête.

Des situations de tension sont parfois apparues en interne dans certaines entreprises, conduisant à une opposition entre les « **opérationnels** » qui demandaient, à juste titre, une disponibilité permanente de leurs applications de production, et les « **analystes** », qui demandaient, également à juste titre, à pouvoir accéder aux données pour réaliser leurs analyses. Mais la différence structurelle entre les requêtes (simples pour les systèmes opérationnels, complexes pour le décisionnel) rendait cette cohabitation presque impossible, à moins pour les analystes de travailler la nuit, et encore !

2.3.1. L'Infocentre

La première solution trouvée était une solution de bon sens. Puisque les analystes avaient besoin des mêmes données que les opérationnels, et en même temps, mais que les requêtes des premiers bloquaient les requêtes des seconds, il suffisait donc de donner à chacun sa copie des données. On a donc dupliqué les bases de données de production, c'est-à-dire que chaque jour, chaque semaine, ou chaque mois, une copie des informations de production était réalisée, à l'identique, sur un autre ordinateur, spécialement pour les analystes (Voir figure 1).

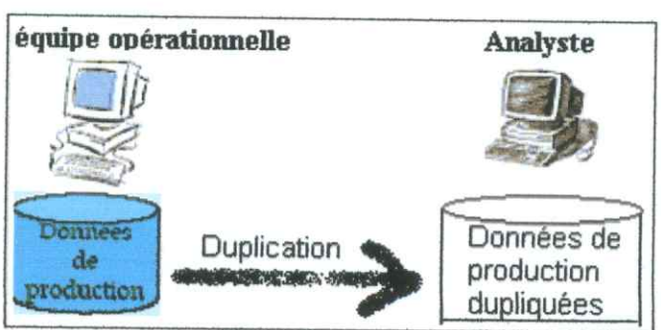


Fig. 1 l'Infocentre.

Les équipes opérationnelles pouvaient continuer à utiliser leurs applications, sans être nullement perturbés par les requêtes analytiques, et les analystes pouvaient prendre le risque de lancer des requêtes complexes, analysant par exemple le chiffre d'affaires suivant plusieurs dimensions (clients, produits, fournisseurs), sur plusieurs années, mois par mois, sans prendre le risque de bloquer le système opérationnel.

Outre sa « simplicité», ce mode de fonctionnement, en doublon total entre les deux systèmes, se révélait très coûteux. Les serveurs, les disques durs, les bases de données, devaient tous être acquis en double, uniquement pour les besoins de l'analyse.

2.3.2. L'Entrepôt de données

Après plusieurs années d'utilisation des infocentres, c'est-à-dire de duplications des données de production, les services informatiques ont imaginé une évolution intelligente de ce mode de stockage. Ils ont en effet constaté que les informations traitées dans les applications opérationnelles étaient très différentes de celles interrogées dans les applications décisionnelles. Le nombre de tables, de fichiers interrogés dans une même requête est bien plus important dans l'aide à la décision, le nombre d'indicateurs calculés également. En revanche, les applications décisionnelles se contentent presque toujours de lire les données. Elles n'ont jamais à écrire de nouvelles informations dans les bases de données. Autre constat, les questions posées par un décideur impliquent fréquemment des informations stockées dans plusieurs applications ou bases de données. Lorsque vous calculez la rentabilité de vos clients, vous exportez des données de la gestion commerciale (factures, commandes), de la comptabilité (délais de règlement, impayés), mais également de la gestion de production (coût des produits fabriqués). Le fait de dupliquer ces bases de données dans un infocentre ne simplifie en rien ces extractions.

Il a donc été imaginé mettre en place, en sortie des bases de production, un entrepôt de données. Cet entrepôt uniquement dédié au stockage des données décisionnelles, permet de réconcilier les différentes sources initiales de données, et les applications de production.

Fréquemment construit à partir d'une base de données relationnelle, cet entrepôt de données sert littéralement d'entrepôt. On y verse une copie des données qui serviront, un jour, à l'analyse et à la prise de décision. (Voir figure 2).

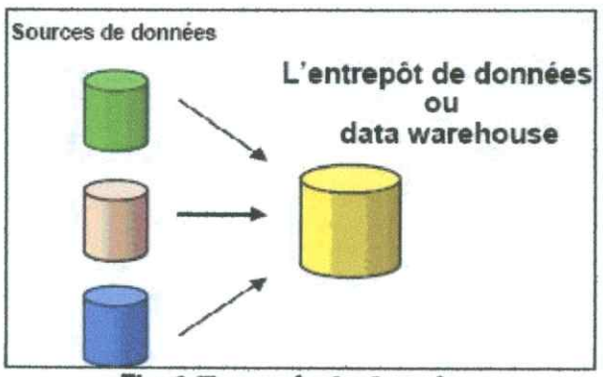


Fig. 2 Entrepôt de données.

2.3.3. Les bases de données multidimensionnelles

L'avez-vous remarqué ? Quand un manager parle de ses résultats, il leur associe toujours une variable. Il parle d'une progression d'un mois sur l'autre, d'une gamme de produit plus performante que l'autre, d'un bénéfice plus important dans un pays que dans l'autre, d'un commercial plus efficace que son collègue.

Tout simplement parce que d'un point de vue métier, l'information de base n'a aucune importance si elle n'est pas affectée d'une dimension. On cherche le chiffre d'affaires par trimestre, la rentabilité par client, la marge par commercial et par région.

Les bases de données relationnelles sont constituées d'un ensemble de tables à deux dimensions. Comme une suite de feuilles Excel, composées de lignes et de colonnes, qui seraient conservées dans un même classeur. Mais l'intérêt des bases de données relationnelles est dans le relationnel, c'est-à-dire dans la capacité à mettre en relation les feuilles les unes avec les autres. Ainsi une base de données **clients** comportera plusieurs tables : une table factures, une table client. Dans la première sera stocké le détail des factures, dans la seconde seront stockées les coordonnées des clients. Inutile donc de répéter les coordonnées dans chaque facture, un lien permettra d'aller chercher pour chaque ligne de la table facture les coordonnées correspondantes dans la table client.

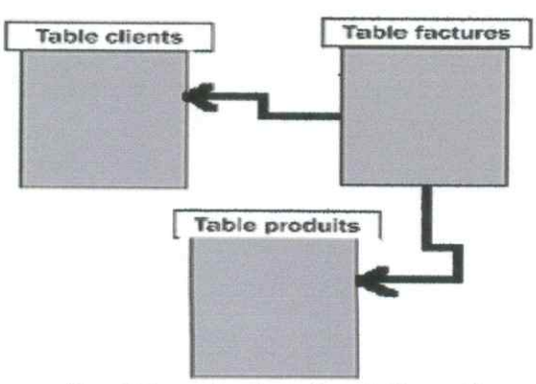


Fig. 3 Base de donnée relationnelle.

Lorsque l'on lance des **requêtes décisionnelles**, comme par exemple,

« Le calcul de la marge par client, et par gamme de produits, mois par mois depuis un an », le système a besoin d'accéder à de nombreuses tables. Une telle requête peut être très longue à exécuter, voir même parfois bloquer totalement le serveur.

On a, pour résoudre ce problème, inventé les bases de **données multidimensionnelles**. Ces bases se présentent sous forme d'un **cube** (voir Figure). Ce nouveau modèle de bases multidimensionnelles a été inventé par le Docteur **Edgard Codd**, déjà considéré comme le Père des bases de données relationnelles.

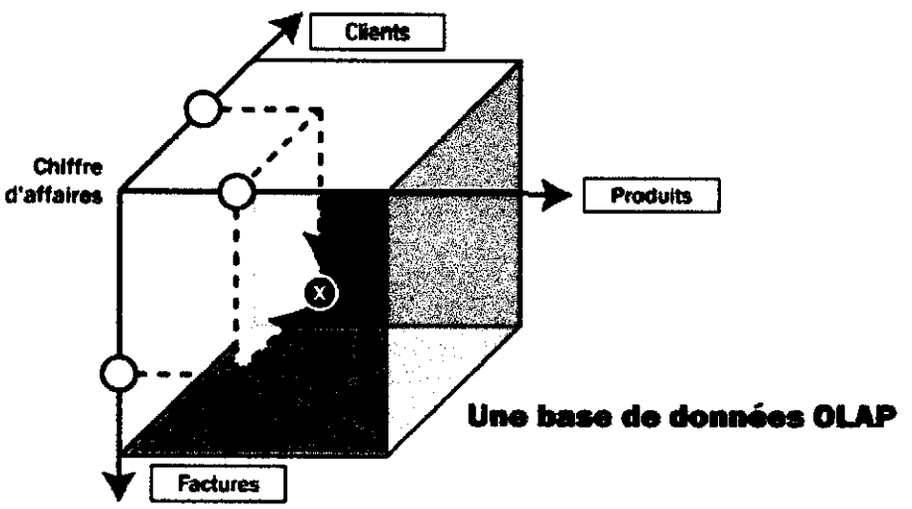


Fig. 4 Base Multidimensionnelle.

Chaque base de données **multidimensionnelle** pré -calcule et stocke toutes les informations, au croisement de chacune des dimensions. Le chiffre d'affaire réalisé par chaque client, sur chaque produit, dans chaque magasin, durant chaque mois, sera pré calculé, et conservé dans le cube, au croisement de chacune des dimensions prévues.

Ainsi, lors du lancement d'une requête décisionnelle complexe, le système aura simplement à lire les croisements de chaque colonne du cube, pour extraire des informations qui auront déjà été calculées.

La base de données **multidimensionnelle** est incontestablement le coeur de tout système décisionnel. La vision multidimensionnelle qu'elle apporte des données métiers en fait la pierre angulaire de la prise de décisions.

3. Conclusion :

Le développement des premiers systèmes d'informations s'est concentré sur l'automatisation des processus opérationnels, ainsi que sur les données liées aux processus.

Ces systèmes opérationnels permettent un gain de productivité non négligeable, ils soutiennent le bon déroulement de l'activité principale de l'entreprise, l'excellence opérationnelle.

Les besoins d'analyse sont arrivés bien plus tard, ils relèvent plus de l'avantage concurrentiel que de l'excellence opérationnelle.

Les SI décisionnels et les SI opérationnels ont des besoins et des structures totalement différents.

Le Système d'Information **opérationnel** doit être performant : les accès en écriture sont importants, même si prédéfinis, Tandis que les Systèmes d'Informations **décisionnels** doivent être flexibles, fournir des possibilités d'analyse conséquente. **[Bol02]**

Ainsi Les systèmes d'informations décisionnels sont nés d'un besoin des entreprises confrontées à une concurrence de plus en plus forte, des clients de plus en plus exigeants, des données de plus en plus surabondantes, non organisées dans une perspective décisionnelle et éparpillées dans de multiples systèmes hétérogènes, en fin, un contexte organisationnel de plus en plus complexe et mouvant.

Pour répondre à ces besoins non satisfaits par les systèmes de gestion de bases de données traditionnels, le nouveau rôle de l'informatique est de définir et d'intégrer une architecture qui serve de fondation aux applications décisionnelles : **le Data Warehouse.**

Le data warehouse correspond à l'exploitation des données de l'entreprise dans le but de faciliter la prise de décision par les décideurs, c'est-à-dire la compréhension du fonctionnement actuel et l'anticipation des actions pour un pilotage éclairé de l'entreprise.

CHAPITRE II

DATAWAREHOUSE : Objectifs, Définitions, Architectures

1. Introduction

L'entreprise actuelle "croule" sous les données. Cette surabondance a comme conséquence directe un rejet par saturation. Pourtant, les données représentent une mine d'informations. Elles sont un avantage dont l'entreprise doit tirer parti.

Pour cela, il est fondamental de mettre en place une nouvelle informatique décisionnelle pour obtenir une meilleure compréhension de la valeur des informations disponibles, de définir des indicateurs pertinents pour faciliter les prises de décisions opérationnelles et garder la mémoire de l'entreprise.

Pour répondre à ces besoins, le nouveau rôle de l'informatique est de définir et d'intégrer une architecture qui sert de fondation aux applications décisionnelles.

Cette architecture globale est le Data Warehouse. Le Data Warehouse est apparu ces dernières années suite à la convergence entre les nouveaux besoins en informations des entreprises et la capacité à intégrer et à mettre en oeuvre des technologies aptes à y répondre.

Ce travail présente les principaux concepts liés au data warehouse. Dans un premier temps, une définition du data warehouse est proposée, puis les principaux objectifs du data warehouse sont expliqués.

2. Objectifs du Data Warehouse

L'information est devenue vitale pour l'entreprise. Toutes les données, qu'elles proviennent du système de production de l'entreprise, des sources externes, vont devoir être organisées, coordonnées, intégrées et enfin stockées pour donner a l'utilisateur une vue intégrée et orientée métier. Le Data Warehouse doit :

- Rendre les informations de l'entreprise compréhensibles, facilement accessibles, rapides....
- Assurer des informations cohérentes.
- Avoir une source d'information souple et adaptable : l'entrepôt de données est conçu dans la perspective d'évoluer dans le temps.
- Produire des informations propres pour faciliter la prise de décision, est le véritable résultat concret du Data warehouse.

3. Data Warehouse :

3.1. Définition :

Un data warehouse est un entrepôt de données. Il s'agit d'un stockage intermédiaire des données issues des applications de production, dans lesquelles les utilisateurs finaux puisent avec des outils de restitution et d'analyse. **[Gog00]**

Bill Inmon est considéré comme le père du concept. Dans son ouvrage de référence "***Building the Data Warehouse***" **[Inm94]**. Il définit l'entrepôt de données comme :

"***Une collection de données orientées sujet, intégrées, non volatiles, historisées, organisées pour le support d'un processus d'aide à la décision,***".

Cette définition englobe différents termes que nous explicitons.

3.1.1. Orientation sujet du Data Warehouse :

Le Data Warehouse est organisé autour des sujets qui ont un intérêt majeur pour l'entreprise. On assemblera à cet effet les informations par thèmes contrairement aux modélisations traditionnelles (transactionnel) qui regroupent les informations par fonctions. L'intérêt de cette organisation est de passer d'une vision vertical de l'entreprise a une vision transversal beaucoup plus riche.

Par exemple, le sujet client pourra être intégré dans un système décisionnel à caractère Marketing, un deuxième orienté vers l'administration des ventes, et un troisième à des fins d'analyse financière.

3.1.2. Données Intégrées :

Les données qui alimentent le data warehouse proviennent de multiples sources de données hétérogènes. Les données des systèmes de production doivent être intégrées de façon à avoir une seule vision globale dans le data warehouse. L'intégration consiste à résoudre les problèmes d'hétérogénéité des systèmes de stockage, des modèles de données, de sémantique de données.

3.1.3. Données historisées :

Dans les systèmes de production, les données sont mises à jour à chaque nouvelle transaction, l'ancienne valeur est perdue, par contre dans un data warehouse la donnée ne doit jamais être mise à jour, elle représente une valeur insérée à un certain moment. Le data warehouse stocke l'historique, c'est-à-dire l'ensemble des valeurs que la donnée aura prise au cours du temps. Dans un Data warehouse la valeur d'un fait à un instant "T" n'a en soi pas beaucoup d'intérêt, mais l'évolution de cette état dans le temps est fondamental : plutôt que de savoir que le client X a achète tel produit tel

jour, il est plus intéressant de constater que ce client consomme davantage, ou que le produit qu'il a acheté se vend mieux dans les périodes de grande chaleur.

Un référentiel de temps doit être donc associé à la donnée afin de tracer son évolution.

3.1.4. Données non volatiles :

Les données du data warehouse sont utilisées en mode consultation, elles ne sont pas modifiées par l'utilisateur. En théorie, une requête lancée à différentes dates sur les mêmes données doit retourner les mêmes résultats.

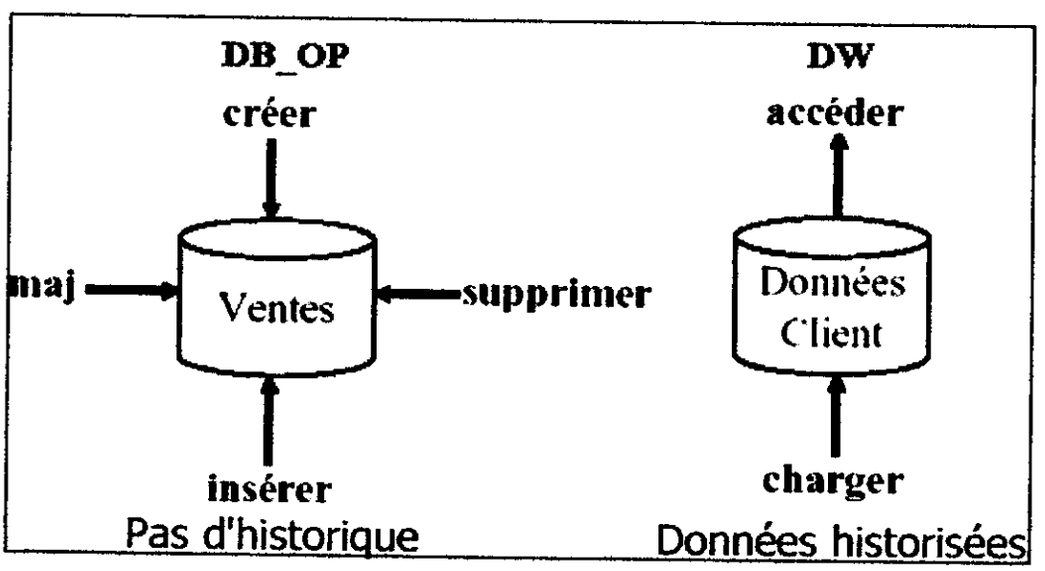


Fig. 5 Données Non Volatiles.

3.2. Structure d'un Data Warehouse :

Un data warehouse se structure en quatre classes de données, organisées selon un axe historique et un axe synthétique comme le montre le schéma suivant :

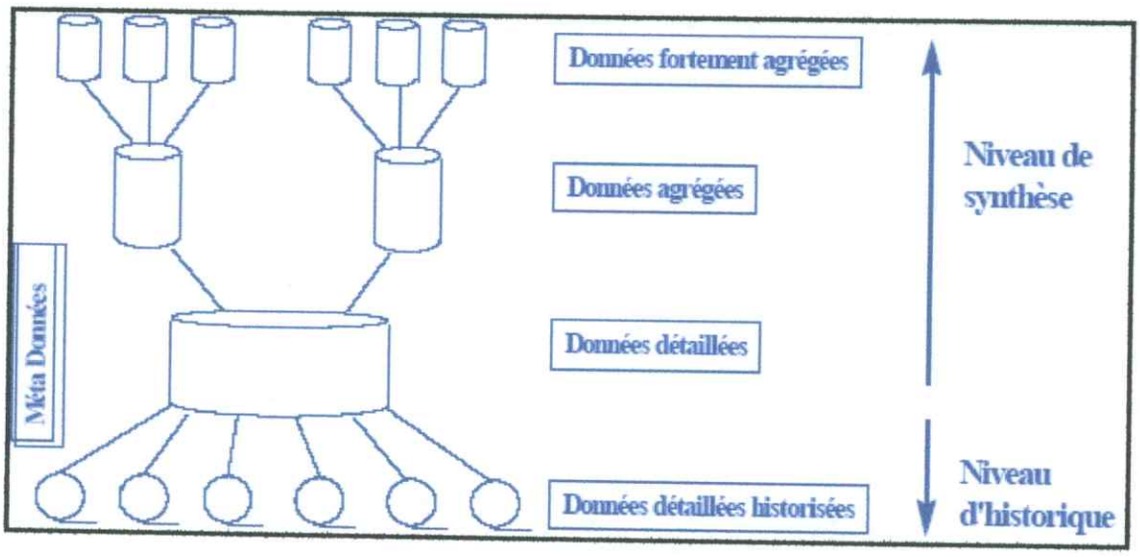


Fig.6. La structure du data warehouse.

3.2.1. Données détaillées : Elles reflètent les événements les plus récents. Les intégrations régulières des données issues des systèmes de production vont habituellement être réalisées à ce niveau. « Les volumes de données peuvent devenir très importants du fait, d'une part, du positionnement transversal, orienté sujet et intégré du Data Warehouse et, d'autre part, de l'historisation des valeurs de détails. »

3.2.2. Données agrégées : Les données agrégées sont très souvent utilisées car elles correspondent à des éléments d'analyse représentatifs des besoins des utilisateurs. Elles constituent Déjà un résultat d'analyse et une synthèse de l'information contenue dans le Système décisionnel.

Les agrégations sont des données de synthèse précalculées qui améliorent les temps de réponse, tout simplement parce que les réponses sont prêtes avant que les questions ne soient posées.

3.2.3. Méta données (données sur les données):

Les Metadonnées constituent l'ensemble des données qui décrivent des règles ou processus attachés à d'autres données concernant le Data Warehouse. Elles sont intégrées dans un référentiel (dictionnaire des données).

Ces dernières permettent de connaître la réponse à chacune des questions suivantes :

- Comment trouver l'information dont j'ai besoin ?
- D'où ces données proviennent-elles ?
- Comment ont-elles été agrégées ?
- Quelles sont les requêtes d'accès disponibles ?

- Ont-elles été modifiées et comment ?
- Comment les définitions des métiers ont-elles évolué ?
- De quel historique dispose-t-on ?

3.2.4. Données historisées : Un des objectifs du Data Warehouse est de conserver en ligne les données historisées. Chaque nouvelle insertion de données provenant du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée.

3.3. Architecture et Implémentation : [Sch02]

Pour implémenter un data warehouse, trois types d'architecture sont possibles :

3.3.1. Architecture réelle : l'architecture réelle est généralement l'architecture utilisé pour les systèmes décisionnels. Le stockage des données du Data Warehouse est réalisé dans un SGBD séparé du système de production. Ce SGBD est alimenté par des extractions périodiques. Avant le chargement, les données subissent d'importants processus d'intégration, de nettoyage et de transformation. L'avantage de cette solution est de disposer de données préparées pour les besoins de la décision et répondant bien aux objectifs du Data Warehouse. Les inconvénients de cette architecture sont le coût de stockage supplémentaire et le manque d'accès en "temps réel".

3.3.2. Architecture virtuelle : Dans une architecture virtuelle, les données du Data Warehouse résident dans le système de production. Elles sont rendues visibles par des produits middleware ou par des passerelles. Il n'y a pas dans cette architecture de coût de stockage supplémentaire et l'accès se fait en temps réel. Cependant l'inconvénient de ce type d'architecture est que les données ne sont pas préparées.

3.3.3. Architecture remote : L'architecture remote est une combinaison des deux types d'architectures décrites précédemment. L'objectif est d'implémenter physiquement les niveaux agrégés afin de faciliter l'accès et garder le niveau de détail dans les systèmes de production en y donnant accès par le biais de middleware ou de gataways.

Cette architecture comme la précédente est rarement utilisée.

Les différents éléments d'appréciation sont repris dans le tableau récapitulatif ci-dessous :

	Architecture Réelle	Architecture Virtuelle	Architecture Remote
Utilisateur	Retenue pour les Systèmes décisionnels	Rarement utilisée	Rarement utilisée
Stockage	SGBD séparé du système de production, alimenté par des extraction périodique	Données résidant dans le système de production	Combinaison des Architectures réelle et virtuelle
Avantages	Données préparées Pour les besoins de la décision	Pas de coût de Stockage Supplémentaire, Accès en temps réel	
Inconvénients	Coût de stockage Supplémentaire, Manqué d'accès en Temps réel	Données non préparées	

Tab.1. comparaison entre différentes architectures.

3.4. Composants de base d'un Data Warehouse :

Les entreprises passent à l'ère de l'information. Transformer leur système d'information qui avait une vocation de production à un système décisionnel dont le but est le pilotage de l'entreprise.

Dans ce paragraphe, on présente les principaux composants de base d'un data warehouse, selon KIMBALL

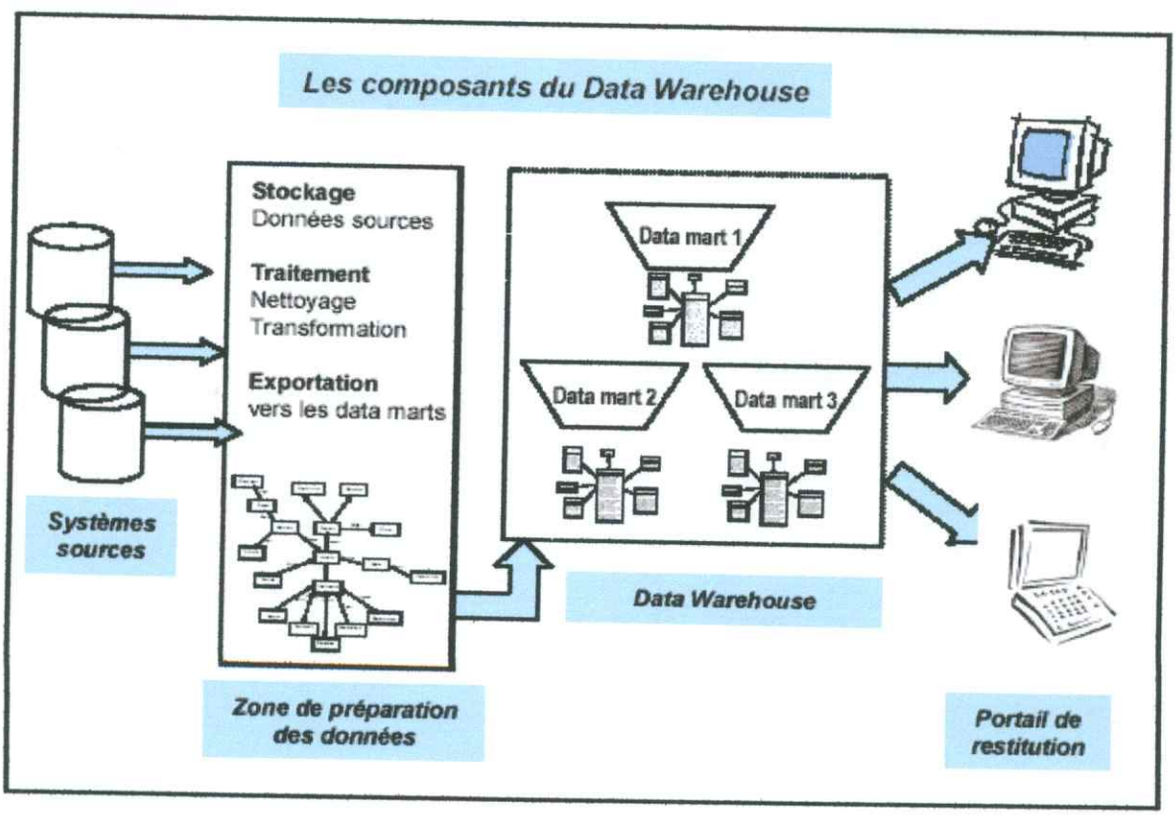


Fig. 7. Composant de base d'un Data Warehouse.

3.4.1. Systèmes Sources :

C'est le système opérationnel d'enregistrement, dont la fonction consiste à capturer les transactions liées à l'activité de l'entreprise **[Kim97]**. Il s'agit souvent de ce que l'on appelle les applications de gestion.

On appelle, d'une façon générale base de production toutes les sources (qu'il s'agisse de données de production, d'informations internes et d'informations externes quel que soit leur mode de stockage) dont il va falloir extraire des données en vue d'alimenter le data warehouse. **[Gog 01]**.

3.4.2. La Zone de Préparation des Données :

Elle regroupe l'ensemble des processus qui nettoient, transforment, combinent, archivent, suppriment les doublons. Elle prépare les données sources en vue de leur intégration et de leur exploitation dans le data warehouse.

3.4.3. Data Warehouse (Base de l'Entrepôt de données) :

Est le lieu de stockage centralisé des informations utiles pour les décideurs, dans lequel les données hétérogènes des sources sont intégrées et stockées durablement.

3.4.4. Le Serveur de Présentation :

Ce composant correspond à la machine cible sur laquelle l'entrepôt de données est stocké et organisé pour répondre en accès direct aux requêtes provenant des utilisateurs, des générateurs d'états ou d'autres applications.

Sur le serveur de présentation, les données sont stockées sous forme dimensionnelle afin de faciliter l'accès aux utilisateurs finaux.

3.4.5. Data Marts (Magasins de Données) :

Un data mart est un magasin de données. Il s'agit d'une solution départementale d'entrepôt de données supportant une partie des données. C'est un sous ensemble du data warehouse qui ne contient que les données d'un métier de l'entreprise alors que le data warehouse contient toutes les données décisionnelles de l'entreprise pour tous les métiers.

3.4.6. Le Portail de Restitution:

C'est la partie publique du data warehouse. Il représente ce que voient les utilisateurs, les outils avec lesquels ils travaillent. Les services offerts par le portail de restitution sont les services d'accès aux données, les applications de modélisations et le data Mining.

Les services d'accès aux données comprennent : la navigation dans le data warehouse, la gestions des requêtes et la généralisation d'états standards. Les applications de modélisation offrent différent type d'analyses basées sur des modèles financiers, système d'évaluation de clientèles, et offrent aussi une analyse sur l'activité centrale du data Mining telles que la catégorisation, la classification et le regroupement par affinité.

3.5 L'alimentation d'un Data Warehouse [W.01]

3.5.1. La Problématique :

Pour alimenter un data warehouse, il est intéressant d'extraire la matière vive des bases de production. Pour cela, il faut avoir préalablement identifier les données intéressantes de celles qui ne le sont pas. On devra alors se poser les questions adéquates :

- ✓ L'adresse complète du client est-elle utile ou le code postal suffit-il ?
- ✓ L'age du client est-il nécessaire ?

Ces données ainsi réorganisées deviennent des informations utiles pour le décisionnel.

La première phase de la construction d'un data warehouse consiste généralement à extraire les données utiles des systèmes opérationnels qui dans de nombreux cas sont hétérogènes, diffusées et complexes.

- Elles sont hétérogènes car bien souvent on rencontre plusieurs SGBD différents
- Elles sont diffusées car on les trouvera au sein de plusieurs environnements matériels, éventuellement relies par plusieurs réseaux interconnectés différents.
- Elles sont complexes, car on rencontrera plusieurs modèles logiques et physiques prioritairement orientés vers les traitements complexes.

Des solutions logicielles sont alors nécessaires à leur intégration et à leur homogénéisation. Ces outils ont pour objet de s'assurer de la cohérence des données du Data Warehouse et d'homogénéiser les différents formats trouvés dans les bases de données opérationnelles.

3.5.2. Les fonctionnalités d'un outil d'alimentation :

Un outil d'alimentation doit être capable d'extraire des données au niveau de la source, de les transformer en vertu des règles très précises, puis de les injecter au sein d'un système décisionnel cible.

L'outil apportera une aide précieuse s'il est capable d'automatiser l'ensemble de ces tâches.
Le choix de l'outil d'alimentation dépendra d'un certains nombres de facteurs, le volumes d'informations à extraire, l'hétérogénéité des sources de données, la fréquences d'extraction ...
Il est de plus nécessaire d'adopter un outil capable d'accéder nativement aux moteurs de SGBD.

3.5.3. Les phases de l'alimentation du Data Warehouse sont les suivantes :

- Découvrir quelles sont les données à faire migrer.
- L'acquisition des données se déroule en trois phases :
  - l'extraction,
  - la transformation
  - le chargement.

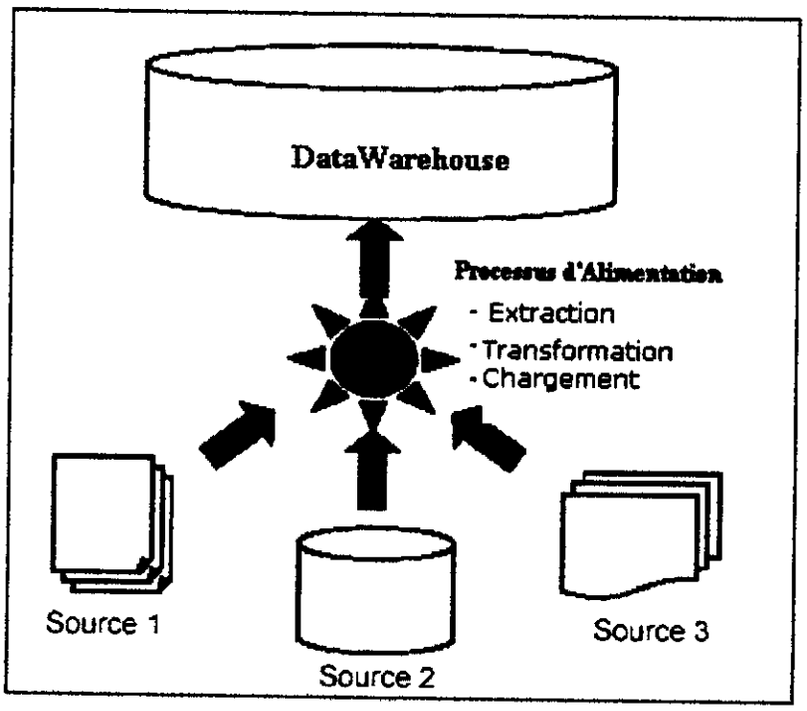


Fig. 8. Processus d'Alimentation.

a) La découverte des données

La découverte des données consiste à les localiser dans le système opérationnel et à prendre les plus judicieuses.
Un mauvais choix des données extraites va complexifier les phases suivantes de l'alimentation.

b) L'Extraction

L'extraction des données consiste à collecter les données utiles dans le système de production à l'aide des outils spécialisés comme « E.T.L » (Extracting-Transforming-Loading). Pour rafraîchir la base

décisionnelle, il faut identifier les données ayant évolué afin d'extraire le minimum de données, puis planifier ces extractions afin d'éviter les saturations du système de production.
De plus, la forme des données externes, qui est souvent totalement anarchique accentue la difficulté. Pour être utiles, ces données nécessitent un reformatage pour pouvoir les incorporer dans une forme exploitable pour l'entreprise.

c) La transformation des données

Le nettoyage des données est une discipline sur laquelle de nombreux éditeurs travaillent actuellement. Outre la qualité des données qu'ils permettent d'auditer et éventuellement d'améliorer, les outils de nettoyage permettent de supprimer les doublons dans les fichiers. Il s'agit à ce stade d'appliquer des filtres prédéfinis sur les données afin d'attribuer des valeurs cohérentes aux variables mal ou non renseignées ou encore d'harmoniser les formats (date : jj/mm/aaaa). On peut également avoir à convertir les données d'un format EBCDIC vers ASCII. Sachant en fin que les données du système opérationnel doivent être agrégées ou calculées avant leur chargement dans la base décisionnelle.
Il faut également pouvoir associer des champs sources avec des champs cibles, comme par exemple :

- le transfert du "*nom du client*" vers un champ cible.
- la décomposition d'une "*adresse*" vers les champs "*numéro*", "*rue*", "*ville*" ou l'inverse.

Certains outils peuvent également réaliser des analyses lexicales des champs sources. Ils seront donc capables de comprendre que les champs suivants signifient la même chose :"*Boulvd*", "*Bd*,"*Boulevard*".

En complément, on trouve des outils d'audit et d'analyse pour assurer le suivi du processus afin notamment de contrôler les rejets.

d) Le chargement des données

Le chargement est la dernière phase de l'alimentation du Data Warehouse. C'est une phase délicate notamment lorsque les volumes sont importants. Pour obtenir de bonnes performances en chargement, il est impératif de maîtriser les structures du SGBD (tables et index) associées aux données chargées afin d'optimiser au mieux ces processus.

3.6. Exploitation et utilisation de l'information

Une fois que les données se trouvent dans le data warehouse, il ne reste plus qu'à les exploiter.
L'utilisateur final doit alors pouvoir interroger les données en ligne à l'aide d'outils simples et conviviaux, qui leurs permettent de répondre à leurs nouveaux besoins.

CHAPITRE III

MODÉLISATION DES DONNÉES DU DATA WAREHOUSE

1. Introduction

Modéliser est une tâche délicate car elle met en jeu des populations différentes. Les administrateurs de données et les utilisateurs "métiers" doivent créer un modèle global et cohérent pour l'entreprise.

Les données du Data Warehouse doivent être orientées " sujet ", facilement compréhensibles et utilisables par les utilisateurs.

La conception d'un entrepôt est très différente de celle d'une base de données pour un système transactionnel, Les concepts sont plus ouverts et plus difficiles à définir. De plus, les besoins des utilisateurs de l'entrepôt ne sont pas aussi clairs que ceux des utilisateurs des systèmes transactionnels. Les modèles de données utilisés dans la conception des systèmes transactionnels traditionnels ne sont pas adaptés aux requêtes complexes. En effet, les requêtes dans les systèmes transactionnels sont simples (liste des commandes d'un client ou adresse et numéro de téléphone d'un client), alors que, dans les entrepôts, les requêtes utilisent beaucoup de jointures, demandent beaucoup de temps de calcul. Pour ce type d'environnement, on a suggéré une nouvelle approche de modélisation : les modèles multidimensionnels.

2. Modélisation relationnelle

2.1 modèle de donnée normalisée

Le modèle présenté ici est relativement simple est très classique, très utilisé dans les entreprises en matière de modélisation. Si le système à mettre en œuvre permet à la fois des sélections et de la mise à jour en ligne, alors le modèle est bien adapté. Dans un contexte décisionnel ou les mises à jour ne sont pas d'actualités, sa pertinence doit être reconsidéré.

La première remarque qu'il faut faire sur ce modèle est que dans un point de vue décisionnel, la sémantique est faible. Les informations intéressantes pour l'utilisateur n'existe pas a priori, elle doivent être extrapolés.

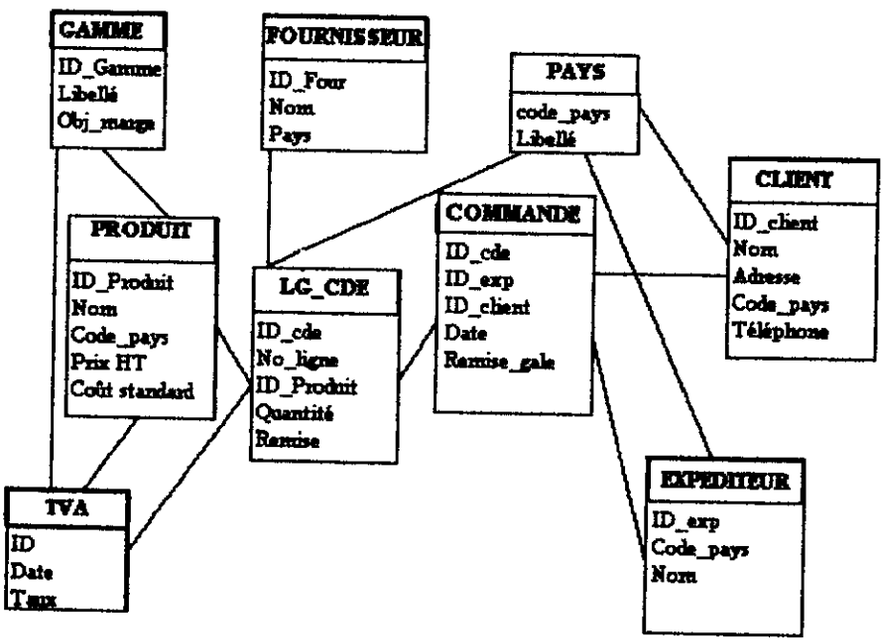


Fig.9. Modèle de Données Normalisé.

Le schéma présenté ici est simple. Pourtant, il est constitué de 9 entités. Un modèle d'entreprise pourra contenir des centaines ou des milliers d'entités, et donc d'autant de tables au niveau physique. Une requête lancé pour corréler des informations, par exemple sur les dépenses et les ventes ou les stocks et les unités vendues, utilisera peut être des dizaines de tables. Ce type de requête sera complexe à formuler pour l'utilisateur et à traiter pour l'optimiseur de la base de données. Ce qui provoque la saturation du système. D'où la nécessite de recourir à la dénormalistion pour le décisionnel.

2.2 Modèle de Donnée Dénormalisé

On prend le modèle normalisé et le simplifier afin qu'il réponde mieux aux exigences du décisionnel. Cette dénormalisation n'est pas réalisée en s'appuyant sur une technique précise mais plutôt en s'intéressant aux besoins des utilisateurs. On va ainsi créer des redondances d'informations et des informations agrégées qui diminueront le nombre de tables. De ce fait, on améliorera les temps de réponse et on facilitera l'accès aux informations par les utilisateurs car le modèle lui apparaîtra comme plus lisible. **[Fro00]**

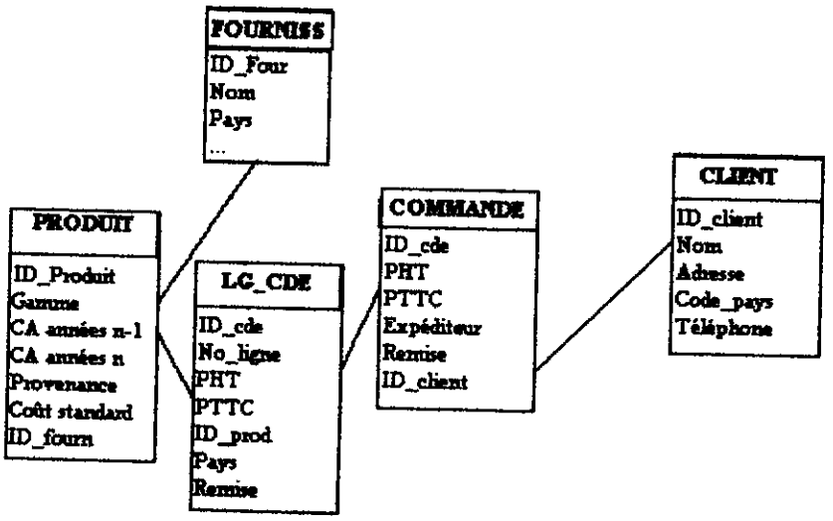


Fig. 10. Modèle de données Dénormalisé.

Une analyse précise des besoins des utilisateurs à montrer que **l'expéditeur** n'était pas un sujet représentant un intérêt majeur. On revanche il, est intéressant d'associer l'attribut « nom de l'expéditeur » au sujet commande ainsi le modèle contient un nombre de tables plus restreint, chaque table étant associe à un sujet d'intérêt.

Ce modèle est nettement moins complexe que le modèle normalisé. Cela ne veut pas dire pour autant que le modèle résultat est simple, car si nous avons un modèle normalisé de 200 tables nous aboutissons a une centaines de tables ce qui reste complexe est peu lisible. **[Fro00]**

3. Modélisation multidimensionnelle

La conception des bases de données est en général basée sur le modèle Entite-Relation (Entity-Relationship).

Ce modèle permet de décrire des relations entre les données élémentaires (entités) en éliminant des redondances, ce qui provoque l'introduction d'un nombre important de nouvelles entités. De ce fait, l'accès aux données devient compliqué et le diagramme généré difficile à comprendre pour un utilisateur. C'est pour cette raison que l'utilisation de la modélisation E-R pour la conception d'un data warehouse n'est pas considéré comme approprié.

Le modèle multidimensionnel de données, par contre, permet d'observer les données sous plusieurs perspectives. Il est plus facilement compréhensible, même pour les personnes qui ne sont pas expertes en informatique.

La modélisation multidimensionnelle vise à présenter les données sous forme standardisée, intuitive et facilitant l'interrogation. Selon cette approche, le sujet à analyser est placé au centre du modèle et les axes d'analyses pertinent autour.

> La modélisation multidimensionnelle consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé (chiffre d'affaire) et les différentes perspectives de l'analyse (produit, région, temps).

Exemple : Considérons les données suivantes.

Ventes en 1999

Catégorie des produits	**Région**	**Montant des ventes**
Electroménager	Midi-Pyrénées	50
Electroménager	Aquitaine	40
Electroménager	Languedoc-Roussillon	30
Papeterie	Midi-Pyrénées	60
Papeterie	Languedoc-Roussillon	50
Bricolage	Midi-Pyrénées	30
Bricolage	Aquitaine	30

Tab.2 .Ventes de produits par région.

On peut distinguer différentes perspectives pour observer ces données : une dimension relative à la catégorie des produits, une dimension relative à la région.

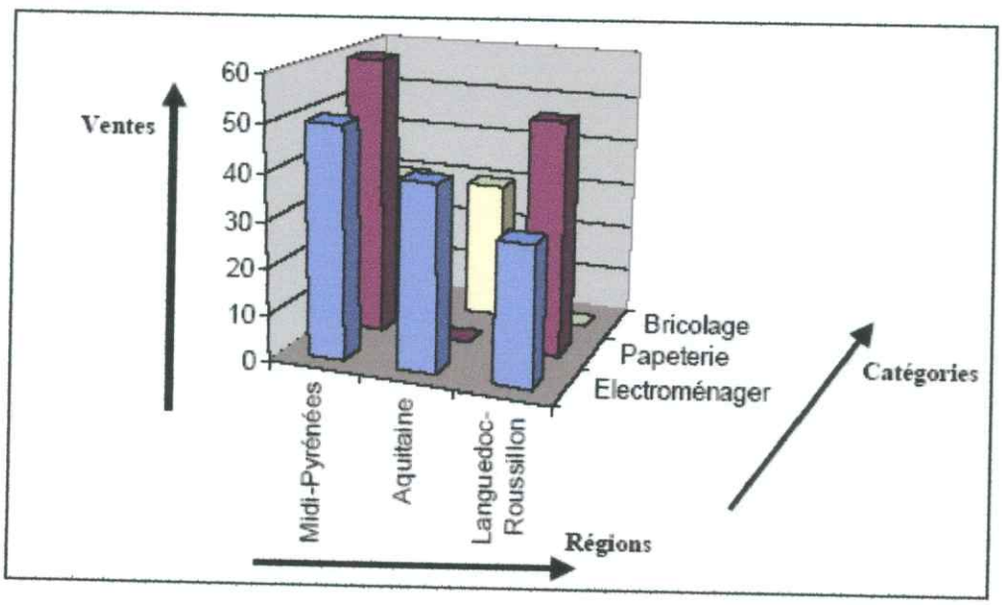


Fig. 11. Visualisation sous forme de barres.

Maintenant, nous considérons plusieurs tables, relatives aux ventes de chaque année entre 1997 et 1999. On peut alors observer les données dans un espace à trois dimensions : la dimension catégorie, la dimension produits et la dimension temps. Chaque intersection de ces dimensions représente une cellule comportant le montant des ventes.

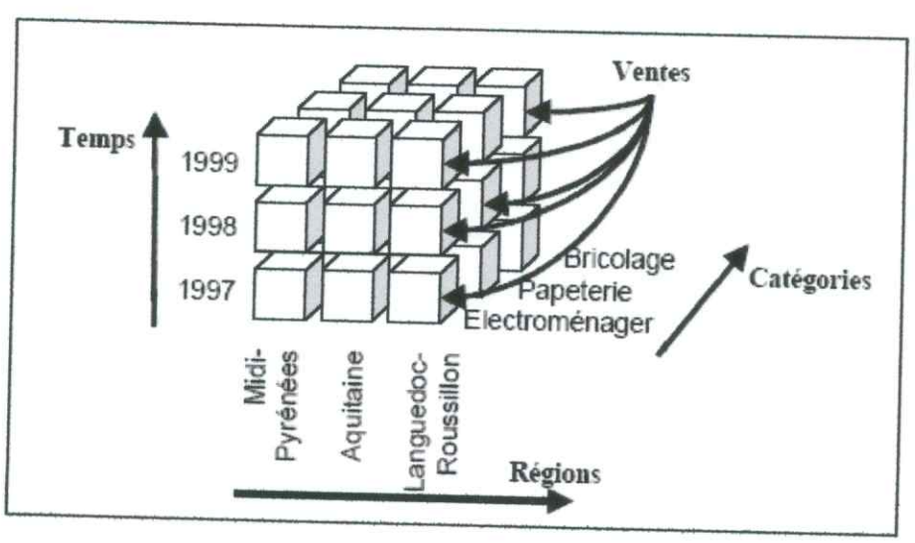


Fig. 12. Représentation multidimensionnelle.

3.1. Modélisation conceptuelle

Conceptuellement, cette modélisation multidimensionnelle a donné naissance aux concepts de fait et de dimension **[Kim96]. [Tes00]**

3.1.1 Concept fait

Le fait modélise le sujet de l'analyse. Un fait est formé de mesures correspondant aux informations de l'activité analysée. Une mesure est numérique, comme le chiffre d'affaires par exemple, souvent additive, apte à être manipulée via des opérateurs arithmétiques.

Exemple : Considérons le fait de Vente pouvant être constitué des mesures d'activités suivantes : quantité de produits vendus et montant total des ventes. Nous représenterons le fait par un rectangle englobant les différentes mesures d'activité qu'il contient. En outre le symbole d'un cube estampille le fait.

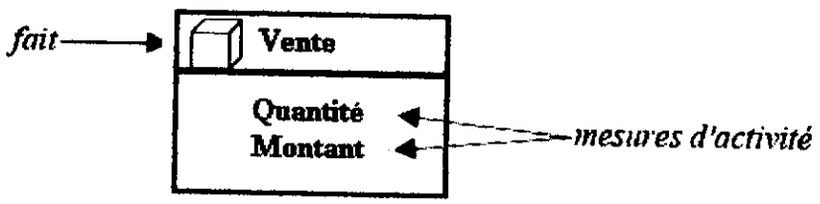


Fig. 13. Exemple de fait.

3.1.2 Concept dimension

Le sujet analysé, c'est à dire le fait, est analysé suivant différentes perspectives. Ces perspectives correspondent à une catégorie utilisée pour caractériser les mesures d'activité analysées. On parle de dimensions.

Une dimension représente une notion liée au métier, comme le client ou le fournisseur. Le sujet à analyser, en d'autres termes le fait, est analysé selon différents axes appelés dimensions. Une dimension représente donc une perspective d'analyse. Elle se compose de paramètres (ou attributs) correspondants aux informations pour lesquelles sont analysées les mesures.

Exemple: Poursuivons l'exemple précédent. Le fait peut être analysé suivant différentes perspectives correspondant à trois dimensions : la dimension Temps, la dimension Géographie et la dimension Catégorie.

Nous représenterons une dimension par un rectangle englobant les différents paramètres qu'elle contient. En outre un symbole représentant trois axes estampille les dimensions pour les distinguer du fait.

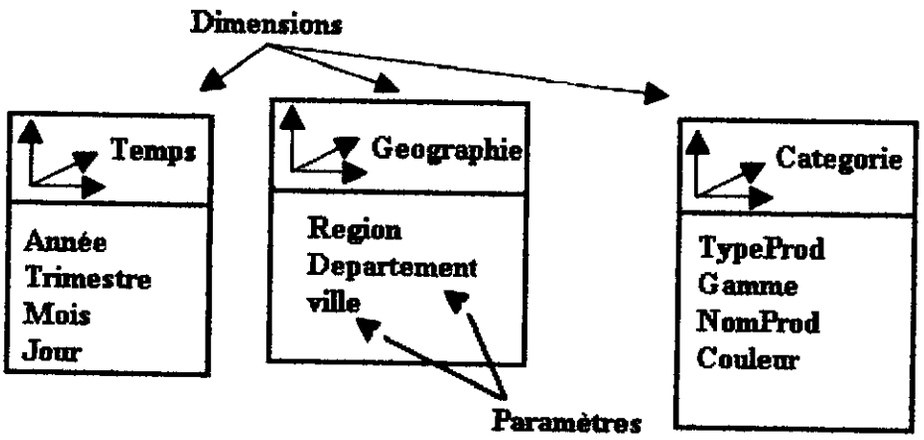


Fig.14. Exemples de dimensions.

a) Concept hiérarchie

Les paramètres d'une dimension peuvent être organisés selon leur niveau de détail. Pour définir ces différents niveaux, chaque dimension est munie d'une ou plusieurs hiérarchies.

> Une hiérarchie organise les paramètres d'une dimension selon une relation :
> « Est plus fin que » conformément à leur niveau de détail.

Exemple

La dimension Géographie peut se composer des paramètres Ville, Département, Région et Pays, présentés du niveau de détail le plus fin au niveau de détail le plus agrégé (une ville fait partie d'un département qui fait partie d'une région, ...).

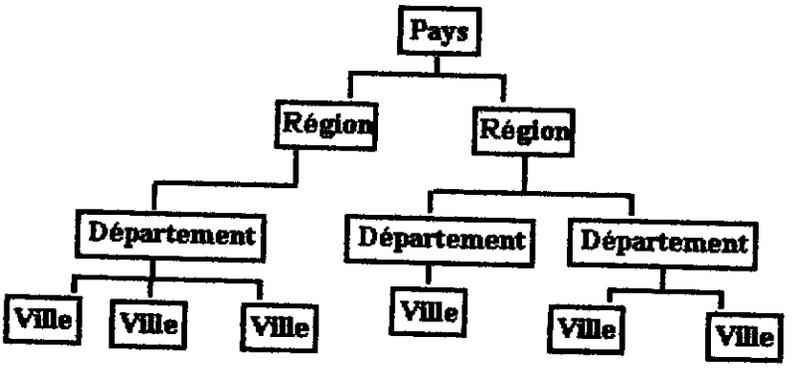


Fig.15. Exemple de hiérarchie.

3.1.3. Modèles en étoile, en flocon et en constellation

a) Modèle en étoile

A partir du fait et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Cette structure est constituée du fait central et des dimensions. Ce modèle représente visuellement une étoile, on parle de modèle en étoile (star schema).

Exemple : un schéma en étoile modélisant les analyses des quantités et des montants selon trois dimensions : Le temps, la catégorie et la situation géographique.

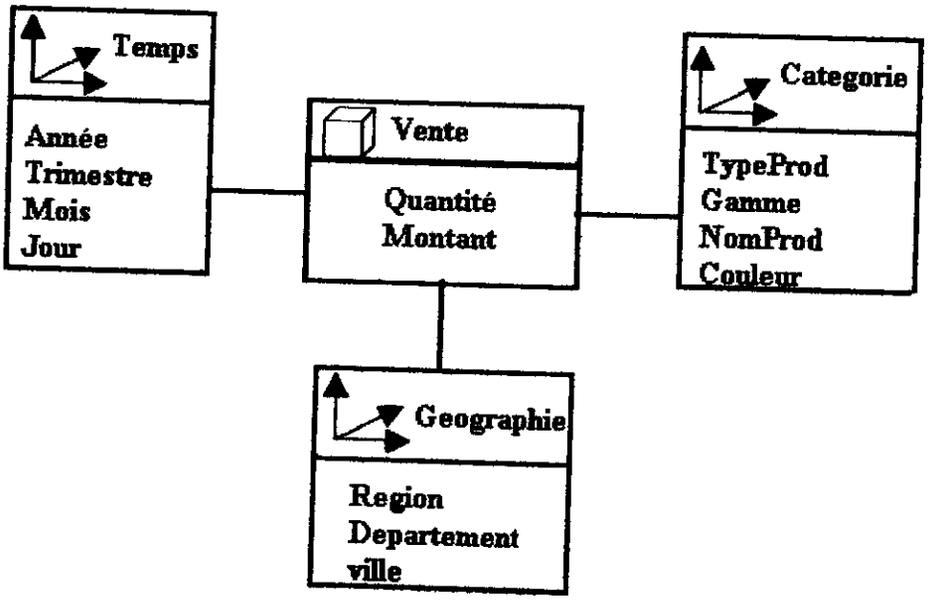


Fig.16. Schéma en étoile.

Avantages :

- Facilité de navigation
- Performances : nombre de jointures limitées.

Inconvénients :

- Toutes les dimensions ne concernent pas les mesures
- Redondances dans les dimensions.

b) Modèle en flocon de neige

Il existe d'autres techniques de modélisation multidimensionnelle, notamment la modélisation en flocon (snowflake). Une modélisation en flocon consiste à décomposer les dimensions du modèle en étoile en sous hiérarchies. La modélisation en flocon est donc une émanation de la modélisation en étoile, le fait est conservé et les dimensions sont éclatées conformément à sa hiérarchie des paramètres. L'avantage de cette modélisation est de formaliser une hiérarchie au sein d'une dimension. Par contre, la modélisation en flocon induit une dénormalisation des dimensions générant une plus grande complexité en termes de lisibilité et de gestion.

EXEMPLE : nous décrivons le modèle en étoile précédant en dénormalisant chacune de ces dimensions, formant ainsi une sorte de flocon.

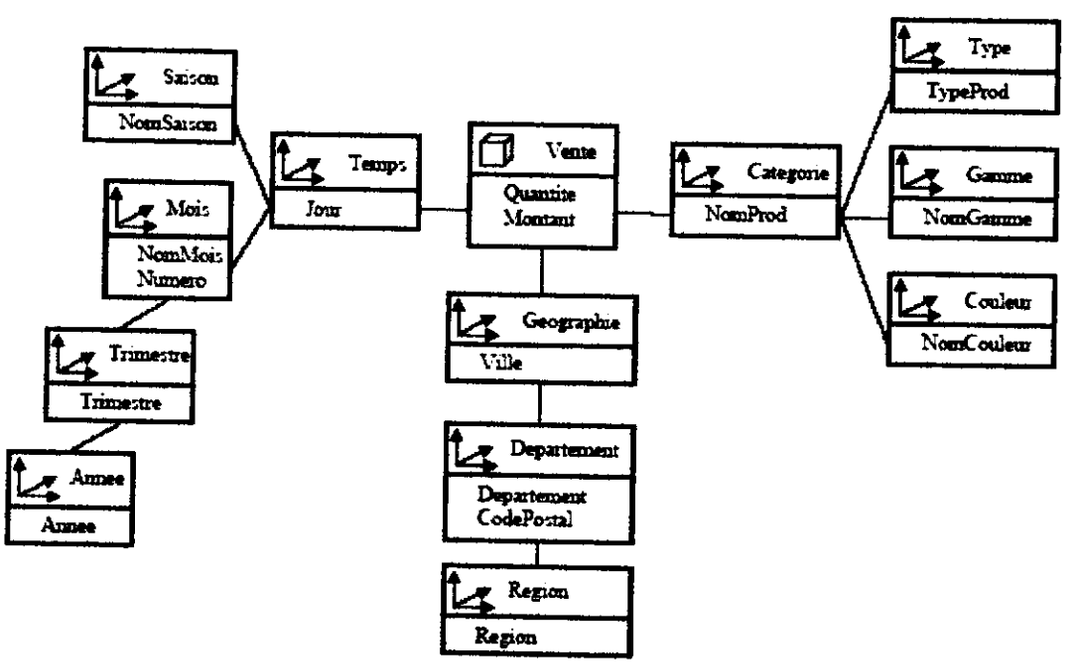


Fig.17. Schéma en flocon de neige.

Avantages :

- Réduction du volume,
- permettre des analyses par palier (drill down) sur la dimension hiérarchisée.

Inconvénients :

- Navigation difficile.
- Nombreuses jointures.

c) Modèle en constellation

Une autre technique de modélisation, issue du modèle en étoile, est la modélisation en constellation. Il s'agit de fusionner plusieurs modèles en étoile qui utilisent des dimensions communes. Un modèle en constellation comprend donc plusieurs faits et des dimensions communes ou non.

Exemple : la modélisation en constellation ; nous décrivons une constellation constituée de deux schémas en étoile : l'un correspond aux ventes effectuées dans les pharmacies et l'autre analyse les prescriptions des médecins.

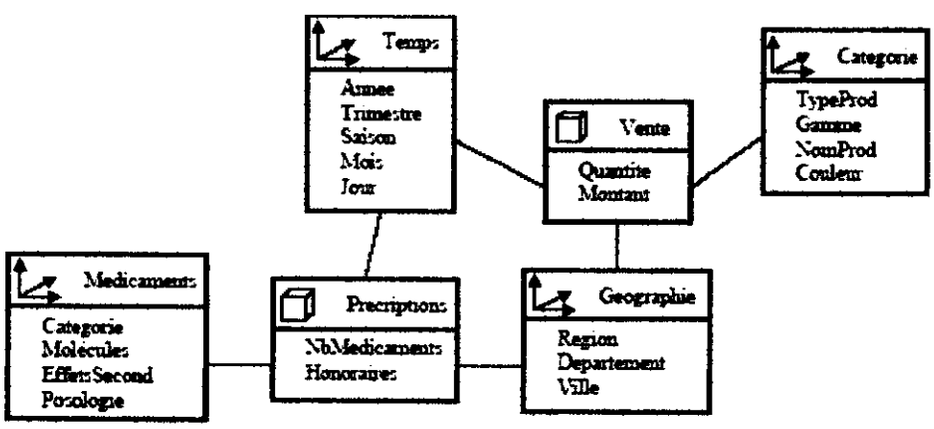


Fig.18. Schéma en constellation.

3.2. Modélisation logique

Au niveau logique plusieurs possibilités sont envisageables pour la modélisation multidimensionnelle. Il est possible d'utiliser :

- Un système de gestion de bases de données (SGBD) existant tel que les SGBD relationnels (ROLAP) ou bien les SGBD orientées objet (OOLAP),
- Un système de gestion de bases de données multidimensionnelles (MOLAP).

3.2.1. ROLAP

L'approche la plus couramment utilisée consiste à utiliser un système de gestion de bases de données relationnelles, on parle de l'approche ROLAP ("Relational On-Line Analytical Processing").

Le modèle muldimensionnel est alors traduit de la manière suivante :

- Chaque fait correspond à une table, appelée table de fait,
- Chaque dimension correspond à une table, appelée table de dimension.

Ainsi, la table de fait est constituée d'attributs représentant les mesures d'activité et les attributs clés étrangères de chacune des tables de dimension. Les tables de dimension contiennent les paramètres et une clé primaire permettant de réaliser des jointures avec la table de fait.

Exemple : Nous considérons l'exemple du schéma en étoile précédant.

La table VENTE correspond au fait et les tables TEMPS, GEOGRAPHIE, CATEGORIE représentent les dimensions.

VENTE (CleTps#, CleGeo#, CleCat#, Quantite, Montant)

TEMPS (CleTps, Annee, Trimestre, Saison, Mois, Jour)

GEOGRAPHIE (CleGeo, Region, Departement, Ville)

CATEGORIE (CleCat, TypeProd, Gamme, NomProd, Couleur)

3.2.2. OOLAP

Plus récemment, une autre approche s'appuie sur le paradigme objet, on parle de l'approche OOLAP ("Object On-Line Analytical Processing"). Le modèle multidimensionnel se traduit ainsi :

- Chaque fait correspond à une classe, appelée **classe de fait**,
- Chaque dimension correspond à une classe, appelée **classe de dimension**.

3.2.3. MOLAP Une alternative à ces deux approches consiste à utiliser un système multidimensionnel "pur" qui gère des structures multidimensionnelles natives, on parle de l'approche MOLAP ("Multidimensional On-Line Analytical Processing").

Les structures multidimensionnelles natives utilisées sont des tableaux à n dimensions. Dans la littérature, les termes de cube, hypercube et table multidimensionnelle sont utilisés de manière interchangeable. Nous utiliserons le terme d'hypercube pour désigner des structures à deux, trois ou à plus de trois dimensions.

Cette approche permet de stocker les données de manière multidimensionnelle. L'intérêt est que les temps d'accès sont optimisés, mais cette approche nécessite de redéfinir des opérations pour manipuler ces structures multidimensionnelles.

4. Modélisation Multidimensionnelle versus Modélisation Relationnelle

Il est intéressant de faire une comparaison entre la modélisation relationnelle (E-R), utilisée au sein des systèmes OLTP ainsi que dans la zone de préparation des données des systèmes décisionnels, et la modélisation multidimensionnelle, utilisée dans la zone de présentation des données des systèmes décisionnels.

Ce que l'on peut noter en premier lieu, c'est qu'en fait, un **schéma E-R peut se décomposer en plusieurs schémas multidimensionnels ([Kim00])**. En d'autres termes, un schéma E-R est composé de plusieurs étoiles. Pour décomposer un schéma E-R en multidimensionnel, on va sélectionner dans le schéma E-R les associations contenants des attributs numériques impliquées dans des relations plusieurs à plusieurs (N:M) entre entités. On fera de ces associations des tables de faits. On dénormalisera ensuite les autres entités et on les reliera aux tables de faits identifiées. On obtient ainsi les dimensions.

En résumé, les principales différences entre les deux approches sont les suivantes :

Tab.3. Modélisation multidimensionnelle versus modélisation relationnelle.

Modélisation E-R	**Modélisation multidimensionnelle**
Elimine les redondances dans le stockage des informations	Facilite l'accès à l'information
Modèle informatique difficile à appréhender par l'utilisateur final (un même schéma représente souvent différents processus métier)	Modèle proche du métier
	Assure une performance d'accès en interrogation

5. Conclusion :

La modélisation des données est un point fondamental dans la construction du data warehouse. Un modèle mal adapté aux besoins des utilisateurs deviendra en effet assez vite inexploitable. La construction du modèle est d'autant plus critique qu'aujourd'hui aucune méthode miracle n'existe et que peu d'outils sont a même d'aider le concepteur dans sa tâche.

Le choix à opérer entre modèle dimensionnel (étoile ou flocon) et modèle relationnel (normalisé ou denormalisé) dépend du type de l'initiative décisionnel à construire.

CHAPITRE IV :
La Technologie OLAP

1. Qu'est ce que OLAP ? :

Le terme OLAP (On Line Analytical Processing) ou traitement analytique en ligne se réfère à une technologie qui facilite l'accès à l'information stockée dans des data wharhouses. Les applications OLAP permettent à l'utilisateur d'extraire des données et de les visualiser sous plusieurs angles (chiffre d'affaire par produit, par client, par région).

Les caractéristiques demandées à une base de données OLAP ont été formalisées par Nigel Pendse et Richard Creeth, les auteurs de l'Olap report (www.olapreport.com) sous le sigle FASMI, pour Fast Analysis of Shared Multidimensional Information :

- **Fast (Rapidité) :** le système doit être capable de répondre aux demandes des utilisateurs dans un laps de temps courts (entre 1 et 20 secondes). Les constructeurs utilisent des astuces comme des précalculs pour réduire les durées des requêtes. Il est donc préférable de choisir une réponse rapide même au détriment d'une analyse moins poussée.
- **Analysis (Analyse) :** le système doit pouvoir faire face à toutes les logiques d'affaire et de statistiques. L'utilisateur doit avoir la possibilité de construire ses propres calculs et ses analyses sans avoir à programmer. Pour cela, les outils utilisés seront fournis par le constructeur ou bien seront des outils extérieurs.
- **Shared (Partage) :** le système doit créer un contexte où la confidentialité est préservée et doit gérer les cas où plusieurs utilisateurs ont des droits en écritures. Ce point est la plus grosse faiblesse des produits actuels.
- **Multidimensional (Multidimensionnel) :** C'est la condition essentielle des produits OLAP. Le système doit fournir des vues conceptuelles multidimensionnelles des données. Les hiérarchies doivent être respectées et doivent être visibles.

- **Informations (Information)** : C'est toutes les données et les informations nécessaires pour un produit OLAP. La capacité d'un produit sera en fonction des données en entrées et non pas en place mémoire qu'elle occupe.

2. Les 12 Règles D'OLAP

Afin de formaliser le concept OLAP, vers fin 1993, Edgar.F.Codd publie un article intitulé *"Providing OLAP to User Analysts"* aux Etats Unis, dans lequel il définit 12 règles que tout système de pilotage multidimensionnel devrait respecter.

1. Vue multidimensionnelle

Le système doit permettre une vue multidimensionnelle des données.

L'utilisateur a l'habitude de raisonner en vue multidimensionnelle comme par exemple lorsqu'il souhaite analyser les ventes par produit mais aussi par région ou par période. Ces modèles permettent des manipulations simples : rotation, pivot ou vues par tranche, analyse de type permutations d'axes (slice-dice) ou en cascade (drill-anywhere).

2. Transparence du serveur OLAP à différents types de logiciels

Cette transparence se traduit pour l'utilisateur par un complément à ses outils habituels garantissant ainsi sa productivité et sa compétence. Elle s'appuie sur une architecture ouverte permettant à l'utilisateur d'implanter le système OLAP sans affecter les fonctionnalités du système central.

3. Accessibilité à de nombreuses sources de données

Le système OLAP doit donner accès aux données nécessaires aux analyses demandées. Les outils OLAP doivent avoir leur propre schéma logique de stockage des données physiques hétérogènes, doivent accéder aux données et réaliser n'importe quelle conversion afin de présenter à l'utilisateur une vue simple et cohérente. Ils doivent aussi savoir de quel type de systèmes proviennent les données.

4. Performance du système de Reporting

L'augmentation du nombre de dimensions ou du volume de la base de données ne doit pas entraîner de dégradation visible par l'utilisateur.

5. Architecture Client/Serveur

La plupart des données pour OLAP sont stockées sur des gros systèmes et sont accessibles via des PC. Il est donc nécessaire que les produits OLAP soient capables de travailler dans un environnement Client/Serveur.

6. Dimensions Génériques

Toutes les dimensions doivent être équivalentes en structure et en calcul. Il ne doit exister qu'une seule structure logique pour toutes les dimensions. Toute fonction qui s'applique à une dimension doit être aussi capable de s'appliquer à une autre dimension.

7. Gestion dynamique des matrices creuses

Le schéma physique des outils OLAP doit s'adapter entièrement au modèle d'analyse spécifique créé pour optimiser la gestion des matrices creuses. En effet, dans une analyse à la fois sur les produits et les régions, tous les produits ne sont pas vendus dans toutes les régions.

Quand on construit le cube multidimensionnel, le produit cartésien des divers axes réserve les espaces nécessaires au stockage des informations même si celle-ci n'existent pas, générant ainsi se que l'on appelle la matrice creuse. Les parties du cube multidimensionnel doivent être stockées afin de ne pas altérer les temps d'accès.

8. Support multi-utilisateurs

Les outils OLAP doivent supporter les accès concurrents, garantir l'intégrité et la sécurité afin que plusieurs utilisateurs accèdent au même modèle d'analyse.

9. Calculs à travers les dimensions

Les opérations doivent pouvoir s'effectuer sur toutes les dimensions et ne doivent pas faire intervenir l'utilisateur pour définir un calcul hiérarchique.

10. Manipulation intuitive des données

Toute manipulation doit être accomplie via une action directe sur les cellules du modèle sans utiliser de menus ou des chemins multiples à travers l'interface utilisateur.

11. Nombre illimité de niveaux d'agrégation et de dimensions

Il ne doit pas y avoir de limites particulières quand aux dimensions et aux niveaux d'agrégation requis.

12. Souplesse et facilité de constitution des rapports

La création des rapports dans les outils OLAP doit permettre aux utilisateurs de présenter comme ils le désirent des données synthétiques ou des résultats en fonction de l'orientation du modèle.

Six nouvelles règles ont été ajoutées pour compléter la définition :

1) Extraction en lots et interprétation
2) Modèle d'analyse OLAP
3) Traitement des données non formalisées

4) Conserver les résultats OLAP

5) Extraction des valeurs manquantes

6) Traitement des valeurs manquantes

3. Systèmes OLTP versus systèmes OLAP

Les bases de données sont utilisées dans les entreprises pour gérer les importants volumes d'informations contenus dans leurs systèmes opérationnels. Ces données sont gérées selon des processus transactionnels en ligne (OLTP : "*On-Line Transactional Processing*"). qui se caractérisent de la manière suivante **[Cod93] [Kim96] :**

- Ils sont nombreux au sein d'une entreprise.
- Ils concernent essentiellement la mise à jour des données.
- Ils traitent un nombre d'enregistrements réduit.
- Ils sont définis et exécutés par de nombreux utilisateurs.

L'exploitation de l'information contenue dans ces systèmes opérationnels est devenue une préoccupation essentielle pour les dirigeants des entreprises qui désirent améliorer leur prise de décision par une meilleure connaissance de leur propre activité, de celle de la concurrence, des employés, des clients et des fournisseurs. Les entreprises sont donc à la recherche de systèmes supportant efficacement les applications d'aide à la décision. Ces applications décisionnelles utilisent des processus d'analyse en ligne de données (OLAP : "*On-Line Analytical Processing*" **[Cod93]**). Ces processus répondent aux besoins spécifiques des analyses d'information.

Ces processus OLAP se caractérisent de la manière suivante **[Cod93][Inm94][Kim96] :**

- ils sont peu nombreux, mais leurs données et traitements sont complexes.
- il s'agit uniquement de traitements semi-automatiques visant à interroger, visualiser et synthétiser les données.
- ils concernent un nombre d'enregistrements importants aux structures hétérogènes.
- ils sont définis et mis en œuvre par un nombre réduit d'utilisateurs qui sont les décideurs.

Le Tableau suivant compare les caractéristiques des processus OLTP et OLAP.

Caractéristique du système	OLTP	OLAP
Portée de l'interaction utilisateur	Transaction	Base de données entière
Application	production	Aide à la décision
Quantité de données affectées par l'interaction	Enregistrements individuels	Groupes d'enregistrements
Temps de réponse	Seconde	Seconde à Minute
Mode d'utilisation machine	Stable	Dynamique
Nature des données	Normalisées, non agrégées	Dénormalisées, agrégées
Taille données	100MB à 1GB	1GB à 1TB
Mode d'accès à la base de données	Accès prédéfini	Accès indéfinis ou Dynamique
Volatilité des données	Elevée (données mise à jour à chaque transaction)	Faible (données rarement mises à jour pendant une requête)
Priorité	Haute performance, grande disponibilité	Grande souplesse, grande autonomie de l'utilisateur

Tab.4. Comparaison des Processus OLTP et OLAP.

Beaucoup de produit utilisent des bases de données multidimensionnelles tandis que d'autres font appel à des bases de données relationnelles associées à un schéma en étoile ou toutes les données sont stockées dans une « table de Fait » avec autour plusieurs autres tables de dimension.

4. Les différents outils OLAP

Les moteurs du décisionnel, les outils OLAP (On Line Analytical Processing), complètent les Data Warehouses. Ils fournissent une vue des données stockées qui transforme celles-ci en informations stratégiques pour l'entreprise. Le stockage des informations peut être réalisé de plusieurs façons : MOLAP (Multidimensionnel OLAP), ROLAP (Relationnel OLAP), HOLAP (Hybride OLAP).

4.1. Les outils MOLAP (Multidimensionnel OLAP)

La base MOLAP est l'application physique du concept OLAP elle adopte réellement une structure multidimensionnelle. MOLAP est conçue exclusivement pour l'analyse multidimensionnelle, avec un mode de stockage optimisé par rapport aux chemins d'accès prédéfinis. Ainsi, toute valeur d'indicateur associée à l'axe temps sera précalculée au chargement pour toutes ses valeurs hebdomadaires, mensuelles, etc.

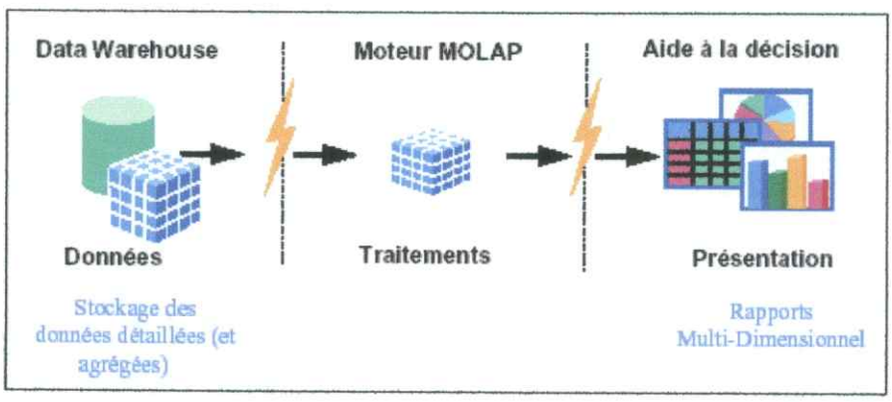


Fig.19. Architecture d'un produit Molap. [W.01]

MOLAP agrége tous par défaut. Cette base est plus rapide et performante en temps d'accès.
Pourtant, ce type de stockage doit se limiter aux données les plus utilisées (celle de l'année en cours, par exemple) afin que les réponses soient instantanées et que le volume de données MOLAP reste raisonnable.

4.2. Les outils ROLAP (Relational OLAP)

ROLAP est une base relationnelle classique organisée pour réagir comme une base OLAP, cette base est sans limite de taille et permet ainsi de gérer de plus grandes bases. Mais ceci se fait au détriment de l'accès aux informations, qui est alors plus lent car les résultats des requêtes sont calculés au moment de leur exécution.

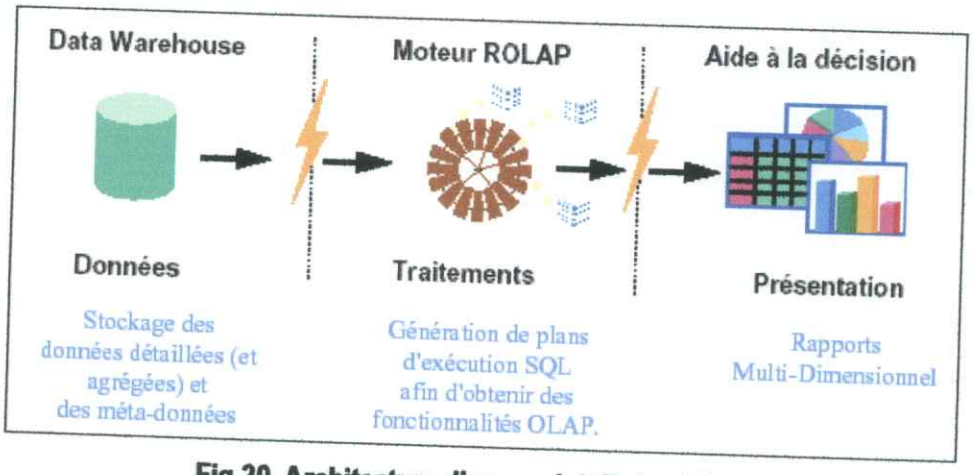


Fig.20. Architecture d'un produit Rolap. [W.01]

ROLAP n'agrège rien, mais tire parti des agrégats s'ils existent. De ce fait ROLAP est plus lourd à administrer que MOLAP, puisqu'il demande de créer explicitement certains agrégats.

Cependant, il est déconseillé d'accéder en direct à des bases de données de production pour faire des analyses sérieuses, pour des raisons de performances.

4.3. Les outils HOLAP (Hybrid OLAP)

La base HOLAP est un compromis entre les deux concepts précédents : une base MOLAP pour les données souvent consultées, une base ROLAP pour les autres données. La solution HOLAP combine les avantages des deux solutions MOLAP et ROLAP.

MOLAP	• les données de la base décisionnelle sont copiées dans une structure (hyper-cube) matricielle qui contient également les agrégats. • requêtes MDX • Gestion des matrices creuses (aucun espace aux cellules vides), ce qui limite la taille de stockage. • Limitation de stockage aux données les plus utilisées (celle de l'année en cours, par exemple) afin que les réponses soient instantanées. • Taille limitée au GigaOctet.

ROLAO	• Toutes les données restent dans la base décisionnelle. • Les requêtes MDX sur ce cube font donc appel à des requêtes SQL impliquant des jointures. • C'est le type de stockage qui offre les temps de réponse les plus lents. Mais c'est aussi le plus économe. Généralement, les données qui remontent à deux ans et plus (elles sont nombreuses et peu consultées).
HOLAP	• seuls les agrégats des niveaux supérieurs sont stockés sous forme matricielle et les données bas niveaux restent dans la base décisionnelle. Il s'agit d'une combinaison entre l'approche MOLAP et l'approche ROLAP. • Seule les requêtes MDX qui utilisent directement les données bas niveaux sont ralenties (drillthrough, par exemple). • Ce type de stockage convient bien aux données de l'année précédente

Tab.5. Tableau récapitulant les différents outils OLAP

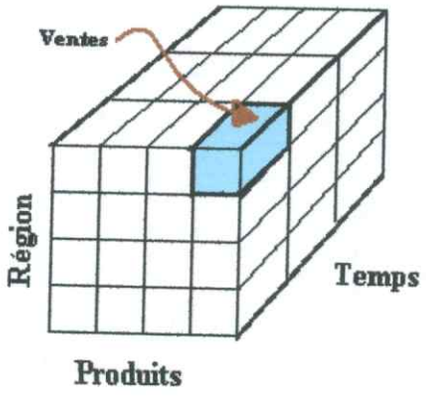
5. Cube :

L'analyse multidimensionnelle revient à voir les données sous forme cubique, une base de données relationnelle ne permet aux utilisateurs que des visions en deux dimensions par exemple l'étude des produits par région. Une base de données miltidimensionnelles permet une analyse intégrant plusieurs dimensions comme par exemple l'étude des produits par région dans le temps ou encore l'étude des ventes de produits par région par couleurs, par taille et ce dans le temps. **[Gog 01]**

Produit	Région	Vente
Ecrous	Est	50
Ecrous	Ouest	60
Ecrous	Centre	100
Vis	Est	40
Vis	Ouest	70
Vis	Centre	80
Boulons	Est	90
Boulons	Ouest	120
Boulons	Centre	140

Tab.6. Etude des ventes de produits par région

Fig.21. Etude des ventes de produits par région dans le temps



6. Les Opération de Base D'OLAP

Plusieurs opérations sont introduites pour offrir des possibilités d'animation dans la représentation du cube à l'écran. Elles consistent à faire pivoter le cube, le couper en tranches, inter changer ou combiner les coordonnées et/ou les contenus.

6.1 Les Techniques de Navigation dans le cube

6.1.1 Opérations liées à la structure

Les opérations agissant sur cette structure multidimensionnelle de l'information sont motivées par l'aspect interactif de l'analyse en ligne de données, et le souci d'offrir des possibilités d'animation de la représentation. De plus, elles illustrent l'importance des liens entre la manipulation des données et la représentation du cube à l'écran. Elles se présentent comme suit :

a) Rotate (pivot) : Cette opération consiste à faire effectuer à un cube une rotation autour d'un des trois axes passant par le centre de deux faces opposées, de manière à présenter un ensemble de faces différent.

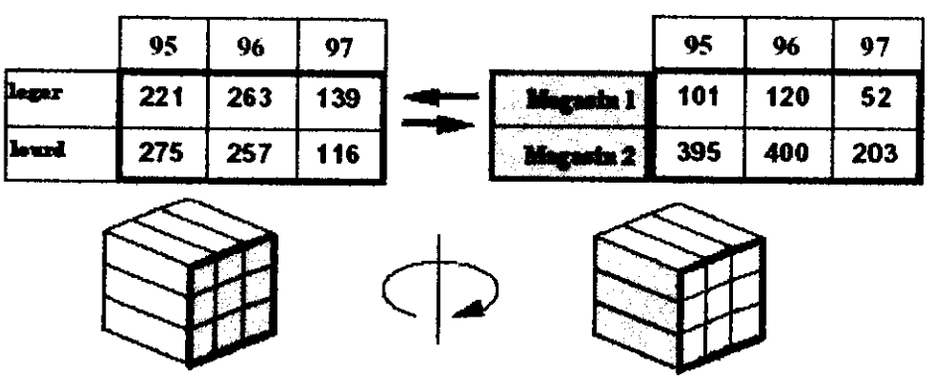


Fig.22. Exemple de rotation.

b) Switch (permutation) : Cette opération consiste à inter changer la position des membres d'une dimension.

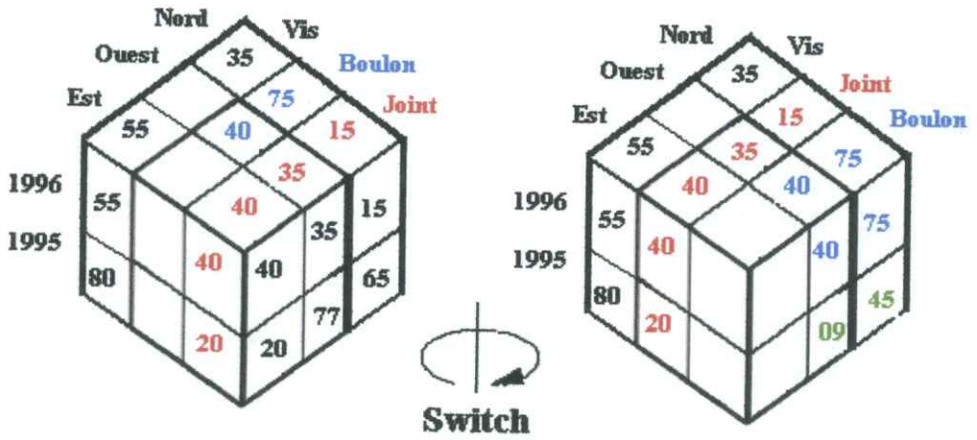


Fig.23. Exemple de permutation.

c) Split (division) : Elle consiste à présenter chaque tranche du cube, et à passer d'une représentation tridimensionnelle d'un cube à sa représentation sous la forme d'un ensemble de tables. D'une manière générale, cette opération permet de réduire le nombre de dimensions d'une représentation. On notera que le nombre de tables résultant d'une opération Split dépend des informations contenues dans le cube de départ et n'est pas connu à l'avance.

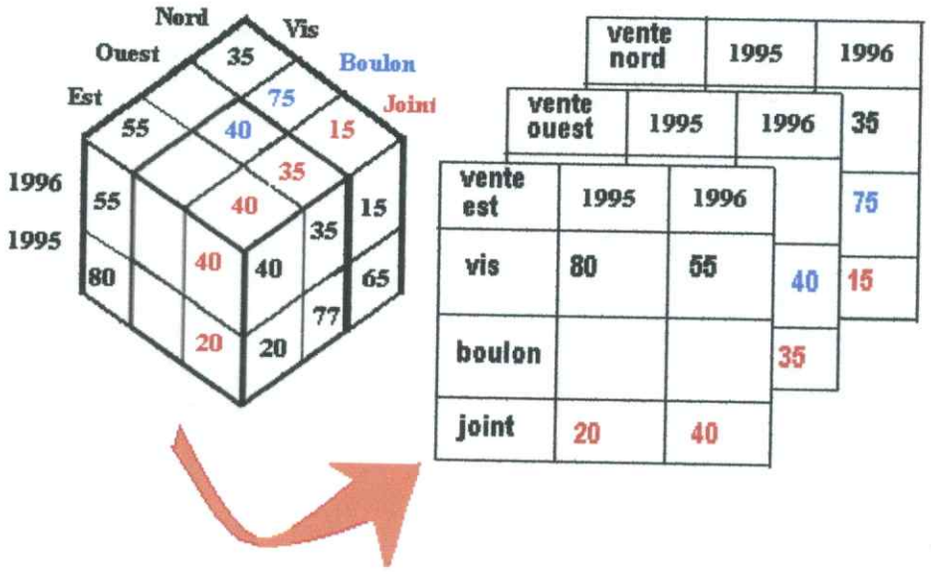
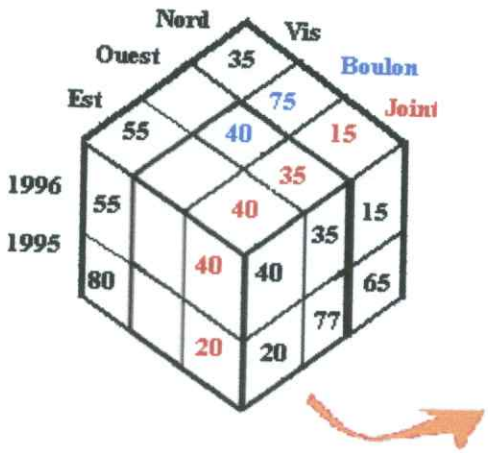


Fig.24. Exemple de division.

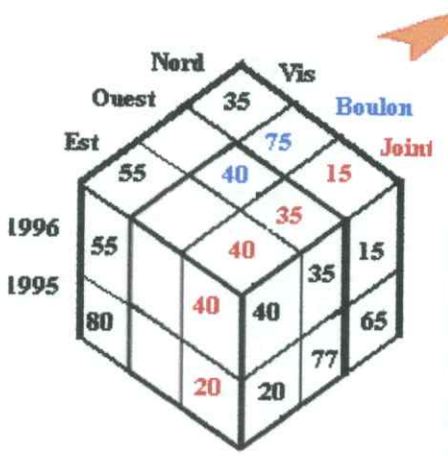
d) Nest (emboîtement) : Cette opération permet d'imbriquer des membres. L'un de ses intérêt est qu'elle permet dégrouper sur une même représentation bidimensionnelle toutes les informations (mesures et membres) d'un cube, quel que soit le nombre de ses dimensions. L'opération réciproque, « unnest », reconstitue une dimension séparée à partir des membres imbriques.



Ventes		1995	1996
Vis	Nord		35
	Est	80	55
	Ouest		45
Joint	Nord	65	15
	Est	20	40
	Ouest	77	35
Boulon	Nord	90	75
	Est		
	Ouest		40

Fig. 25. Exemple d'emboîtement.

e) Push (enfoncement) : Cette opération consiste à combiner les membres d'une dimension aux mesures du cube, et donc de faire passer des membres comme contenus de cellules. L'opération réciproque appelée pull, permet de changer le statut de certaines mesures d'un cube en membres, et de constituer une nouvelle dimension pour la représentation du cube, à partir de ces nouveaux membres.



Ventes	Est	Ouest	Nord
vis	1996 55 1995 80	1996 1995 10	1996 35 1995 10
boulon	1996 1995	1996 40 1995 25	1996 75 1995 40
joint	1996 40 1995 20	1996 35 1995 90	1996 15 1995 65

Fig. 26. Exemple de l'enfoncement.

6.1.2. Opérations associées à la granularité :

Les opérations agissant sur la granularité d'observation des données caractérisent la hiérarchie de navigation entre les différents niveaux. Elles correspondent aux opérations suivantes :

- **Drill Down & Roll Up :**

Le drill Down/UP c'est la possibilité d'aller du niveau global vers le niveau détaillé, et vice-versa. Ce mécanisme et basé sur la hiérarchie des différentes dimensions ; chaque dimension se décompose en attributs reliés entre eux par des relations père/fils. Cela n'est pas toujours facile car une dimension peut contenir plusieurs sous hiérarchies.

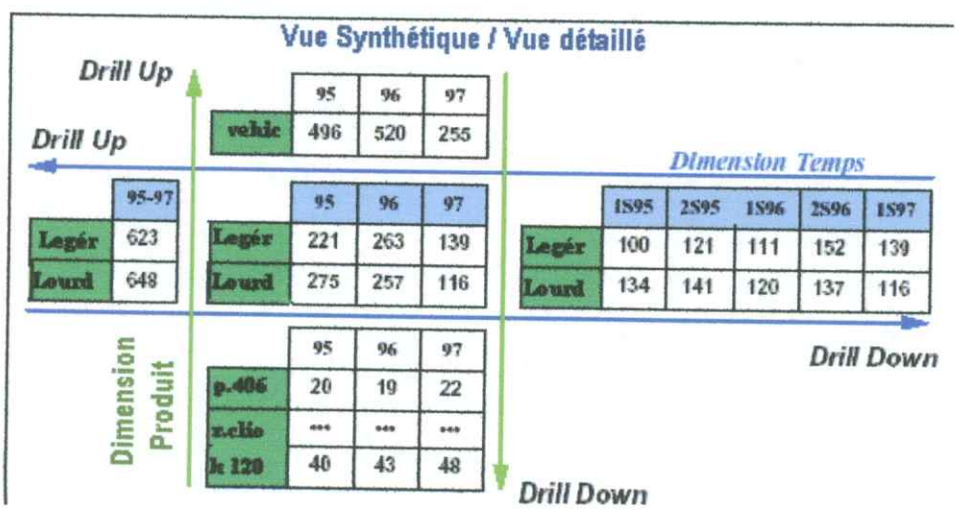


Fig.27 Illustration de Drill-Up/Drill-Down.

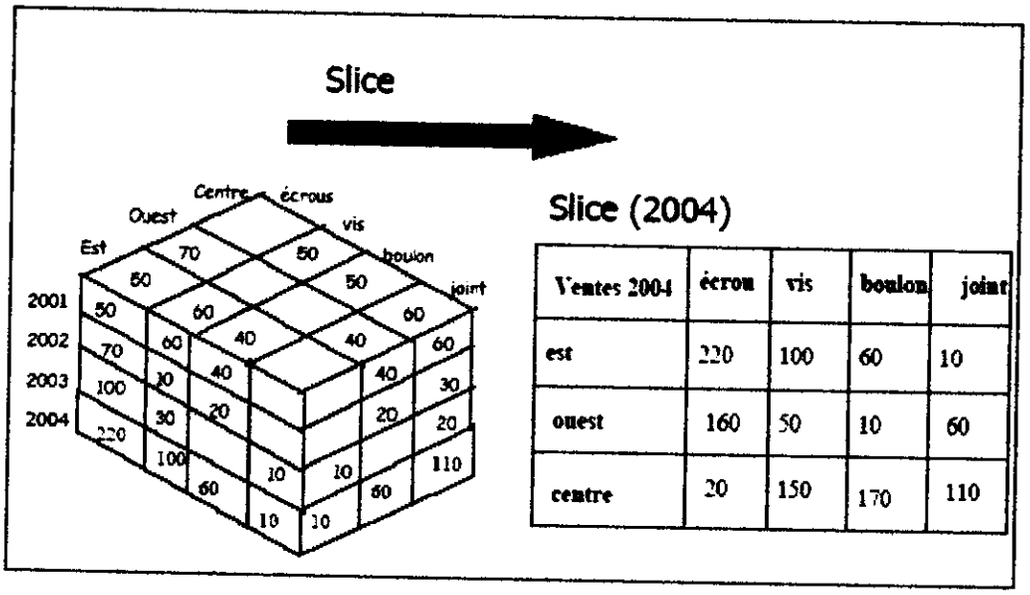
- **Slice & Dice :**

Le Slice and Dice désigne la possibilité de faire des coupes et de pivoter dynamiquement les cubes ou axes du tableau d'analyse croisé.

Slicing: Sélection de tranches du cube par des prédicats selon une dimension

- filtrer une dimension selon une valeur
- Exemple: Slice (2004) : on ne retient que la partie du cube qui correspond à cette date

Dicing: extraction d'un sous-cube



Ventes 2004	écrou	vis	boulon	joint
est	220	100	60	10
ouest	160	50	10	60
centre	20	150	170	110

Fig.28. Exemple de slicing

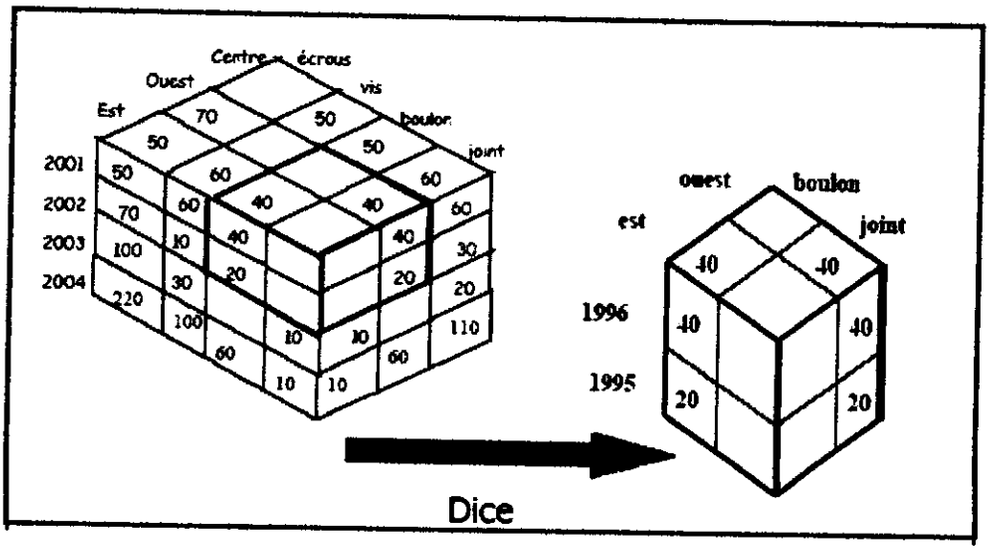


Fig.29. Exemple de Dicing

7. Conclusion :

Le traitement analytique en ligne ou OLAP (On Line Analytical Processing) est une technologie centrée sur l'analyse des données. Les applications OLAP permettent aux utilisateurs de sélectionner, visualiser et analyser des données transactionnelles à partir d'une variété de sources et elles donnent aux entreprises la possibilité d'extraire des informations supplémentaires de leurs systèmes OLTP (On Line Transaction Processing) et datawarehouse traditionnels. OLAP représente une extension, ou un affinement, des systèmes d'information d'aide a la décision ou SIAD et EIS (Executive Information System).ces types d'applications se destinent a la délivrance d'information de synthèse de haut niveau exploitable a un niveau décisionnaire. Olap va encore plus loin en fournissant d'autres possibilités d'analyse des données d'entreprise : par découpage, rotation, agrégation et exploration vers le bas (vers des niveaux plus détaillés).

Mais ce qui distingue la technologie OLAP d'un système OLTP traditionnel est sa capacité à fournir des vues multidimensionnelles de données transactionnelles.

CHAPITRE V :

Méthodologie de Construction d'un Data Warehouse

Introduction

Construire un entrepôt de données dimensionnel revient à faire correspondre les besoins de la communauté des utilisateurs avec la réalité des informations disponibles

L'entrepôt de données est donc bien différent des bases de données de production car les besoins pour lesquels on veut le construire sont différents. Il contient des informations historisées, globalement cohérentes, organisées selon les métiers de l'entreprise pour le processus de décision. L'entrepôt n'est pas un produit ou un logiciel mais un environnement. Il se bâtit et ne s'achète pas. Les données sont puisées dans les bases de production, nettoyées, normalisées, puis intégrées. Des métadonnées décrivent les informations dans cette nouvelle base pour lever toute ambiguïté quant à leur origine et leur signification. **[Kim97]. [AS04]**

Nous avons relevé trois parties interdépendantes qui relèvent de la construction d'un entrepôt de données :

1. L'étude préalable qui définit les besoins, objectifs et précise la démarche.
2. La conception du modèle dimensionnelle de donnée qui représente l'entrepôt conceptuellement et logiquement.
3. La mise en œuvre de l'architecture, par une suite de trois sous étapes :

    a. Construction de l'entrepôt, la base de l'entrepôt
    b. Construction des cubes OLAP, la base multidimensionnelle
    c. Construction de la zone d'alimentation, qui reprend à un niveau plus précis l'examen des données, le choix des méthodes et des dates auxquelles les données entreront dans l'entrepôt.

1. Etude et Définition des besoins

La définition des besoins métier et l'identification des données associées constituent l'étape fondamentale sinon crucial pour la construction d'un data warehouse.

Le recueil des besoins est une opération extrêmement subtile et probablement parmi les plus délicate pour un informaticien.

Les utilisateurs de l'entreprise et leurs besoins affectent presque toutes les décisions prisent au cours de l'implémentation du data warehouse .comme le montre la figure, nous plaçons ces besoins au centre de l'« univers du data warehouse ». D'où les chances de succès d'un data warehouse se trouvent considérablement accrues par la bonne compréhension des utilisateurs et de leurs besoins.

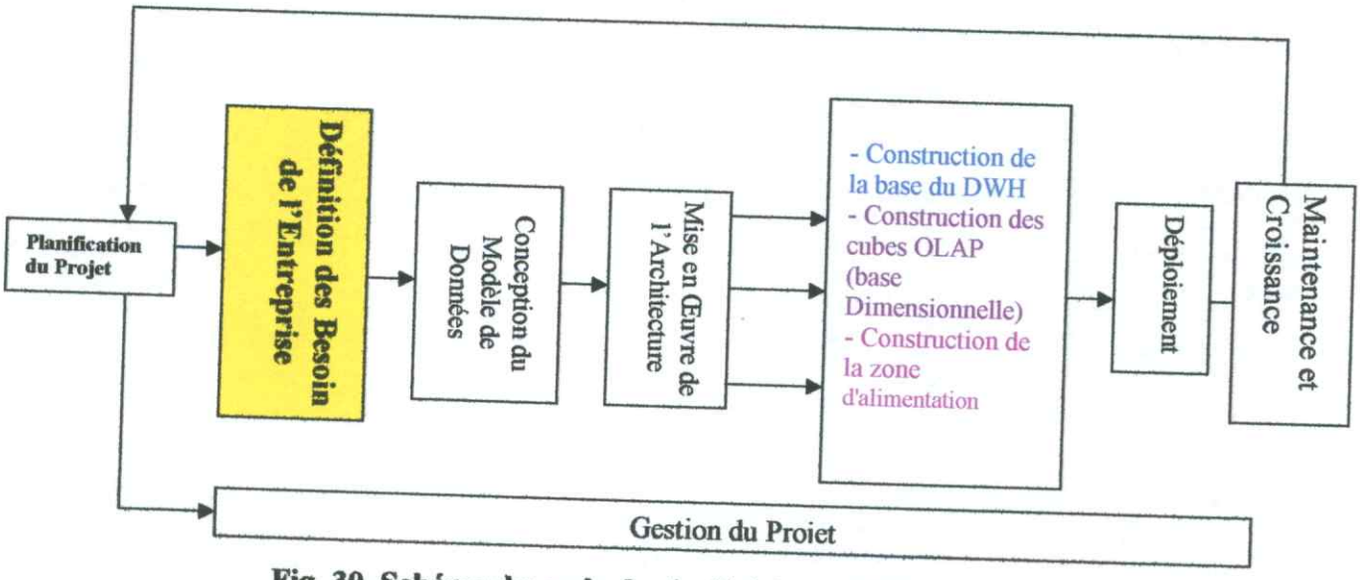


Fig. 30. Schéma du cycle de vie décisionnel [Kim97].

L'expression des besoins par les utilisateurs met souvent en évidence la volonté d'obtenir :

- Des analyses sur ce qui s'est passé (par exemple comparer les performances actuelles d'un magasin avec celles de l'année dernière)
- Des analyses prédictives (par exemple déterminer les achats potentiels pour un type de client, déterminer les clients qui risquent d'abandonner l'entreprise, ...).

Il faut alors recenser les données disponibles dans les bases de production, toutes les données de production ne sont pas utiles dans l'entrepôt donc il faut identifier les besoins des utilisateurs en terme de données utiles. On présente seulement les données utiles aux preneurs de décisions. Il faut aussi identifier les données supplémentaires requises et s'assurer la possibilité de se les procurer (achat de bases géographiques, démographiques...).

2. Conception du modèle de données

Ralph KIMBALL propose une démarche qui se résume à neuf décisions, dans un ordre bien définie, qui sont à la base pour la conception complète d'un entrepôt de données dimensionnel, ces décisions portent sur les point suivants **[Kim02]** :

2.1. Choisir les processus d'activité à modéliser :

La première étape de la conception consiste à décider quels processus d'entreprise doivent être modélisés, en confrontant notre compréhension des exigences de l'activité à la réalité des données disponibles au niveau des sources d'informations.

Un processus d'entreprise est une activité normale de l'organisation, généralement assistée par un système source collectant des informations au profit de l'entrepôt de données à partir d'une application existante. Des exemples de processus d'activité sont les achats de matières premières, les commandes, la facturation, les expéditions, les stocks, les comptes clients, les ventes et la comptabilité générale. Une fois les processus identifiés, une ou plusieurs tables de faits sont construites à partir de chacun des processus choisis.

2.2. Choisir le grain de chaque table de faits :

Déclarer le grain veut dire spécifier exactement ce que représente une ligne individuelle de table de faits. Le grain représente le niveau de détail des mesures de la table de faits. Il répond à la question « **comment décrivez-vous une ligne unique de la table de faits ?** ».

Généralement, la granularité de la table des faits choisie est la plus fine possible. Plus le niveau de détail est fin plus la conception est robuste, et lorsque les données sont détaillées, elles réagissent mieux à des requêtes inhabituelles et à l'introduction de données nouvelles.

Voici des exemples de déclarations de grain :

- Un instantané quotidien du niveau de stock de chaque produit dans un magasin.
- Un instantané mensuel de la situation de chaque compte d'une banque.
- Le total quotidien des ventes d'un produit dans un magasin.
- Transaction effectuée au guichet automatique de banque.
- Une ligne individuelle d'un article d'un bon de commande.

2.3. Choisir les dimensions de chaque table de faits :

Une fois que la granularité de la table des faits est bien établie, le choix des dimensions est assez simple. Bien souvent, la granularité est exprime en termes de dimensions. « **Le total quotidien des ventes d'un produit dans un magasin** » évoque inévitablement la dimension Temps, la dimension Produit, la dimension Magasin, voir la dimension lieu.

Le choix des dimensions s'accompagne de la définition de tous les attributs textuels (les champs) qui garniront la table de dimension.

2.4. Choisir les faits mesurés que contiendra chaque enregistrement de table de faits :

Des faits mesurés typique sont des quantités numériques additives telles que la quantité vendue ou les montants de vente. Identifier les faits numériques qui vont renseigner chaque ligne de la table de faits. Les faits sont déterminés par la réponse à la question, **« que mesurerons-nous ? ».**

2.5. Les attributs des dimensions :

Les attributs jouent un rôle vital dans un data warehouse. Ils sont la source de pratiquement toutes les contraintes et de tous les intitulés intéressants dans les états, il sont aussi le moyen par lequel le data warehouse est rendu utilisable et compréhensible.

Naturellement, la liste des attributs est généralement assez longue. Plus nous saisissons de données descriptives sur nos tables de dimension, plus celle-ci sera riche, et plus les analyses seront intéressantes.

A ce stade, la conception de la structure logique principale est terminée, les autres étapes concernent plus généralement la structure physique.

2.6. Comment suivre les dimensions à évolution lente :

Dans la réalité les dimensions ne sont pas indépendantes les une par rapport au autres. En particulier, comme exemple les dimensions « produit » et « client » ne sont pas indépendantes de la dimension temps; sur la durée, les descriptions et la formulation des produits réels subissent de lentes évolutions. Le client, notamment, change constamment. Les humains changent de nom, se marient et divorcent, ont davantage d'enfants et change d'adresse. Les équipes de vente changent périodiquement ...etc. Par exemple dans une compagnie d'assurances, il est essentiel que la description de la personne assurée reflète bien le statut de cette personne à l'époque d'un sinistre passé et non le statut actuel.

Nous appelons ces dimensions quasi constantes, des dimensions à évolution lente. Nous devons décider de quelle manière nous allons traiter ces changements, ce choix se résume entre trois solutions, chacune assurant un suivi plus ou moins strict des modifications dans le temps :

- **Ecrasement de la valeur précédente :** Recouvrir et perdre les valeurs anciennes dans l'enregistrement de la dimension, renonçant ainsi à la possibilité de suivre les événements ou situations passées.

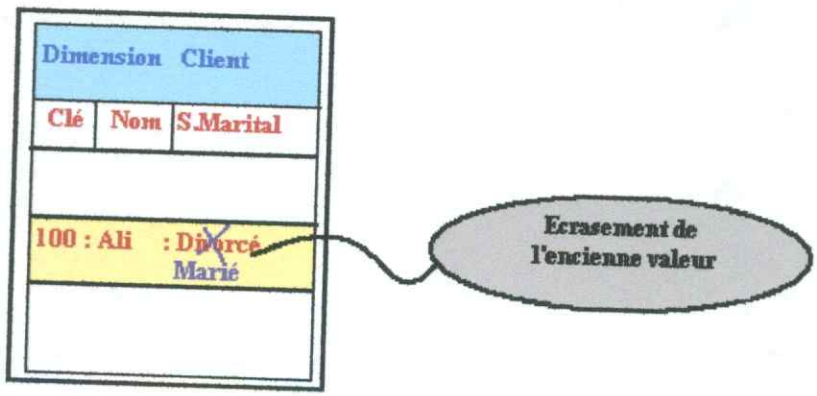


Fig. 31. Écrasement de la valeur précédente.

- **Ajout d'une ligne de dimension :** Créer un enregistrement de dimension supplémentaire lors du changement, comportant les nouvelles valeurs de l'attribut, ce qui revient à segmenter l'historique très exactement selon l'ancienne et la nouvelle description.

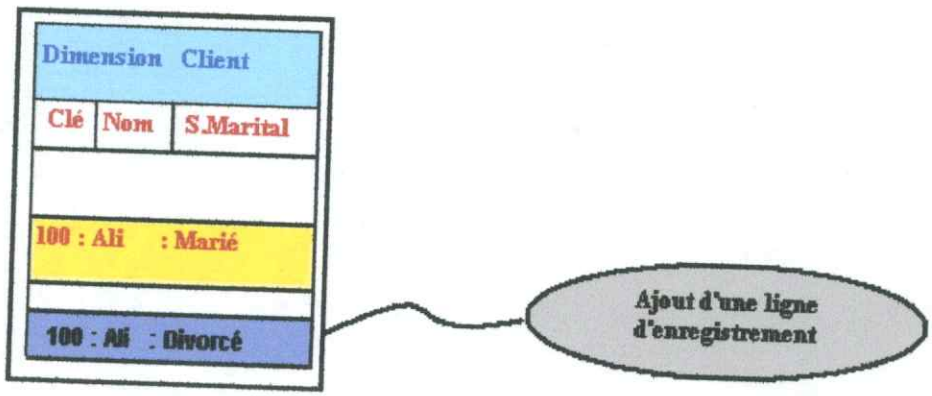


Fig.32. Ajout d'une ligne d'enregistrement.

- **Ajout d'une colonne de dimension :** Créer de nouveaux champs « actuel » à l'intérieur de l'enregistrement d'origine de la dimension, tout en conservant en même temps les valeurs enregistrées en premier lieu. Ceci permet de décrire l'historique en amont du changement en

utilisant les valeurs d'origine, et de décrire l'historique aval selon les nouvelles valeurs des attributs.

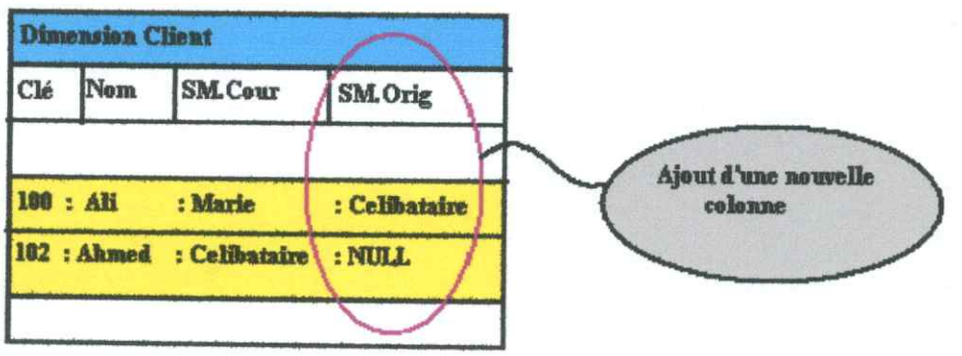


Fig. 33.Ajout d'une nouvelle colonne.

2.7. Les agrégats :

Les requêtes faisant appel à des données agrégées forment 80 % des demandes effectuées dans un système décisionnel. La construction des tables d'agrégats permet d'améliorer très nettement les temps de réponse de ces requêtes.

Agréger des donnes consiste à effectuer une sommation des données sur un axe hiérarchique afin de constituer les totaux correspondant à des étages sur cet axe.

Par exemple, les ventes seront stockées avec un niveau de détail correspondant aux ventes par produit par fournisseur et par magasin. Or la requête d'un utilisateur correspond aux ventes par famille de produits, par groupe de fournisseurs et par région, ne correspond pas au niveau d'informations disponible, il va falloir créer le niveau d'agrégations correspondant.

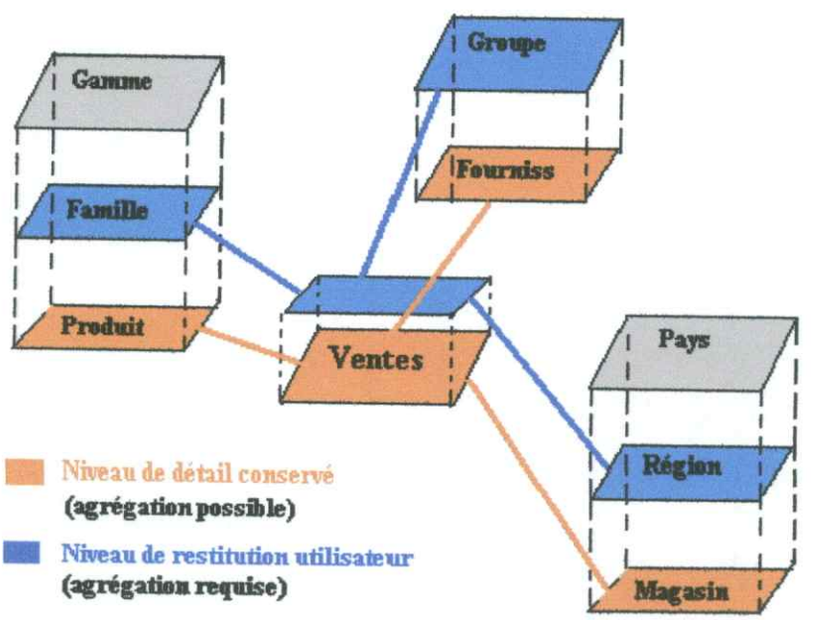


Fig. 34.Les différents niveaux d'agrégations.

2.8. L'étendue historique de la base de données :

Une des propriétés du data warehouse tient à ce qu'il permet d'historiser les données. Chaque nouvelle insertion de données provenant du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée. Cependant, il est fréquent que des niveaux de détail différents soient associes aux données selon leur récence.

Même s'il est toujours intéressant de conserver un important historique, il est rarement utile de conservera un niveau très détaillé les données les plus anciennes. Souvent on requière un niveau de détail moindre pour les données de plus de trois ans que pour les données plus récentes.

2.9. L'urgence avec laquelle les données doivent être extraite et chargées dans l'entrepôt de données :

Cette étape consiste à collecter les données utiles dans le système de production. Il faut donc d'abord identifier les données qui ont évolué afin d'en extraire aussi peu que possible, puis planifier ces extractions de façon à éviter les saturations (réseau, entrées/sorties, unité centrale).

Nous pouvons choisir de charger les données sur une base hebdomadaire ou plutôt en fonction de certaines phases d'inventaire ou de reporting. Le moment idéal pour réaliser ce chargement se situe de toutes façon en dehors des heures de fort trafic, c-à-d la nuit ou le week-end.

Enfin, nous devons garder à l'esprit qu'une bonne conception ne peut que résulter d'un équilibre entre les besoins des utilisateurs finaux et les réalités des données disponibles à partir des applications existantes.

3. La mise en œuvre de l'architecture

Cette étape se déroule en trois sous phases :

3.1. Construction de l'entrepôt

C'est une base de donnée relationnelle ayant pour rôle d'organiser et de ranger les données de plusieurs sources afin d'obtenir un format satisfaisant aux exigences des traqueurs d'informations. Elles se comportent comme un serveur d'aide a la décision qui transforme, agrège et ajoute de la valeur aux données en provenance de différentes sources de production. On emploie la modélisation dimensionnelle lors de la conception de base d'entrepôt de données pour organiser les informations et assurer ainsi l'efficacité des requêtes et permettre un accès rapide aux informations en vue de leur analyse et la création d'états les concernant.

3.2. Construction des cubes OLAP :

Un cube est un ensemble de données généralement bâti à partir d'un sous-ensemble d'un Data Warehouse et est organisé et synthétisé en une structure multidimensionnelle définie par un ensemble de **dimensions** et de **mesures.** »

Les cubes représentent la modélisation multidimensionnelle des données, il s'agit d'une part de données utiles pour un besoin d'analyse bien précis. Ils constituent une façon de faciliter l'analyse

3.3. Construction et étude de l'alimentation :

L'alimentation est la procédure qui permet de transférer des données du système opérationnel vers l'entrepôt de données en les adaptant. La conception de cette opération est une tâche assez complexe. Elle doit être faite en collaboration avec les administrateurs des bases de production qui connaissent les données disponibles et vérifient la faisabilité des demandes. Il est nécessaire de déterminer quelles données seront chargées, quelles transformations et vérifications seront nécessaires, la périodicité et le moment auxquels les transferts auront lieu.

3.3.1. Transformation et vérification

L'étude des besoins a déterminé le contenu de l'entrepôt en partant des ambitions des utilisateurs. Néanmoins, la forme, le contenu des données de production ne convient pas toujours immédiatement au format choisi pour les données de l'entrepôt. Par conséquent, des transformations sont souvent nécessaires.

a. Format

Le format physique des données provenant de la production peut ne pas être adéquat avec le système hôte de l'entrepôt. Les données pouvant provenir de serveurs différents dans des services différents, il est nécessaire d'uniformiser les noms et les formats des données manipulées au niveau de l'entrepôt.

b. Consolidation

Selon les choix des unités pour les dimensions, des opérations de consolidation devront accompagner le chargement des données (par exemple sommer les ventes pour obtenir et enregistrer un total par jour et non pas toutes les transactions).

c. Uniformisation d'échelle

Pour éviter de trop grandes dispersions dans les valeurs numériques, une homogénéisation des échelles de valeurs est utile. Ne pas la réaliser peut pénaliser les outils d'analyse et de visualisation et peut-être simplement remplir inutilement les disques.

d. Autres

Des transformations qui permettent de mieux analyser les données sont aussi réalisées pendant la phase de chargement. Par exemple, la transformation de la date de naissance en âge, assure une plus grande lisibilité des données et permet de pallier les problèmes apparus avec l'introduction de la dimension temps.

3.3.2. Métadonnées

Des choix conceptuels, logiques et organisationnels ont été opérés tout au long de cette démarche d'informatisation du système d'information de décision. La perte de l'information sur ces choix peut induire de mauvaises interprétations. Donc, un annuaire spécialisé conserve toutes les informations (les métadonnées) au sujet du système d'information qui régit l'entrepôt. Cet annuaire contient entre autres choses :

- Les adresses et descriptions des objets de l'entrepôt : tables, champs, ... En particulier chaque donnée est définie par une description précise et claire, son origine, ... surtout lorsque les données proviennent de bases différentes et qu'elles peuvent avoir d'autres significations par ailleurs.
- les utilisateurs, autorisations, ...
- les règles de gestion (transformation, de vérification, contrôle qualitatif...).
- les résultats d'exécution.
- l'historique de l'entrepôt avec les dates de chargement, ...

Donc cet annuaire ou dictionnaire de métadonnées est le référentiel de système d'information décisionnel, sans référentiel qui qualifie de façon précise ce que signifie chaque valeur dans la base, il n'est pas possible de conduire une analyse et interpréter les résultats. C'est ce rôle que joue l'annuaire des métadonnées.

3.3.3. Certification et publication

La cohérence de l'entrepôt est globale et ce n'est qu'à la fin de la procédure de chargement qu'il est possible de la vérifier. Une procédure de certification de la qualité des données chargées doit suivre le chargement. Cette procédure est particulière à l'entrepôt et ne peut être décrite précisément. Elle va par exemple vérifier la cohérence des données au niveau des activités. En cas de succès, la nouvelle version de l'entrepôt sera publiée. Dans le cas contraire, c'est souvent le chargement complet qui est

annulé et repoussé à une date ultérieure. En fait, le chargement peut être vu comme une très grosse (et longue) transaction.

3.3.4. Les outils ETL

Cependant il existe des outils qui facilites les étapes d'alimentations déjà présenter ; l'outil ETL (Extract, Transform and Load) récupère toutes les données et les centralise dans le Data warehouse, quels que soit leur sources et les système qui les supportent (Système d'exploitation, SGBD, fichiers plats, bases hiérarchiques...), d'automatiser et d'industrialiser le processus d'alimentation, de faciliter la maintenance des données et de limiter les développements spécifiques. Ces outils permettent de construire et de maître à jour et de maintenir le dictionnaire de métadonnées.

On évoque des ratios atteignant jusqu'à 35 % du budget total du projet pour la maîtrise de ce processus et 70 % des coûts et du temps dans sa réalisation **[Fra00]**. De nombreux facteurs ont une incidence sur ces chiffres : l'hétérogénéité de l'environnement, la complexité des règles de gestion pour la conversion et le contrôle des données sources et enfin la réactivité attendue du système.

Le processus ETL doit aussi s'effectuer périodiquement. La cible doit donc être optimisée en terme de temps de réponse (agrégats, rapprochements, redondances) et cela détermine la façon dont le processus de transformation doit être effectué.

Un processus ETL se décompose en trois phases : l'extraction, la préparation/transformation et le chargement.

a. L'extraction

Il s'agit en premier lieu d'aller chercher les données là où elles se trouvent. L'outil ETL a la capacité de se connecter aux différentes applications, bases de données ou fichiers. Pour cela, plusieurs technologies sont utilisables :

- Les passerelles fournies par les éditeurs de base de données.
- Les utilitaires de réplication, utilisables si les systèmes de production et décisionnels, sources et cibles, sont homogènes.
- Les outils spécifiques d'extraction.
- L'outil doit être capable de lire sélectivement les données sources, et donc de filtrer les données en lecture afin de n'extraire que l'information pertinente.
- En matière de rafraîchissement, l'idéal est de ne recharger la base cible qu'avec des données modifiées ou ajoutées depuis la dernière extraction.

De même, s'il est nécessaire de relancer une extraction, l'idéal est de ne pas devoir solliciter à nouveau les données sources d'autant que les données d'origine risquent de ne plus y être présentes, il est donc souhaitable de disposer d'une couche de stockage des données avant transformation : l'Operational Data Store (ODS) que certains outils intègrent dans leur gestion de flux. L'ODS permettra aussi de s'affranchir de certaines contraintes de temps en limitant la durée réelle de mobilisation de la source (production).

b. La préparation / transformation

Les bases sources sont organisées dans un format répondant aux besoins d'une application. Ce format n'a que peu de chances d'être compatible avec celui d'une autre application. Les données rendues accessibles puis acquises dans la phase extraction doivent donc être transformées.
Il s'agit d'un prétraitement incluant la mise en correspondance des formats de données, le nettoyage, la transformation et l'agrégation.
L'étape de contrôle s'effectue par application de règles adaptées sur les flux de données entrant. Bien souvent les incohérences ne sont découvertes que dans cette phase.
Plusieurs règles président au traitement des données suspectes :

La première consiste à les rejeter. Un rejet peut être :

- Bloquant, si l'un des éléments est suspect, on rejette tout.
- Non bloquant, dans ce cas les données cohérentes alimentent la base cible alors que les autres sont rejetées.

La seconde approche consiste à intégrer dans la base cible les données suspectes, avec une valeur par défaut. L'inconvénient de cette méthode est double : elle fausse certaines consolidations de données et elle rend difficile la reprise de l'historique.

Durant cette phase, les rejets doivent êtres journalisés et accompagnés de la cause des erreurs.
Les ETL sont des ateliers spécialisés dans la migration de données, leur transformation. Ils doivent fournir une fonction permettant de vérifier qu'une donnée est cohérente par rapport aux données déjà existantes dans la base cible. Ils doivent aussi fournir d'excellents outils pour convertir les données (par exemple un langage ou une interface graphique de description de transformation).
Enfin, ils doivent être conçus pour manipuler de gros volumes de données.

c. Le chargement

Le chargement prend en compte la gestion du format final des données.

Pour la mise en oeuvre du transfert de données, on distingue deux approches possibles :

Le transfert de fichiers : l'ETL transporte les données du système source vers le système cible via un moteur.

Le transfert de base à base. Dans ce cas, les outils travaillent en mode connecté, d'une source de données à une cible. Les données sont extraites ensemble à la source, puis transférées à la cible en y appliquant éventuellement des transformations à la volée. Un seul processus, plus rapide, a ainsi l'avantage de pouvoir à la fois effectuer les transferts et toutes les autres opérations d'alimentation, sans rupture. Ces processus sont spécifiques aux différentes bases et sont plus difficiles à normaliser.

CHAPITRE VI

Conception et Mise en Oeuvre du Data Warehouse

Introduction :

Apres avoir présenté les concepts concernant les data warehouses et leurs constructions dans cette première partie, nous proposons maintenant notre propre solution data warehouse.
Dans notre cas nous avons pris une base de donnée **« Gestion de Stock »** qui nous a été recommandé, surtout pour la richesse et la fiabilité en matière de données qu'elles la composes.
Commençons par une brève description de l'entreprise, elle comporte cinq magasins de stock repartis sur un ensemble de cinq régions (ROUIBA, DAR ELBIDA, MEFTAH, SIDI BELABAS, EL KHROUBE).chacun de ces magasins est un dépôt.

Comme nous l'avons évoqué dans la première partie de notre document (chapitre 5), la construction du data warehouse passe par trois étapes :

1. L'étude préalable qui définit les besoins, objectifs et précise la démarche.
2. La conception du modèle dimensionnel de données qui représente l'entrepôt conceptuellement et logiquement.
3. La mise en œuvre de l'architecture, par une suite de trois sous étapes :
    - Construction de l'entrepôt, la base de l'entrepôt ;
    - Construction des cubes OLAP, la base multidimensionnelle ;
    - Construction de la zone d'alimentation, qui reprend à un niveau plus précis l'examen des données, le choix des méthodes et des dates auxquelles les données entreront dans l'entrepôt.

1. Présentation de l'exemple : « Gestion des Stocks »

Dans notre projet, nous avons utilisé une base de données transactionnelle (base de production relationnelle) relative à une gestion de stock d'une entreprise d'un exercice complet (année 2004).

La première phase dans la mise en œuvre du système décisionnel a consisté d'abord à comprendre le métier (gestion stock) et le détail du contenu de la base de données afin de pouvoir dégager les besoins essentiels concernant l'activité gestion des stocks.

Comme la base de données choisie est très consistante, nous avons préféré retenir que les aspects les plus importants pour la construction du Data Warehouse. Voici en détail le contenue de la partie retenue :

- **Table Produit :**

Colonne	Désignation
Code_produit	Code produit
Designation	Désignation
Libelle_famille	Libelle famille
Libelle_sfamille	Libelle sous famille
Libelle_groupe	Libelle groupe
Libelle_organe	Libelle organe
Référence	Référence produit
U_M	Unité de mesure
.....	.....

- **Table Magasin :**

Colonne	Désignation
CODE_MAGASIN	Code Magasin
CODE_DOS	Désignation de l'entreprise
LIBELLE_MAGASIN	Désignation du magasin
.....	.....

- **Table Bon_Consommation :**

Colonne	Désignation
N_CONS	Numéro de consommation
CODE_MAGASIN	Code Magasin
DESIGNATION_MAGASIN	Désignation Magasin
DATE_BON	Date Bon
N_CONTRAT	Numéro de contrat
.....	.....

➢ **Table Ligne_Consommation :**

Colonne	Désignation
N_cons	Numéro de consommation
Code_produit	Code produit
Quantité_demandé	Quantité demandée
Quantité_sortie	Quantité sortie
CMUP	Coût moyen unitaire pondéré
Depot	Code dépôt
U_M	Unité de mesure
.....	.....

➢ **Table Bon_Réception :**

Colonne	Désignation
N_BRCP	Numéro bon réception
Num_contrat	Numéro contrat
Code_magasin	Code magasin
Code_fourn	Code fournisseur
Nom_frs	Nom fournisseur
Date_bon	Date bon
.....	.....

➢ **Table Ligne_Réception :**

Colonne	Désignation
N_BRCP	Numéro bon réception
Code_ produit	Code du produit
Quantité	Quantité reçue
Prix_ unitaire	Prix unitaire
Prix _ total	Prix total
Taxe	Taxe
Depot	Code depot
.....	.....

➢ **Table Bon_Réintégration :**

Colonne	Désignation
N_reint	Numéro de réintégration
Date_bon	Date bon
Code_mag	Code magasin
.....	.....

➢ **Table Ligne_DA :**

Colonne	Désignation
N_Da	
Région	
Structure	
Date_bon	
.....	.....

➢ **Table Ligne_Réintégration :**

Colonne	désignation
N_REINT	Numero reintegration
Code_produit	Code produit
Quantite	Quantité réintégré
Prix_unitaire	Prix unitaire
Prix_total	Prix total
Taxe	Taxe
Dépôt	Code dépôt
.....	.....

➢ **Table Demande_Achat :**

Colonne	Désignation
N_Da	
Région	
Structure	
Date_bon	
.....	.....

2. Modélisation dimensionnelle :

Le but d'un data warehouse est de fournir de l'information. Les sources de ces informations sont diverses. Par conséquent en dehors de la véracité des données il convient de construire un modèle de données performant, qui correspond aux besoins des utilisateurs.

Rappelons que la modélisation d'un data warehouse est un processus de conception de haut vers le bas, dans lequel nous appliquons notre connaissance des activités pour choisir les dimensions et les faits qui figureront dans notre base.

Comme nous l'avons invoqué dans la première partie de notre mémoire (chapitre 5), la conception logique d'un data warehouse passe par quatre premières étapes :

- Choix du processus d'activité à modéliser.
- Choix du grain du processus d'activité.
- Choix des dimensions applicables à chaque table de fait.
- Choix des mesures que contiendra chaque enregistrement de la table de fait.

2.1. Modélisation dimensionnelle de l'activité « DEMANDE d'ACHAT»

2.1.1. Le processus d'activité : dans une division stock, on ne gère pas les contrats ou les commandes d'achats. C'est pour cela, qu'en cas de rupture ou de forte demande de consommation sur certains produits, la division stock lance une demande d'achat auprès de la division Achats.

La demande d'achat contient les besoins en termes de produits pour l'approvisionnement des magasins de l'entreprise.

2.1.2. Le grain du processus d'activité :

Concernant cette activité, il est important de voir les produits demandés par chaque structure, par date. Il s'agit du niveau de détail le plus atomique fourni par le système opérationnel de gestions de stock. Les dimensions découlent immédiatement de cette déclaration du grain : temps, structure, produit.

2.1.3. Les dimensions :

Les dimensions concernées pour étudier l'activité Demande d'Achat sont :

a. dimension temps :

La définition de la dimension temps selon Ralph KIMBALL « la seule dimension qui figure systématiquement dans tous les entrepôts de données, car en pratique tout entrepôt est une série temporelle. Le temps est le plus souvent la première dimension dans le classement sous-jacent (subordonné) de la base de données » **[Kim97].**

Dans notre structure pour l'activité d'achat nous avons opté pour un grain quotidien des données. Ainsi chacun des enregistrements de notre table de dimensions temps représente un jour.

Fig.35. Dimension Temps.

Date	Jour	Mois	Trimestre	Année
01/01/2004	01	Janvier	Trimestre 1	2004
02/01/2004	02	Janvier	Trimestre1	2004
.....	.....	.....	.....	....
01/05/2004	01	Mai	Trimestre2	2004
29/10/2004	29	Octobre	Trimestre4	2004

Tab.7. Détails d'une table dimension Temps.

b. dimension produit :

La dimension « produit » décrit chaque produit du magasin, c'est l'une des dimensions principales de tout entrepôt de données. Elle doit comporter les plus grand nombre possible d'attributs descriptifs. Les produits sont identifiés par une clé unique, regroupés par groupe, sous famille et famille.

Fig.36. Dimension Produit.

Clé Produit	Produit	Groupe	Sous famille	Famille
100112103	Porte avant	504	Peugeot	Véhicules légers
1402123	Carburateur	504	Peugeot	Véhicules légers
140242172	Filtre à huile	205	Peugeot	Véhicules légers
05020040	Contre plaqué	Article bois	Article bois	Bois et dérives
3803006	Vitre	Vitrerie	Vitrerie	Droguerie miroiterie

Tab.8. Détails d'une table dimension Produit.

c. dimension structure :

La dimension structure nous permet de reconnaître qui était à l'origine des différentes transactions (réceptions, consommations, réintégrations…) des produits durant une tranche de temps donnée.

Les structures sont identifiées par une clé unique, regroupés par région ainsi que le nom de la structure.

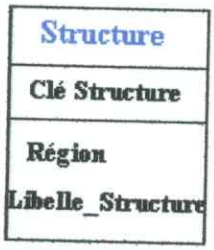


Fig.37. Dimension Structure

Code Structure	Région	Nom Structure
5002	Centre	DHW Blida Rénovation 06 Stations de pompages
5012	Centre	Véhicules Légers
5008	Est	Atelier Forage et Fabrication
5103	Ouest	Tôlerie et Peinture
5056	Centre	APC Blida Station de Pompage

Tab.9. Détails d'une table dimension Temps.

2.1.4. Les faits mesurés : Les faits mesurés que nous avons enregistrés sont :

- Quantité_Demandé

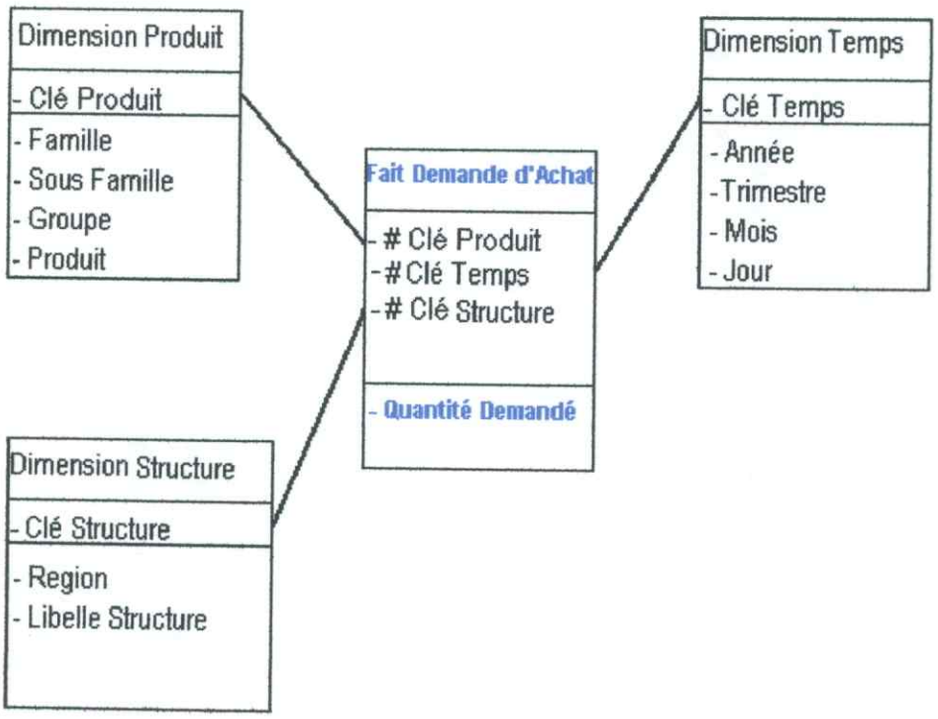


Fig.38. Modèle dimensionnel en étoile de l'activité « Demande d'Achat».

2.2. Modélisation dimensionnelle de l'activité « RECEPTION / ACHAT »

2.2.1. Le processus d'activité : Sur la base de notre étude et de la connaissance de l'activité et des ressources de données disponibles, la première activité importante, Après celle des demandes d'achats, est l'activité « **RECEPTION / ACHAT** ». Nous nous intéressons d'avantage sur cette activité étant donné que c'est l'activité clé des services « gestion stock » et elle consiste à approvisionner les différents magasins de l'entreprise en passant par des achats chez différents fournisseurs, ainsi la modélisation de cette activité va nous permettre de répondre aux questions suivantes :

- Quels produits sont achetés le plus souvent ?
- Combien de fournisseurs vendent ces produits ?
- Quelles sont les performances de nos fournisseurs ?

Cette activité correspond à l'opération de réception au niveau système opérationnel.

2.2.2. Le grain du processus d'activité :

Le grain est important parce qu'il détermine les différentes dimensions de la base de données. En général, la granularité de la table des faits choisie est la plus fine possible, plus le niveau de détail est fin plus la conception est robuste.

Pour l'activité d'achat, il est important de voir les produits achetés dans chaque magasin, par date et par fournisseur, il s'agit du niveau de détail le plus atomique fourni par le système opérationnel de gestions de stock. Les dimensions découlent immédiatement de cette déclaration du grain : temps, magasin, produit, fournisseur.

2.2.3. Les dimensions :

D'après la granularité de la table de faits, nous avons déterminé une série de dimensions principales. Cette série comporte les dimensions suivantes : Temps, Produit, Magasin, Fournisseur.

Les tables de dimensions servent à enregistrer les descriptions textuelles des dimensions de l'activité, c'est les attributs, ils sont utilisés pour décrire des aspects d'une dimension qui peuvent être utilisé comme source de contraintes par les utilisateurs.

Les dimensions retenues pour ce processus d'activité sont :

- « Dimension Temps ».
- « Dimension Produit ».

Ainsi que :

- **dimension magasin :**

La dimension magasin sert à reconnaître le lieu où s'est effectuer les différentes transactions (réceptions, consommations, réintégrations, transfert…).

La dimension magasin est la principale dimension géographique de notre entrepôt de données.

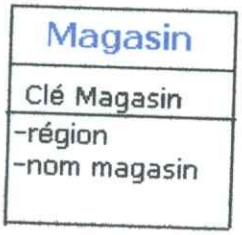


Fig.39. Table de dimension Magasin.

Code Magasin	Région	Nom magasin
1001	Centre	Magasin D.E.B
1002	Centre	Magasin Rouiba
1003	Est	Magasin Direction Kharoub
1004	Ouest	Magasin S.B.A
1005	Centre	Magasin Meftah

Tab.10. Détails d'une table dimension Magasin

Dimension fournisseur :

La dimension Fournisseur permet de décrire chaque fournisseur travaillant avec l'entreprise. Ils sont identifiés par une clé unique, un nom et sont regroupés par Type.

Fig.40. Table de dimension Fournisseur.

Clé fournisseur	Nom fournisseur	Type fournisseur
02	Sonatrach	Etatique
013	Ediel El-Achour "Equipement Electrique"	Privé
E.005	K.S.B ALLEMAGNE	Etranger
E.011	Schneider Electric France	Etranger

Tab.11. Détails d'une table dimension Fournisseur.

2.2.4. Les faits mesurés : la quatrième et dernière étape de la conception est la détermination des faits qui apparaîtront dans la table de faits. Ici encore, notre choix est en fonction de la granularité, voir les produits achetés dans chaque magasin, par date et par fournisseur. Les faits collectés par le

système opérationnel incluent la quantité achetée (par exemple le nombre ...) et le montant d'achat en dinars.

Les faits mesurés que nous avons enregistrés sont :

- Quantité_entrée. (acheté/reçue)
- Montant_TTC

Ces faits sont parfaitement additifs sur toutes les dimensions. Nous pouvons couper la table en tranches et en dés sans risques, toute somme de ces faits est valide et correcte.

Après avoir défini les faits et les attributs des dimensions, notre schéma conceptuel de l'activité **« ACHAT»** se présente comme suit :

Que choisir ?

Le schéma en étoile est, certes, plus redondant que le schéma en flocon mais :

La redondance n'est pas un problème pour le décisionnel, puisqu'il n'y a que des requêtes de sélection

- ✓ L'espace occupe par les tables de dimension est négligeable devant celui de la table des faits.
- ✓ Les requêtes sont plus rapides sur un schéma en étoile.
- ✓ L'ETL est plus simple avec un schéma en étoile.

Pour toutes ces raisons, le schéma en flocon est peu recommandé.

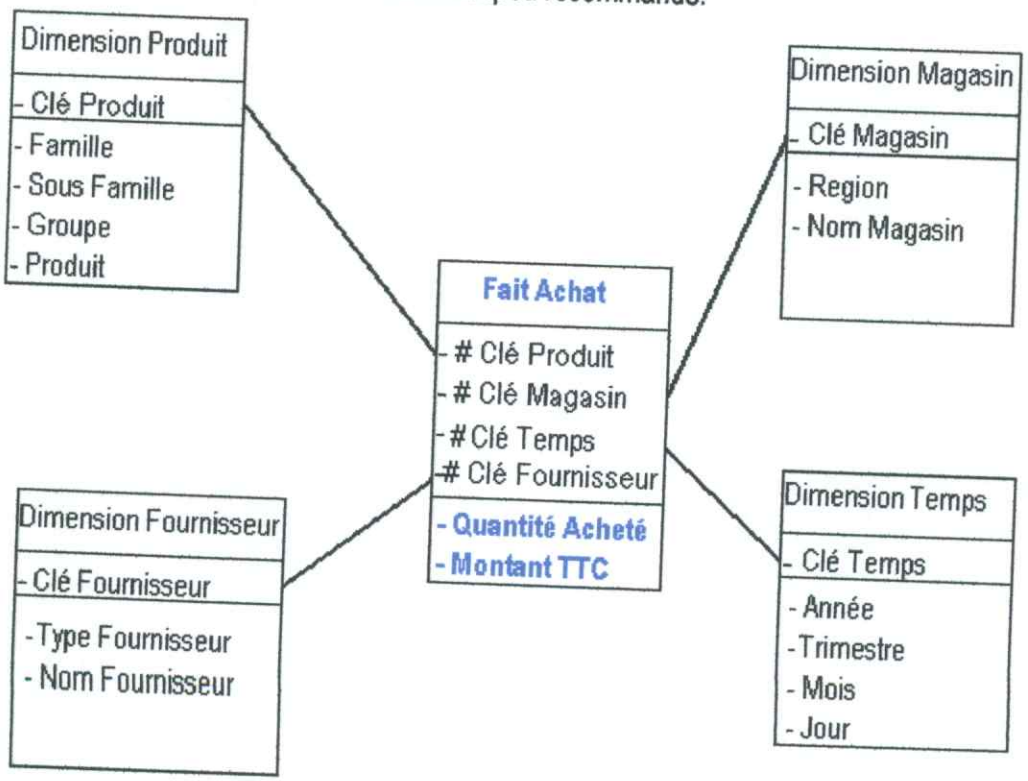


Fig.41. Modèle dimensionnel en étoile de l'activité « RECEPTION / ACHAT »

2.3. Modélisation dimensionnelle de l'activité « CONSOMMATION »

2.3.1. Le processus d'activité : la consommation est le processus inverse de la réception puisqu'il s'agit d'une sortie de stock pour des fins de consommation des produits.

2.3.2. Le grain du processus d'activité :

Pour cette activité, il est important de voir les produits consommés par différentes structures dans chaque magasin, par date, il s'agit du niveau de détail le plus atomique fourni par le système opérationnel de gestions de stock. Les dimensions découlent immédiatement de cette déclaration du grain : temps, magasin, produit, structure.

2.3.3. Les dimensions :

C'est les mêmes dimensions étudiées qu'au par avant pour l'activité « achat ».

- « Dimension Temps »
- « Dimension Produit »
- « Dimension Magasin »
- « Dimension Structure»

2.3.4. Les faits mesurés : Les faits mesurés que nous avons enregistrés sont :

- Quantité_sortie. (Consommé)
- Montant des consommations.

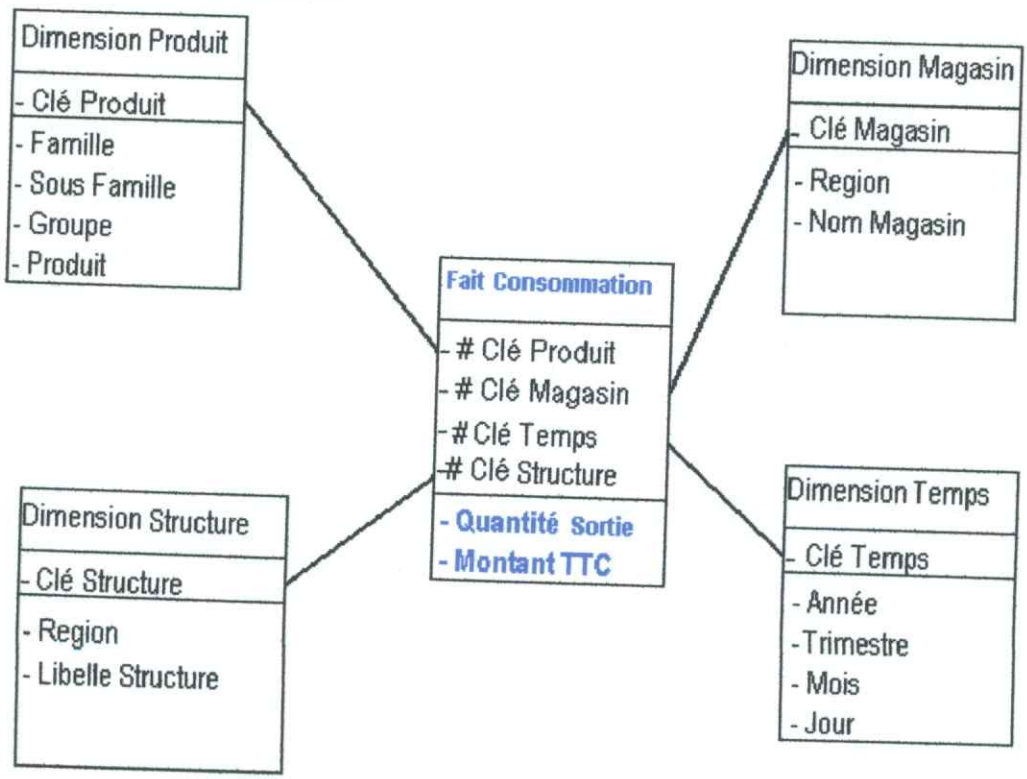


Fig.42. Modèle dimensionnel en étoile de l'activité « CONSOMMATION »

2.4. Modélisation dimensionnelle de l'activité « RÉINTÉGRATION »

2.4.1. Le processus d'activité : une réintégration se produit lorsque les produits demandés pour consommation ne sont pas totalement consommés. La partie restante (non consommé) sera réintégrée au stock.

2.4.2. Le grain du processus d'activité :

Pour cette activité, il est important de voir les produits réintégrés dans chaque magasin, par différentes structures de l'entreprise après les avoirs octroyer pendant un certain temps. Il s'agit du niveau de détail le plus atomique fourni par le système opérationnel de gestions de stock. Les dimensions découlent immédiatement de cette déclaration du grain : temps, magasin, produit, structure.

2.4.3. Les dimensions :

Les dimensions concernées pour étudier l'activité Réintégration sont :

- « Dimension Temps »
- « Dimension Produit »
- « Dimension Magasin »
- « Dimension Structure »

2.4.4. Les faits mesurés : Les faits mesurés que nous avons enregistrés sont :

- Quantité_réintégré
- Montant des Réintégration

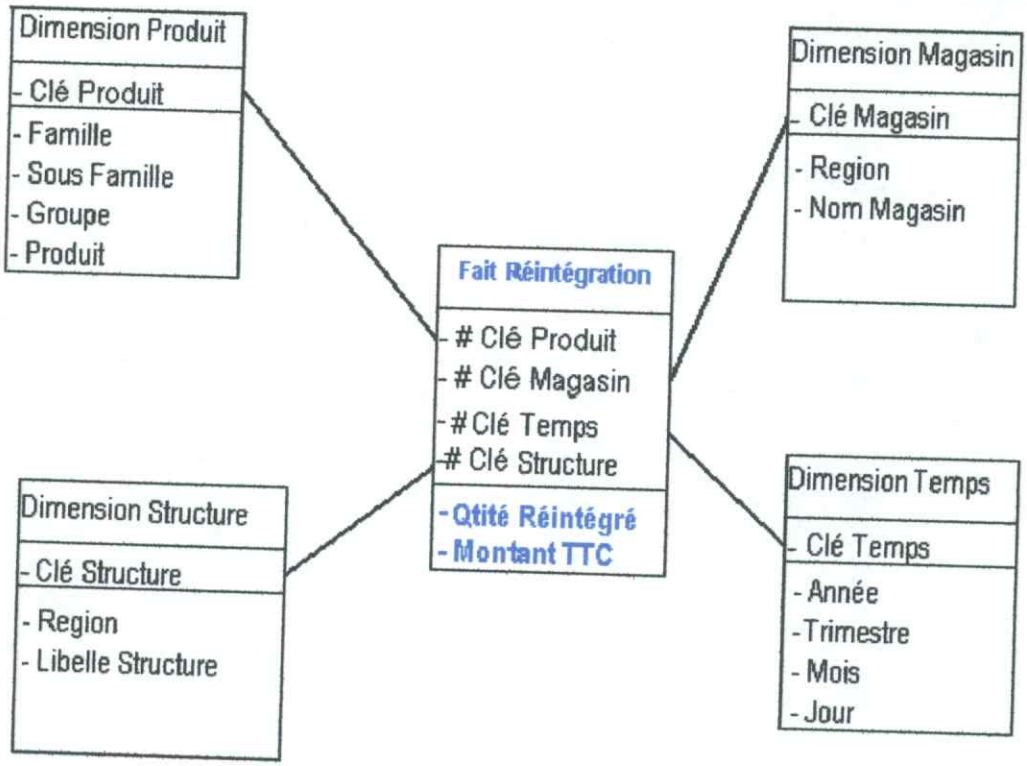


Fig.43. Modèle dimensionnel en étoile de l'activité « Réintégration»

3. Mise en œuvre de l'entrepôt : Dans cette partie, nous allons présenter les processus suivi pour la mise en oeuvre de notre data warehouse. La mise en œuvre de l'entrepôt comme nous l'avons déjà vu passe par trois grandes étapes, chaque une de ces étapes représente un composant de l'architecture d'un data warehouse. L'architecture de notre data warehouse est représenté dans le schéma suivant, avec exposition des correspondances entre chaque composant et l'étape qui lui correspond dans le processus de mise en oeuvre.

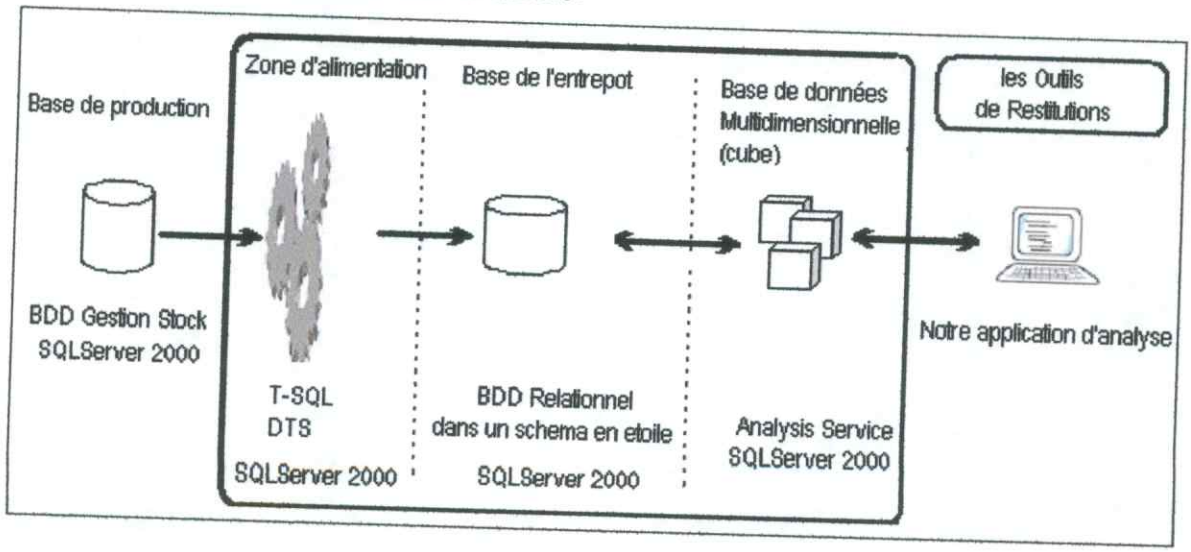


Fig.44. Architecture du Data WareHouse G-S-M

3.1. Construction de la base de l'entrepôt :

C'est une base de données (relationnelle) dont l'objectif est de centraliser l'information décisionnelle, elle représente la base de données physique de notre entrepôt. Cette base répond à notre modèle de données, en d'autres termes c'est la transformation de la conception logique en une base de données physique. Chaque entité dans notre modélisation, table de fait ou dimension, est transformée en une table de base de données relationnelle.

A chaque dimension on associe une table de base de données relationnelle avec :

- ✓ Une clé primaire.
- ✓ Et une colonne par niveau (pour y stocker les membres).

Par exemple, à la dimension produit on associe une table ayant pour clé primaire Clé Produit et comme autres colonnes : Famille, Sous Famille, Groupe et Produit

Et a chaque table de fait on associe une table de base de données relationnelle avec :

- ✓ Des clés étrangères.(clé primaire de chaque table dimension)
- ✓ Et une colonne par mesure.

Donc, notre base décisionnelle (base d'entrepôt) contient les tables suivantes :

	Nom de la table	Clé
01	Produit	Clé_Produit
02	Magasin	Clé_Magasin
03	Structure	Clé_Structure
04	Fournisseur	Clé_Fournisseur
05	Fait_demande_achat	Clé_Produit ;Clé_Structure ;Clé_Temps
06	Fait_réception	Clé_Produit ;Clé_Fournisseur ;Clé_Magasin ;Clé_Temps ;
07	Fait_consommation	Clé_Produit ;Clé_Magasin ;Clé_Temps ;Clé_Structure ;
08	Fait_réintégration	Clé_Produit ;Clé_Structure ;Clé_MagasinClé_Temps
09	Temps	Clé_Temps

Tab.12. Tables de la base Décisionnelle.

Le schéma relationnel de l'entrepôt regroupe les schémas relationnels de Ses cubes (schéma en étoile). La base de données qui correspond à ce schéma relationnel global, s'appelle la base décisionnelle (Par opposition aux bases de production).

Certaines dimensions sont communes a plusieurs cubes (la dimension produit est commune aux Cubes Réception et Consommation, par exemple). Leurs tables ne sont évidemment pas répétées dans le

schéma relationnel de l'entrepôt, mais utilisées par plusieurs tables des faits. C'est pourquoi Analysis Services emploie le terme de dimensions partagées.

Voici Le schéma relationnel de notre base décisionnelle :

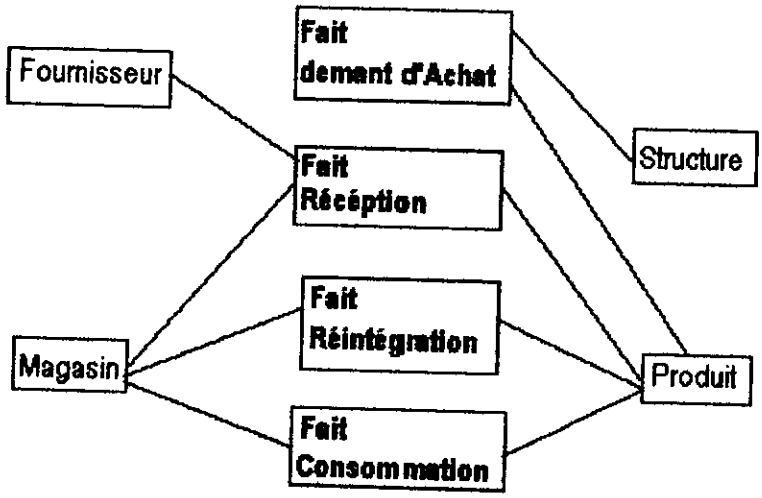


Fig.45. Le schéma relationnel de la base décisionnelle.

3.2. Construction des cubes OLAP :

Les cubes représentent notre base de données multidimensionnelle, Chaque cube représente un fait ou un besoin d'analyse bien précis, Ils facilitent l'analyse aux utilisateurs finaux et les performances des requêtes, en stockant les agrégations des données de l'entrepôt.

Chaque cube possède un schéma de données (étoile ou flocon), qui est l'ensemble des tables jointes appartenant au Data Warehouse. La table centrale du schéma est la table de faits, qui contient les mesures du cube. Les autres tables sont les tables de dimension, qui contiennent les dimensions du cube.

Nous avons choisi comme technique de modélisation (mode de stockage) pour la mise en œuvre des cubes la modélisation MOLAP, parce qu'elle :

- ✓ Offre un meilleur temps de réponse aux requêtes MDX
- ✓ Elle peut être adoptée pour les bases allant jusqu'à une dizaine de giga octet, ce qui nous semble être suffisant pour notre petit entrepôt en tenant compte d'un taux de croissance important.
- ✓ Est la simple à maître en oeuvre, on a pas à gérer les agrégats elle agrège tout par défaut
- ✓ Le serveur OLAP utilisé est MS SQL Server 2000 OLAP Service.

Les figures suivantes illustrent notre modèle physique de chaque cube MOLAP en utilisant le formalisme Suivant :

Concept de fait : nous représentons le fait par un cube englobant les différentes mesures d'activité qu'il contient. Tout en l'indiquant à l'aide du symbole suivant " " :

Fig.46. Concept de Fait.

Concept de dimension : nous représentons les dimensions par un rectangle englobant les différents paramètres d'analyse qu'il contient. Tout en l'indiquant à l'aide du symbole" " représentant trois axes estampille pour le distinguer des tables de faits.

Fig.47. Concept de Dimension.

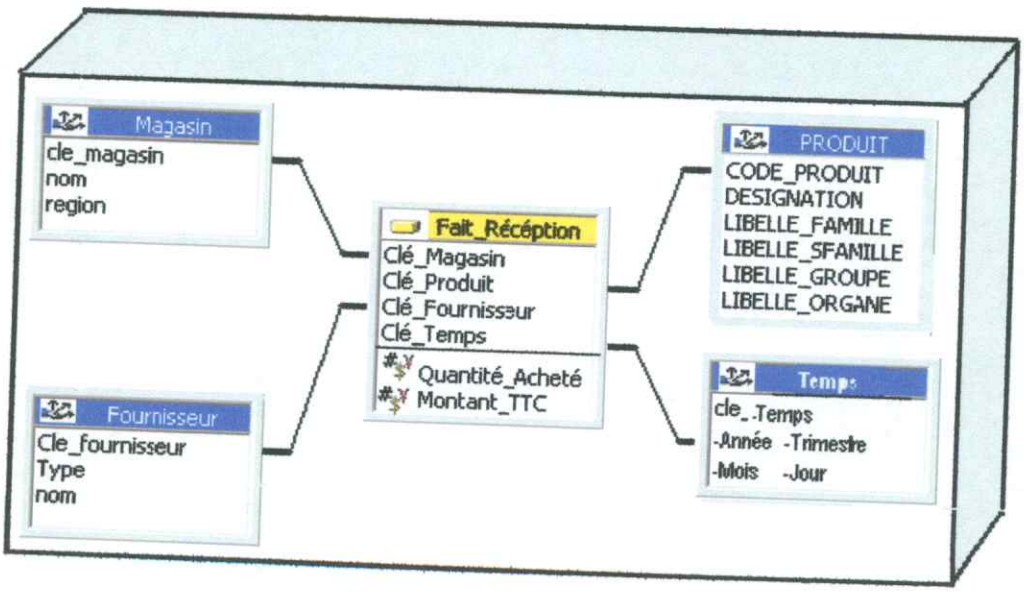


Fig.48. Modélisation du cube Réception.

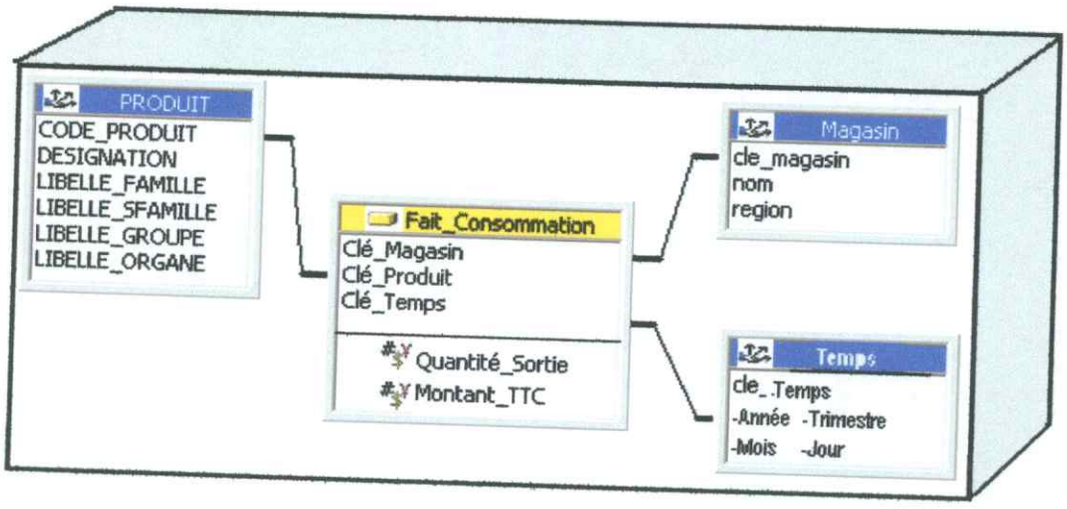


Fig.49. Modélisation du cube Consommation.

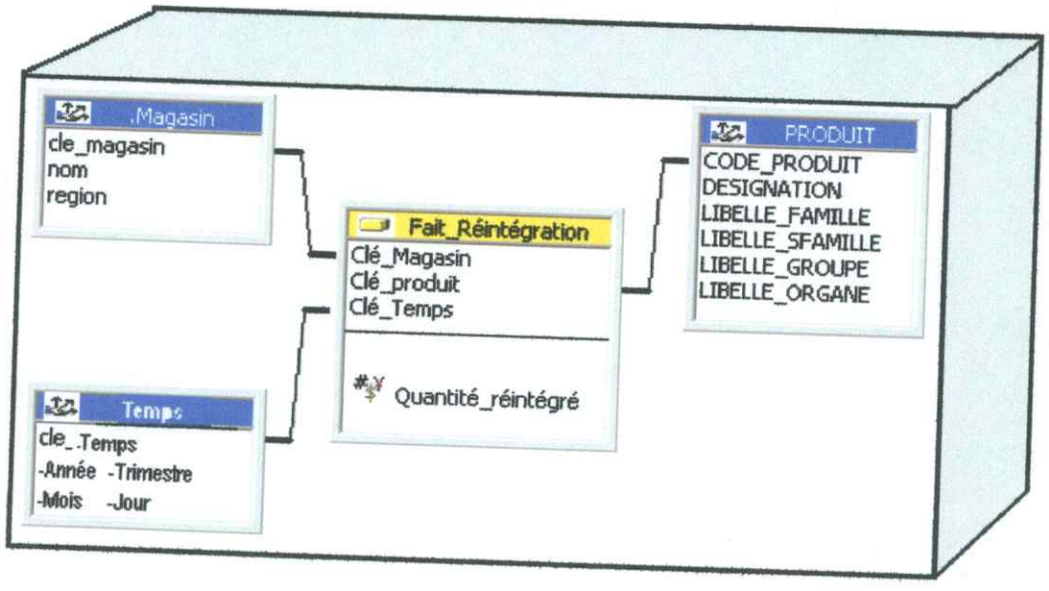


Fig.50. Modélisation du cube Réintégration.

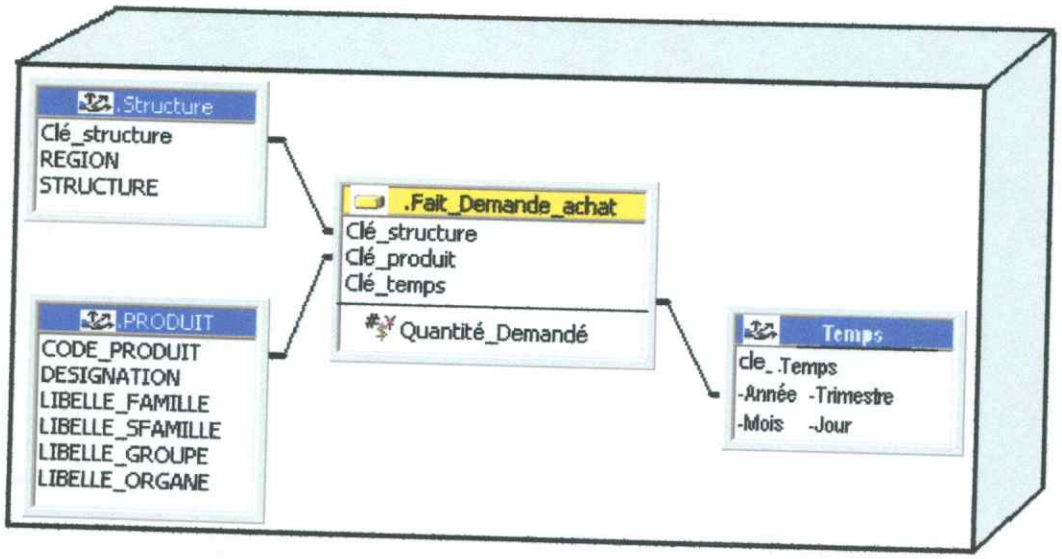


Fig.51. Modélisation du cube Demande Achat.

3.3. Construction de la Zone d'alimentation :

Les données telles qu'elles existent dans les systèmes OLTP peuvent rarement être exploitées directement pour une aide à la décision complexe. Il est nécessaire de déterminer, à partir d'une foultitude de données, lesquelles seront chargées, quelles transformations et vérifications seront nécessaires, la périodicité et le moment auxquels les transferts auront lieu.

L'alimentation est la procédure qui permet de transférer ces données du système opérationnel vers l'entrepôt de données en les adaptant à la structure réceptrice sans pour autant en modifier l'information utile.

La conception de cette opération est une tâche assez complexe. Elle doit être faite en collaboration avec les administrateurs des bases de production qui connaissent les données disponibles et vérifiant la faisabilité des demandes.

Comme le montre le schéma ci-dessous ce processus se déroule en trois grandes étapes (extraction, transformation, chargement) :

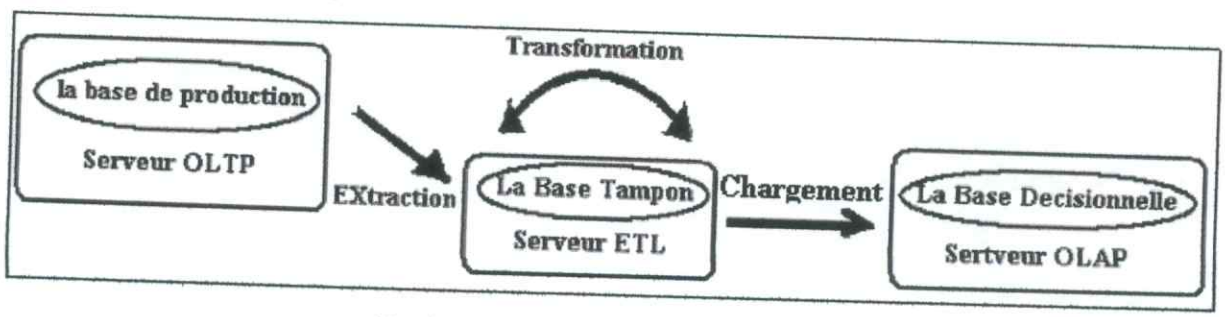


Fig.52. Les étapes du processus ETL.

Au cours de la construction de notre data warehouse nous avons effectué un chargement initial très simple qui consiste à peupler les tables puis à vérifier que les données sont prêtes à être utilisé. Mais c'est historisé périodiquement et automatiquement les données qui constituent la tâche la plus complexe et que nous détaillerons dans ce qui suit.

3.3.1. La base tampon

Nous avons proposé cette base de données intermédiaire Étant donné que l'ETL des données OLTP pour le système OLAP entre en concurrence avec les sauvegardes des bases de production. Il faut donc que cette phase perturbe le moins longtemps possible les systèmes OLTP.

Ainsi les données extraites doivent atterrir sur une autre base, appelée base tampon (staging area).

Une fois l'étape d'extraction terminée, les transformations nécessaires. Peuvent être effectuées tranquillement dans la base tampon. Il ne faut pas non plus que le système OLAP ne soit perturbe par la phase ETL (en particulier, par l'étape de transformation). Autrement dit, cette base tampon ne doit pas être la base décisionnelle et doit être gérée par un serveur dédie à l'ETL.

3.3.2. Etapes ETL

Détaillons maintenant les étapes ETL de la figure d'en haut :

a)Extraction

L'approche qui convient le mieux est de charger dans le data warehouse uniquement les nouvelles informations. Ainsi nous établissons des règles d'extraction de données opérant seulement sur celles qui ont changé depuis le dernier chargement.

Pour que l'étape d'extraction dure le moins longtemps possible, il faut que la requête de sélection ne comporte aucune jointure (il faut donc extraire les tables une par une).

Par ailleurs, il est bon que dans les systèmes OLTP, chaque table concernée par l'extraction (Magasin, produits, etc.) soit munie d'une colonne pour la date de création et une autre pour la date de dernière modification. Sans ces colonnes, on serait oblige d'extraire toutes les lignes et il serait complique de déterminer (dans la base tampon) les lignes réellement modifiées depuis la dernière extraction.

Un lot DTS implémenté sous forme de requête SQL pourrait par exemple être planifié pour importer (extraire) toutes les commandes de produit qui correspondent à une plage spécifique, comme le montre le schéma suivant :

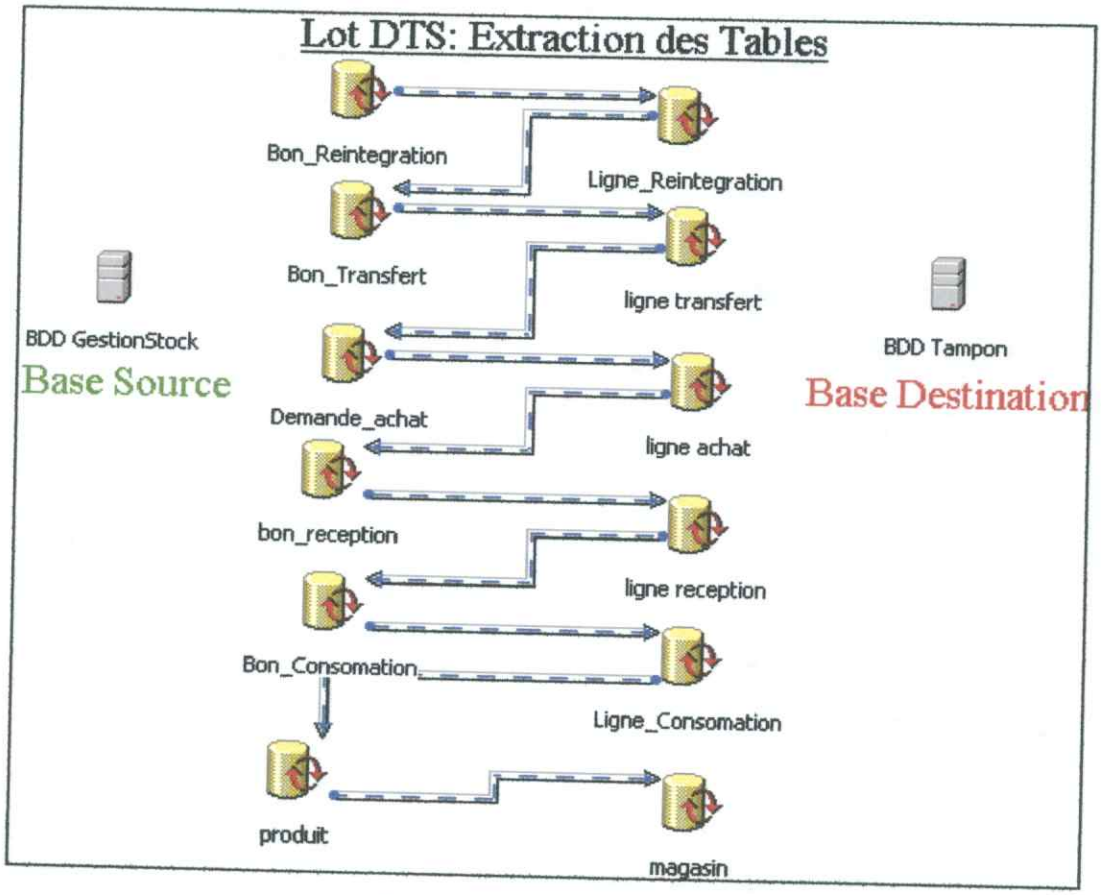


Fig.53. Lot DTS Extraction.

b) Transformation

Ce n'est pas parce que les données proviennent des bases de production qui fonctionnent rigoureusement bien, que ces données sont valides pour le système décisionnel. Il faut presque toujours les transformer. Pendant que les données sont insérées dans les tables tampon, on peut les uniformiser, c'est-à-dire les réparer, les compléter, les synchroniser et les formater.

Les transformations se font

- ✓ d'abord, pendant le passage des données des tables temporaires aux tables tampon ;
- ✓ ensuite, des modifications sont apportées au sein des tables tampon en vue du chargement.

Le schéma suivant illustre un exemple de transformation nous permettant de réparer des données du système opérationnelle.

Après avoir soigneusement parcouru notre base de données opérationnelle, nous avons trouvé que certain colonnes de la table produit comportent des champs "null".

Comme la montre la figure suivante :

CODE_PRODUIT	LIBELLE_FAMILLE	LIBELLE_SFAMILLE	LIBELLE_GROUP	LIBELLE_ORGANE
0100001	VENTE EN ETAT SPECIFIQUE-----	<NULL>	<NULL>	<NULL>
0101001	VENTE EN ETAT SPECIFIQUE-----	VENTE EN ETAT SEPCIFIQUE	FONTE	<NU L>
0101002	VENTE EN ETAT SPECIFIQUE-----	VENTE EN ETAT SEPCIFIQUE	FONTE	<NULL>
0101003	VENTE EN ETAT SPECIFIQUE-----	VENTE EN ETAT SEPCIFIQUE	FONTE	<NULL>
0101004	VENTE EN ETAT SPECIFIQUE-----	<NULL>	FONTE	<NULL>
0101005	VENTE EN ETAT SPECIFIQUE-----	VENTE EN ETAT SEPCIFIQUE	FONTE	<NULL>
0301001	MATERIAUX DE CONSTRUCTION	LIANTS CIMENT ET DERIVES	<NULL>	<NULL>

Fig.54. Table produit avant la correction.

Ce qui provoquerait d'une manière certaine l'arrêt du traitement du cube au moment du calcul de touts les agrégats possibles.

Afin de pallier à ce problème nous avons procédé à la correction de ces données en définissant des tâches de transformation à l'aide des lots DTS de telle sorte que :

-Si un champ comporte un "Null" alors nous lui affectons le contenu du champ parent.

Le schéma DTS réalisant ces tâches est configuré comme suivant :

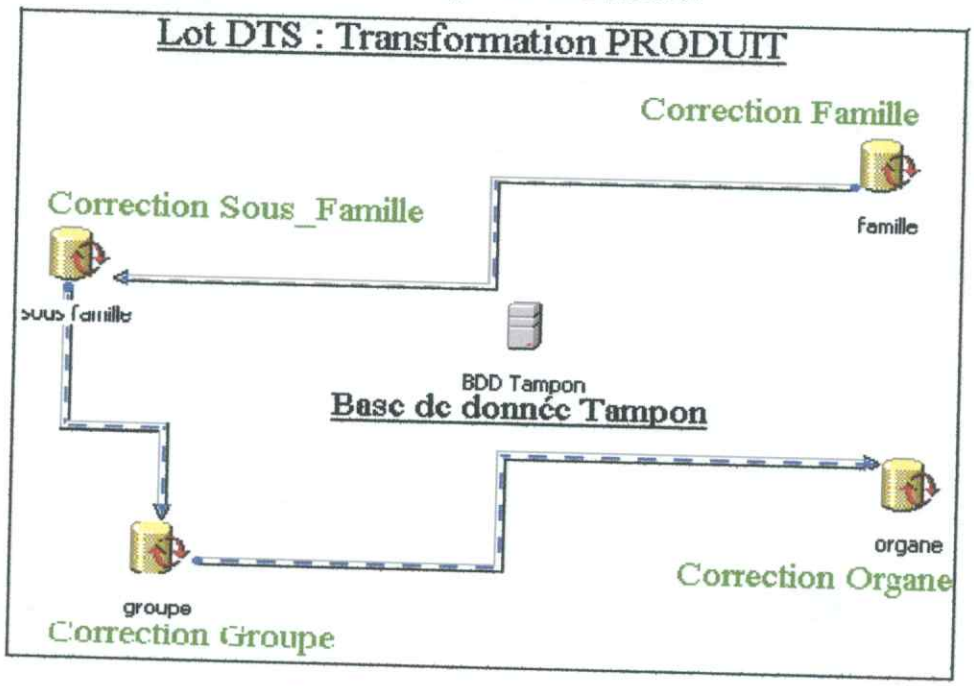


Fig. 55.Lot DTS Transformation.

Et la table "Produit" au niveau de la base tampon, après correction, devient comme suivant :

CODE_PRODUIT	LIBELLE_FAMILLE	LIBELLE_SFAMILLE	LIBELLE_GROUPE	LIBELLE_ORGANE
0100001	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SPECIFIQUE-	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT S
0101001	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SEPCIFIQUE	FONTE	FONTE
0101002	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SEPCIFIQUE	FONTE	FONTE
0101003	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SEPCIFIQUE	FONTE	FONTE
0101004	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SEPCIFIQUE	FONTE	FONTE
0101005	VENTE EN ETAT SPECIFIQUE	VENTE EN ETAT SEPCIFIQUE	FONTE	FONTE

Fig.56. Table produit après la correction.

c)Chargement

Une fois que nous avons sélectionné les informations à stocker et leur avons appliqué les transformations nécessaires, nous serons confronté au chargement et à l'actualisation périodiques de données dans le data warehouse. Le moment idéal pour réaliser ce chargement se situe de toute façon en dehors des heures de fort trafic, c'est-à-dire la nuit ou le week-end.

Comme les données sont chargées dans la base décisionnelle qui est muni d'un schéma relationnel (en étoile), il faut charger ses tables dans cet ordre :

- D'abord les tables qui ne contiennent aucune clé étranger.
- Ensuite les tables qui ne contiennent que des clés étrangères vers des tables déjà chargées.

Le schéma ci-dessous illustre un exemple de tâche planifiée permettant de charger périodiquement et automatiquement la table de fait consommation de la base tampon vers la base décisionnelle (Data Warehouse) :

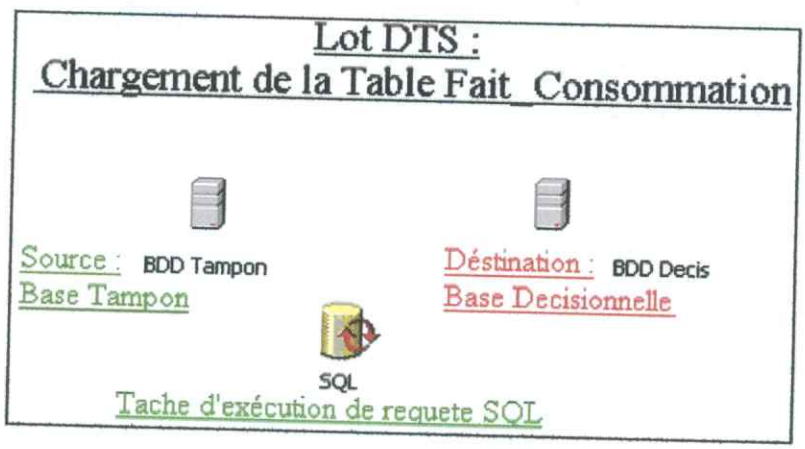


Fig.57. Lot DTS Chargement.

Une fois le data warehouse implémenté, les routines d'actualisations périodiques représentent la part la plus importante du travail de maintenance.

4. Conclusion

Dans ce chapitre, Nous avons présenté la partie qui constitue l'arrière plan de notre système décisionnel, la modélisation et la mise en œuvre de notre data warehouse.

Notre modélisation repose sur un schéma multidimensionnel, dont l'intérêt est de présenter l'information d'une manière adaptée aux analyses OLAP. L'avantage de notre modèle réside dans sa capacité à se détacher des choix ROLAP, MOLAP, HOLAP. Ce modèle repose sur un schéma en étoile comportant un fait et des dimensions, ces dimensions peuvent être partagées par plusieurs tables de faits formant un schéma en constellation. Ce partage des dimensions entre plusieurs faits limite la redondance, la complexité, et l'espace de stockage à l'implémentation.

Dans la mise en œuvre, nous avons implémenté notre base décisionnelle ainsi que les cubes OLAP et la zone d'alimentation des données, répondant ainsi à l'architecture globale de data warehouse.

Présentation de l'Application G.S.M 2005

G.S.M 2005 est l'application cliente utilisé pour présenter les données analytiques contenues dans le serveur d'analyse (Analysis services), afin de faire une analyse en ligne.

Ce rapport illustre la méthode de fonctionnement de notre Application **G.S.M 2005**. Et cela à travers l'explication des différentes étapes suivantes :

1. Lancement de l'application **G.S.M 2005**
2. Identification de l'utilisateur.
3. Choix de la base de données décisionnelle à laquelle l'utilisateur souhaite se connecter.
4. Choix du cube de données sur lequel l'utilisateur souhaite faire l'analyse.
5. Exploration des données du cube choisi à l'aide d'un tableau Croisé dynamique.
6. Exemple d'analyse à l'aide du tableau.
7. Présentation des données graphiquement en choisissant un model de représentation graphique.
8. Exemple d'analyse à l'aide des graphes.

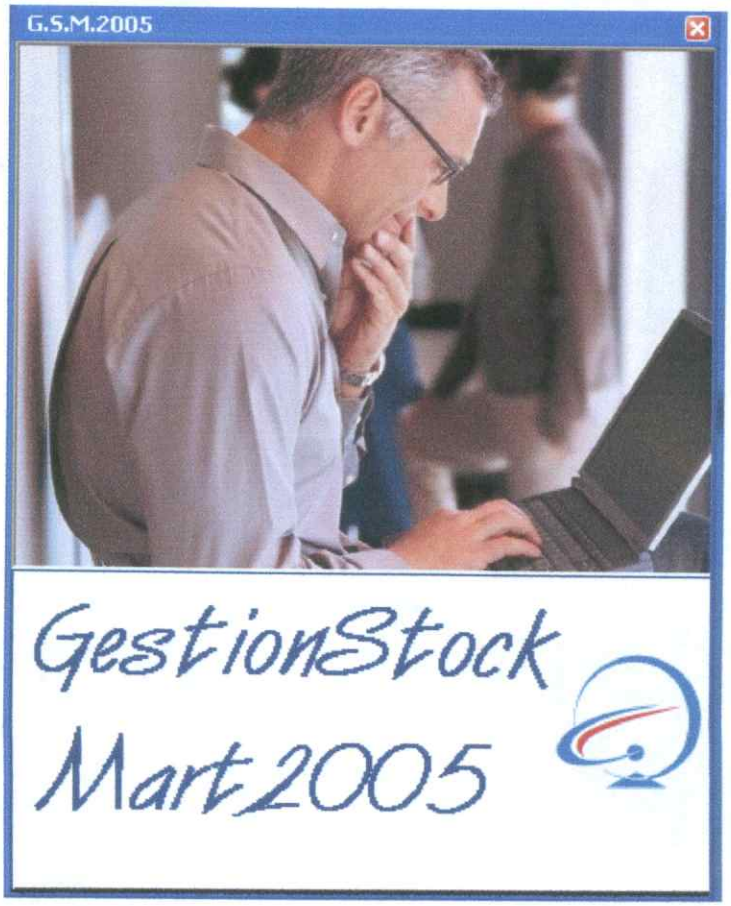


Fig. Fenêtre d'accueil de « G.S.M.2005 »

Après avoir lancé l'application, à l'apparition de la **fenêtre d'accueil**, L'utilisateur survole à coup de souris la fenêtre d'accueil. Dans la zone ou le curseur de la souris prend la forme d'une main il faut **« Cliquer »** pour valider son entré, une fois que c'est fait, la **fenêtre principale** apparaît.

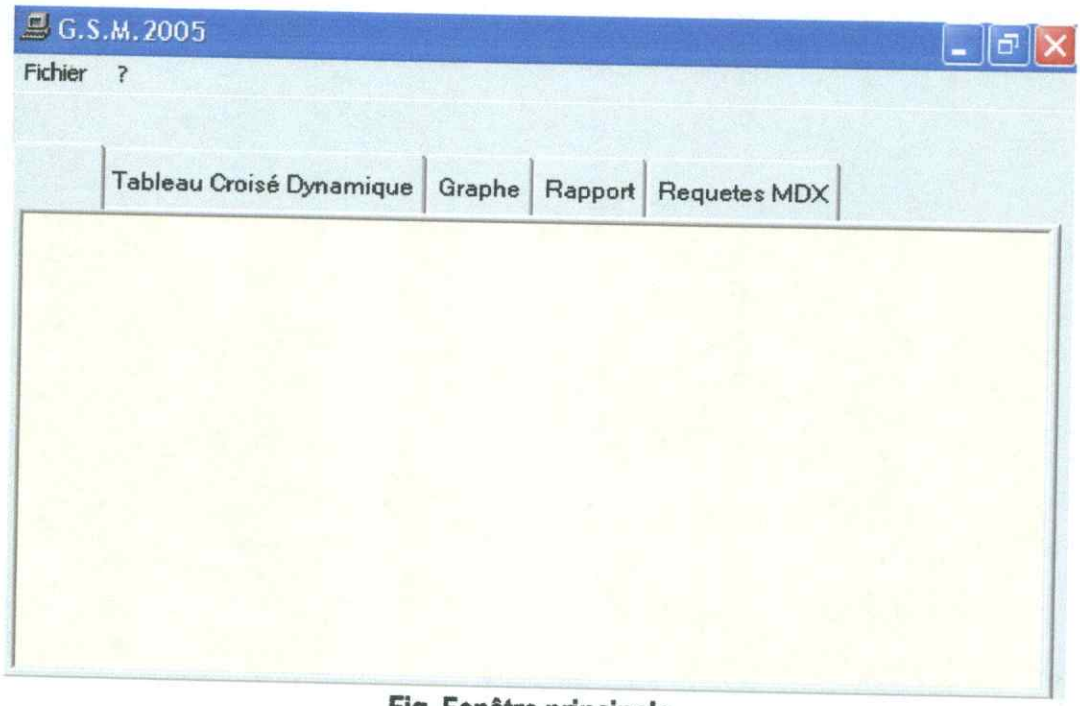


Fig. Fenêtre principale.

La fenêtre principale est constitue des éléments suivants :

- **Une barre de Menu** : qui elle aussi contient
  - ✓ Le menu **Fichier** comportant :
    - -La commande **Ouvrir une Session.**
    - -La commande **Quitter.**
  - ✓ Le menu **? (Aide)** comportant :
    - **- Aide**
    - **- A propos de G.S.M 2005**
- **Une barre d'Outils constitué d'un Ensemble d'Onglets**
  - ✓ Onglet Tableau Croisé Dynamique.
  - ✓ Graphe.
  - ✓ Rapport.
  - ✓ Requêtes MDX.

Explication de chaque partie de la fenêtre principale :

Le Menu ? contient les options suivantes :

Aide : permet d'afficher l'aide sur **G.S.M. 2005.**

A propos de : affiche la fenêtre A propos de **G.S.M. 2005.**

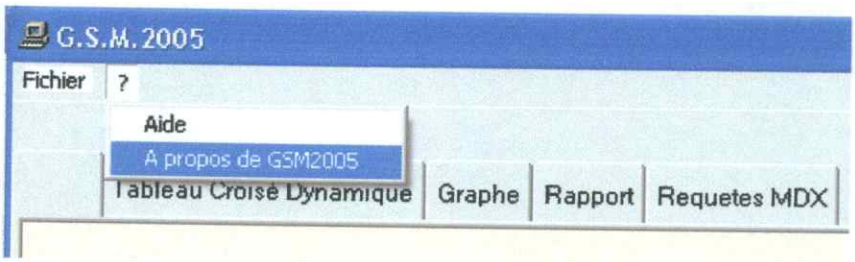


Fig. Menu « ? ».

Vu que la fenêtre principale demeure avec un ensemble de fonctionnalités réduites tant que l'utilisateur n'a pas ouvert une session de travail. L'utilisateur doit alors ouvrir une session afin d'accéder à toutes les fonctionnalités comme présenté dans la figure ci-dessous :

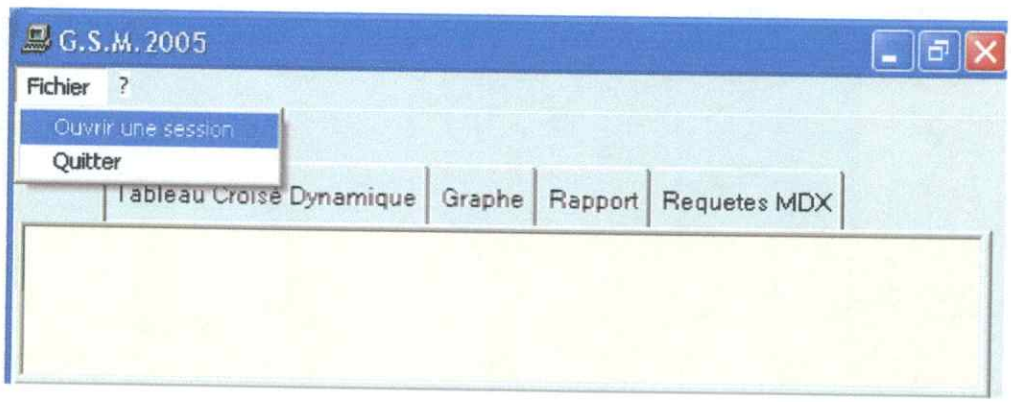


Fig. Menu Fichier.

Une fois que l'utilisateur lance l'ouverture de session, une fenêtre de Contrôle apparaît et invite l'utilisateur à saisir le nom et le mot de passe requis,comme est montré dans la figure ci-après :

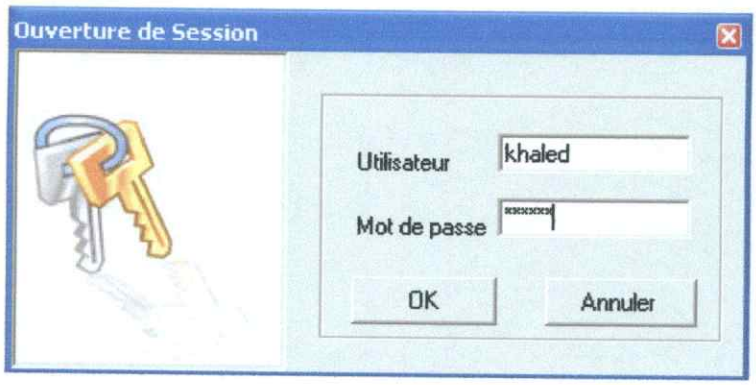


Fig. Fenêtre ouverture de session.

Cette fenêtre est constituée de :

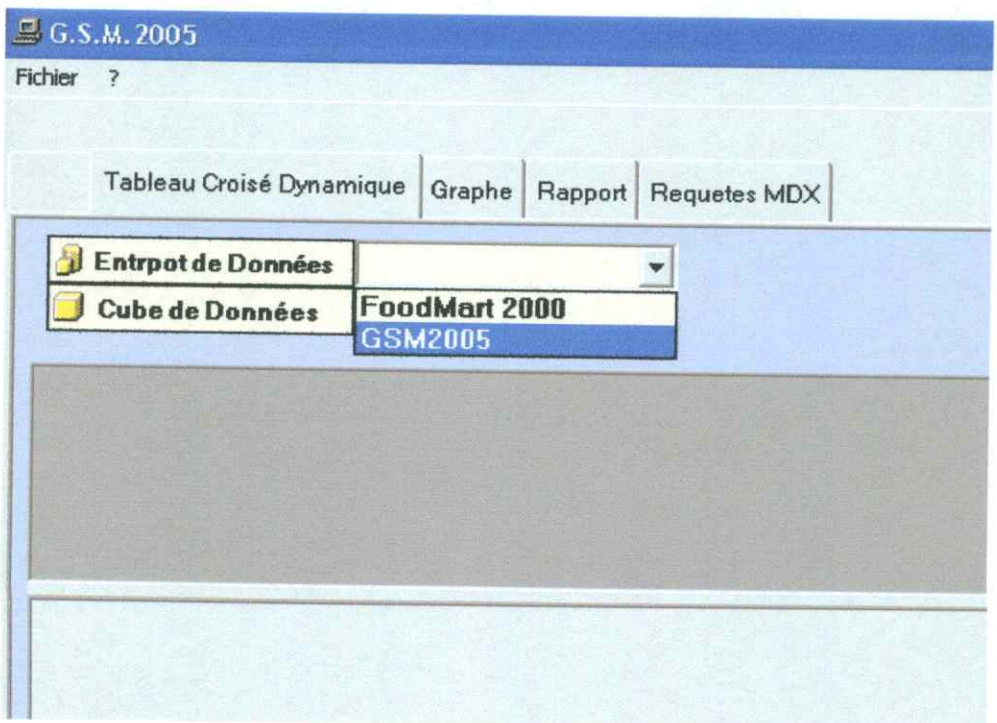
- **La zone de texte utilisateur**, permet de saisir l'identification de l'utilisateur.
- **La zone de texte mot de passe**, permet de saisir le mot de passe.
- **La commande ok**, pour valider la connexion et accéder a la fenêtre principale.
- **La commande Annuler**, pour annuler l'ouverture de session.

La barre d'outil : elle contient

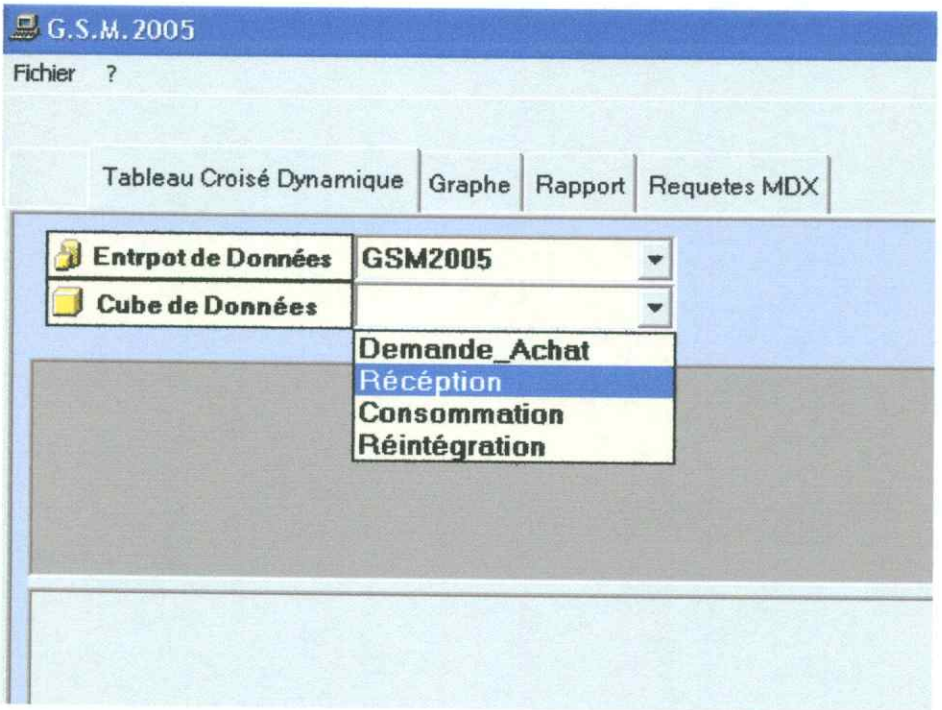
Onglet Tableau Croisé Dynamique : permet d'analyser les données sous forme de tableau.

Avant tout autre chose il faut d'abord :

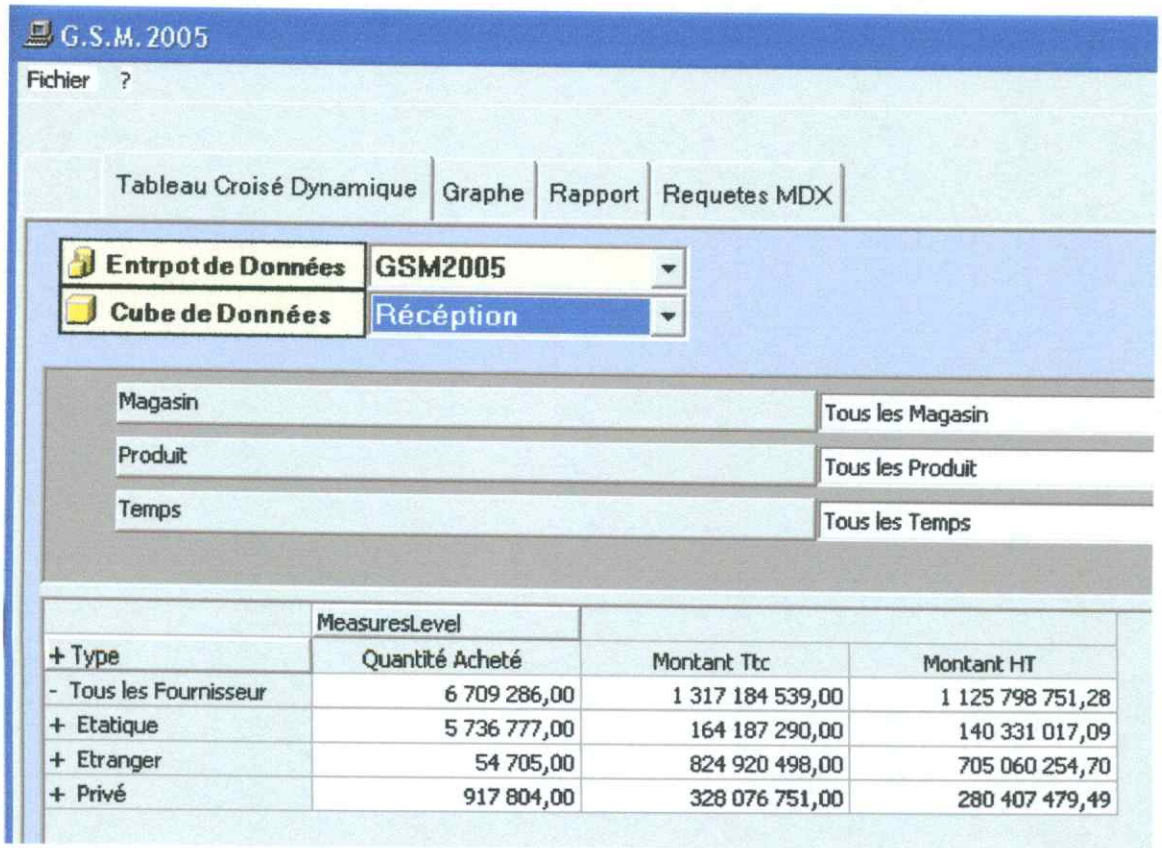
1. **choisir un entrepôt de données**



2. **choisir un cube de données :**

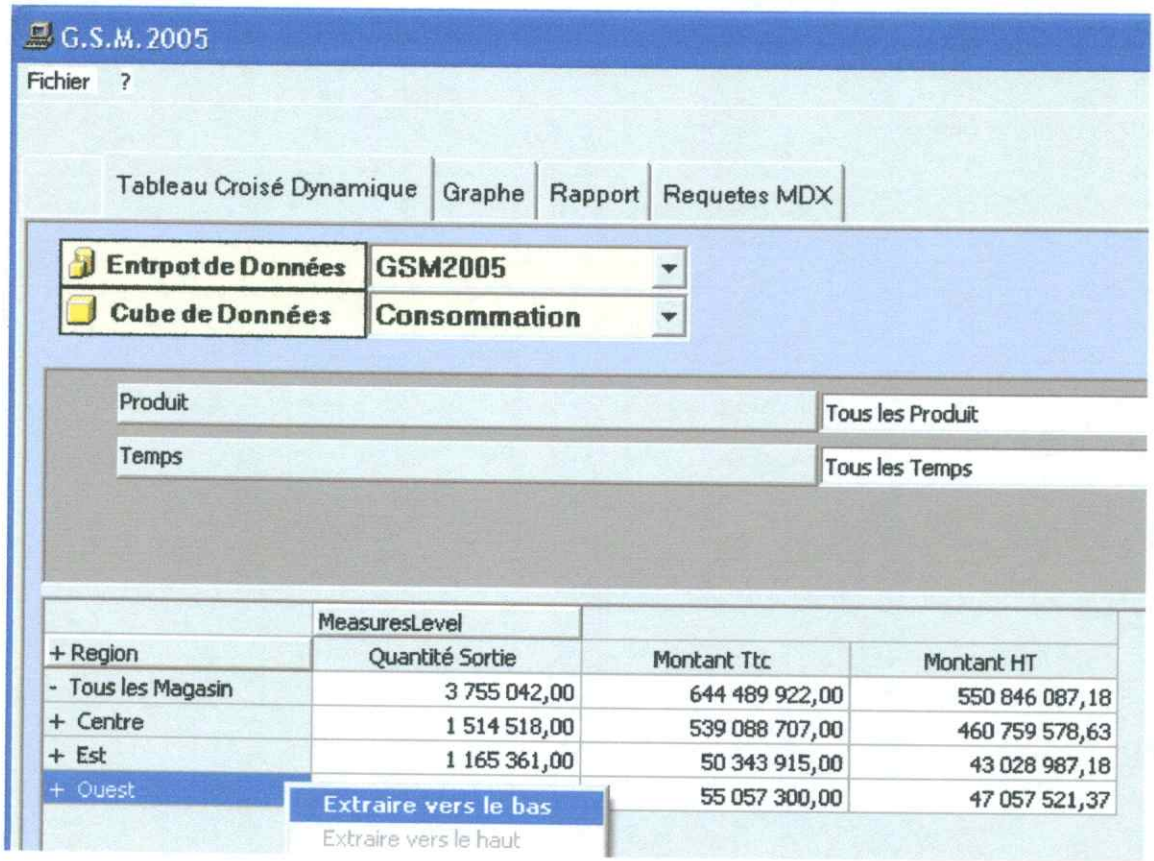


La fenêtre d'exploration de données apparaît avec toutes les données du cube Réception de la base G.S.M 2005. Ainsi la fenêtre ci-dessous contient les Quantités et Montant des produits achetés auprès des différents fournisseurs travaillant avec l'entreprise.

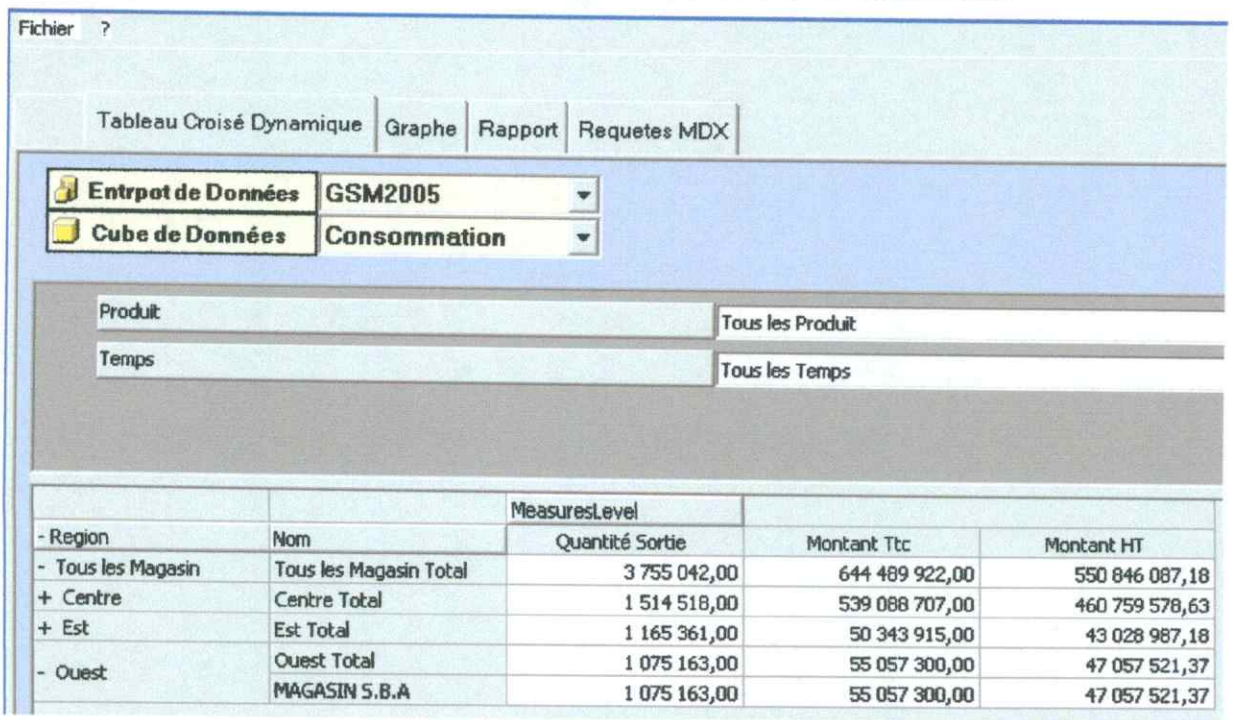


	MeasuresLevel		
+ Type	Quantité Acheté	Montant Ttc	Montant HT
- Tous les Fournisseur	6 709 286,00	1 317 184 539,00	1 125 798 751,28
+ Etatique	5 736 777,00	164 187 290,00	140 331 017,09
+ Etranger	54 705,00	824 920 498,00	705 060 254,70
+ Privé	917 804,00	328 076 751,00	280 407 479,49

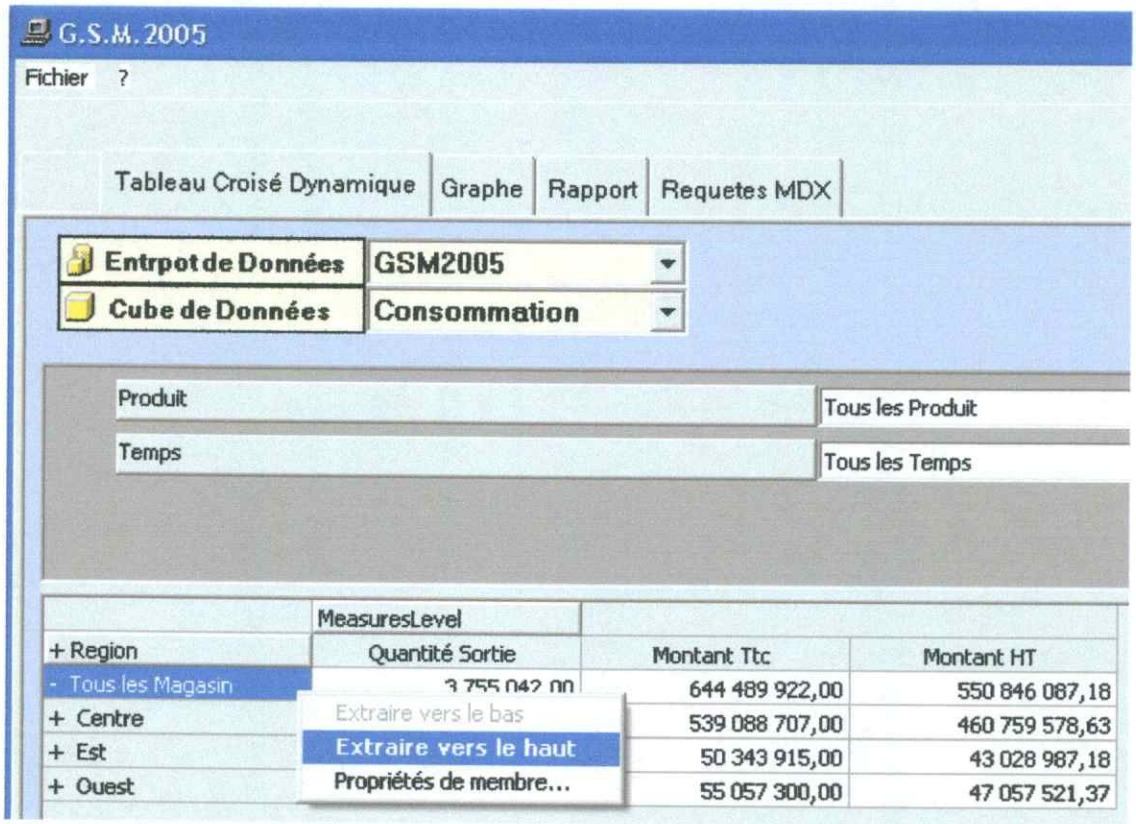
Cependant le tableau suivant représente les données du cube Consommation et illustre les Quantités Sorties ainsi que le montant correspondant de chaque produit et cela dans chaque magasin de l'entreprise.



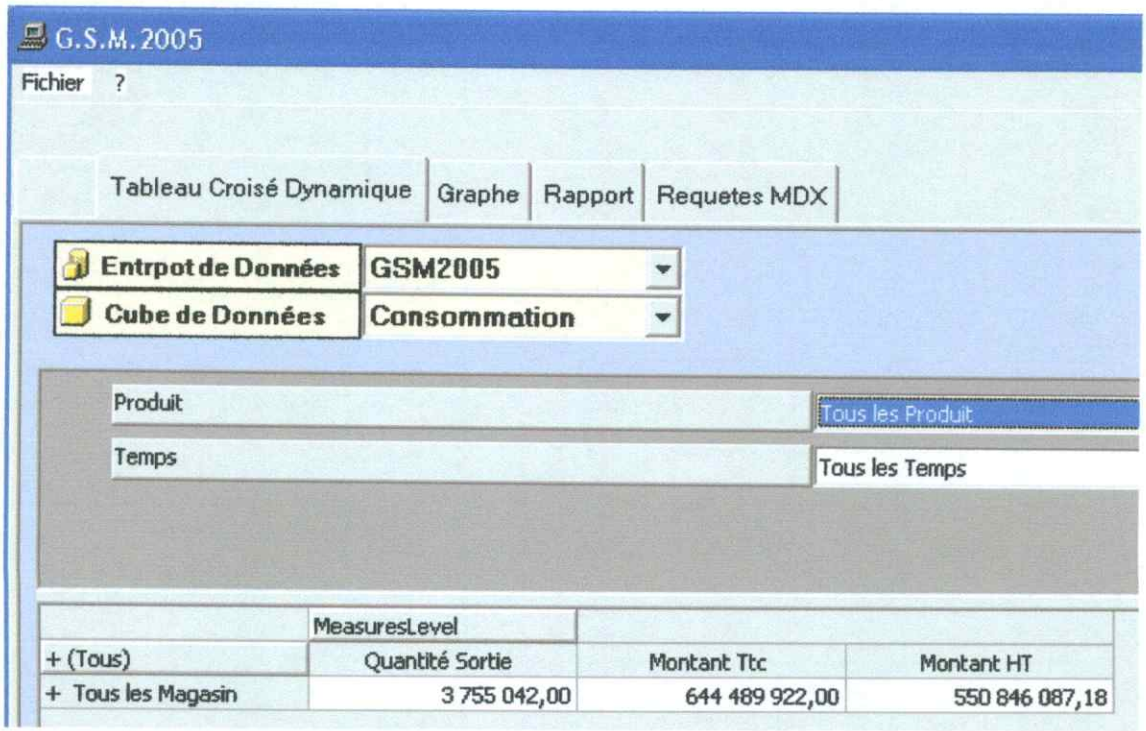
En appliquant quelques technique de navigation au tableau précédant, entre autre, le Forage vers le bas (drill_down) sur le niveau Ouest de la dimension magasin. Et on aura le Résultat suivant :



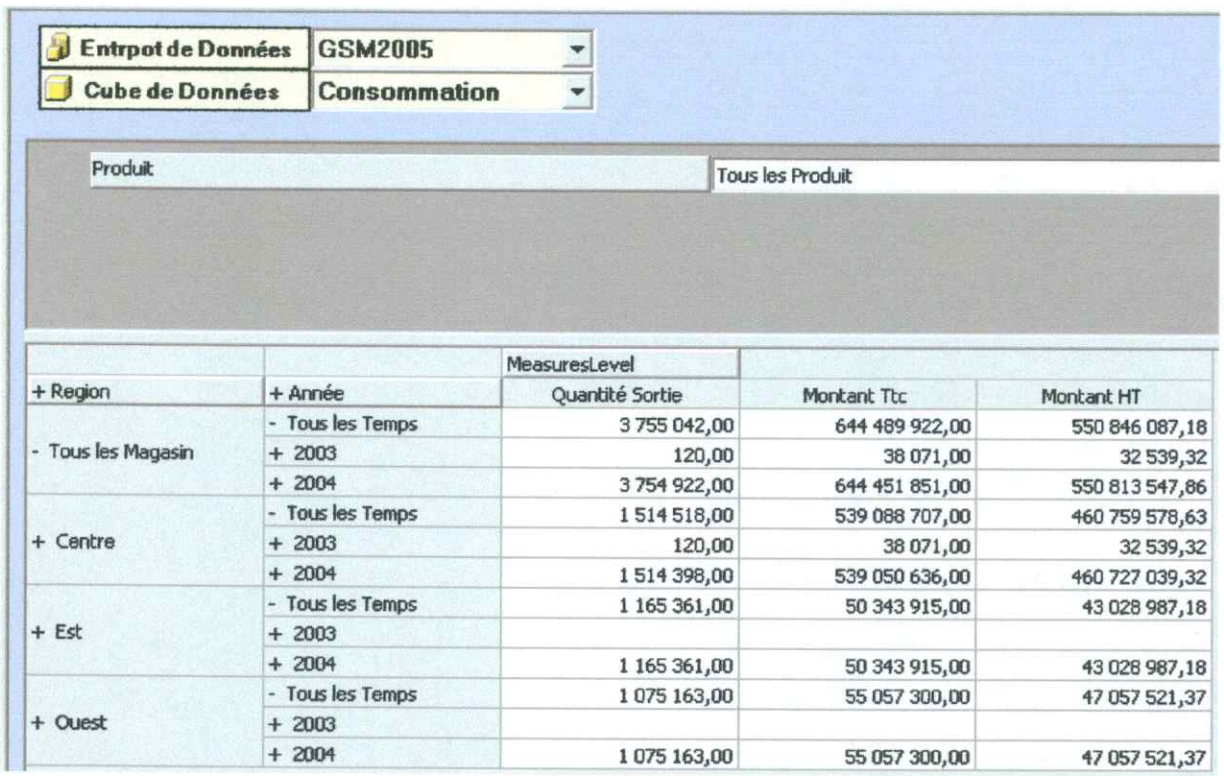
De la même manière nous pouvons regrouper touts les niveaux de la dimension magasin et appliqué à cet effet le Forage vers le haut (**drill_up**)



Et nous aurons le Résultat suivant :

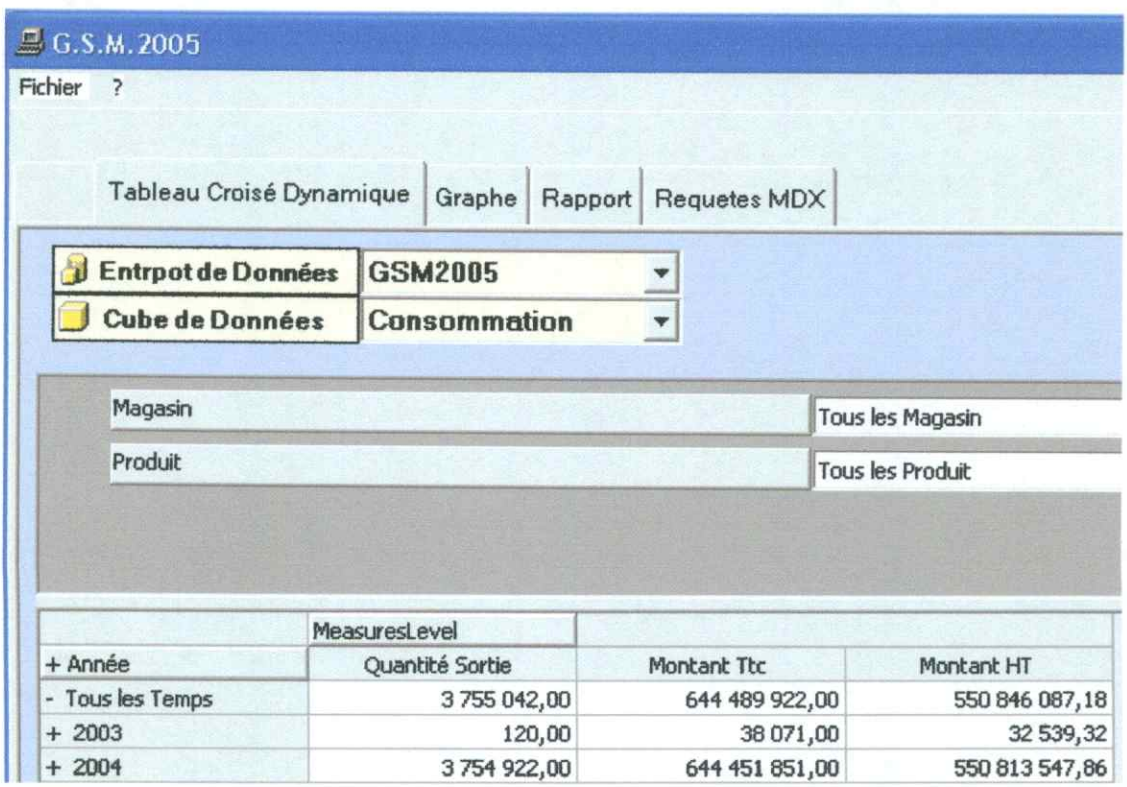


Nous pouvons également glisser de nouvelle Dimension qui servira comme de nouvel axe d'analyse. Ainsi la dimension Temps glissé nous permet, cette fois-ci d'analyser les Quantités et les Montants correspondant de chaque produit, dans chaque magasin et cela par rapport à une tranche de Temps (Jour, Mois, Trimestre, année) !!! .



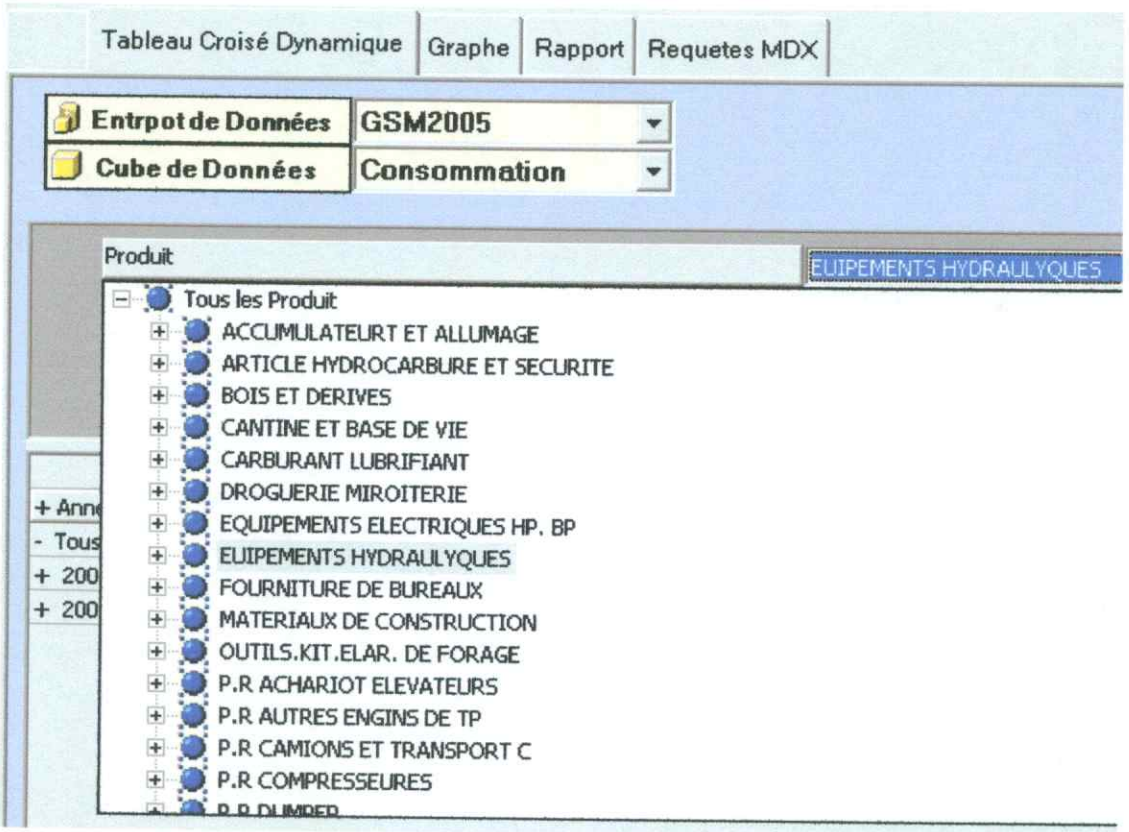
		MeasuresLevel		
+ Region	+ Année	Quantité Sortie	Montant Ttc	Montant HT
- Tous les Magasin	- Tous les Temps	3 755 042,00	644 489 922,00	550 846 087,18
	+ 2003	120,00	38 071,00	32 539,32
	+ 2004	3 754 922,00	644 451 851,00	550 813 547,86
+ Centre	- Tous les Temps	1 514 518,00	539 088 707,00	460 759 578,63
	+ 2003	120,00	38 071,00	32 539,32
	+ 2004	1 514 398,00	539 050 636,00	460 727 039,32
+ Est	- Tous les Temps	1 165 361,00	50 343 915,00	43 028 987,18
	+ 2003			
	+ 2004	1 165 361,00	50 343 915,00	43 028 987,18
+ Ouest	- Tous les Temps	1 075 163,00	55 057 300,00	47 057 521,37
	+ 2003			
	+ 2004	1 075 163,00	55 057 300,00	47 057 521,37

Comme nous pouvons également jouer sur les dimensions et les Permuter à volonté :

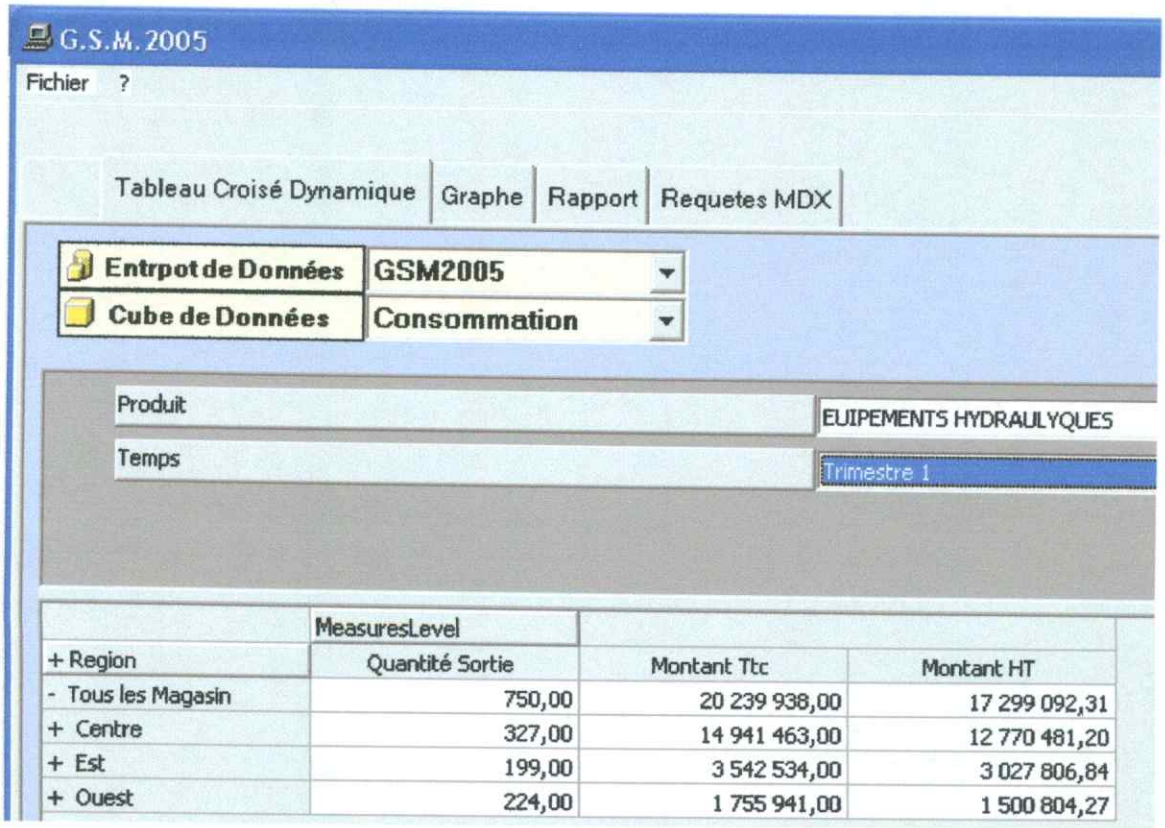


	MeasuresLevel		
+ Année	Quantité Sortie	Montant Ttc	Montant HT
- Tous les Temps	3 755 042,00	644 489 922,00	550 846 087,18
+ 2003	120,00	38 071,00	32 539,32
+ 2004	3 754 922,00	644 451 851,00	550 813 547,86

Nous pouvons aussi, affiner notre analyse en choisissant une famille de produit uniquement



Et les résultats suivants montrent les consommation de la Famille de Produit « Equipement Hydraulique » de touts les magasins durant le trimestre 1 de l'année 2004.

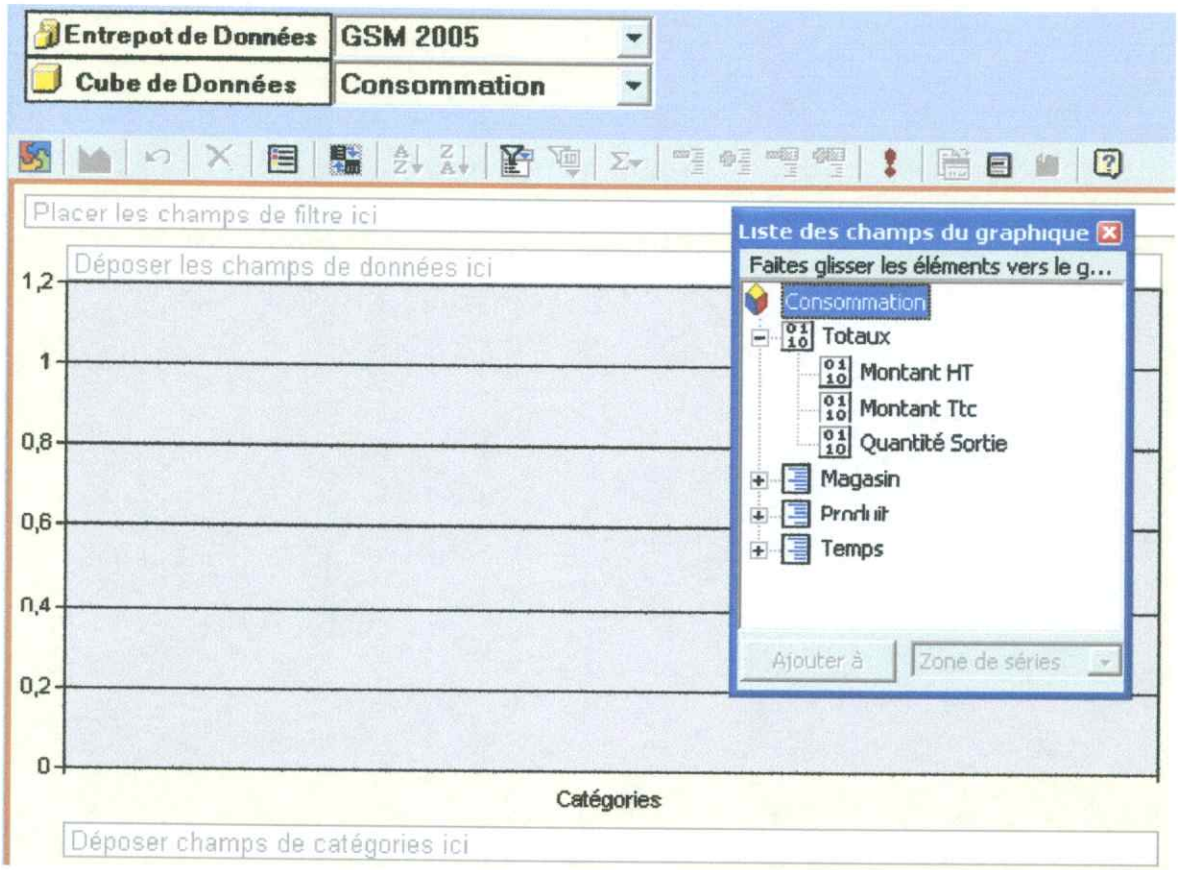


	MeasuresLevel		
+ Region	Quantité Sortie	Montant Ttc	Montant HT
- Tous les Magasin	750,00	20 239 938,00	17 299 092,31
+ Centre	327,00	14 941 463,00	12 770 481,20
+ Est	199,00	3 542 534,00	3 027 806,84
+ Ouest	224,00	1 755 941,00	1 500 804,27

Onglet Graphe : permet d'analyser les données sous forme graphique.

1. choisir un entrepôt de données
2. Choisir le cube
3. déplacer les dimension et mesures vers les champs

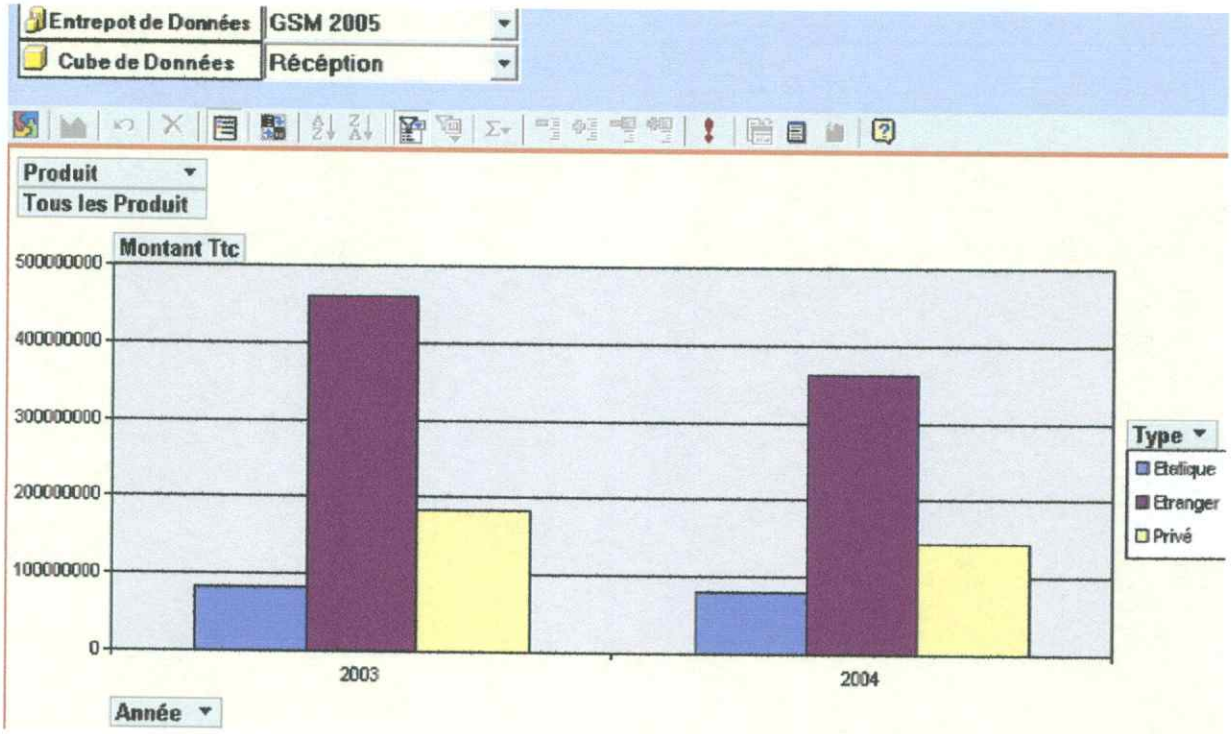
A partir de l'icône comportant la liste de champs graphique faites glisser les mesures ainsi que les dimension vers leurs champs, selon notre choix, dans le tableau de l'onglet graphe afin d'obtenir une représentation et pouvoir l'analysé.



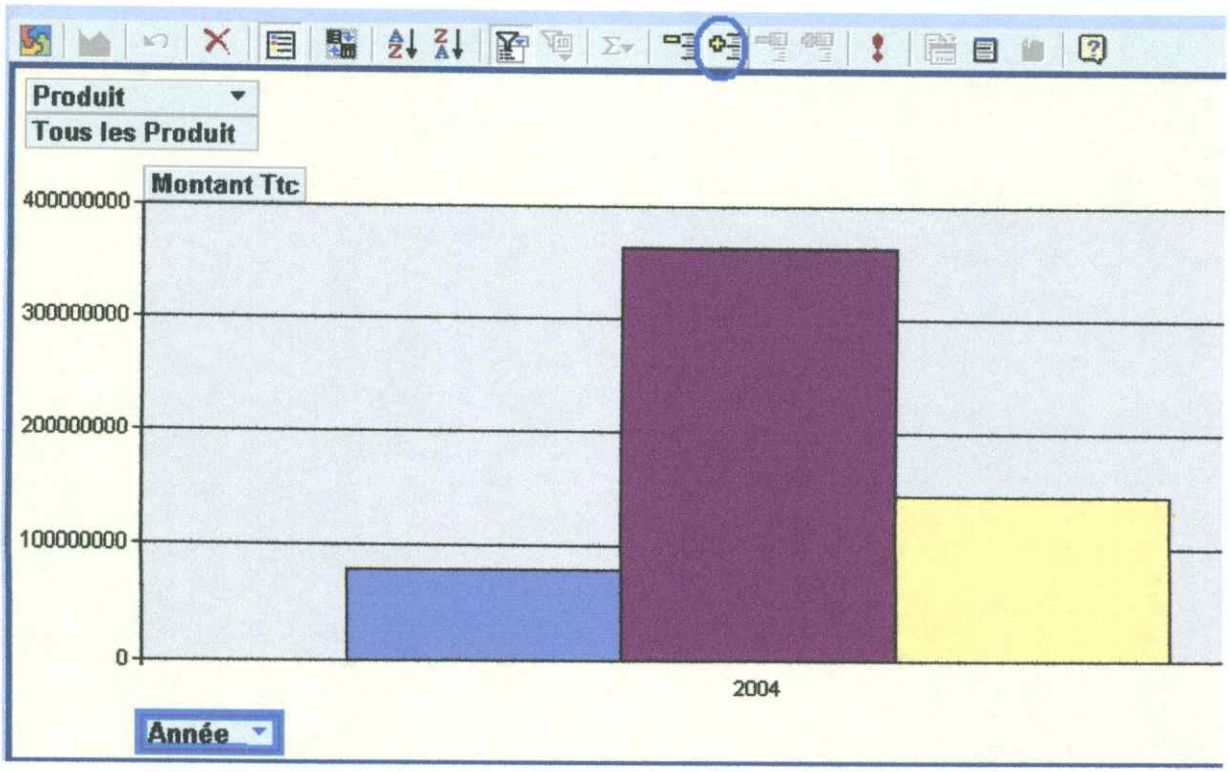
Regardons par exemple la figure ci-après qui représente les données du cube Réception.
Après avoir choisi les perspectives d'analyse comme Produit, Fournisseur, Temps, l'entreprise peux puiser des réponses aux préoccupations suivante telles que :

- Auprès de quel fournisseur l'entreprise effectue d'importants achats ?
- Suivre la trace des fournisseurs au cours du temps (2003-2004) ?

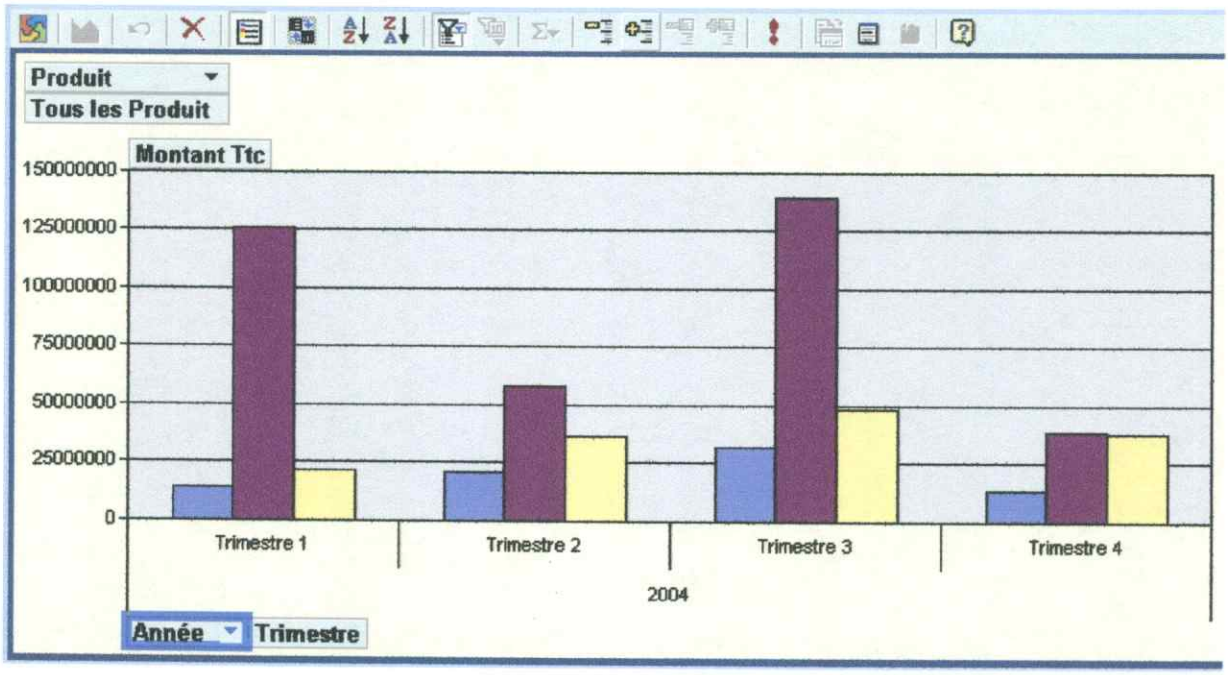
Résultat d'Analyse de tous les produits fournis au cours de l'année 2003-2004 par les trois types de fournisseurs (Etatique, Etranger, Privé).



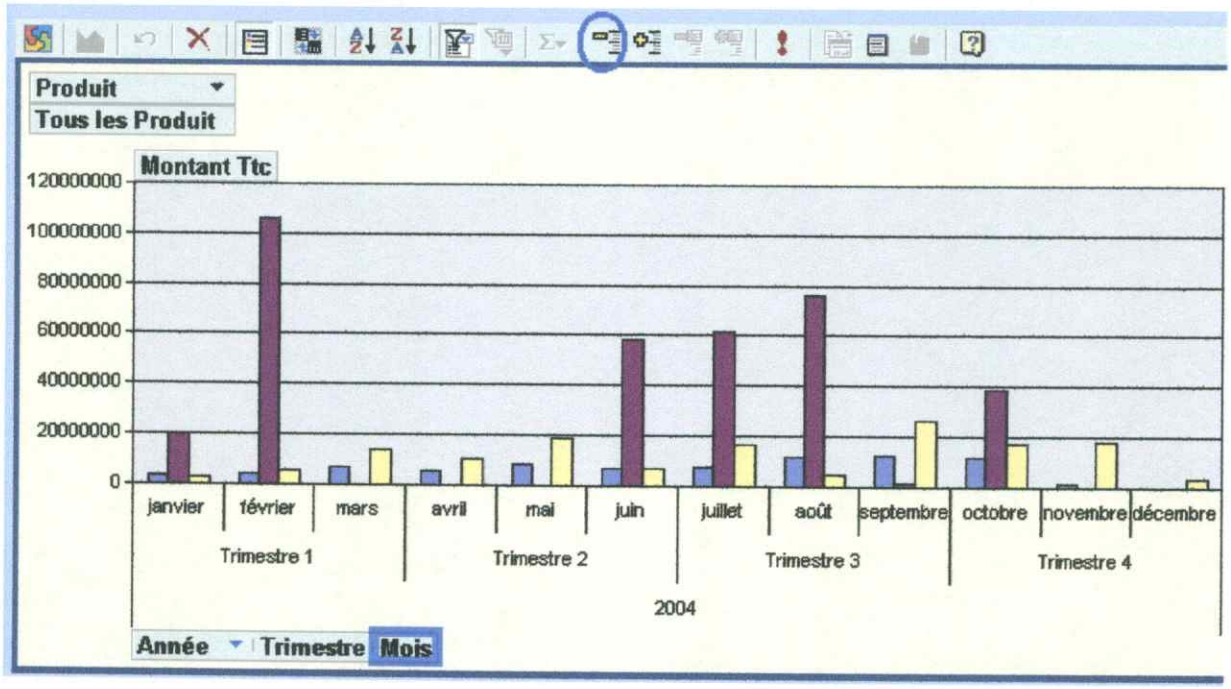
Maintenant concentrons nous uniquement sur l'exercice de l'année 2004, pour affiner notre analyse nous essayons d'appliquer quelques techniques de navigation à savoir le **« Forage vers le bas »**.



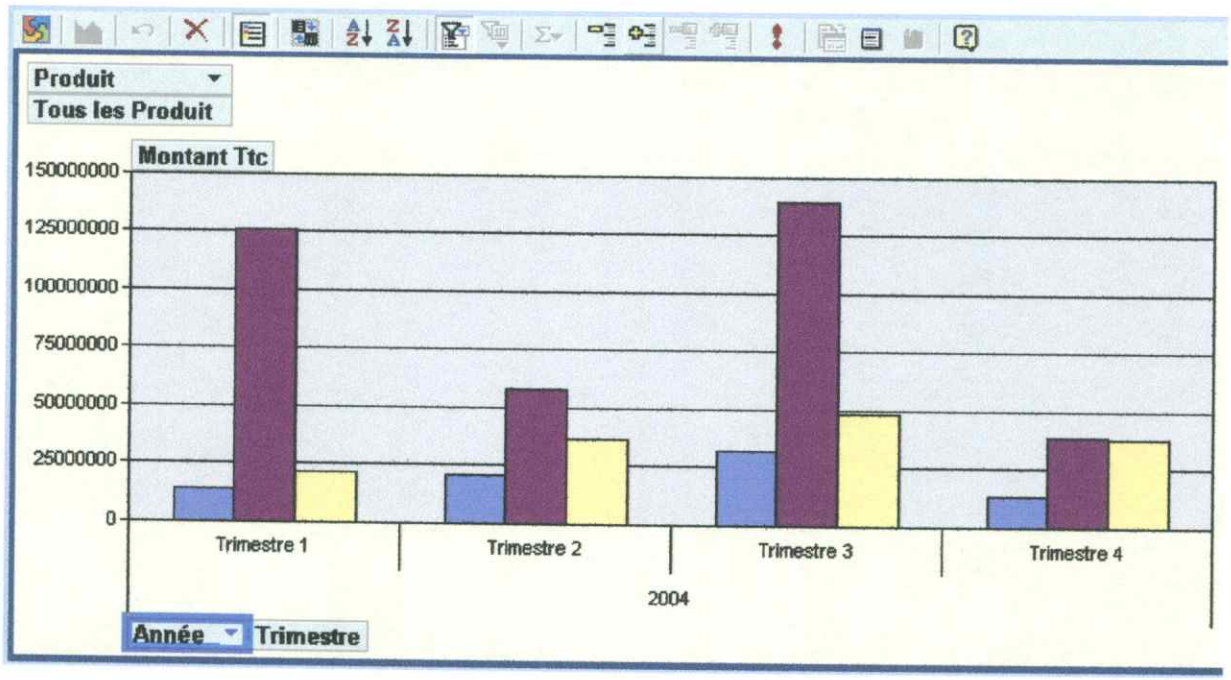
Nous obtenons ainsi le résultat suivant qui est de permettre une représentation à un niveau de détail plus fin des données. Donc des données trimestrielles au lieu annuelles.



De même pour un « **Forage vers le haut** » de la représentation graphique suivante.

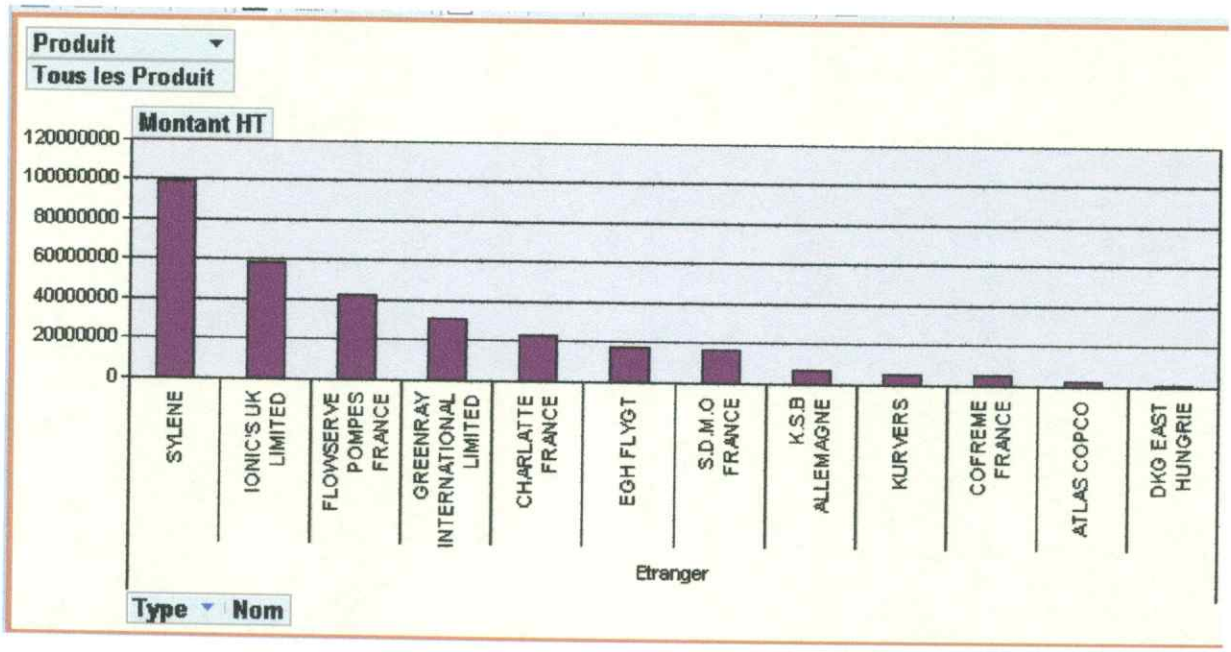


Le **Forage vers le haut** a permet de consolider les mois en trimestre comme c'est le cas dans la figure ci-dessous :

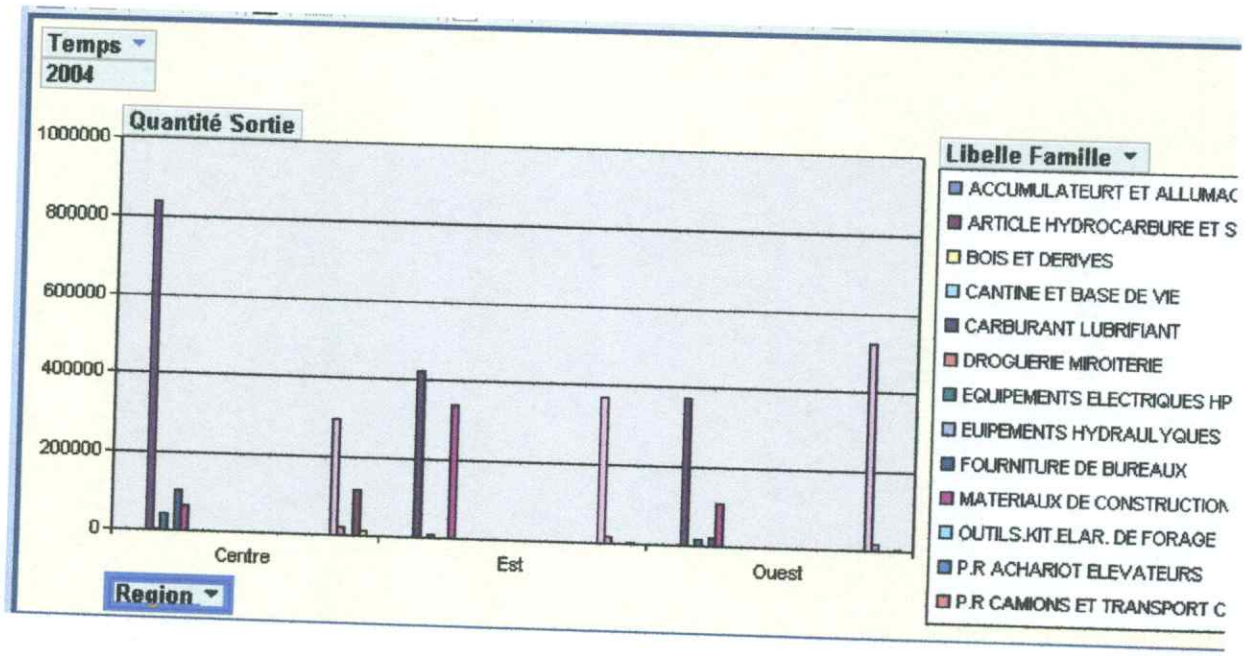


Exemple d'analyse :

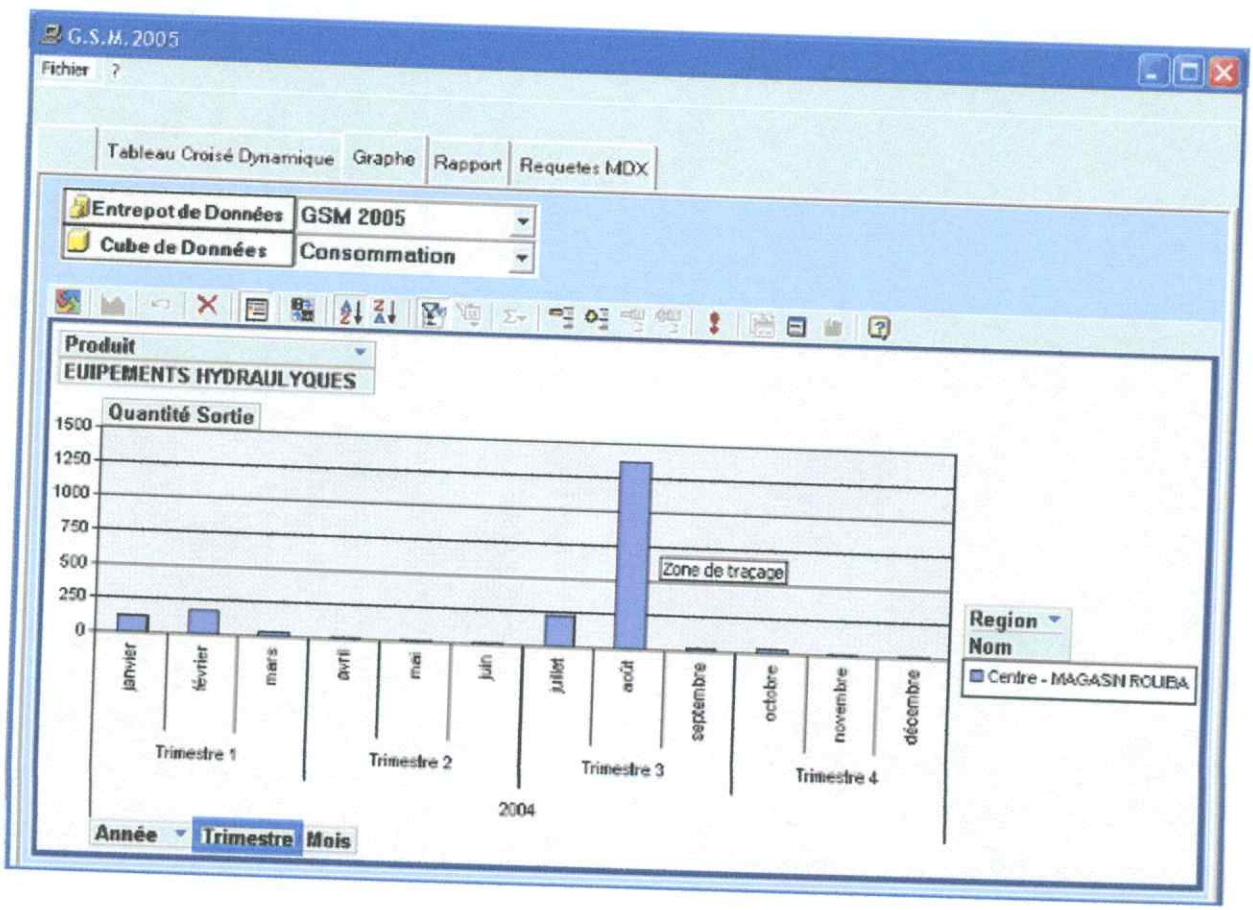
- **Les fournisseurs étrangers les plus performants, pour l'année 2004**



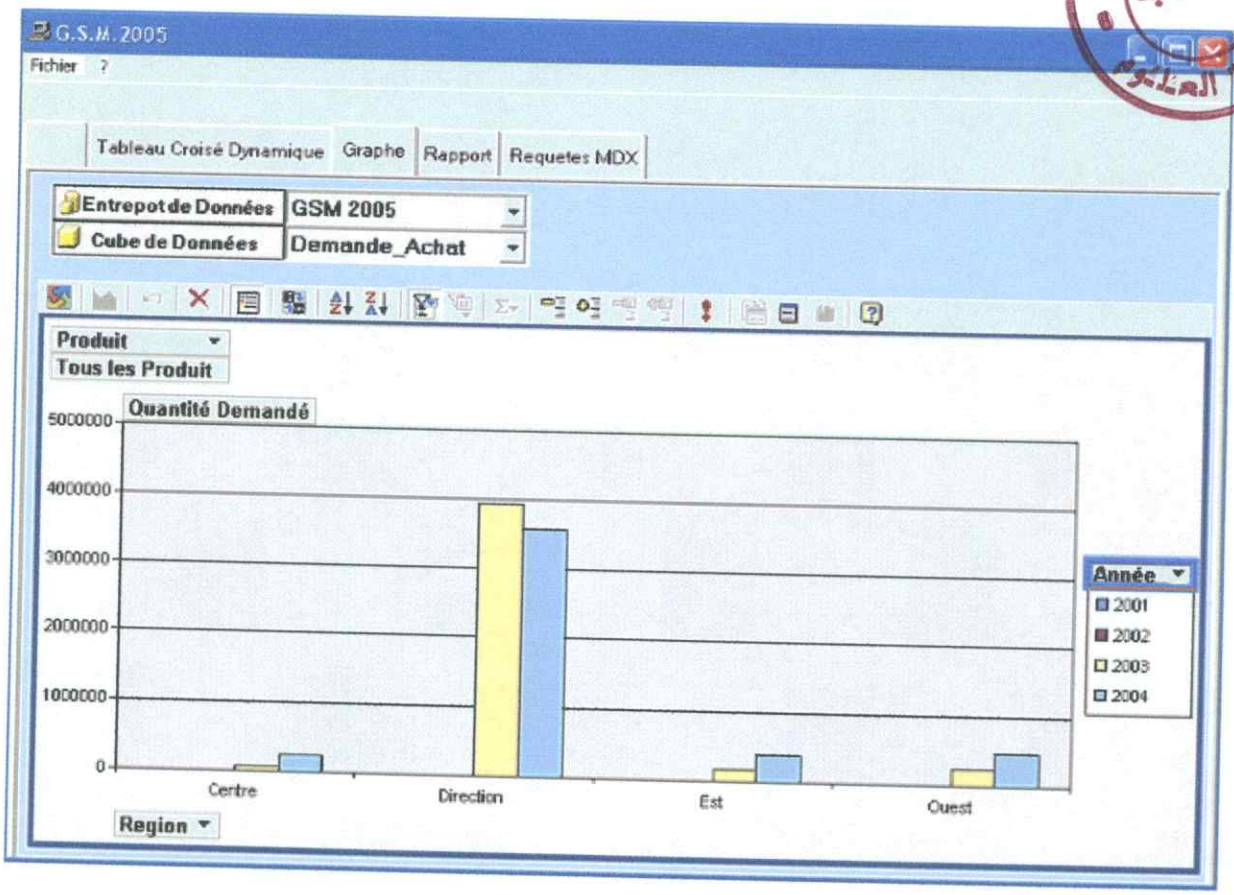
➢ **Les produits les plus consommés dans chaque magasin pour l'année 2004**



➢ **Suivre la trace de consommation des produit « Equipements Hydrauliques »**



➢ **Les demandes d'achat par région, année**



Nous pouvons ainsi effectuer toute sorte d'analyse en choisissant nos axes d'intérêt et appliquant les diverses techniques de navigations. Tout cela en très peu de temps, sans faire ni recours aux requêtes fastidieuse et complexes ni pénalisé les usagés de la base de données de production.

Glossaire

Agrégation Action de calculer les valeurs associées aux positions parents des dimensions hiérarchiques. Cette agrégation peut être une somme, une moyenne ou tout autre processus plus complexe comme la deuxième plus forte valeur.

Attribut Un fait décrivant chaque position d'une dimension.

Cellule Une donnée définie par une position de chaque dimension. Les cellules d'un hypercube peuvent être vides ou remplies. Lorsqu'un grand nombre de cellules sont vides, on parle de données éparses.

Datamart L'ensemble des données se rapportant à un des métiers de l'entreprise. Plusieurs datamart forment le data warehouse de l'entreprise.

Data Mining Définition un peu floue car récupérée par beaucoup d'éditeurs d'outils d'aide à la décision. A l'origine, le data mining correspondait à toutes les technologies avancées susceptibles d'analyser l'information d'un Data Warehouse pour en tirer des tendances, pour segmenter l'informations, ou pour trouver des corrélations dans les données. Aujourd'hui, le terme a tendance à caractériser tous les outils d'aide à la décision, le " mineur " étant soit l'outil lui-même soit l'utilisateur.

Data Warehouse « Entrepôt de données ». Base de données spécifique au monde décisionnel et destiné principalement à analyser les leviers « business » potentiels. D'après Bill Inmon, un Data Warehouse est intégré, orienté sujet et contient des données non volatiles et historisées

Data Warehousing Processus de mise en oeuvre d'un projet de Data Warehouse.

Dimension Axe d'analyse associé aux *indicateurs* ; correspond le plus souvent aux sujets d'intérêts du *Data Warehouse* ; exemple : dimension temporelle, dimension client...

Données creuses Dans une structure multidimensionnelle, les *données creuses* sont des intersections de dimensions pour lesquels un fait ne s'est pas produit (exemple : pas de vente de produit X à la date T) ou n'est pas physiquement stocké (exemple : pas d'agrégations physique associée aux vente de produits par gammes et par mois).

DSS Decision Support System ou système d'information décisionnel. C'est un système d'interrogation et de présentation des données adapté pour l'aide à la décision. Le terme français équivalent est SIAD ou Système d'Information d'Aide à la Décision. Un autre terme anglais est EIS ou Executive Information System.

EIS Executive Information Systems (littéralement, système d'information des cadres). Environnement de présentation de tableau de bord présentant de manière synthétique et graphiques les performances d'une activité (ex : santé d'une entreprise, bilan des ventes...).

Fait Donnée numérique servant de base à la définition des indicateurs dans un modèle multidimensionnel. Attention, ce terme est parfois utilisé dans la littérature pour décrire plus généralement tout indicateur.

FASMI Fast Analysis of Shared Multidimensional Information ou analyse rapide d'information multidimensionnelle partagée. Ces cinq termes ont tous leur importance dans la définition de la technologie OLAP.

Hiérarchie Les positions d'une dimension organisées selon une série de relations 1-n en cascade. Cette organisation de données est comparable à un arbre logique ou chaque membre n'a pas plus d'un père mais un nombre quelconque d'enfants.

Hypercube Cube à n dimensions. Structure sur laquelle repose la plupart des outils multidimensionnels.

Indicateur Information permettant de mesurer la performance de telle ou telle activité de l'entreprise (ventes, gestion des stocks...). La plupart du temps, cette information est numérique (ex : chiffre d'affaires, quantité en stock...).

Jointure Rapprochement entre deux tables par comparaison de valeurs communes, sur la base d'un attribut commun.

Méta-données Donnée décrivant une donnée

Middleware Le logiciel du centre : infrastructure logicielle permettant de rendre l'architecture Client/Serveur la plus transparente possible. Dans un contexte décisionnel, il est situé entre les outils d'aide à la décision et la base de données décisionnelle. Un bon middleware permet d'avoir indépendance entre ces deux types de composants

Modèle dimensionnel (ou multidimensionnel) Technique de modélisation consistant à modéliser une base décisionnelle à partir de l'identification des *faits* à analyser et des *dimensions* d'analyses qui leur sont associées

Modèle en étoile Technique de modélisation dimensionnelle, consistant à créer physiquement ce modèle sur une base de données relationnelle. Ce modèle distingue physiquement les tables de faits des tables de dimensions

Modèle en flocon (ou snowflake) Technique de modélisation dimensionnelle, dérivée de la *modélisation en étoile*. Dans ce modèle, les tables de dimensions y sont dénormalisées, c'est à dire dénuées de redondances.

Modèle relationnel Technique de modélisation consistant à modéliser une base de données en la décomposant en entité et en relations corrélant ces entités

MOLAP Caractérise l'architecture nécessaire à la mise en place d'un système multidimensionnel en s'appuyant sur les bases de données multidimensionnelles

OLAP Caractérise l'architecture nécessaire à la mise en place d'un système d'information décisionnel. S'oppose à OLTP (On Line Transaction Processing), adressant les systèmes d'information transactionnels. OLAP est souvent utilisé pour faire référence exclusivement aux bases de données multidimensionnelles. En effet, le concept a été formalisé par le Dr Codd, sous la forme de douze règles, décrivant un modèle idéal d'analyse d'information. Il a été montré depuis qu'il a été possible de respecter ces règles indépendamment de la structure de stockage utilisée. De plus en plus, le terme est souvent utilisé pour désigner plus généralement le décisionnel dans ses aspects techniques.

Optimiseur L'optimisation des questions est un aspect tout à fait central des systèmes relationnels; La requête SQL est transformée en une succession d'opérateurs relationnels (restriction, projection, jointure, union, ...); L'optimiseur est un composant logiciel chargé de choisir l'ordre dans lequel ces opérations vont être exécutées.

Passerelle Matériel d'interconnexion de réseaux locaux employant des protocoles de haut niveau différents. Logiciel de traduction situé sur un serveur et permettant à deux Applications d'interfaces différentes de dialoguer.

Requête C'est une demande envoyée au gestionnaire de Base de Données serveur. Si celui-ci permet la gestion des données, le langage utilisé est le SQL. Dans un contexte d'infocentre, l'exécution des questions sur un serveur est le plus souvent interprétée

ROLAP Caractérise l'architecture nécessaire à la mise en place d'un système multidimensionnel en s'appuyant sur les technologies relationnelles.

SGBDR Système de Gestion de Base de Données Relationnelle

SQL Langage de Requête Structuré. Le langage SQL est un standard défini par l'ANSI et l'ISO. Il est dérivé de l'algèbre relationnel et de SEQUEL (System R74). Il constitue aujourd'hui le plus petit commun dénominateur des langages du marché.

BIBLIOGRAPHIE

Livres :

- **[Fra97]** : J.Michel FRANCO et EDS-institut prométheus, « Le data warehouse et le data mining », Eyrolles, 1997.
- **[Fra00]** : Jean-Michel Franco et Sandrine de Lignerolles, « Piloter l'entreprise grâce au data warehouse », Eyrolles 2000.
- **[Kim97]** : Ralph KIMBALL, « Entrepôts de données », International Thomson Publishing France, 1997.
- **[Kim02]** : Ralph KIMBALL et Margy Ross « Entrepôts de données », Deuxième édition,Vuilbert, Paris,2003(Traduction de Claude Raimond).
- **[Kim03]** : Ralph KIMBALL et Margy Ross, « Guide Pratique de Modélisation Dimensionnelle ». Edition Vuibert, Paris 2003. Traduction de Claude Raimond.
- **[Gou98]** : J.Marie GOUARNE, « Le projet décisionnel, Enjeux, Modèles architecture du data warehouse », Eyrolles, 1998
- **[Gog98]** : Jean-francois Goglin ,« La construction du data warehouse », Edition HERMES 1998
- **[Gar00]** : Georges GARDARIN, « Internet /intranet et bases de données data web, data media, data warehouse, data mining », Eyrolles, 2000.
- **[Kev02]** : Kevin VIERS, « Introduction au Data Warehousing ». LE COMPUS, 2002.
- **[Sup01]**:Support de Microsoft «Programmation SQL (My SQL & SQL Server) », MICRO-APPLICATION, 2001.

Mémoires et thèses :

- **[AS04]** : Ammar RAMDANE-CHERIF & Layla SID, « Conception et Réalisation d'un Data Warehouse Métier Et d'un Outil Web d'Analyse ». mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en informatique, INI ,2004.
- **[BR04]** : BENZARGA Mehdi & RAMDANI Farid, « Conception et Réalisation d'une Solution Décisionnelle pour la Fonction Vente Basée sur une Architecture Data Warehouse », mémoire de fin d'étude d'ingénieur d'état en informatique, USTHB ,2004.

- **[DA04] :** A.Djellah & B.Allouche, « Elaboration d'un tableau de bord prospectif », mémoire de fin d'étude d'ingénieur d'état en informatique, USTHB ,2004.
- **[Bol02] :** Nathalie RYSER BOLOGNINI, Etude pour la création d'un entrepôt de données dans le cadre de l'assurance vie et transformation des données en informations utiles en vue d'une prise de décision, Mémoire en vue de l'obtention du Diplôme post grade en informatique et organisation, Université de LAUSANNE, 2000-2002.
- **[Tes00]:** Olivier TESTE, 'Modélisation et manipulation d'entrepôts de données complexes et historisées', thèse en vue de l'obtention du doctorat en informatique, université Paul SABATIER (Toulouse), 2000.
- **[Nie98]** Philippe Nieuwbourg avec la participation active de l'équipe Business Intelligence de Microsoft. "Quel système décisionnel pour les entreprises agiles "

Sites :

- **[W.01] :** www.nordnet.fr/dnakache/valeur
- **[W.02] :** www.decisionnel.net/datawarehouse/dwh.htm
- **[W.03] :** www.grappa.univ-lille3.fr
- **[W.04] :** www.irit.fr/recherche/IRI/SIG/personnes/these/these.pdf
- **[W.05] :** www.prometheus.eds.fr
- **[W.06] :** www.1keydata.com/datawarehousing/datawarehouse.html
- **[W.07] :** www.dwinfocenter.org
- **[W.08] :** www.datawarehousingonline.com
- **[W.09] :** www.dw-institute.com
- **[W.10] :** www.rkimball.com
- **[W.11] :** www.billimmon.com
- **[W.13] :** www.univ-valenciennes.fr/limav/donsez/cours
- **[W.14] :** www.indexel.net/doc
- **[W.15] :** www.solutions.journaldunet.com/0301/030113_datawarehouse.shtml
- **[W.16] :** www.dbasupport.com
- **[W.17] :** www.guidecomparatif.com
- **[W.18] :** www.grd-publications.com/art/ls033/ls033014.htm
- **[W.19]:** www.rd.francetelecom.com/fr/conseil/mento14/chap6.html