Les Sélections
Techniques de l'Ingénieur

DOSSIER

TECHNIQUES DE L'INGÉNIEUR

l'expertise technique et scientifique de référence

h3870

Entrepôts de données

Date de publication : 10/02/2005

Par :

Claude CHRISMENT
Professeur à l'université Toulouse-3

Geneviève PUJOLLE
Maître de conférences à l'université Toulouse-1

Franck RAVAT
Maître de conférences à l'université Toulouse-1

Olivier TESTE
Maître de conférences à l'université Toulouse-3

Gilles ZURFLUH
Professeur à l'université Toulouse-1

Ce dossier fait partie de la base documentaire
Bases de données
dans le thème Technologies logicielles Architectures des systèmes
et dans l'univers Technologies de l'information

Pour toute question :
Service Relation Clientèle • Éditions Techniques de l'Ingénieur • 249, rue de Crimée
75019 Paris – France

par mail : infos.clients@teching.com ou au téléphone : 00 33 (0)1 53 35 20 20

Techniques
de l'Ingénieur

Entrepôts de données

par **Claude CHRISMENT**
Professeur à l'université Toulouse-3

Geneviève PUJOLLE
Maître de conférences à l'université Toulouse-1

Franck RAVAT
Maître de conférences à l'université Toulouse-1

Olivier TESTE
Maître de conférences à l'université Toulouse-3

et **Gilles ZURFLUH**
Professeur à l'université Toulouse-1

1.	**Bases de données décisionnelles**	H 3 870 — 2
1.1	Bases de données et système d'information	— 2
1.2	Des bases de données aux entrepôts de données	— 2
2.	**Architecture d'un système décisionnel**	— 3
3.	**Extraction des sources**	— 4
4.	**Constitution d'un entrepôt**	— 5
4.1	Analyse des besoins	— 5
4.2	Définition de l'entrepôt	— 6
4.3	Choix des outils d'analyse	— 6
5.	**Modélisation multidimensionnelle des magasins**	— 6
5.1	Nécessité de modèles adaptés	— 6
5.2	Étude des modèles par niveau d'abstraction	— 6
	5.2.1 Niveau conceptuel	— 7
	5.2.2 Niveau logique	— 7
	5.2.3 Niveau physique : Oracle 9i	— 8
6.	**Analyse décisionnelle**	— 10
6.1	Contexte	— 10
6.2	Outils de manipulation décisionnelle	— 10
	6.2.1 Requêteurs	— 10
	6.2.2 Tableurs	— 10
	6.2.3 Outils d'analyse OLAP	— 10
	6.2.4 Outils de fouille de données	— 11
6.3	Algèbre multidimensionnelle	— 11
	6.3.1 Table multidimensionnelle	— 11
	6.3.2 Opérateur de construction	— 12
	6.3.3 Opérateurs de transformation de la granularité des données	— 12
	6.3.4 Opérateurs de transformation de la structure des données	— 12
7.	**L'exemple d'Oracle**	— 14
Pour en savoir plus		Doc. H 3 870

Les entrepôts de données (ou « data warehouse ») sont des bases de données (BD) spécifiques utilisées par les applications d'aide à la décision.

La mise en place et l'exploitation d'un entrepôt au sein d'une entreprise suivent des processus particuliers, distincts des démarches utilisées pour l'élaboration des BD.

*En ce qui concerne l'**extraction des données**, les entrepôts sont alimentés à partir de sources de données diverses telles que des BD, des fichiers et des documents web. Il convient de s'assurer de la cohérence de l'ensemble de ces données et de permettre leur mise à jour régulière (rafraîchissement) en accord avec les besoins des décideurs.*

*La **structuration de l'entrepôt** doit être adaptée à l'usage que l'on en fait. Les modèles de données utilisés pour structurer et manipuler les BD classiques sont généralement inadaptés aux entrepôts ; de nouveaux modèles multidimensionnels ont été proposés pour offrir aux décideurs une représentation simple des données.*

*La **manipulation des données** d'un entrepôt s'effectue souvent au travers de logiciels d'analyse de données. C'est pourquoi les données doivent être sélectionnées selon certains critères ou certaines dimensions grâce à des opérateurs ad hoc qui les agrègent ou, au contraire, les répartissent selon les axes d'étude.*

*Enfin, l'**évolution de l'entrepôt** n'est pas uniquement liée aux extractions des données qu'il reçoit régulièrement des sources. Son schéma peut aussi être modifié au fil du temps pour s'adapter à l'évolution des processus d'analyse.*

1. Bases de données décisionnelles

La gestion et le pilotage des entreprises, dans le cadre d'une mondialisation croissante de l'économie, nécessitent des **systèmes d'information performants**. Les décideurs, quel que soit leur niveau de responsabilité, doivent pouvoir accéder aux informations qui leur sont utiles le plus rapidement possible. Les entrepôts de données tentent de répondre à cette nécessité.

1.1 Bases de données et système d'information

Au sein de l'entreprise, les bases de données contiennent l'information nécessaire aux applications de gestion de l'entreprise : données commerciales, données liées à la gestion du personnel ou à la gestion de la production. Ces bases de données, dites de **production**, ont été conçues et mises en place par les informaticiens pour répondre aux besoins des **applications transactionnelles** (paie du personnel, gestion des stocks, facturation, etc.).

Mais les bases de production, même si elles constituent le cœur du système d'information, ne contiennent qu'une partie de l'information utile aux décideurs. Ceux-ci souhaitent accéder aussi à l'ensemble des données situées dans l'environnement de l'entreprise au sens large, telles que :

— les informations fournies ou disponibles chez les partenaires (clients et fournisseurs principalement) ;

— les données (économiques, fiscales, etc.) produites par des administrations ou des organismes d'État ;

— les données stockées dans des sites web du réseau Internet (études de marché, prospectives ministérielles, etc.).

Les décideurs sont souvent démunis pour accéder simplement à ces données ou pour les manipuler et en extraire des informations pertinentes en raison :

— du volume considérable des données et de l'hétérogénéité des sites ;

— de la diversité des structures, des supports, des langages et des modes d'exploitation de ces données (BD relationnelles [H 2 038], BD objet [H 3 840], documents XML [H 7 148]) ;

— de la difficulté de traiter, manipuler ou mettre en corrélation (croiser) des données dans des formats hétérogènes.

Or, un système d'information doit pouvoir répondre à des requêtes du type « obtenir le chiffre d'affaires mensuel des hypermarchés du groupe par catégorie de produits pour les six derniers mois ». L'efficacité d'une décision peut être liée à la rapidité et à la fiabilité de la réponse fournie par le système.

1.2 Des bases de données aux entrepôts de données

Les bases de production, au sein d'une entreprise, sont alimentées par les applications informatiques transactionnelles (paye du personnel, gestion des stocks de produits, comptabilité). Leur structure (schéma) a été conçue pour satisfaire les exigences de ces applications ; le modèle de données relationnel [H 2 038] est le formalisme le plus utilisé par les informaticiens chargés de concevoir et mettre en œuvre ces bases de données.

Dès les années 1970, avec l'avènement massif des micro-ordinateurs dans les entreprises, des « infocentres » ont offert aux décideurs une interrogation directe des bases de données grâce aux langages de type SQL [H 3 128]. Mais les structures de données relationnelles s'avèrent inadaptées aux applications décisionnelles, rendant l'accès aux données fastidieux pour des non-informaticiens. Ajoutons à ce constat que le système d'information ne se limite pas aux bases de données internes mais intègre aussi les bases de données des partenaires et des sites web contenant des documents au

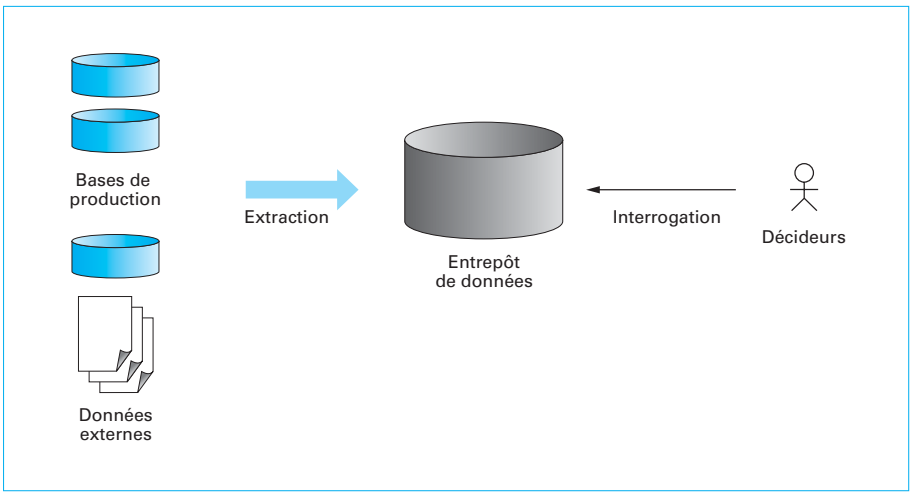


Figure 1 – Entrepôt de données : interface entre les sources d'information et les décideurs

format HTML ou XML. L'interrogation d'un système d'information par un décideur relève alors du parcours du combattant.

Dans les années 1980, face à la généralisation des réseaux, les informaticiens ont constitué des réseaux de bases de données qui ont permis d'exprimer des requêtes sur une BD « virtuelle » qui était en fait constituée d'un ensemble de BD (BD réparties [H 3 850], BD multibases). Mais la disponibilité des sources de données et la complexité des systèmes de gestion de bases de données (SGBD) répartis [H 2 918] sont un frein important à ce type d'approche.

Dès le début des années 1990, des entreprises ont adopté une autre approche : elles ont mis en place des entrepôts de données, c'est-à-dire des bases de données spécifiques entièrement dédiées aux décideurs. Ces bases de données, véritables **interfaces** entre les décideurs et les diverses sources de données (figure **1**), permettent un accès simplifié aux données en masquant l'hétérogénéité des sources. L'évolution des données contenues dans les sources, tant au niveau des valeurs qu'au niveau de leur structure, est répercutée régulièrement dans l'entrepôt.

Un entrepôt est une base de données thématiques extraite de diverses sources d'information et organisée de manière à faciliter l'accès des données aux décideurs.

Pour des besoins d'analyse, les données de l'entrepôt peuvent être **historisées** ; on enregistre alors l'évolution de leurs états successifs dans le temps. Mais cette historisation entraîne le stockage d'un volume important de données et nécessite de constituer des entrepôts par fonction ou par thème.

2. Architecture d'un système décisionnel

On considère généralement une architecture à trois niveaux (figure **2**) :

— les **sources** d'information qui correspondent à l'ensemble des bases de données de production et sites dont sont extraites les informations décisionnelles ;

— l'**entrepôt** qui contient l'ensemble des données extraites de ces sources ;

— les **magasins** extraits de l'entrepôt et dédiés aux différentes classes de décideurs.

Les trois niveaux de cette architecture ne sont pas nécessairement présents dans les systèmes décisionnels. Par exemple, des logiciels commerciaux permettent de constituer des magasins directement à partir des sources. Cependant, à notre avis, l'architecture à trois niveaux assure l'indépendance de l'entrepôt par rapport aux diverses manipulations faites par les décideurs.

Sources : les informations utiles aux décideurs peuvent être stockées sur des sites de natures diverses (sites web, bases de données relationnelles, fichiers). Les langages de codification, les structures de données sont généralement hétérogènes, ce qui rend délicate l'extraction des données en vue d'alimenter l'entrepôt.

Entrepôt : il s'agit d'une véritable base de données qui contient les données extraites des sources. Elles y sont généralement organisées selon le modèle relationnel.

Seules les informations utiles aux décideurs sont stockées dans l'entrepôt, ce qui nécessite une analyse fine des besoins en matière de données.

L'alimentation de l'entrepôt consiste à extraire périodiquement des données des sources pour « rafraîchir » l'entrepôt. Les **fréquences d'extraction** sont naturellement liées à la volonté des décideurs d'analyser des données récentes.

Au sein d'un entrepôt, des données peuvent être historisées, même si elles ne le sont pas dans les sources.

Exemple : pour connaître l'évolution du salaire d'un employé, il convient de constituer une série chronologique des salaires dans l'entrepôt et de compléter la série à chaque extraction de la source.

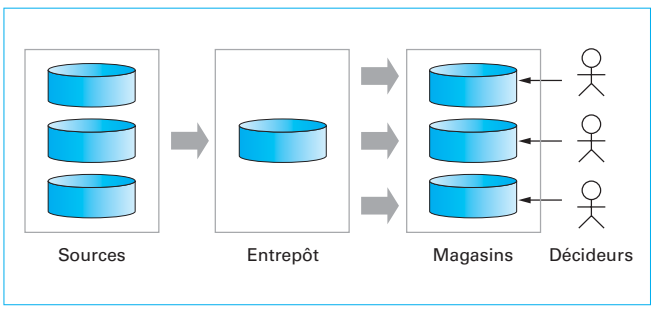


Figure 2 – Architecture à trois niveaux

Un entrepôt de données n'est généralement pas utilisé en l'état par les décideurs dans la mesure où :

— il contient l'ensemble des données décisionnelles d'un domaine d'analyse et concerne donc plusieurs classes de décideurs ;

— les structures de données relationnelles sont inadaptées à certains types d'analyses de données.

- **Magasin** : cette base de données est :

  — extraite d'un entrepôt ;

  — destinée à une classe particulière de décideurs ;

  — organisée selon un modèle adapté aux analyses à réaliser.

Les modèles multidimensionnels sont fréquemment utilisés pour structurer les magasins : modèles en étoile ou en flocon. Ils ont l'avantage d'organiser les données en mettant en évidence l'objet de l'analyse (les faits) et en privilégiant des axes d'étude (les dimensions).

Les données d'un magasin (ou *datamart*) peuvent être aisément manipulées grâce à des requêteurs, des tableurs, des logiciels d'analyse ou de fouille de données [H 3 744].

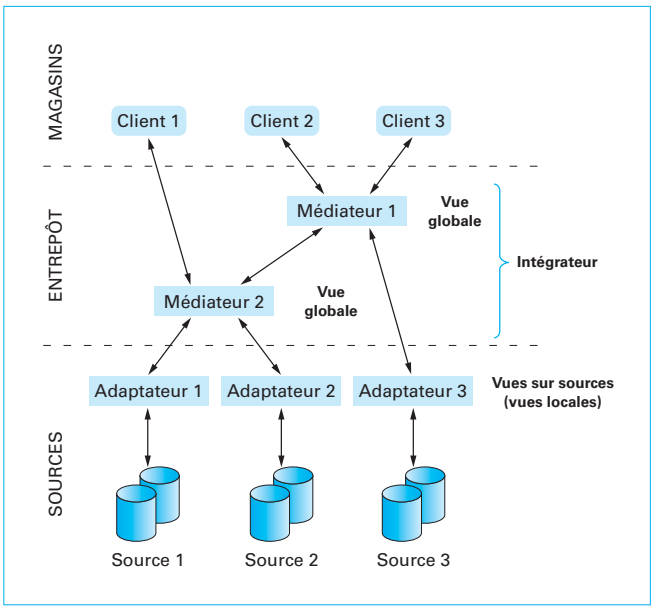


Figure 3 – Approche à base de médiateurs

3. Extraction des sources

Les entrepôts de données sont alimentés à partir de sources multiples, autonomes (gérées par des systèmes différents et indépendants), hétérogènes (du point de vue structurel ou sémantique), éventuellement peu structurées (données semi-structurées) ou non.

Une première démarche pour construire un entrepôt à partir de telles sources consiste à écrire pour chacune un programme *ad hoc* réalisant la sélection des données pertinentes de la source pour les adapter aux exigences du système qui gère l'entrepôt de données. Cette démarche est particulièrement contraignante notamment pour le rafraîchissement de l'entrepôt ainsi que pour s'adapter à ses évolutions.

Une démarche intermédiaire pour s'accorder avec plus de souplesse à l'hétérogénéité des systèmes qui gèrent les sources tout en préservant leur autonomie, consiste à générer pour chaque source (rôle de l'adaptateur) une image dans un modèle compatible avec celui de l'entrepôt. Un processus générique d'unification permet ensuite l'alimentation de l'entrepôt. Généralement, le modèle pivot utilisé pour les images est le modèle relationnel : il suffit généralement d'utiliser la fonction d'**exportation de données** intégrée au système de gestion de la source pour obtenir cette image. Ce processus se simplifie lorsque les sources sont relationnelles ou pseudo-relationnelles, et cette tâche peut être pilotée par des systèmes génériques avec des interfaces graphiques spécifiant les caractéristiques de l'image à partir de la source (voir l'environnement Oracle, § 7). À partir de cet ensemble d'images, on alimente soit directement les tables de l'entrepôt de données, soit les tables associées aux vues d'un système d'intégration de données (appelé aussi système médiateur), dans un **schéma médiateur** (figure **3**).

Il existe principalement deux approches pour définir le schéma médiateur d'un système d'intégration de données :

— ascendante : approche GAV (*global as view*) ;

— descendante : approche LAV (*local as view*).

Dans l'**approche GAV**, le schéma médiateur est défini comme un ensemble de vues sur les sources. Un ensemble de règles spécifie comment se calculent les vues du schéma médiateur à partir des vues image des sources.

Supposons que l'on dispose de vues image de deux sources :
— source S1 : deux vues image : Film[titre,année,genre] et Joue[titre,nom] ;
— source S2 : deux vues image : Movie[mid,title,year,type] et Plays[mid,actor].
À partir de ces vues, les règles suivantes permettent la construction des vues du schéma médiateur :
— règle 1 : Comédie(x,y) :- Film(x,y,'comédie') ∨ Movie(-,x,y,'comedy') ;
— règle 2 : Acteurs(x) :- Joue(y,x) ∧ Film(y,-,'comédie') ∨ Plays(y,x) ∧ Movie(y,-,-,'comedy').
Le schéma médiateur comporte deux vues : Comédie[titre,année] et Acteurs[nom,titre].

Dans l'**approche LAV**, le processus s'inverse dans la mesure où le schéma médiateur est spécifié *a priori* et les vues image sur les sources sont dérivées du schéma médiateur.

On a donc au départ un schéma médiateur : Comédie[titre,année] et Acteurs[nom,titre]. Sur ce schéma sont spécifiées les règles de dérivation des vues image des sources, soit :
— règle 1 : Film(x,y,z) :- Comédie(x,y,null) ∧ z = 'comédie' ;
— règle 2 : Movie(x,y,z,w) :- Comédie(y,z,x) ∧ w = 'comedy' ;
— règle 3 : Joue(x,y) :- Comédie(x,-,-) ∧ Acteurs(y,x) ;
— règle 4 : Plays(x,y) :- Comédie(z,-,x) ∧ Acteurs(y,z).
Les vues image sur les sources sont identiques à celles de l'exemple précédent.

Une approche comme la précédente, basée sur un schéma médiateur, est intéressante du fait de sa **généricité** : il est en effet possible en prenant en compte de nouvelles sources (ensemble de vues) et un ou plusieurs schémas médiateurs (ensembles de vues) de spécifier de nouvelles règles (processus incrémental) pour obtenir un nouveau schéma médiateur.

Le choix d'une approche GAV ou LAV dépend des caractéristiques fonctionnelles souhaitées pour le système d'intégration. Dans une approche GAV, ce sont les sources qui sont le centre d'intérêt et l'adjonction ou la modification structurelle d'une source se répercute potentiellement sur le schéma médiateur. Dans l'approche LAV, c'est le schéma médiateur qui est le centre d'intérêt et l'adjonction d'une source ou une modification structurelle d'une source se

traduit par la création ou l'adaptation de vues sans impact sur le schéma médiateur. C'est plutôt dans ce contexte que s'inscrit l'approche d'extraction pour l'alimentation des entrepôts de données.

Les exemples précédents donnent un aperçu partiel des **problématiques d'extraction et d'unification** pour la construction de l'entrepôt :

1. **hétérogénéité sémantique** : des attributs de sens différents peuvent porter le même nom dans des vues différentes, ou au contraire des attributs de même nom peuvent avoir des sens différents ;
2. pour des attributs sémantiquement compatibles, les valeurs peuvent être **codées différemment** (comédie pour genre et comedy pour type) ;
3. compte tenu des contextes, les **mécanismes d'identification** (clés primaires) peuvent être différents : il faut spécifier les mécanismes de correspondance pour pouvoir concaténer des n-uplets associés à des vues différentes ;
4. ces concaténations permettent de compléter des **descriptions d'entités** à partir de sources différentes à condition d'établir correctement les correspondances. Il faut notamment veiller à ce que les sources utilisées s'inscrivent dans un même référentiel spatio-temporel ;
5. dans le prolongement de ce processus de concaténation, il peut y avoir des entités avec des **descriptions plus ou moins complètes** (ce qui impose la présence de mécanismes de gestion de valeurs nulles).

Pour faciliter le traitement de ces problématiques, chaque nouvelle source à prendre en compte peut être associée à un *data guide* contenant des métadonnées jouant le rôle de descripteur de la source de données, descripteur qui sera utilisé pour spécifier les nouvelles règles de fusion et d'unification.

4. Constitution d'un entrepôt

Si la nécessité de constituer un entrepôt peut paraître évidente dans le cadre de la mise en œuvre d'un système décisionnel de l'entreprise, le premier écueil rencontré concerne la méthode à employer.

Nous présentons ici quelques éléments méthodologiques dans les principales étapes permettant de concevoir et mettre en œuvre un entrepôt.

Les méthodes utilisées pour le développement des traitements transactionnels s'avèrent globalement inadaptées, notamment les démarches en cascade ou en V. Notons cependant que certains de leurs modèles ou étapes peuvent être avantageusement réutilisés dans le développement des systèmes décisionnels.

Par exemple, la méthode MERISE ou le processus unifié (basé sur UML) proposent une démarche pour définir et implanter données et traitements. Le point de départ d'un tel processus est constitué par les besoins spécifiés par les utilisateurs opérationnels, toujours identifiables et généralement clairement définis. Le résultat est un logiciel fermé pour l'utilisateur, sorte de boîte noire activée chaque fois que la fonction qu'il assure est sollicitée par un événement.

Le cas des traitements décisionnels est différent. Il s'agit généralement de traitements non structurés, peu ou pas répétitifs et évolutifs.

Exemple : lors de la définition d'un plan de recrutement, extraire des indicateurs des données opérationnelles concernant les fonctions et âges des salariés de l'entreprise.

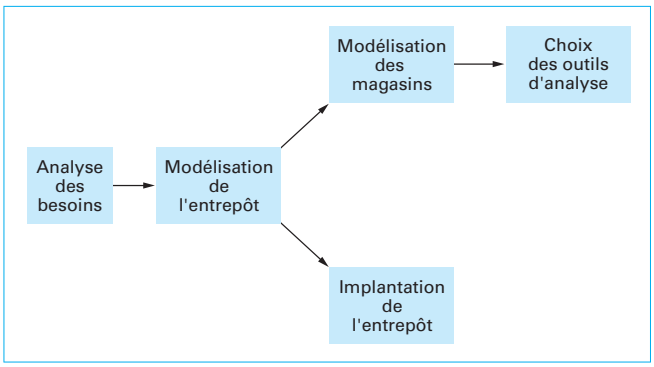


Figure 4 – Démarche de conception

Les décideurs doivent donc être capables d'accéder eux-mêmes aux données du système d'information et d'effectuer les traitements qui leur seront nécessaires au moment où ils en exprimeront le besoin.

Notons que la spécificité des traitements décisionnels exige une démarche itérative comportant des cycles courts. Certains professionnels disent qu'il faut voir grand mais faire petit à petit.

En effet, la constitution d'un entrepôt de données principal (unique) représente un investissement trop important pour une équipe d'informaticiens au sein d'une entreprise ; il est donc nécessaire d'étaler l'activité correspondante sur plusieurs années.

Commencer par le développement d'un projet pilote, unanimement accepté par les différents acteurs impliqués, constitue une solution intéressante pour un démarrage. Puis, si le succès de ce projet facilite l'adhésion des plus réticents, lancer de nouveaux projets plus ambitieux.

De plus, la tendance qui semble se dessiner consiste à faire cohabiter :

— de gros serveurs dédiés aux analyses de données gourmandes en ressources machines ;

— des micro-ordinateurs supportant des magasins de données sur lesquels seront appliqués des requêteurs ou des navigateurs.

Une telle architecture modulaire peut aussi faciliter un développement incrémental du système décisionnel de l'entreprise.

D'autre part, le système décisionnel doit s'adapter à l'évolution constante de l'entreprise et de son environnement. Par conséquent, le périmètre fonctionnel du système décisionnel doit être clairement défini et faire l'objet d'un consensus au sein de l'entreprise.

La figure **4** présente les différentes étapes permettant d'élaborer un entrepôt et les magasins associés.

4.1 Analyse des besoins

Les logiciels de pilotage utilisés par les décideurs permettent d'élaborer des mesures, d'analyser des données, de constituer des tableaux de bord, de suivre en temps réel les informations commerciales ou techniques.

Les besoins du métier correspondant à ces informations de pilotage sont élaborés à partir de données opérationnelles. Celles-ci sont stockées soit dans les bases de données de l'entreprise, soit dans des sources d'information extérieures.

La première phase consiste donc à identifier des **classes de décideurs** et à recenser pour chacune d'elles :

— la **nature des analyses** effectuées régulièrement ou exceptionnellement ;

— les **données brutes** permettant d'élaborer ces analyses ;

— les **sources** de ces données et leur localisation.

Exemple : un gestionnaire produit analyse régulièrement le montant des ventes pour les produits relevant de sa responsabilité. Le montant est obtenu en cumulant les ventes de produits réalisées dans différents magasins sur une période donnée.

Mais les besoins des décideurs sont souvent volatiles, ou au moins fluctuant ; ils exigent donc de mettre en œuvre une démarche itérative pour prendre en compte les évolutions des besoins fonctionnels.

L'implication des utilisateurs dans la définition des besoins est primordiale. Or, ils ne sont pas toujours en mesure d'exprimer leurs attentes. Leurs besoins s'affinent dès l'instant où ils disposent de premiers résultats, d'où la nécessité d'une démarche itérative à cycles courts.

La fiabilité des données brutes issues des bases de production s'avère un point crucial. Elle conditionne naturellement la qualité des analyses et des décisions qui en découlent.

D'autre part, le coût lié à la transformation et à la disponibilité des données brutes doit être étudié, afin d'éviter de lourds calculs ou des coûts d'obtention de données prohibitifs.

4.2 Définition de l'entrepôt

L'entrepôt doit contenir l'ensemble des informations de pilotage extraites de différentes sources. L'entrepôt est géré par les informaticiens qui en extraient des vues virtuelles ou matérialisées (appelées magasins) destinées aux différentes classes de décideurs.

En théorie, on pourrait être tenté de constituer un entrepôt gigantesque pour répondre aux besoins connus mais aussi aux besoins futurs des décideurs. Même si elle est techniquement envisageable, cette solution est à éviter parce que difficilement maîtrisable.

Le modèle relationnel est généralement utilisé pour organiser les données de l'entrepôt. Cependant, les règles de normalisation, qui permettent d'éliminer la redondance, ne sont pas appliquées dans l'entrepôt. En effet, sa mise à jour s'effectue par extraction périodique dans des bases de données de production (sources) supposées fiables.

La dimension temporelle, si utile aux décideurs pour analyser l'évolution des données dans le temps, exige le stockage dans l'entrepôt des différentes versions des objets sur une période. Par exemple, une base de production peut uniquement contenir les derniers montants des salaires du personnel pour effectuer la paye ; mais l'analyse de l'évolution de ces salaires sur les cinq dernières années nécessite que l'entrepôt stocke les différents montants des salaires sur cette période. La gestion des versions peut naturellement entraîner un volume considérable de données à stocker et à gérer.

Le rafraîchissement des données à partir des sources peut être global pour l'entrepôt mais peut aussi faire l'objet d'une étude plus fine. Différents scénarios peuvent être étudiés : les multiples analyses effectuées par les décideurs nécessitant des degrés différents dans la « fraîcheur des données ». Le rafraîchissement de l'entrepôt ayant un coût non négligeable, les informaticiens proposeront plusieurs solutions évaluées financièrement et soumises aux acteurs concernés.

4.3 Choix des outils d'analyse

Les outils de « requêtage », d'analyse et de restitution (*reporting*, OLAP : On-Line Analytical Processing, *data mining*, etc.) représentent la face visible de l'iceberg décisionnel.

Il est bien entendu que la performance et la qualité des mesures sont des facteurs déterminants dans l'efficacité des analyses. Les outils d'analyse doivent donc être adaptés au périmètre fonctionnel.

Le développement du système décisionnel dans une entreprise ne doit pas être guidé par des outils du marché. Mais les décideurs doivent connaître leurs possibilités.

5. Modélisation multidimensionnelle des magasins

5.1 Nécessité de modèles adaptés

Les données à analyser doivent refléter la vision des analystes, c'est-à-dire apparaître sous une forme facilitant les prises de décision. Cette vision correspond à une structuration des données selon plusieurs axes d'analyse représentant des notions diverses telles que le temps, la localisation géographique, une nomenclature de produits, etc. On parle d'analyse multidimensionnelle.

La modélisation traditionnelle sous la forme de relations est inadéquate pour supporter efficacement les analyses multidimensionnelles. Pour s'en convaincre, considérons la relation de la figure **5*a***. On peut distinguer deux axes selon lesquels les ventes de l'année 2000 peuvent être observées : un axe relatif aux catégories de produits vendus et un axe relatif aux départements des points de vente. La représentation relationnelle de ces données ne facilite pas le travail de l'analyste ; par exemple, l'observation des ventes dans la Haute-Garonne (Département = 31) met en jeu des enregistrements disséminés dans la relation.

Une vision plus proche de celle des analystes consiste à organiser les données dans un tableau où les axes d'analyse sont croisés (figure **5*b***) ; il s'agit d'une représentation de données à deux dimensions. Dans ce contexte, l'observation des ventes dans la Haute-Garonne consiste simplement à analyser la première ligne. Pour construire cette représentation à partir de la relation initiale, les enregistrements sont répartis en lignes et en colonnes.

Par extension, si l'on considère plusieurs tableaux relatifs aux ventes annuelles de l'année 2000 à l'année 2002, on observe alors les données dans un espace à trois dimensions (figure **5*c***).

La **modélisation multidimensionnelle** consiste donc à considérer les données à analyser – on parle de sujet analysé – comme un point dans un espace à plusieurs dimensions. Les données sont ainsi organisées de telle sorte que le sujet analysé et les axes de l'analyse soient mis en évidence. Cette organisation multidimensionnelle a favorisé l'utilisation de la métaphore du **cube de données** (figure **6**).

5.2 Étude des modèles par niveau d'abstraction

Nous proposons d'étudier la modélisation des bases multidimensionnelles, de manière classique en base de données, au travers des différents niveaux d'abstraction :

Ventes 2000	Catégorie	Département	Montant
	Papeterie	31	150.00
	Papeterie	81	100.00
	Papeterie	82	100.00
	Micro	31	250.00
	Micro	81	200.00

(a) relation ventes 2000

Ventes 2000	Catégorie	Papeterie	Micro
Département			
31		150.00	250.00
81		100.00	200.00
82		100.00	

(b) tableau croisé des ventes 2000

Ventes 2000	Catégorie	Papeterie	Micro
Ventes 2001	Catégorie	Papeterie	Micro
Ventes 2002	Catégorie	Papeterie	Micro
Département			
31		200.00	260.00
81		140.00	220.00
82		130.00	

(c) tableaux croisés des ventes 2000, 2001, 2002

Figure 5 – De la modélisation traditionnelle à la modélisation multidimensionnelle

— le **niveau conceptuel** (§ 5.2.1) qui consiste à décrire la base multidimensionnelle indépendamment des choix d'implantation ;
— le **niveau logique** (§ 5.2.2) qui consiste à définir la base multidimensionnelle en utilisant la technologie liée au type de logiciel utilisé (R-OLAP, M-OLAP, etc.) ;
— le **niveau physique** (§ 5.2.3) qui revient à spécifier la base multidimensionnelle avec les mécanismes offerts par le logiciel utilisé (Oracle 9i, etc.).

5.2.1 Niveau conceptuel

La modélisation multidimensionnelle repose sur les concepts de **fait**, de **dimension** et de **hiérarchie**. Cette modélisation consiste à décrire les données analysées au travers d'un **schéma en étoile**.

Un schéma en étoile décrit le sujet analysé au travers d'un **fait**. Celui-ci est défini par un nom et un ensemble d'attributs appelés **mesures** ou **indicateurs**. Par exemple, on peut analyser les ventes au travers de la mesure montant. Les axes d'analyse selon lesquels le fait est observé sont modélisés par des **dimensions**. Par exemple, le montant des ventes peut s'analyser suivant une dimension temporelle, une dimension géographique ou encore une dimension représentant les produits et leurs catégories. Les dimensions comportent un ou plusieurs attributs qui sont le plus souvent organisés suivant des **hiérarchies**. Une hiérarchie modélise les niveaux de granularité auxquels les mesures sont observées. Par exemple, une dimension temporelle comportant les attributs année, mois, libellé_mois, jour supporte une hiérarchie indiquant que l'année représente une granularité plus élevée que le mois, lui-même plus général que le jour. Les attributs définissant les niveaux de granularité sont appelés **paramètres** tandis que les attributs informationnels (libellé_mois), qui sont reliés à un paramètre, sont dits **attributs faibles**.

La modélisation multidimensionnelle intègre donc des concepts spécifiques pour lesquels les notations existantes (entité-relation, UML : Unified Modeling Language [H 3 238]) s'avèrent imparfaite. L'inadéquation des notations existantes est par exemple liée à la représentation des hiérarchies associées à chaque dimension. Des notations spécifiques doivent être proposées. La figure **7** présente un exemple de schéma conceptuel décrit avec des notations spécifiques adaptées. Sur cette figure, le fait représenté concerne des VENTES dont les mesures le caractérisant sont montant et quantité. Les ventes sont analysées suivant trois dimensions : TEMPS, PRODUITS et CLIENTS. Sur chacune d'elles, sont définies des hiérarchies dont les paramètres sont représentés par un cercle tandis que les attributs faibles associés sont soulignés.

Par extension, il est possible de décrire plusieurs sujets d'analyse (faits) observables suivant des dimensions communes et/ou spécifiques. La modélisation d'une telle réalité analytique consiste à décrire un **schéma en constellation** (figure **8**) issu de l'intégration de plusieurs sous-schémas en étoile.

5.2.2 Niveau logique

La modélisation multidimensionnelle au niveau logique peut se baser sur la technologie des SGBD relationnels (SGBDR) qui représentent l'immense majorité des SGBD du marché ; le terme de modélisation R-OLAP (Relational OLAP) désigne cette approche. D'autres options sont possibles comme le développement de SGBD spécifiques, intégrant des structures natives adaptées aux exigences des structures multidimensionnelles. Il s'agit de solutions M-OLAP (Multidimensional OLAP). Enfin, des techniques hybrides H-OLAP (Hybrid OLAP) sont également possibles.

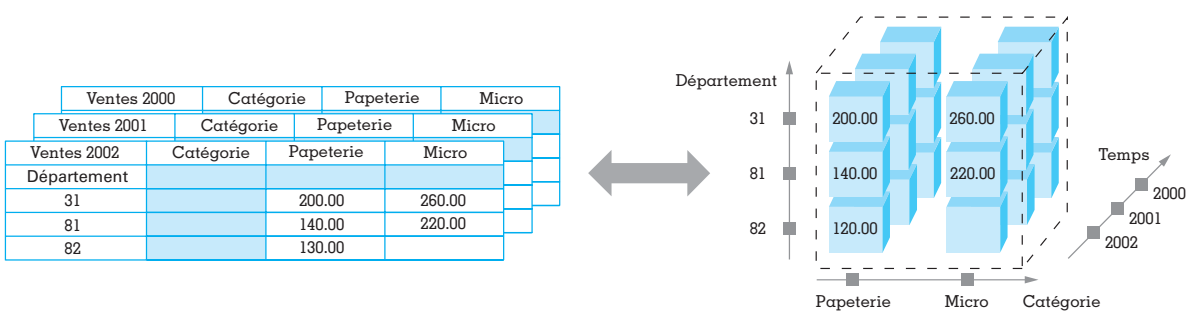


Figure 6 – Cube de données

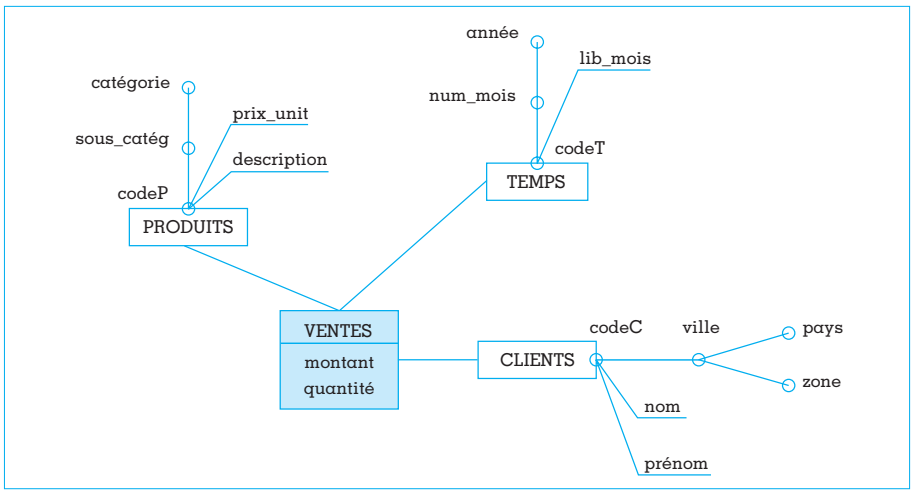


Figure 7 – Schéma conceptuel en étoile

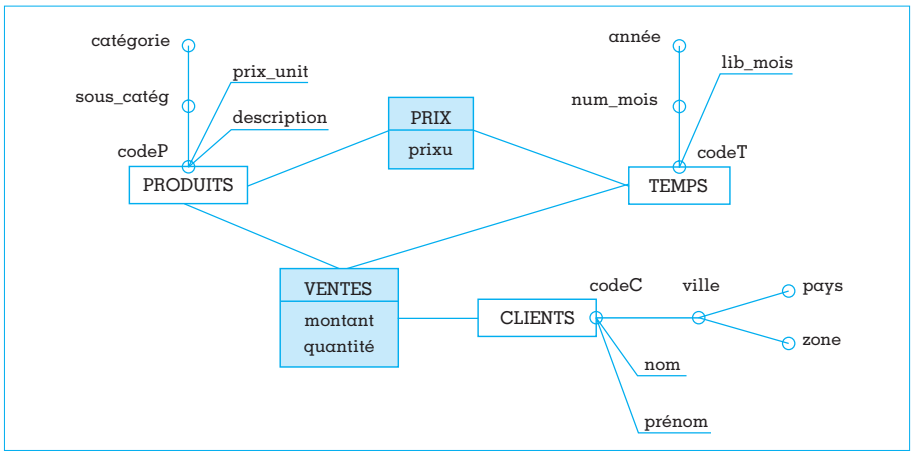


Figure 8 – Schéma conceptuel en constellation

Ici, nous considérons l'approche la plus courante, R-OLAP. Le modèle multidimensionnel conceptuel est alors traduit de la manière suivante :

— à chaque dimension correspond une relation de même nom dont les attributs sont dérivés des paramètres et attributs faibles de la dimension. Parmi les attributs de la dimension, on distingue un ou plusieurs attributs formant la clé primaire (paramètre correspondant au niveau de granularité le plus fin) ;

— à chaque fait correspond une relation de même nom. Le schéma de la relation est constitué d'attributs représentant les mesures, des clés étrangères référençant les dimensions liées au fait. La clé primaire du fait peut être constituée par la concaténation des clés étrangères ou bien par un attribut supplémentaire comptant les enregistrements.

La figure **9** présente un exemple de schéma R-OLAP en étoile issu de la traduction du schéma conceptuel en étoile de la figure **7**.

La normalisation des tables de dimension permet de spécifier de manière explicite les hiérarchies ; nous parlons alors de **schéma en flocon** (figure **10**).

5.2.3 Niveau physique : Oracle 9i

Les ordres classiques du langage SQL permettent d'implanter un schéma en étoile, en flocon ou en constellation, mais ils s'avèrent rapidement insuffisants.

En effet, la commande CREATE TABLE...AS... permet effectivement de construire une table valuée par recopie de données éventuellement distantes (utilisation de DATABASE LINK). Cependant, lors de l'évolution des tables source, cette technique s'avère inadéquate puisqu'il est nécessaire de recalculer l'ensemble des données à copier et lors d'une évolution du schéma, la requête d'extraction doit être redéfinie.

De même, la commande CREATE VIEW...AS... permet de construire des vues virtuelles au-dessus de tables source afin de donner une vision multidimensionnelle des données. Néanmoins, les performances d'une telle infrastructure sont très vite mises à mal compte tenu du fait que la vue doit être recalculée à chaque utilisation alors que des volumes de données très importants sont mis en jeu dans le contexte des systèmes décisionnels. Les temps de réponse aux requêtes des utilisateurs dans un tel contexte deviennent très vite inacceptables.

Devant ces insuffisances, les constructeurs de SGBDR ont étendu les ordres SQL. Ainsi Oracle propose plusieurs commandes spécifiques telles que :

— CREATE MATERIALIZED VIEW...AS... qui permet de calculer des vues dont les données sont stockées et rafraîchies (mises à jour) par le système avec une périodicité définie par le concepteur ;

— CREATE DIMENSION... qui permet la définition de hiérarchies pouvant être exploitées pour calculer des préagrégats.

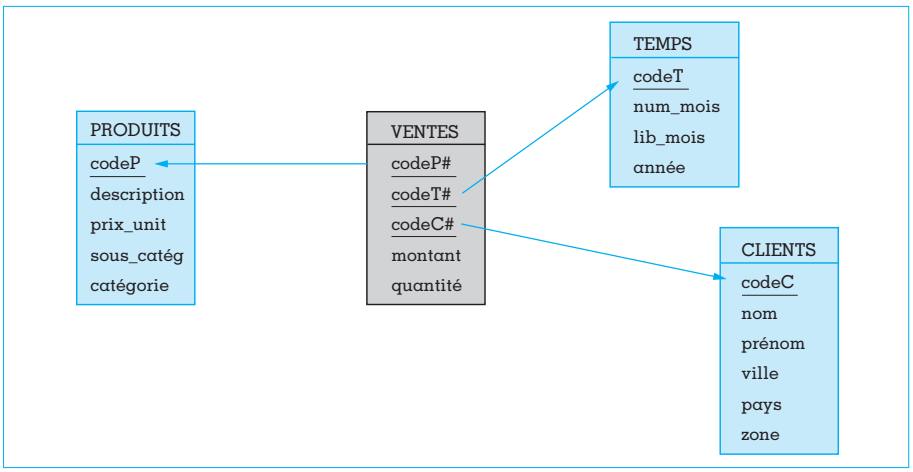


Figure 9 – Schéma logique R-OLAP en étoile

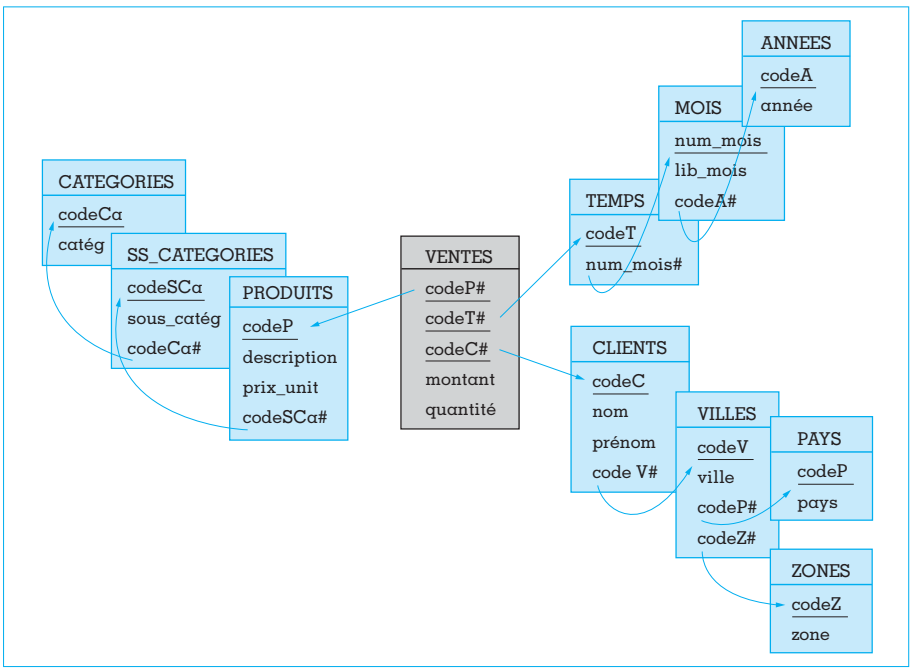


Figure 10 – Schéma logique R-OLAP en flocon

■ La commande de création des **vues matérialisées** a pour syntaxe :

```
CREATE MATERIALIZED VIEW <nomvue>
BUILD { IMMEDIATE | DEFERRED }
REFRESH { COMPLETE | FAST | FORCE | NEVER }
                { ON DEMAND | ON COMMIT }
AS SELECT ... ;
```

Plusieurs options permettent de configurer le fonctionnement de la vue matérialisée :

— IMMEDIATE : création de la vue matérialisée et population de la vue ;

— DEFERRED : création de la vue matérialisée sans être alimentée en données. DBMS_MVIEW.REFRESH (<liste_vues>) alimente la vue ;

— ON COMMIT : rafraîchissement à chaque fin de transaction modifiant les tables source ;

— ON DEMAND : rafraîchissement avec DBMS_MVIEW.REFRESH ;

— COMPLETE : recalcul complet de la vue ;

— FAST : application d'un rafraîchissement incrémental ;

— FORCE : FAST si possible, COMPLETE sinon ;

— NEVER : pas de rafraîchissement.

Oracle propose également des vues matérialisées spécifiques, portant directement sur les tables source et permettant des rafraîchissements plus rapides. La syntaxe de ces vues dites de chargement est la suivante :

```
CREATE MATERIALIZED VIEW LOG ON <tablesource>
WITH SEQUENCE, ROWID (<liste_attributs>)
INCLUDING NEW VALUES ;
```

La clause SEQUENCE autorise les opérations de manipulation complexes (combinant INSERT, UPDATE ou DELETE sur plusieurs

tables). Les vues de chargement doivent être définies sur les tables source pour autoriser :

— le rafraîchissement rapide (REFRESH FAST) ;

— la définition de vues agrégées ;

— la définition de vues avec jointures (chaque ROWID des tables source doit être projeté en résultat).

■ La commande de création des **dimensions** a été proposée dans Oracle afin de permettre la définition des hiérarchies sur les tables de dimension. La syntaxe de cette commande est la suivante :

```
CREATE DIMENSION <nomdimension>
   LEVEL <niveau1> IS (<nomtable.nomattribut1>)
   LEVEL <niveau2> IS (<nomtable.nomattribut2>)
   ...
WITH HIERARCHY <nomhierarchie1> (
   <niveaui1> CHILD OF <niveaui2> CHILD OF ...)
WITH HIERARCHY <nomhierarchie2> (
   <niveaui1> CHILD OF <niveaui2> CHILD OF ...)
...
ATTRIBUTE <niveauj1> DETERMINES <nomattributx>
ATTRIBUTE <niveauj2> DETERMINES <nomattributy>
... ;
```

La démarche de construction d'une dimension comprend plusieurs étapes faisant appel à plusieurs ordres SQL :

— définition d'une vue matérialisée ;

— définition de la clé primaire de la vue matérialisée ;

— définition de la dimension associée à la vue matérialisée.

6. Analyse décisionnelle

6.1 Contexte

Les magasins constituent un extrait des données contenues dans l'entrepôt de données dédié à une fonction dans l'entreprise ou à une classe d'utilisateurs. Les décideurs manipulent les données de ces magasins au travers d'outils spécifiques. Pour répondre à ce besoin, chaque magasin repose sur un modèle de données particulier. Ces modèles de données peuvent prendre la forme suivante :

— une BD relationnelle ;

— un tableau manipulé au travers d'un tableur du marché ;

— un environnement spécifique pour la prise de décision (univers Business Objects, catalogue de Cognos) ;

— une BD multidimensionnelle.

La figure **11** schématise ces différents types de magasins de données.

Une fois que ces magasins ont été définis, ils sont manipulés avec des outils de manipulation décisionnelle. Nous pouvons classer ces outils en différentes catégories :

— requêteurs pour bases de données relationnelles ou multidimensionnelles ;

— tableurs avec leurs différentes fonctions ;

— outils spécifiques d'analyse OLAP ;

— outils de fouille de données.

6.2 Outils de manipulation décisionnelle

6.2.1 Requêteurs

Dans cette catégorie, nous pouvons recenser les requêteurs SQL. Ils permettent de manipuler des bases de données relationnelles. La connaissance du schéma de la BD et du langage SQL ainsi que la restitution du résultat sous forme de tableau ne sont pas toujours en adéquation avec les besoins d'analyse des décideurs. Aussi, sont proposés sur le marché des requêteurs présentant les données sous une forme plus adaptée. Ces outils intègrent une couche conceptuelle permettant aux utilisateurs de spécifier leurs requêtes à partir d'un référentiel. Ce référentiel permet de stocker des noms de données ou d'indicateurs en utilisant le vocabulaire professionnel courant des décideurs. Ce référentiel conserve également le lien entre les noms métiers et les noms des attributs physiques dans la BD.

Lors de l'utilisation d'un requêteur, le décideur sélectionne les éléments désirés dans le référentiel, y intègre d'éventuelles sélections et l'outil joint les différents éléments sélectionnés du référentiel pour construire un état. Ces outils comprennent différentes possibilités de définition d'états plus ou moins complexes (multiétats, graphiques, etc.). La plupart des éditeurs proposent également de diffuser ces rapports via un navigateur internet pour alimenter un intranet ou un extranet. Certains éditeurs parlent d'**EIS** (Executive Information System) correspondant à un environnement offrant à l'utilisateur des facilités de présentation synthétiques et graphiques de compte rendu d'activité.

6.2.2 Tableurs

Dans ce cas, le décideur utilise un tableur du marché pour représenter les informations sous forme de tableau. Une fois que le tableau est construit, il peut :

— l'analyser directement ; la structure du tableau correspond à ses besoins ;

— compléter ce tableau en y associant des graphiques (fonctions graphiques du tableur choisi) ;

— le manipuler pour créer de nouveaux tableaux simples (fonctions de calcul plus ou moins complexes présentes dans un tableur) ou à *n* dimensions (tableaux croisés dynamiques d'Excel).

6.2.3 Outils d'analyse OLAP

Ces outils intègrent les aspects multidimensionnels dans la manipulation des données. Ainsi, ils permettent de manipuler les données en effectuant des rotations de dimensions ou de hiérarchies, des forages vers le bas ou vers le haut (parcours des niveaux de granularité d'une hiérarchie de dimension).

Dans cette catégorie, nous trouvons les extensions du langage SQL proposées par les éditeurs de SGBD.

Nota : nous pouvons remarquer qu'Oracle dans ses dernières versions proposait une extension de la commande Group By tandis que Microsoft proposait une extension de la clause Select afin de présenter les données sous une forme adaptée à la prise de décision.

À l'heure actuelle, il n'existe pas de norme pour la manipulation multidimensionnelle des bases de données.

Cette catégorie regroupe également des outils graphiques permettant de manipuler les données au travers de tableaux à deux (ou *n*) dimensions. Certains éditeurs permettent également d'effectuer des analyses sur ces rapports (sélection, visualisation des données en détails). Nous pouvons parler de rapports dynamiques. Ces outils graphiques conviviaux sont destinés aux décideurs non informaticiens.

Ces deux types de manipulation reposent sur les principes de l'algèbre multidimensionnelle (§ 6.3).

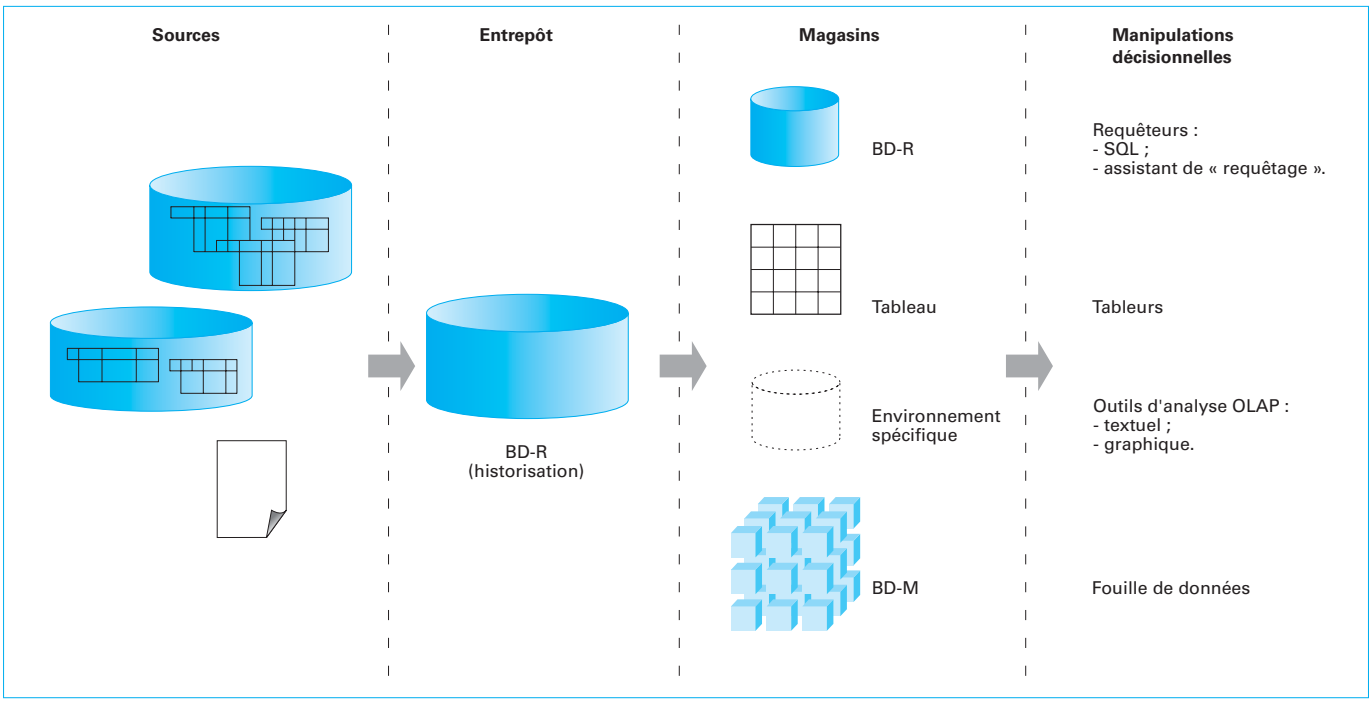


Figure 11 – Typologie des magasins

6.2.4 Outils de fouille de données

Les outils de fouille de données (*data mining*) regroupent l'ensemble des techniques permettant d'aller puiser des informations pertinentes dans les montagnes de données stockées ces dernières décennies dans les entreprises. La fouille de données repose sur de nombreux modèles mathématiques et statistiques.

6.3 Algèbre multidimensionnelle

Lors de la manipulation graphique ou textuelle de données au moyen d'outils d'analyse multidimensionnelle, les données sont visualisées au travers de tables à deux dimensions. La représentation sous forme de tables est motivée par sa simplicité et sa précision ; il s'agit d'une représentation très répandue à laquelle les décideurs sont habitués [1] [2], notamment par différents outils du marché.

Nous définissons plus loin ce concept de table multidimensionnelle (§ 6.3.1) ainsi que l'opérateur de construction de cette table (§ 6.3.2). Nous terminons par la présentation des opérateurs de l'algèbre multidimensionnelle, à savoir les opérateurs de transformation de granularité des données (§ 6.3.3) et de transformation de structure des données (§ 6.3.4).

6.3.1 Table multidimensionnelle

Une table dimensionnelle est la représentation des instances d'un fait et de ses dimensions. Plus précisément, une table multidimensionnelle présente les valeurs des mesures d'un fait en fonction des valeurs des paramètres des dimensions représentées en lignes et colonnes ainsi que des autres dimensions représentées de manière non détaillée. Une telle table correspond à une tranche de cube multidimensionnel. Nous pouvons définir une table multidimensionnelle comme suit :

TD = (F, {m1, m2, ...},
< (D1, <h1, h'1 , ...>, <p11, p12, ...>, <(v_{p11}, v_{p12}, ...),
(v'_{p11} , v'_{p12} , ...), ...>),
(D2, <h2, h'2 , ...>, <p21, p22, ...>, <(v_{p21}, v_{p22}, ...),
(v'_{p21} , v'_{p22} , ...), ...>),
(D3, <h3, h'3 , ...>, <p31, p32, ...>, <>), ...
(D*n*, <h*n*, h'*n* , ...>, <p*n*1, p*n*2, ...>, <>) >,
pred)

avec

F	le fait,
m1, m2	les mesures affichées,
D1	la dimension affichée en ligne,
D2	la dimension affichée en colonne,
h1	la hiérarchie utilisée,
pij	les paramètres affichés,
v_{pij}, v'_{pij}	les valeurs affichées,
D3 à D*n*	les dimensions non détaillées,
pred	le prédicat de sélection sur le fait et/ou les dimensions.

La table dont la définition est la suivante est relative à l'analyse des montants des ventes des produits de la catégorie C2 pour chacun des pays des clients. La figure **12** donne sa représentation graphique.

TD = (VENTES, {montant},
< (TEMPS, <h_an> <annee>, < (2001), (2000), (1999) >),
(CLIENTS, <h_cli, h_zon>, <pays>, < (Allemagne),
(France) , (Espagne) >)
(PRODUITS, <h_cat>, <>, <>) >,
PRODUITS.categories='C2')

VENTES (montant)		CLIENTS / h_cli			
		Pays	Allemagne	France	Espagne
TEMPS / h_an	année				
	2001		200	150	300
	2000		250	240	260
	1999		200	210	220
PRODUITS.categories = 'C2'					

Figure 12 – Représentation graphique d'une table multidimensionnelle

6.3.2 Opérateur de construction

Le premier opérateur de l'algèbre consiste à construire une telle table à partir des données d'une BD multidimensionnelle :

DISPLAY (Nom^S, Nom^F [, {m1, m2,...}] [,(D1 [,h]), (D2 [, h'])]) = TD

Cet opérateur possède les arguments suivants :

— Nom^S : nom du schéma en constellation ;

— Nom^F : sujet d'analyse (fait) ;

— {m1, m2, ...} : ensemble des mesures affichées (courantes) ;

— D1, D2 : dimensions affichées respectivement en ligne et en colonne ;

— h, h' : hiérarchies utilisées pour naviguer respectivement sur D1 et D2.

Par défaut, cet opérateur intègre les paramètres suivants :

— affichage de toutes les mesures du fait courant ;

— utilisation de deux premières dimensions associées à F ;

— positionnement sur la première hiérarchie des dimensions courantes ;

— positionnement sur le paramètre de granularité maximale.

6.3.3 Opérateurs de transformation de la granularité des données

Nous retrouvons les quatre opérateurs définis dans le tableau **1**.

Tableau 1 – Opérateurs de transformation de la granularité des données

Catégorie	Opérateur	Description
Forage	DrillDown	Forage vers le bas
	RollUp	Forage vers le haut
Calcul	Cube	Agrégations en ligne et colonne
	UnCube	Suppression des agrégations

■ **Forages** : ils permettent de visualiser les mesures du fait de manière plus ou moins détaillée en parcourant les niveaux d'une hiérarchie d'une dimension (figure **13**). Dans cette catégorie, nous retrouvons les deux opérateurs suivants :

DrillDown (TD, Di, pk) = TDr
— diminution de la granularité des données visualisées ;
— forage vers le bas sur la hiérarchie courante de la dimension Di (en ligne ou colonne) jusqu'au paramètre pk ;
— ajout d'un ou de plusieurs paramètres en ligne ou en colonne.

RollUp (TD, Di, pk) = TDr
— augmentation de la granularité des données visualisées ;
— forage vers le haut sur la hiérarchie courante de la dimension Di (en ligne ou colonne) jusqu'au paramètre pk ;
— suppression d'un ou de plusieurs paramètres en ligne ou en colonne.

■ **Calculs** : ces opérateurs permettent d'ajouter ou de supprimer dans la table multidimensionnelle une colonne et une ligne contenant une agrégation calculée des données contenues dans la table. Nous retrouvons les opérateurs Cube et Uncube définis comme suit :

Cube (TD, Fnct) = TDr
— Fnct ∈ {SUM, COUNT, AVG, MIN, MAX, DVT,...} ;
— calculs d'agrégation ajoutés pour chaque ligne et colonne ;
— ajout d'une ligne et d'une colonne contenant les calculs d'agrégation.

UnCube(TD) = TDr
— opération inverse ;
— supprime la ligne et la colonne contenant les calculs d'agrégation.

6.3.4 Opérateurs de transformation de la structure des données

Ces opérateurs permettent d'effectuer des analyses en réalisant des sélections, des rotations, des permutations, voire des transformations de mesure en paramètre et *vice versa*. Le tableau **2** présente succintement ces opérateurs.

Tableau 2 – Opérateurs de transformation de la structure des données

Catégorie	Opérateur	Description
Classique	Select	Sélection des positions d'un paramètre Sélection des valeurs d'une mesure
	AddM	Ajout d'une mesure visualisée
	DelM	Suppression d'une mesure visualisée
Rotation	FRotate	Rotation de faits (partageant des dimensions)
	DRotate	Rotation de dimensions
	HRotate	Rotation de hiérarchies d'une dimension
Permutation	Switch	Permutation de positions (valeurs) d'un paramètre
	Order	Ordonnancement des positions d'un paramètre
	Nest	Permutation de paramètres
Transformation	Push	Conversion d'un paramètre en mesure
	Pull	Conversion d'une mesure en paramètre

VENTES (montant)		CLIENTS / h_cli			
		Pays	Allemagne	France	Espagne
TEMPS / h_an	année				
	2001		200	150	300
	2000		250	240	260
	1999		200	210	220
PRODUITS.all					

DrillDown (TDVentes1, CLIENTS, ville) = TDVentes2

RollUp (TDVentes2, CLIENTS, pays) = TDVentes1

VENTES (montant)		CLIENTS / h_cli					
		Pays	Allemagne		France		Espagne
		Ville	Berlin	Hambourg	Paris	Toulouse	Madrid
TEMPS / h_an	année						
	2001		150	50	100	50	300
	2000		160	90	100	140	260
	1999		100	100	110	100	220
PRODUITS.all							

Figure 13 – Opérations de forage

VENTES (montant)		CLIENTS / h_cli			
		Pays	Allemagne	France	Espagne
TEMPS / h_an	année				
	2001		200	150	300
	2000		250	240	260
	1999		200	210	220
PRODUITS.all					

HRotate (TDVentes1, CLIENTS, H_CLI, H_ZON) = TDVentes3

VENTES (montant)		CLIENTS / h_zon				
		Zone	Nord	Sud	Est	Ouest
TEMPS / h_an	année					
	2001		100	120	200	180
	2000		230	210	200	140
	1999		180	150	190	160
PRODUITS.all						

Figure 14 – Rotation de hiérarchies

■ **Rotations** : ces opérateurs permettent de changer de sujet d'analyse (rotation de faits), d'axe d'analyse (rotations de dimensions) ou de vues d'une même dimension (rotation de hiérarchies). Ils sont définis comme suit :

FRotate (TD, F2) = TDr
— rotation du fait courant pour visualiser les mesures du fait F2 ;
— partage au minimum les deux dimensions visualisées.

DRotate (TD, D1, D2 [, Hi]) = TDr
— rotation de la dimension D1 avec la dimension D2 ;
— il est possible de préciser la hiérarchie Hi (définie sur la dimension D2) à utiliser pour la visualisation ;
— positionnement sur le paramètre de granularité maximale de la nouvelle hiérarchie.

HRotate (TD, D, H1, H2) = TDr
— rotation de la hiérarchie H1 avec la hiérarchie H2 (utilisée pour visualiser les données de la dimension D).

Exemple : sur la figure **14**, les décideurs analysent le montant des ventes des produits par année et par pays des clients. Ils souhaitent modifier cette analyse afin de visualiser ce montant des ventes par année et par zone de chalandise. Pour ce faire, ils doivent utiliser l'opérateur de rotation de dimension.

■ **Permutations** : ces opérateurs agissent sur les paramètres (ainsi que sur leurs valeurs) servant de base à la définition de la table multidimensionnelle (figure **15**). Les trois opérateurs de cette catégorie sont définis comme suit :

Switch (TD, D, p, v1, v2) = TDr
— permutation des valeurs v1 et v2 du paramètre p avec répercussion sur les valeurs des paramètres de granularité inférieure.

Order (TD, D, p, s) = TDr
— ordonnancement des valeurs du paramètre p ;
— s = 'ASC' : classement croissant (ascendant) ;
— s = 'DESC' : classement décroissant (descendant).

Nest (TD, D, p1, p2) = TDr
— permutation des paramètres p1 et p2 sur la hiérarchie courante ;
— le paramètre p1 est imbriqué dans le paramètre p2.

■ **Transformations** : ces opérateurs permettent de transformer une mesure en paramètre et *vice versa* (figure **16**). Nous pouvons définir ces deux opérateurs comme suit :

Push(TD, D, p) = TDr
— conversion (transformation) du paramètre p en mesure dans le fait courant ;
— p ne doit pas être le paramètre de plus bas niveau ! ;
— D est affichée avec au moins deux paramètres.

Pull(TD, D, m) = TDr
— conversion (transformation) d'une mesure du fait courant en paramètre de la dimension courante D ;
— le nouveau paramètre est positionné comme granularité minimale des paramètres affichés ;
— le fait courant est visualisé avec au moins deux mesures.

■ **Opérateurs classiques** : ils permettent d'affecter un prédicat de sélection aux paramètres des dimensions et/ou aux mesures du fait analysé, voire d'ajouter ou de supprimer une mesure dans la table multidimensionnelle (figure **17**). Ces opérateurs sont définis comme suit :

Select (TD, predσ) = TDr
— restriction sur les valeurs restituées ;
— le prédicat de sélection predσ porte sur les dimensions et/ou le fait.

AddM / DelM (TD, m) = TDr
— ajout / suppression d'une mesure dans un fait visualisé.

7. L'exemple d'Oracle

La tendance suivie actuellement par la majorité des constructeurs de logiciels en décisionnel, est de proposer une offre complète couvrant toute la chaîne logicielle nécessaire à l'élaboration d'un système décisionnel constitué d'un entrepôt et de magasins multidimensionnels. Parmi ces acteurs, nous citons Oracle dont l'offre comporte :

— un logiciel de stockage des données avec Oracle Application Server ;
— un logiciel de type ETL (*extract, transform, load* ou extraction, transformation, chargement) avec Oracle Warehouse Builder ;
— un logiciel d'interrogation, d'analyse et de *reporting* avec Oracle Discoverer.

Nous présentons ici le principe général de l'ETL Oracle Warehouse Builder qui permet de définir et de construire une base de données cible à partir de sources de données (BD, fichiers, etc.).

Dans ce logiciel, la constitution d'un système décisionnel est organisée au sein d'un projet. Ce dernier comprend un ensemble de modules ; chaque module représente une base de données source ou cible. Un module, comme l'illustre la figure **18**, est défini au travers d'un ensemble d'objets : tables, vues, vues matérialisées, transformations PL/SQL, séquences, faits, dimensions, *mapping*, c'est-à-dire des processus d'extraction.

VENTES (montant)		CLIENTS / h_cli			
		Pays	France	Allemagne	Espagne
TEMPS / h_an	année				
	2001		150	200	300
	2000		240	250	260
	1999		210	200	220
PRODUITS.all					

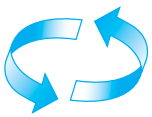
Nest (Switch (DrillDown (TDVentes1, TEMPS, lib_mois), CLIENTS, pays, Allemagne, France), TEMPS, année, mois_lib) = TDVentes4

VENTES (montant)			CLIENTS / h_cli			
			Pays	Allemagne	France	Espagne
TEMPS / h_an	mois_lib	année				
	janv	2001		50	35	50
		2000		40	40	50
		1999		40	35	45
	févr	2001		25	15	40
		2000		20	20	20
		1999		15	20	25
	...	...		...	...	...
PRODUITS.all						

Figure 15 – Combinaison d'opérateurs de permutation

VENTES (montant)		CLIENTS / h_cli				
		Pays	Allemagne		France	Espagne
		Ville	Berlin	Hambourg	Paris	Toulouse	Madrid
TEMPS / h_an	année						
	2001		150	50	100	50	300
	2000		160	90	100	140	260
	1999		100	100	110	100	220
PRODUITS.all							

Push (TDVentes2, CLIENTS, pays) = TDVentes5



Nest (Pull (TDVentes5, CLIENTS, pays), CLIENTS, ville, pays) = TDVentes2

VENTES (montant, pays)		CLIENTS / h_cli					
		Ville	Berlin	Hambourg	Paris	Toulouse	Madrid
TEMPS / h_an	année						
	2001		150, Allemagne	50, Allemagne	100, France	50, France	300, Espagne
	2000		160, Allemagne	90, Allemagne	100, France	140, France	260, Espagne
	1999		100, Allemagne	100, Allemagne	110, France	100, France	220, Espagne
PRODUITS.all							

Figure 16 – Opérations de transformation

VENTES (montant)		CLIENTS / h_cli			
		Pays	Allemagne	France	Espagne
TEMPS / h_an	année				
	2001		200	150	300
	2000		250	240	260
	1999		200	210	220
PRODUITS.all					

AddM (Select (TDVentes1, pays = 'Allemagne' ∧ montant > 200), quantité) = TDVentes6

VENTES. montant > 200 (montant)		CLIENTS / h_cli	
		Pays	Allemagne
TEMPS / h_an	année		
	2001		_
	2000		250
	1999		_
PRODUITS.all			

Figure 17 – Opérations classiques

La constitution d'un module source est produite automatiquement par l'ETL lors d'une phase dite d'importation, qui consiste à extraire les descriptions de la structure de la source (métadonnées) et d'en donner une représentation relationnelle dans l'outil.

La définition du module cible s'effectue en deux temps :

— **définition du schéma cible**. La construction d'un schéma multidimensionnel est effectuée au travers des faits, des dimensions ;

— **définition des processus d'extraction**. Ces derniers sont mis en place grâce aux *mappings* qui permettent d'expliciter le processus d'extraction et de transformation à effectuer pour alimenter les objets cibles (faits et dimensions) à partir des objets sources (tables), de séquences et de transformations PL/SQL.

La figure **19** présente le *mapping* d'une dimension MAGASINS_DM à partir de deux relations nommées GEOGRAPHIE et MAGASINS. Ce *mapping* est constitué d'un opérateur de jointure qui combine les enregistrements des deux relations sources, et qui alimente la dimension attribut par attribut. D'autre part, nous visualisons également le schéma de cette dimension ; elle est constituée de deux hiérarchies hiera_enseigne et hiera_geographie.

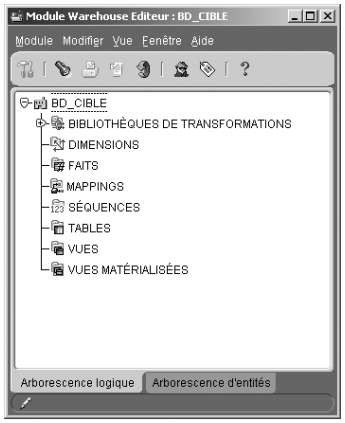


Figure 18 – Objets constituant un module

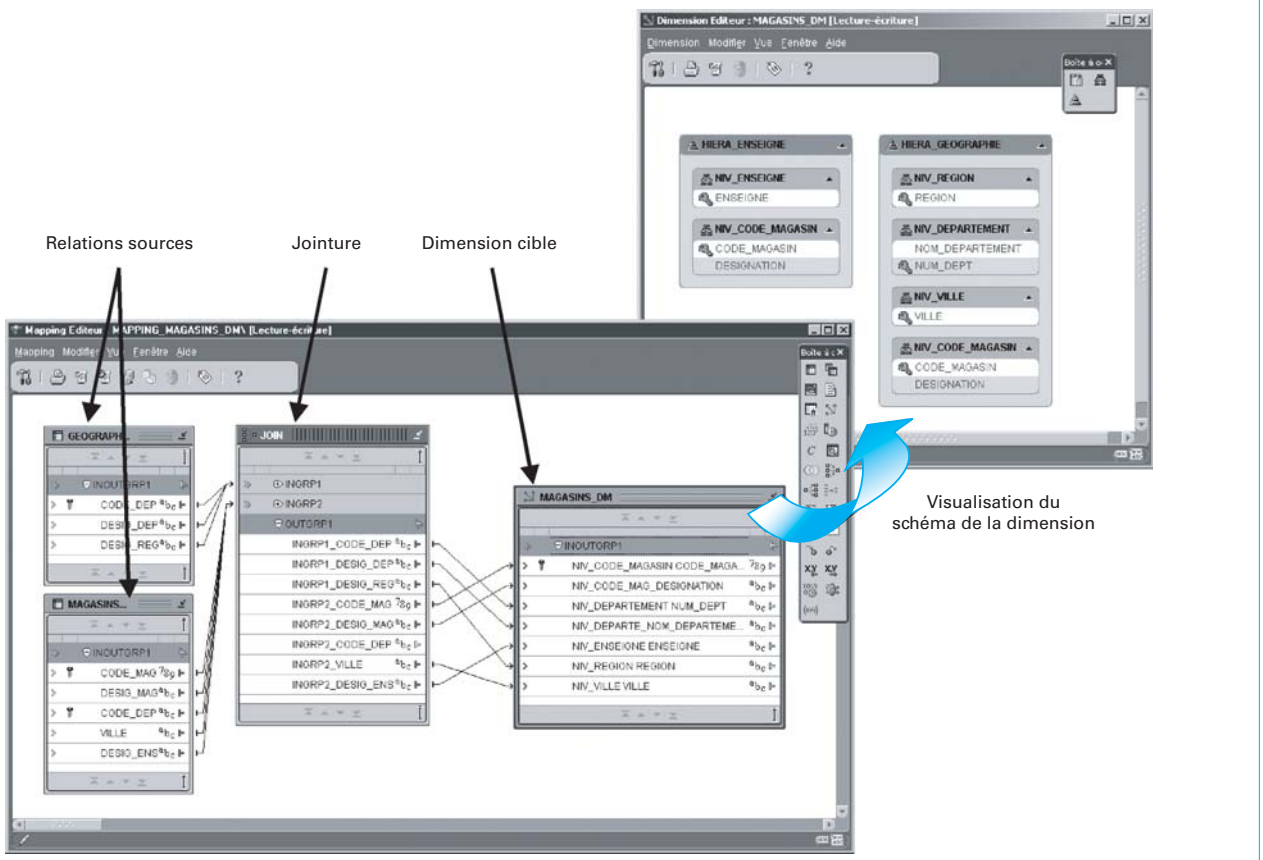


Figure 19 – Exemple de *mapping* d'une dimension MAGASINS_DM

Entrepôts de données

par **Claude CHRISMENT**
Professeur à l'université Toulouse-3

Geneviève PUJOLLE
Maître de conférences à l'université Toulouse-1

Franck RAVAT
Maître de conférences à l'université Toulouse-1

Olivier TESTE
Maître de conférences à l'université Toulouse-3

et **Gilles ZURFLUH**
Professeur à l'université Toulouse-1

Bibliographie

Références

[1] AGRAWAL (R.), GUPTA (A.) et SARAWAGI (S.). – *Modeling Multidimensional Databases*. Research Report, IBM Almaden Research Center, San Jose, Californie (1995). Paru dans les actes de ICDE'97.

[2] GYSSEN (M.) et LAKSHMANAN (L.V.S.). – *A Foundation for Multi-Dimensional Databases*. 23rd International Conference on Very Large Data Bases – VLDB'97, Athènes, Grèce (25 au 29 août 1997).

Ouvrages

INMON (W.H.). – *Building the Data Warehouse*. Wiley (2002).

KIMBALL (R.) et ROSS (M.). – *Entrepôts de données. Guide pratique de modélisation dimensionnelle*. Vuibert (2003).

Dans les Techniques de l'Ingénieur

CHRISMENT (C.) et ZURFLUH (G.). – *Bases de données. Introduction*. [H 3 800], Technologies logicielles – Architectures des systèmes (1995).

CHRISMENT (C.), PUJOLLE (G.) et ZURFLUH (G.). – *Bases de données orientées objets*. [H 3 840], Technologies logicielles – Architectures des systèmes (1992).

CHRISMENT (C.), PUJOLLE (G.) et ZURFLUH (G.). – *Bases de données réparties*. [H 3 850], Technologies logicielles – Architectures des systèmes (1993).

CHRISMENT (C.), LUGUET (J.), PUJOLLE (G.) et ZURFLUH (G.). – *Bases de données relationnelles*. [H 2 038], Technologies logicielles – Architectures des systèmes (1997).

CHRISMENT (C.), PUJOLLE (G.) et ZURFLUH (G.). – *Langages de bases de données : SQL et les évolutions vers l'objet*. [H 3 128], Technologies logicielles – Architectures des systèmes (1999).

THÉVENIN (J.-M.) et VIALLET (F.). – *Architecture des systèmes de gestion de bases de données*. [H 2 918], Technologies logicielles – Architectures des systèmes (1996).

ZIGHED (D.A.) et RAKOTOMALALA (R.). – *Extraction de connaissances à partir de données (ECD)*. [H 3 744], Technologies logicielles – Architectures des systèmes (2002).

GIROUX (P.). – *Langage UML : développement de logiciel et modélisation visuelle*. [H 3 238], Technologies logicielles – Architectures des systèmes (2004).

CHAHUNEAU (F.). – *XML*. [H 7 148], Documents numériques – Gestion de contenu (2001).

Logiciels

Cette liste n'est pas exhaustive.

Oracle
http://www.oracle.com

■ **Requêteurs**

Impromptu, Cognos
http://www.cognos.com/impromptu

Business Objects, Business Objects SA
http://www.businessobjects.com

Discoverer, Oracle
http://www.oracle.com/technology/products/discoverer

■ **Requêteurs avec rapport *via* un navigateur**

WebIntelligence, Business Objects SA
http://www.businessobjects.com

Impromptu Web Reports, Cognos
http://www.cognos.com/impromptu

■ **Outils d'analyse OLAP**

Oracle OLAP, Oracle
http://www.oracle.com/technology/products/bi/olap/olap.html

Microsoft SQL Server, Microsoft
http://www.microsoft.com/sql

■ **Outils graphiques d'analyse OLAP**

PowerPlay, Cognos
http://www.cognos.com/products/business_intelligence/analysis

Business Objects, Business Objects SA
http://www.businessobjects.com

■ **Outils ETL**

Oracle Warehouse Builder, Oracle
http://www.oracle.com/technology/products/warehouse

DataStage, Ascential
http://www.ascential.com/products/datastage.html