République Algérienne Démocratique Et Populaire

Ministère De L'enseignement Supérieur Et De La Recherche Scientifique

Université SAAD DAHLAB de Blida

Faculté Des Sciences

Département D'informatique

Mémoire De Fin D'études

Pour Obtention Du Diplôme De Master En Informatique

Thème

Conception Et Réalisation Dun Système Personnalisé Dans L'analyse En Ligne (OLAP)

présenté par :

Mr. KHATIRI MOHAMED

Mr. STAMBOULI MOHAMED FATEH

Dirigé par :

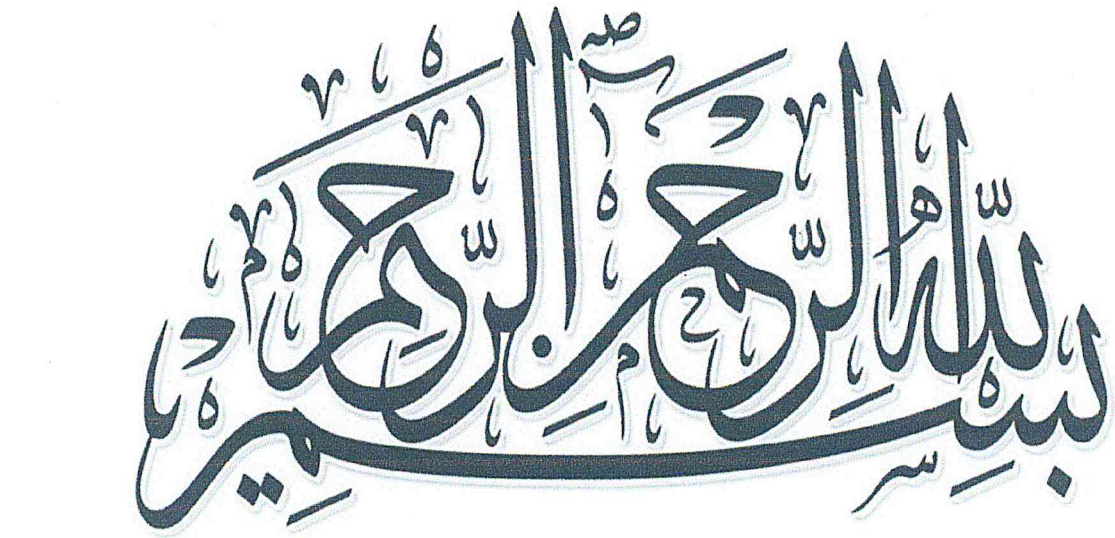
Mlle. ALIMAZIGHI HADJER

Encadré par :

M.Oukid

- Président - Mr BALA
- Membre de jury = Mr NAHAL.

PROMOTION 2011-2012


بسم الله الرحمن الرحيم

آية الكرسي

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

اللَّهُ لَا إِلَٰهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ

لَا تَأْخُذُهُ سِنَةٌ وَلَا نَوْمٌ

لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ

مَنْ ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلَّا بِإِذْنِهِ

يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ

وَلَا يُحِيطُونَ بِشَيْءٍ مِنْ عِلْمِهِ إِلَّا بِمَا شَاءَ

وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالْأَرْضَ

وَلَا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ ﴿٢٥٥﴾

لَا إِكْرَاهَ فِي الدِّينِ قَد تَّبَيَّنَ الرُّشْدُ مِنَ الْغَيِّ

فَمَن يَكْفُرْ بِالطَّاغُوتِ وَيُؤْمِن بِاللَّهِ

فَقَدِ اسْتَمْسَكَ بِالْعُرْوَةِ الْوُثْقَىٰ

لَا انفِصَامَ لَهَا وَاللَّهُ سَمِيعٌ عَلِيمٌ ﴿٢٥٦﴾

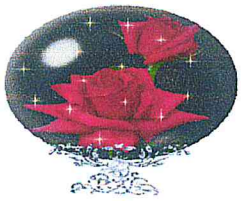
اللَّهُ وَلِيُّ الَّذِينَ آمَنُوا يُخْرِجُهُم

مِّنَ الظُّلُمَاتِ إِلَى النُّورِ

وَالَّذِينَ كَفَرُوا أَوْلِيَاؤُهُمُ الطَّاغُوتُ

يُخْرِجُونَهُم مِّنَ النُّورِ إِلَى الظُّلُمَاتِ

أُولَٰئِكَ أَصْحَابُ النَّارِ هُمْ فِيهَا خَالِدُونَ ﴿٢٥٧﴾



Remerciements

Nous tenons tout d'abord à remercier Dieu tout puissant pour nous avoir guidés vers le bon chemin de la lumière et du savoir et pour nous avoir donné du courage et de la volonté afin de pouvoir réaliser ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à notre promotrice Mlle.AliMazighi Hadjer pour l'orientation de ce mémoire pour son aide, sa patience et la confiance qu'elle nous a accordée.

Nous tenons à remercier : les membres du jury pour avoir accepte d'apprécier notre travail.

Nos sincères remerciements vont également à tous les enseignants du département d'informatique de l'université SAAD DAHLAB de BLIDA, et à tous les enseignants qui ont participé à notre formation.

Enfin nous remercions tous ceux qui ont contribué de prés ou de loin à la réalisation de ce mémoire.

Dédicace

C'est avec un très grand plaisir que je dédie ce modeste travail aux personnes les plus chères au monde après mon DIEU tout puissant pour leurs bonnes intentions à mon égare,

A Mes chers parents que Dieu les garde et les préserve, pour leurs tendresses, leurs sacrifices et pour leurs précieux conseils. Pour leurs soutiens dans les moments difficiles. Sachez que Vous êtes extraordinaires et que, c'est grâce à Vous que je suis là ou j'en suis. Avec ma profonde affection,

A Mes grands parents que je leur souhaite une longue vie,

A Mes Sœurs au leurs souhaitant une vie heureuse,

À tout les membres de ma famille notamment KHATIRI et BENKALI,

A Tous mes collègues et à mes amis sans exceptions,

Je dédie enfin ce modeste travail à ceux qui me connaissent, et surtout à ceux qui m'aiment,

Mohamed KTR

Dédicace

Je dédie ce mémoire de fin d'études

A

Mon très cher père et ma très chère mère

en témoignage de ma reconnaissance envers le soutien, les sacrifies et tous les efforts qu'ils ont fait pour mon éducation ainsi que ma formation

A

mes frères et soeurs, pour leur soutien et leurs encouragements. Que dieu vous protège et vous prête bonne santé et longue vie.

A

tous ceux qui ont une relation de proche ou de loin avec la réalisation du présent rapport.

A

tous ceux qui me sont chers.

Fateh...

Résume :

Ces dernières années, on assiste à une forte inflation de volume d'information en raison de l'accroissement des capacités de calcul et de stockage, dans ce contexte de surcharge d'informations, il ya a lieu de développer des outils informatique pour faciliter la recherche et l'éxtration rapide de l'information pertinente. Un de ces outils est la personnalisation qui consiste à répondre au mieux aux attentes de chaque utilisateur.

Le travail à réaliser s'inscrit dans le cadre de la personnalisation de l'information dans le contexte de l'interrogation des entrepôts de données.

les requêtes OLAP permettent l'analyse interactive d'un gros volume de donnée multidimensionnelle, ces données proviennent d'un entrepôt de données, les résultats massifs délivrés en réponse a ces requêtes, génèrent une surcharge informationnelle dans la quel il est difficile de distinguer l'information pertinente d'une information secondaire.

Ce travail consiste en la conception d'un système de personnalisation des requêtes MDX selon des critéres de l'analyse.

Pour réalisé notre objectif, nous avons développé un système en language JAVA sous la plate forme NetBeans, il est accéssible via des pages Web JSP. Afin d'implémenter notre système, nous avons utilisé un entrepôt de données de Microsoft utilisé dans la documentation en ligne de SQL Server

Mot-clés : Entrepôt de données, Presonnalisation, Recommendation personnalisé, Requûtes MDX, Préférences.

Abstract:

Because of the big boom of data in our days the information search became very slow and more complicated, this why the development of tools that facilitate the search and the extraction of information became crucial. In our case we use the personalization that allows better responding of every user's needs.

Our work is to personalize the information in the context of the data warehouse requesting. The OLAP queries allow an interactive analyze of a huge multidimensional volume data, this data became from a data warehouse. The massive results issued as response of these queries generate an informational overload in which it's difficult to distinguish the relevant information from the secondary.

This work consists of the conception of a personalization system of the MDX queries by the analytics criteria.

To reach our goal we develop a system with the JAVA language on the netbeans platform, it's reachable with JSP web pages. We used also a data warehouse of Microsoft used in the online documentation of SQL Server.

Keywords : Data Warehouse, Personalization, Recommendation, MDX Queries, Preferences.

الملخص:

بسبب الازدهار الكبير من البيانات في أيامنا أصبح البحث عن المعلومات بطيء و معقد أكثر ، لهذا أصبح تطوير الأدوات التي تسهل عملية البحث واستخراج المعلومات مهم جدا

في حالتنا نحن نستخدم نظام التشخيص الذي يسمح بتقديم إجابة دقيقة لكل مستخدم

عملنا هو تشخيص نظام المعلومات في مستودع البيانات ذات الكم الضخم من البيانات المتعددة الابعاد حتى يستطيع المستخدم تمييز المعلومة الهامة من المعلومة الثانوية

هذا العمل يتكون من مفهوم نظام تخصيص استفسارات متعددة الأبعاد التعبير و هدفنا هو تطوير نظام لتقديم بيانات دقيقة و هامة للمستخدم.

كلمات بحث مفتاحية:

استعلامات متعددة الابعاد تشخيص المعلومات مستودع بيانات تفضيلات

Introduction générale

La partie 01: L'état de l'art.

Chapitre 01 : Entrepôt de données et les systèmes d'OLAP.

Chapitre 02 : Les systèmes de personnalisation d'OLAP.

La partie 02: Conception du système.

Chapitre 03: Proposition.

Chapitre 04: La conception.

La partie 03 : Mise en œuvre.

Chapitre 05 : Implémentation.

Liste des figures

○ Entrepôt de données et les systèmes d'OLAP

○ Les systèmes de personnalisation d'OLAP

- **Proposition**

- **Conception**

- **Implémentation**

Liste des tableaux

Introduction Générale

Introduction Générale :

Aujourd'hui la situation de marché est telle, que toute entreprise possède un système d'information décisionnel. Celui-ci pourra être plus ou moins complexe et élaboré, allant du simple tableau sous Excel à l'entrepôt de données.

Le service attendu d'un système décisionnel est d'avoir une vue synthétique de l'entreprise, ceci dans le but de pouvoir prendre des décisions stratégiques concernant les directions et engagement à prendre.

En général, le marché des solutions décisionnelles basées sur les entrepôts de données et les outils OLAP (OnLine Analytical Processing).

L'OLAP a fait l'objet de nombreux travaux de recherche et devenue une technologie bien maitrisée et largement utilisée dans les systèmes d'information décisionnelle. Son apport est de rendre possible la navigation interactive dans les données, la visualisation rapide des informations et d'exploration de la structure multidimensionnelle des données.

Une des principales limites de ce système est leur incapacité à discriminer les utilisateurs en fonction de leurs centres d'intérêt, de leurs préférences. Donc nous essayons à travers de ce projet d'exploiter les algorithmes de personnalisation pour recommander des requêtes pertinentes et d'aider l'utilisateur dans son exploitation de cube OLAP.

➢ Problématique:

Les entrepôts de données stockent de gros volumes de données, multidimensionnelles, consolidées et historiées dans le but d'être explorées et analysées par différents utilisateurs. L'exploration de données est un processus de recherche d'informations pertinentes au sein d'un ensemble de données. L'ensemble de données à explorer est un cube de données qui est un extrait de l'entrepôt de données que les utilisateurs interrogent en lançant des séquences de requêtes OLAP. Cependant Les cubes sont souvent de taille importante et il est difficile pour l'utilisateur de naviguer dedans et de dégager l'information intéressante.

➢ Objectifs:

Donc il y a lieu de développer des outils informatiques pour faciliter la recherche et l'extraction rapide de l'information pertinente. La personnalisation dans le cadre des entrepôts de données présente un réel intérêt dans un contexte où les analyses qui permettent la prise de décision sont réalisées par l'utilisateur lui-même (Le décideur), on utilise l'approche qui est venue palier aux insuffisances de la méthode introduite initialement par **Börzsönyi et al. (2001)**, on utilise l'operateur skyline appliqué par l'algorithme déviser pour régner.

➢ Plan de mémoire:

La partie 01 : L'état de l'art.

Le Chapitre 01 : Les entrepôts de données et les système OLAP.

Dans ce chapitre on va évoquer les principaux concepts liées à la conception et à l'exploitation des entrepôts de données, ou nous avons vu les caractéristiques des données de l'entrepôt et l'architecture associée.

Le Chapitre 02 : Les systèmes de personnalisation d'OLAP.

Dans ce chapitre, on va présenter un état de l'art général des approches de personnalisation des systèmes OLAP et les travaux réalisés. Puis nous exposons les constatations de cet état de l'art qui vont conduire à nos propositions.

La partie 02 : Conception du système.

Le Chapitre 03 : Proposition.

Dans ce chapitre, on va présenter d'abord l'opérateur Skyline ainsi que la problématique à laquelle il répond. Ensuite on va expliquer les algorithmes de calcul du Skyline.

Le chapitre 04 : Conception.

Dans ce chapitre on va développer notre contribution sur la personnalisation des réponses aux requêtes OLAP. On va montrer l'architecture globale de notre système, nous formalisons par la suite le problème de personnalisation traité, puis nous détaillons notre approche en donnant l'algorithme proposés.

La parite 03 : Mise en œuvre.

Le chapitre 05 : Réalisation.

Cette parite décrit l'environnement et les outils de réalisation de notre prototype, cela en présentant premier lieu les choix effectués pour sa réalisation.

Partie

I

ETAT DE L'ART

- Entrepôt de données et les systèmes d'OLAP
- Les systèmes de personnalisation d'OLAP

Entrepôts de données et les systèmes d'OLAP

1 Introduction :

Aujourd'hui la situation de marché est telle, que toute entreprise possède un système d'information décisionnel. Celui-ci pourra être plus ou moins complexe et élaboré, allant du simple tableau sous Excel à l'entrepôt de données.

Le service attendu d'un système décisionnel est d'avoir une vue synthétique de l'entreprise, ceci dans le but de pouvoir prendre des décisions stratégiques concernant les directions et engagement à prendre.

En général, le marché des solutions décisionnelles basées sur les entrepôts de données et les outils OLAP.

2 Système Décisionnel :

L'**informatique décisionnelle** désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données d'une entreprise en vue d'offrir une aide à la décision et de permettre aux responsables de la stratégie d'entreprise d'avoir une vue d'ensemble de l'activité traitée.

Ce type d'application utilise en règle générale un entrepôt de données (ou datawarehouse en anglais) pour stocker des données transverses provenant de plusieurs sources hétérogènes (techniquement Excel, DB2, Oracle, SQL SERVEUR..., et fonctionnellement RH, Production, Compta, finance...) et fait appel à des traitements par lots pour la collecte de ces informations.

L'informatique décisionnelle s'insère dans l'architecture plus large d'un système d'information. Néanmoins l'informatique décisionnelle n'est pas un concept concurrent du management du système d'information. Au même titre que le management relève de la sociologie et de l'économie, la gestion par l'informatique est constitutive de deux domaines radicalement différents que sont le management et l'informatique. Afin d'enrichir le concept avec ces deux modes de pensées, il est possible d'envisager un versant orienté ingénierie de l'informatique portant le nom

d'informatique décisionnelle, et un autre versant servant plus particulièrement les approches de gestion appelé management du système d'information.

Le **système décisionnel** regroupe un ensemble d'informations et d'outils mis à la disposition des décideurs pour supporter de manière efficace la prise de décision **[COD 93] [INM 94].**

L'**architecture** des systèmes décisionnels met en jeu quatre éléments essentiels : les sources de données, l'entrepôt de données, les magasins de données et les outils d'analyse et d'interrogation.

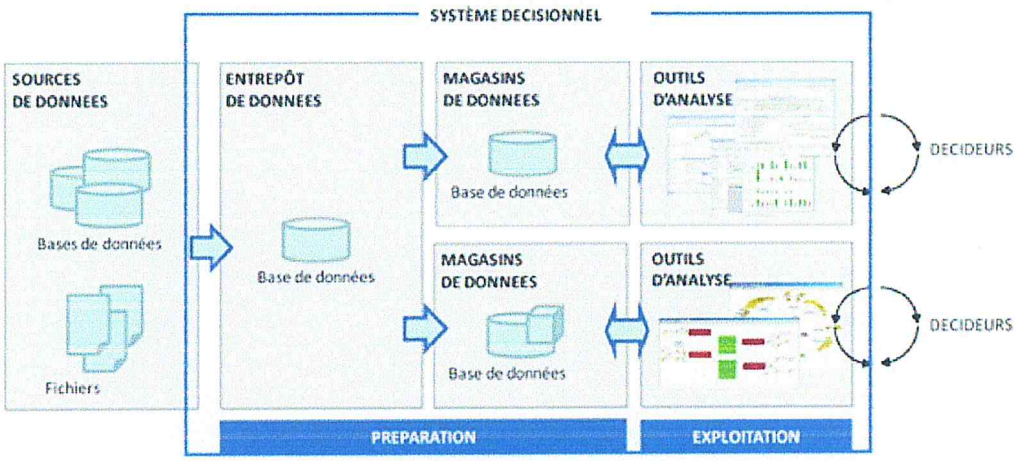


Figure.01 : Architecture des systèmes décisionnels [TES 00]

- Les **sources de données** sont nombreuses, variées, distribuées et autonomes. Elles peuvent être internes (bases de production) ou externes (Internet, bases des partenaires) à l'entreprise.
- L'**entrepôt de données** est le lieu de stockage centralisé des informations utiles pour les décideurs. Il met en commun les données provenant des différentes sources et conserve leurs évolutions.
- Les **magasins de données** sont des extraits de l'entrepôt orientés sujet. Les données sont organisées de manière adéquate pour permettre des analyses rapides à des fins de prise de décision.

- Les **outils d'analyse** permettent de manipuler les données suivant des axes d'analyses. L'information est visualisée à travers d'interfaces interactives et fonctionnelles dédiées à des décideurs souvent non informaticiens (directeurs, chefs de services,…) **[TES 00]**.

Dans un système d'information décisionnel, l'utilisateur final formulera des questions du type:

- Comment se comporte le produit X par rapport au produit Y ?
- Et par rapport à l'année dernière ?
- Quel type de client peut bien acheter mon produit Z ?

Ces exemples permettent de mettre en évidence les faits suivants :

- les questions doivent pouvoir être formulées dans le langage de l'utilisateur en fonction de son « métier », c'est-à-dire de son secteur d'activité (service marketing, service économique, service gestion des ressources humaines, ...).
- la prévision des interrogations est difficile car elles sont du ressort de l'utilisateur. De plus, ses questions vont varier selon les réponses obtenues: si le produit X s'est vendu moins bien que l'année précédente, il va être utile d'en comprendre les raisons et donc de détailler les ventes du produit X (par région, par type de magasin, ...).

Ce qui caractérise d'abord les besoins, c'est donc la possibilité de poser une grande variété de questions au système, certaines prévisibles et planifiées comme des tableaux de bord et d'autres imprévisibles. Si des outils d'édition automatiques préprogrammés peuvent être envisagés, il est nécessaire de permettre à l'utilisateur d'effectuer les requêtes qu'il souhaite, par lui-même, sans intervention de programmeurs. Deux contraintes apparaissent alors immédiatement: la simplicité du modèle des données, la performance malgré les grands volumes **[*KIM 97*]**.

3 L'entrepôt de données :

Dans les années 90, Bill Inmon, est l'un des premiers à avoir employé le terme d'entrepôt de données. Ce dernier le définit comme : « Un entrepôt de données est une collection de données portant sur des sujets touchant une organisation, intégrée, variant dans le temps, et non-volatile pour supporter le processus de prise de décision d'une organisation». **[INM 96].**

- Orientée sujet : les données constituent des granules d'information concernant des sujets d'analyse plutôt que les opérations de gestion se déroulant au sein de L'entreprise.

- Intégrée : les données sont centralisées dans un entrepôt à partir d'un ensemble de sources de données variées. Les données sont fusionnées et agencées au sein d'une vision cohérente.

- Variant selon le temps : Toutes les données d'un entrepôt sont identifiées par des périodes temporelles spécifiques. On parle d'historisation des données **[KIM 96], [INM 96], [TES 00].**

- Non volatile : Les données d'un entrepôt sont stables. Il est possible d'ajouter des données, mais on ne modifie pas les données déjà intégrées. Il est toutefois possible de les archiver.

De son cote, Ralph Kimball a fourni une définition plus simple d'un entrepôt de données, mais qui n'en est pas moins précise : « un entrepôt de données est une copie des données transactionnelles d'une entreprise structurée de manière spécifique pour l'interrogation et l'analyse » **[KIM 96].**

Un entrepôt de données est l'espace de stockage centralise d'un extrait des sources de données pertinentes pour les décideurs. Son organisation doit faciliter la gestion des données sous la forme d'une vision unifiée et doit permettre la conservation des évolutions nécessaires pour les prises de décisions.

Un entrepôt de données offre une vision uniforme des données qui seront extraites en fonction de besoins d'analyse pour alimenter les magasins de données.

3.1 L'intérêt de l'entrepôt de données :

1. Vision transversale de l'entreprise
2. Intégration de différentes bases
3. Données non volatiles (pas de suppression)
4. Historisation
5. Organisation vers prise de décision

3.2 Architecture d'un système décisionnel :

	Systèmes opérationnels	Entrepôt de données
But	Exécution d'un processus métier	Evaluation d'un processus métier
Interaction avec l'utilisateur	Insertion, Modification, Interrogation, Suppression	Interrogation
Données	Courantes	Courantes et Historiques
Usage	Support de l'opération de l'entreprise	Support de l'analyse de l'entreprise
Requêtes	Simples, Prédéterminées	Complexes, Ad-hoc
Type d'utilisateur	Employé	Décideur
Principe de conception	Troisième forme normale	Conception multidimensionnelle

Tableau.01 : Base de données vs Entrepôt de données [BOU 09]

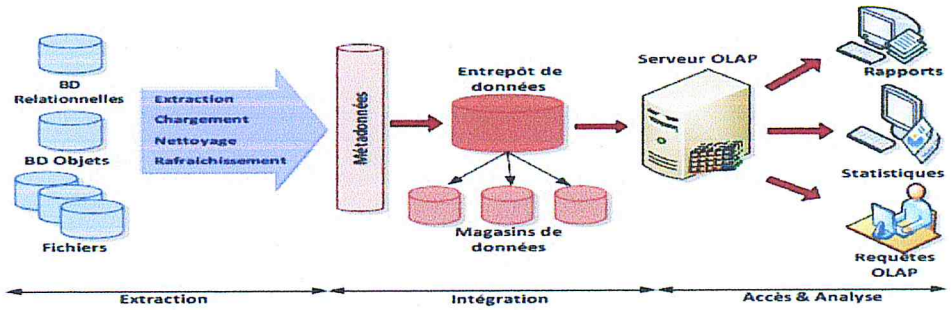


Figure.02 : Architecture des systèmes décisionnels [BOU 09]

L'entrepôt de données joue un rôle stratégique dans la vie d'une entreprise. Il stocke des données pertinentes aux besoins de prise de décision en provenance des systèmes opérationnels de l'entreprise et d'autres sources externes. A la différence d'une base de données classique supportant des requêtes transactionnelles de type OLTP (On-Line Transaction Processing), un entrepôt de données est conçu pour supporter des requêtes de type OLAP (On-Line Analytical Processing). L'interrogation est l'opération la plus utilisée dans le contexte d'entrepôt de données où la mise à jour consiste seulement à alimenter l'entrepôt. Le tableau.01 montre la différence entre un système opérationnel doté d'une base de données classique et un entrepôt de données. Le processus de construction d'un entrepôt de données est composé de trois principales phases (Figure.02) :

(1) extraction des données à partir des différentes sources,
(2) organisation et intégration des données dans l'entrepôt et
(3) accès aux données intégrées et analyse de ces dernières dans une forme efficace et flexible.

Dans la première et la deuxième phase, les données issues de différentes sources de données sont extraites, nettoyées et intégrées dans l'entrepôt de données. Les métadonnées contiennent des informations utiles sur la création, l'utilisation et la gestion de l'entrepôt. Durant la troisième phase, un serveur OLAP se charge de présenter les informations demandées par les utilisateurs sous plusieurs formes : tableaux, rapports, statistiques, etc… **[BOU 09].**

4 Concepts fondamentaux :

Nous avons vu précédents ce que comprenait un environnement décisionnel et qu'il avait comme concept central l'entrepôt de données ou le Data Warehouse. Intéressons-nous maintenant les concepts fondamentaux de l'informatique décisionnelle.

4.1 Concept de fait :

Le **fait** modélise le sujet de l'analyse. Un fait est formé de mesures correspondant aux informations de l'activité analysée **[TES 00].**

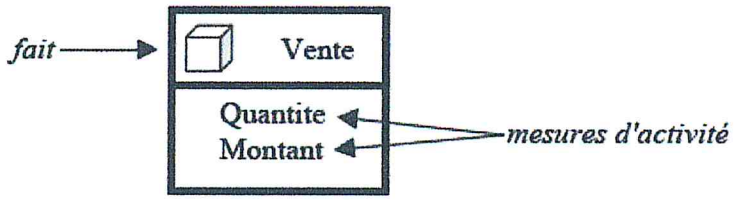


Figure.03: Exemple de faits (TES 00).

4.2 Concept de dimensions :

Une **dimension** modélise une perspective de l'analyse. Une dimension se compose de paramètres correspondant aux informations faisant varier les mesures de l'activité.

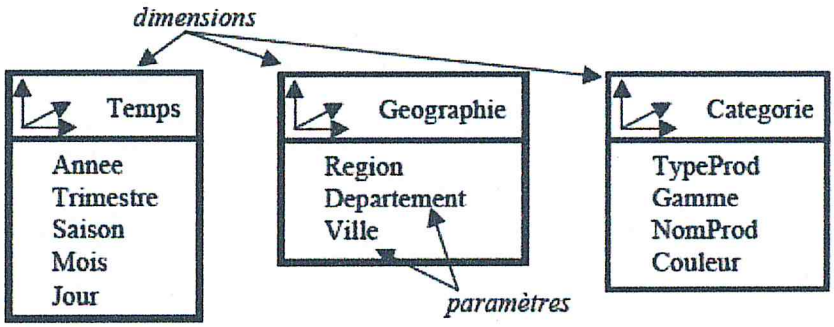


Figure.04 : Exemple de dimension (TES 00).

On part du principe que les données sont des faits à analyser selon plusieurs dimensions. Il est ainsi possible de réaliser une structure de données simple qui correspond à ce besoin de modélisation multidimensionnelle. Cette structure est constituée du fait central et des dimensions.

Au niveau logique cela peut se traduire par trois modèles différents : en étoile, en flocon de neige ou en constellation.

4.3 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 **[KIM 96]).**

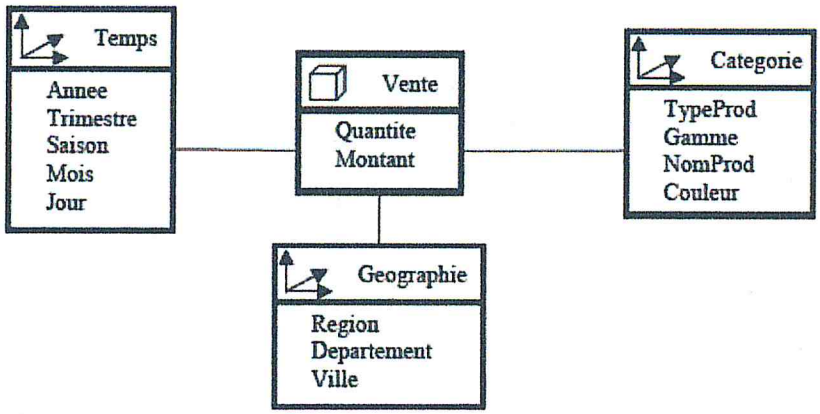


Figure.05: Exemple d'une modélisation en étoile (TES 00).

4.4 Modèle en flocon:

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 **[TES 00].**

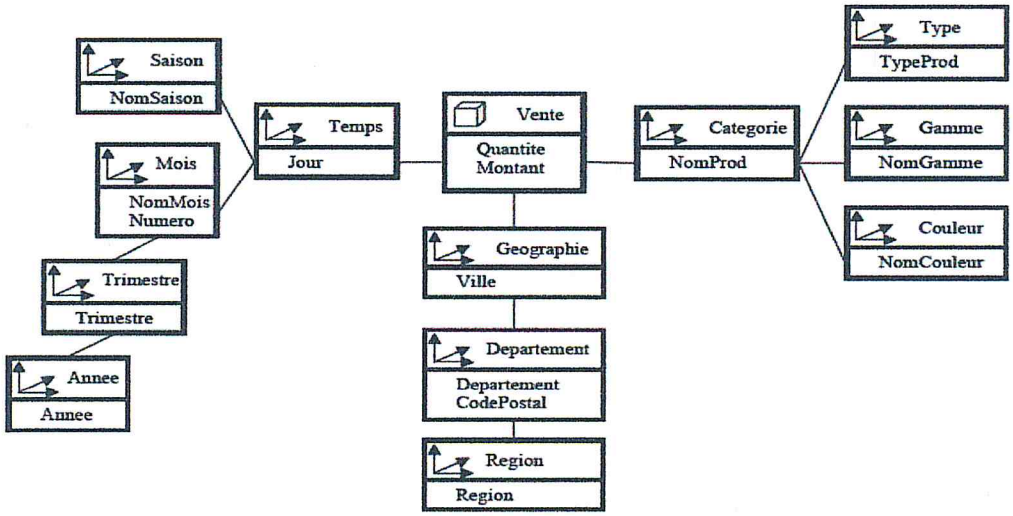


Figure.06: Exemple d'une modélisation en flocon (TES 00).

4.5 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. **[TES 00].**

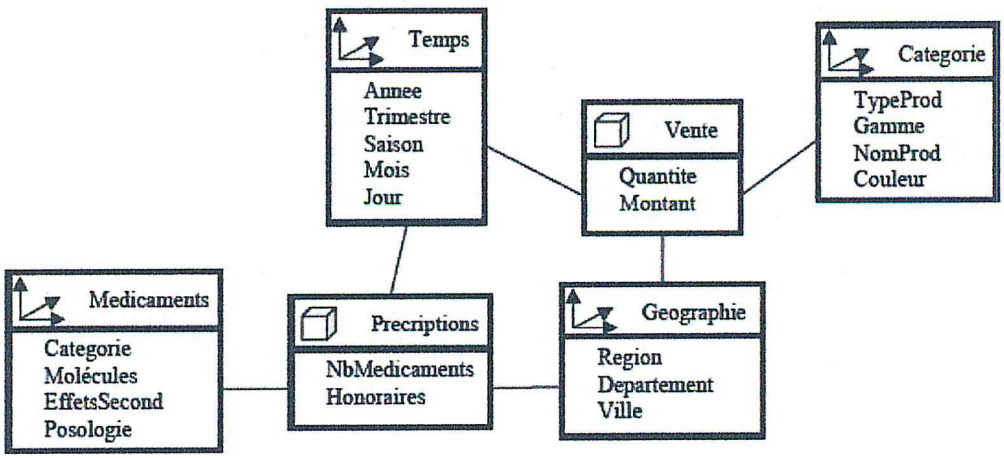


Figure.07: Exemple d'une modélisation en constellation (TES 00).

4.6 Datamart :

Un datamart (ou magasin de données) est une vue partielle du datawarehouse mais orientée métier. C'est un sous-ensemble du datawarehouse contenant des informations se rapportant à un secteur d'activité particulier de l'entreprise ou à un métier qui y est exercé. Il se situe en aval du datawarehouse et est alimenté par celui-ci. On peut donc créer plusieurs datamart correspondant au différent besoin des utilisateurs.

Cela permet de réduire le nombre d'opération sur les bases de production. De plus cela permet d'offrir aux utilisateurs un outil spécifiquement adapté à leurs besoins. Cet outil sera plus petit et permettra donc un accès plus rapide à l'information.

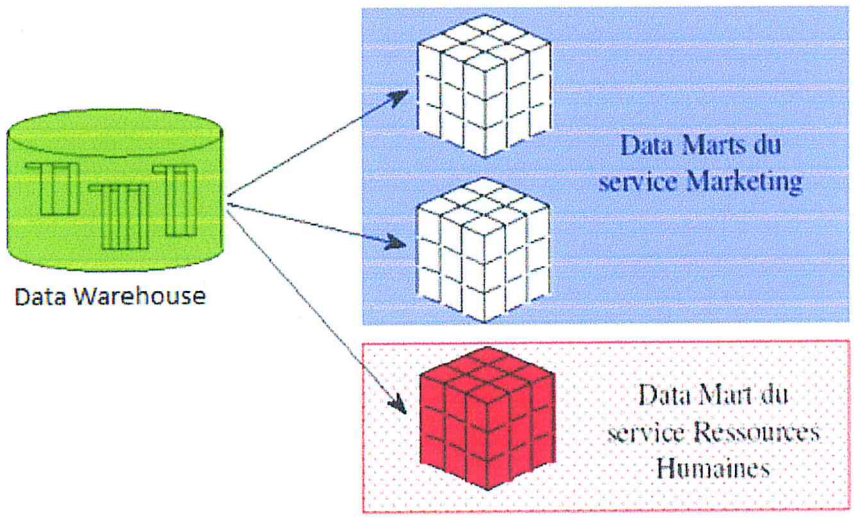


Figure.08 : Un datawarehouse vs des datamarts (TES 00).

5 OLAP :

En informatique, et plus particulièrement dans le domaine des bases de données, le traitement analytique en ligne (anglais online analytical processing abr. OLAP) est un type d'application informatique orienté vers l'analyse sur-le-champ d'informations selon plusieurs axes, dans le but d'obtenir des rapports de synthèse

tels que ceux utilisés en analyse financière. Les applications de type OLAP sont couramment utilisées en informatique décisionnelle, dans le but d'aider la direction à avoir une vue transversale de l'activité d'une entreprise.

Ce type d'application s'oppose au traitement de transactions en ligne (anglais online transaction processing abr. OLTP) qui s'inscrit dans un système opérationnel, c'est-à-dire dédié aux métiers de l'entreprise pour les assister dans leurs tâches de gestion quotidiennes.

Ce terme a été défini par Edgar Frank Codd en 1993 au travers de 12 règles que doit respecter une base de données si elle veut adhérer au concept OLAP :

- **Multi dimensionnalité** - le modèle OLAP est multidimensionnel par nature.
- **Transparence** - l'emplacement physique du serveur OLAP est transparent pour l'utilisateur.
- **Accessibilité** - l'utilisateur OLAP dispose de l'accessibilité à toutes les données nécessaires à ses analyses.
- **Stabilité** - la performance des interrogations reste stable indépendamment du nombre de dimensions.
- **Client-Serveur** - le serveur OLAP s'intègre dans une architecture client-serveur.
- **Dimensionnement** - le dimensionnement est générique afin de ne pas fausser les analyses.
- **Gestion complète** - le serveur OLAP assure la gestion des données clairsemées.
- **Multi-Utilisateurs** - le serveur OLAP offre un support multi-utilisateurs (Gestion des mises à jour, intégrité, sécurité).
- **Inter Dimension** - le serveur OLAP permet la réalisation d'opérations inter-dimensions sans restriction.
- **Intuitif** - le serveur OLAP permet une manipulation intuitive des données.
- **Flexibilité** - la flexibilité (ou souplesse) de l'édition des rapports est intrinsèque au modèle.

❖ **Analyse sans limites** - le nombre de dimensions et de niveaux d'agrégation possibles est suffisant pour autoriser les analyses les plus poussées.

Ce concept a été appliqué à un modèle virtuel de représentation de donnée appelé cube ou hypercube OLAP qui peut être mis en œuvre de différentes façons.

Le but de l'**OLAP** est de permettre une analyse multidimensionnelle sur des bases de données volumineuses afin de mettre en évidence une analyse particulière des données (il est l'objet d'un questionnement particulier).

Grâce à l'OLAP, les utilisateurs peuvent créer des représentations multidimensionnelles (appelées **hypercubes** ou « cubes OLAP ») selon les critères qu'ils définissent afin de simuler des situations.

Un hypercube OLAP (ou cube OLAP) est une représentation abstraite d'informations multidimensionnelles exclusivement numérique utilisée par l'approche OLAP. Cette structure est prévue à des fins d'analyses interactives par une ou plusieurs personnes (souvent ni informaticiens ni statisticiens) du métier que ces données sont censées représenter.

Les cubes OLAP ont les **caractéristique**s suivantes :

- obtenir des informations déjà agrégées selon les besoins de l'utilisateur.
- simplicité et rapidité d'accès
- capacité à manipuler les données agrégées selon différentes dimensions
- un cube utilise les fonctions classiques d'agrégation : min, max, count, sum, avg, mais peut utiliser des fonctions d'agrégations spécifiques.

L'hypercube OLAP donne accès à des **fonctions d'extraction de l'information** (pour visualisation, analyse ou traitement), et à des fonctions de requête en langage MDX (comparable à SQL pour une base de données relationnelles).

✓ **Roll-up** Passage de mesures détaillées `a résumées en remontant dans la hiérarchie de la dimension.

✓ **Drill-down** Descendre dans la hiérarchie de la dimension.

- ✓ **Rotate** Rotation des axes du cube pour fournir une vue alternative des données.
- ✓ **Slicing** Extraction d'une tranche d'informations : S'élection d'une dimension pour passer à un sous-cube.
- ✓ **Dice** Extraction d'un bloc de données : S'élection de deux ou plusieurs dimensions.
- ✓ **Drill-accross** Exécution de requêtes impliquant plus d'un cube ayant une dimension commune.
- ✓ **Drill-through** Passage d'une mesure à l'autre ou d'un membre d'une dimension à un autre. **[Rav, al 07].**

5.1 Implémentation

Le concept de l'hypercube OLAP peut être utilisé par le biais de différentes interfaces selon les besoins et les capacités. Il en existe actuellement trois.

5.2 ROLAP (Relational OLAP)

Dans les systèmes relationnels OLAP, l'entrepôt de données utilise une base de données relationnelle. Le stockage et la gestion de données sont relationnels. Toutefois, le modèle relationnel requiert des extensions pour supporter les requêtes d'analyses multidimensionnelles du niveau d'application. Le moteur ROLAP traduit dynamiquement le modèle logique de données multidimensionnel M en modèle de stockage relationnel R (la plupart des outils requièrent que la donnée soit structurée en utilisant un schéma en étoile ou un schéma en flocon de neige). L'efficacité du résultat de la requête est le facteur dominant pour la performance et le passage à l'échelle global du système. Ainsi, les stratégies d'optimisation représentent le point principal qui distingue les systèmes ROLAP existants. **[DSB 99] [Sat03].**
La figure.08 montre une architecture pour le serveur ROLAP.

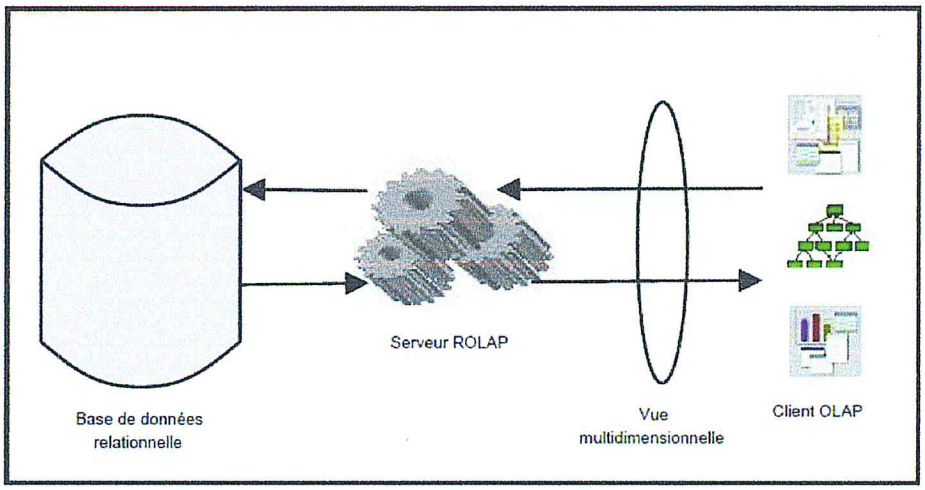


Figure.09 : Architecture ROLAP (MTS 05).

La technologie ROLAP a deux avantages principaux : (1) elle permet la définition de données complexes et multidimensionnelles en utilisant un modèle relativement simple, et (2) elle réduit le nombre de jointures à réaliser dans l'exécution d'une requête. Le désavantage est que le langage de requêtes tel qu'il existe, n'est pas assez puisant ou n'est pas assez flexible pour supporter de vraies capacités d'OLAP. **[VS 99] [BSH 98].**

Nous décrivons quelques produits commerciaux existants **[How 03] [CHJM 02]:**

- **IBM DB2 UDB :** C'est un SGBD tournant sur une variété de plate-formes.Il supporte le concept de fédération. Il dispose d'une large gamme d'outils, par exemple :

  DB2 Performance Expert : Cet outil pour multiplateformes permet la création des rapports, des analyses et recommande des changements par rapport à la performance.

  DB2 DataJoiner : Outil pour l'optimisation des requêtes SQL.

- **Oracle 9i :** C'est un SGBD tournant aussi sur une variété de plate-formes. Oracle supporte des partitions de hash, range et list.

  Oracle 9i : Supporte la consolidation (focalisée sur une base de données centralisée).

Real Application Clusters : Il permet de désigner certains processeurs comme processeurs OLAP et d'autres comme processeurs de requêtes.
L'optimiser : Basé sur les coûts ou sur les règles.

- **SQL Server 2000 :** Supporte le concept de fédération (liaison entre bases de données distribuées et hétérogènes via le logiciel).
  L'optimiser de SQL Server : Basé sur les coûts avec création automatique de statistiques et leur rafraîchissement.
  Le Query Processor : Il supporte des requêtes multidimensionnelles, ainsi que les index composites et semi-jointures.
  Le SQL Query Analyzer : Il peut faire des suggestions par rapport à l'implantation des index additionnels et des statistiques complémentaires.
  Microsoft DTS (Data Transformation Services) : L'outil ETL intégré dans Microsoft SQL Server.

5.3 MOLAP (Multidimensional OLAP)

Les systèmes multidimensionnels OLAP utilisent une base de données multidimensionnelle pour stocker les données de l'entrepôt et les applications analytiques sont construites directement sur elle. Dans cette architecture, le système de base de données multidimensionnel sert tant au niveau de stockage qu'au niveau de gestion des données. Les données des sources sont conformes au modèle multidimensionnel, et dans toutes les dimensions, les différentes agrégations sont pré calculées pour des raisons de performance.
La figure.09 montre une architecture pour les systèmes MOLAP.

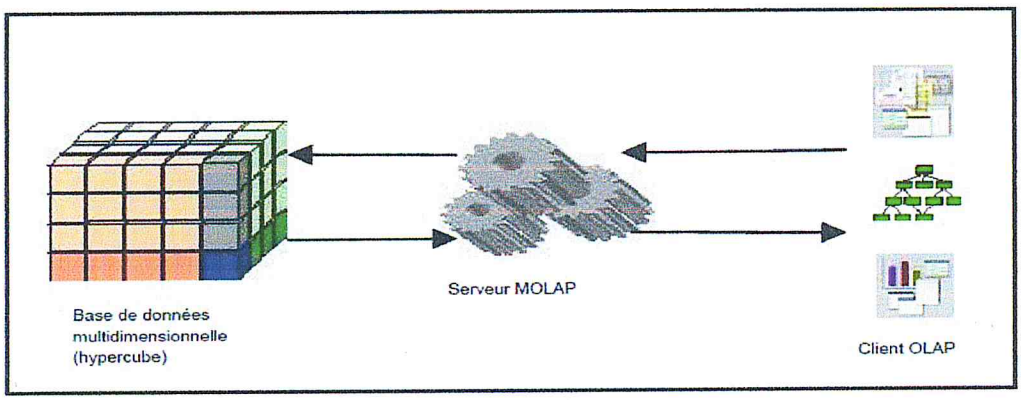


Figure.10 : Architecture MOLAP. (MTS 05).

Les avantages des systèmes MOLAP sont basés sur les désavantages des systèmes ROLAP et elles représentent la raison de leur création. D'un côté, les requêtes MOLAP sont très puissantes et flexibles en termes du processus OLAP, tandis que, d'un autre côté, le modèle physique correspond plus étroitement au modèle multidimensionnel. Néanmoins, il existe des désavantages au modèle physique MOLAP. **[BEN 02] DSBH 99].**

Les produits commerciaux existants sont :

- **Hyperion Essbase OLAP Server** : C'est une base de données multi-utilisateurs,
  multi-thread et multidimensionnelle. Les composants d'Essbase OLAP sont :
  Hyperion Essbase Application Manager : Il fournit des outils graphiques. Il inclut des modules pour la construction et le chargement des structures OLAP, pour le chargement des données, pour la définition des processus de calcul, pour la gestion des partitions de la base de données,...
  Hyperion Essbase Query Designer : Ce composant remplace le Query Wizard d'Essbase 6.0. Il a été conçu pour faciliter la navigation des utilisateurs finaux.
  Hyperion Essbase Partition Option : Il permet aux développeurs des applications la création logique et physique de sous-ensembles des bases de données. Les deux types de partitions qui sont supportés sont : transparents et liée. Le premier cherche à montrer à l'utilisateur final une base de données centralisée. Avec le deuxième type, nous pouvons définir deux partitions liées si elles partagent au moins une partie d'une dimension. Ainsi, nous pouvons naviguer entre applications et non entre partitions. Par exemple, nous pouvons naviguer entre l'application de ventes et celle d'achats, car elles relient le même ensemble de produits.
- **SAS OLAP Server :** C'est une base multidimensionnelle qui fournit un accès rapide aux données agrégées [SAS04]. Leurs composants incluent:

Un modèle de données général et un schéma multidimensionnel global :
Pour aboutir à la transparence du premier point, tant le modèle de données général que le langage de requête uniforme doivent être fournis. Etant donné qu'il n'existe pas un modèle standard, cette condition est difficile à réaliser.

Une allocation optimale dans le système de stockage : Le système HOLAP doit bénéficier des stratégies d'allocation qui existent dans les systèmes distribués tels que : le profil de requêtes, le temps d'accès, l'équilibrage de chargement,...

Une réallocation automatique : Toutes les caractéristiques traitées ci-dessus changent dans le temps. Ces changements peuvent provoquer la réorganisation de la distribution des données dans le système de stockage multidimensionnel et relationnel, pour assurer des performances optimales.

Actuellement, la plupart des systèmes commerciaux utilisent une approche hybride.

Cette approche permet de manipuler des informations de l'entrepôt de données avec un moteur ROLAP, tandis que pour la gestion des data marts, ils utilisent l'approche multidimensionnelle. Dans la figure.10 nous montrons une architecture en utilisant les types de serveurs ROLAP et MOLAP pour le stockage de données.

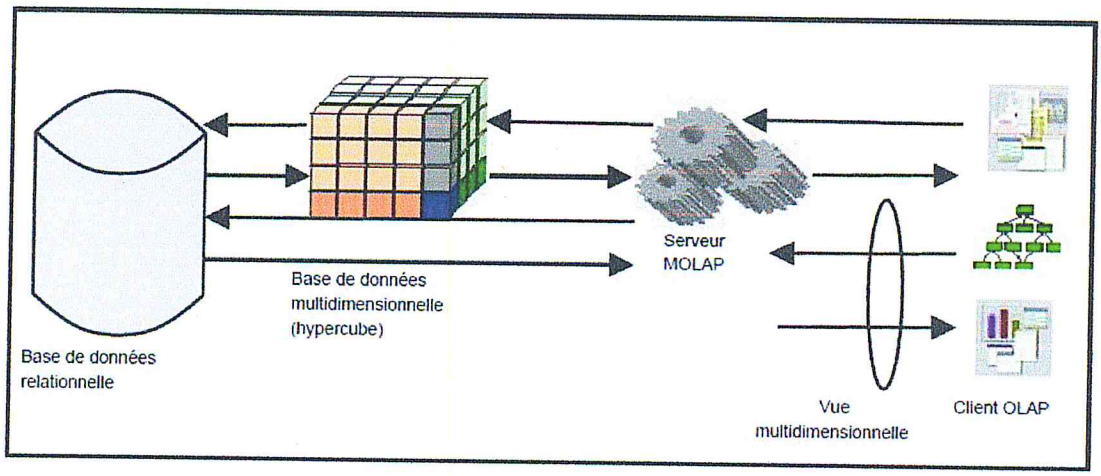


Figure.11 : Architecture HOLAP (MTS 05).

Les produits commerciaux existants sont :

- **DB2 OLAP Server :** Il permet de calculer, de consolider et d'accéder l'information à partir des bases de données multidimensionnelles, relationnelles ou les deux.Certains de ses composants sont :
  DB2 OLAP Integration Server : Il utilise des outils graphiques et des services pour l'intégration des données qui réduisent le coût pour créer, déployer et gérer les applications de DB2 OLAP Server.
  DB2 OLAP Server Administration Services : Il fournit des outils pour améliorer et faciliter des tâches d'administration.

- **Oracle Express Server :** Ce serveur exploite un modèle de données multidimensionnel.
  Il gère un ensemble d'indicateurs à n dimensions, dont les valeurs sont stockées ou calculées dynamiquement.

  Le stockage des données peut se faire dans une base de données multidimensionnelle ou relationnelle.

  La base Oracle Express Server : Stocke les agrégats multidimensionnels, tandis que les données de détail sont stockées dans la base relationnelle.
  En utilisant un **4GL (Langage de 4ème génération),** Express Server propose des fonctions avancées pour la présentation et l'analyse des résultats, tels que : traitement de séries chronologiques, consolidation, statistiques, prévisions,...

6 Conclusion :

Dans ce chapitre nous avons évoqué les principaux concepts liées à la conception et à l'exploitation des entrepôts de données, ou nous avons vu les caractéristiques des données de l'entrepôt et l'architecture associée.

Nous avons vu également les différents modèles OLAP (ROLAP, MOLAP et HOLAP) et les opérations de manipulation de ces dernières qui nous permettent l'interrogation efficace des entrepôts de données, ainsi que les vues matérialisées et son fonctionnement.

Les systèmes de personnalisation d'OLAP

1 Introduction :

L'entrepôt de données (Data Warehouse), qui a pris forme au début des années 1990 et il est devenu depuis, sous de multiples variantes, la clé de voûte de ce que nous appelons, « l'informatique décisionnelle ».

L'informatique décisionnelle a pour objectif d'élaborer des systèmes d'analyse de données dédiés au soutien et à l'amélioration des processus décisionnels des entreprises.

Les entrepôts de données constituent l'outil essentiel de collecte et de mise à disposition des données en vue de leur analyse. Le but est de collecter des données décrites de manière multidimensionnelle afin de les mettre à la disposition des décideurs à des fins d'analyse. Cette analyse fait appel à des traitements OLAP **[COD, al 93]**, qui se distinguent des processus OLTP principalement par leur complexité et par le nombre de données.

En effet, les bases multidimensionnelles sont l'un des nouveaux développements remarquables de la conception des bases de données qui étend de façon considérable les possibilités d'analyse de grands ensembles de données multidimensionnels.

Les données d'une base de données multidimensionnelle proviennent d'un entrepôt de données et sont organisées sous forme de cube de données.

Un cube est une représentation multidimensionnelle des données dans cet entrepôt. Il est décrit par les tables de dimensions (les axes d'analyse) et une table de faits (les mesures) et interrogé par des requêtes de type OLAP.

Les approches de recommandation sont généralement classées en fonction de leurs algorithmes de calcul en des approches basées sur le contenu, des approches de filtrage collaboratif et des approches hybrides **[ADO et TUZ 05]**.

1.2.1 La recommandation basée sur le contenu :

Consiste à proposer à l'usager des objets qui sont similaires à ceux qu'il a appréciés dans le passé **[MAE 94 ; PAZ et BIL 07 ; ZHA et al 02]**. Une mesure de similarité entre les objets est souvent utilisée afin d'identifier ceux qui sont susceptibles d'être utiles pour l'usager.

1.2.2 La recommandation basée sur le filtrage collaboratif :

Exploite les appréciations d'une communauté d'utilisateurs sur les objets afin de découvrir des corrélations entre les utilisateurs. L'utilisateur courant se verra recommander des objets que des utilisateurs similaires ont appréciés **[KON et al 97 ; RES et al 94]**. Ainsi une mesure de similarité entre utilisateurs est généralement établie **[SAT et al 06]**.

1.2.3 La recommandation hybride :

Des approches qualifiées d'hybrides combinent entre le calcul basé sur le contenu et le filtrage collaboratif **[BAL et SHO 97].**

La recommandation fut parmi les premières approches suivies pour fournir un accès personnalisé à l'utilisateur. Un système de recommandation a pour but de suggérer à l'utilisateur un contenu informationnel susceptible de répondre à ses besoins.

1. Un profil utilisateur est une collection d'information sur l'utilisateur et cette collection peut être vue comme un ensemble de caractéristiques avec des valeurs associées contenant par exemple les préférences d'utilisateurs. [**MOU 07**].

2. Un profil utilisateur regroupe l'ensemble des connaissances nécessaires à une évaluation efficace des requêtes et à une production d'une information pertinente adaptée à chaque utilisateur [**BOU et al 04**].

1.4.1 Le contenu du profil:

La représentation du profil d'un utilisateur varie selon le domaine d'application. Par exemple, en recherche d'information, les profils sont généralement représentés sous forme de mots clés pondérés [**FER et SIL 01**; **SOL et CRA 98**] ou d'un ensemble de fonctions d'utilité sur un domaine d'intérêt [**CHE et al 03**], tandis qu'en bases de données, les profils peuvent contenir des conditions de sélection ou de jointure des requêtes SQL [**KOU et IOA 04, 05**].

Les données du profil peuvent être entrées manuellement par l'utilisateur, ou inférées automatiquement à partir de ses interactions précédentes avec le système.

Comme indiqué précédemment, un profil utilisateur peut être représenté simplement par un ensemble de caractéristiques permettant au système d'identifier et modéliser l'utilisateur tout en respectant sa vie privée **[FAV 07],** un profil utilisateur peut être composé de :

- des informations personnelles, que l'on appelle aussi démographiques, telles que le nom, le prénom, l'âge, etc.
- des préférences sur le contenu de la réponse à la requête.
- des préférences sur la présentation de la réponse, et / ou des conditions d'exploitation.

Ces informations peuvent être complétées pour adapter encore mieux le processus de la personnalisation de l'information **[KOS 08].**

1.4.2 Construction du profil :

La construction du profil traduit un processus qui permet d'instancier sa représentation. Elle s'effectue en deux étapes :

(1) l'acquisition et la collecte des données utilisateur ;

(2) puis la construction proprement dite du profil **[TAM et al 08].**

La première phase consiste à collecter les informations pertinentes pour instancier le profil de l'utilisateur. Ce processus d'acquisition peut être explicite et/ou implicite :

a) L'approche explicite :

L'approche explicite est suffisante pour couvrir toutes les préférences exprimée par l'utilisateur. Cette technique constitue une approche simple pour obtenir des informations sur l'utilisateur. On interroge directement l'utilisateur ou on lui demande par exemple de remplir des formulaires pour collecter ses préférences sur les dimensions, les membres ainsi que des informations décrivant son environnement à savoir la taille de son écran, la vitesse de son processeur et la taille de sa mémoire, etc.

En effet, l'utilisateur émet directement son jugement d'intérêt en donnant une valeur de pertinence sur une échelle graduée allant du moins intéressant au plus intéressant.

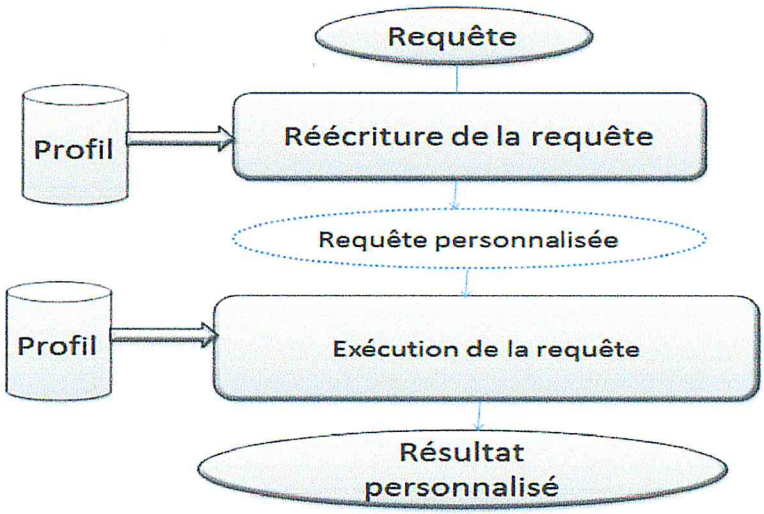


Figure.13 : L'intégration du profil lors de la phase d'exécution de la requête et la visualisation des résultats (BOU 10).

3) La phase d'affinement de la requête : Les requêtes utilisateur sont assurément une source évidente importante pour l'identification des besoins en information de l'utilisateur. Néanmoins, les utilisateurs soumettent souvent des requêtes très courtes et ambiguës. L'objectif de la personnalisation à ce stade du cycle de vie de la requête est de clarifier le besoin en information de l'utilisateur en se basant sur son profil. Ainsi, la reformulation de requête dans ce cadre intègre les composantes informationnelles issues du profil de l'utilisateur pour identifier, enrichir et cibler son intention. Dans le cadre d'un accès personnalisé aux bases de données, l'approche de reformulation de requête consiste à proposer des algorithmes d'enrichissement de la requête utilisateur par des prédicats de préférence candidats choisis du profil.

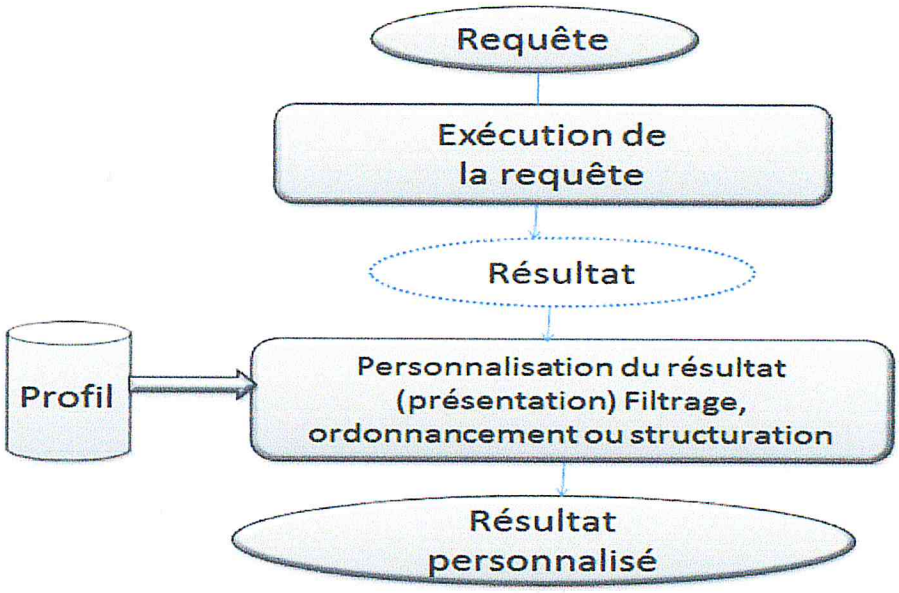


Figure.14 : L'intégration du profil lors de la phase de présentation des résultats (BOU 10).

1.5 La notion de préférences :

Les préférences utilisateur correspondent à un ensemble de critères permettant pour un utilisateur spécifique :

- De mesurer la pertinence d'une information, et
- D'évaluer si une information est plus pertinente qu'une autre information.

Dans le contexte des bases de données multidimensionnel, les préférences utilisateur permettent d'ordonner les tuples de la réponse selon leur importance et ainsi de déterminer quels sont les tuples les plus intéressants. Les préférences dans ce domaine sont exprimées sur le contenu (attributs, tuples ou requêtes) **[BOU 10]**.

1.5.1 Formulation de préférences :

Les modèles des préférences OLAP sont définis à divers niveaux selon deux approches de modélisation différentes.

Les travaux de modélisation des préférences OLAP se sont basés sur des approches initialement définies pour les bases de données. Les préférences sont formulées dans ces approches selon une approche quantitative **[AGR et WIM 00; KOU et IOA, 04, 05a]** ou qualitative **[KIE 02 ; CHO 03].**

1.5.1.1 Approches qualitatives Vs Approches quantitatives :

Les approches qualitatives permettent une formulation relative des préférences à travers une comparaison entre deux éléments (par exemple « je préfère le domaine des entrepôts de données aux réseaux »). Cette formulation est intuitive pour les usagers. Par contre, **les approches quantitatives** permettent de formuler des préférences d'une manière absolue sur les éléments désirés (par exemple « j'aime beaucoup le domaine des entrepôts de données » et « je préfère le domaine des réseaux avec un degré inférieur ») **[JER 12]**.

En termes d'expressivité, les approches qualitatives sont plus générales puisqu'on ne peut pas traduire toutes les relations d'ordre par des fonctions de score. Cependant, ces approches ne permettent pas de traduire la différence de l'intensité des préférences. Par exemple, elles ne permettent pas de distinguer les préférences « j'aime beaucoup les bases de données » et « j'aime un peu les bases de données » **[JER 12]**.

1.5.2 Expression des préférences:

Les préférences qui expriment des besoins spécifiques de l'usager à long terme sont stockées. D'autres préférences qui représentent des besoins à court terme sont exprimées explicitement au moment de la requête.

Les préférences quantitatives sont formulées dans **[RAV et TES 08]** par des règles ECA (évènement, condition, action) qui associent un poids à un attribut de dimension lorsqu'une opération OLAP est invoquée. Cette action est contrainte par la satisfaction d'une condition.

Exemple. Considérons une préférence de l'usager pour le niveau de granularité Année de la dimension temporelle. Cette préférence est décrite par la règle suivante.

CREATE RULE R1 ON Dates
WHEN displayed
THEN priority(Dates. Année, 1);

Golfarelli et Rizzi (2009) ont défini une algèbre de formulation de préférences qualitatives simples (POS, NEG, CONTAIN, …). Cette algèbre est enrichie par deux opérateurs de composition de préférences : Pareto (⊗) pour indiquer le même degré d'importance entre deux préférences et Priorisation (▷) pour définir un ordre de priorité entre préférences. Ces différents opérateurs sont ensuite intégrés dans une requête MDX à l'aide de clause
PREFERRING.

Exemple. La préférence de l'exemple précédent est formulée à l'aide de l'opérateur suivant : CONTAIN (Dates,Année). Lors de la définition d'une requête MDX, cette préférence sera intégrée à l'aide de la clause suivante : PREFERRING Dates CONTAIN Année.

Les préférences de **Xin et Han (2008)** sont exprimées par des fonctions de score. **Par exemple**, afin de rechercher les meilleurs doctorants qui publient dans des conférences avec un taux d'acceptation proche de 20% et qui sont organisées depuis 1980, la clause suivante est définie au niveau de la requête SQL :
Order By (Tx_accep − 20)_+ _(Date_deb − 1980)_ ; _ étant un paramètre de pondération des deux préférences.

D'autres méthodes d'expression explicite de préférences ont été étudiées en bases de données. Nous pouvons citer les modèles de description logique de **Bunningen et al. (2006) et de Chomicki (2003).**

Les méthodes explicites demandent un effort de formulation de la part de l'usager qui ne sait parfois pas exprimer ses préférences par un langage descriptif. Afin de remédier à ces problèmes, certains travaux proposent d'acquérir implicitement les préférences à partir de l'historique des interactions de l'usager avec le système **[Holland et al 2003].** Ces préférences sont stockées dans un profil.

1.6 Contextualisation :

Une préférence peut être associée à un contexte. Dans ce cas, elle est dite contextuelle (ou conditionnelle). Le contexte d'une préférence définit sa portée, c'est à-dire l'environnement dans lequel elle doit être prise en compte.

Une préférence contextuelle est un couple (P, C), où P est une préférence et C est un contexte. La partie contexte spécifie les conditions sous lesquelles la préférence P sera activée, où P peut être formulé selon une approche quantitative ou qualitative **[JER 12].**

2 Personnalisation des systèmes OLAP :

Selon **[GAR et al 09],** la personnalisation est un processus d'adaptation du système OLAP à certaines informations relatives à l'usager telles que ses objectifs, ses caractéristiques, son comportement et son contexte. Ainsi, et selon **[JER 12],** les travaux sur la personnalisation dans la communauté des bases de données multidimensionnelles ont porté sur les différents niveaux de l'architecture des systèmes OLAP. Nous présentons dans la suite les travaux de personnalisation par niveau :

- la personnalisation du schéma multidimensionnel.
- La personnalisation de l'interrogation des données, Deux catégories de travaux peuvent être distinguées:
  - des travaux permettant la personnalisation des requêtes de l'usager.
  - des travaux de recommandation.
- la personnalisation de la visualisation des données.

2.1 Personnalisation du schéma OLAP

Après la conception du schéma de la base de données multidimensionnel, les besoins de l'usager peuvent évoluer dans le temps **[FAV et al 07].** Les travaux sur la personnalisation du schéma multidimensionnel répondent à la problématique suivante : *Comment adapter le schéma d'une base de données multidimensionnelle aux besoins évolutifs de chaque usager ?*

2.1.1 Travaux de [Favre 2007].

L'approche de personnalisation proposée dans **[FAV 07],** est une solution basée sur une évolution du schéma de l'entrepôt guidée par les utilisateurs. Il s'agit en effet de recueillir les connaissances de l'utilisateur et de les intégrer dans l'entrepôt de données afin de créer de nouveaux axes d'analyse.

Cette approche permet aux utilisateurs d'intégrer leurs propres connaissances sur la façon d'agréger les données sous la forme de règles de type « si-alors ». Une évolution du modèle de l'entrepôt permet le partage des chemins d'agrégation nouvellement créés entre les utilisateurs du système. Supposons qu'un utilisateur veuille analyser les ventes du produit « food », non pas par « year » ni par « day » et non plus par « month », mais par semaine, information qui n'est pas présente dans l'entrepôt. L'utilisateur peut alors exprimer ses connaissances sur les types d'agence, afin de créer un niveau « semaine », correspondant à une nouvelle hiérarchie de la dimension « Time ». L'utilisateur pourra ainsi réaliser des analyses des ventes du produit « food » par semaine.

2.1.2 Travaux de [Garrigós et al 2009].

Contrairement aux approches précédentes où la personnalisation est effectuée après la mise en œuvre du système OLAP, le mécanisme de personnalisation selon **Garrigós et al. (2009)** débute depuis l'étape de conception de la base de données multidimensionnelle.

Durant cette étape, un profil de l'usager est défini ainsi qu'un ensemble de règles ECA précisant les actions de personnalisation à effectuer en ligne. La deuxième étape de ce mécanisme est effectuée au moment de l'interrogation de la base de données multidimensionnelle. Elle permet de générer une vue du schéma en fonction du profil de l'usager courant afin de faciliter la tâche de formulation de requête. Il s'agit d'une approche d'assistance à la définition de requête.

2.2 Personnalisation de l'interrogation des données :

Les travaux de personnalisation de l'interrogation des données se situent au niveau « restitution et analyse » du système OLAP. Ils répondent à la problématique suivante :

Parmi les éléments du schéma multidimensionnel et le volume important des données stockées, comment déterminer une requête qui répond au mieux aux besoins de l'usager et comment renvoyer ensuite un résultat pertinent ?

Ainsi, ces travaux permettent de personnaliser l'interrogation du schéma et/ou des instances de la base de données multidimensionnelles. Deux catégories de travaux peuvent être distinguées:

- Des travaux permettant la personnalisation des requêtes de l'usager
- Des travaux visant à assister l'usager dans la définition des requêtes, appelés communément des travaux de recommandation.

2.2.1 Personnalisation de requêtes :

La personnalisation de requête est basée sur le constat que des usagers peuvent juger des résultats différents pertinents lors du requêtage des données **[PIT et al 02]**. L'objectif de cette approche est de restituer les données les plus pertinentes pour chaque usager.

Définition. La personnalisation de requête est un mécanisme effectué avant ou après l'évaluation de la requête afin de changer la requête ou l'ordre du résultat.

Les méthodes d'expansion des requêtes supposent l'existence d'un profil de l'usager. Des éléments du profil sont utilisés pour étendre la requête de l'usager. Ceci se traduit par l'ajout de conditions de sélection **[BEL et al 05, 06],** d'attributs d'agrégation **[RAV et al 07],** la version étendue de la requête est ensuite exécutée en substitution de la requête initiale afin de générer un résultat adapté à l'usager.

2.2.1.1 Travaux de [Golfarelli et al, 2009].

Les travaux **de Golfarelli et al. (Golfarelli et Rizzi, 2009 ; Golfarelli et al, 2011)** traitent la problématique des résultats volumineux ou vides des requêtes OLAP. Inspirés des travaux de **(Kießling, 2002)** en bases de données, ils se basent sur le modèle BMO (« Best Matches Only ») d'exécution de requêtes où seulement les tuples du résultat qui ne sont pas dominés par d'autres sont renvoyés.

En ce qui concerne l'*approche*, il s'agit d'une approche de personnalisation de requête. La requête est exécutée sans personnalisation, puis les préférences sont exploitées pour déterminer les tuples du résultat qui sont meilleurs que tous les autres. Plus précisément, l'exécution de la requête suit une démarche non proactive qui évalue d'une manière incrémentale des requêtes enrichies par des combinaisons de préférences jusqu'à trouver une requête qui renvoie un résultat non vide.

En ce qui concerne *l'algorithme* de personnalisation, il prend en entrée une requête, un ensemble de préférences et l'instance de la base de donnée multidimensionnel. Il produit en sortie l'ensemble des tuples non dominés qui représente un sous-ensemble du résultat généré sans personnalisation. L'algorithme se déroule en deux étapes : la construction du graphe de dominance de préférences, puis le parcours du graphe. Le graphe de dominance est construit à partir des préférences exprimées dans la requête. Chaque préférence induit le partitionnement des tuples du résultat en plusieurs classes regroupant chacune des tuples équivalents. Chaque noeud du graphe correspond à un ensemble de classes qui ne sont pas dominées par d'autres ; et est représenté par un prédicat permettant de sélectionner les tuples de ses classes. Les arcs traduisent les liens de dominance entre les classes des noeuds. Le parcours du graphe est effectué selon l'ordre de dominance des noeuds.

Pour chaque noeud visité, une requête MDX (Multidimensional Expressions) est générée par l'expansion de la requête initiale avec le prédicat correspondant au noeud. Si l'exécution de la requête enrichie génère un résultat non vide, alors ce résultat est renvoyé à l'usager, sinon le noeud suivant est visité.

En conclusion, ces travaux ont les points positifs suivant : la réduction du volume important du résultat et l'expressivité du modèle de préférences. Les points négatifs majeurs sont l'effort manuel et cognitif de formulation des préférences et la nécessité d'exécuter plusieurs requêtes intermédiaires afin de répondre à une seule requête utilisateur. **[JER 12].**

2.2.1.2 Travaux de [Ravat et al, 2007].

Les travaux **de Ravat et al. (Ravat et al., 2007a)** se situent dans le cadre de la personnalisation des requêtes par expansion. Malgré la proposition d'une algèbre d'opérateurs OLAP, cette approche n'est pas basée sur un modèle d'analyse OLAP. La personnalisation d'une requête est indépendante des requêtes déjà lancées au cours de l'analyse courante.

L'algorithme de personnalisation prend en entrée la requête de l'usager, le schéma de la BDM et un profil de l'usager comportant un ensemble de règles. Il génère une requête étendue. Une règle détermine les attributs de dimension à afficher. Ainsi, les règles avec un score supérieur à un seuil sont utilisées pour enrichir la requête.

Du point de vue *système*, il est demandé à l'usager de définir les règles et de spécifier manuellement un seuil de personnalisation. L'implantation de cette approche est effectuée au niveau applicatif par l'ajout d'un module permettant d'analyser la requête et de l'enrichir en fonction des règles.

L'inconvénient majeur de cette approche réside dans la subjectivité de la précision du seuil de la requête qui déterminera les attributs de dimension à afficher. Par ailleurs, les règles sont limitées au niveau des structures d'une base de donnée multidimensionnel. L'approche ne permet pas par exemple de focaliser l'analyse sur l'année en cours. **[JER 12].**

2.2.1.3 Travaux de [BELLATRECHE et AL, 2005].

Les auteurs ont proposé une méthode pour la personnalisation des requêtes OLAP en exploitant le profil utilisateur pour apporter une meilleure visualisation à ce dernier (l'utilisateur). Dans cette approche, la personnalisation consiste à transformer la requête utilisateur avant l'accès au cube de données. Dans cette méthode, l'utilisateur est interrogé afin de donner ses préférences et une contrainte de visualisation. Ces préférences utilisateurs qui classent les membres, les dimensions et la contrainte de visualisation qui contrôlent les résultats sont stockées dans le profil utilisateur. Et une définition du profil utilisateur dans le contexte OLAP est proposée.

Le problème traité consiste à trouver les sous-cubes les plus intéressants d'un cube C qui peuvent être visualisés.

Les préférences utilisateur définissent une relation de pré-ordre total pour l'ensemble des membres de toutes les dimensions. Cette relation de pré-ordre total sur les membres utilisée pour définir une relation monotone de pré-ordre total sur les cubes. Par exemple, pour les membres m1 est plus intéressant que m2 veut dire que $m1 \leq m2$. Pour les cubes si le cube C1 est le sous cube de C2 ($C1 \subseteq C2$), alors C1 moins intéressant que C2 et $C1 \leq C2$.

Les auteurs introduisent les contraintes de visualisation ou d'ordre machine dans ce contexte. Ces contraintes précisent la taille de l'écran d'affichage du dispositif (machine), ou plus précisément le nombre maximal de graduations qui peuvent être affichées sur les différents axes d'analyse.

On note que plusieurs sous-cubes peuvent être résultat de cette personnalisation puisque la relation entre les membres est un pré-ordre. Ces sous-cubes sont dits modulo équivalents ≡. Intuitivement, on dit qu'ils sont également intéressants. Et plusieurs structures peuvent être proposées pour le même sous-cube.

2.2.1.4 Les requêtes de skyline :

[BOR01] est défini le concept skyline dans le contexte d'une base de données relationnelle interrogeable par le langage SQL. Les tuples d'une requête de skyline sont ceux qui ne sont pas dominés par d'autres tuples. Le concept de requête skyline a été introduit pour formuler des recherches multicritères. Lorsque ces critères sont conflictuels, l'opérateur skyline retourne les meilleurs compromis entre plusieurs critères. Ces résultats sont incomparables entre eux, on va détailler l'opérateur skyline dans le chapitre 03.

2.2.2 Recommandation de requêtes :

Les approches de recommandation ont été largement étudiées dans le domaine du web. Mais, peu de travaux **[Giacometti et al, 2008; Bentayeb et al, 2009]** ont considéré la recommandation dans le contexte OLAP.

Contrairement au domaine du web où les objets recommandés sont de différents niveaux de granularité (un article, plusieurs articles, une page web, …), les travaux en OLAP se focalisent sur la recommandation de requêtes afin de prendre en compte l'aspect navigationnel de l'interrogation des données OLAP. Ils répondent à la problématique suivante : « *comment guider l'usager dans l'exploration de la BDM afin de l'aider dans son processus de prise de décision?* ».

Définition. La recommandation de requête est l'action de proposer à l'usager une requête ou des parties de requête d'une manière adaptée à ses intérêts et/ou à son analyse en cours afin de l'assister dans l'exploration des données.

La recommandation de requête fournit deux fonctionnalités :

1) L'assistance à la définition de requête par la proposition de parties de requête

2) La proposition de requêtes complètes afin de faciliter l'exploration de l'espace multidimensionnel. **[JER 12].**

Cependant, seule la deuxième fonctionnalité a été étudiée par la communauté des bases de données multidimensionnelles **[GIA et al 08 ; BEN et al 09]** Aucun travail sur l'assistance à la formulation de requête OLAP n'a été proposé à l'image des travaux sur les requêtes SQL en base de données.

Par les opérateurs classiques d'OLAP. L'utilisateur doit fixer les paramètres des algorithmes d'une manière interactive pour obtenir les partitions appropriées. Puis, le système recommande à l'utilisateur les partitions obtenues. Si ces derniers sont validés par l'utilisateur, elles sont intégrées dans l'entrepôt de données et un nouveau niveau de hiérarchie est alors créé. Cette nouvelle hiérarchie propose à l'utilisateur des nouvelles requêtes OLAP qui n'ont pas possible avec l'ancien schéma.

2.2.2.1 Travaux de [Giacometti et al, 2008].

Une session d'analyse OLAP peut être définie comme une session interactive durant laquelle un utilisateur lance une requête pour naviguer au sein du cube. Très souvent, le choix de la partie du cube à explorer et donc la requête correspondante à cette partie est une tâche difficile.

[GIA et al 08] Proposent un système de recommandation d'analyses multidimensionnelles en se basant sur la navigation qu'effectue un utilisateur donné par rapport aux navigations réalisées par les autres utilisateurs.

La proposition de **Giacometti et al. (2008)** permet la recommandation de requêtes pour anticiper sur une séquence de requêtes d'un utilisateur grâce à l'analyse des historiques de navigations réalisées par les autres utilisateurs. Par exemple, supposons qu'un utilisateur a réalisé une analyse des ventes par année et par country, puis avec un forage vers le bas par année et par région, et enfin par city (avec un second forage vers le bas). Si un nouvel utilisateur réalise une analyse des ventes par année et par country, puis une analyse par année et par région, une analyse par année et par city lui sera recommandée, sa navigation étant similaire à une navigation réalisée précédemment.

2.2.2.2 Travaux de [Bentayeb et al, 2009].

[BEN et al 09] propose deux types de personnalisation OLAP dans les entrepôts de données : l'adaptation et la recommandation.

- **Personnalisation basée sur l'adaptation :**

Selon **[BEN et al 09],** l'adaptation permet aux utilisateurs d'exprimer ses propres besoins en termes de règles d'agrégation définies d'un niveau fils (niveau existant) à un niveau parent (nouveau niveau). La clause si, en effet, détermine des conditions sur les attributs du niveau fils pour les instances groupées et qui forment une partition. La clause Then détermine les agrégats du niveau parent, chacun correspond à un sous-ensemble de la partition.

Dans ce cas-ci, le système est adaptatif puisqu'il s'adapte en évoluant le schéma d'entrepôt de données pour tenir compte de les nouvelles informations de l'utilisateur.

- **Personnalisation basée sur la recommandation :**

Les opérateurs OLAP classiques sont conçus pour créer des agrégats intuitifs. Cependant, pour permet aux utilisateurs de trouver les agrégats non prévus et appropriés qui expriment des relations profondes dans un entrepôt de données, **[BEN et al 09]** proposent de combiner les techniques de data mining et OLAP. Ils choisissent d'employer la méthode K-means, en raison du format de partition que prendre son résultat et ainsi pour présenter les agrégats qui sont sémantiquement plus riches que ceux fournis.

2.3 Personnalisation de la visualisation des données :

Le problème du volume souvent important du résultat des requêtes OLAP a fait l'objet d'étude de la personnalisation de la visualisation dans les base de donnée multidimensionnel **[STO 03].** Certains travaux considèrent la personnalisation de la visualisation comme une approche de personnalisation de l'interrogation des données **[GOL 10].** Il s'agit d'un axe de personnalisation indépendant. D'une part, la personnalisation de la visualisation relève de l'interaction Homme-Machine et concerne un autre niveau de l'architecture des systèmes OLAP (structures de visualisation). D'autre part, il s'agit d'un mécanisme qui est généralement effectué après la personnalisation de l'interrogation de données.

La personnalisation de la visualisation des données est l'action d'adapter l'interface de visualisation des données en fonction d'un modèle de l'usager ou de critères externes tels que le type et la taille du dispositif utilisé.
Les travaux dans cet axe ont porté sur :

- La personnalisation de la structure d'affichage du résultat d'une requête, en définissant par exemple la disposition des données sur les axes d'une table multidimensionnelle **[BEL et al 05).**
- La mise en valeur de certains indicateurs décisionnels, par exemple les chemins de navigation les plus visités **[GAR et al 09].**

3 Comparaison des différentes méthodes

	Personnalisation de forme	Personnalisation de requête	Recommandation	Timing	Sortie
Travaux de [Favre 2007].					Schéma (S')
Travaux de [Garrigós et al 2009].				Durant conception, durant définition de requête	Schéma (S')
Travaux de [Golfarelli et al, 2009].		x		Après l'évaluation de requête	Tuples
Travaux de [Ravat et al, 2007].	x	x		Avant l'évaluation de requête	Requête Q'
Travaux de [BELLATRECHE et AL, 2005].	x	x	x	Avant l'évaluation de requête	$Q' \subseteq Q$
Travaux de [Giacometti et al, 2008].			x	Après l'évaluation de requête	Requête Q'
Travaux de [Bentayeb et al, 2009].			x		Requête Q'

Tableau.02 : Tableau récapitulatif des travaux de personnalisation dans les entrepôts de données.

Nous avons présenté un panorama des travaux portant sur la personnalisation des systèmes OLAP. Nous avons pu observer que certains problèmes ont été traités et que d'autres problèmes existent encore. Nous avons préféré dresser une comparaison des différents travaux étudiés.

4 Conclusion

Dans ce chapitre, nous avons préféré évoquer d'abord les différents concepts liés à la personnalisation dans les entrepôts, que ce soit au niveau de la modélisation, de la stratégie de conception, de l'exploitation, etc.

Par la suite, nous avons attaqué le noyau de notre thème qui est la personnalisation, nous avons éclaircit la différence entre personnalisation et recommandation, à quel niveau la personnalisation peut être effectuée : au niveau conception de l'entrepôt ou au niveau présentation (visualisation) des résultats d'une requête OLAP.

Et pour chaque type de personnalisation nous avons fait un rapide survol des différents travaux effectués dessous.

Il faut signaler que ce domaine (la personnalisation dans les entrepôts de données) est encore au stade de recherche. De ce fait, il résulte un manque de démarches de conception complètes qui traitent de toutes les étapes de conception.

Partie

II

Conception du système

- Proposition
- La conception

1 Introduction:

Dans ce chapitre, on va présenter d'abord l'opérateur Skyline ainsi que la problématique à laquelle il répond. Ensuite on va expliquer les algorithmes de calcul du Skyline.

Après, on va continuer par une description globale de notre approche qui est venue palier aux insuffisances de la méthode introduite initialement par **Börzsönyi et al. (2001).**

2 L'opérateur Skyline:

Le concept de requête skyline a été introduit pour formuler des recherches multicritères. Lorsque ces critères sont conflictuels, l'opérateur skyline retourne les meilleurs compromis entre plusieurs critères. Ces résultats sont incomparables entre eux.

Le calcul de la Skyline est connu comme le problème vecteur maximal.

Dans [BOR01] l'opérateur skyline [BOR01] est défini dans le contexte d'une base de données relationnelle interrogeable par le langage SQL. Les tuples d'une requête de skyline sont ceux qui ne sont pas dominés par d'autres tuples.
Un tuple domine un autre s'il est aussi bon ou meilleur dans toutes les dimensions et mieux dans au moins une dimension. Un élément peut être vu comme un point dans un espace à « m » dimensions, où « m » est le nombre d'attributs qu'il possède. [BOR01]

La figure.14 montre la Skyline d'hôtels bon marché près de la plage pour un ensemble d'échantillons d'hôtels. Un hôtel avec prix = 50$ et la distance = 0,8 miles domine un hôtel avec des prix = $100 et la distance = 1,0 miles.

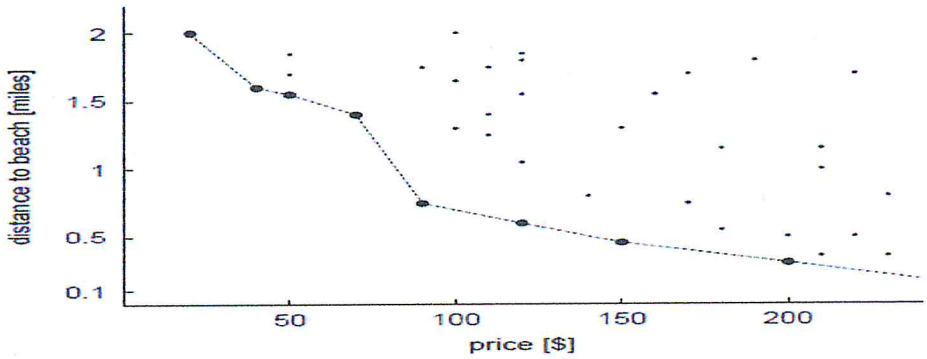


Figure.15 : le Skyline d'hôtels (Börzsönyi et al. 2001)

2.1 Syntaxe et Sémantique des requêtes de Skyline :

La syntaxe et la sémantique des requêtes skyline ont d'abord été présentées officiellement dans [BOR01]. La syntaxe de base des requêtes skyline est définie par la syntaxe suivante (extension de SQL):

SELECT ... FROM ... WHERE...
GROUP BY ... HAVING...
SKYLINE OF [DISTINCT] d1 [MIN | MAX | DIFF]...dm [MIN | MAX | DIFF]
ORDER BY ...

d1,..., dm indiquent les dimensions de skyline, ou d'une qualification. MIN, MAX, et DIFF spécifient si la valeur de cette dimension devrait être réduite, agrandie ou simplement différente.

La sémantique de la clause de skyline est simple, le skyline de la clause est exécutée après l'instruction SELECT... FROM ... WHERE ... GROUP BY ... HAVING..., mais avant la clause ORDER BY .La clause de skyline sélectionne tous les tuples intéressants, ces tuples qui ne sont pas dominés par d'autres tuples.

2.2 Exemple :

RowId	Propriétaire	...	Ville	Prix	Éloignement	Consommation	Voisins
1	Dupont		Marseille	220000	15	275	5
2	Dupond		Aix-en-Provence	100000	15	85	1
3	Martin		Marseille	220000	7	180	1
4	Sanchez		Aubagne	340000	7	85	3
5	LIF		Aix-en-Provence	100000	7	180	1

Table.03 La relation logements [SEB 09]

La relation illustrée par la table 03 est typique pour l'utilisation du Skyline. Elle répertorie différents logements. Les attributs classiques sont ici Propriétaire et Ville, et les critères de choix pour trouver le « meilleur logement » sont:

- Le Prix de vente,
- L'éloignement par rapport au lieu de travail en kilomètres,
- La Consommation énergétique en kilowattheures par an et par mètre carré,
- Le nombre de Voisins.

Pour des raisons de clarté, la table est divisée en trois parties. La première contient le RowId. C'est un attribut implicite ne faisant pas partie du schéma de la relation. Sa valeur sert d'identifiant unique à chaque tuple. Nous notons Ti le tuple ayant i pour RowId. La deuxième colonne contient toutes les informations qui n'entrent pas en compte des critères de préférence de l'utilisateur.Il n'y a pas de restriction particulière sur leur domaine. Enfin, la troisième partie regroupe les attributs préférences qui vont être pris en considération pour le calcul du Skyline.

Avec la relation précédente (**Table.03**), en supposant que l'utilisateur souhaite trouver les meilleurs logements suivant ces critèresC = {Eloignement; Prix}, **la table.03** donne la projection de cette relation selon l'ensemble d'attributs C.

Le tuple T1 est dominé par le tupleT2 (T2 $\leq_C$ T1) car ils ont le même éloignement mais que T2.Prix < T1.Prix. Pour l'utilisateur le tuple T2 répond mieux à ses souhaits que le tuple T1, il n'est donc pas intéressant de conserver le tuple T1. Pour les tuples T2 et T3, on a T2 $\geq_C$ T3 (éloignement) et T3 $\geq_C$ T2 (Prix), ils sont donc tous les deux incomparables par rapport à la relation de dominance.

Lorsqu'un tuple T domine un tuple U(T ≥C U), cela signifie que T est équivalent ou« Meilleur » que le tuple u pour tous les critères choisis. Comme nous considérons que les critères sont minimisés, les valeurs de T pour tous les critères sont inférieurs ou égales à celles de U **[SEB 09]**.

2.3 La mise en œuvre de l'opérateur de Skyline :

L'approche pour mettre en œuvre les requêtes Skyline est d'étendre un système existant (relationnel, orienté-objet) de base de données avec un nouvel opérateur logiquequi s'appelle « opérateur Skyline ».

2.3.1 Traduire une requête Skyline dans une requête SQL imbriquée :

On va commencer par montrer comment les requêtes Skyline peuvent être mises en œuvre sur une base de données relationnelleen traduisant la requête Skyline dans une requête imbriquée SQL.

```
SELECT *
FROM R t
WHERE NOT EXISTS ( SELECT *
                   FROM R u
                   WHERE u .c1 <= t .c1
                   AND . . .
                   AND u .cd<= t . cd
                   AND ( u. c1 <t . c1 OR . . . OR u . cd <t . cd ) ) ;
```

On peut très bien exprimer dans SQL que nous sommes intéressés dans des Logements qui ne sont pas dominés par d'autres logements. Cependant, cette approche présente un rendement très médiocre pour un certain nombre de raisons:

1. **Le tuple t est dominé par un tuple de la fenêtre**

t est alors directement écarté et ne sera plus pris en compte pour le reste du calcul : il est dominé et ne fera donc jamais partie du Skyline. On élimine t dés la premiére occurrence de ce cas et il n'est pas nécessaire de poursuivre la comparaison avec les autres tuples de la fenêtre.

2. **Le tuple t domine un ou plusieurs tuples de la fenêtre**

Les tuples de la fenêtre dominés par t sont écartés et ne seront plus pris en compte pour les itérations futures. t est inséré dans la fenêtre sans probléme, puisqu'il y a au moins une place libre.

3. **t est incomparable avec l'ensemble des tuples de la fenêtre**

C'est le cas le plus compliqué : t doit étre ajouté à la fenêtre mais il n'a éliminé aucuntuple sur son passage et il se peut que celle-ci soit pleine. Si ce n'est pas le cas, t est ajouté normalement à la fenêtre. Sinon, t est mis de côté dans le fichier temporaire et sera examiné à nouveau au cours de la prochaine itération de l'algorithme.

À la fin de chaque itération, on peut extraire les tuples candidats qui ont été comparés à tous les autres, y compris ceux du fichier temporaire : ils ne sont dominés par aucun autre et font donc partie du Skyline. Il n'est pas nécessaire de les garder dans la fenêtre plus longtemps. Celle-ci ne contient donc que les tuples réellement candidats, c'est à dire dont on ne peut affirmer qu'il font partie du Skyline. Pour savoir facilement quels tuples ont déjà été comparés entre eux, un systéme d'estampillage est utilisé : chaque tuple étiqueté t a été comparé avec tous les tuples d'étiquette inférieure à t. Ceci permet de garantir que l'algorithme se termine toujours et que deux tuples ne seront jamais comparés ensemble plusieurs fois.

Cet algorithme fonctionne particuliérement bien si le Skyline est petit et qu'il n'y a pas beaucoup de candidats à la fois : le fichier temporaire n'est alors pas utilisé puisqu'il y a toujours de la place dans la fenêtre. L'algorithme se termine ainsi aprés une seule itération, avec sa meilleure complexité O(n), avec n le nombre de tuples de l'entrée à examiner. Inversement, dans le pire des cas, sa complexité est quadratique $O(n^2)$. Il reste cependant toujours bien plus efficace que la requéte Sql puisqu'il limite considérablement les E/S, la fenêtre résidant en mémoire centrale.

2.3.2.2 L'algorithme Divide-and-Conquer (DC) :

Pour résoudre le probléme du « MaximalVector Computation » un algorithme éfficace a été proposé par **Preparata et Shamos (1985); Kung et al. (1975).**, ce probléme est équivalent au calcul du Skyline mais défini dans un contexte extrêmement différent de celui des bases de données.

Börzsönyi et al. (2001) en ont proposé une adaptation qui prend en compte les contraintes associées à ce nouveau contexte.

Pour comprendre les idées sous-tendant cet algorithme, il faut se représenter le probléme de l'opérateur Skyline de maniére géométrique. Les tuples à examiner sont alors considérés comme des points, dont les coordonnées sont les valeurs numériques des attributs-critéres du Skyline. Le principe général de cette méthode est de travailler de maniére récursive en divisant l'espace en des sous-espaces plus petits jusqu'à rendre le calcul du Skyline évident. Puis, il faut reconstruire le résultat final en réunissant tous les Skylines partiels. Voici les trois étapes principales de l'algorithme:

1. **Calculer la médiane**

La médiane Mi est calculée pour l'entrée par rapport à la dimension di. Elle est utilisée pour diviser l'entrée en deux partitions, P1 dans laquelle se trouvent tous les tuples T pour lesquels T.di est meilleur que mi et P2 contient les tuples restants.

2. Calculer les Skylines

C'est la partie récursive de l'algorithme : les Skylines SKY1 et SKY2 des partitions P1 et P2 sont calculés. L'espace est partitionné de maniére récursive jusqu'à ce que chaque partition ne contienne plus que très peu de tuples. Le calcul du Skyline devient alors simple.

3. Fusionner les Skylines

Le Skyline résultant de la fusion de SKY1 et SKY2 est calculé. Remarquons qu'aucun des tuples de SKY1 ne peut être dominé par un tuple de SKY2 pour la dimension di, puisque la médiane Mi a été utilisée pour diviser l'espace. De plus, une autre propriété géométrique du Skyline peut être exploitée en divisant SKY1 et SKY2 à l'aide d'une médiane Mj pour une autre dimension dj ≠ di. Il en résulte alors quatre partitions SKY1.1, SKY1.2, SKY2.1, SKY2.2. L'avantage est que nous n'avons pas à essayer de fusionner SKY1.2 et SKY2.1 car leurs tuples sont forcément incomparables. Il faut maintenant fusionner les paires restantes en procédant de maniére récursive.

La complexité de cet algorithme **[PER SHA, KUN 85, 75]** est del'ordre de O(n (log n)$d-^{2}$) +O(n log n) (avec d le nombre de dimensions sur lesquelles porte le Skyline et n le nombre de tuples en entrée). Ce résultat est meilleur que la complexité en O(n^{2}) de la requête Sql et de BNL. Cet algorithme se comporte donc en général bien mieux que BNL lorsque le résultat est de grande taille et ne tienne pas en mémoire.

2.3.2.3 B-trees :

Pour calculer le Skyline, il est également possible d'utiliser un index ordonné, B-tree Une façon d'utiliser un index ordonnée pour un skyline à deux dimensions est de parcourir tous les indices, obtenir les tuples dans l'ordre et filtrer les tuples de la Skyline.

Cet algorithme peut également être appliquée si la requête a prédicats dans sa clause WHERE, dans ce cas, nous avons besoinà chercher dans les indices jusqu'à ce qu'on trouve le premièr correspondant qui qualifie ces prédicats. En présence des prédicats, l'algorithme devient moins intéressant, car il prend plus de temps pour trouver un correspondant. En général,cetalgorithme est intéressant si le résultat de la Skyline est limité et un premier match peut être trouvé trèsrapidement.

2.3.2.4 R-tree:

Il s'agit d'une structure de données conçue pour indexer spatialement des objets. Cette structure est particulièrement performante lorsque le nombre de dimensions n'est pas trop important, c'est typiquement le cas pour les requêtes Skyline. En effet si le nombre de critéres est trop important, le nombre de résultats de l'opérateur devient alors très grand, l'interprétation de ces résultats devient plus délicate. Si l'utilisateur souhaite malgré cela prendre en compte un grand nombre de critéres.

3 Proposition :

D'après ces algorithmes, on va choisir Divises pour régner et on va montrer maintenant pourquoi cet algorithme est important .

4 Justifucation de la proposition :

Cela fonctionne près décomposant un problème en deux sous-problèmes ou plus du même (ou relié) type, jusqu'à ce que ceux-ci deviennent assez simples pour être résolus directement. Les solutions aux sous-problèmes sont alors combinées pour donner une solution au problème original. Cette technique est la base des algorithmes efficaces pour toutes sortes de problèmes.

Divisez pour régner est un outil puissant pour résoudre des problèmes conceptuellement difficiles, tels que le classique : tout qu'il exige est une manière de casser le problème en sous-problèmes, de résoudre les cas insignifiants et de combiner des sous-problèmes au problème original. La division du problème en sous-problèmes de sorte que les sous-problèmes puissent être combinés encore est souvent la difficulté principale en concevant un nouvel algorithme. En effet, parce que beaucoup de tels problèmes le paradigme offre seulement solution simple.

D'ailleurs, **Divisez pour régner** fournit souvent une manière normale de concevoir efficacement un algorithmes.

Divisez pour régner est naturellement adaptés pour l'exécution dans des processeurs multiples, particulièrement systèmes de partager-mémoire où la communication des données entre les processeurs n'a pas besoin d'être projetée à l'avance, parce que des sous-problèmes distincts peuvent être exécutés sur différents processeurs.

Divisez pour régner tende naturellement à faire l'utilisation efficace de la mémoire . La raison est qu'une fois qu'un sous-problème est assez petit, c'et tous ses sous-problèmes peuvent, en principe, être résolus dans la cachette, sans accéder à la mémoire centrale plus lente. Un algorithme a conçu pour exploiter la cachette de cette façon s'appelle cachette inconsciente.

Cet algorithme est théoriquement le meilleur algorithme connu pour le pire des cas. Dansle pire des cas, sa complexitéasymptotique estde l'ordre de« $O(n (\log n) d^{-2}) + O(n \log n)$ », n est le nombre des tuples d'entrée et d est le nombre des dimensions du Skyline**[HIG 93].**

5 Conclusion :

Dans ce chapitre nous avons vu l'opérateur skyline et la syntaxe associé proposé par **(Börzsönyi et al. 2001),** et comment on peux traduire la requête skyline en requête SQL imbriqué.

Aussi, Nous avons pris une vue sur les algorithmes de calcule de skyline et pourquoi nous avons préféré **Devisé pour régner.** Nous faisons un résumé sur les avantages de ce dernier :

- Efficacité
- Parallélisme
- Accès mémoire
- La complexité

La conception

1 Introduction :

Dans ce chapitre on va développer notre contribution sur la personnalisation des réponses aux requêtes OLAP. D'abord on va présenter l'entrepôt de donnée. Ensuite on va montrer l'architecture globale de notre système.

Nous présentons par la suite notre approche de personnalisation en donnant l'algorithme proposé suivi par un exemple illustratif, puis on va présenter la solution abstraite (conception), on utilise le langage de modélisation UML pour la présentation (Conception orienté objet) dans deux diagramme :

- Diagramme des cas d'utilisation.
- Diagramme de classe.

Pour parvenir à la mise en œuvre de notre système, nous formalisons le problème de personnalisation traité. Puis nous détaillons notre approche en donnant l'algorithme proposé.

2 Formalisation du problème

2.1 Entrées :

- *Requête MDX Q;*
- *Préférences utilisateur sur les mesures ;*

2.2 Sortie:

- *Résultat personnalisé;*

Le problème consiste à trouver l'ensemble skyline de la requête formalisée de manière à ce que la réponse à cette requête soit la meilleure par rapport aux préférences d'utilisateur et que cette réponse soit satisfaite.

3 Architecture du système :

L'architecture de système est une discipline multiforme et transversale qui traite de "comment concevoir et construire un système". Son objectif est de maîtriser l'évolution du système d'information et la structure des systèmes informatiques à travers l'anticipation des problèmes de frontières, de faisabilité et de performance tout au long du processus logiciel.

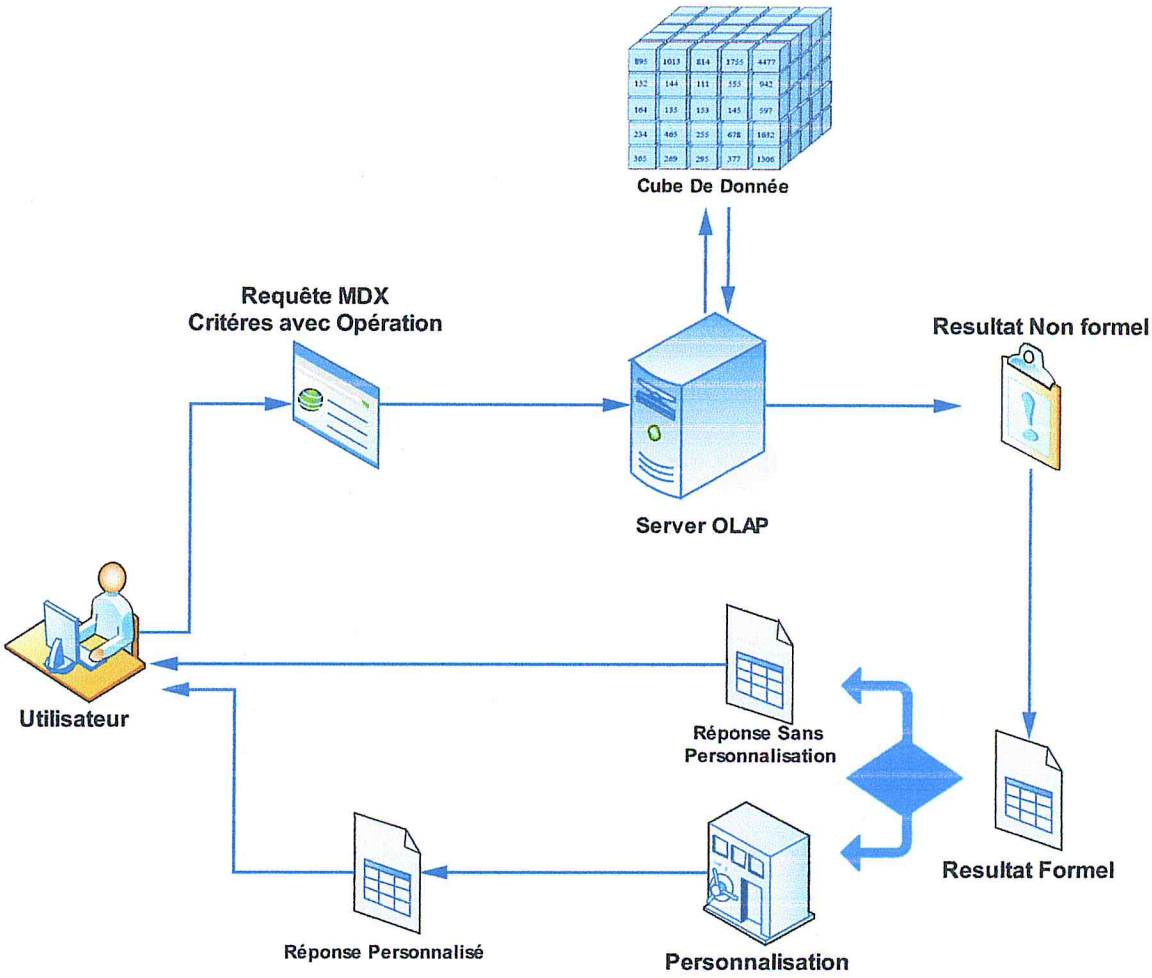


Figure.16 : Architecture globale du système de personnalisation

4 Description générale le déroulement de la personnalisation :

L'approche de la personnalisation que nous proposons consiste à trouver l'ensemble de tuples non dominés à l'utilisateur d'après sa requête MDX qui correspond aux préférences qu'il a choisi.

Pour atteindre cet objectif on divise système en trois modules :

Premier module : Ce module comprend la préparation et la modélisation des critères qui sont venus de la requête MDX (les mesures) avec ses type d'opération (MIN, MAX et Plus proche de).

Lorsque la requête est écrite, les préférences sont posées à l'utilisateur (à partir les mesures qu'il a choisi) pour introduire les différent critères qui rendent le système trouver l'ensemble skyline.

Deuxième module : Après avoir exécuté la requête MDXoriginale émise par l'utilisateur et obtenir la réponse à partir du serveur, ce module organise la réponse d'OLAP, c'est l'étape où l'on change la forme de la réponse de format non formel en format formel pour être prêt à la personnalisation.

- **Format non formel** : la réponse du serveur OLAP est un objet qui englobe plusieurs objets (Axis(), Values(), Cells(), …etc), elle est d'une structure de N dimensions.

- **Format formel** : C'est une structure transformé à partir de la réponse de serveur OLAP en deux dimensions (matrice).

Troisième module : Ce module est le noyau du système, c'est là qu'on applique l'algorithme de calcul de skyline sur la réponse du server OLAP qui est fourni par les critères d'utilisateur.

5 Algorithmes de personnalisation :

L'algorithme que nous avons choisi est introduit par Börzsönyi et al. (2001), il a défini comment calculer l'ensemble skyline en utilisant l'algorithme diviser pour régner.

Diviser pour régner est une technique algorithmique consistant à diviser un problème de grande taille en plusieurs sous-problèmes analogues. L'étape de subdivision est appliquée récursivement, Les algorithmes récursifs utilisent naturellement cette technique : ils s'appellent eux-mêmes une ou plusieurs fois sur une partition du problème initial et combinent les solutions pour retrouver une solution au problème initial.

Dans notre contexte nous appliquons l'algorithme divisé pour régner sur une réponse OLAP (table multidimensionnelles). On calcule le médiane pour chaque critère et on divise l'ensemble des tuples en deux partitions, ou la première partition est meilleur que l'autre. Ensuite on cherche l'ensemble non dominé (skyline) dans chaque partition et on commence de combiner ces ensembles.

Pour avoir réalisé cet algorithme, on créer un arbre binaire ou la tête contient les tuples initiales, et les fils contiennent les partitions, après, on cherche l'ensemble non dominé dans chaque fils puis on combine ces ensembles entre eux jusqu'à arriver à la tête.

Fonction Construire (créer l'arbre)

Entrée :

- *Indice des colonnes des critères*
- *La médiane de chaque ensemble de critères.*
- *Sommet courant*

Sortie :

- *Arbre*

fonction **Construire**(Nœud tête, Liste tuples, Médiane : réel, indice critères : entier):sommet;

```
var m:entier;
var c,s:sommet;

début
   si indice critére = critère.taille
      si médianne ≤ tuple.critère alors
         new(c);
         setValeur(c,tuple);
         c^.gauche=NIL;
         c^.droit=NIL;
         tête^.droite = c ;
        c.pére = tête ;
        retourner(c);
      sinon
         new(c);
         setValeur(c,tuple);
         c^.gauche=NIL;
         c^.droit=NIL;
         tête^.gauche = c ;
         c.pére = tête ;
         retourner(c);
     finsi
   sinon
       s= Construire(tête.Droite, Liste tuples,  Médiane : reel,
indice critèere) ;
       s= Construire(tête.gauche, Liste tuples,  Médiane : reel,
indice critèere) ;

);
   finsi
 fin
```

Fonction dominerNoeud

Entrée :

- *Indice des colonnes des critères*
- *La médiane de chaque ensemble de critères.*
- *Sommet courant*

Sortie :

- *Entienr*

```
fonction DominerNoeud(val x:sommet, Critère c):entier
 var cpt=0 ;
  début
    si estFeuille(x)alors
      si (x.tuple1.c1 ≤ x.tuple2.c2)
        cpt = cpt + 1
        retourner cpt ;

    sinon
      si
        cpt = cpt + DominerNoeud (x.droite, Critère c) ;
        cpt = cpt + DominerNoeud (x.droite, Critère c) ;
      finsi
      finsi
    finsi

   fin
```

Fonction DominerArbre

Entrée :

- *Sommet*

Sortie :

- *Sommet*

```
fonction DominerArbre(tête:sommet):sommet;
  var s:sommet;
  début
    si (estFeuille(tête)) alors
      Si DominerNoeud(tête, Critère) == nombreDeCritère;
         tête.père.tuples = tête.tuples
      sinon
        tête.tuples = DominerArbre(tête.droite)
        tête.tuples = DominerArbre(tête.gauche)
      finsi
  fin
```

6 Exemple :

Supposons que l'utilisateur veut exécuter la requête suivante

```
SELECT { {[Measures].[Internet Sales Amount], [Measures].[Internet Gross Profit]} *[Product].[Style].CHILDREN} ON COLUMNS,{[Date].[Month of Year].CHILDREN} ON ROWS FROM [Adventure Works]
```

Avec les préférences:

- Montant Moyen des Ventes (Internet) MIN
- Bénéfice Brut (Internet) MIN

Le résultat de la requête sans personnalisation est affiché dans le tableau.10

		Montant des Ventes (Internet)			Bénéfice Brut (Internet)		
		Not Applicable	Unisex	Womens	Not Applicable	Unisex	Womens
0	Janvier	$56,456.93	$2,056,647.96	$262,751.79	$35,341.94	$842,896.27	$100,769.92
1	Février	$56,995.90	$2,133,643.67	$311,747.29	$35,679.33	$875,184.07	$118,747.70
2	Mars	$60,097.80	$2,225,581.75	$324,935.62	$37,621.12	$914,246.64	$124,334.06
3	Avril	$62,673.58	$2,321,242.04	$394,926.46	$39,233.55	$952,421.19	$150,598.80
4	Mai	$71,880.47	$2,574,200.69	$468,565.11	$44,997.06	$1,067,509.91	$178,065.92
5	Juin	$65,200.93	$2,640,635.89	$475,087.17	$40,815.67	$1,092,695.78	$180,872.37
6	Juillet	$48,212.84	$1,721,719.63	$141,330.32	$30,181.16	$698,754.48	$54,525.82
7	Août	$52,056.61	$1,678,393.47	$169,156.59	$32,587.35	$680,628.74	$65,506.29
8	Septembre	$52,149.72	$1,585,959.14	$196,559.30	$32,645.64	$642,713.32	$75,802.65
9	Octobre	$54,595.17	$1,761,081.15	$193,492.97	$34,176.49	$718,254.38	$74,944.17
10	Novembre	$54,832.02	$1,776,937.37	$244,300.22	$34,324.75	$724,185.48	$93,410.61
11	Décembre	$65,607.99	$2,597,879.55	$401,142.13	$41,070.49	$1,061,817.17	$153,323.33

Tableau.4: Résultat de la requête sans personnalisation

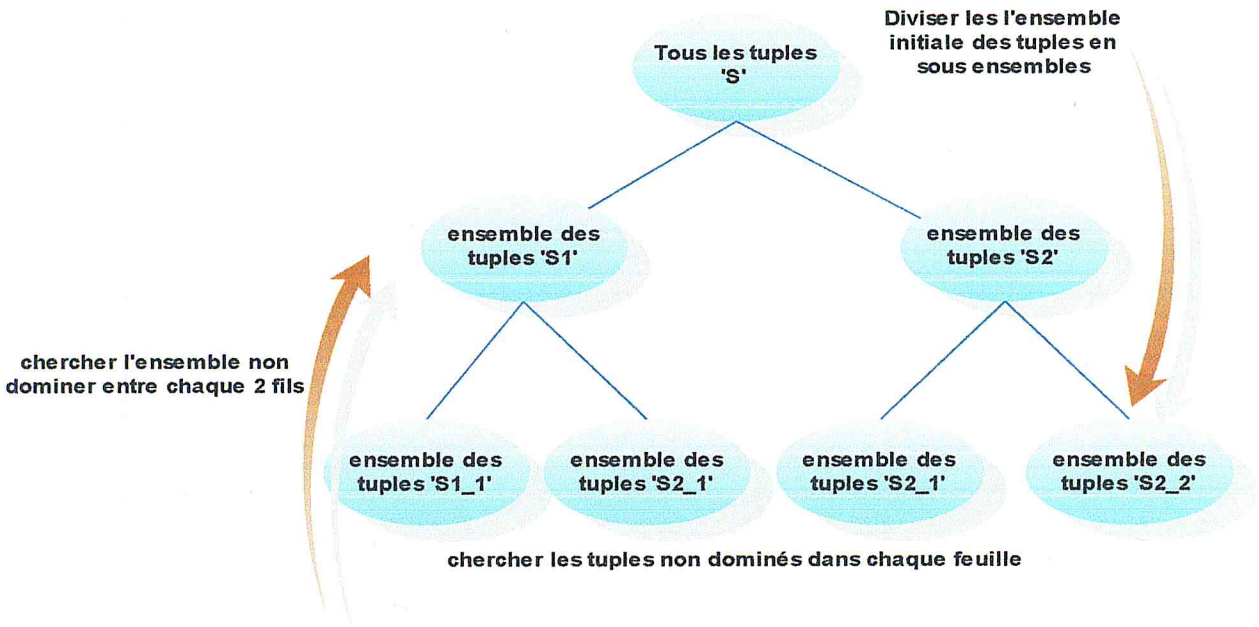


Figure.17 : Arbre de personnalisation

Le résultat avec personnalisions est affiché dans le tableau.11

		Montant des Ventes (Internet)			Bénéfice Brut (Internet)		
		Not Applicable	Unisex	Womens	Not Applicable	Unisex	Womens
6	Juillet	$48,212.84	$1,721,719.63	$141,330.32	$30,181.16	$698,754.48	$54,525.82
7	Août	$52,056.61	$1,678,393.47	$169,156.59	$32,587.35	$680,628.74	$65,506.29
8	Septembre	$52,149.72	$1,585,959.14	$196,559.30	$32,645.64	$642,713.32	$75,802.65

Tableau.5 : Résultat de la requête avec personnalisation

Nous allons détailler un autre exemple dans le chapitre **réalisation**.

7 La Modélisation :

La modélisation a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client. Parmi les méthodes les plus connues on peut notamment citer UML

UML (en anglais Unified Modeling Language ou langage de modélisation unifié) est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique **[IVA 05].**

On va présenter la solution abstraite (conception), on utilise le langage de modélisation UML version 2.4 pour la présentation (Conception orienté objet) dans deux diagramme :

- Diagramme des cas d'utilisation.
- Diagramme de classe.

7.1 L'outil utilisé :

L'outil utilisé s'appelle **Edraw Max**, c'est un logiciel de création des diagrammes 2D avec des symboles et des exemples riches, ce qui rend facile de créer des organigrammes.

7.2 Diagramme des cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). **[IVA 05]**

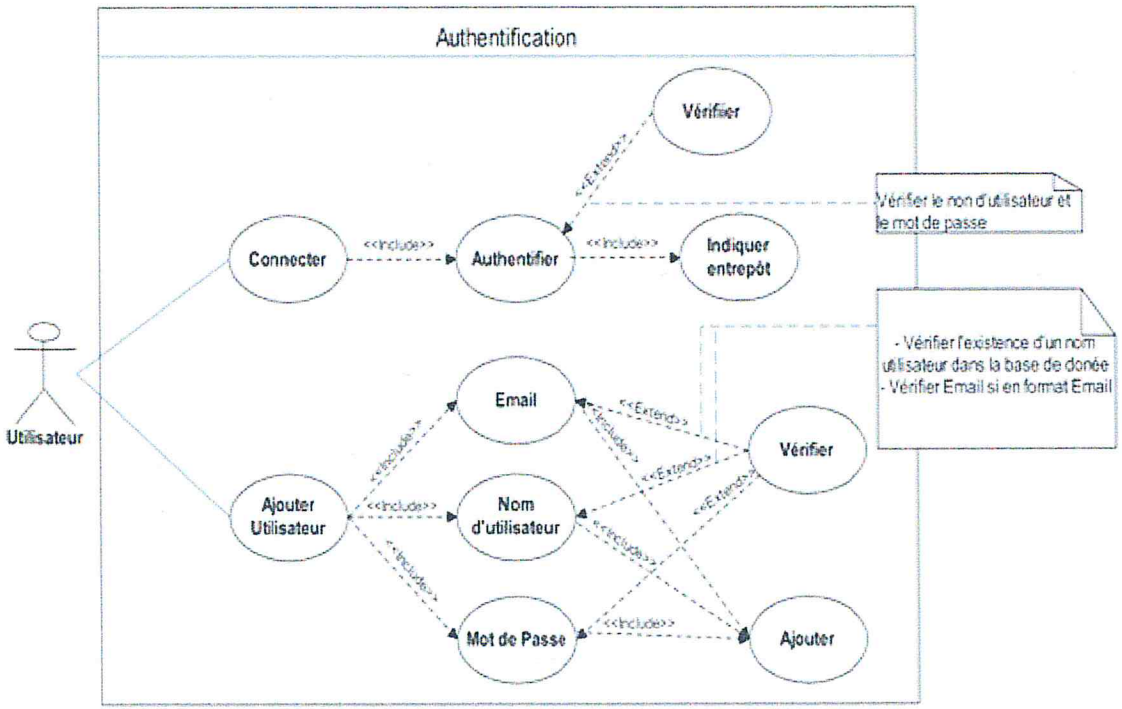
Diagramme de cas d'utilisation	Scénario
	Authentification permettant de vérifier l'identité d'une personne, largement utilisé dans les réseaux informatiques. L'utilisateur doit alors fournir un nom d'utilisateur et un mot de passe associé. **L'ajout de l'utilisateur :** Si un nouvel utilisateur veut utiliser le système pour la première fois il doit créer un compte utilisateur déterminant leur Email, nom d'utilisateur et le mot de passe. **La connexion :** L'utilisateur doit s'authentifier pour utiliser le système, il suffit de saisir un mot de passe valide avec son nom d'utilisateur. L'utilisateur doit indiquer le nom de l'entrepôt de donné que veut-il faire analyser.

Figure.18 : Diagramme de cas d'utilisation(Authentification).

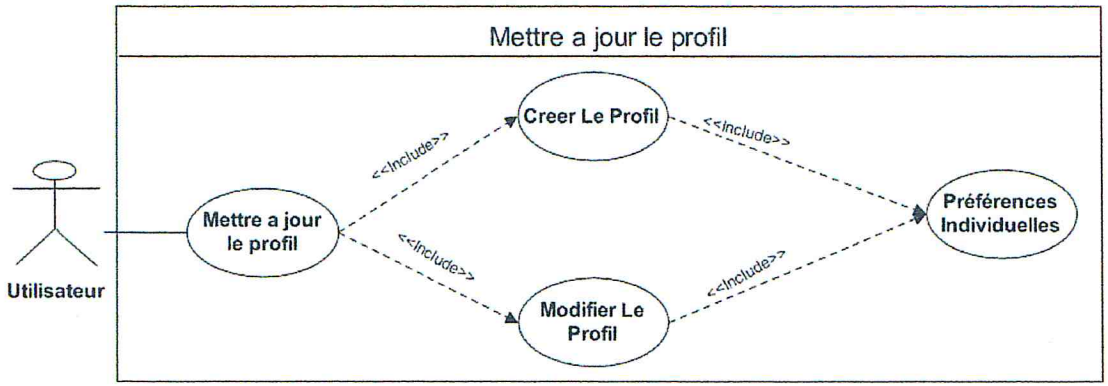


Figure.19 : Diagramme de cas d'utilisation (Mettre à jour le profil).

C'est l'action qui consiste à mettre à jour le profil utilisateur en spécifiant les informations sur le personnel.

Créer profil :

La création d'un profil consiste à définir les informations personnelles (nom, âge,...) et les préférences.

Modifier profil :

La modification des profils c'est tout changement sur les préférences.

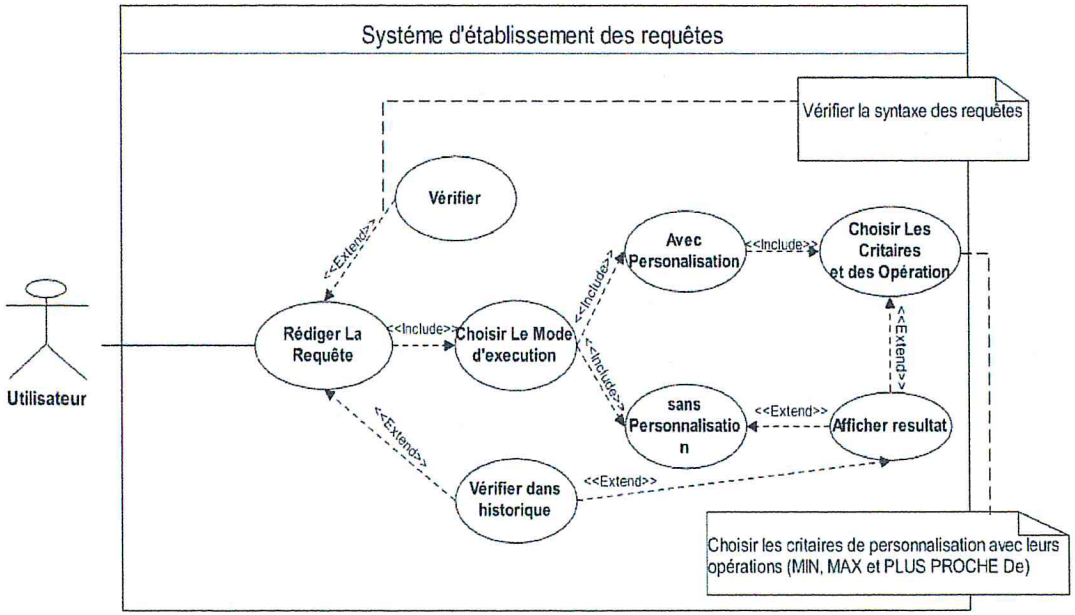


Figure.20 : Diagramme de cas d'utilisation (Système d'établissement d'une requête).

L'établissement des requêtes est la phase à laquelle l'utilisateur peut faire les analyses et exécuter des requêtes vers l'entrepôt.

Rédiger la requête :

L'utilisateur peut rédiger une nouvelle requête MDX visuellement, le système vérifie si la requête existe dans historique, il affiche le résultat immédiatement. Sinon l'utilisateur doit choisir le mode d'exécution, s'il choisit le mode sans personnalisation le résultat s'affiche directement, sinon avec le mode personnalisé il doit définir les critères de personnalisation avec leurs types d'opérations.

7.3 Diagramme de classes :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces les systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensembles par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence les relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

Le diagramme de classe permet d'exprimer comment les objets et les classes vont être définies, ainsi que les relations qui existent entre les différentes classes. **[IVA 05]**

On va présenter d'abord le diagramme de package, ensuite nous détaillons le diagramme de classes et les classes associés

7.4 Diagramme de classe par package :

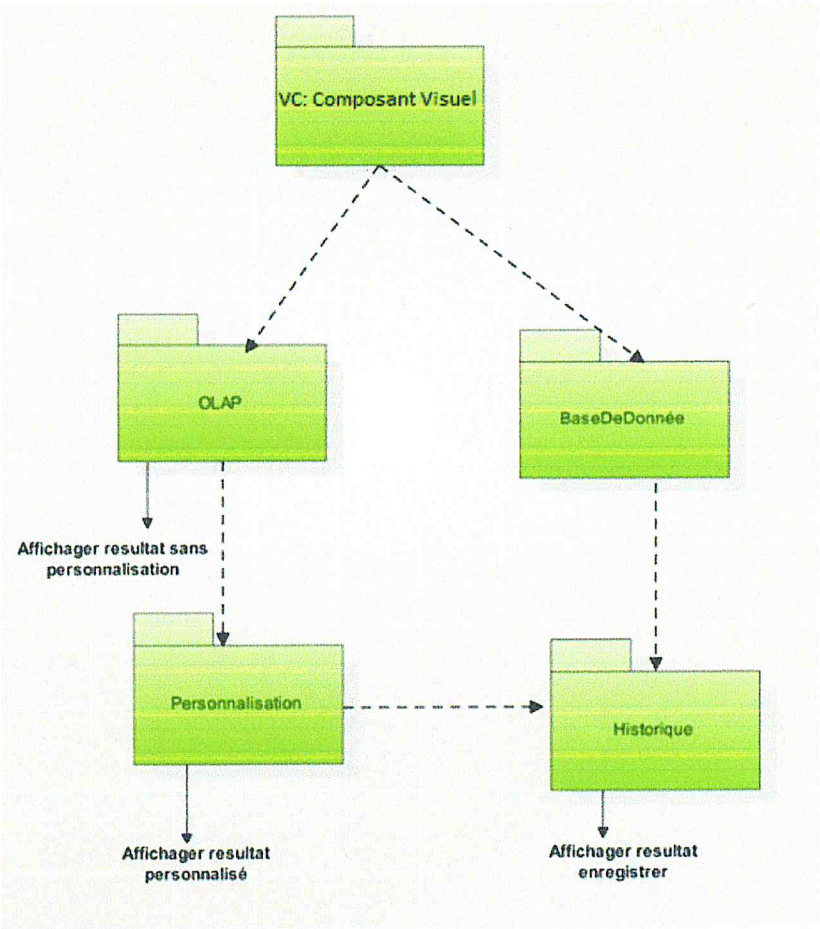


Figure.21: Le Diagramme de classes Par Package.

Composant visuel : Le système est flexible pour accepter tous les différents type de GUI (API Windows, API Unix, Web, …etc), ce package contient les classes qui représentent les interfaces graphique à l'utilisateur.

OLAP : Le package OLAP contiens les classes qui peuvent connecter à l'entrepôt de donnée, retourner les informations de l'entrepôt et exécuter les requêtes MDX …etc.

Personnalisation : ce package contiens les classes qui permettent d'exécuter l'algorithme de personnalisation.

Historique : le package Historique englobe les classes qui permettent de sauvegarder les requêtes et exploiter le résultat de l'historique.

BaseDeDonnée : ce dernier a les classes qui consistent de traiter toutes les données, et gère l'entré sortie de la base données

7.5 Diagramme de classes :

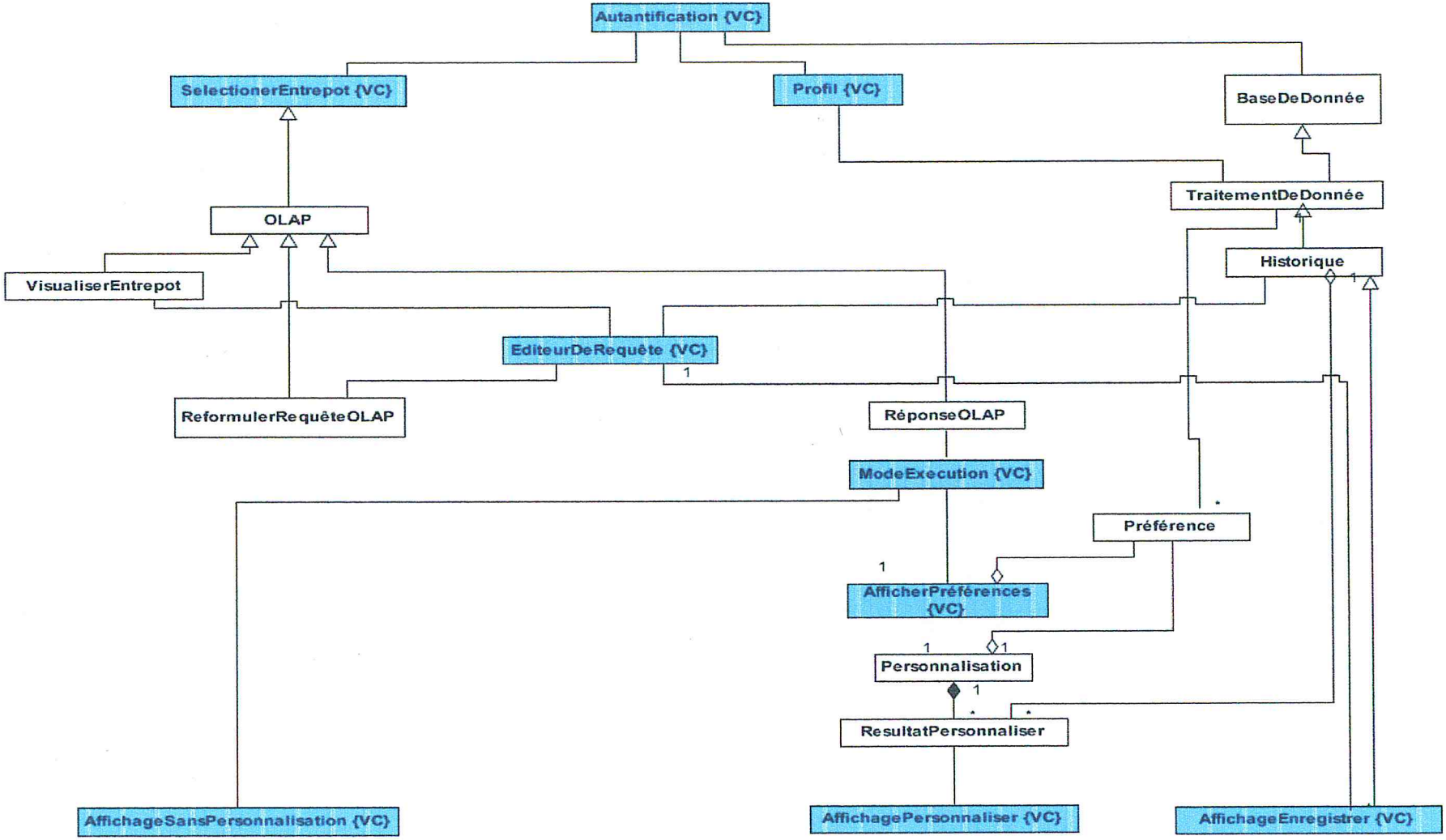


Figure.22 : Le Diagramme de classes.

7.6 Les classes :

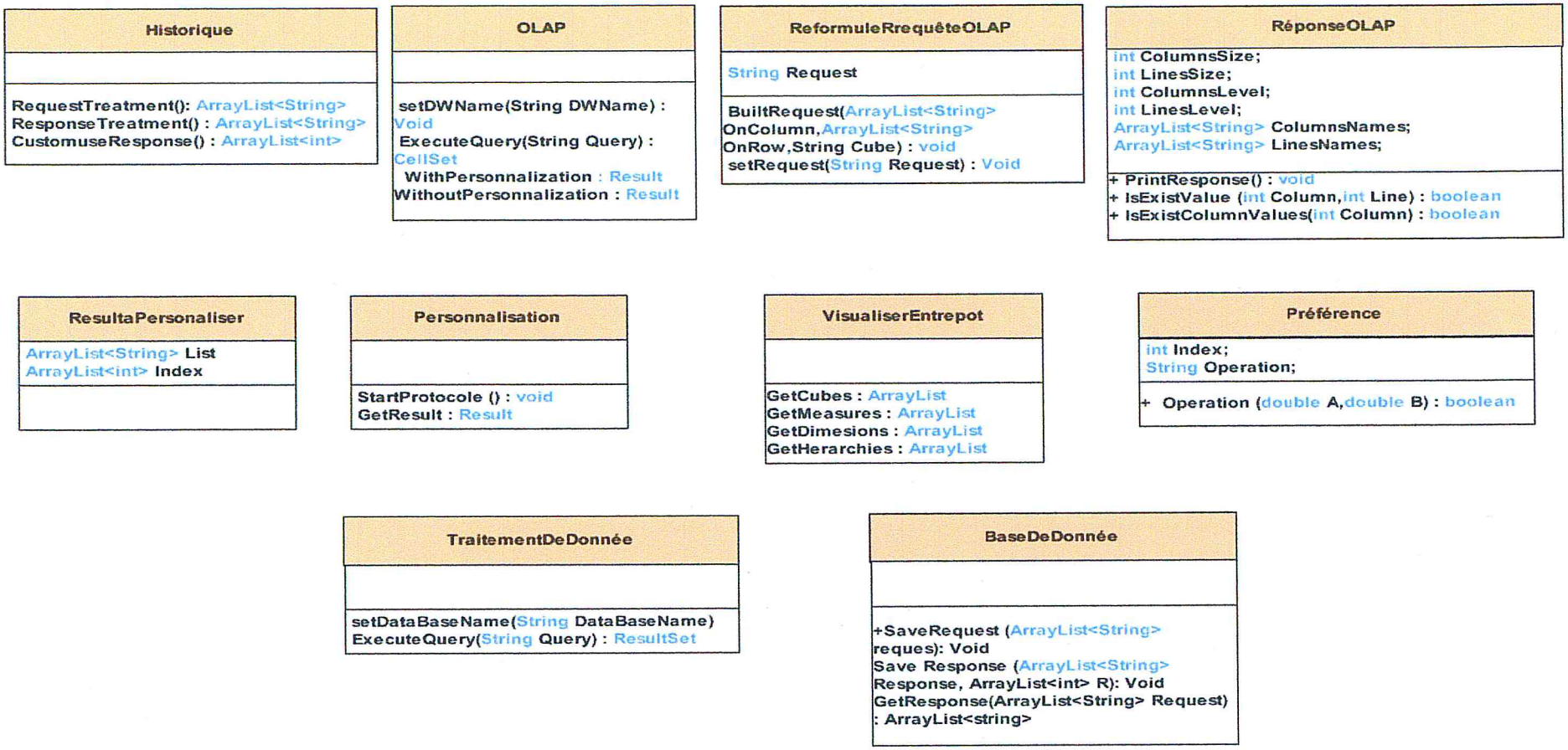


Figure.23 : Les classes.

➢ VC (Composant visuel) :

C'est un composent graphique qui contient un groupe des classes présentent les interfaces de l'application, on peut les changer par n'importe quel autre système graphique (détaille dans chapitre -05-). Par exemple on a utilisé un composent visuel web JSP qui contient :

1. Authentification :

Elle est responsable de l'authentification, il suffit un nom d'utilisateur et un mot de passe valident pour activer l'application, après elle ouvre **Profil** et **SelectionnerEntrepot.**

2. Profil :

Elle fournit à **BaseDeDonnée** les données de l'utilisateur. D'après elle l'utilisateur peut mettre à jour leurs propres données de profil pour les enregistrer dans la base, puis elle assure que l'utilisateur peut choisir l'entrepôt de données pour l'envoyer à **OLAP**.

3. SelectionnerEnrepôt :

D'après cette interface l'utilisateur peut choisir l'entrepôt de donnée, pour l'envoyer à la classe **OLAP**.

4. EditeurDeRequête :

C'est l'interface qui permet à l'utilisateur de facilité la formulation de la requête MDX (par glissement), ou elle retourner le résultat directement de l'historique.

Elle participe avec **ReformulerRequêteOALP** pour lui envoyer la requête à la classe **OLAP**.

5. **ModeExecution :**

Cette classe garantie à l'utilisateur de choisir le mode d'exécution de la requête, sans personnalisation ou bien avec personnalisation.

6. **AffichageSansPersonnalisation:**

Elle affiche la réponse de la requête sans personnalisation.

7. **AffichagePersonnaliser:**

Elle affiche la réponse de la requête avec personnalisation.

8. **AfficherPréférnces :**

D'après cette interface l'utilisateur peut choisir les préférences (par MIN, MAX, Plus proche de, …etc.) pour l'utiliser dans l'algorithme en cas de personnalisation.

9. **AfficherEnregister :**

Elle affiche le résultat d'une requête directement de l'historique.

- **OLAP :**

C'est elle qui fait la connexion vers l'entrepôt de donné, et exécute les requêtes MDX, elle fournit les cubes de l'entrepôt ainsi que les dimensions, les hiérarchies et les niveaux (toutes les informations concernant l'entrepôt) à la classe **VisualiserEntrepot**.

- **VisualiserEntrepôt:**

Elle reçoit les données de l'entrepôt (classe **OLAP**) qui permet de récupérer et formuler tous les informations concerne l'entrepôt de donnée (cubes, les dimensions, ...etc.) pour les afficher dans **EditeurDeRequête** .

- **RefomulerRequêteOLAP :**

Elle formule la requête MDX et la transmet vers **OLAP**.

- **RéponseOLAP :**

Elle réussit la réponse informelle de l'exécution de la classe **OLAP** (héritage) et applique des traitements pour convertir à un format matriciel (avec le protocole XMLA).

- **Préférence :**

C'est une structure qui formule les préférences avec leurs opérations.

- **Personnalisation :**

Elle est responsable d'implémenter l'algorithme de la personnalisation (implémenter par arbre binaire), elle hérite **RéponseOLAP.**

Elle lance le protocole par la création de l'arbre descendant récursivement.
Elle lance la dominance dans chaque feuille de l'arbre récursivement.
Elle commence la dominance ascendant par les feuilles jusqu'à arriver à la tête de l'arbre récursivement (entre chaque 2 fils).
Elle formule le résultat dans la tête de l'arbre pour l'enregistrer dans la base et même pour l'afficher.

- **RésultatPersonnaliser :**

C'est une structure qui formule le résultat de la personnalisation.

- **Historique :**

Cette classe responsable de sauvegarder la requête, les préférences choisis et la réponse dans la base de donnée, elle donne aussi le résultat d'une requête sauvegardé précédemment.

- **TraitementDeDonnée :**

Elle hérite **BaseDeDonnée**, elle est responsable de tous les traitements sur les données.

BaseDeDonnée :

C'est elle qui fait la connexion vers la base de données, et exécute les requêtes.

8 Conclusion :

Dans ce chapitre, nous avons vu la description des différents principaux modules dans le système de l'application et le fonctionnement en général.

Nous avons défini l'algorithme choisis pour la personnalisation et la présentation abstraite avec le langage de modélisation UML (02 diagrammes), ça nous donne un itinéraire vers le chapitre suivant pour présenter la solution réelle (l'implémentation).

Partie

III

Mise en œuvre

- Implémentation

Implémentation

1 Introduction :

La réponse à une requête OLAP interrogeant des quantités de données importantes et même pour la taille. En effet, l'utilisateur doit naviguer longuement afin d'obtenir l'information pertinente qu'il recherche.

Pour palier à ce problème, nous avons présenté dans le chapitre précédent notre approche de la personnalisation des requêtes MDX prenant en compte les opérateurs skylines. Cette approche est concrétisée par notre système qu'on va présenter dans ce chapitre.

2 Présentation de l'entrepôt utilisé :

Les données à manipuler sont celles de « Adevnture Works 2008». Entrepôt de données de Microsoft utilisé dans la documentation en ligne de SQL Server.L'entrepôt contient des données relatives à deux sous-domaines d'activité, les finances et les ventes.

2.1 Les tables de fait de l'entrepot :

L'entrepôt contient des données relatives à deux sous-domaines d'activité, les finances et les ventes.

2.1.1 Finance

Dans l'entrepôt de données, les données financières sont réparties dans deux schémas qui présentent les caractéristiques suivantes :

Finance

- Contient les données financières de la société Adventure Works et de ses filiales.
- Contient les données en devise locale de l'organisation à laquelle elles sont associées.
- Prend en charge le groupe de mesures Finance Analysis Services.

Currency Rates

- Contient les données de conversion de devise, y compris les cours moyens journaliers et les cours en clôture par rapport au dollar US.
- Prend en charge le groupe de mesures Currency Rates Analysis Services.

2.1.2 Sales

Les données relatives aux ventes sont réparties en quatre schémas qui présentent les caractéristiques suivantes :

Reseller Sales

- Contient les ventes uniquement des revendeurs.
- Contient uniquement les commandes expédiées.
- Contient les données en dollar US et conserve ces données dans leur devise d'origine.
- Prend en charge le groupe de mesures Reseller Sales Analysis Services.

Sales Summary

- Contient une vue de synthèse des données des ventes effectuées par les revendeurs et des ventes Internet.
- Les dimensions sont limitées par rapport aux schémas des revendeurs et des ventes Internet.

Internet Sales

- Contient les commandes individuelles des clients Internet et des données détaillées.
- Contient uniquement les commandes expédiées.
- Contient les données en dollar US et conserve ces données dans leur devise d'origine.
- Prend en charge le groupe de mesures Internet Sales Analysis Services.

Sales Quota

- Contient les données des quotas de ventes des représentants.
- Prend en charge le groupe de mesures Sales Quotas Analysis Services.

2.2 Schéma en constulation de l'entrepôt :

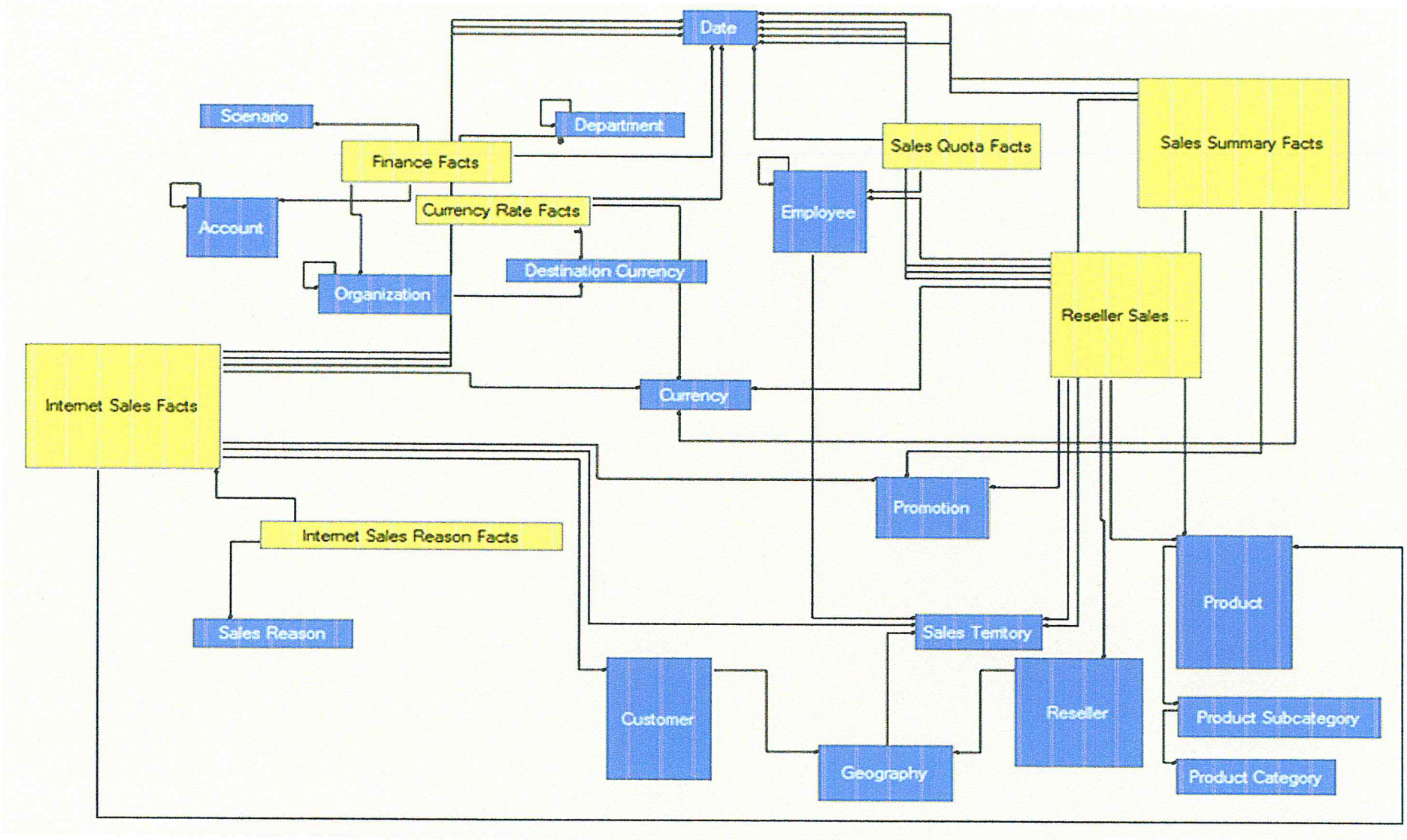


Figure.24 : Schéma en constellation de l'entrepôt Adventure works 2008.

3 Environnement de développement :

3.1 Langage de programmation :

Le langage utilisé pour la réalisation de système est le Java. Le choix de ce langage n'est pas dû au hasard, Java met à la disposition du développeur une API (Application Programming Interface) très riche lui permettant de faire de très nombreuses choses.

Le choix approprié :

- Il a une bibliothèque d'exécution indépendante de la plateforme: en théorie, il vous est possible d'utiliser le même code pour Windows 95/98/NT, Solaris UNIX Macintosh, etc.
- La mémoire dans java est allouée et libérée automatiquement pour ne jamais se préoccuper des pertes de mémoires. Les concepteurs ont supprimé l'allocation et la libération de mémoire manuelles
- Ils ont éliminé l'arithmétique des pointeurs introduisant du même coup une vraie gestion de tableau pour supprimer la possibilité d'écraser toute zone mémoire à cause d'un compteur erroné.
- Ils ont éliminé toute possibilité de confusion entre une affectation et un test d'égalité ans une instruction conditionnelle. Une instruction if (ntries - 3) ne pourra pas franchir l'étape de la compilation.
- Il utilise la notion d'interface pour remplacer l'héritage multiple. Les interfaces offrent tous ce que nous pouvons obtenir à partir de l'héritage multiples.

3.2 Plate-forme :

Nous avons choisis la plate-forme NetBeans pour le développement de notre système.

Aperçu : Il est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).

Objectifs : le principal objectif de netbeans.org est de fournir tous les outils nécessaire au développeur Java en un seul téléchargement, sans nécessiter de mise à jour ou de configuration supplémentaire pour commencer un travail productif avec l'IDE.

3.3 SQL Server Analysis Services 2008 :

Il fournit des fonctions OLAP et d'exploration de données pour les applications décisionnelles. Analysis Services prend en charge OLAP en permettant de concevoir, de créer et de gérer des structures multidimensionnelles qui contiennent des données agrégées provenant d'autres sources de données, telles que des bases de données relationnelles.

3.4 Serveurs OLAP :

Le serveur utilisé dans notre application est SQL Server Analysis Services (SSAS).

SQL Server Analysis Services est la solution de traitement analytique en ligne (OLAP) de Microsoft, et c'est un outil de SQL Server. Des solutions OLAP sont au cœur de l'intelligence d'affaires à l'aide de données sur les activités à comprendre l'entreprise et à prendre des décisions qui feront la réussite en affaires.

SSAS permet aux utilisateurs de rassembler les données de plusieurs bases de données relationnelles dans un emplacement central et présente alors que les données d'une façon qui le rend facile pour les utilisateurs d'interroger les données, afficher les données à différents niveaux de détail et effectuer des calculs avancés contre les données, avec l'objectif de dérivation des informations utiles à partir des millions ou des milliards de transactions qui se produisent dans une entreprise sur une base quotidienne.

Outre ses fonctionnalités OLAP, fournit un ensemble riche de fonctionnalités d'exploration de données qui permettent aux utilisateurs de construire des modèles statistiques avancées de leurs données et d'utiliser ces modèles pour classer les données et faire des prédictions basées sur les données .

SSAS exécute des requêtes utilisant le langage MDX, Ce langage permet de créer des requêtes dont l'équivalent en langue SQL nécessiterait un grand nombre de requêtes et de temps d'exécution beaucoup plus longs.

3.5 Le MDX :

MDX est un langage de requêtes pour les bases de données multidimensionnelles, de la même manière que SQL est utilisé pour les requêtes sur les bases de données relationnelles. Dans son approche, MDX est proche du SQL sur son aspect select et where même si la similarité ne va pas plus loin. Le but des expressions multidimensionnelles MDX est de rendre aisé et intuitif l'accès aux données de différentes dimensions.

MDX est fait pour naviguer dans les bases multidimensionnelles et pour définir des requêtes sur tous les objets (dimensions, hiérarchies, niveaux, membres et cellules) afin d'obtenir une représentation sous forme de tableaux croisés.

Il en découle une approche très hiérarchisée. Tout d'abord un cube est composé de dimensions. Une dimension peut contenir une ou plusieurs hiérarchies.

3.6 Olap4j (open Java API for OLAP):

Il a été conçu pour être agnostique plate-forme Java, de sorte que vous avez la liberté de choisir la plate-forme que vous préférez. La principale distribution d'Olap4j est compatible avec Java versions 5 et 6. Il prend en charge les spécifications JDBC 3 ainsi que 4.

Olap4j est une API OLAP, il est également une spécification pour l'implémentation des bases de données d'OLAP.

Afin d'établir une connexion à un serveur distant qui n'est pas instancié par l'objet de connexion lui-même, olap4j contient un driver d'implémentation intégré XMLA.

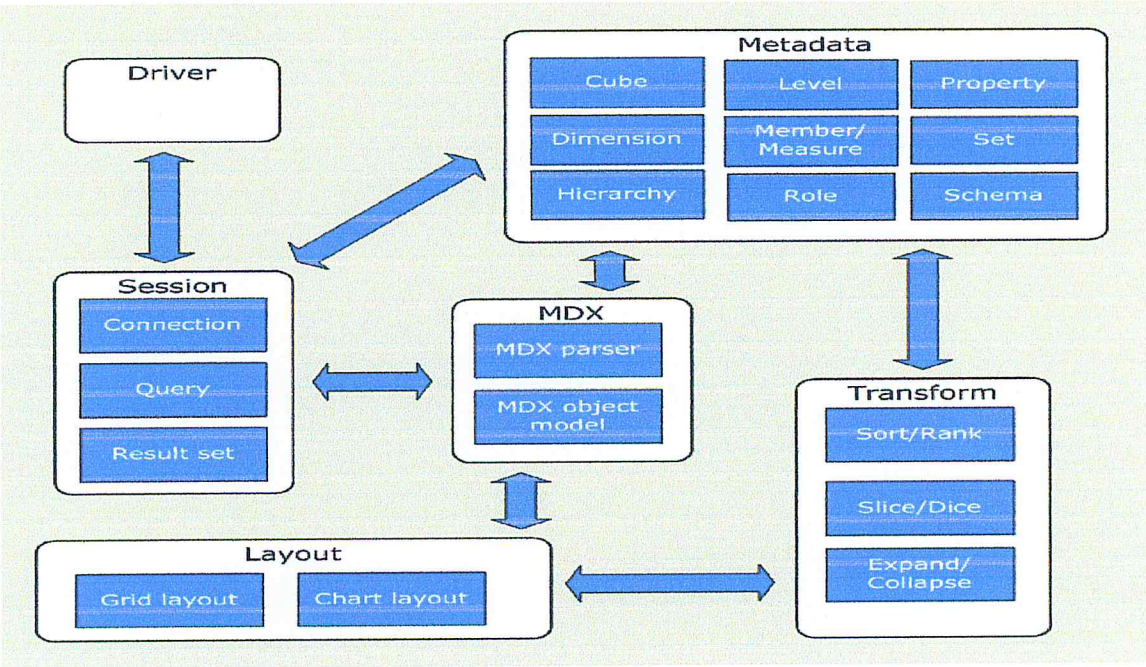


Figure.25 : API de JAVA pour OLAP

3.7 XMLA (XML for Analysis) :

Les communications avec un serveur distant OLAP sont effectuées via le protocole de transport HTTP. Il est très simple, des moyens bien connus sont utilisés pour la communication entre les clients et les serveurs. La spécification XMLA exploite la puissance de HTTP, il suffise de dire que c'est le moyen utilisé par le pilote XMLA. XMLA utilise également l'authentification HTTP comme moyen de sécurité. Il utilise l'authentification de base « BASIC security », comme son nom l'indique, elle est très basique dans les assurances, il suffit de prévoir un mécanisme d'authentification minimale.

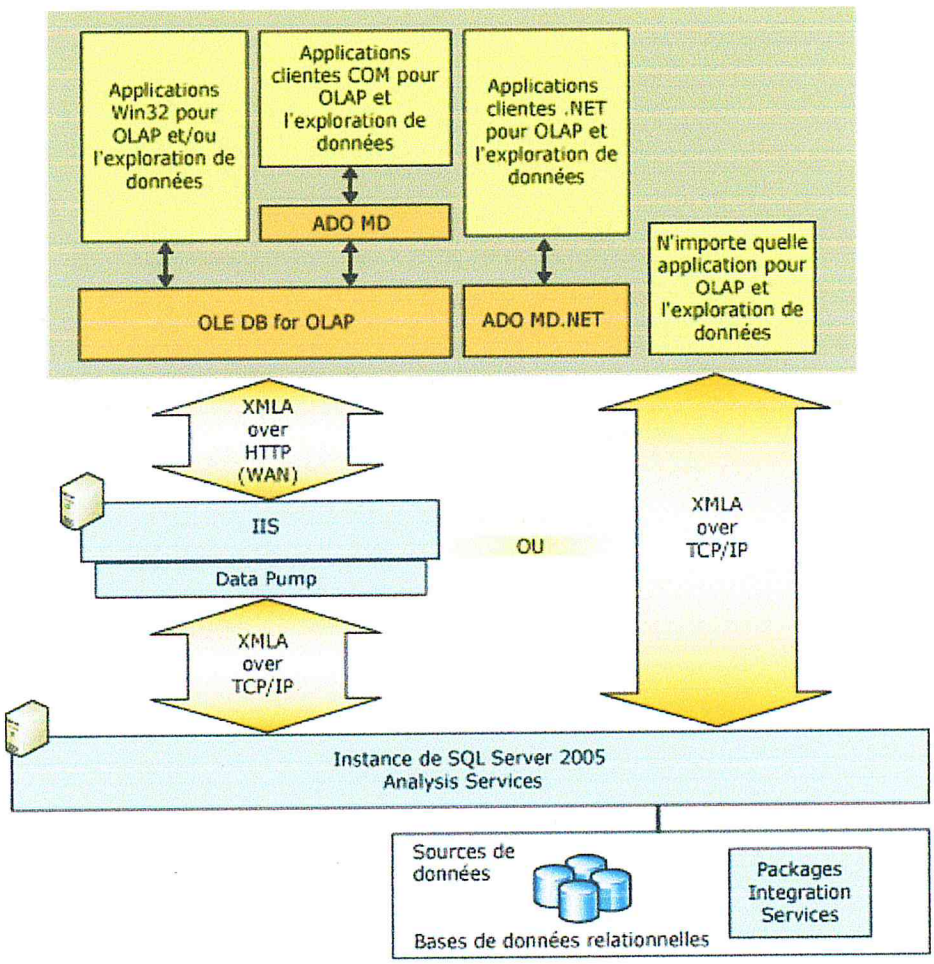


Figure.26 : XMLA permet aux applications clientes de communiquer avec les sources de données OLAP.

Pour résumer ce qui précède. Le programme Java utilise un client de services Web ou d'un conducteur de JDBC comme communicateur avec IIS. Le client de services Web ou le pilote parle XMLA à IIS. IIS traite les requêtes HTTP utilisant msmdpump.dll de les passer à SSAS.

La mise en œuvre XMLA est plus rapide à utiliser et plus léger sur le développement de Système.

4 Le Composent visuel :

Le système de l'application est flexible pour accepter n'importe quel GUI (flash, pages web, API Windows, API Lunix…etc), il faut interger essentiellement dans le package **VisualComponent.**

On prend un exemple des interfaces web, on utilise un composent qui contient des pages web JSP pour offrir à l'utilisateur les fonctions de ce système.

Notre composent visuel se compose de :

4.1 Page d'authentification :

L'utilisateur qui accède le système, il faut passer par la page web d'authentification pour introduire son nom d'utilisateur et son mot de passe.

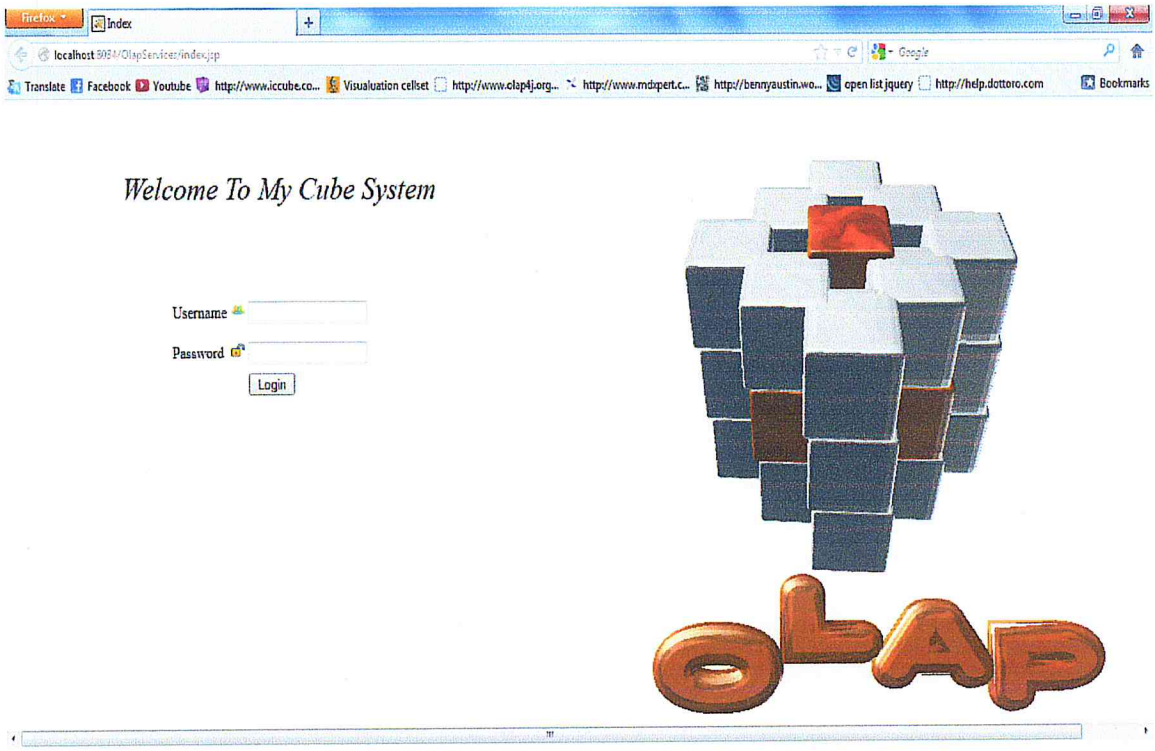


Figure.27 : page d'authentification

4.2 Editeur de requête :

A partir une classe qui s'appelle **VisualiserEntrepôt** dans notre système qu'il peut retourner n'importe quelles informations sur l'entrepôt de données analysé, c'est une classe très importante pour visualiser l'entrepôt dans un éditeur de requête pour faciliter l'édition des requêtes.

Dans cette page, les cubes, les dimensions et les hiérarchies sont visualisées de manière qui permet à l'utilisateur de rédiger les requêtes MDX par une manière très simple et facile (par glissement), elle inclut aussi certaine des fonctions MDX (MEMBERS, CHLDREN, JOIN, …etc).

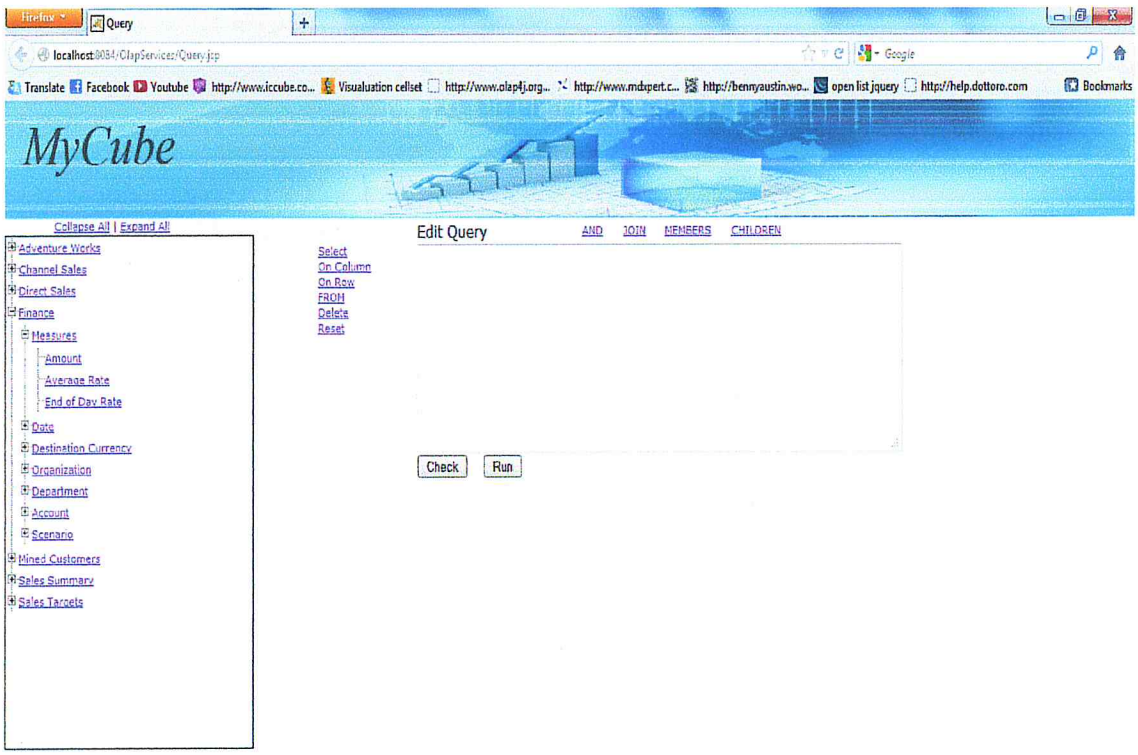


Figure.28: éditeur des requêtes.

4.3 Afficher le résultat :

C'est la page qui affiche le résultat des requêtes MDX à l'utilisateur.

Query Result

			Order Count	Order Count	Order Count	Order Count	Sales Amount Quota	Sales Amount Quota	Sales Amount Quota	Sales Amount Quota
			FY 2002	FY 2003	FY 2004	FY 2005	FY 2002	FY 2003	FY 2004	FY 2005
Accessories	All Promotions	All Employees	245	434	17,977	867	Null	Null	Null	Null
Bikes	All Promotions	All Employees	2,903	4,215	11,240	Null	Null	Null	Null	Null
Clothing	All Promotions	All Employees	465	824	8,231	351	Null	Null	Null	Null
Components	All Promotions	All Employees	392	903	1,351	Null	Null	Null	Null	Null

Figure.29: Page web présente résultat d'une requête MDX

5 Le Système :

5.1 Le déroulement de système de l'application:

Le déroulement de système de personnalisation passe à trois étapes :

- Authentification
- Sélection de l'entrepôt
- Détermination de la requête

La figure.42 montre les étapes utilisé pour arriver à obtenir une réponse personnalisé

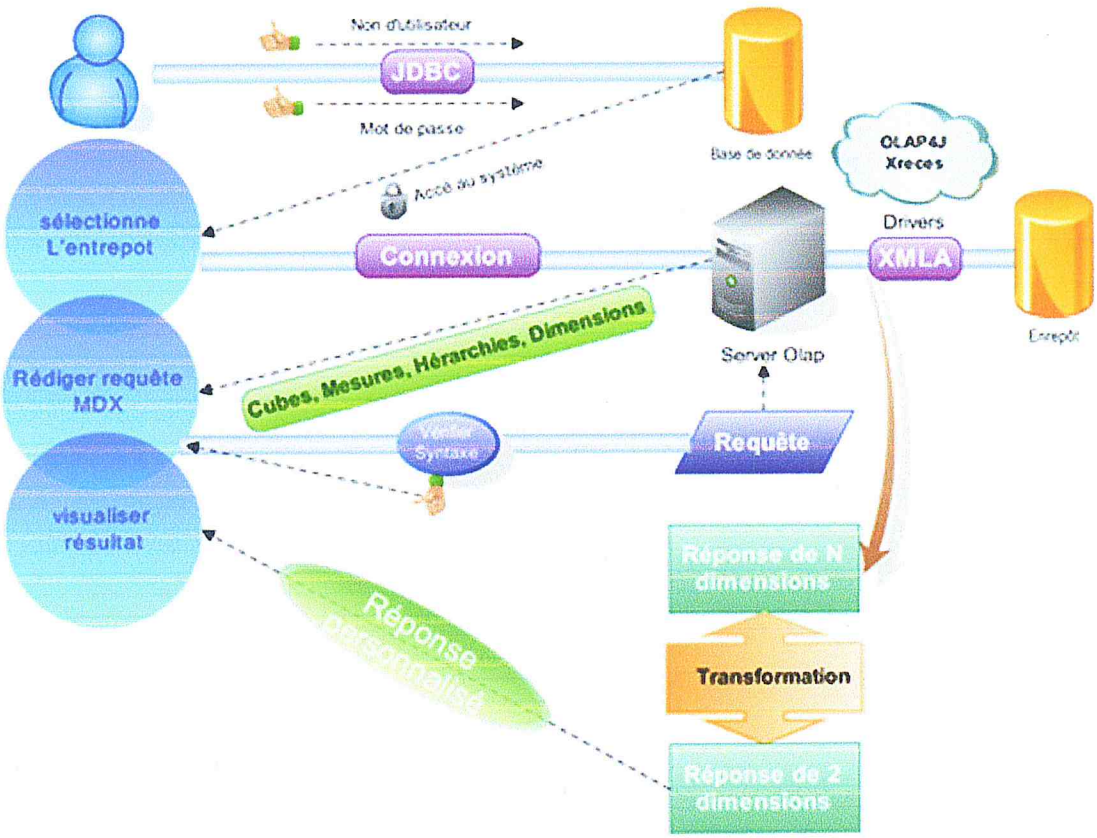


Figure.30: Le déroulement de système.

Exemple : Supposons que l'utilisateur veut exécuter la requête suivante

```
SELECT {[Measures].[Internet Average Sales Amount],
[Measures].[Internet Sales Amount], [Measures].[Internet Average Unit
Price]} ONCOLUMNS,
{[Date].[Month of Year].CHILDREN}  ONROWS
FROM [Adventure Works]
```

Avec les critères :

- Montant Moyen des Ventes (Internet) MIN
- Montant des Ventes (Internet) MIN
- Prix Unitaire Moyen (Internet) MIN

La réponse sans personnalisation est présentée dans le tableau 06

		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	$ 1,041.13	$ 2,375,856.68	$ 473.56
1	Février	$ 1,088.94	$ 2,502,386.86	$ 494.64
2	Mars	$ 1,085.95	$ 2,610,615.17	$ 504.17
3	Avril	$ 1,095.76	$ 2,778,842.08	$ 497.20
4	Mai	$ 1,130.54	$ 3,114,646.27	$ 513.63
5	Juin	$ 1,165.17	$ 3,180,923.99	$ 523.18
6	Juillet	$ 1,011.79	$ 1,911,262.79	$ 475.56
7	Août	$ 944.61	$ 1,899,606.67	$ 446.34
8	Septembre	$ 940.37	$ 1,834,668.15	$ 433.83
9	Octobre	$ 970.61	$ 2,009,169.29	$ 442.94
10	Novembre	$ 1,001.96	$ 2,076,069.60	$ 457.69
11	Décembre	$ 1,151.68	$ 3,064,629.66	$ 525.22

Tableu.06 : La réponse d'OLAP de la requête.

5.2 Le déroulement de l'algorithme de personnalisation:

On va montrer par la suite le déroulement de l'algorithme déviser pour régner implémenté par un arbre binaire.

5.2.1 Structure de données :

NœudCritère : il contient l'indice de critère et son opération, formé une liste,

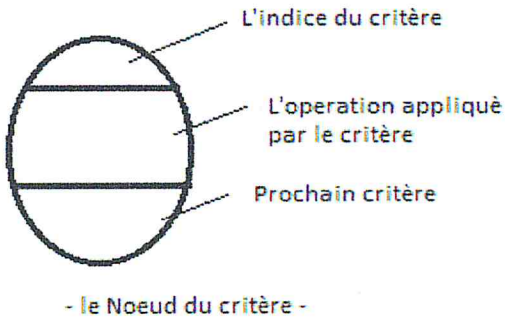


Figure.31 : Le nœud de critère.

ListeCritère : Est une liste qui contient les **NœudCritère** utilisé dans l'algorithme.

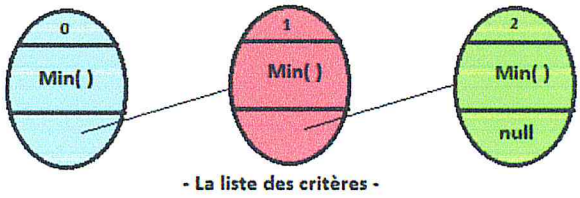


Figure.32 : Liste des critères

Nœud : Est une structure formée un nœud de l'arbre

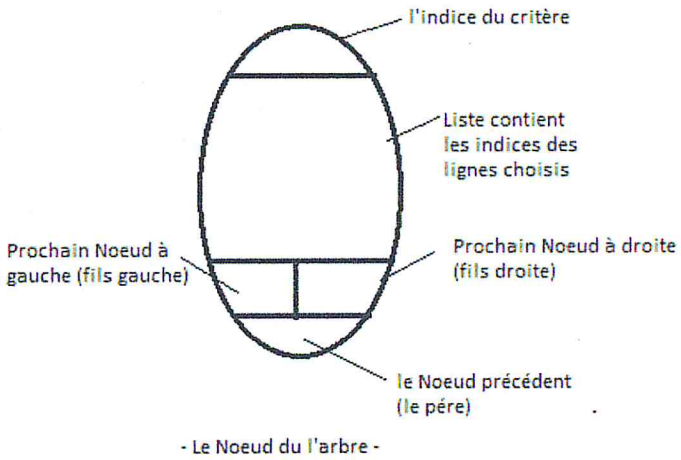


Figure.33 : Nœud de l'arbre

Arbre : Est un arbre binaire utilisé pour exécuter l'algorithme.

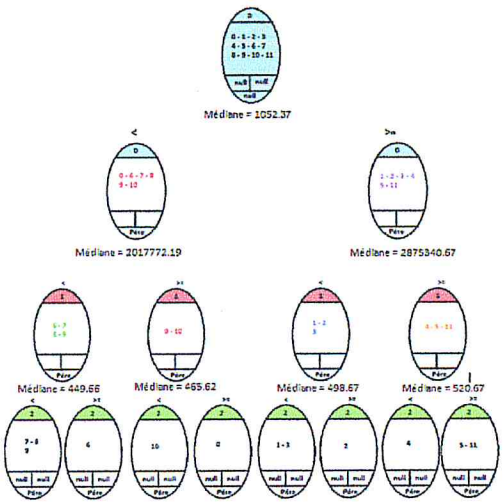


Figure.34 : L'arbre de personnalisation.

Matrice : présente la réponse d'une requête MDX (le format matricielle)

Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

- La Matrice de la personnalisation -

Tableau.07 : Réponse d'OLAP sous format matricielle.

5.2.2 L'algorithme de Personnalisation :

```
FonctionDémarerProtocole() ;
    Debut
        Initialiser() ;
        CréerArbre(Arbre T) ;
        DominationFondamentale() ;
        DominationGlobale() ;
    Fin ;
```

FonctionInitialiser ()

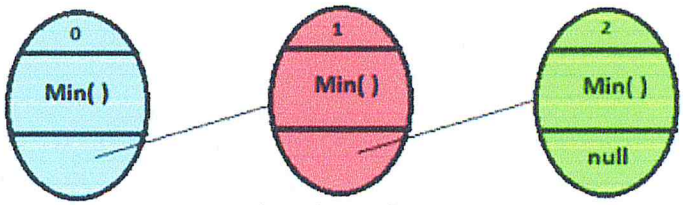
Debut

Créer le premier nœud **de l'Arbre**

Tête.IndiceCritère = 0 ;

Tête.Liste = les indices des valeurs de colonne (0) ;

Fin ;



- La liste des critères -

Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

- La Matrice de la personnalisation -

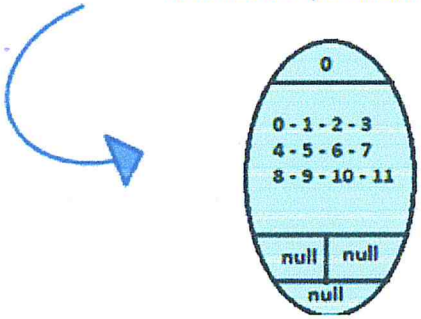


Figure.35 : 1er fonction de l'algorithme, création la tête de l'arbre.

Fonction Médiane(Arbre Courant**)**

Debut

Calculer le médiane des éléments de Courant ;

Fin ;

Fonction CreerArbre(Arbre Courant**)**

Debut

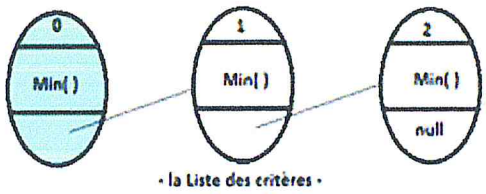
Pour chaque critère on crée deux fils du courant

Le **FilsGauche** contient les indices éléments qui sont inférieur ou égale le **médiane**(courant) ;

Le **FilsDroite** contient les indices éléments qui sont inférieur ou égale le **médiane** (courant) ;

Fin ;

Créer le premier niveau de l'arbre (1er critère) :



Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

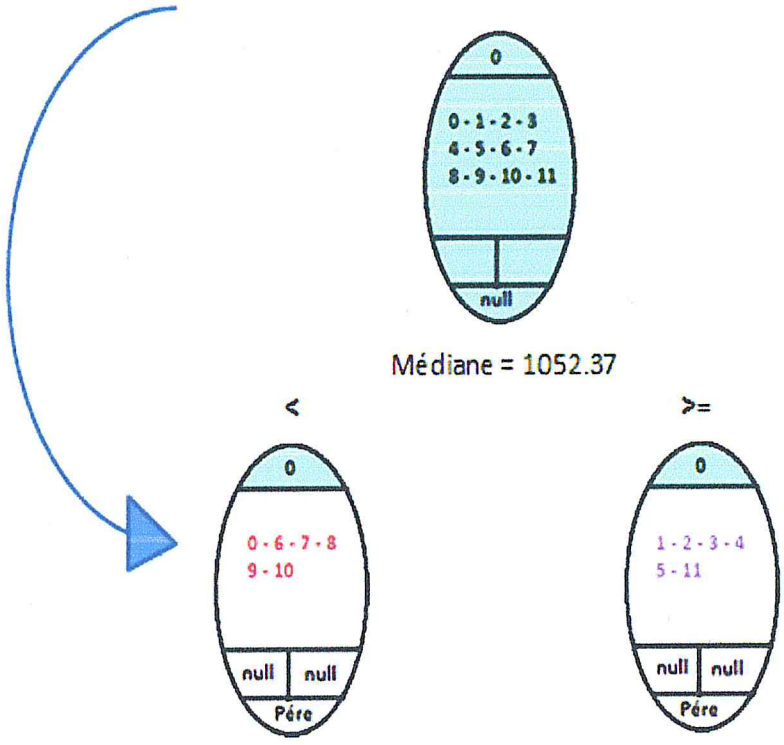
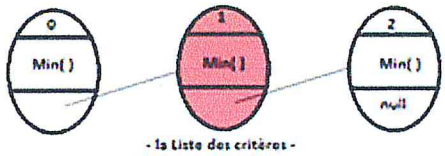


Figure.36 : Création de 1er niveau de l'arbre.

Créer le 2éme niveau de l'arbre (2éme critère) :



Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

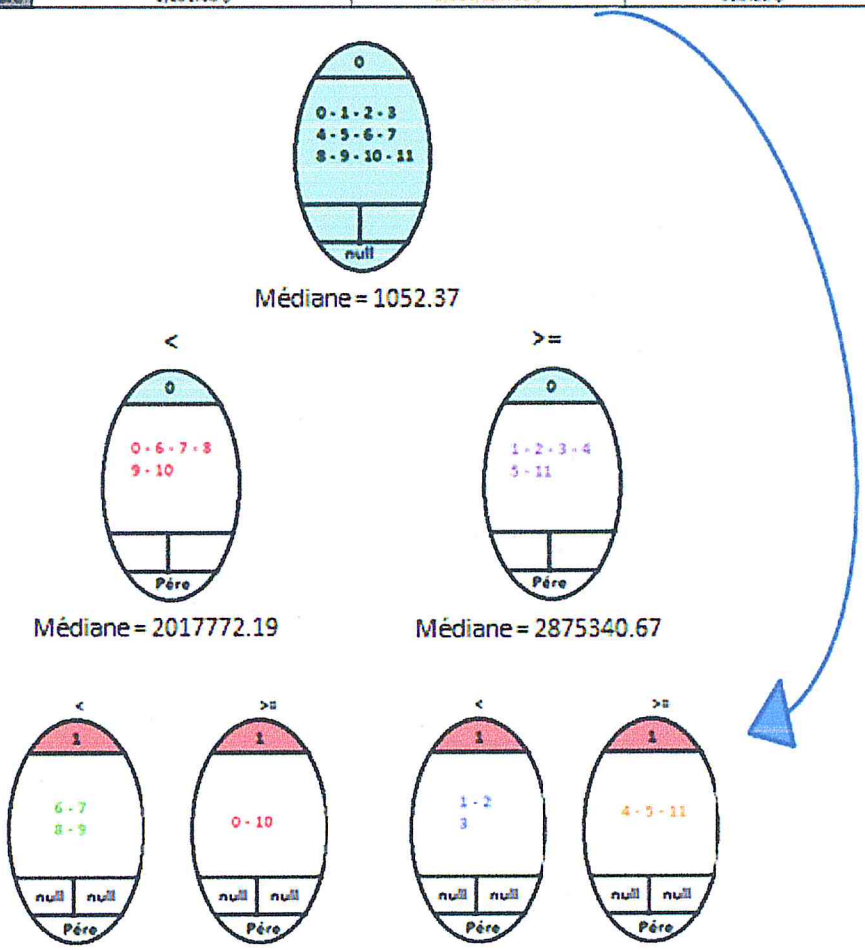
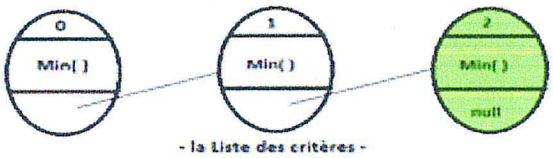


Figure.37 : Création de 2éme niveau de l'arbre.

Créer le 3éme niveau de l'arbre (3éme critère)



- la Liste des critères -

Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

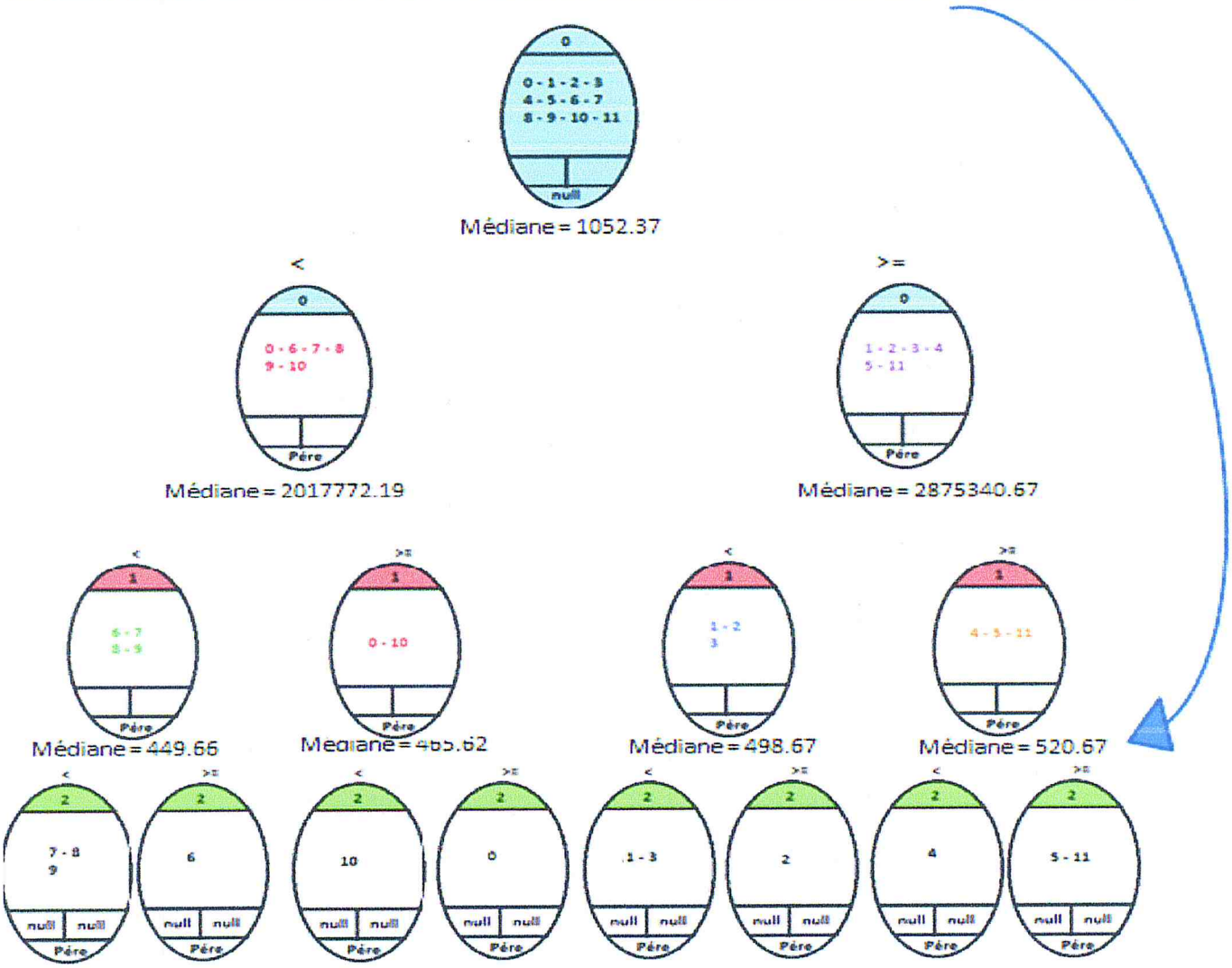


Figure.38 : Création de 3éme niveau de l'arbre.

FonctionDominationFondamentale()

Debut

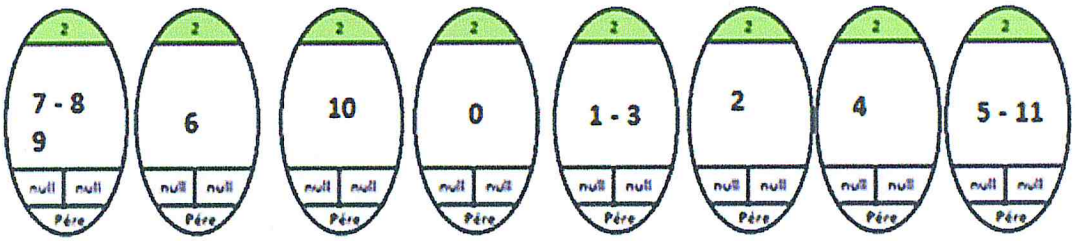
Pour chaque feuille de l'arbre

Lancer la dominance entre ses éléments (elle-même);

Fin ;

Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

- La Matrice de la personnalisation -



Aprés la dominance (DominationFondamentale)

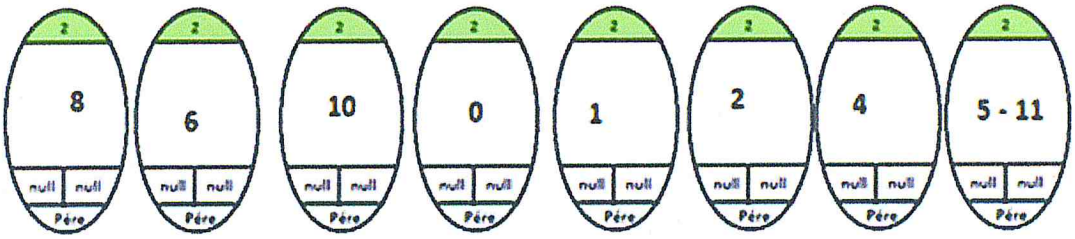


Figure.39 : Dominance dans chaque feuille.

FonctionDominationGlobale()

Debut

Commencer par les feuilles jusqu'à arriver à la tête de l'arbre

Lancer la dominance entre chaque 2 fils ;

Fin ;

Indice		0	1	2
		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
0	Janvier	1,041.13 $	2,375,856.68 $	473.56 $
1	Février	1,088.94 $	2,502,386.86 $	494.64 $
2	Mars	1,085.95 $	2,610,615.17 $	504.17 $
3	Avril	1,095.76 $	2,778,842.08 $	497.20 $
4	Mai	1,130.54 $	3,114,646.27 $	513.63 $
5	Juin	1,165.17 $	3,180,923.99 $	523.18 $
6	Juillet	1,011.79 $	1,911,262.79 $	475.56 $
7	Août	944.61 $	1,899,606.67 $	446.34 $
8	Septembre	940.37 $	1,834,668.15 $	433.83 $
9	Octobre	970.61 $	2,009,169.29 $	442.94 $
10	Novembre	1,001.96 $	2,076,069.60 $	457.69 $
11	Décembre	1,151.68 $	3,064,629.66 $	525.22 $

- La Matrice de la personnalisation -

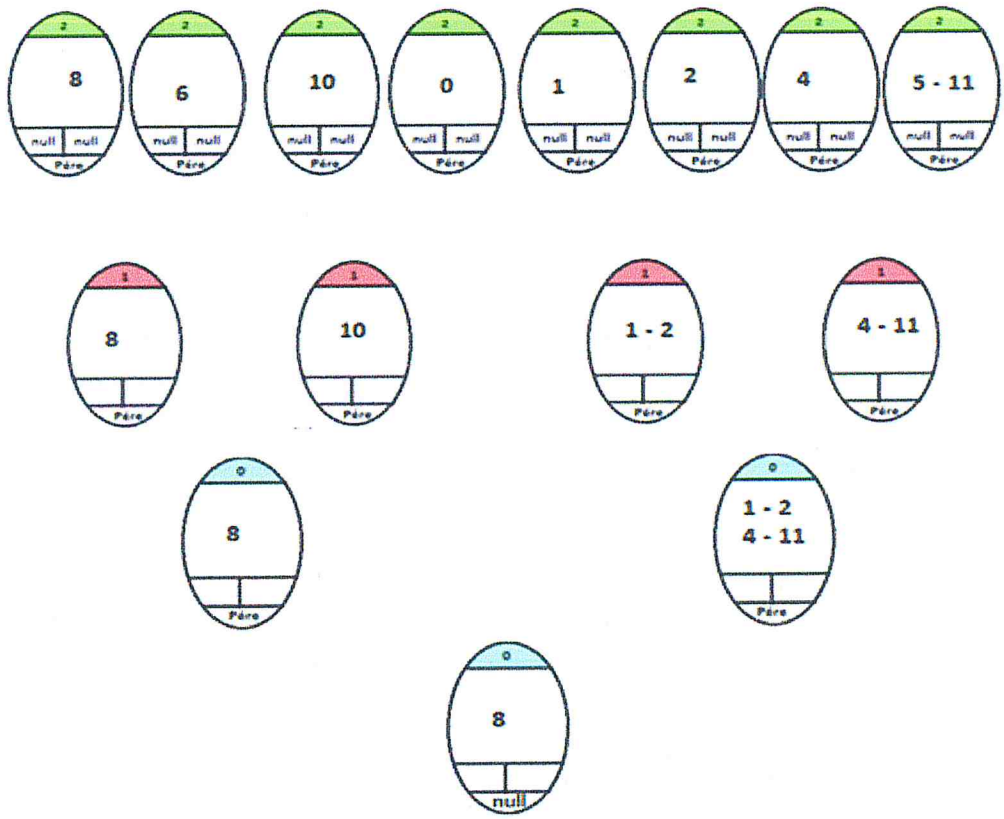


Figure.40 : Dominance entre chaque deux fils jusqu'à arriver à la tête.

FonctionResult()

Debut

Resultat = liste des indices de la tête ;

Fin ;

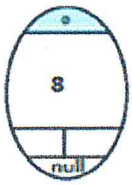


Figure.41: Résultat.

La réponse personnalisée est présentée dans le tableau 07.

		Montant Moyen des Ventes (Internet)	Montant des Ventes (Internet)	Prix Unitaire Moyen (Internet)
8	Septembre	$ 940.37	$ 1,834,668.15	$ 433.83

Tableau.08 : Réponse personnalisée.

5.3 Système d'erreur :

Pour le but de donner à l'utilisateur un système fiable, on a ajouté un mini système qui s'appelle **Interface**, c'est une interface entre le système interne de l'application et l'utilisateur, elle gère les erreurs et les exceptions.

Les exceptions représentent le mécanisme de gestion des erreurs intégré au langage Java. Il se compose d'objets représentant les erreurs et d'un ensemble des mots clés qui permettent de détecter et de traiter ces erreurs.

Lors de la détection d'une erreur, un objet qui hérite de la classe **Interface** informe l'utilisateur quel est l'obstacle qui perturbe le bon déroulement du système par message d'erreur.

La classe **Interface** identifie les zones du programme dans lesquelles une opération peut conduire à une situation exceptionnelle, récupéré les exceptions qui seront créés en cas de problème pour informer l'utilisateur.

Ces mécanismes permettent de renforcer la sécurité du code Java et garantir le bon fonctionnement du sytème intrne.

6 Le système en exécution :

Pour mettre notre système fournir un service fiable à l'utilisateur, on doit prendre le temps d'exécution en considération et poser la question combien de temps faut prendre le système pour lui répondre ?

Pour répondre à cette question, une comparaison entre l'exécution avec personnalisation et l'exécution sans personnalisation est faite.

	Temps d'exécution	
requête	Sans personnalisation	Avec personnalisation
SELECT {[Measures].[Internet Average Sales Amount], [Measures].[Internet Sales Amount], [Measures].[Internet Average Unit Price]} ON COLUMNS, {[Date].[Month of Year].CHILDREN} ON ROWS FROM [Adventure Works]	4 second	6 second
SELECT {[Measures].[Order Count], [Measures].[Customer Count], [Measures].[Internet Average Unit Price]} ON COLUMNS, {[Promotion].[Promotion Category].[All Promotions].children * [Date].[Fiscal].[Fiscal Year].Members * [Product].[Product Categories].[Category].Members * [Sales Channel].[Sales Channel].Members} ON ROWS FROM [Adventure Works]	7 second	9 second

Tableau.09 : Comparaison entre les modes d'exécution

Solution : pour avoir amélioré le temps d'exécution de notre système on a proposé un mécanisme pour d'enregistrer la réponse personnalisé dans la base de donnée, alors que l'utilisateur veux exécuter une requête déjà réalisé, le système affiche la réponse directement sans passer par l'algorithme, donc on a amélioré le temps d'exécution par un temps d'accès directe a une valeur.

7 Conclusion :

La partie réalisation représente la dernière phase dans le développement de l'application, nous avons défini l'architecture matérielle de notre système et aussi, nous avons présenté les choix effectuées pour les différents outils de développement et nous avons montré les interfaces de ce système.

Nous avons eu un aperçu sur le système interne et le fonctionnement en générale ainsi que le temps d'exécution.

Conclusion Générale

Conclusion générale

Au terme de ce rapport, nous présentons une synthèse sur les acquis dans le domaine de la personnalisation des requêtes MDX et plus précisément dans le contexte des entrepôts de données.

Le but de la personnalisation est de faciliter l'expression du besoin d'utilisateur et de lui permettre d'obtenir des informations pertinentes lors de ses accès à un système d'information

La personnalisation dans les bases de données multidimensionnelle, où L'entrepôt est conçu pour supporter des requêtes complexes de décision (requêtes OLAP) dont les résultats sont visualisés sous forme de tableaux croisés (ces résultats peuvent être très volumineux et souvent ils ne peuvent être visualisés

Notre système de personnalisation reçoit une requête MDX provenant de l'utilisateur, puis il applique l'algorithme (deviser pour régner) de personnalisation pour trouver l'ensemble skyline de la requête émise par l'utilisateur et d'après ses choix de préférences.

Enfin, notons qu'il existe peu de travaux sur la personnalisation de l'information dans le domaine des entrepôts de donnée, et notre travail n'est qu'une démarche initiale dans ce domaine et ne peux être qualifié de parfait.

Nous avons donc présenté les entrepôts de données et illustré, à l'aide d'exemples, les différentes possibilités qu'ils offrent. Ensuite, nous avons décrit les systèmes OLAP et les opérations effectuées sur les cubes de données.

Nous avons fait un tour de table sur les travaux entrepris dans le domaine de personnalisation dans les entrepôts de données, ainsi une étude comparative entre les différentes méthodes de personnalisation a été faite.

La personnalisation dans le domaine des entrepôts de données est un axe émergent et fait l'objet de recherches actives. De ce fait nous avons rencontré des difficultés durant la réalisation de notre système.

Rappelons que nous avons utilisé l'approche de l'opérateur skyline définit par **Börzsönyi et al. (2001)** pour la personnalisation et nous avons choisi l'algorithme deviser pour régner implémenter par arbre binaire pour la réaliser.

Finalement, le résultat du test est effectué sur notre prototype, en vue de confirmer son efficacité.

Perspectives

Nos propositions ouvrent plusieurs perspectives de recherche qui permettent d'améliorer en générale la personnalisation de l'accès à l'information dans les systèmes d'information et en particulier le fonctionnement de notre approche de personnalisation, nous pouvons ajouter à notre système :

1) Amélioration de la personnalisation de la présentation

La personnalisation de la présentation dans notre système dépend de la personnalisation du contenu. Cette personnalisation concerne essentiellement la présentation des informations personnalisées dans l'interface de navigation. Aucune adaptation, proprement dite, de l'interface n'est prévue dans *ce* système. Donc il est intéressant d'enrichir et d'approfondir cette personnalisation des outils qui permettent de personnaliser également l'interface.

2) Acquisition des données des profils

Définition d'un prototype qui permet l'acquisition implicite des données des utilisateurs à travers l'analyse des historiques de leurs interactions avec le système en utilisant par exemple un fichier log.

3) Développer le système de l'application pour fonctionner sous réseau.

4) Amélioration de temps d'exécution :

Ajouter la programmation de multithreading (calculer la médiane, ... etc.).

Bibliographie

Bibliographie

[Bellatreche et al, 2005]	Article universitaire Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., and Laurent, D. (2005). **Apersonalization framework for olap queries.** Université de Cergy, FRANCE. 2005
[Bellatreche et al, 2006]	Article universitaire Bellatreche, L., Giacometti, A., Marcel, P., and Mouloudi, H. (2006). **Personalization of mdx queries.** In BDA'06 : 22èmes journées Bases de Données Avancées, Lille, 17-20 octobre 2006, Actes.
[Borzsonyi et al, 2001]	Article universitaire Borzsonyi, S., Kossmann, D., and K.Stocker (2001). **The skyline operator.** Technische Universit¨at M¨unchen D-81667 M¨unchen, Germany 2001
[Bouzeghoub et al, 2005]	Article universitaire Mokrane Bouzeghoub, Dimitre Kostadinov. **Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils.** Laboratoire PRiSM, Université de Versailles 2005
[Chomicki, 2002]	Article universitaire Chomicki, J. **Querying with intrinsic preferences.** Dept Of computer science and engeneering. University at buffalo. NewYork 2002
[Favre, 2007]	Thése Université Lumière Lyon 2 Cecile Favre. **Évolution de schémas dans les entrepôts de données : mise à jour de hiérarchies de dimension pour la personnalisation des analyses.** (2007)
[Giacometti et al, 2008]	Article universitaire Arnaud Giacometti, Patrick Marcel, Elsa Negre **A Framework for Recommending OLAP Queries** Laboratoire d'Informatique Université François Rabelais de Tours – France 2008

[Robert Vieira 2008]	**Livre** Robert Vieira **Professional Microsoft SQL Server 2008 Programming** 2009 India , ISBN 9780-470-2570-9
[Bertrand Burquier 2007]	**Livre** Thierry Petibon **Business Intelligence avec SQL Server 2005 - Mise en œuvre d'un projet décisionnel: Mise en œuvre d'un projet décisionnel** Dunod, 2007 ISBN 9782100528141
[Lakshman Bulusu 2009]	**Livre** Lakshman Bulusu **Open Source Data Warehousing and Business Intelligence** CRC Press 2013 978-1-4398-1640-0
[Christian Soutou 2012]	**Livre** Christian Soutou **UML 2 pour les bases de données** ISBN 978-2-212-13-413-1
[B.S.Ainapure 2011]	**Livre** James Rumbaugh **Object Oriented Modeling And Design** ISBN 9788184315110

Références Webographies

http://www.olap4j.org/	Drivers
http://www.mdxpert.com/	Langage MDX
http://msdn.microsoft.com/en-us/library/ms170208	Adventure Works DW
http://msdn.microsoft.com/fr-fr/library/microsoft.analysisservices.adomdclient.cellset	L'objet CellSet
http://technet.microsoft.com/fr-fr/sqlserver/bb265254.aspx	SQL Server 2008
http://www.virtualprofessors.com/mehran-sahami-programming-methodology-stanford-cs106a	**Video** Mehran Sahami Programing Methodology
http://www.virtualprofessors.com/computing-with-randomness-probability-theory-and-the-internet	**Video** Meharn sahami Probability Theory and the internet