



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Logiciel

Par :

**MECHEBAK Souad
GACEM Zohra**

Sur le thème

Personnalisation de requêtes OLAP

Soutenu publiquement le .. / 11 / 2019 à Tiaret devant le jury composé de :

Mr. MEGHAZI El Hadj	MCA Université Ibn Khaldoun	Président
Mr. BENNOUDA Habib	MCA Université Ibn Khaldoun	Encadreur
Mr. OUARED Abdelkader	MCA Université Ibn Khaldoun	Examineur

R *emerciements*

A l'issue de ce travail, nous voulons d'abord remercier Allah de nous avoir guidé et donner la force, le courage et la patience pour terminer ce travail : MERCI "Dieu"

Nous tenons à exprimer nos vifs remerciements à notre encadreur Mr. BENNOUDA Habib, pour nous avoir guider tout au long de ce projet, pour ses conseils, patiences et aides, sans lesquels notre projet n'aurait pu aboutir.

Nous tenons particulièrement à remercier très sincèrement l'ensemble des membres du jury qui me font le grand honneur d'avoir accepté de juger notre travail.

Chaleureux remerciements vont à tous nos enseignants au cours de notre formation au fil des années.

Souad

Zohra

Résumé

Un entrepôt de données stocke de gros volumes de données multidimensionnelles, consolidées et historisées dans le but d'être analysés. L'entrepôt est notamment conçu pour supporter des requêtes OLAP complexes dont les résultats sont visualisés sous forme de tableaux croisés, l'utilisateur doit naviguer longuement afin de trouver l'information pertinente recherchée. La pertinence de l'information se définit par un ensemble de critères et de préférences personnalisables spécifiques à chaque utilisateur. Les données décrivant les utilisateurs sont souvent regroupées sous forme de profils. Le contenu du profil d'un utilisateur varie selon les approches et les applications.

Ce travail a pour objectif d'étudier la personnalisation de l'information pour un utilisateur interrogeant un entrepôt par requêtes OLAP. En prenant en compte les profils des utilisateurs. Pour ce faire, chaque utilisateur de l'entrepôt de données est invité à donner ses préférences et une contrainte de visualisation. On a utilisé l'entrepôt de données FoodMart comme une base de données de test.

Mots-Clés : Entrepôt de données, personnalisation, préférences utilisateur, profil utilisateur, visualisation, contrainte, requêtes OLAP.

Abstract

A data warehouse stores large volumes of consolidated and histories multidimensional data to be analyzed. The data warehouse is in particular designed to support complex decision queries (OLAP queries) whose results are displayed under the form of cross tables, the user must navigate long to find the relevant information sought. The relevance of the information is defined by a set of criteria and preferences customizable to each user. Data describing users is often grouped as profiles. The content of a user's profile varies by approach and application.

This work aims to study the personalization of information for a user querying a warehouse by OLAP queries. Taking into account user profiles. To do this, each user of the data warehouse is prompted to give his preferences and a visualization constraint. The FoodMart data warehouse was used as a test database

Keywords : Data warehouse, personalization, user preferences, user profile, visualization, constraint, OLAP queries,

Liste des abréviations

OLAP: (On-Line Analytical Processing

BD : base de données

BDM : base de données multidimensionnelle

MDX : Multi-Dimensionnelle eXpression

DOLAP: Desktop OLAP

OOLAP: Object OLAP

ROLAP : Relational OLAP

MOLAP : Multidimensionnel OLAP)

HOLAP : Hybride OLAP)

Liste des figures

<i>Figure I.1.</i> Exemple d'un modèle en étoile [MAHMOUDI, BABA AHMED2016].....	08
<i>Figure I.2.</i> Exemple d'un modèle en flocon de neige [MAHMOUDI, BABA AHMED2016].....	08
<i>Figure I.3.</i> Exemple d'un modèle en constellation [MAHMOUDI, BABA AHMED2016]	09
<i>Figure II.1.</i> Architecture générale d'un système de recommandation [Trousse, 2001].....	25
<i>Figure II.2.</i> L'algorithme <i>Maxsubset</i> [Bellatreche et al, 2006]	37
<i>Figure II.3.</i> l'algorithme <i>Perso</i> [Bellatreche et al, 2006].....	38
<i>Figure III.1.</i> Architecteur de système de personnalisation	44

Liste de tableaux

<i>Tab II.1.</i> Comparaison des travaux dans le domaine des entrepôts [Benttayeb, 2009].....	18
<i>Tab II.2.</i> Récapitulatif des travaux sur la modélisation et l'usage des préférences OLAP.....	24

TABLE DES MATIERES

Remerciements	
Résumé	
Abstract	
Liste des abréviations	
Liste des figures	
Liste des tableaux	
Introduction générale	1

Chapitre I : Entrepôt de données

1. Introduction :	02
I.2. Datawarehouse :	05
I.2.1Caractéristiques des données d'un ED :	05
I.2.1.1. Orientées sujet et intégrées :	05
I.2.1.2. La variante de temps (historisées) :	06
I.2.1.3. Non volatiles :	06
I.3 Implémentation des entrepôts des données	06
I.3.1 Systèmes ROLAP (Relational OLAP) :	06
I.3.2 Systèmes MOLAP (Multidimensionnel OLAP) :	06
I.3.3 Systèmes HOLAP (Hybride OLAP) :	06
I.3.4 Base DOLAP (Desktop OLAP) :	07
I.3.5 Base OOLAP (Object OLAP) :	07
I.4. Construction d'un DataWarehouse :	07
I.4.1. Modèle en étoile:	07
I.4.2. Modèle en flocon de neige :	08
I.4.3 Modèles en constellation :	09
I.4.4. Différence entre le modèle en étoile et flocon de neige :	09
I.5. Architecture d'un entrepôt de données	10
I.6. Modélisation et interrogation des bases de données multidimensionnelles	10
I.6.1 Modélisation des données OLAP	10
I.6.2 Manipulations des données OLAP.....	11

Chapitre II : La personnalisation

II.1. Introduction:	15
II.2. La personnalisation :	15
II.3. Profil :	17
II.3.1. Le contenu du profil :	17
II.3.2. Modélisation du profil utilisateur :	17
II.3.3. Construction du profil :	18
II.3.4. Exploitation de profil :	18
II.3.5. Évolution du profil utilisateur :	19
II.4. Les Préférences :	19
II.4.1 Modélisation de préférences	20
II.4.1.1 Niveau des préférences.....	20
II.4.1.2 Formulation de préférences	21
II.4.2. Synthèse sur la modélisation et l'exploitation des préférences OLAP	23
II.5. La recommandation :	24
II.6. Personnalisation des systèmes OLAP	25
II.6.1 Personnalisation du schéma OLAP	26
II.6.2. Personnalisation de l'interrogation des données.....	27
II.6.3. Personnalisation de requêtes OLAP :	27
II.6.3.1. Recommandation de requêtes	29
II.6.4. Personnalisation de la visualisation des données.....	30
II.6.5. Personnalisation de la prise de décision	30
II.6.6. Personnalisation de l'interrogation des bases de données multidimensionnelles	31
II.6.6.1 Critères d'étude des travaux de personnalisation de l'interrogation des BDM	31
II.6.6.2. Personnalisation portant sur le schéma de l'entrepôt de données :	31
II.6.6.2.1 Personnalisation basée sur l'adaptation :	31
II.6.6.2.2. Personnalisation basée sur la recommandation :	31
II.7. Algorithmes pour la personnalisation	32
II.7.1 Approche de [Bellatreche et al, 2005].....	32
II.7.2 Approche de [Bellatreche et al, 2006].....	34

Chapitre III : La personnalisation des requête OLAP

III.1. Introduction :	44
III.2. Technologies employées :	44
III.3. Configuration de Mondrian avec Tomcat et MySQL Server	45
III.3.1. La base de données test FoodMart de MySQL	45
III.3.2. Création et population de la base de données.....	46
III.3.3. Configuration de Tomcat	47
III.3.4. Installation et configuration de Mondrian.....	48
III.3.5. Configuration de Mondrian avec MySQL	50
III.4 Présentation de prototype :	52
III.5. Exemple 1: Une seule requête, deux profils utilisateur, deux résultats différents :	62
III.5.1. Discussion des résultats.....	65
III.6. Exemple 2 : Discrimination entre les sous-ensembles de références	66
Conclusion générale	
Bibliographie.....	

Introduction Général

Introduction générale

Les systèmes décisionnels visent à transformer les données opérationnelles en informations pouvant être interprétées pour répondre à un ensemble de besoins d'analyse recensés auprès des utilisateurs à un moment donné. Mais les utilisateurs peuvent avoir des besoins variés auxquels l'entrepôt n'est pas forcément en mesure de répondre, a fortiori dans une grande entreprise dans laquelle les utilisateurs exercent de nombreux métiers.

Les outils OLAP sont connus pour être intuitifs car leurs utilisateurs finaux ne sont pas forcément informaticiens. Cependant, la grande volumétrie des données et la complexité des requêtes d'analyse qui impliquent beaucoup d'agrégations rendent plus difficile la tâche d'analyse aux utilisateurs.

Les utilisateurs des systèmes OLAP (On-Line Analytical Processing) formulent des requêtes décisionnelles pour répondre à des besoins d'analyse spécifiques pour l'aide à la décision.

Il est donc nécessaire d'offrir à ces derniers des solutions mieux adaptées à leur mode de raisonnement à travers des processus de recommandation et d'enrichissement de leurs requêtes d'analyse afin de ne leur retenir que les informations qualifiées pertinentes. Cependant, cette pertinence est définie par un ensemble de critères et de préférences personnalisables à chaque utilisateur ou ensemble d'utilisateurs et stockés dans une structure appelée Profil Utilisateur.

Ce processus, appelé personnalisation, constitue un champ de recherche qui reste à explorer dans le domaine des systèmes de exploration dans le domaine des systèmes décisionnels intégrant des magasins multidimensionnels [Rizzi, 2007].

L'objectif du présent travail est d'éclairer la notion de la personnalisation de l'information dans les bases de données multidimensionnelles (entrepôts de données).

Pour ce faire, notre travail est subdivisé en deux parties complémentaires dont la première évoque la notion de la personnalisation dans le domaine des bases de données multidimensionnelles (entrepôts des données) et la seconde présente un cas d'application de la personnalisation des requêtes MDX.

Problématique :

La problématique traitée dans ce rapport consiste à personnaliser les systèmes OLAP en fonction de besoins analytiques individuels. Les mécanismes de personnalisation dans les systèmes OLAP, visant à mieux prendre en compte l'utilisateur, ne sont que très peu étudiés. La personnalisation peut prendre la forme de recommandations de données décisionnelles afin de simplifier voire d'optimiser les analyses OLAP.

Introduction générale

Comment améliorer, pour un utilisateur donné, ses possibilités d'interaction avec un système décisionnel donné et achever un bon processus d'analyse ?

L'extraction d'une telle information pose un certain nombre de problèmes. Le premier est le temps de réponse élevé. Ce problème est d'autant plus crucial que l'utilisateur ne sachant pas a priori ce qu'il cherche, pose plusieurs requêtes pour trouver ce qui l'intéresse.

Le deuxième est la taille importante des réponses aux requêtes utilisateur. Le nombre d'informations trouvées est généralement très important. L'exploitation de ces réponses n'est pas toujours possible. On peut se retrouver dans des situations où des réponses ne peuvent pas être toutes explorées pour être utilisées.

C'est pourquoi on fait appel aux outils de personnalisation ou de recommandation pour trouver rapidement un ensemble réduit de réponses intéressantes et pouvant satisfaire l'utilisateur. Notre objectif est de personnaliser les requêtes décisionnelles interrogeant des bases de données multidimensionnelles en restituant les données en fonction des préférences utilisateur et de ses contraintes. Pour ce faire, chaque utilisateur (décideur) de l'entrepôt de données est invité à donner ses préférences (son profil) et une contrainte de visualisation.

Cette contrainte peut être, par exemple, les limitations imposées par le dispositif utilisé pour afficher la réponse d'une requête.

Organisation de notre travail :

Dans le premier chapitre, on introduit d'une façon générale la notion de la personnalisation de l'information, et comment le processus de personnalisation est abordé aujourd'hui de façon générale.

Dans le second chapitre, nous avons présenté d'abord les concepts généraux manipulés dans les entrepôts de données. Puis nous avons étudié de manière détaillée la personnalisation des visualisations l'algorithme de (Bellatreche 2006) : personnalisation des requêtes OLAP.

Le dernier chapitre qui est le troisième comportera une présentation de notre prototype qui implémente les algorithmes de personnalisation, ainsi que des exemples illustratifs validant nos propositions.

Chapitre I

Entrepôt de données

I. Introduction :

Toutes les entreprises disposent aujourd'hui d'un volume de données conséquentes jamais analysé. Et si nous vous disions que ces informations peuvent améliorer votre compétitivité ? L'informatique décisionnelle permet d'analyser les données sous tous les angles pour en tirer des conclusions utiles à l'amélioration des performances d'entreprise. Les preneurs de décisions sont alors en mesure de faire des choix constructifs pour leur société, qu'ils n'auraient pas pu prendre en connaissance de cause sans les outils de l'informatique décisionnelle.

I.2. Datawarehouse :

L'entrepôt de données, ou Datawarehouse, est une méthode spécifique de l'informatique décisionnelle.

Un Datawarehouse se définit habituellement comme « un ensemble de données orienté sujet (thématique), intégré, prenant en compte la variante temps, opérant une collecte des données non volatiles et organisées pour le support d'un processus d'aide à la décision ».

L'objectif d'un Datawarehouse est la prise de décision autour des activités majeures de l'entreprise.

La finalité d'un datawarehouse est d'amener la bonne information disséminée dans l'entreprise ou exogène vers l'utilisateur final (le décideur), au niveau de qualité souhaité dans le meilleur délai au format adapté à ses besoins.

De nombreuses définitions ont été proposées, soit académiques, soit par des éditeurs d'outils, de bases de données ou par des constructeurs, cherchant à orienter ces définitions dans un sens mettent en valeur leur produit.

La définition la plus appropriées d'un Datawarehouse est :

« Le Datawarehouse est une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision ».

I.2.1 Caractéristiques des données d'un ED :**I.2.1.1. Orientées sujet et intégrées :**

Signifie que le Datawarehouse est conçu pour enjammer les frontières fonctionnelles et permettre une analyse orientée sujet (produit, coût...etc.), contrairement aux données des systèmes traditionnels qui sont généralement organisées par processus fonctionnel.

Le Datawarehouse doit permettre de prendre en compte les données qui sont dans les systèmes traditionnels (les bases précédentes conçues sur des modèles plus anciens, mais qui contiennent des données capitales pour l'entreprise), afin de fournir au décideur une vue intégrée des données (c'est-à-dire une vue unique).

I.2.1.2. La variante de temps (historisées) :

Concerne la nature historique des données (c'est-à-dire que les données sont positionnées dans le temps), ce qui permet l'étude de leur évolution, en effet un des objectifs du Datawarehouse est de conserver en ligne les données historisées, donc datées : chaque nouvelle insertion de données provenant du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée. Le support de stockage dépend du volume des données, de la fréquence d'accès, du type d'accès.

Les supports les plus couramment utilisés sont les disques, les disques optiques numériques et les cassettes.

I.2.1.3. Non volatiles :

Implique que le datawarehouse contrairement aux systèmes de production, est mis à jour de façon constante, les données stockées au sein de Datawarehouse ne peuvent être ni altérées, ni modifiées, ni supprimées après exploitation (ce qui signifie qu'une donnée entrée dans l'entrepôt l'est pour de bon et n'a pas vocation à être supprimée). Elle devienne une partie prenante de l'historique de l'entreprise.

De ce fait, la donnée sera considérée comme volatile dans un système de production alors qu'elle doit nécessairement être non volatile dans un Datawarehouse.

I.3 Implémentation des entrepôts des données

Il existe différentes stratégies d'implémentation d'ED et d'analyse OLAP, Ces implémentations reprennent toutes les concepts OLAP. Parmi ces implémentations nous citons les plus utilisés :

I.3.1 Systèmes ROLAP (Relational OLAP) :

- Utilisent un SGBD relationnel classique avec des adaptations spécifiques à l'OLAP.
- La base relationnelle de l'entrepôt est organisée pour réagir comme une base OLAP.
- Lents et peu performants mais sans limites de taille.

I.3.2 Systèmes MOLAP (Multidimensionnel OLAP) :

- Utilisent un SGBD multidimensionnel (MOLAP), ils sont l'application physique du concept

OLAP (réellement d'une structure multidimensionnelle).

- Très rapides et performants mais limité au gigaoctet.

I.3.3 Systèmes HOLAP (Hybride OLAP) :

- C'est un compromis :
- Une base MOLAP pour les données souvent consultées (la minorité),

– Une base ROLAP pour les autres (la majorité).

I.3.4 Base DOLAP (Desktop OLAP):

Base OLAP très limitée en taille, hébergée sur le poste client, et très rapide.

I.3.5 Base OOLAP (Object OLAP):

Utilise un SGBD Orienté Object : peu utilisé.

Les faits et les dimensions :

Les deux composants principaux de ces modèles sont les dimensions et les faits.

Les dimensions :

Les dimensions sont les axes sur lesquels on veut faire l'analyse, il peut est avoir une dimensions client, une dimension produite, une dimension géographique (pour faire des analyses sur secteur géographique), etc...

Les faits :

Les faits sont ceux sur quoi va porter l'analyse. Ce sont des tables qui contiennent les informations opérationnelles et relatant la vie d'une entreprise. Par exemple on peut avoir une table des faits pour la vente qui permet d'évaluer le chiffre d'affaire net, quantités et montants commandés et quantités facturées. Aussi une table des faits pour les ressources humaines qui permet d'évaluer le nombre d'exemplaires d'un produit en stock, niveau de remplissage d'un stock. Un fait est ce que nous voulons analyser.

I.4. Construction d'un DataWarehouse :

A partir des faits et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Nous distinguons trois groupes de modèles :

I.4.1. Modèle en étoile :

C'est une manière de relier une **dimension** à un **fait** dans un entrepôt de données. Dans le modèle en étoile on a une table de fait centrale qui est liée par les tables de dimensions dénormalisées. Les **dimensions** ne sont pas liées entre elles. Dans l'exemple ci-dessous nous avons la table des faits Ventes qui permet d'analyser les ventes et le profit sur des axes (Dimensions) Magasin, Temps, et Produit.

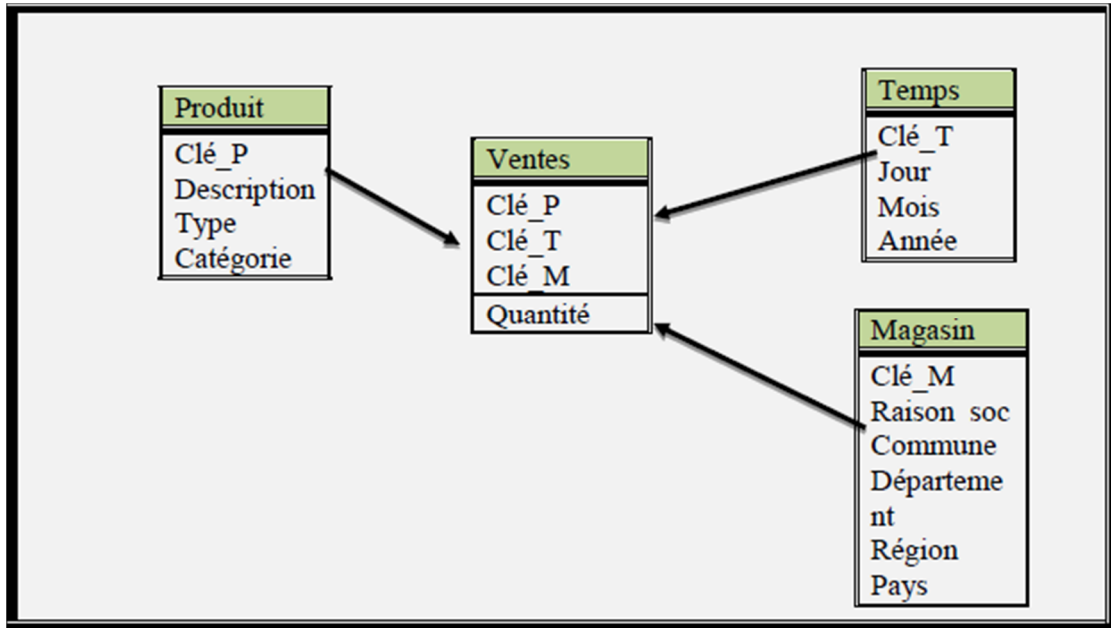


Figure I.1. Exemple d'un modèle en étoile [MAHMOUDI, BABA AHMED2016]

I.4.2. Modèle en flocon de neige :

C'est une manière de relier une dimension à un fait dans un entrepôt de données. C'est le modèle en étoile avec une normalisation des dimensions. Il peut exister des hiérarchies des dimensions pour diviser les tables de dimensions lorsqu'elles sont trop volumineuses.

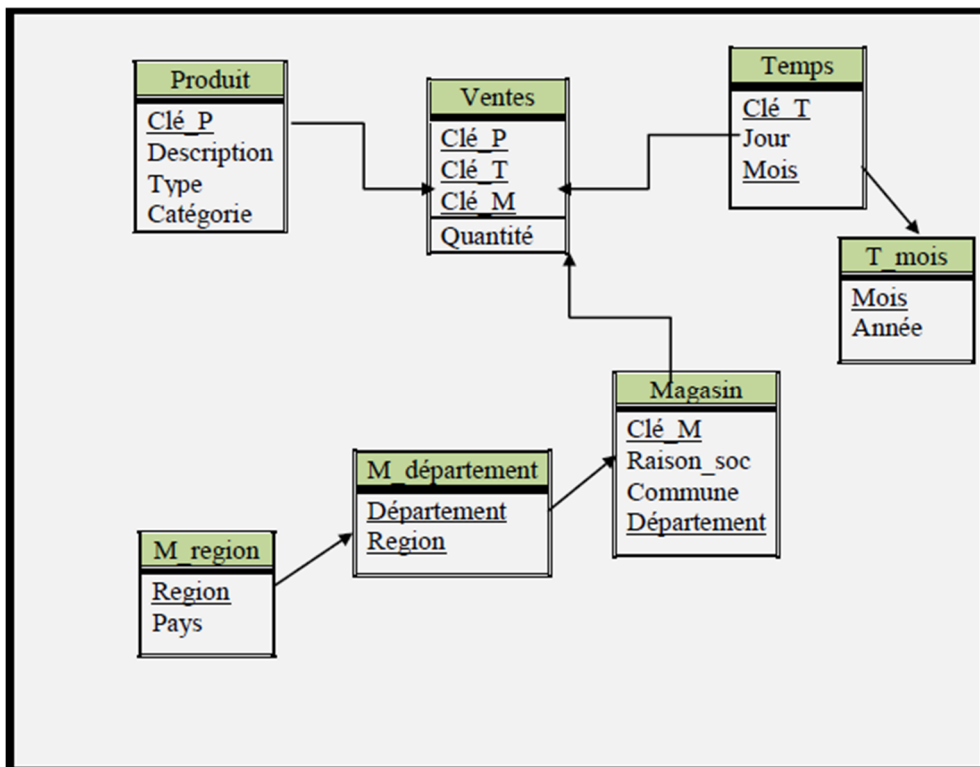


Figure I.2 Exemple d'un modèle en flocon de neige [MAHMOUDI, BABA AHMED2016]

I.4.3 Modèles en constellation :

Ce schéma est une extension du schéma en étoile. Il consiste à fusionner plusieurs schémas en étoile qui utilisent des dimensions communes. Un schéma en constellation comprend donc plusieurs faits reliés à un ensemble de dimensions qui peuvent être partagées.

Ce schéma présente l'avantage de pouvoir corréler les sujets d'analyse tels que la comparaison des montants des locations réalisées dans les différentes agences par rapport aux chiffres d'affaires réalisés par son personnel. En outre, le partage des dimensions par plusieurs faits permet d'éviter de les définir plusieurs fois.

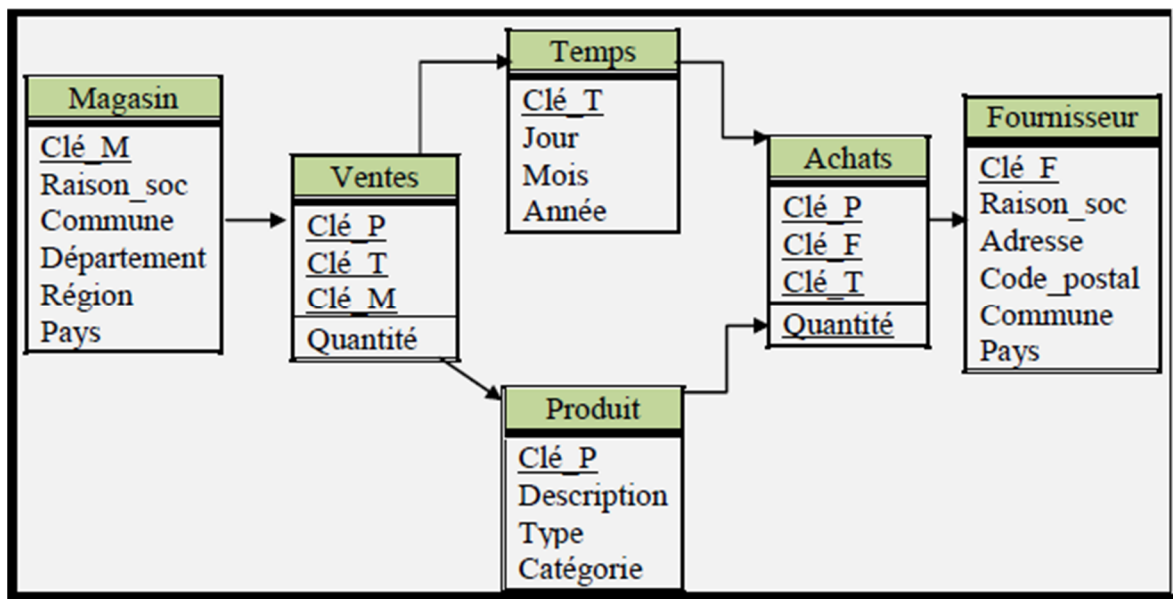


Figure 1.3 Exemple d'un modèle en constellation [MAHMOUDI, BABA AHMED2016]

La relation de faits Ventes partage les dimensions Temps et Produits avec la table Achats. Néanmoins, la dimension Magasin appartient seulement à Ventes. Également, la dimension Fournisseur est liée seulement à la relation Achats.

I.4.4. Différence entre le modèle en étoile et flocon de neige :

Le modèle en flocon de neige est le modèle en étoile avec une normalisation des dimensions. Les dimensions ne sont pas liées entre elles dans le modèle en étoile or au niveau du modèle en flocon de neige il peut exister une hiérarchie entre les dimensions. Au niveau du modèle en étoile les dimensions sont dénormalisées contrairement au modèle en flocon de neige où les dimensions sont normalisées. Le modèle en flocon de neige est complexe avec de nombreuses jointures souvent coûteuses contrairement au modèle en étoile qui évite les jointures.

I.5. Architecture d'un entrepôt de données

La phase essentielle d'un entrepôt de données est l'alimentation des données provenant des bases de production. Afin d'effectuer cette phase l'utilisation d'outils logiciels est nécessaires. On parle d'ETL (Extract, Transform, Load).

L'alimentation d'un entrepôt est organisée selon les phases suivantes :

- **Découverte des données** : Identification des données à importer dans l'entrepôt de donnée qui se trouve dans les systèmes sources.
- **Extraction des données** : Collection des données importantes dans les systèmes de production.
- **Transformation des données** : Rendre les données cohérentes avec la structure d'un entrepôt de données c'est-à-dire les données des systèmes de production doivent être agrégées ou calculées avant leur chargement.
- **Chargement des données** : Insertion des données au sein de l'entrepôt de données.

I.6. Modélisation et interrogation des bases de données multidimensionnelles

I.6.1 Modélisation des données OLAP

Issue originellement du monde industriel, la modélisation multidimensionnelle vise à organiser les données de telle sorte que les applications OLAP soient performantes et efficaces [Kimball, 1996]. Deux approches existent.

Modélisation en cube

Les premiers modèles proposés reposent directement sur la métaphore de cube [Agrawal et al., 1997 ; Datta et Thomas, 1999 ; Gray et al., 1996 ; Gyssens et Lakshmanan, 1997 ; Li et Wang, 1996]. Cette approche supporte une séparation entre les éléments de structure et les valeurs [Torlone, 2003] : modélisation des axes de l'analyse peu expressive (difficulté à représenter l'organisation hiérarchique des données). Elle se heurte aussi à la représentation des espaces multidimensionnels constitués de plus de trois axes d'analyse. Elle s'avère enfin limitée lorsqu'il s'agit de représenter des constellations de faits et de dimensions potentiellement partagées.

Modélisation multidimensionnelle

Face à ces limites, d'autres approches sont apparues par développement de modèles multidimensionnels [Cabibbo et Torlone, 1998, 2000 ; Golfarelli, et al., 1998 ; Ravat et al., 2008 ; Schneider, 2003] qui distinguent les éléments de structuration des valeurs tout en maintenant un nombre limité de concepts : fait, dimension et hiérarchie. Ces modèles restent spécialisés dans la représentation de données multidimensionnelles et ne reposent pas sur des

notations standards [Torlone, 2003]. Les concepts et les formalismes associés à la modélisation multidimensionnelle souffrent de l'absence d'un consensus standardisé [Rizzi, et al., 2006].

I.6.2 Manipulations des données OLAP

La communauté des bases de données multidimensionnelles s'est intéressée à la définition d'un langage d'interrogation des données OLAP. De nombreux opérateurs et langages ont été proposés. Ces propositions visent à répondre aux besoins d'analyse OLAP des décideurs en définissant des opérateurs interactifs facilitant la navigation au sein des données multidimensionnelles [Abelló et al., 2003]. Différentes études comparatives ont été réalisées dans [Abelló et al., 2006 ; Rafanelli, 2003 ; Ravat et al., 2007b, 2008 ; Torlone, 2003]. Les premiers travaux sur les manipulations OLAP ont étendu les opérateurs de l'algèbre relationnelle pour le modèle en cube [Agrawal et al., 1997 ; Datta et Thomas, 1999 ; Gray et al., 1996 ; Gyssen et Lakshmanan, 1997 ; Li et Wang, 1996].

Pour mieux prendre en compte les structures multidimensionnelles, d'autres travaux ont proposé des opérateurs pour spécifier et manipuler un cube [Abelló et al., 2003 ; Cabibbo et Torlone, 1997 ; Franconi et Kamble, 2004 ; Pedersen et al., 2001]. La majorité des travaux reposent sur une structure de visualisation simplifiée dans laquelle le concept de hiérarchie n'est pas exploité. Certaines des propositions [Gyssen et Lakshmanan, 1997] ne supportent qu'un niveau de paramètre en entête des lignes et des colonnes de la structure de visualisation. Certains travaux [Sarawagi, 1999, 2000] ; [Sathe and Sarawagi, 2001] ont défini des opérateurs avancés pour l'exploration d'un cube de données en se basant sur des techniques de la fouille des données. Ces opérateurs ne sont pas largement utilisés dans les outils commerciaux et au niveau de la recherche à l'image des opérations classiques telles que Drilldown et Roll-up.

En pratique, certains serveurs OLAP (Mondrian1, Oracle2) utilisent des langages qui ne traduisent pas l'aspect navigationnel des analyses OLAP. Nous pouvons citer le langage MDX qui est doté d'une syntaxe de type SQL, ainsi que l'extension de SQL par les opérateurs Roll-up et Cube [Gray et al., 1996] qui a été intégrée à Oracle 11g.

Analyse des données OLAP

Contrairement à l'interrogation d'une base de données, l'interrogation d'une BDM est une succession d'opérations d'exploration. Elle est souvent désignée par *analyse multidimensionnelle* relativement au type des données manipulées, ou par *analyse OLAP* en faisant référence à la technologie utilisée. Bien qu'elle représente le cœur d'un système

OLAP, l'analyse OLAP souffre d'un manque de formalisation. Différents travaux la qualifient d'une *navigation* [Dittrich et al., 2005 ; Kumar et al., 2006 ; Sarawagi et al., 1998 ; Thalhammer et al., 2001], un concept emprunté du web. Ainsi, une analyse OLAP se déroule comme suit :

1. Sélection d'une première requête
2. Navigation des données par
 - Forage vers le bas
 - Forage vers le haut
 - Restriction des données
 - Etc.

Différentes visions de l'intuition de la navigation sont adoptées :

- Une analyse est basée sur l'objectif [Dittrich et al., 2005] ; [Thalhammer et al.,2001].

L'utilisateur cherchant à répondre à une question ou à expliquer un phénomène commence par définir une première requête qui traduit son besoin. Puis, il modifie la requête suite à l'observation du résultat. Une requête est généralement la transformation de la requête précédente.

- Une analyse est pilotée par la découverte [Kumar et al., 2006] ; [Sarawagi et al., 1998].

L'objectif de l'analyse est de détecter les anomalies dans les valeurs de la BDM.

L'analyse démarre généralement au niveau hiérarchique le plus haut d'une dimension. Des opérateurs spécifiques sont utilisés en plus des manipulations OLAP classiques afin de découvrir les anomalies cachées.

D'après [Kumar et al., 2006], une analyse OLAP est un chemin au sein du treillis de cuboïdes¹. La transition d'un nœud à un autre est assurée par un forage vers le bas suivant la dimension du nœud précédent ou par l'ajout d'une nouvelle dimension. Des manipulations OLAP comme la rotation ou les restrictions ne sont pas prises en compte.

D'une manière plus générale, [Dittrich et al., 2005] définissent une analyse par un graphe.

Chaque nœud traduit un état de l'analyse et est représenté par les deux dimensions affichées et les conditions de restriction courantes. Un arc traduit la requête utilisateur appliquée. Ainsi, la représentation d'un état de l'analyse n'intègre pas les structures multidimensionnelles fait et mesures et se limite à deux dimensions.

Par ailleurs dans [Thalhammer et al. 2001], une analyse est vue comme un graphe de cubes de données, où le nœud racine est le cube le plus général (dont les données sont

agrégées selon le niveau le plus général de chaque dimension). L'état d'analyse suivant représente un cube où les données sont agrégées selon un niveau de granularité plus détaillé.

Il faut noter que [Sathe et Sarawagi ; 2001] adoptent une vision contraire des transitions d'une analyse OLAP. En effet, le passage d'un état d'analyse (un cube de données) à un autre est effectué par un forage vers le haut.

Conclusion

Après avoir présenté légèrement le concept de l'entrepôt de données (base de données multidimensionnelle) et ses enjeux, le chapitre suivant va éclairer la notion de la personnalisation et son application dans ce domaine de base de données multidimensionnelle.

Chapitre II

La personnalisation des cubes OLAP

II.1. Introduction :

Avant d'aborder la personnalisation de l'information, il est indispensable de découvrir les différents concepts utilisés tel que la préférence et le profil utilisateur, ensuite, nous allons présenter la personnalisation de l'information d'une manière générale.

Si la personnalisation n'est pas une nouvelle idée dans les domaines précédemment évoqués, elle constitue un axe de recherche émergent dans le domaine des entrepôts de données. L'intérêt de cet axe de recherche est d'élaborer des systèmes d'analyse de données dédiés au soutien et à l'amélioration des processus décisionnels des entreprises.

Les utilisateurs des systèmes OLAP (On-Line Analytical Processing) formulent des requêtes décisionnelles pour répondre à des besoins d'analyse spécifiques pour l'aide à la décision.

II.2. La personnalisation :

L'idée de développer des mécanismes permettant de personnaliser un système informatique n'est pas une idée nouvelle selon [F. Ravat, O. Teste, G. Zurfluh, 2010]. Notamment, nous pouvons citer les travaux du domaine de la recherche d'information (RI) [Korfhage, 1997] [Bouzeghoub, Kostadinov, 2005]. La personnalisation en RI consiste généralement à définir des profils utilisateurs (ensembles plus ou moins structurés de caractéristiques) qui sont exploités aux différentes étapes du processus de RI : indexation, recherche, ...

La personnalisation de BD décisionnelles a fait l'objet d'une première proposition [Bellatreche et al., 2005] visant à fournir aux décideurs la visualisation la plus adaptée à ces requêtes. Pour ce faire, les auteurs proposent des contraintes de visualisation précisant la structure du cube de données résultat et des préférences utilisateurs définies à l'aide du concept de pré-ordre total [Koutriba, Ionnidis, 2004] portant sur un attribut d'une dimension. Cette proposition se limite à des cubes de données dont les dimensions contiennent un seul attribut et à l'opération de sélection des données pour calculer le cube résultat.

Dans le web, la personnalisation implique la création automatique de pages web qui s'adaptent aux intérêts de l'internaute. La personnalisation s'organise selon des données implicites (fureteurs, provenances, parcours observé dans un site, etc.) ou explicites (informations fournies par l'internaute sur ses préférences). L'expression « sur mesure » est parfois utilisée lorsqu'il est question de données explicites.

Dans les médias écrits, la personnalisation s'effectue à partir de bases de données d'informations et utilise parfois la technique de fusion de fichiers dans le marketing direct.

Non seulement le document écrit s'adresse-t-il directement par son nom au lecteur concerné, mais le contenu (les articles ou même la publicité) est plus particulièrement ciblé selon les caractéristiques de ce dernier.

Le même principe de personnalisation a été repris du reste par l'industrie de l'objet promotionnel (tasse, t-shirt, porte-clés, etc.), du jeu vidéo (par exemple, The Sims) ou même de la littérature pour enfants qui offre maintenant des livres personnalisés.

Des chercheurs associés à des industriels font avancer la qualité de la personnalisation des artefacts en nourrissant l'espoir d'arriver un jour à produire des objets complexes standards (donc bon marché pour une grande distribution) et aussi extrêmement personnalisables, notamment le « digital human laboratory » à Tokyo et nombres d'études cognitives et sociologiques [Jameson, 1999].

Nous nous intéressons à la personnalisation de l'information dans le contexte de l'interrogation des entrepôts de données par requêtes OLAP. (On-Line Analytical Processing) permettent l'analyse interactive d'un gros volume de données multidimensionnelles à l'aide d'opérateurs spécifiques de

- (a) consolidation et agrégation pour résumer l'information et de
- (b) présentation et structuration pour naviguer et explorer l'information.

Ces données sont organisées sous forme de cubes de données et proviennent d'un entrepôt de données.

Le processus de personnalisation est décrit par [Rosenberg, 2001] comme « **une technologie cognitive** » dans la mesure où il s'appuie sur un apprentissage, par la machine, de ce que veut l'utilisateur. Cet apprentissage est basé sur une observation de son comportement au cours des sessions. Le système a, par ailleurs, la capacité de s'adapter au cours du temps. Pour Kostadinov [Kostadinov, 2003], la personnalisation de l'information se définit par un ensemble de préférences individuelles, par des ordonnancements de critères ou par des règles sémantiques spécifiques à chaque utilisateur ou communauté d'utilisateurs. Ces modes de spécification servent à décrire le centre d'intérêt de l'utilisateur, le niveau de qualité des données qu'il désire ou des modalités de présentation de ces données.

La personnalisation consiste à utiliser les données comportementales recueillies sur l'utilisateur pour cibler ses intérêts. Il s'agit donc de fournir à l'utilisateur un contenu créé en amont en fonction de son profil. Ainsi, la personnalisation se retrouve dans tant de travaux de recherches traitant pourtant des sujets très différents. Suite à cet engouement, nous constatons une difficulté croissante à en saisir la signification exacte ce qui nous incite à clarifier ce vaste sujet. Ainsi, nous nous intéressons à définir ce que c'est la personnalisation après une revue

des définitions qui existent dans la littérature. Par ailleurs, l'étude de la littérature a montré une certaine ambiguïté quant à la définition de la personnalisation et de la recommandation. Alors, il est temps de faire la distinction entre la recommandation et la personnalisation pour mieux comprendre les problèmes posés et les solutions proposées.

Le terme personnalisation est utilisé pour expliquer comment recevoir à partir d'une grande quantité d'informations seulement la partie qui intéresse un individu ou un groupe d'individus. Pour connaître ce qui intéresse l'utilisateur, il faut connaître son profil (ses intérêts, ses préférences ou même ses contraintes, ses comportements, etc.). Le terme personnalisation est alors utilisé pour la prise en compte du profil utilisateur.

De ce fait, aujourd'hui, on utilise les systèmes personnalisés qui tiennent compte des besoins, des intérêts et des préférences individuels des utilisateurs. Les utilisateurs s'attendent qu'on exploite l'information qu'on a sur eux et de continuer à apprendre à partir de leurs relations avec le système (comportement). Ils ne veulent pas perdre leur temps avec des informations inappropriées ou non pertinentes.

II.3. Profile :

Un profil utilisateur est une collection d'informations sur l'utilisateur. Cette collection peut être vue comme un ensemble de caractéristiques avec des valeurs associées contenant par exemple les préférences de l'utilisateur. Il couvre des aspects larges tels que son environnement cognitif, social et professionnel qui déterminent ses intentions au cours d'une session de recherche [C. Cool et A. Spink].

II.3.1. Le contenu du profil :

Dans un processus d'accès à des informations structurées comme les bases de données ou les entrepôts de données, un profil utilisateur peut être composé de :

- (1) préférences sur le contenu de la réponse à la requête,
- (2) préférences sur la présentation de la réponse, et/ou
- (3) des conditions d'exploitation (les conditions matérielles et/ou les contraintes utilisateur).

II.3.2. Modélisation du profil utilisateur :

L'introduction de la dimension utilisateur dans un processus d'accès à l'information, mérite, voire nécessite une réflexion sur la modélisation de l'entité **utilisateur**.

Différents travaux ont considéré que la modélisation de l'utilisateur ; dans le cadre d'un système d'accès personnalisé à l'information, est un processus caractérisé par trois phases :

1. La première porte sur la définition d'une **représentation** des unités d'information caractérisant l'utilisateur du système. Elle correspond à la définition de la structure du profil utilisateur présentée dans la section précédente,

2. La deuxième phase est liée à l'**instanciation** de cette représentation au cours d'une activité d'accès à l'information pour un utilisateur particulier. Elle regroupe des techniques d'acquisition des données utilisateur ainsi que des approches de construction pour agencer les informations collectées selon la structure représentative définie lors de l'étape précédente,

3. Enfin, la troisième phase concerne l'**évolution** du profil au cours du temps. Elle nécessite la mise en place de stratégies de mise à jour du contenu informationnel du profil.

Dans ce qui suit nous allons exposer les différentes approches utilisées pour la mise en œuvre de chaque étape.

II.3.3. 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 [TB, 2008].

La première phase consiste à collecter les informations pertinentes pour instancier le profil de l'utilisateur. Ce processus peut collecter ces informations soit directement à partir la machine de l'utilisateur (côté client) ou à partir de l'application (côté serveur). Ce processus d'acquisition peut être explicite et/ou implicite :

- i. l'approche explicite consiste à obtenir les informations directement de l'utilisateur,
- ii. l'approche implicite, largement motivée par les travaux actuels dans le domaine, implique l'exploitation des données de comportement de l'utilisateur pour inférer son profil.

II.3.4. Exploitation de profil :

Le profil peut être utilisé soit pour une transformation du système, soit pour une recommandation faite par le système.

Le tableau de la figure 1.1 permet de caractériser chacun des travaux présentés en les positionnant par rapport aux critères définis précédemment.

Références	Profil	Collecte information	Type perso	Objet perso	Perso évolutive		Type approche	Exploitation profil
					Mode mise à jour	Extensibilité		
[Bellatreche et al., 2005]	Oui	Explicite	Cognitif	Présentation	Structure	Non	Quantitative	Transformation
[Espil et Vaisman, 2001]	Non	Explicite	Cognitif	Contenu	Valeur	Non	-	Transformation
[Favre et al., 2007]	Non	Explicite	Cognitif	Contenu	Structure	Non	-	Transformation
[Giacometti et al., 2008]	Oui	Implicite	Social	Contenu	Structure	Non	-	Recommandation
[Jerbi et al., 2008-2009]	Oui	Explicite	Cognitif	Interaction	Structure	Non	Quantitative	Transformation
[Ravat et al., 2008]	Oui	Explicite	Cognitif	Interaction	Structure	Non	Quantitative	Transformation
[Thalhammer et al., 2001]	Non	Explicite	Cognitif	Interaction	Valeur	Non	-	Transformation

Tab II.1 Comparaison des travaux dans le domaine des entrepôts [Benttayeb,2009]

Nous remarquons que les techniques de collecte d'information implicite n'ont été exploitées que dans un des travaux, de façon très récente au niveau des entrepôts de données.

Ceci implique que dans la plupart des cas, l'utilisateur est amené à paramétrer l'entrepôt de données en spécifiant ses préférences ; cette approche explicite peut s'avérer parfois ardue, en particulier lorsque les préférences doivent être quantifiées. À ce jour, l'aspect social de la personnalisation dans les entrepôts de données n'a été également exploité que par ce seul travail. L'approche proposée est la plupart du temps une approche cognitive où l'utilisateur est considéré de manière individuelle, sans tirer profit de l'expérience des autres utilisateurs.

Par ailleurs, notons que les travaux se proposent souvent d'exploiter le profil pour transformer le système, l'aspect recommandation restant presque inexploré dans le contexte des entrepôts de données, à l'exception d'un des récents travaux. Enfin, nous notons que les travaux proposés jusque-là n'ont pas abordé la possibilité d'extensibilité du profil avec des critères non prédéfinis. Ces remarques servent de support à la section suivante dans laquelle nous traçons différentes perspectives de recherche [F. Bentayeb et al].

II.3.5. Évolution du profil utilisateur :

L'évolution des profils désigne leur adaptation à la variation des préférences des utilisateurs et au changement de leur environnement d'interaction, et par conséquent, de leurs besoins en information au cours du temps.

La technique utilisée pour faire évoluer le profil est inhérente à celle de la construction.

Néanmoins, adapter le profil utilisateur implique des changements au niveau de contenu qui conduisent éventuellement à la suppression de quelques dimensions (ou membres) ou à l'émergence d'autres dimensions (ou membres) avec l'augmentation ou la diminution de l'intérêt.

Si l'acquisition du profil est implicite et afin de faire face aux changements d'intérêts dynamiques, le système doit non seulement dépister des intérêts dérivant de l'utilisateur afin d'identifier de nouveaux intérêts émergents, mais également inférer à chaque connexion les caractéristiques de son environnement à savoir la taille de son écran, la capacité de sa mémoire, etc.

II.4. Les 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 notre contexte 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.

II.4.1 Modélisation de préférences

De nombreuses recherches sur la modélisation des préférences ont été menées par les communautés de recherche d'information et de base de données. Toutefois, peu de travaux ont étudié les préférences dans un environnement OLAP.

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

II.4.1.1 Niveau des préférences

Au niveau des bases de données, les préférences sont généralement exprimées sur les n-uplets.

Ces modèles sont inadaptés aux BDM où les préférences doivent spécifier le chemin des données que l'utilisateur désire analyser [Golfarelli et al., 2011]. Ainsi, les modèles des préférences OLAP ont porté sur le schéma ainsi que les valeurs de la BDM.

Les préférences sur le schéma de la BDM sont définies sur deux niveaux :

- Les préférences sur les dimensions [Bellatreche et al., 2006] décrivent l'ensemble des dimensions pertinentes pour l'analyse d'un fait.
- Les préférences sur les paramètres [Golfarelli et Rizzi, 2009, Ravat et Teste, 2008] spécifient les niveaux de granularité préférés au long d'une dimension.

Les modèles proposés dans [Bellatreche et al., 2006 ; Golfarelli et Rizzi, 2009 ; Xin et Han, 2008] supportent des préférences sur des valeurs de mesures élémentaires (par exemple, $Nb_publis > 2$) ou agrégées (par exemple, $SUM(Nb_publis) > 10$). Il peut s'agir également de préférences sur les valeurs des attributs de dimension [Bellatreche et al., 2006 ; Golfarelli et Rizzi, 2009], par exemple $Année > 2010$.

La sémantique des préférences définies sur le même élément du schéma varie d'une approche à une autre. Considérons par exemple une préférence de l'utilisateur sur l'attribut « Ville ».

Selon [Ravat et Teste 2008], cette préférence qualifie l'importance de l'attribut ville par rapport aux autres attributs de la dimension géographique. Cette préférence spécifie selon [Golfarelli et Rizzi 2009] l'importance des valeurs de mesures agrégées par ville par rapport aux valeurs des mêmes mesures lorsqu'elles sont agrégées selon d'autres attributs.

Il faut noter que dans le modèle de [Golfarelli et Rizzi 2009], une préférence sur les valeurs d'un attribut est propagée suivant la hiérarchie. Par exemple, une préférence de la ville de Toulouse signifie que l'utilisateur préfère, en plus des données de cette ville, les données de la

France (pays de cette ville) et des différents départements toulousains.

II.4.1.2 Formulation de préférences

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 [Agrawal et Wimmers, 2000 ; Koutrika et Ioannidis, 2004, 2005] ou qualitative [Kießling, 2002 ; Chomicki, 2003].

Selon les approches **qualitatives**, les préférences sur un ensemble sont formulées par des relations d'ordre entre ses éléments.

Soit E un ensemble et une relation binaire sur cet ensemble notée « \leq », cette relation est une relation d'ordre si pour tous x, y et z éléments de E :

- $x \leq x$ (réflexivité)
- $(x \leq y \text{ et } y \leq x) \Rightarrow x = y$ (antisymétrie)
- $(x \leq y \text{ et } y \leq z) \Rightarrow x \leq z$ (transitivité)

La plupart des travaux proposent la définition d'un ordre strict partiel sur les attributs de dimensions [Bellatreche et al., 2006] ou sur les n-uplets de la BDM [Golfarelli et Rizzi, 2009]. Un ordre partiel strict sur un ensemble E est une relation binaire irreflexive, transitive et asymétrique [Chomicki, 2003] sur les éléments de E.

Les approches **quantitatives** permettent d'exprimer les préférences d'une façon indirecte par l'utilisation de fonctions de score qui associent un nombre réel à chaque n-uplet du résultat de la requête. Dans ce cas, un n-uplet t_1 est préféré à un n-uplet t_2 si son score est supérieur à celui de t_2 .

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 »).

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 ».

Quant à l'acquisition des préférences, il est plus facile de définir des relations de préférences entre des couples d'éléments que de spécifier des scores. Cependant, l'emploi de telles préférences pour ordonner des valeurs agrégées est plus compliqué que l'évaluation d'une fonction de score.

Il faut noter qu'une méthode typique pour inférer implicitement les préférences qualitatives est de calculer à partir du log le nombre d'occurrences des éléments sélectionnés au passé [Holland et al., 2003]. Ces occurrences, qui peuvent être assimilées à des scores de préférences, sont ensuite utilisées pour déduire les relations d'ordre. Récemment [Aligon et al. 2011] ont proposé une approche d'extraction de règles d'association à partir de l'historique des requêtes MDX, qui sont ensuite traduites en préférences.

Expression des préférences

Les préférences qui expriment des besoins spécifiques de l'utilisateur à long terme sont stockées.

D'autres préférences qui représentent des besoins à court terme sont exprimées explicitement au moment de la requête suivant différentes manières.

Les préférences quantitatives sont formulées dans [Ravat et Teste, 2008] par des règles ECA (Evènement-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'utilisateur 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 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 (\otimes) pour indiquer le même degré d'importance entre deux préférences et Priorisation (\triangleright) pour définir un ordre de priorité entre préférences. Ces différents opérateurs sont ensuite intégrés dans une requête MDX à l'aide de clause PREFERRING.

Une deuxième méthode a été proposée en bases de données où les préférences sont sélectionnées si leurs contextes appartiennent avec le contexte de la requête.

– Si une préférence P est rattachée à un contexte interne C, la sélection de P dépend d'une confrontation entre le contexte C et les tuples de la base de données [Stefanidis et al., 2009] ou les attributs de la requête [Agrawal et al. 2006].

– Dans le cas de contexte externe, le contexte courant de l'utilisateur $CC(U)$ est d'abord détecté. Une préférence est sélectionnée si son contexte apparie avec $CC(U)$. Selon [Bunningen et al., 2006 ; Stefanidis et al., 2007], l'appariement de contexte revient à déterminer les contextes qui sont égaux ou plus généraux que $CC(U)$. Par exemple, si l'utilisateur est localisé à Toulouse au moment de la requête, les préférences qui sont associées à la localisation France sont sélectionnées.

Sélection des préférences

Une étape de sélection des préférences est nécessaire pour déterminer celles qui seront utilisées dans le processus de personnalisation.

Une première méthode est centrée sur l'applicabilité de la préférence. Une préférence P est applicable à une requête Q si l'exécution de Q combinée conjonctivement avec P ne renvoie pas un résultat vide [Golfarelli et al., 2011] ou si le score de P est supérieur à un seuil exprimé dans Q [Ravat et Teste, 2008]. Nous pouvons aussi citer l'approche de [Cuppens et Demolombe, 1991] où une préférence est applicable à une requête Q si son prédicat est impliqué logiquement par un prédicat de Q .

Une deuxième méthode a été proposée en bases de données où les préférences sont sélectionnées si leurs contextes apparient avec le contexte de la requête.

– Si une préférence P est rattachée à un contexte interne C , la sélection de P dépend d'une confrontation entre le contexte C et les tuples de la base de données [Stefanidis et al., 2009] ou les attributs de la requête [Agrawal et al. 2006].

Impact des préférences

Les préférences OLAP ont des impacts différents sur la restitution des données. Selon une première approche, les préférences sont intégrées dans la requête en tant que contraintes optionnelles. Les n-uplets qui satisfont les préférences autant que possible sont restitués, même si aucun n-uplet ne satisfait toutes les préférences [Golfarelli et Rizzi, 2009 ; Golfarelli et al, 2011]. Dans une deuxième approche, les préférences sont intégrées dans la requête comme des contraintes fortes. Par exemple, les préférences sur les valeurs d'un attribut de dimension sont intégrées en tant que conditions de sélection au sein de la requête initiale [Bellatreche et al., 2006]. Les préférences sur les attributs de dimension sont intégrées au niveau de la clause Group-By d'une requête SQL [Ravat et Teste, 2008].

II.4.2. Synthèse sur la modélisation et l'exploitation des préférences OLAP

En résumé, les modèles de préférences OLAP varient selon le niveau de définition (schéma ou valeurs de la BDM) et la méthode de formulation (quantitative ou qualitative). Les préférences exprimées implicitement ou explicitement sont utilisées en tant que contraintes

fortes ou optionnelles. Le Tableau II.2 donne une vision synthétique de ces approches. A partir de ce tableau, nous pouvons constater des insuffisances au niveau de la modélisation et de l’usage des préférences OLAP.

En termes de modélisation, nous avons relevé le manque de modèles de préférences portant sur toutes les structures multidimensionnelles de la BDM. De plus, nous avons constaté l’absence de modèle de préférence OLAP contextuelle. La prise en compte du contexte permettrait de décrire plus précisément les préférences de l’usager [Garrigós et al., 2009]. Il faut noter que malgré de nombreuses propositions de modèles de préférences contextuels en bases de données, aucune n’est parfaitement adaptée au domaine OLAP. Par exemple, dans [Agrawal et al. 2006] la modélisation des contextes internes sous forme de vecteur composé des attributs de la requête et de leurs valeurs ne prend pas en compte l’organisation multidimensionnelle et hiérarchique des structures et valeurs d’une BDM.

En ce qui concerne l’usage des préférences, la prise en compte du contexte courant permettrait de restreindre l’ensemble des préférences sélectionnées.

				Bellatreche et al., 2006	Golfarelli et al., 2010 ; Aligon et al., 2011	Ravat et Teste, 2008	Xin et Han, 2008
Modélisation	Niveau	Schéma	Mesures			×	
			Dimensions	×			
		Paramètres		×	×		
	Valeurs	×	×		×		
	Forme				Qual.	Qual.	Quant.
Contextualisation				Non	Non	Non	Non
Exploitation	Expression	explicite			×	×	×
		implicite			×		
	Sélection		Ordre	BMO ⁵	Score>seuil		
	Intégration		Condition de sélection	Clause Preferring	Clause Group-By	Clause Order By	
	Impact		forte	optionnelle	forte	optionnelle	

Tab II.2.Récapitulatif des travaux sur la modélisation et l’usage des préférences OLAP

II.5. La recommandation :

L’objectif d’un système de recommandations est d’aider les utilisateurs à faire leurs choix dans un domaine où ils disposent de peu d’informations pour trier et évaluer les alternatives possibles. Un système de recommandations peut être décomposé en trois entités

de base (Figure II.1) le groupe d'agents *producteurs* de données brutes, le module de calcul de recommandations et le groupe de consommateurs des recommandations [Trousse, 2001].

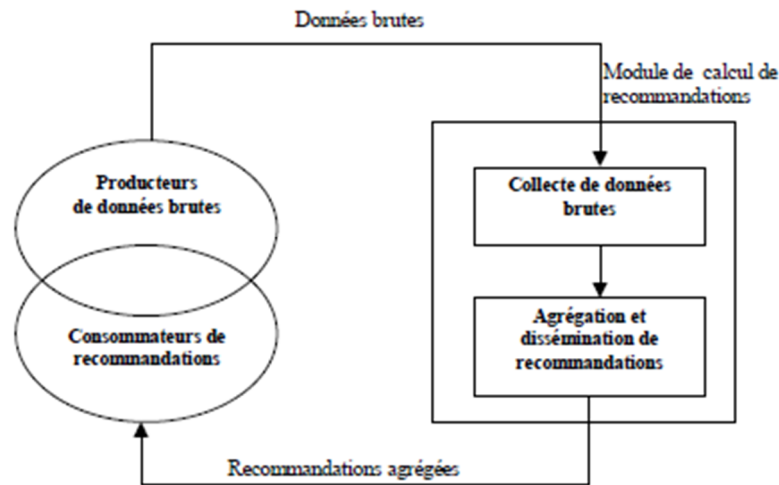


Figure II.1 Architecture générale d'un système de recommandation [Trousse, 2001]

GroupLens [Fink et Kobsa, 2000] est un exemple de système de recommandation. Il offre des services de recommandation sur la base de la gestion de groupes d'utilisateurs ayant les mêmes intérêts. Il propose trois types de recommandations : personnelle, anonyme et consultation rapide.

Une autre forme de recommandation consiste à formuler des propositions en fonction du profil de l'utilisateur et du contenu visualisé par ce dernier. C'est notamment cette technique qui est utilisée par **Amazon.com** [Amazon, 2004] pour conseiller ses clients : si le visiteur s'intéresse à un livre, le site lui propose les ouvrages ayant également plu aux acheteurs de ce même livre. Pour Amazon, ce système vise à remplacer le travail que ferait un vendeur dans une librairie traditionnelle.

Concluons ce point en remarquant que les recommandations n'appartiennent pas qu'au domaine de la personnalisation. Certaines pratiques de marketing en délivrent de manière « aveugle ». Bazsalicza et Naim [Bazsalicza et Naim, 2001] démontrent toutefois qu'un système de recommandations personnalisées fournit de meilleurs résultats qu'un système aléatoire.

II.6. Personnalisation des systèmes OLAP

Selon [Garrigós et al., 2009], la personnalisation est un processus d'adaptation du système OLAP à certaines informations relatives à l'utilisateur telles que ses objectifs, ses caractéristiques, son comportement et son contexte. Ainsi, 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. dans ce qui suis on va se baser sur le travail

de[Bellatrech,2006] pour faire évoquer le paradigme de personnalisation pour les entrepôts de données» Nous présentons dans la suite les travaux de personnalisation par niveau : la personnalisation du schéma multidimensionnel, la personnalisation de l'interrogation des données (couvrant les trois services de personnalisation de l'accès à l'information), la personnalisation de la visualisation des données et la personnalisation de la prise de décision.

II.6.1 Personnalisation du schéma OLAP

Après la conception du schéma de la BDM, les besoins de l'utilisateur peuvent évoluer dans le temps [Favre et al., 2007 ; Hurtado et al., 1999 ; Garrigós et al. 2009]. 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 ?

Deux approches ont été proposées : l'évolution du schéma et la gestion de vues du schéma.

La première approche est centrée sur la mise à jour des hiérarchies des dimensions. Deux niveaux de mise à jour sont distingués :

- 1) l'ajout ou la suppression d'un niveau de granularité et
- 2) l'ajout ou la suppression d'une instance de dimension.

Dans [Blaschka et al., 1999 ; Hurtado et al., 1999], des opérateurs d'évolution sont définis pour mettre à jour les hiérarchies des dimensions, tandis que plus récemment, [Favre et al. 2007] ont proposé une approche à base de règles. Les travaux d'évolution du schéma engendrent la mise à jour du schéma physique des bases de données multidimensionnelles.

Une étape de maintenance est effectuée en conséquence afin de propager les mises à jour au niveau des agrégats. Ceci implique des mises à jour, qui peuvent s'avérer lourdes et complexes, des processus d'alimentation et de rafraîchissement de la BDM.

La deuxième approche de personnalisation du schéma permet de générer une vue personnalisée du schéma en fonction du profil de l'utilisateur courant [Garrigós et al., 2009].

Il s'agit de cacher des éléments de la constellation à l'utilisateur. La personnalisation du schéma est répartie en deux étapes. Les profils utilisateurs sont définis durant la phase de conception. Ils sont ensuite combinés à des règles ECA afin de générer des vues personnalisées du schéma au moment de sa consultation ou de son interrogation. Cette approche n'induit pas de modification du schéma physique de la base. Son inconvénient majeur est l'impossibilité d'ajout d'un élément de la constellation pour répondre à un nouveau besoin de l'utilisateur.

II.6.2. 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'utilisateur 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'utilisateur
- Des travaux visant à assister l'utilisateur dans la définition des requêtes, appelés communément des travaux de recommandation.

II.6.3. Personnalisation de requêtes OLAP :

La personnalisation de requête est basée sur le constat que des utilisateurs peuvent juger des résultats différents pertinents lors de la requête des données (Pitkow et al. 2002). L'objectif de cette approche est de restituer les données les plus pertinentes pour chaque utilisateur.

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.

Deux approches de personnalisation des requêtes OLAP ont été proposées :

- Expansion de la requête afin de mieux traduire les besoins de l'utilisateur.
- Tri du résultat en fonction de préférences afin de retourner les meilleurs objets. On parle dans ce cas de requêtes de tri ou de préférences.

Expansion de requêtes

Les méthodes d'expansion des requêtes supposent l'existence d'un profil de l'utilisateur. Des éléments du profil sont utilisés pour étendre la requête de l'utilisateur. Ceci se traduit par l'ajout de conditions de sélection [Bellatreche et al., 2005, 2006], d'attributs d'agrégation [Ravat et al., 2007], ... La version étendue de la requête est ensuite exécutée en substitution de la requête initiale afin de générer un résultat adapté à l'utilisateur.

Le processus de personnalisation de requêtes par expansion se déroule en deux étapes :

- Sélection d'un sous-ensemble des éléments du profil ou des préférences définies en ligne qui sera utilisé pour personnaliser la requête. Notons que plusieurs travaux effectués dans la communauté des bases de données [Agrawal et al., 2006 ;

Bunningen et al., 2006 ; Holland et Kießling, 2004 ; Stefanidis et al., 2007] considèrent cette étape comme un processus de résolution de contexte. Il s'agit du cas de préférences contextuelles où celles qui sont rattachées au contexte de la requête sont sélectionnées.

– Intégration des éléments sélectionnés dans la requête initiale.

Requêtes de tri

La catégorie des requêtes de tri la plus étudiée est celles des *requêtes Top-K* qui permettent de renvoyer seulement les k meilleurs objets du résultat [Li et al., 2007a ; Li et al., 2007b ; Loh et al., 2002 ; Xin et Han, 2008]. Les critères de tri sont définis en tant que conditions faibles (soft) afin de rendre la sélection des n-uplets flexible.

Les requêtes top-k permettent de trier le résultat selon des fonctions de score. Ces fonctions portent sur un ou plusieurs attributs (par exemple, les 10 appartements avec le loyer le moins cher et la surface la plus grande), d'une ou plusieurs tables de la base (par exemple, les 10 appartements avec le loyer et le coût de vie de la ville les moins chers).

Typiquement, une fonction de score est utilisée pour associer chaque n-uplet du résultat avec un score, puis les k n-uplets avec les scores les plus élevés sont restitués. Cependant, d'autres critères de tri du résultat ont été étudiés. Selon [Ilyas et al. 2008], il existe trois modèles de requêtes de type top-k selon le type des objets auxquels on attribue les scores : le modèle top-k sélections, le modèle top-k jointures et le modèle top-k groupes d'agrégation.

Dans le modèle *top-k sélections*, les scores sont attachés aux n-uplets de la base. Le processus d'exécution d'une requête est effectué comme suit. Les n-uplets sont sélectionnés selon les conditions de base de la requête. Puis les n-uplets sélectionnés sont triés selon la fonction de score. L'ensemble non vide des meilleurs n-uplets est ensuite renvoyé.

Dans le modèle *top-k jointures*, les scores sont attribués aux résultats de jointures entre différentes tables de la base. Les jointures entre les tables sont effectuées, puis les n-uplets qui résultent des jointures sont triés pour en retourner les meilleurs.

Selon le modèle *top-k groupes d'agrégation*, les scores sont calculés pour des groupes de n-uplets.

Le traitement d'une requête nécessite d'abord de calculer les groupes de n-tuples puis les trier selon une fonction de score de groupes [Li et al., 2006], telle que la somme. Le regroupement peut être basé sur les attributs d'agrégation de la requête ou sur d'autres attributs [Li et al., 2007].

Ainsi, le processus de tri peut être effectué avant (modèle top-k jointures) ou après (modèles top-k sélections et top-k groupes d'agrégation) l'évaluation de la requête.

Il faut noter que lorsqu'il est difficile de définir une fonction de score ou lorsque les préférences ne sont pas précises, d'autres types de requêtes de tri sont utilisés. Il s'agit essentiellement de requêtes Skyline où seulement les objets non dominés du résultat sont retournés. Introduit initialement par [Börzsönyi et al. 2001], plusieurs travaux ont étudié l'évaluation et l'optimisation de l'opérateur Skyline.

II.6.3.1. Recommandation de requêtes

Les approches de recommandation ont été largement étudiées dans le domaine du web. Certains travaux se basent sur le paradigme OLAP afin de générer des recommandations [Adomavicius et Tuzhilin, 2001] pour les applications de commercial électronique. Mais, peu de travaux [Giacometti et al., 2008, 2009] ; [Sarawagi, 1999, 2000] 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 [Marcel et Negre, 2011] afin de prendre en compte l'aspect navigationnel de l'interrogation des données OLAP. Ils répondent à la problématique suivante :

« Comment guider l'utilisateur 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'utilisateur 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 et
- 2) la proposition de requêtes complètes afin de faciliter l'exploration de l'espace multidimensionnel.

Cependant, seule la deuxième fonctionnalité a été étudiée par la communauté des bases de données multidimensionnelles

[Giacometti et al., 2008, 2009]. 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 [Fan et al., 2011].

Outre les approches de recommandation des domaines du web et des bases de données, une nouvelle approche a été proposée en OLAP, à savoir la recommandation basée sur les exceptions [Kumar et al., 2006];[Sarawagi et al., 1998] permettant de retrouver des données qui sont difficilement repérées à l'aide des analyses OLAP classiques.

Il faut noter que malgré la richesse des travaux sur la recommandation des requêtes SQL de bases de données [Akbarnejad et al. 2010] ; [Chatzopoulou et al., 2009] ; [Khoussainova et al., 2009], les approches proposées ne peuvent pas être directement appliquées à l'OLAP à cause de l'incohérence de leurs modèles de données avec les concepts de base de l'OLAP, à savoir la modélisation multidimensionnelle.

II.6.4. 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 BDM [Stolte, 2003]. Certains travaux considèrent la personnalisation de la visualisation comme une approche de personnalisation de l'interrogation des données [Golfarelli, 2010]. A notre sens, 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'utilisateur 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 [Bellatreche et al., 2005].
- La mise en valeur de certains indicateurs décisionnels, par exemple les chemins de navigation les plus visités [Garrigós et al., 2009].

II.6.5. Personnalisation de la prise de décision

Les travaux de personnalisation dans cet axe se situent au niveau de la dernière étape d'un processus de prise de décision.

Une première approche permet d'aider l'utilisateur à mieux interpréter les données multidimensionnelles afin de prendre une décision pertinente. [Cabanac et al. 2007] proposent d'intégrer des annotations sur le schéma et sur les valeurs d'une BDM afin de conserver les commentaires, les réflexions et les explications formulés lors des analyses. Ces annotations sont restituées par le système OLAP conjointement aux éléments auxquels elles sont rattachées [Jerbi, 2007]. Elles sont utilisées à des fins personnelles (réutilisation des réflexions précédentes) ou collectives (confrontation de différentes interprétations).

Une deuxième approche permet d'aller au-delà de l'assistance à la prise de décision par l'automatisation de la prise de décision [Thalhammer et al., 2001]. Des règles d'analyse sont définies par l'utilisateur afin de traduire le processus de prise de décision manuel. Une action, correspondant à une décision (par exemple changer le prix d'un article) est effectuée suite à un événement au niveau des bases de données sources (par exemple la diminution de la quantité en stock).

II.6.6. Personnalisation de l'interrogation des bases de données multidimensionnelles

Nous avons considéré les travaux les plus significatifs qui répondent au mieux à notre problématique de recherche. Nous examinons les travaux retenus selon les axes *approche globale*, *algorithme de personnalisation* et *implantation du système* et relativement à des critères d'évaluation jugés importants pour ces axes.

II.6.6.1 Critères d'étude des travaux de personnalisation de l'interrogation des BDM

Soit O l'objet renvoyé par la fonctionnalité de personnalisation, O peut être une requête, un ensemble de requêtes, ou un ensemble de tuples.

Les travaux considérés sont évalués suivant trois axes :

- L'approche de personnalisation, concernant la nature et les caractéristiques de la fonctionnalité de personnalisation qui est offerte à l'utilisateur
- L'algorithme appliqué, précisant comment O est généré
- l'implantation du système, indiquant comment la fonctionnalité de personnalisation est mise en œuvre.

II.6.6.2. Personnalisation portant sur le schéma de l'entrepôt de données :

[Bentayeb et al, 2009] propose deux types de personnalisation OLAP dans les entrepôts de données : l'adaptation et la recommandation.

II.6.6.2.1 Personnalisation basée sur l'adaptation :

Selon [Bentayeb et al, 2009], 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 les schémas d'entrepôt de données pour tenir compte des nouvelles informations de l'utilisateur.

II.6.6.2.2. 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, [Bentayeb et al, 2009] proposent de combiner les techniques de datamining 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 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.

II.7. Algorithmes pour la personnalisation

Dans cette partie les travaux de [Bellatreche et al, 2005] et [Bellatreche et al, 2006] et les algorithmes destinés à la personnalisation dans les bases de données multidimensionnelles.

II.7.1 Approche de [Bellatreche et al, 2005]

Les préférences stockées dans un profil utilisateur sont exprimées par l'ordonnancement des membres de toutes les dimensions. Ces préférences sont ajoutées à la requête originale lors de processus de personnalisation. En plus de ces préférences, il y a des contraintes dites de visualisations qui font partie de ce profil, et son rôle réside dans la spécification de la forme de l'affichage pour le résultat.

Dans l'algorithme proposé dans [Bellatreche et al, 2005], la personnalisation de requête utilisateur et ses visualisations sont combinées dès qu'en calcul la partie de réponse la plus intéressante en respectant les contraintes de visualisation.

Si les contraintes de visualisation permettent de définir la structure d'affichage de résultats, l'algorithme a pour seulement de trouver le cube le plus intéressant et visualisable par cette structure, sinon, le travail s'étend vers la recherche d'une structure adéquate pour visualiser ce cube.

Le pré-ordre total défini sur les membres de toutes les dimensions de cube est utilisé pour définir une relation de pré-ordre entre les personnalisations possibles (sous cubes) de ce cube.

Ce pré-ordre entre les sous cubes est défini en respectant la propriété de monotonie suivante :

Etant donné que C_1 est un sous cube d'un cube C_2 , alors C_1 est moins intéressant que C_2 par le fait que C_2 contient plus d'informations que C_1 .

Le principe de cet algorithme est de construire pour un cube donné C , l'ensemble de toutes les personnalisations possibles (sous cubes respectant les préférences utilisateur et les contraintes de visualisation), ensuite, faire ressortir de cet ensemble, un sous cube C^* ayant les caractéristiques suivantes :

- Il existe S^* une structure qui permet de le visualiser.
- Il n'existe pas un autre sous cube plus intéressant que C^* et visualisable (C^* est maximal par rapport la relation de pré-ordre entre les sous cubes).

Dans le cas où les contraintes de visualisation ne précisent pas exactement la structure de sous cube optimal C^* , une structure respectant les préférences de l'utilisateur et ses contraintes de visualisation doit être trouvée. De ce fait, cet algorithme personnalise à la fois, l'ensemble des mesures a présentées à l'utilisateur, et ça présentation.

Dans ce travail, le problème est formulé de la manière suivante :

Pour un cube C , une contrainte de visualisation v et un pré-ordre sur C , trouver une visualisation $\langle C^*, S^* \rangle$ de C tel que : $C^* \in \leq_p \{ C' \subseteq C \mid \exists S', v(C', S') = \text{true} \}$ et $v(C^*, S^*) = \text{true}$. La proposition d'associer des utilités (des entiers dans $[0,1]$) pour chaque membre de toutes les dimensions permet de définir une fonction d'utilité pour un cube par :

Et ensuite : l'ordre \leq_u sur C par : $C \leq_u C'$ si seulement si $Fu(C) \leq Fu(C')$.

Enfin, étant donné deux cubes C et C' tel que : $C \subseteq C'$, en peut dire que $C \leq_u C'$.

Concernant la contrainte de visualisation, cette dernière est obtenue à partir de structure $S = \langle S_1, \dots, S_k \rangle$ et un entier G par : $V_G(C, S) = \text{true}$ si seulement si $\prod_{k \in [1, K]} \epsilon_k \leq D_k \leq G$ avec $k \in [1, K]$.

Etant donné un ensemble fixe de dimensions D , et un pré-ordre total \leq_p entre les membres dans D . pour chaque paire de membres $(m, m') \in D^2$, on dit que m' est préféré que m si seulement si :

$m \leq_p m'$. Et lorsque $m \leq_p m'$ et $m' \leq_p m$, on dit qu'elles sont également préférées.

Par conséquent, on définit un pré-ordre total entre les cubes par : pour deux cubes C_1 et C_2 , C_1 est moins intéressant que C_2 , noté par $C_1 \leq_p C_2$ si pour chaque $m_1 \in M(C_1) \setminus M(C_2)$, il existe $m_2 \in M(C_2) \setminus M(C_1)$ tel que : $m_1 \leq_p m_2$.

De ce fait, deux cubes C_1 et C_2 sont toujours comparables avec respect au \leq_p si l'ensemble des membres qui distingue C_1 de C_2 n'est pas vide, c'est-à-dire que :

$$(M(C_1) \setminus M(C_2)) \cap (M(C_2) \setminus M(C_1)) \neq \emptyset$$

II.7.2 Approche de [Bellatreche et al, 2006]

Contrairement au pré-ordre total défini dans [Bellatreche et al, 2005], le modèle de préférence défini dans [Bellatreche et al, 2006] est basé sur un ordonnancement partiel qui est et selon les auteurs plus flexible et réalisable que le premier.

Ainsi, l'objet personnalisé dans les deux propositions se diffère dès que dans [Bellatreche et al, 2005], les auteurs cherchent à personnaliser la réponse à une requête OLAP (le cube résultant après l'exécution de la requête), tandis que, dans [Bellatreche et al, 2006], l'idée consiste à personnaliser la requête elle-même avant l'exécution (en basant sur ses références).

Dans cette nouvelle proposition, étant donné une requête MDX q , un profil d'utilisateur et une contrainte, l'algorithme proposé calcule une version personnalisée de q de la façon suivante :

- Cette requête personnalisée est une sous requête de q (dans le sens de l'inclusion des requêtes), dont l'objectif est de réduire la cardinalité de l'ensemble de réponse.
- La réponse à cette requête personnalisée doit satisfaire la contrainte.
- Cette requête personnalisée est également la plus intéressante en respectant le profil d'utilisateur.

Dans cette proposition, le profil utilisateur Γ est défini par le tuple $\Gamma = \{<_d, \{<_1, <_N\}\}$ où :

- $<_d$ est un ordonnancement partiel sur l'ensemble de dimensions

$D(C)$ de cube C . Etant donné deux dimensions D_i et D_j , on dit que

D_i est moins intéressant que D_j si $D_i <_d D_j$.

- Pour chaque $i \in [1, N]$, $<_i$ est un ordonnancement partiel sur l'ensemble des membres de D_i , $D_i \in (C)$. Pour deux membres m_1 et m_2 dans $\text{dom}(D_i)$, m_1 est moins intéressant que m_2 si $m_1 <_i m_2$.

L'ordonnancement spécifié d'après le profil utilisateur permet d'obtenir un ordonnancement pour les références de cellules. Supposons t et t_0 deux références de cellules de schéma $D_0 \in D(C)$. Si $\Delta(t, t_0)$ est l'ensemble de dimensions où t et t_0 diffère, alors t et t_0 sont comparables seulement avec le respect aux dimensions les plus importantes dans $\Delta(t, t_0) = \{D_i \in D \mid t(D_i) \neq t_0(D_i)\}$, c.-à-d. l'ensemble de dimensions $\max <_d (\Delta(t, t_0))$.

On dit que t est moins intéressant que t_0 avec respect au Γ noté $t \leq \Gamma t_0$, si pour chaque dimension $D_i \in \max <_d (\Delta(t, t_0))$, on a $t(D_i) <_i t_0(D_i)$.

☞ Exemple 1 :

Etant donné le cube défini comme suivant :

$C = \{D^*_1, D^*_2, D^*_3, f^*\}$ avec
 $D^*_1 = \{\text{north, south, east, west}\}$,
 $D^*_2 = \{\text{drink, food}\}$,
 $D^*_3 = \{2004, 2005\}$.

Supposons $\Gamma = \{<_d, \{<_{\text{Time}}, <_{\text{Product}}, <_{\text{Location}}, <_{\text{Measures}}\}\}$ le profil utilisateur défini sur C par :

- (Year $<_d$ Location) et (Product $<_d$ Location).
- 2002 $<_{\text{Time}}$ 2003 $<_{\text{Time}}$ 2004 $<_{\text{Time}}$ 2005 $<_{\text{Time}}$ 2006
- electronics $<_{\text{Product}}$ shoes $<_{\text{Product}}$ cloth $<_{\text{Product}}$ food $<_{\text{Product}}$ drink.
- (Centre $<_{\text{Location}}$ Tours) et (Centre $<_{\text{Location}}$ Orleans).
- quantity $<_{\text{Measures}}$ price.

Supposons $t = \{2005, \text{food, Centre, price}\}$ et $t_0 = \{2006, \text{drink, Orleans, price}\}$ deux références de schéma $\text{sch}(t) = \text{sch}(t_0) = \{\text{Time, Product, Location, Measures}\}$.

Nous avons $D(t, t_0) = \{\text{Time, Product, Location}\}$.

Puisque $\max <_d (D(t, t_0)) = \{\text{Location}\}$ et $t(\text{Location}) = \text{Centre} <_{\text{Location}} t_0(\text{Location}) = \text{Orleans}$, on a : $t < \Gamma t_0$.

Dans [Bellatreche et al, 2006], la personnalisation est procédée au niveau de la requête en comparant leurs ensembles de références. Pour un profil défini sur un cube C et deux requêtes MDX q et q_0 définies sur C , q est moins intéressante que q_0 dénoté par $q \leq \Gamma q_0$ si $\text{sch}(q) = \text{sch}(q_0)$ et $(\forall t \in \text{ref}(q)) (\exists t_0 \in \text{ref}(q_0)) (t \leq \Gamma t_0)$.

☞ Exemple 2:

Supposons les deux requêtes MDX q et q_0 définies par :

```
q : SELECT [Location].members ON COLUMNS,
CROSSJOIN({[year].2005, [year].2006},
{[category].shoes, [category].cloth}) ON ROWS
FROM Sales Cube
```

```
WHERE [Measures].quantity
```

```
q0 : SELECT [Location].members ON COLUMNS,
CROSSJOIN({[year].2003, [year].2004,
```

```
[year].2005, [year].2006},
{[category]. food, [category].drink}) ON ROWS
FROM Sales Cube
WHERE [Measures].quantity
```

On a $\text{ref}(q) = \{2005, 2006\} \times \{\text{shoes, cloth}\} \times \text{adom}(\text{Location}) \times \{\text{quantity}\}$ et $\text{ref}(q_0) = \{2003, 2004, 2005, 2006\} \times \{\text{food, drink}\} \times \text{adom}(\text{Location}) \times \{\text{quantity}\}$.

D'après le profil utilisateur défini dans l'exemple précédent, on peut remarquer que q_0 est préférée que q , c.-à-d. $q \leq_{\Gamma} q_0$ puisque : $\forall t \in \text{ref}(q), \exists t_0 \in \text{ref}(q_0)$ tel que : $t \leq_{\Gamma} t_0$.

Par exemple, pour le tuple $t = \{2005, \text{shoes, Orleans, quantity}\} \in \text{ref}(q)$, il existe un le tuple $t_0 = \{2005, \text{food, Orleans, quantity}\} \in \text{ref}(q_0)$ où $\Delta(t, t_0) = \{\text{Product}\}$.

Dès que $\max_{<d}(\Delta(t, t_0)) = \{\text{Product}\}$ et $t(\text{Product}) <_{\text{Product}} t_0(\text{Product})$, alors $t \leq_{\Gamma} t_0$.

L'un des critères que la requête personnalisée doit satisfaire est la contrainte de visualisation.

Cette satisfaction est définie comme suit : étant donné un cube N-dimensionnel

$C = \langle D_1, \dots, D_N, f \rangle$, une structure K-dimensionnel $T = \langle T_1, \dots, T_k \rangle$ et un K-tuple des entiers $G = \langle G_1, \dots, G_K \rangle$. La définition d'une contrainte de visualisation $V_{T, G}$ pour chaque requête $q \in Q_{\text{MDX}}$ est donnée par : $V_{T, G}(q) = \text{true}$ si q contient K axes et :

- Pour $k \in [1, K]$, on a $T_k \subseteq S_k(q)$, cette contrainte signifie que l'utilisateur veut voir les dimensions dans T_k sur l'axe k de q .
- Pour $k \in [1, K]$, on a $|\text{ref}_k(q)| \leq G_k$, cette contrainte signifie que l'utilisateur peut avoir un nombre maximal G_k sur l'axe K de q .

On arrive maintenant à la formulation de problème dans [Bellatreche et al, 2006] tel que :

Etant donné :

- Une requête MDX $q \in Q_{\text{MDX}}$.
- Une contrainte $v \in V$ définie sur Q_{MDX} , et
- Un profil utilisateur Γ sur un cube C .

Le but est de calculer l'ensemble Q^* des requêtes personnalisées défini par :

$$Q^* = \max_{\leq_{\Gamma}} \{q' \in Q_{\text{MDX}} \mid q' \subseteq q \wedge v(q') = \text{true}\}.$$

Ce calcul peut être effectué par l'algorithme en deux phases :

- **Phase 1** : Calculer des sous-ensembles de références de q les plus intéressants qui peuvent être visualisés en respectant la contrainte v . Cet ensemble est dénoté dans la fonction *Perso* (voir la figure) par $Mi-1$. En utilisant la fonction *MaxSubset* présentée sur la figure, cet

ensemble est calculé itérativement dans la boucle principale de fonction *Perso* (voir les étapes 3-11).

• **Phase 2** : Déterminer les structures qui permettent de visualiser les sous-ensembles de références de q les plus intéressants calculés dans la première phase. Noter que ces structures sont calculées en utilisant la fonction *FindStruct* de [Bellatreche et al, 2005] dans la deuxième boucle de la fonction *Perso* (étapes 13-15).

Pour la première phase, l'ensemble des références T de q est partitionné en basant sur l'ordonnement \leq_{Γ} comme suit : $T = T_1 \cup \dots \cup T_P$

Tel que :

- $T_1 = \max_{\leq_{\Gamma}}(T)$. T_1 contient les références de T les plus intéressants.
- $T_{i+1} = \max_{\leq_{\Gamma}}(T \setminus (\bigcup_{k=1}^i T_k))$ pour $i \in [0, P-1]$.

Pour deux sous-ensembles X et Y de $\text{ref}(q)$, on dit que X est moins intéressant que Y dénoté par $X \leq_{\Gamma} Y$ si pour chaque $x \in X$, il existe $y \in Y$ tel que $x \leq_{\Gamma} y$.

En plus, pour chaque sous-ensemble X de $\text{ref}(q)$, une fonction booléenne C_v est défini par : $C_v(X) = \text{true}$ s'il existe une structure S (pour visualiser X) et une requête $q' \in q[X, S]$ tel que : $v(q') = \text{true}$.

<p>Input: A set of references R such that $C_v(R) = \text{true}$ A set of references T_i that are not comparable w.r.t. \leq_{Γ}</p> <p>Output: The set $\mathcal{M} = \max_{\subseteq} \{R \cup X \mid X \subseteq T_i \wedge C_v(R \cup X) = \text{true}\}$</p> <hr/> <ol style="list-style-type: none"> 1. Let $k = 1$ 2. Let $\mathcal{I}_0 = \{\emptyset\}$ and $\mathcal{C}_1 = \{\{t\} \mid t \in T_i\}$ 3. While $(\mathcal{C}_k \neq \emptyset)$ do 4. Let $\mathcal{I}_k = \{X \in \mathcal{C}_k \mid C_v(R \cup X) = \text{true}\}$ 5. Let $\mathcal{C}_{k+1} = \emptyset$ 6. For every $S_1 \in \mathcal{I}_k$ and $S_2 \in \mathcal{I}_k$ do 7. If $S_1 \cap S_2 = k - 1$ then 8. Insert $(S_1 \cup S_2)$ in \mathcal{C}_{k+1} 9. For every $S \in \mathcal{C}_{k+1}$ do 10. For every $S' \subset S$ such that $S' = k$ do 11. If $S' \notin \mathcal{I}_k$ then 12. Remove S from \mathcal{C}_{k+1} 13. $k = k + 1$ 14. end while 15. Let $\mathcal{M} = \max_{\subseteq} (\bigcup_{i=0}^{k-1} \{R \cup X \mid X \in \mathcal{I}_i\})$ 16. Return \mathcal{M}

Figure II.2.L'algorithme *Maxsubset* [Bellatreche et al, 2006]

<p>Input: A MDX query $q \in Q_{MDX}$ An anti-monotone constraint $v \in \mathcal{V}$ A user profile I</p> <p>Output: The set Q^* of most-interesting sub-queries of q w.r.t. \preceq_I that satisfy v</p> <hr/> <ol style="list-style-type: none"> 1. Let $i = 1$ 2. Let $\mathcal{M}_0 = \{\emptyset\}$ 3. Let $T_1 = \max_{\preceq_I}(ref(q))$ 4. While $(T_i \neq \emptyset)$ do 5. Let $\mathcal{M}_i = \emptyset$ 6. For every $R \in \mathcal{M}_{i-1}$ 7. Let $\mathcal{M}_i = \mathcal{M}_i \cup MaxSubset(R, T_i)$ 8. Let $T_{i+1} = \max_{\preceq_I}(ref(q) \setminus (\cup_{k=1}^i T_k))$ 9. $i = i + 1$ 11. end while 12. Let $Q^* = \emptyset$ 13. For every $R \in \mathcal{M}_{i-1}$ 14. For every structure S such that $v(q[R, S]) = true$ 15. $Q^* = Q^* \cup \{q[R, S]\}$ 16. Return Q^*
--

Figure II.3. l'algorithme *Perso* [Bellatreche et al, 2006]

☞ Exemple 2 :

Cet exemple illustre le fonctionnement de fonctions *Perso* et *MaxSubset* pour personnaliser une requête MDX. Prenons le profil utilisateur de l'exemple 8, et q la requête MDX suivante avec la contrainte de visualisation VT, G définie par $T = \langle \emptyset, \emptyset \rangle$ et $G = \langle 4, 4 \rangle$:

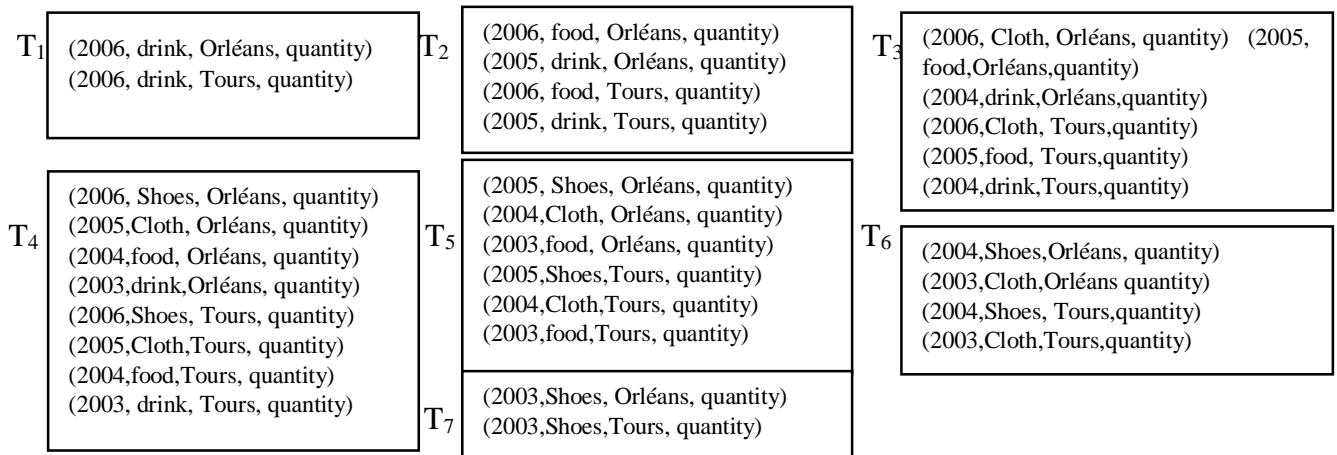
```
q : SELECT {[Location].Tours, [Location].Orleans},
      {[year].2003,[year].2004, [year].2005,[year].2006},
      {[category].shoes,[category].cloth,[category].food, [category].drink}
FROM Sales Cube
WHERE [Measures].quantity
```

Noter que l'utilisateur ne rien dit concernant la structure utilisée pour afficher la réponse.

Pour cette requête, l'ensemble de références $ref(q) = \{2006, 2005, 2004, 2003\} \times \{\text{Orleans, Tours}\} \times \{\text{drink, food}\} \times \{\text{quantity}\}$.

D'ailleurs, $\text{ref}(q) = T_1 \cup \dots \cup T_7$ où les ensembles des références T_k ($k \in [1, 7]$)

sont :



La fonction *Perso* calcul l'ensemble des requêtes personnalisées $Q^* = \text{Perso}(q, VT, G, \Gamma)$ de la façon suivante :

- A la première itération ($i = 1$), on a $M_1 = \text{MaxSubset}(\emptyset, T_1) = \{T_1\}$ lorsque $VT, G(T_1) = \text{true}$.
- De la même manière, nous pouvons voir que $M_2 = \text{MaxSubset}(T_1, T_2) = \{T_1 \cup T_2\}$.
- A la troisième itération ($i = 3$), on a $M_3 = \text{MaxSubset}(T_1 \cup T_2, T_3)$. Lorsque il n'est pas possible de visualiser les mesures concernant 3 années, 3 produits et 2 locations avec une table croisée qui n'a plus de 4 lignes et 4 colonnes, on a $VT, G(T_1 \cup T_2 \cup T_3) = \text{false}$. Alors, nous n'avons pas $M_3 = \{T_1 \cup T_2 \cup T_3\}$.

En effet, nous pouvons montrer que $M_3 = \{R_3, R'_3\}$ lorsque :

$R_3 = T_1 \cup T_2 \cup \{(2005, \text{food}, \text{Orleans}, \text{quantity}), (2004, \text{drink}, \text{Orleans}, \text{quantity}), (2005, \text{food}, \text{Tours}, \text{quantity}), (2004, \text{drink}, \text{Tours}, \text{quantity})\}$, et

$R'_3 = T_1 \cup T_2 \cup \{(2006, \text{cloth}, \text{Orleans}, \text{quantity}), (2005, \text{food}, \text{Orleans}, \text{quantity}), (2006, \text{cloth}, \text{Tours}, \text{quantity}), (2005, \text{food}, \text{Tours}, \text{quantity})\}$.

Nous détaillons maintenant comment M_3 est calculé par la fonction *MaxSubset* :

1. A la première itération ($k = 1$) de fonction *MaxSubset*, C_1 contient toutes les 1- sous-ensembles de T_3 , c.-à-d. 6 éléments. D'ailleurs, on a $I_1 = C_1$ lorsque toutes les candidates dans C_1 sont intéressantes avec le respect au C_v .
2. A la deuxième itération ($k = 2$) de fonction *MaxSubset*, C_2 contient toutes les 2-sous-ensembles de T_3 , c.-à-d., 15 éléments. Noter que quatre éléments de C_2 sont intéressants avec respect au C_v quand il est possible de visualiser seulement 2 années, 3 produits et 2 locations, ou 3 années, 2 produits et 2 locations. Les éléments de C_2 qui ne sont pas intéressants :

{(2006, cloth, Orleans, quantity), (2004, drink, Orleans, quantity)},
 {(2006, cloth, Orleans, quantity), (2004, drink, Tours, quantity)},
 {(2004, drink, Orléans, quantity), (2006, cloth, Tours, quantity)},
 {(2006, cloth, Tours, quantity), (2004, drink, Tours, quantity)} dès que l'ajout de l'un de ces 2-sous-ensembles va donner 3 années, 3 produits et 2 locations, ce qui ne peut pas être visualisé.

3. A la troisième itération ($k = 3$), nous pouvons montrer que C_3 contient 8 éléments et que sont tous intéressants, c.-à-d. $I_3 = C_3$.

4. A la quatrième et dernière itération ($k = 4$), nous pouvons montrer que C_4 contient 2 éléments et que sont tous intéressants, c.-à-d. $I_4 = C_4$ (voir R_3 et R'_3 quand ces 4-sous-ensembles de T_4 sont ajoutés).

– A la quatrième itération ($i = 4$) de fonction *Perso*, on a $M_4 = \text{MaxSubset}(R_3, T_4) = \{R_4, R'_4\}$ lorsque :

$R_4 = R_3 \cup \{(2004, \text{food}, \text{Orleans}, \text{quantity}), (2003, \text{drink}, \text{Orleans}, \text{quantity}),$
 $(2004, \text{food}, \text{Tours}, \text{quantity}), (2003, \text{drink}, \text{Tours}, \text{quantity})\}$, et

$R'_4 = R'_3 \cup \{(2006, \text{shoes}, \text{Orleans}, \text{quantity}), (2005, \text{cloth}, \text{Orleans}, \text{quantity}),$
 $(2006, \text{shoes}, \text{Tours}, \text{quantity}), (2005, \text{cloth}, \text{Tours}, \text{quantity})\}$.

– A cinquième itération ($i = 5$), on $M_5 = \text{MaxSubset}(R_4, T_5) = \{R_5, R_{05}\}$ lorsque :

$R_5 = R_4 \cup \{(2003, \text{food}, \text{Orleans}, \text{quantity}), (2003, \text{food}, \text{Tours}, \text{quantity})\}$, et

$R'_5 = R'_4 \cup \{(2005, \text{shoes}, \text{Orleans}, \text{quantity}), (2005, \text{shoes}, \text{Tours}, \text{quantity})\}$.

– A la sixième itération ($i = 6$), on a $M_6 = \text{MaxSubset}(R_5, T_6) = \{R_5, R'_5\}$

lorsque plus de catégories de product et years peuvent être ajoutés à l'ensemble de références R_5 or R'_5 de sorte qu'ils puissent être visualisés en utilisant une table croisée qui n'a plus de 4 lignes et 4 colonnes.

– Pour la même raison, quand $i = 7$, on a $M_7 = \text{MaxSubset}(R_5, T_7) = \{R_5, R_{\cdot 5}\}$ lorsque ni $R_5 \cup T_7$ ni $R'_5 \cup T_7$ est affichable.

– a la fin, on a $Q^* = \{q[R_5, S_{5a}], q[R_5, S_{5b}], q[R'_5, S'_{5a}], q[R'_5, S'_{5b}]\}$ lorsque :

$R_5 = \{2006, 2005, 2004, 2003\} \times \{\text{Orleans}, \text{Tours}\} \times \{\text{drink}, \text{food}\} \times \{\text{quantity}\}$,

$S_{5a} = \langle \{\text{Year}\}, \{\text{Location}, \text{Product}\}, \{\text{Measures}\} \rangle$,

$S_{5b} = \langle \{\text{Location}, \text{Product}\}, \{\text{Year}\}, \{\text{Measures}\} \rangle$,

$R'_5 = \{2005, 2006\} \times \{\text{Orleans}, \text{Tours}\} \times \{\text{drink}, \text{food}, \text{cloth}, \text{shoes}\} \times \{\text{quantity}\}$,

$S'_{5a} = \langle \{\text{Location}, \text{Year}\}, \{\text{Product}\}, \{\text{Measures}\} \rangle$,

$S'_{5b} = \langle \{Product\}, \{Location, Year\}, \{Measures\} \rangle$.

Par exemple on peut présenter à l'utilisateur la requête MDX personnalisée correspond à $q[R_5, S_{5a}]$. S_{5a} est essentiellement utilisée pour déterminer comment les dimensions apparaîtront dans la clause ON de la requête MDX. R_5 est utilisé pour fournir pour chaque dimension l'extension de l'ensemble de membres apparaissant dans la table croisée résultante.

De ce fait, la requête personnalisée correspondant à $q[R_5, S_{5a}]$ qui peut être proposée à l'utilisateur sera :

```
SELECT {[year].2003,[year].2004,
[year].2005, [year].2006} ON COLUMNS,
CROSSJOIN ({[Location].Tours,[Location].Orleans},
{[category].food,[category].drink}) ON ROWS
FROM Sales Cube
WHERE [Measures].quantity
```

D'après cet exemple de personnalisation, on peut remarquer que le résultat contient deux sous-ensembles de références et pour chacun deux structures d'affichages possibles.

Cela est due à l'utilisation de l'ordre partiel strict qui ne permet pas d'établir la relation de préférence entre tous les éléments de profil utilisateur (membres et dimensions) et par conséquent on peut obtenir lors de calcul de sélection des références les plus intéressantes en terme de préférences utilisateur et contraintes de visualisation, plusieurs sous-ensembles de références également importants. Cette égalité d'importance entre les sous-ensembles de références est traduit par l'existence des références (l'ordre entre les sous-ensembles des références est obtenu à partir de l'ordre entre les références) non comparables en fonction de la relation de préférence définie par les préférences utilisateur exprimés dans son profil.

Absence d'une approche globale de personnalisation

L'une des limites des approches existantes est le manque de capacité de fournir des services de personnalisation à différents niveaux de détail (une partie d'une requête, une requête complète, ...) et à des moments différents de la phase d'analyse (lors de la définition de la requête, après la définition de la requête, après la génération du résultat).

Conclusion :

Nous avons constaté l'absence d'une approche d'assistance à la définition des requêtes OLAP ainsi que l'absence d'un mécanisme global combinant les fonctionnalités de personnalisation et de recommandation.

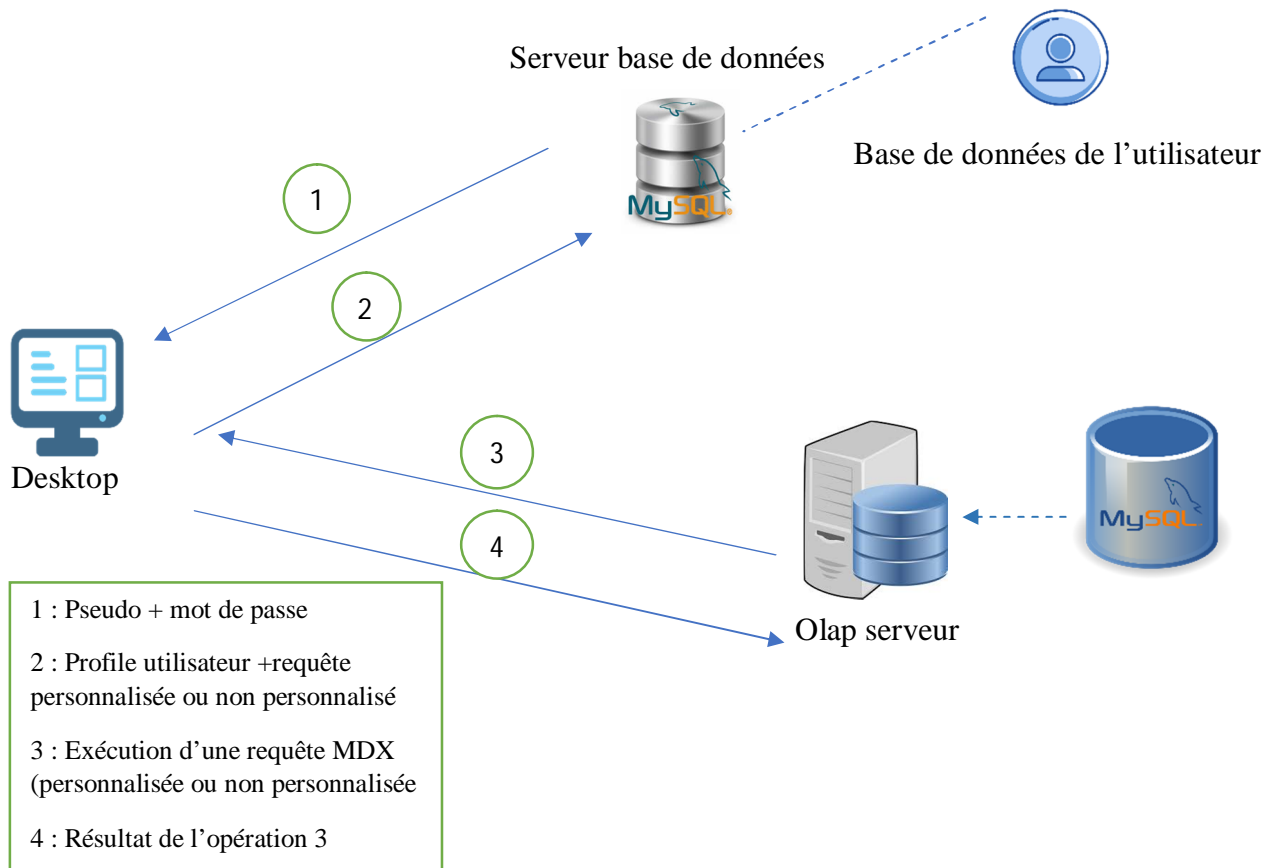
Chapitre III

L'implémentation

III.1. Introduction :

Après avoir exploré précédemment les algorithmes de personnalisation des requêtes OLAP, ce chapitre a pour objectif de montrer et discuter l'application qui implémente ces algorithmes de personnalisation prenant en compte les profils des utilisateurs.

Le schéma suivant décrit le principe de base pour le fonctionnement de notre système de personnalisation :



P

III.2. Technologies employées :

Le schéma montre plusieurs technologies participées dans la réalisation de ce système tel que :

Mondrian : est un serveur OLAP (On Line Analytical Processing) disponible sous licence open source.

Il fait partie de la catégorie des serveurs R-OLAP, c'est-à-dire qu'il accède à des données contenues dans une base relationnelle.

Mondrian exécute des requêtes utilisant le langage MDX, également utilisé dans

Microsoft SQL Server. Ce langage permet de créer des requêtes dont l'équivalent en langue SQL nécessiterait un grand nombre de requêtes et des temps d'exécution beaucoup plus longs. Ce serveur est le plus souvent utilisé conjointement avec JPivot ou JRubik (Présentés plus loin), outils qui proposent une interface graphique de consultation et manipulation des données.

Le projet Mondrian a maintenant rejoint le projet Pentaho.

III.3. Configuration de Mondrian avec Tomcat et MySQL Server

Dans cet article nous essayons d'installer et configurer Mondrian en utilisant le serveur de base de données MySQL.

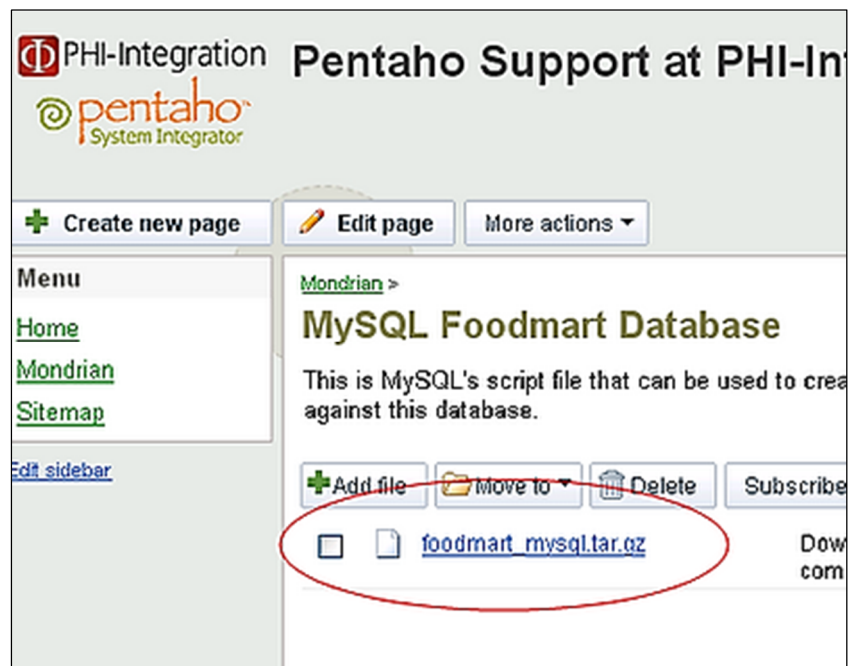
Prés requis :

Il y a des applications qui sont nécessaires avant d'utiliser Mondrian :

- **Java Development Kit 5 ou récent**
- **Apache Tomcat**
- **MySQL 5**
- **MySQL 5 JDBC driver**

III.3.1. La base de données test FoodMart de MySQL

Vous pouvez récupérer le fichier compressé de la base de données test Foodmart pour notre application sur le site de projet Pentaho et le décompresser dans un répertoire de votre convenance.



III.3.2. Création et population de la base de données

Ouvrir l'invite de commande et placez-vous sur le répertoire \bin de MySQL, dans notre cas c'est **C:\Program Files\MySQL\MySQL Server 5.0\bin** avec le fichier décompressé précédemment dans **C:\foodmart_mysql.sql**.

Tapez la commande **mysqladmin -u root -p create foodmart**. Cette commande permet de créer une base de données et un utilisateur foodmart. Le mot de passe par défaut de l'utilisateur foodmart est foodmart. Connectez-vous à l'aide de l'utilisateur root et donnez ensuite les privilèges à l'utilisateur foodmart : **grant all privileges on *.* to 'foodmart'@'localhost' identified by 'foodmart'** ; Ces privilèges permettent à l'utilisateurs "foodmart" de se connecter en local et d'exploiter la base de données "foodmart".

Ensuite, Utilisez cette commande **mysql.exe uroot Foodmart < C:\foodmart_mysql.sql** pour peupler la base de données "FoodMart".

Et finalement, utilisez cette commande **mysql.exe -uroot Foodmart -e "show tables"** pour visualiser les tables que contient notre base de données "Foodmart", vous devez avoir cette liste des tables :

```

+-----+
| Tables_in_foodmart |
+-----+
| account |
| agg_c_10_sales_fact_1997 |
| agg_c_14_sales_fact_1997 |
| agg_c_special_sales_fact_1997 |
| agg_g_ms_pcat_sales_fact_1997 |
| agg_l_03_sales_fact_1997 |
| agg_l_04_sales_fact_1997 |
| agg_l_05_sales_fact_1997 |
| agg_lc_06_sales_fact_1997 |
| agg_lc_100_sales_fact_1997 |
| agg_ll_01_sales_fact_1997 |
| agg_pl_01_sales_fact_1997 |
| category |
| currency |
| customer |
| days |

```

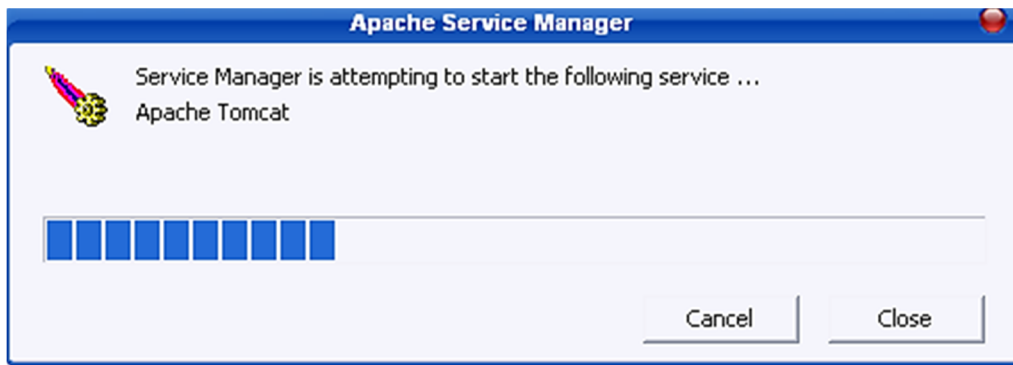
department	
employee	
employee_closure	
expense_fact	
inventory_fact_1997	
inventory_fact_1998	
position	
product	
product_class	
promotion	
region	
reserve_employee	
salary	
sales_fact_1997	
sales_fact_1998	
sales_fact_dec_1998	
store	
store_ragged	
time_by_day	
warehouse	
warehouse_class	
+-----+	

III.3.3. Configuration de Tomcat

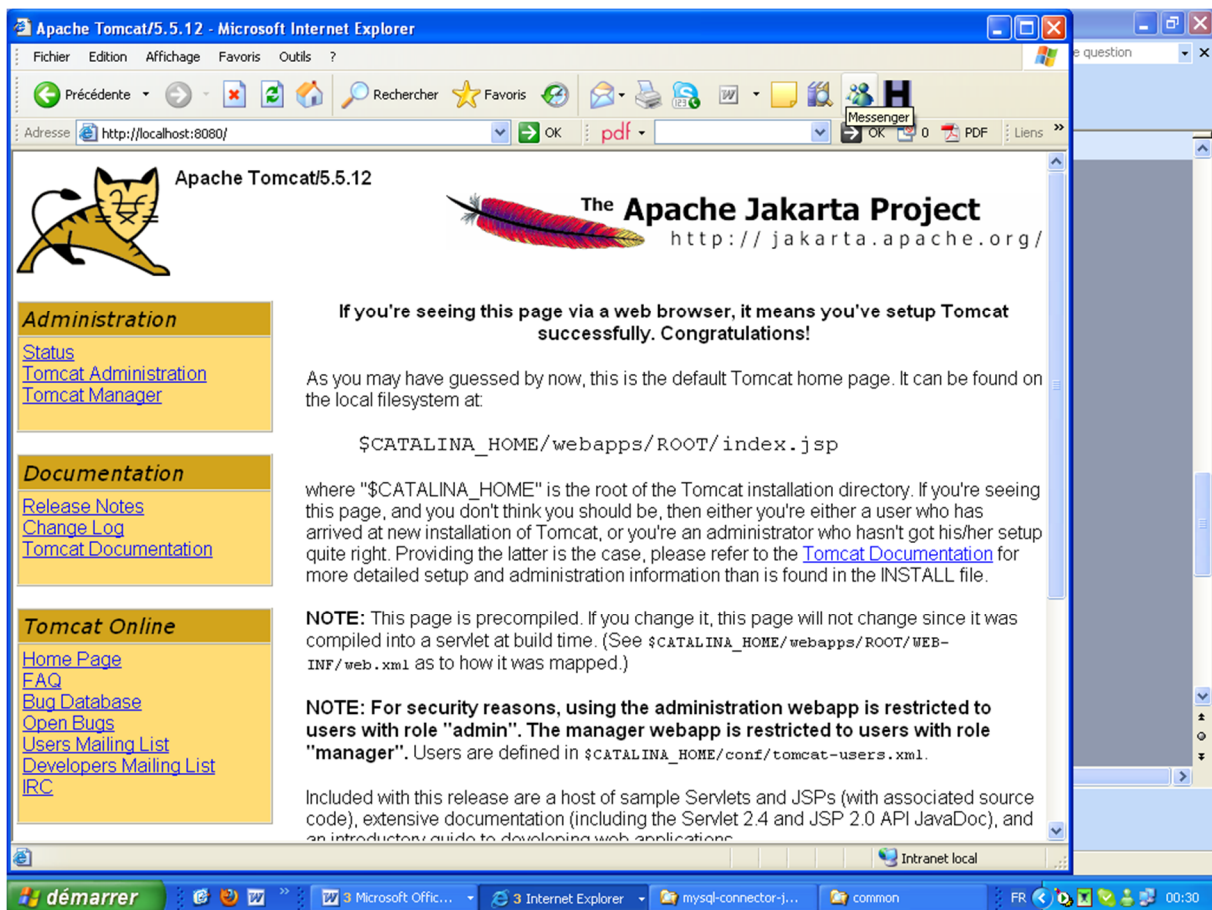
Premièrement, il faut copier le pilote MySQL, i.e. le fichier **mysql-connector-java-5.1.7-bin.jar** dans le dossier librairie de Tomcat.

- Avant d'obtenir le fichier jar vous devez décompresser le fichier de pilote MySQL zippé.
- Copier ce jar driver dans le dossier **C:\Program Files\Apache Software Foundation\Tomcat 5.5\common\lib**.

Pour lancer Tomcat, vous pouvez utiliser « Apache Service Manager », allez à Démarrer, Programmes, Apache Tomcat 5.5, Monitor Tomcat.

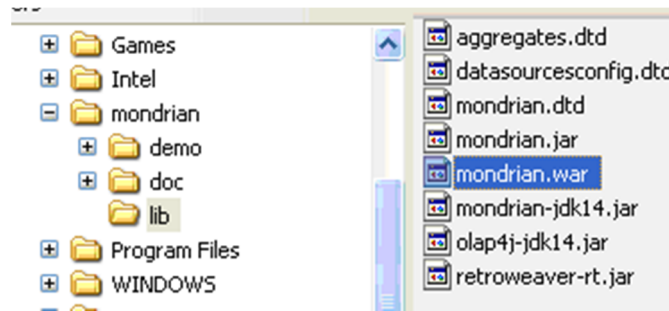


Pour s'assurer que Tomcat fonctionne, accédez à l'adresse **http://localhost:8080/** avec votre navigateur. Si tout fonctionne, vous devrez avoir une page qui ressemble à celle-ci.

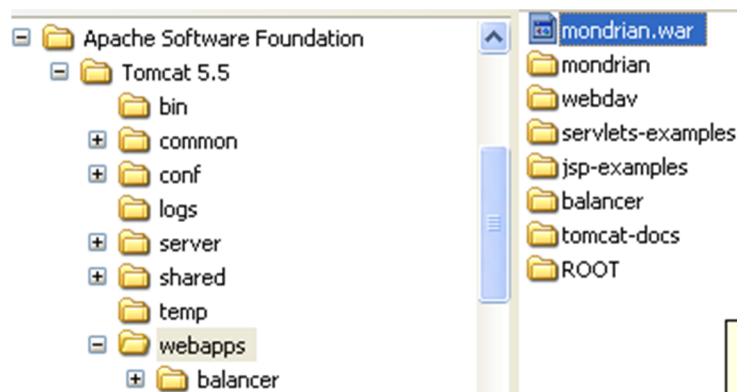


III.3.4. Installation et configuration de Mondrian

Décompresser le package de distribution Mondrian et allez sur son répertoire **lib** où vous devez trouver le fichier **mondrian.war**.



Copier ce fichier war dans le répertoire **webapps** de Tomcat le temps où ce dernier est toujours démarré. Le fichier war sera automatiquement décompressé pour donner un nouveau sous répertoire porte le même nom que le fichier war.



Pour vérifier si tout ça marche bien, tapez cette adresse dans votre navigateur **http://localhost:8080/mondrian**, vous devez obtenir la page de bienvenue de Mondrian



Mondrian exemples:

- [JPivot pivot table](#)
- [JPivot pivot table by XMLA](#)
- [JPivot with 4 hierarchies](#)
- [JPivot with role 'California Manager' set](#)
- [JPivot with arrows](#)
- [JPivot with colors](#)
- [Various queries formatted using the Mondrian tag-library](#)
- [Basic interface for ad hoc queries](#)
- [XML for Analysis tester](#)

Other links:

- [Mondrian home page](#)
- [Mondrian project page](#)
- [JPivot home page](#)
- [JPivot project page](#)

III.3.5. Configuration de Mondrian avec MySQL

Modifier les fichiers :

- Fourheir.jsp
- Mondrian.jsp
- Colors.jsp
- Arrows.jsp

Qui se trouvent dans **C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\mondrian\WEB-INF\queries** en Remplaçant la ligne

```
<jp:mondrianQueryid="query01"jdbcDriver="sun.jdbc.odbc.JdbcOdbcDriver"
jdbcUrl="jdbc:odbc:MondrianFoodMart"catalogUri="/WEB-INF/queries/FoodMart.xml">
```

Par

```
<jp:mondrianQueryid="query01"jdbcDriver="com.mysql.jdbc.Driver"
jdbcUrl="jdbc:mysql://localhost/foodmart?user=foodmart&password=foodmart"
catalogUri="/WEB-INF/queries/FoodMart.xml">
```

☞ **Note :**

Pour notre application, on va utiliser pour le serveur **MySQL Server** le port 3307 au lieu 3306, puisque ce dernier sera utilisé pour le serveur **MySQL** de **EasyPHP** qui va gérer la base de données comportant les préférences utilisateur, alors la ligne précédente doit être sous la forme :

```
<jp:mondrianQueryid="query01"jdbcDriver="com.mysql.jdbc.Driver"
jdbcUrl="jdbc:mysql://localhost:3307/foodmart?user=foodmart&password=foodmart"
catalogUri="/WEB-INF/queries/FoodMart.xml">
```

Exploration de données multidimensionnelles:

Maintenant vous êtes prêt à travailler avec JPivot, allez sur:

http://localhost:8080/mondrian, et choisissez par exemple le premier lien "JPivot pivot table".

vous allez obtenir cette page qui montre que Mondrian fonctionne correctement :

Adresse http://localhost:8080/mondrian/testpage.jsp?query=mondrian

Test Query uses Mondrian OLAP

MDX

		Mesures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
+All Media	+All Products	266 773	225 627,23	565 238,13

Slicer: [Year=1997]

[back to index](#)

Tomcat : Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration

XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

Tomcat a été écrit en langage Java, il peut donc s'exécuter via la JVM (machine virtuelle java) sur n'importe quel système d'exploitation la supportant.

Easyphp : fut le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur descript (PHP), ainsi qu'une administration SQL php MyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP.

Par défaut, le serveur Apache crée un nom de domaine virtuel (en local) 127.0.0.1 ou localhost. Ainsi, quand on choisit « Web local » dans le menu d'EasyPHP, le navigateur s'ouvre sur cette URL et affiche la page index. php de ce site qui correspond en fait au contenu du dossier www d'EasyPHP.

MySQL : est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle ou Microsoft SQL Server.

Java : Apparu en 1991, le langage JAVA a commencé à être intéressant à partir de 1995 avec sa prise en charge par le navigateur phare de l'époque, Netscape.

Ce langage ne cesse de se développer. Il s'agit d'un langage orienté objet dont la syntaxe est très proche de celle du C++.

C'est également un langage portable, c'est à dire qu'il s'adapte à une foule de plateformes différentes. C'est là l'une des qualités de JAVA.

III.4 Présentation de prototype :

Les données de l'entrepôt sont interrogées indirectement par le système. Par la baie de Mondrian, un serveur OLAP destinée à l'exécution des requêtes multidimensionnelles (MDX) sur les données d'un entrepôt de données et le formatage des résultats sous forme d'une table croisée.

L'interface utilisateur :

C'est l'ensemble des différentes fenêtres constituant l'interface de l'application et qui permet à un utilisateur de :

- S'authentifier sur le serveur (nom d'utilisateur et mot de passe) :

Chaque utilisateur a son propre nom d'utilisateur et mot de passe pour naviguer sur ce système.

- Définir ces préférences ou les modifier : Ces préférences représentent l'ordonnement choisi par l'utilisateur sur l'ensemble des dimensions de cube de données, ainsi que sur les niveaux des dimensions et bien sûr, un autre ordonnement concerne les membres des niveaux. Cet ordonnement est partiel, c.-à-d. que l'utilisateur n'est pas obligé de classer toutes les dimensions, les niveaux de chaque dimension et les membres de chaque niveau dans chaque dimension.

En plus de ces ordonnancements, l'utilisateur peut spécifier quelques contraintes concernant la forme que doit prendre un résultat sur l'écran de son mobile. Ces contraintes permettent à l'utilisateur de désigner un ensemble de dimensions qu'il souhaite voir toujours sur le premier axe et un autre ensemble sur le second. Ainsi, l'utilisateur peut indiquer le nombre maximal des membres que doit comporter chaque axe, on parle de nombre de graduation pour chaque axe.

- **Parcourir ses requêtes prédéfinies** : Ces requêtes ont été personnalisées ou non, et sont stockées sur son profil. L'utilisateur peut effectuer les différentes opérations offertes sur ses requêtes tel que :

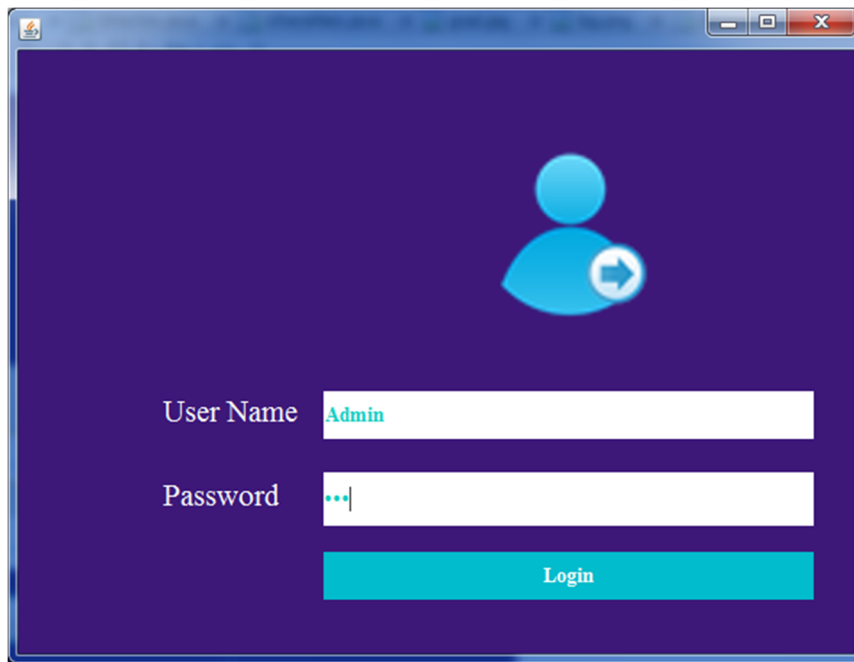
- L'édition d'une requête non personnalisée afin de la modifier,
- Le changement le nom d'une requête,
- La suppression d'une requête,
- La personnalisation d'une requête non personnalisée,
- L'exécution d'une requête personnalisée.

- **Construire une requête a zéro** : C'est l'interface qui aide l'utilisateur à construire une nouvelle requête en passant par les étapes suivantes :

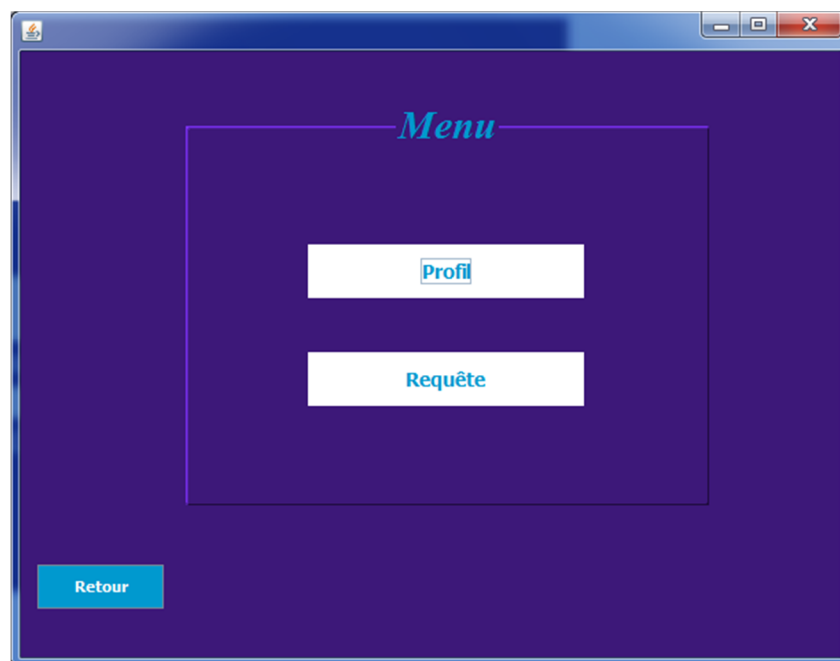
- Sélection de cube.
- Sélection de l'ensemble des dimensions pour la clause« SELECT ».
- Sélection pour chaque dimension sélectionnée dans l'étape précédente, un ensemble des niveaux.
- Sélection pour chaque niveau sélectionné dans l'étape précédente, un ensemble des membres.
- Sélection de l'ensemble des dimensions pour la clause« WHERE ». Cet ensemble doit être différente de celle de la clause« SELECT ». (Cette étape est optionnelle, c.-à-d. que l'utilisateur peut ne choisit aucune dimension)
- Sélection pour chaque dimension sélectionnée dans l'étape précédente, un niveau et un membre. (Cette étape aura lieu si l'utilisateur a choisi dans l'étape précédente au moins une dimension).
- La sauvegarde de la requête générée sur le profil de l'utilisateur sous un nom bien défini.
- **Personnalisée une requête non personnalisée ou exécuter une autre personnalisée** : Ces requêtes ont été des requêtes générées nouvellement ou des requêtes de l'ensemble stockées sur le profil utilisateur.

Le résultat de la personnalisation d'une requête non personnalisée peut contenir plus d'une requête personnalisée due à l'utilisation de l'ordre partiel. Dans ce cas, l'utilisateur peut utiliser la discrimination entre les requêtes résultantes afin de réduire leur nombre si possible.

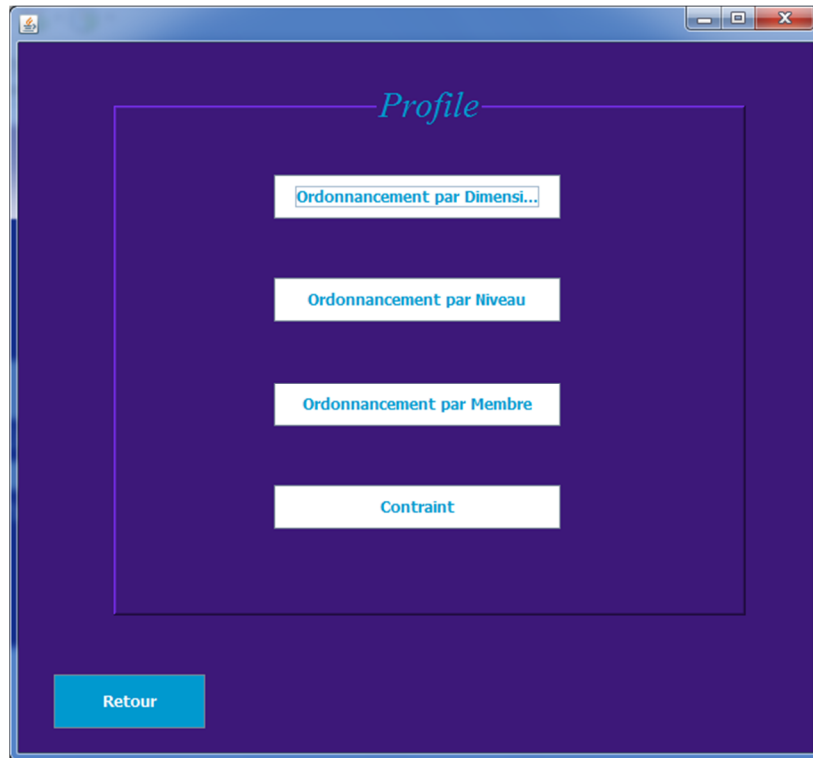
- L'exécution de la requête personnalisée afin d'afficher le résultat.

A screenshot of a web application window with a dark blue background. At the top center is a light blue icon of a person with a circular arrow. Below it are two white input fields. The first is labeled 'User Name' and contains the text 'Admin'. The second is labeled 'Password' and contains three dots. Below the password field is a red button labeled 'Login'.

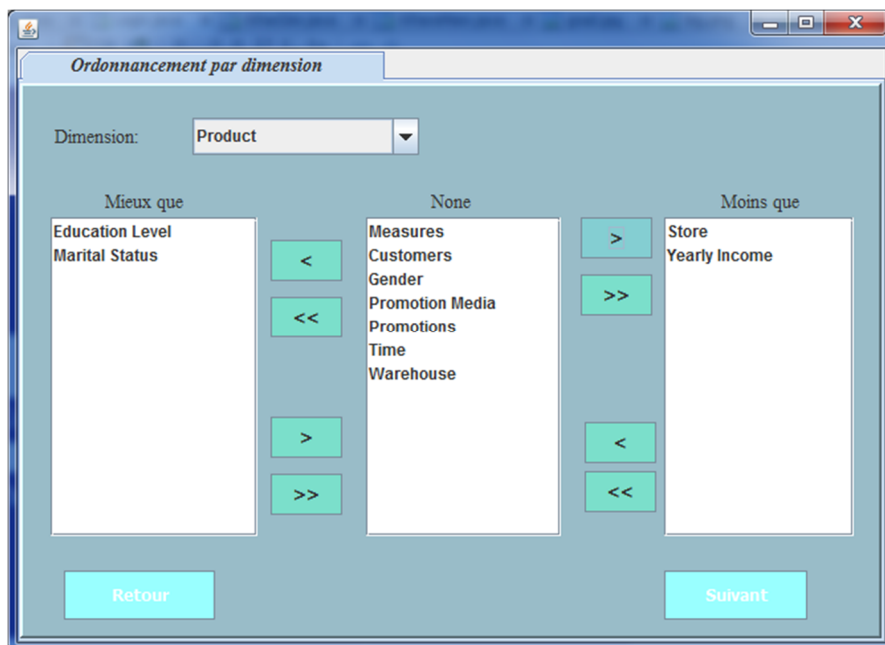
L'interface graphique de l'utilisateur permet à l'utilisateur de construire des requêtes à zéro ou d'utiliser celles de son profil.

A screenshot of a web application window with a dark blue background. The word 'Menu' is centered at the top in a light blue, italicized font. Below it is a white rectangular box containing two buttons: 'Profil' and 'Requête'. At the bottom left of the window is a red button labeled 'Retour'.

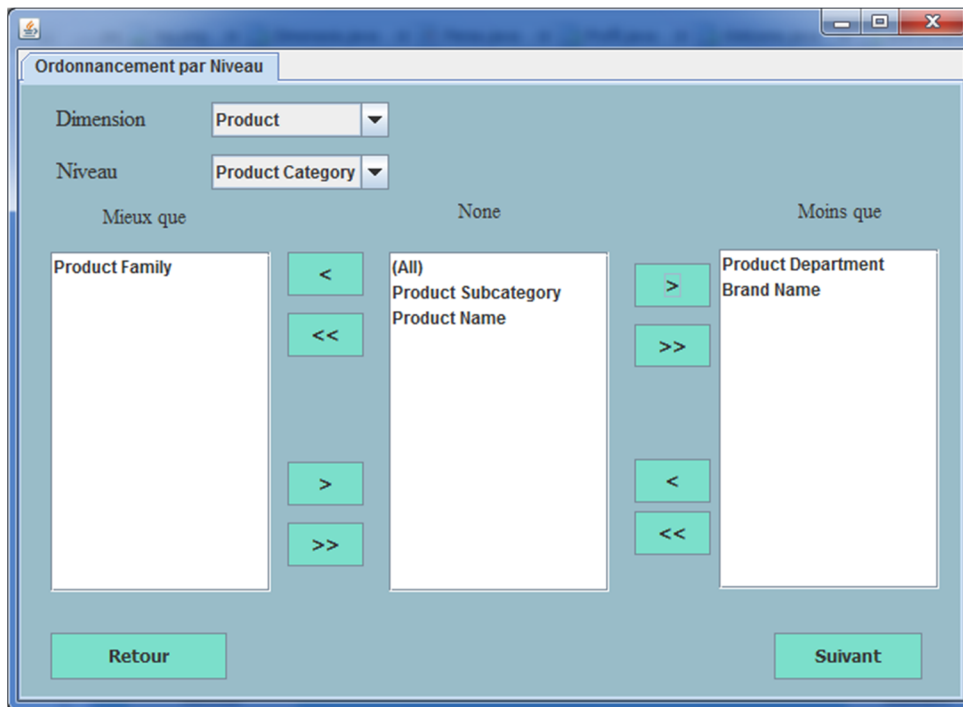
Le menu principal



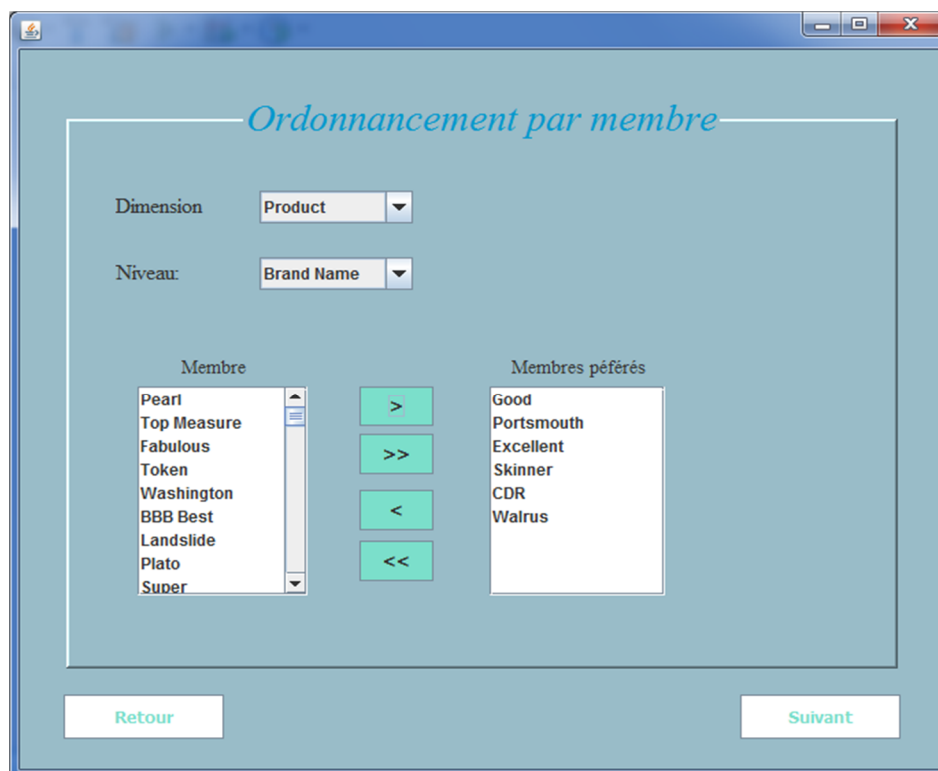
Menu de profile utilisateur pour sélectionner ses préférences



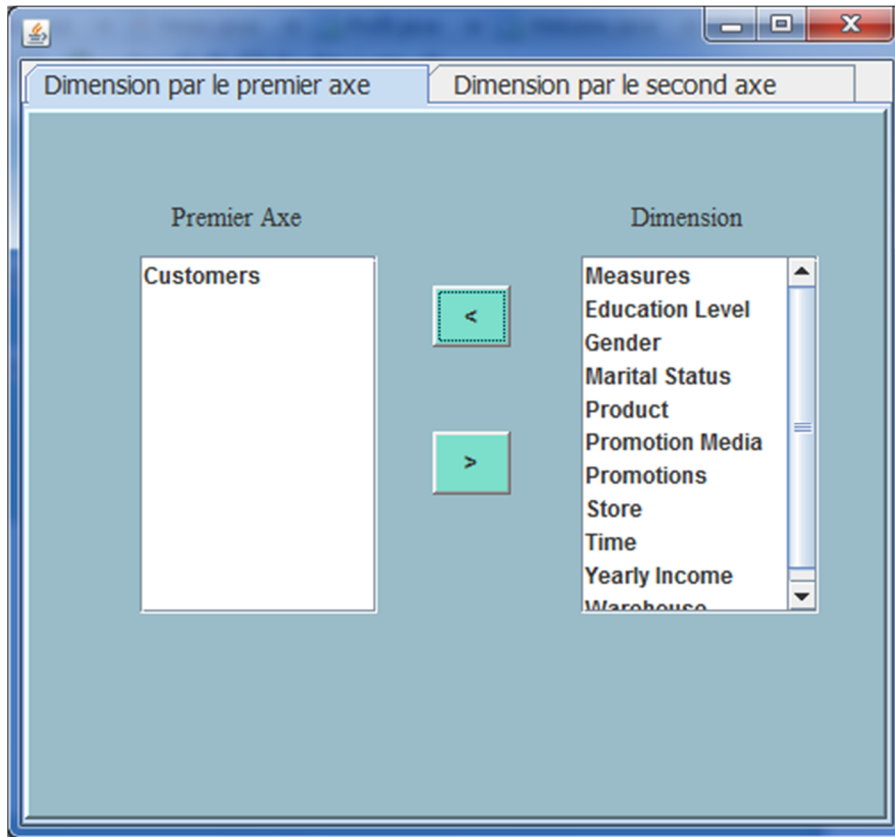
Ordonnement entre les dimensions



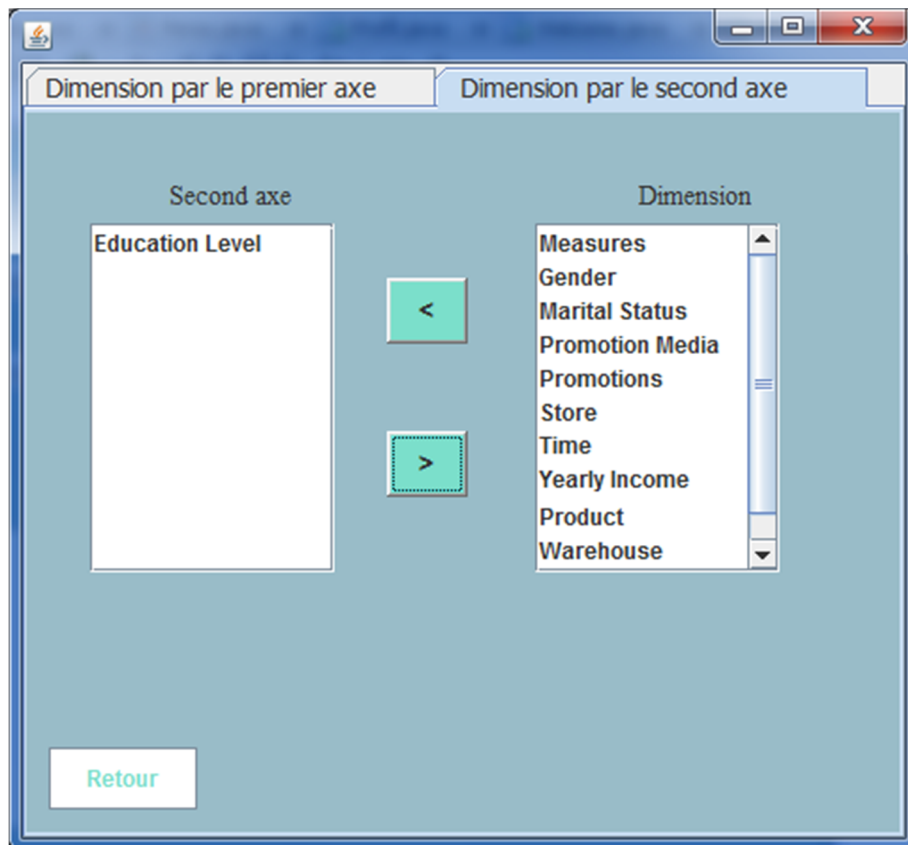
Ordonnement entre les niveaux d'une dimension



Ordonnement entre les membres préférés d'un niveau d'une dimension

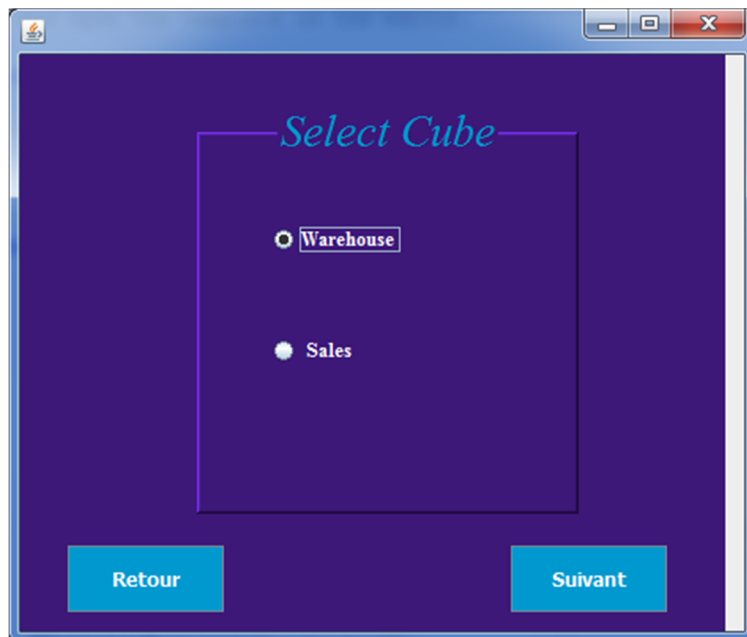


Sélection des dimensions que l'utilisateur souhaite voir sur le premier axe.

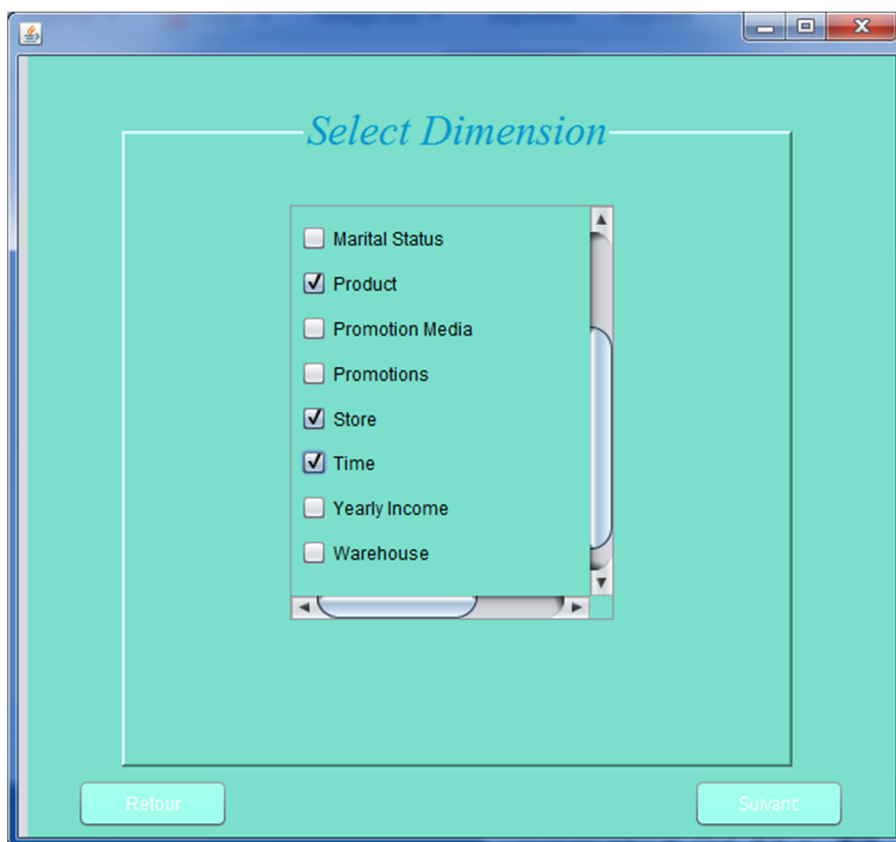


Sélection des dimensions que l'utilisateur souhaite voir sur le second axe.

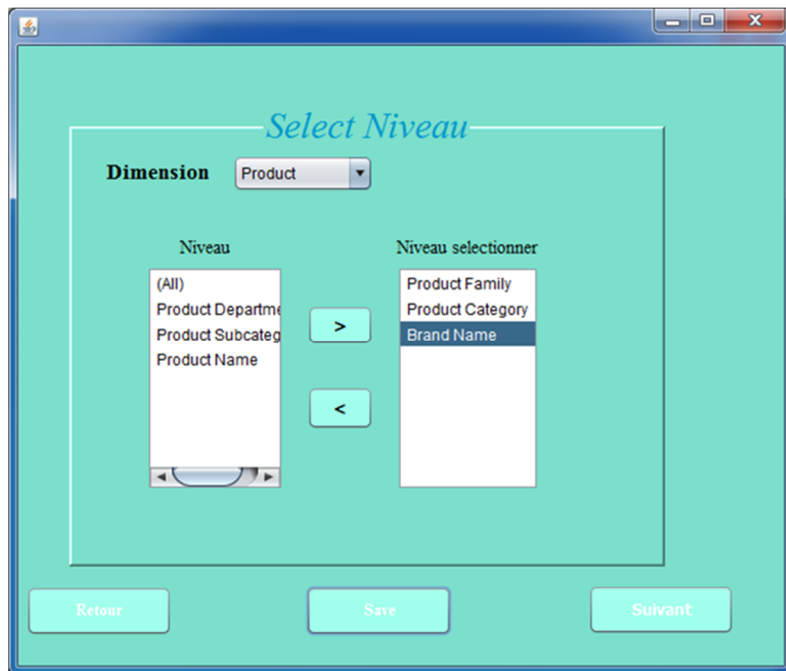
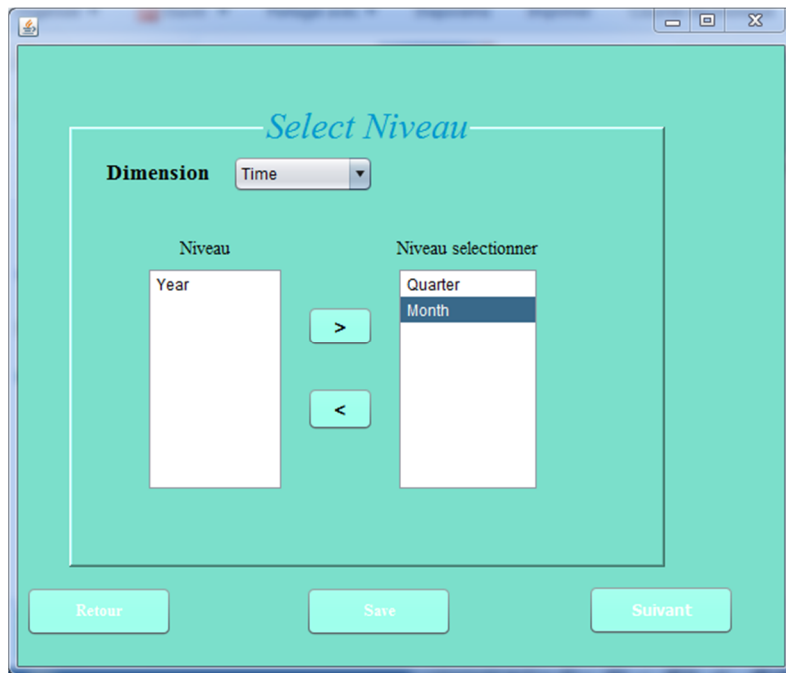
Après la sélection des préférences l'utilisateur doit sélectionner un cube de l'entrepôt de données



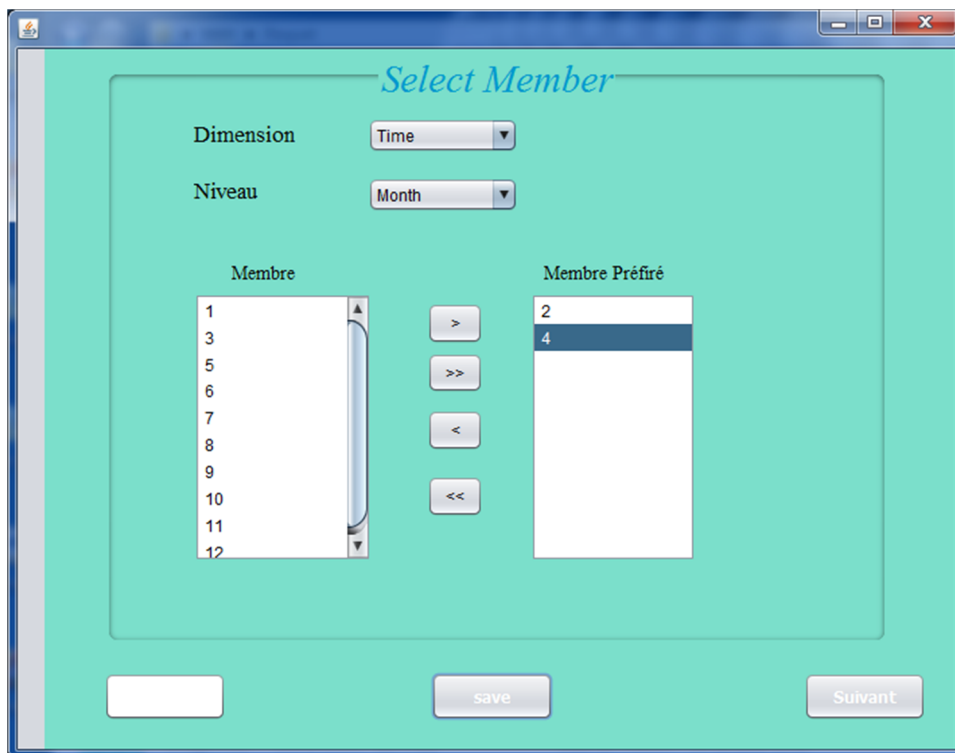
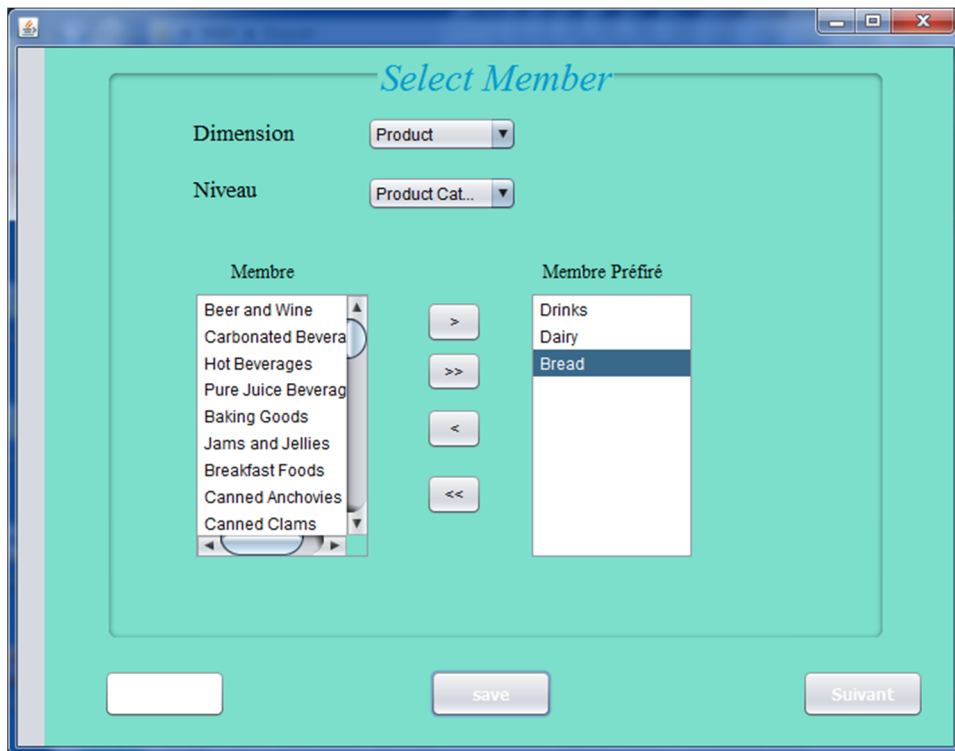
Sélection de cube utilisé pour la requête.



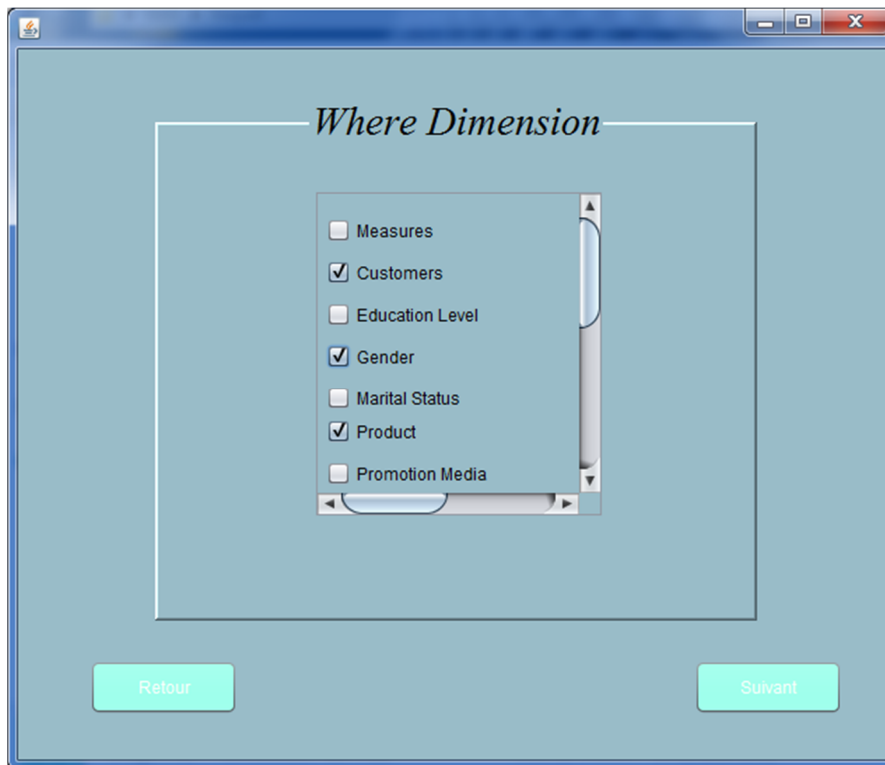
Sélection de l'ensemble des dimensions pour la clause « SELECT ».



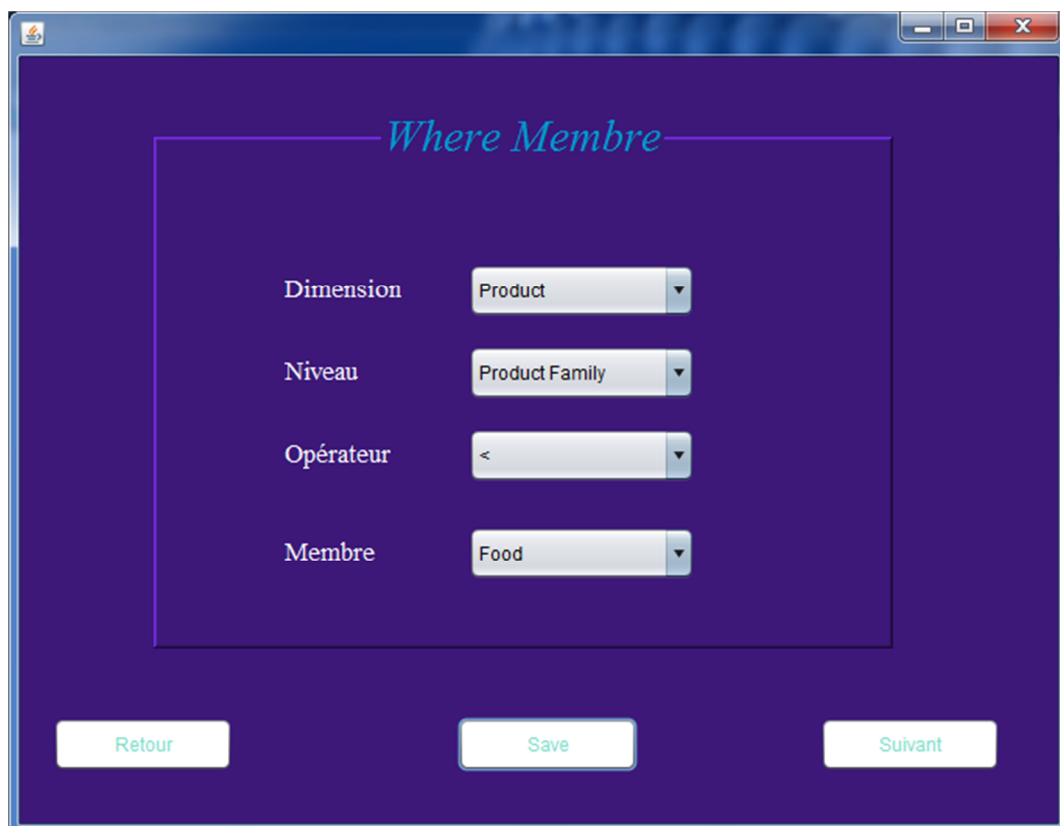
Sélection de l'ensemble des niveaux pour chaque dimension sélectionnée pour la clause « SELECT ».



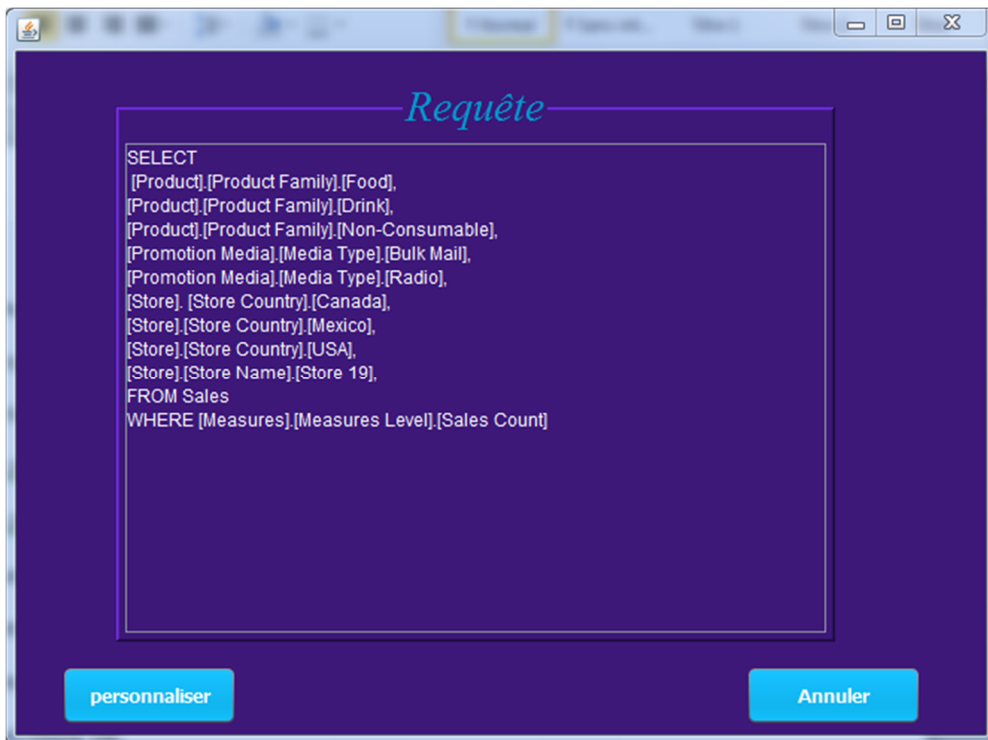
Sélection de l'ensemble des membres pour chaque niveau sélectionné de chaque dimension sélectionnée pour la clause « SELECT ».



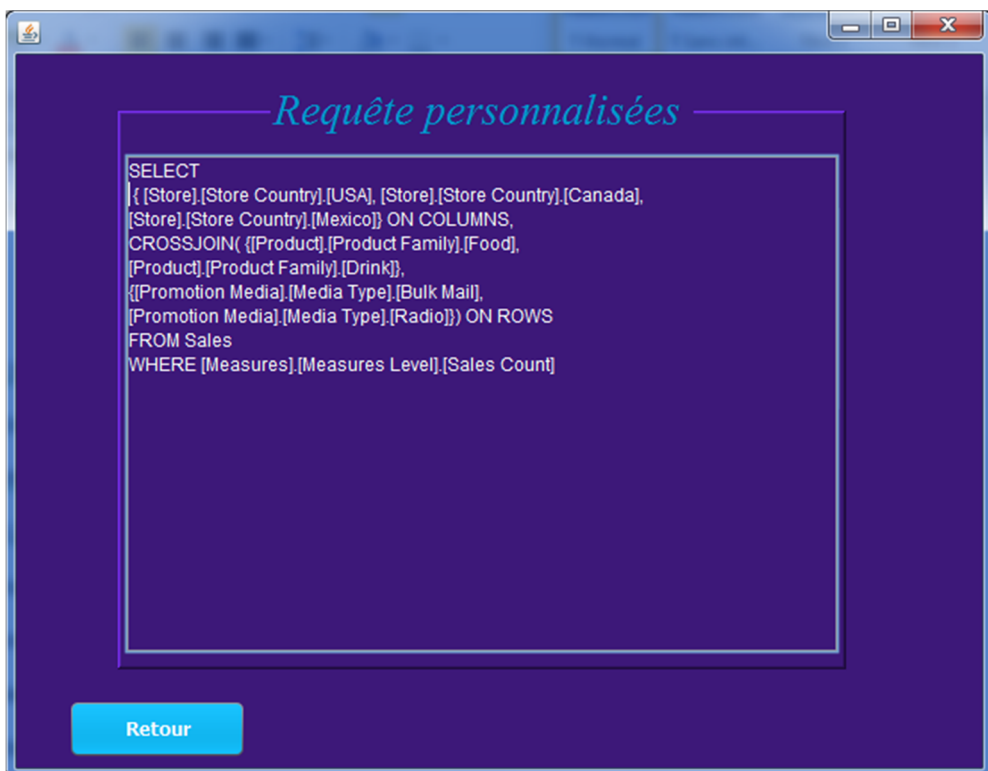
Sélection de l'ensemble des dimensions pour la clause « WHERE ».



Sélectionner pour chaque dimension choisit pour la clause « WHERE », un niveau et un membre pour ce niveau.



Et voilà la requête générée



La requête personnalisée générée

III.5. Exemple 1: Une seule requête, deux profils utilisateur, deux résultats différents :
Soit Souad et Asma deux utilisateurs inscrits sur notre système de personnalisation.

Les préférences des deux utilisateurs ainsi que leurs contraintes de visualisation sont définies comme suit :

Profil de Souad :

L'ordonnancement des dimensions :

Store > Time

Product > Time

Product > Marital Status

Time > Warehouse > Customers > Measures > Gender

L'ordonnancement des niveaux pour chaque dimension :

Customers : State Province > City > Name

Product : Product Category > Product Name > Product Family > Product

Departement > Product > Brand Name > SubCategory.

Store : Store Country > Store Name > Store State > Store City.

Time : Quarter > Year > Month.

Warehouse : Country > Warehouse Name > State Province > City.

L'ordonnancement des membres préférés pour chaque niveau de chaque dimension :

Measures : Measures Level : Sales Count > Store Coste > Store Sales

Customers : Country: USA > Canada > Mexico

Gender : Gender : M > F

Marital Status : Marital Status : M > S

Product :

Product Family : Food > Drink > Non-Consumable

Product Departement : Meat > Eggs > Breakfast Foods>Dairy > SeaFood

Product Category : Meat > Vegetables > Fruit > Bread > Eggs

Promotion Media : Media Type : Product Attachement > TV > Radio

Promotions: Promotion Name : Big Promo > Free For All > Three For One

Store : Store Country : USA > Canada > Mexico

Store Name : Store 19 > Store 20 > Store 9

Time : Year : 1998 > 1997

Quarter : Q3 > Q2 > Q4 > Q1

Month : 9 > 8 > 7 > 6 > 5 > 4 > 10 > 11 > 12 > 3 > 2 > 1

Yearly Income :Yearly Income : \$90K – \$110K > \$70K – \$90K > \$50K – \$70K

Warehouse : Country : USA > Canada > Mexico

Placement des dimensions sur les deux axes : Axe1 = {Product}

Axe2 = {Customers,Time}

Contrainte de visualisation : G = (3,4) // c.-à-d. que l'utilisateur veut voir au maximum trois graduations sur le premier axe et quatre sur le second.

Profil de Asma :

L'ordonnancement des dimensions :

Measures > Gender

Product > Gender

L'ordonnancement des niveaux pour chaque dimension :

Product: Product Family > Product Category

L'ordonnancement des membres préférés pour chaque niveau de chaque dimension :

Measures :

Measures Level : Store Coste > Sales Count > Store Sales > Unit Sales

Gender : Gender : M > F

Product : Product Family : Food > Drink > Non-Consumable

Product Category : Dairy > Bread > Vegetables > Fruit

Placement des dimensions sur les deux axes : Axe1 = { }

Axe2 = { }

Contrainte de visualisation : G = (2,3) // c.-à-d. que l'utilisateur veut voir au maximum deux graduations sur le premier axe et trois sur le second.

Requête :

Soit la requête MDX suivante que les deux utilisateurs iront exécuter :

```
SELECT [Measures].[Measures Level].Store Cost,
[Measures].[Measures Level].Store Sales,
[Measures].[Measures Level].Unit Sales,
[Gender].[Gender].F,
[Gender].[Gender].M,
[Product].[Product Family].Food,
[Product].[Product Family].Drink,
[Product].[Product Family].Non-Consumable,
[Product].[Product Category].Dairy,
[Product].[Product Category].Bread,
[Product].[Product Category].Meat
FROM Sales
```

WHERE [Time].[Year].1997

D'après les deux profils utilisateur, on remarque que les dimensions de cette requête sont toutes ordonnées par l'utilisateur Souad tandis que l'utilisateur Asma n'as pas un choix précis entre les deux dimensions Product et Measures de la requête.

Les niveaux des dimensions de la requête sont tous ordonnés par les deux utilisateurs. Et ce qui concerne les membres employés par la requête, on remarque que l'ordonnancement est presque totale pour les deux utilisateurs sauf que l'utilisateur Asma n'a pas de choix pour le membre Meat et la même chose pour l'utilisateur Souad par rapport le membre Bread.

On passe maintenant à l'exécution de la requête par les deux utilisateurs :

Souad :

```
SELECT { [Product].[Product Category].[Dairy],
        [Product].[Product Category].[Meat],
        [Product].[Product Category].[Bread]} ON COLUMNS,
CROSSJOIN( {[Measures].[Measures Level].[Unit Sales],
            [Measures].[Measures Level].[Store Cost]},
            {[Gender].[Gender].[M], [Gender].[Gender].[F]}
)ON ROWS
FROM Sales
WHERE [Time].[Year].[1997]
```

Asma :

```
SELECT CROSSJOIN({[Gender].[Gender].[M], [Gender].[Gender].[F]},
                { [Product].[Product Family].[Food]})ON COLUMNS,
                {[Measures].[Measures Level].[Store Cost],
                [Measures].[Measures Level].[Store Sles],
                [Measures].[Measures Level].[Unit Sales]}
ON ROWS
FROM Sales
WHERE [Time].[Year].[1997]
```

III.5.1. Discussion des résultats

On peut remarquer que les deux utilisateurs ont interrogé le système afin d'obtenir des réponses à leurs requêtes qui était identiques au départ, mais au cours des différentes phases du processus de personnalisation, les deux utilisateurs ont eu des ensembles de références et des structures différentes qui ont mené finalement à des réponses différentes.

Pour ceux qui n'ont pas une idée claire sur le concept de personnalisation vont sûrement imaginer qu'il y a une anomalie dans le système qui mène à ces réponses différentes. Mais il faut savoir que c'est l'une des principales caractéristiques d'un système de personnalisation basé sur l'utilisation des profils utilisateur. Puisque c'est le contenu (préférences) de ces derniers qui détermine les éléments qui doivent apparaître dans le résultat et même la forme que ce dernier doit prendre dans l'affichage.

III.6. Exemple 2 : Discrimination entre les sous-ensembles de références

Pour cet exemple, et pour certaines raisons d'analyse, l'utilisateur 1 formule la requête suivante :

```
SELECT [Product]. [Product Family].[Food],  
[Product]. [Product Family].[Drink],  
[Product]. [Product Family].[Non-Consumable],  
[Promotion Media]. [Media Type].[Bulk Mail],  
[Promotion Media]. [Media Type].[Radio],  
[Store]. [Store Country]. [Canada],  
[Store]. [Store Country].[Mexico],  
[Store]. [Store Country].[USA],  
[Store]. [Store Name].[Store 19],  
FROM Sales  
WHERE [Measures]. [Measures Level].[Sales Count]
```

Lors de l'exécution de la requête, le système renvoie deux sous-ensembles de références due au soutien de l'ordre partiel (l'utilisateur n'a pas de préférences entre quelques dimensions parmi celles utilisées dans la requête). Les deux sous-ensembles de références sont :

(Avec comme niveaux (Levels) : "Product Family", "Media Type", "Store Country" pour respectivement les dimensions Product, Promotion Media et Store)

R0	R1
Food, Bulk Mail, USA	Food, Bulk Mail, USA
Drink, Bulk Mail, USA	Drink, Bulk Mail, USA
Food, Bulk Mail, Canada	Food, Bulk Mail, Canada
Food, Radio, USA	Food, Radio, USA
Drink, Bulk Mail, Canada	Drink, Bulk Mail, Canada
Food, Bulk Mail, Mexico	Non-Consumable, Bulk Mail, USA
Drink, Radio, USA	Drink, Radio, USA
Food, Radio, Canada	Food, Radio, Canada
Drink, Bulk Mail, Mexico	Drink, Radio, Canada
Drink, Radio, Canada	Non-Consumable, Bulk Mail, Canada
Food, Radio, Mexico	Non-Consumable, Radio, USA
Drink, Radio, Mexico	Non-Consumable, Radio, Canada

Pour l'ensemble R0 :

Structure : {Store} AND {Product, Promotion Media}

Requête :

```
SELECT {[Store]. [Store Country].[USA], [Store].[Store Country].[Canada],
[Store]. [Store Country].[Mexico]} ON COLUMNS,
CROSSJOIN( {[Product]. [Product Family].[Food],
[Product]. [Product Family].[Drink]},
{[Promotion Media]. [Media Type].[Bulk Mail],
[Promotion Media]. [Media Type].[Radio]}) ON ROWS
FROM Sales
WHERE [Measures]. [Measures Level].[Sales Count].
```

Pour l'ensemble R1 :

Structure : {Product} AND {Promotion Media, Store}

Requête :

```
SELECT {[Product]. [Product Family].[Food],
[Product].[Product Family].[Drink],
[Product].[Product Family].[Non-Consumable] } ON COLUMNS,
CROSSJOIN ( {[Promotion Media].[Media Type].[Bulk Mail],
[Promotion Media].[Media Type].[Radio] },
{[Store]. [Store Country]. [USA],
```

[Store]. [Store Country]. [Canada]]) ON ROWS

FROM Sales

WHERE [Measures]. [Measures Level].[Sales Count]

Conclusion

Ce chapitre été un champ d'essai pour donner un aperçu pratique pour le prototype développé ainsi que son fonctionnement de base. Les exemples traités dans ce chapitre et le but de notre system.

Conclusion Général

Conclusion générale

Notre travail se situe dans le cadre des systèmes décisionnels reposant sur une modélisation multidimensionnelle des données. Plus précisément, nous avons défini des concepts et des mécanismes permettant de personnaliser l'information de données multidimensionnelles.

À travers ce travail, on a essayé d'éclairer la notion de personnalisation, ses enjeux dans les diverses disciplines et surtout d'enlever certaines ambiguïtés qui entourent ce concept dans les entrepôts de données.

D'après cette étude, on a constaté que la personnalisation dans les entrepôts de données est conçue pour supporter des requêtes complexes de décision. Elle est basée sur l'utilisation de profil utilisateur pour lui donner une réponse appropriée à ses préférences. Le principe est lorsqu'un utilisateur interroge un entrepôt de données par une requête OLAP, ses préférences stockées dans son profil doivent être prises en considération.

Dans notre travail, nous avons étudié dans un premier temps le profil utilisateur, car ce dernier constitue le noyau de la personnalisation, puis nous avons étudié la personnalisation dans les bases de données multidimensionnelle, qui est le domaine le plus proche à celui des entrepôts de données. Ensuite nous avons présenté brièvement les différents travaux de personnalisation sur les entrepôts de données. Et comme l'objectif de notre travail est la «personnalisation de visualisation», nous nous sommes concentrés sur le travail de [Bel et al,2006], la personnalisation de visualisation, méthode permettant de personnaliser les visualisations obtenues en réponse à des requêtes OLAP, sur la base de profil utilisateur.

Cette méthode repose sur l'ordonnement de références de cube générées par la requête utilisateur. Cet ordre est basé sur les préférences utilisateur contenues dans son profil.

Références

- [Abbar, S., Bouzeghoub, M., Kostadinov, D., Lopes, S., Aghazaryan, A., Betge-Brezetz S. 2008]. Personalized Access Model : Concepts and Services for Content Delivery Platform. Journées Bases de Données Avancées (BDA'08).
- [Barbera et al., 2004] Barbera, S., Bossert, W., and Pattanaik, P. (2004). *Ranking Sets of Objects*, volume II Extensions. Kluwer Academic.
- [Bazzalicza et Naim, 2001] M. Bazzalicza et P. Naim. *Data Mining pour le Web*. Edition Eyrolles, Paris, 2001.
- [Belkin et Croft 1992] N. J. Belkin, W. B. Croft. Information filtering and information retrieval : two sides of the same coin ? *Communications of the ACM*, vol. 35, n° 12, p. 29-38, Décembre 1992.
- [Bellatreche et al. 2005], A personalization framework for OLAP queries, 8th International Workshop on Data Warehousing and OLAP, DOLAP'05, Bremen, Germany, pp. 9-18
- [Bouzeghoub M., Kostadinov D. 2005], Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils, *2ème conférence francophone en Recherche d'Information et Applications CORIA '05*, Grenoble, 5-11 Mars 2005, pp. 201-218.
- [Bradley et al., 2000] K. Bradley, R Rafter, B. Smyth, Case-Based User Profiling for Content
- [Chaudhuri and Dayal, 1997] Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1) :65–74.
- [Chevalier M. et al] Collaborative and Social Information Retrieval and Access : Techniques for Improved User Modeling », IGI Global ,2008.
- [Chomicki, 2002]. Querying with Intrinsic preferences. In *EDBT*, pages 34–51.
- [Chomicki, 2003]. Preference formulas in relational queries. *ACM Trans. Database Sys.*, 28(4) :427–466.
- [F. Bentayeb et al] personnalisation dans les entrepôts de données : bilan et perspectives
- [F.Ravat, O. Teste, G. Zurfluh, 2010]. Personnalisation des bases de données multidimensionnelle
- [Fink et Kobsa, 2000] J. Fink et A. Kobsa. A Review and Analysis of Commercial User
- [Frédérique, Odile,2007] Analyses multidimensionnelles de contenus documentaires dans un ENT universitaire au service de l'acteur enseignant-chercheur. Gartner Group, pp. 1, 05/2001.
- In Proceedings of the 11th IEEE International Conference on Tools with Artificial
- [Inmon, W. H. 1996] Building the Data Warehouse. John Wiley & Sons. Intelligence, pp. 391-398, Chicago, November 1999.

- [**Jagadish et al., 1999**] Jagadish, H.V., Lakshmanan, L.V. S., and Srivastava, D. (1999). What can hierarchies do for data warehouses ? In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 530–541.
- [**Jameson, 1999**] Jameson A., User Adaptive Systems An integrated Overview. Tutorial presented at the 7th International Conference on User Modeling, June 20-24, 1999.
- [**Janowski et Sarner, 2001**] W. Janowski, A. Sarner. Five Opportunities for Personalization.
- [**Jerbi, Ravat, 2010**] Personnalisation du contenu des bases de données multidimensionnelles Article · June 2010
- [**Kießling, W. 2002**]. Foundations of preferences in Database Systems. VLDB, pp. 311–322.
- [**Kimball, 1996**] Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley & Sons.
- [**Korfhage R. 1997**]. *Information Storage and retrieval*, Wiley Computer Publishing, ISBN 0-471-14-338.
- [**Kostadinov, 2003**] D. Kostadinov. Personnalisation de l'information et gestion des profils utilisateurs, Rapport de DEA, Université de Versailles, France, 2003.
- [**Kostadinov, 2007**]. Personnalisation de l'information : une approche de gestion du profil et de reformulation de requêtes. Thèse de Doctorat, L'université de Varseilles.
- [**Koutrika, G., Ioannidis, Y. E. 2005**]. Personalized Queries under a Generalized Preference Model. ICDE, pp. 841-852.
- Modeling Servers for Personalization on the World Wide Web, International Journal of Personalisation, Lecture Notes in Computer Science, 2000.
- [**Pretschner et Gauch, 1999**] A. Pretschner et S. Gauch. Ontology Based Personalized Search,
- [**Rizzi, S. 2007**] OLAP Preferences : a Research Agenda. In DOLAP 07, pp. 99–100.
- [**Rossi et al., 2001**] G. Rossi, D. Schwabe et R. Guimarães. Designing Personalized Web Applications, 10th International World Wide Web Conference (WWW10), May 1-5, Hong Kong, 2001.