



Mémoire de fin d'études

Pour l'obtention du diplôme de Master en
Informatique

**Option : Ingénierie des Systèmes d'Information
Avancés**

Thème

**Prise en compte du contexte de décision
pour la recommandation des requêtes**

Réalisé par :

Mr. METCHAT Lakhdar

Mr. HANDAOUI Mohamed

Encadré par :

Mme Hammouche Djamila

Promotion: 2015/2016

Dédicaces

Je dédie ce modeste travail :

A la mémoire de Mes très chers parents,

Aucun hommage ne pourrait être à la hauteur de l'amour et de respect que je les porte

A ma très chère femme

*Aucune dédicace, aucun mot ne saurait exprimer tout le respect et tout l'amour que je
te dois*

A mes enfants Amani et Mohamed Amine

A mon frère Foucef et sa famille

A ma sœur et sa famille

A toute ma grande famille

Aux parents de ma femme et son frère Zohir et sa sœur et à toute sa grande famille

A mon binôme Mohamed

Pour tout ce qu'on a partagé

A tous mes amis et collègues de travail sans exception

A toute ma promotion

A toutes les personnes que je connais et que je n'ai pas citées

METCHAT Lakhdar

Dédicaces

Je dédie ce modeste travail :

A la mémoire de Ma très chère mère

A mon très cher père,

Aucun hommage ne pourrait être à la hauteur de l'amour et de respect que je les porte

A ma très chère femme

Aucune dédicace, aucun mot ne saurait exprimer tout le respect et tout l'amour que je te dois

A toute ma grande famille

Aux parents de ma femme et sa grande famille

A mon binôme Lakhdar

Pour tout ce qu'on a partagé

A tous mes amis et collègues de travail sans exception

A toute ma promotion

A toutes les personnes que je connais et que je n'ai pas citées

HANDAOUI Mohamed

Remerciements

Merci au bon dieu avant tout,

Tout le grand remerciement à notre encadreur Mme HAMMOUCHE de nous fait confiance et accepter de nous attribuer ce sujet et pour son soutien moral et technique et ses précieux conseils. De simples mots ne sont pas assez fort pour décrire toute notre gratitude envers elle.

Nos remerciements vont également à l'ensemble des employés de la direction de l'éducation de la wilaya de Ain Defla et surtout le personnel du service informatique de nous avoir fourni leurs données sur les résultats du BAC afin de pouvoir réaliser ce projet.

Nous exprimons aussi notre gratitude aux membres de jury pour avoir accepté d'examiner notre travail.

Notre remerciement va en particulier à Mr BOUGHEDDA Ahmed, Ingénieur d'état en informatique et consultant freelance BI pour nous avoir aidé à surmonter les difficultés durant toute la durée de notre projet.

Nous tenons aussi à remercier tous les enseignants du département informatique pour leurs efforts et le savoir qu'ils ont su transmettre tout au long de notre cursus universitaire.

Enfin, nous exprimons notre grande gratitude à toutes les personnes qui ont participé de près ou de loin à la réalisation de ce projet.

Résumé

Les décideurs dans le domaine d'éducation en Algérie et dans la wilaya de Ain Defla en particulier où nous avons fait notre étude pratique ont besoin de connaître et d'analyser chaque fin d'année scolaire après l'achèvement de l'opération du Baccalauréat national le taux de succès selon différents axes d'analyses, en particulier par Lycée, afin de pouvoir prendre des décisions stratégiques et opérationnelles nécessaires pour améliorer ce taux de succès pour l'année prochaine.

Les logiciels existants de genre ERP de gestion, de scolarité par exemple sont incapables à satisfaire les besoins des décideurs pour leur offrir des analyses de données fiables, efficaces et à des moments déterminés relatives au taux de succès du BAC.

Ainsi, se résumant les principales difficultés rencontrées :

- La tâche d'analyse est fastidieuse et trop difficile ;
- La procédure de reporting et d'analyse des résultats du BAC n'est pas performante et elle est manuelle ;
- Les données relatives à l'analyse des différentes bases de production sont volumineuses et non exploitées par la meilleure façon vu la lenteur des requêtes statiques.

Notre travail sert à relever ces lacunes et difficultés en offrant une solution BI qui permettra aux décideurs d'avoir une vision réelle, fiable, efficace en temps opportun et leur recommander des résultats d'analyse exploitables et satisfaisantes qui ne doivent en aucun cas être un ensemble vide ou un ensemble volumineux selon leurs préférences personnelles. En effet, notre projet consiste à réaliser un entrepôt de données flou avec une partie de restitution des résultats d'analyse pour générer des états et des tableaux croisés dynamiques.

Mots clés : Systèmes décisionnels, entrepôt de données, entrepôt de données flou, système de recommandation des requêtes, Mondrian, langage MDX.

Abstrat

The decisions-makers in the education field in Algeria and in the wilaya of Ain Defla in particular that we did our practical study need to know and analyze each end of the school year after the completion of the National Baccalaureate transaction the success rate according to different analysis axes, particularly High School, to make strategic and operational decisions necessary to improve the success rate for next year.

Existing software ERP kind of example schooling management are unable to meet the needs of decision makers for their offer reliable data analyzes, effective and at specific times relating to BAC success rate.

Thus, summarizes the main challenges as follows:

- The analyze task is tedious and difficult;
- The reporting process and analyze the results of the BAC is not performing and it is manual;
- Data for the analysis of different production bases are large and untapped by how best the slow pace static queries.

Our job is to address these gaps and challenges by offering a BI solution that will enable decision makers to have a real vision, reliable, efficient, timely and recommend actionable results and satisfactory analyzes that must in no case be empty set or a large set according to their personal preferences. Indeed, our project is to achieve a fuzzy data warehouse with a refund of part of the analysis results to generate reports and pivot tables.

Keywords: decision-making systems, data warehouse, fuzzy data warehouse, recommendation system queries, Mondrian, MDX language.

تلخيص

في مجال التعليم في الجزائر وفي ولاية عين الدفلى على وجه الخصوص أين قمنا بتطبيقية بحاجة إلى معرفة وتحليل كل نهاية العام الدراسي بعد الانتهاء من عملية البكالوريا نسبة النجاح الوطنية وفقا لمحاور التحليل المختلفة، وخاصة ثانوية، لاتخاذ القرارات الاستراتيجية والتشغيلية اللازمة لتحسين معدل النجاح في العام المقبل.

برمجيات تسيير إدارة التعليم سبيل المثال غير قادرة على تلبية احتياجات صانعي القرار لمساعدتهم من أجل الحصول على تحليل بيانات موثوقة وفعالة وفي أوقات

وهكذا، لخص التحديات الرئيسية على النحو التالي:

- إن مهمة التحليل
- عملية إعداد التقارير وتحليل BAC غير دقيقة ويدوية.
- البيانات ثي وغير مستغلة جيدا في عملية تحليل تطبيق استخراج البيانات.
- مهمتنا معالجة هذه الثغرات والتحديات من خلال تقديم حل استقصاء المعلومات التي من شأنها تمكين صناع القرار ن يكون له رؤية حقيقية وموثوقة وفعالة وفي الوقت المناسب ويساعد في عملية توصي للتنفيذ وتحليلات مرضية التي يجب أن تكون في أي حال من الأحوال مجموعة فارغة أو مجموعة كبيرة وفقا لتفضيلاتهم الشخصية. في الواقع، مشروعنا يسعى لصناعة مستودع بيانات استرداد نتائج التحليل تقارير وجدول محورية.

: أنظمة صنع القرار، مستودع البيانات، مستودع البيانات نظام توصية

.MDX Mondrian

TABLE DE MATIERES

Résumé	5
Abstrat.....	6
تلخيص.....	7
Liste des Abréviations.....	12
Listes de Figures.....	13
Listes de Tableaux.....	14
Introduction Générale	16
1. Contexte du Projet.....	16
2. Problématique	17
3. Objectifs et Contributions	18
4. Organisation du Mémoire	19
1. Systèmes d'Information Décisionnels.....	22
1.1. Introduction.....	22
1.2. Définition d'un Système d'Information Décisionnel.....	22
1.3. Architecture des systèmes décisionnels	23
1.3.1. Les sources des données.....	24
1.3.2. L'entrepôt de données.....	25
1.3.3. Serveur d'analyse ou serveur OLAP.....	25
1.3.4. Générateur d'état ou le Reporting.....	25
1.4. Typologie des systèmes décisionnels	26
1.5. Conclusion.....	26
2. Entrepôt de Données	28
2.1. Introduction.....	28
2.2. Définition d'un Entrepôt de données.....	28
2.3. Fonctionnalités d'un Entrepôt des données.....	28
2.4. Caractéristiques d'un Entrepôt des données.....	28
2.5. Les Datamart	29
2.6. Modélisation multidimensionnelle.....	29

2.6.1.	<i>Niveau Conceptuel</i>	30
2.6.2.	<i>Niveau Logique</i>	31
2.6.3.	<i>Niveau Physique</i>	32
2.7.	Etapes de réalisation d'un entrepôt des données	33
2.7.1.	<i>Conception</i>	33
2.7.2.	<i>Acquisition des données</i>	34
2.7.3.	<i>Aspects techniques</i>	34
2.7.4.	<i>Restitution</i>	34
2.7.5.	<i>Administration et maintenance</i>	35
2.8.	Manipulation d'un entrepôt des données	35
2.8.1.	<i>Opérations de forage</i>	35
2.8.2.	<i>Opérations de Sélection/Projection</i>	35
2.8.3.	<i>Opérations de Restructuration/Réorientation</i>	36
2.9.	Conclusion	36
3.	Logique Floue	38
3.1.	Introduction	38
3.2.	Définition de la logique floue	38
3.3.	Les sous-ensembles flous.....	38
3.4.	Les variables linguistiques	40
3.5.	Les opérateurs flous	41
3.6.	Le raisonnement en logique floue	42
3.7.	Entrepôt de Données Flou	43
3.7.1.	<i>Concept du Data Warehouse Flou</i>	43
3.7.2.	<i>Définition de base et les méta tables floues</i>	43
3.7.3.	<i>Modèle d'un entrepôt de données flou</i>	46
3.8.	Conclusion	46
4.	La Recommandation des Requêtes	48
4.1.	Introduction	48
4.2.	Définition de la recommandation des requêtes	49
4.3.	Types des recommandations des requêtes	50
4.3.1.	<i>La personnalisation</i>	50
4.3.2.	<i>La recommandation</i>	50

4.3.2.1.	<i>Recommandation Cognitive ou basée sur le contenu</i>	51
4.3.2.2.	<i>Recommandation Collaborative</i>	52
4.3.3.3.	<i>Recommandation Hybride</i>	52
4.4.	Conclusion.....	53
5.	Travaux Connexes	55
5.1.	Introduction.....	55
5.2.	Travaux de Bellatreche [Bellatreche, 2005&2006].....	55
5.3.	Travaux de Ravat [Ravat, 2007&2008].....	56
5.4.	Travaux de Giacometti [Giacometti, 2008&2009].....	57
5.5.	Travaux de Golfarelli [Golfarelli, 2009].....	58
5.6.	Travaux de Garrigós [Garrigos, 2009].....	59
5.7.	Travaux de H. Jerbi [Jerbi, 2012].....	60
5.8.	Travaux de R.Khemiri [Khemiri, 2015].....	60
5.9.	Synthèse des travaux de recommandation des requêtes.....	61
5.10.	Tableau Comparatif des travaux existants.....	62
5.11.	Conclusion.....	63
6.	Conception	66
6.1.	Introduction.....	66
6.2.	Interrogation des Requêtes.....	67
6.2.1.	<i>Langage d'Interrogation Flou</i>	67
6.2.2.	<i>Modélisation d'un terme flou</i>	69
6.2.3.	<i>Requêtes MDX Floues</i>	69
6.3.	Serveur OLAP MONDRIAN.....	70
6.3.1.	<i>Introduction</i>	70
6.3.2.	<i>Architecture de Mondrian</i>	70
6.3.3.	<i>Pentaho Schema Workbench</i>	72
6.3.4.	<i>Interface de requêtage web (JPivot)</i>	73
6.3.5.	<i>Installation de Mondrian</i>	73
6.4.	Le Langage MDX.....	76
6.4.1.	<i>Définition</i>	76
6.4.2.	<i>Les concepts</i>	77
6.4.3.	<i>Construction d'une requête MDX</i>	77

6.4.4.	<i>Fonctions</i>	78
6.4.5.	<i>Exemple de fonctions avancées MDX</i>	80
6.5.	Conclusion.....	80
7.	Réalisation & Implémentation	82
7.1.	Introduction.....	82
7.2.	Modèle du Data Warehouse BAC.....	82
7.3.	Présentation de la solution.....	84
7.4.	Architecture Logicielle Utilisée.....	89
7.5.	Fonctionnement Général.....	89
7.6.	Présentation des résultats.....	92
7.6.1.	<i>ROLAP MDX QUERY : Taux de succès BAC /Lycée</i>	94
7.6.2.	<i>FUZZY MDX QUERY : Taux de succès du BAC Elevé/Lyce</i>	95
7.6.3.	<i>FUZZY MDX QUERY : Taux de succès du BAC Moyen/Lyce</i>	96
7.6.4.	<i>FUZZY MDX QUERY : Taux de succès du BAC Faible/Lyce</i>	97
7.6.5.	<i>RECOMMANDED MDX QUERY Avec Profil : ADMIN</i>	98
7.6.6.	<i>RECOMMANDED MDX QUERY Avec Profil : Directeur WILAYA 44</i>	99
7.6.7.	<i>RECOMMANDED MDX QUERY Avec Profil : Directeur LYCEE 2444</i>	100
7.7.	Indicateurs de performance du système.....	101
7.7.1.	<i>Test Benchmark (MDX Query VS SQL Query)</i>	101
	Conclusion & Perspectives	105
	Références Bibliographiques	108
1.	Bibliographie.....	108
2.	Webographie.....	112

Liste des Abréviations

BDD : Base De Données

BI: Business Intelligence

DM: Data Mart

DW: Data Warehouse

FDW: Fuzzy Data Warehouse

ETL: Extract Transform Load

HOLAP: Hybrid On Line Analytical Processing

MOLAP: Multidimensional On Line Analytical Processing

OLAP: On Line Analytical Processing

ROLAP: Relational On Line Analytical Processing

SGBDR : Système de Gestion de Base de Données Relationnel

SID : Système d'Information Décisionnel

SQL : Structured Query Language

MDX : MultiDimensionnal language eXpression

Listes de Figures

Figure 1: Architecture des systèmes décisionnels	23
Figure 2: Les Datamart.....	29
Figure 3: Représent graph d'un ensemble classique et d'un ensemble Flou	39
Figure 4: Le processus de prise de compte de l'utilisateur	53
Figure 5: Architecture BI	71
Figure 6: la page principale mondrian	74
Figure 7: Résultat de requête test.....	76
Figure 8: Modèle du DW BAC.....	83
Figure 9 : Algorithme de recommandation à base de profil utilisateur	88
Figure 10: Architecture de la solution utilisée.....	91
Figure 11: Boite d'identification.....	92
Figure 12: Le menu principal de logiciel « BI Analysis for BAC Succes »	93
Figure 13: ROLAP MDX QUERY : Taux de succès BAC /Lycée.....	94
Figure 14: FUZZY MDX QUERY : Taux de succès du BAC Elevé/Lyce.....	95
Figure 15: FUZZY MDX QUERY : Taux de succès du BAC Moyen/Lyce	96
Figure 16: FUZZY MDX QUERY : Taux de succès du BAC Faible/Lyce	97
Figure 17: RECOMMANDED MDX QUERY Avec Profil : ADMIN	98
Figure 18: RECOMMANDED MDX QUERY Avec Profil : Dir Wilaya 44.....	99
Figure 19: RECOMMANDED MDX QUERY Avec Profil : Dir Lyc 2444....	100
Figure 20 : Test de Benchmarking (MDX Query VS SQL Query).....	103

Listes de Tableaux

Tableau 1: Les opérateurs flous	41
Tableau 2: Définitions de l'implication floue	42
Tableau 3: Tableau Comparatif des travaux existants	62

Introduction Générale

"The future is not inevitable. We can influence it, if we know what we want it to be".

-Charles Handy-

Introduction Générale

1. Contexte du Projet

Avec le développement des technologies de l'information, les entreprises ou les administrations produisent de très grands volumes de données à travers leurs systèmes opérationnels. Leurs données doivent être manipulées et exploitées dans le but de visualiser l'activité de l'administration et plus particulièrement visualiser la rentabilité de ses affaires.

En revanche, Il s'avère très complexe de prendre les bonnes décisions aux bons moments afin de prévoir les changements et de s'adapter à un contexte décisionnel. D'où la nécessité de mettre en place un système décisionnel performant.

L'arrivée de la notion de Business Intelligence (informatique décisionnelle) vient nous répondre au besoin d'avoir une vision globale et optimiser le patrimoine informationnel de l'entreprise, et elle est utilisée dans divers domaines comme l'éducation qui est le domaine de notre cas d'étude.

Afin de mieux exploiter les outils offerts par la BI, les entrepôts de données servent non pas à faire des analyses globales dans un gisement de données mais de prendre en compte de l'utilisateur dans ses différentes navigations multidimensionnelles. En se basant sur ce constat, l'utilité des préférences utilisateurs sont primordiales dans l'analyse et l'exploration des données intéressantes du cube. Dans ce contexte, il est indispensable d'utiliser un système de recommandation des requêtes en prenant en compte le profil utilisateur afin de lui permettre meilleure prise en main de système OLAP, qui

pourra lui fournir des informations se rapprochant à ses intérêts individuels ainsi que lui aider dans son processus décisionnel.

2. Problématique

La recommandation des requêtes dans les bases de données est un axe de recherche prometteur, en particulier dans les systèmes OLAP où l'utilisateur navigue interactivement dans un cube en lançant une séquence de requêtes (et plus particulièrement des requêtes MDX sur un entrepôt de données. Ce processus est souvent pénible puisque l'utilisateur peut avoir des résultats qui ne les satisfont pas.

Les systèmes de recommandation existants sont généralement classés selon deux catégories : les méthodes basées sur le contenu et celles basées sur le filtrage collaboratif. Les méthodes basées sur le contenu recommandent à l'utilisateur des objets similaires à ceux qui l'ont intéressé dans le passé, tandis que les méthodes basées sur le filtrage collaboratif recommandent des objets qui ont intéressé des utilisateurs similaires.

La problématique dans notre cas est de trouver une certaine similarité entre le profil de l'utilisateur courant par rapport aux précédents profils et ainsi lui recommander des requêtes ou des parties du cubes qui les intéressent dans leur processus de prise de décision.

En outre, nous sommes devant la problématique d'interpréter quelques termes flous extrait du langage naturel dans le langage MDX afin de faciliter sa tâche de navigation dans le cube de données en lui recommandant des requêtes intéressantes.

3. Objectifs et Contributions

Au vu de la nature multi-utilisateur des entrepôts de données nous avons proposé des techniques issues du filtrage collaboratif pour recommander des requêtes OLAP à l'utilisateur.

L'idée principale de l'approche proposée est de calculer une similarité entre le profil de l'utilisateur courant et les profils des précédents utilisateurs selon une méta table de définition des profils et ainsi filtrer le profil recherché et afficher des vues matérialisées, dans lesquelles il peut naviguer et faire ses analyses OLAP dans un contexte décisionnel.

Notre projet touche principalement à l'axe de l'informatique décisionnelle (Business Intelligence) et plus précisément la recommandation des requêtes pour des décideurs en prenant en compte leurs profils afin de leur fournir des alternatives utiles pour faire face à leurs problèmes de gestion et ainsi faciliter leurs tâches décisionnelles.

Notre sujet s'étale sur deux parties d'implémentation essentielles :

- 1- Modélisation et implémentation d'un entrepôt de données relationnel contenant des données du BAC de la wilaya de Ain Defla en exécutant des requêtes floues d'une part ;
- 2- Recommandation d'un ensemble bien défini des requêtes résultantes intéressantes et pertinentes selon les préférences du décideur.

4. Organisation du Mémoire

Dans ce qui suit, nous présentons un aperçu théorique sur les systèmes d'information décisionnels d'une façon général, suivi des entrepôts de données standards, la logique floue et les entrepôts de données flous puis nous parlons enfin sur la recommandation des requêtes avec un panorama des travaux connexes de ce domaine dans un tableau comparatif. Cela nous permet de développer notre propre modèle afin de répondre à la problématique citée avant.

La formulation des requêtes floues dans les entrepôts de données flous a pour objectif de servir au mieux le processus d'aide à la décision, qui est très utile dans la pratique mais elle reste comme même un problème très complexe qui dépend énormément des termes flous du domaine d'application utilisés dans la requête.

Notre tâche principale consiste à :

- 1- Réaliser et intégrer quelques termes flous dans le langage MDX standard pour exécuter des requêtes floues relatives aux données du BAC,
- 2- Recommander les résultats des requêtes les plus utiles et les plus intéressantes selon le profil du décideur par l'affichage des vues matérialisées personnalisables.

Enfin, nous concluons par des perspectives relatives à notre projet qui sera une base pour d'autres projets d'étude ou de recherche.

Partie I

ETAT DE L'ART & ETUDE THEORIQUE

CHAPITRE I : SYSTEMES D'INFORMATION DECISIONNELS

CHAPITRE II : ENTREPOT DES DONNEES

CHAPITRE III : LOGIQUE FLOUE

CHAPITRE IV : RECOMMANDATION DES REQUETES

CHAPITRE V : TRAVAUX CONNEXES

CHAPITRE I

SYSTEMES D'INFORMATION

DECISIONNELS

"BI is about providing the right data at the right time to the right people so that they can take the right decisions"

-Níc Smith -

1. Systèmes d'Information Décisionnels

1.1. Introduction

Depuis les années 2000-2001, le marché du décisionnel ne cesse d'exploser dans le monde surtout avec l'investissement de plusieurs grandes sociétés qui souhaitent instaurer un système de Business Intelligence (B.I.) dans leur organisation pour des fins d'analyse et de prise des décisions stratégiques à long terme. Ce système difficile à mettre en œuvre, demandant une expertise et nécessitant une maîtrise d'ouvrage de la part des informaticiens concepteurs du système, ce qui rend souvent difficile le recrutement de ces profils. Nous expliquons dans ce qui suit ce qui est un système d'information décisionnel.

[Web2]

1.2. Définition d'un Système d'Information Décisionnel

Le décisionnel ne concerne souvent que les entreprises qui gèrent un historique de leurs événements passés (faits, transactions etc.). Les entreprises qui viennent de naître n'ont souvent pas besoin de faire du décisionnel car elles n'ont pas encore besoin de catégoriser ou de fidéliser leurs clients. Le souci majeur pour elles serait plutôt d'avoir le maximum de clients et c'est après en avoir récupéré un grand nombre qu'elles penseront certainement à les fidéliser et leur proposer d'autres produits susceptibles de les intéresser. C'est ce que l'on appelle Customer Relationship Management (CRM ou gestion des relations clients).

Comme cela peut se deviner, les décideurs sont les principaux utilisateurs des systèmes décisionnels. Les décideurs sont généralement des « marketeurs » ou analystes en général. Ces derniers établissent généralement des plans de

marketing qui leur permettent de mieux cibler leurs clients, de les fidéliser etc. Et pour cela, ils ont besoin d'indicateurs et des données résumées de leurs activités (ils n'ont souvent besoin de détail que pour des cas spécifiques). Par exemple, contrairement aux systèmes relationnels ou transactionnels où les utilisateurs chercheront à connaître leurs transactions pour faire un bilan, les systèmes décisionnels quant à eux cherchent plutôt à donner un aperçu global pour connaître les tendances des clients (d'où l'opposition des deux modes [quantitatif contre qualitatif]). [Web 1]

Donc un système d'information décisionnel est un ensemble de données organisé de façon spécifique, approprié à la prise de décision dans le but de piloter l'entreprise [Inmon, 1996]

1.3. Architecture des systèmes décisionnels

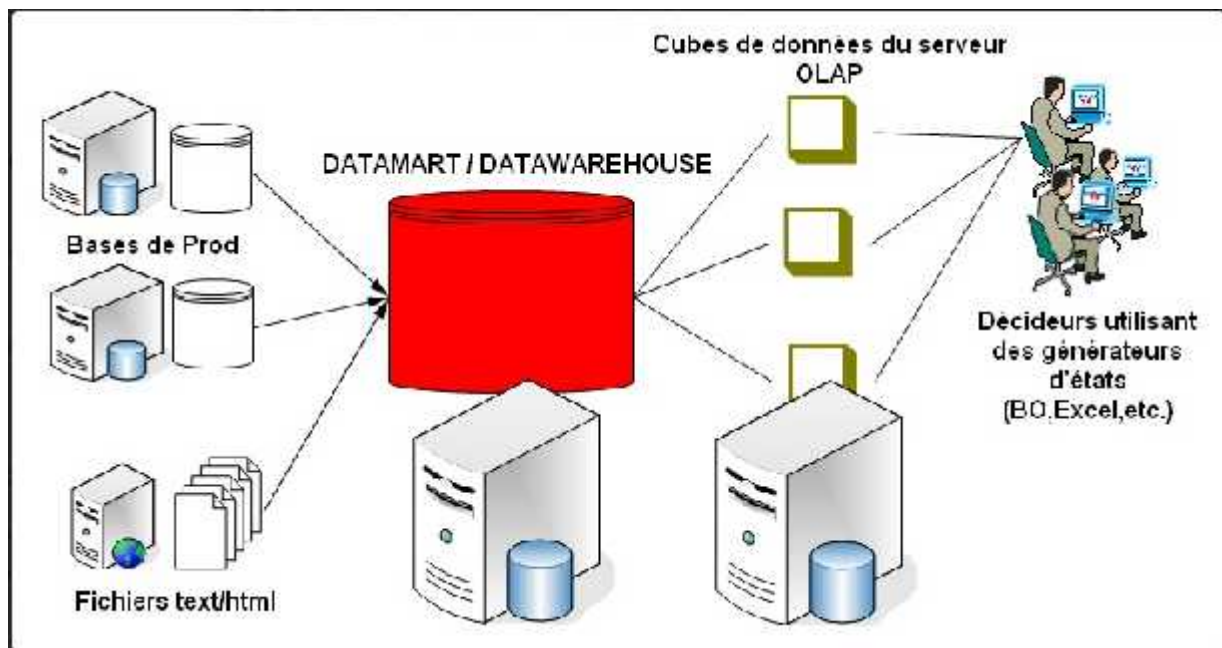


Figure 1: Architecture des systèmes décisionnels [Web 2]

Ceci est l'architecture standard des systèmes décisionnels la plus utilisée. Dans cette architecture, nous disposons d'un entrepôt de données ou DataWarehouse (généralement, il s'agit plutôt d'un DataMart qui est plus petit que le DW et qui concerne un domaine bien particulier [finance, ressources humaines, etc.]). L'entrepôt (ou encore infocentre) centralise les données issues de plusieurs sources (bases de production de l'entreprise, fichiers textes, documents web [html, xml, etc.] etc.). Ces données sont fusionnées dans l'entrepôt qui sera généralement une grosse base de données (MySQL, Oracle etc.). Ensuite, une fois l'entrepôt confectionné, les données sont extraites dans des serveurs d'analyse ou serveurs OLAP sous forme de cubes de données (Mondrian, Analysis Server, etc.) afin d'être analysées. Enfin, des générateurs d'états (Jaspersoft, Crystal Report etc.) sont utilisés afin de présenter l'étude aux utilisateurs finaux ou décideurs (ex : analystes marketing). [Kimball, 2004]

1.3.1. Les sources des données

Les sources de données sont souvent diverses et variées (BDD de production, BDD externes [fournisseurs, clients, ...] ou autres [pages web, rapport de réunion,..]) et le but est de trouver des outils ETL (Extraction / Transformation / Loading) afin de les extraire, de les nettoyer, de les transformer et de les mettre dans l'entrepôt de données (DTS de SQL Server est un exemple d'outil ETL). Des outils comme Datastage ou Talend (monde open source) sont spécialisés en la matière.

1.3.2. L'entrepôt de données

Il est le cœur du système décisionnel et demande une analyse profonde de la part de la maîtrise d'ouvrage. La conception d'un DataWarehouse diffère de la conception d'une base de données relationnelle, qu'elle est normalisée alors que les bases de données multidimensionnelles sont plutôt dénormalisées, en respectant un modèle physique donné (schéma en étoile ou en flocon).

1.3.3. Serveur d'analyse ou serveur OLAP

OLAP (On-Line Analytical Processing) est opposé à OLTP (On-Line Transactional Processing) et a pour but d'organiser les données à analyser par domaine/thème et d'en ressortir des résultats pertinents pour le décideur. Les résultats sont donc des résumés et peuvent être obtenus par différents algorithmes de datamining (fouille de données) du serveur d'analyse. Nous pouvons par exemple établir le résultat suivant : « Les clients qui achètent généralement du beurre et du pain achètent aussi du lait ». Ces résultats pourraient amener l'organisation à disposer ses rayons de telle façon qu'à côté de l'emplacement du beurre, elle mettra le pain et le lait.

1.3.4. Générateur d'état ou le Reporting

Le générateur d'état permet seulement de mieux appréhender le résultat de l'analyse. L'utilisateur final n'étant pas forcément un informaticien, il aura plus de facilité dans des états Business Objects (exemple des feuilles de données Excel) avec des diagrammes et courbes statistiques que d'aller directement requêter dans le serveur d'analyse. Il faut rappeler l'utilisation d'un serveur OLAP comme

Mondrian avec le langage de requête multidimensionnel MDX (qui ressemble au SQL mais n'est pas du SQL). Les états permettent également de faire de l'exploration (navigation) de données (comme Roll-up / Drill-Down).

1.4. Typologie des systèmes décisionnels

- ✓ Entrepôt de données ;
- ✓ Outils de pilotage ou tableaux de bord ;
- ✓ Systèmes experts ;
- ✓ Réseaux de neurones ;
- ✓ Systèmes de décision de groupe (groupware) comme la messagerie électronique ou la visioconférence ;
- ✓ Outils de raisonnement à base de cas ;
- ✓ Outils de Dataming.

1.5. Conclusion

Les Systèmes décisionnels ou la Business Intelligence est devenu un sujet incontournable et une composante essentielle du système d'information au sein d'une entreprise afin de piloter leurs activités.

Nous avons abordé dans ce chapitre la notion des Systèmes d'Information Décisionnels, son historique, ses objectifs et son architecture et ses composants notamment les entrepôts de données. Ce dernier est considéré comme étant le cœur de la BI, que nous aborderons dans le chapitre suivant.

CHAPITRE II

ENTREPOT DES DONNEES

"Data is the foundation of Digital Business. Every touch point, every click, every bite of digital exhaust"

-R.Ray Wang-

2. Entrepôt de Données

2.1. Introduction

Les besoins décisionnels exprimés par les entreprises n'étaient pas satisfaits par les systèmes transactionnels de bases des données, ce qui a donné naissance aux systèmes décisionnels basés sur les entrepôts de données.

2.2. Définition d'un Entrepôt de données

Est une collecte des données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus de prise des décisions.

[Immon, 1996].

Pratiquement c'est une grande base des données dédiée à des fins d'analyse.

2.3. Fonctionnalités d'un Entrepôt des données

Les principales fonctions d'un entrepôt de données :

- Intégrer les systèmes OLAP ;
- Regrouper et organiser des informations provenant des sources diverses ;
- Intégrer et stocker les données pour une vue orientée métier (vue matérialisée) ;
- Retrouver et analyser l'information rapidement et facilement.

2.4. Caractéristiques d'un Entrepôt des données

Nous citons les avantages suivants :

- Données orientées sujet ou métier et non pas pour application, en groupant toutes les informations d'un sujet donné ;

- Données intégrées après leurs normalisations suivant un référentiel unique pour l'ensemble des bases de production ;
- Données non volatiles en gardant la traçabilité des informations et des décisions prises ;
- Données historisées et datées dans le temps.

Le principal inconvénient est : sa grande taille d'où la nécessité de le diviser en des magasins de données que nous appelons des DATAMART afin de faciliter le processus de prise de décision. [Web 3]

2.5. Les Datamart

Est un sous ensemble d'un entrepôt de données, destiné à répondre à un thème, sujet ou secteur d'une entreprise.

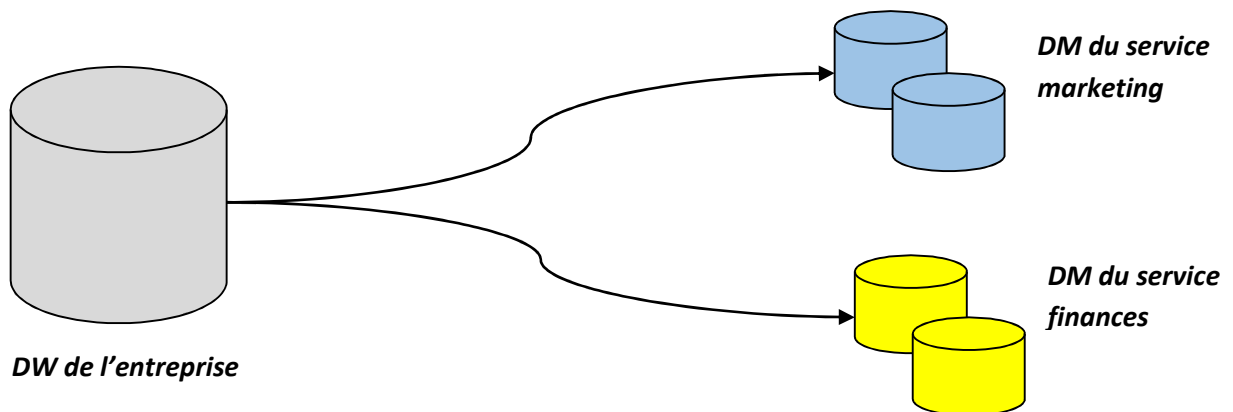


Figure 2: Les Datamart [Web 3]

2.6. Modélisation multidimensionnelle

Pour que nous construisons un entrepôt de données, nous devons utiliser un schéma multidimensionnel et cela à travers (03) niveaux, comme suit :

- Niveau conceptuel
- Niveau logique
- Niveau physique

2.6.1. Niveau Conceptuel

Pour concevoir un modèle dédié à un entrepôt de données, il faut définir deux concepts essentiels : **les faits** avec **les mesures** et **les dimensions** avec **les hiérarchies** ou **les niveaux de granularité**.

a. Les Faits :

Les faits sont les sujets à analyser. Un fait représente la valeur d'une mesure, calculée ou mesurée, selon un membre de chacune des dimensions.

Donc la table des faits contient les valeurs des **mesures** et les clés vers les tables des dimensions.

b. Les Mesures :

Une mesure est un élément des données dans la table des faits sur lequel portent les analyses, en fonction des différentes dimensions. Ces mesures peuvent être des membres simples ou des membres calculés.

Ces valeurs sont le résultat d'opérations d'agrégation sur les données.

Exemple : - Le nombre de produits vendus par manager : *une mesure simple*

- Le bénéfice par produit pour une période donnée : *une mesure calculée*

c. Les Dimensions :

Une dimension est l'axe d'analyse avec lequel on veut faire l'analyse, qui comporte des attributs ou des membres.

Une dimension est tout ce qu'on utilisera pour faire nos analyses.

d. Les Hiérarchies ou les niveaux de granularité :

Chaque attribut ou membre d'une dimension peut appartenir à un ou à des niveaux hiérarchiques ou des niveaux de granularité.

Les attributs qui définissent les niveaux de granularité sont appelés des paramètres.

Au niveau conceptuel, il existe deux modèles :

- En étoile (Star Schema)
- En constellation (Fact Constellation Schema)

1. Le modèle en étoile :

Contient une table de fait centrale et des tables de dimensions telle que la table des faits contienne une clé étrangère pour chaque table de dimension

2. Le modèle en Constellation (Séries d'étoiles) :

C'est une fusion de plusieurs modèles en étoiles avec plusieurs tables de fait et des dimensions communes.

2.6.2. Niveau Logique

Concerne la description de la base multidimensionnelle suivant la technologie utilisée : ROLAP, MOLAP, HOLAP.

a. ROLAP (Relational OLAP) :

- Les données sont stockées dans une Base des Données relationnelle
- Le moteur OLAP (OnLine Analytical Process) permet de simuler le comportement des BD multidimensionnelle.

b. MOLAP (Multidimensional OLAP) :

- Les données sont stockées comme des matrices à plusieurs dimensions comme des cubes $[1 : m, 1 : n, 1 : p]$.

c. HOLAP (Hybrid OLAP) :

- Les données sont stockées dans un SGBD relationnel (table des faits et dimensions) et les données agrégées se sont stockées dans un cube.
- C'est une solution hybride entre ROLAP et MOLAP

Dans le niveau logique, il existe un modèle : **En Flocon**, c'est le modèle de neige (Snow Flake Schema).

➤ Le modèle en Flocon :

- Modèle en étoile avec normalisation des dimensions à travers des niveaux hiérarchiques
- La table de dimension de niveau hiérarchique le plus bas est reliée à la table de fait (elle a la granularité la plus fine).

➤ Le Cube de données :

- Modélisation multidimensionnelle des données facilitant l'analyse des données selon différentes dimensions (temps, lieu, produit, ...)
- Les calculs sont réalisés lors du chargement ou de la mise à jour du cube.

2.6.3. Niveau Physique

Il s'agit de l'implantation et dépend donc du logiciel utilisé. Nous utilisons trois techniques d'analyse pour réaliser physiquement notre entrepôt de données : TOP DOWN, BOTTOM UP, MIDDLE OUT.

a. TOP DOWN :

Conception de l'intégralité de l'entrepôt de données à condition de la connaissance préalable de toutes les dimensions et les faits.

b. BOTTOM UP :

Création des DATAMART un par un puis les regrouper par des niveaux intermédiaires jusqu'à l'obtention d'un véritable entrepôt.

c. MIDDLE OUT :

Conception intégrale de l'entrepôt de données (toutes les dimensions, tous les faits, toutes les relations) puis créer des divisions plus petites et plus gérables.

2.7. Etapes de réalisation d'un entrepôt de données

Les cinq étapes importantes pour la réalisation d'un entrepôt de données [Web 3] sont :

- La conception
- L'acquisition des données
- La définition des aspects techniques de la réalisation
- La définition des modes de restitution
- Les stratégies d'administration, évolution, maintenance.

2.7.1. Conception

- ***Définir la finalité du DW***
 - ✓ Quelle activité ou secteur de l'entreprise faut-il piloter ?
 - ✓ Quel est le processus de l'entreprise à modéliser ?
 - ✓ Qui sont les décideurs ?
 - ✓ Quels sont les faits à mesurer ?
 - ✓ Quelles sont les dimensions à utiliser dans l'analyse ?

- ***Définir le modèle de données***

- ✓ Modèle en étoile/flocon ?
- ✓ Et/ou Cube ?
- ✓ Et/ou vues matérialisées ?

2.7.2. Acquisition des données

- ***Pour alimentation ou mise à jour de l'entrepôt***

- ✓ Mise à jour régulière à travers d'un outil d'automatisation de chargement de l'entrepôt : ETL (*Extract Transform Load*).

- ***ETL***

- ✓ Permet de découvrir, analyser et extraire les données à partir de sources hétérogènes
- ✓ Permet de nettoyer et standardiser les données
- ✓ Permet de charger les données dans un entrepôt

2.7.3. Aspects techniques

- Contraintes logicielles, matérielles, humaines,

2.7.4. Restitution

- Est le but du processus d'entreposage et conditionne le choix de l'architecture et de construction du DW
- Toutes les analyses doivent être réalisables
- Types d'outils de restitution : requêteurs et outils d'analyse, outils de datamining

2.7.5. Administration et maintenance

- Toutes les stratégies à mettre en place pour l'administration, l'évolution et la maintenance

2.8. Manipulation d'un entrepôt des données

2.8.1. Opérations de forage

- **ROLL UP (Forage vers le Haut)**
 - ✓ Représente les données à un niveau de granularité supérieur selon la hiérarchie de la dimension désirée. Ici, on agrège selon une dimension.
- **DRILL DOWN (Forage vers le Bas)**
 - ✓ L'inverse de Roll Up
 - ✓ Représente les données à un niveau de granularité inférieur. Ici, on détaille selon une dimension.

2.8.2. Opérations de Sélection/Projection

- **SLICE**
 - ✓ Opération de sélection sur une tranche de cube obtenue par prédicats selon une dimension
- **DICE**
 - ✓ Opération de projection selon un axe ou une dimension. C'est le cumul de sélection

2.8.3. Opérations de Restructuration/Réorientation

- **PIVOT (ROTATE)**
 - ✓ Tourner le cube pour visualiser une face différente
- **SWITCH (PERMUTATION)**
 - ✓ Inter changer la position des membres d'une dimension
- **NEST**
 - ✓ Imbriquer des membres issus de dimensions différentes
- **PUSH (ENFONCEMENT)**
 - ✓ Combiner les membres d'une dimension aux mesures (les membres deviennent le contenu des cellules)
- **AddM, DelM**
 - ✓ Pour l'ajout et la suppression des mesures à afficher

2.9. Conclusion

A travers ce chapitre, nous avons abordé le concept des entrepôts de données ou les Data Warehouse, qui a pour but primordial répondre aux besoins décisionnels.

La construction d'un Data Warehouse passe par plusieurs étapes essentielles dans ce chapitre, à savoir le processus d'alimentation ou ETL jusqu'aux techniques de modélisation et de navigation dans les cubes de données.

CHAPITRE III

LOGIQUE FLOUE

“If you are willing the flexibility of your approach, you can usually do something better”

-John Carmack-

3. Logique Floue

3.1. Introduction

Les bases théoriques de la logique floue (fuzzy logic) ont été établies au début des années 1965 par le professeur Zadeh de l'université de Californie de Berkeley. Cette technique associe les notions de « sous-ensemble flou » et de « la théorie des possibilités ».

Il s'agit d'une approche calquée sur le raisonnement humain plutôt que sur des calculs rigides pour des problèmes mal définis.

3.2. Définition de la logique floue

La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en 1965 en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, nous permettons à une condition d'être dans un autre état que vrai ou faux. La logique floue confère ainsi une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes. Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. [Web 4]

3.3. Les sous-ensembles flous

La logique floue repose sur la théorie des ensembles flous, qui est une généralisation de la théorie des ensembles classiques [Zadeh, 1965]. Par abus de langage, suivant les us de la littérature, nous utiliserons indifféremment les

termes sous-ensemble flou et ensemble flou. Les ensembles classiques sont également appelés ensemble nets, par opposition a flou, et de même la logique classique est également appelée logique booléenne ou binaire. [Web 4]

Définition 1

Soit X un ensemble. Un sous-ensemble flou A de X est caractérisé par une fonction d'appartenance. $f^a : X \rightarrow [0, 1]$. [Web 4]

Note : Cette fonction d'appartenance est l'équivalent de la fonction caractéristique d'un ensemble classique.

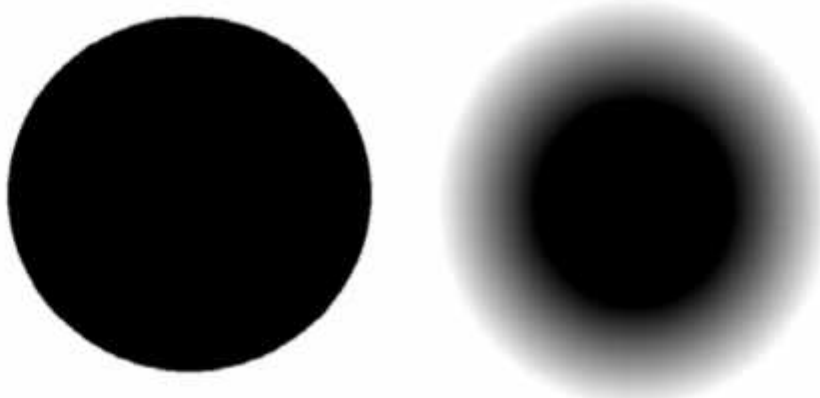


Figure 3: Représentation graphique d'un ensemble classique et d'un ensemble Flou [Web 4]

Soit X un ensemble, A un sous-ensemble flou de X et μ_A la fonction d'appartenance le caractérisant.

Définition 2

La hauteur de A , notée $h(A)$, correspond à la borne supérieure de l'ensemble d'arrivé de sa fonction d'appartenance : $h(A) = \sup\{\mu_A(x) \mid x \in X\}$. [Web 4]

Définition 3

A est dit normalisé si et seulement si $h(A) = 1$. En pratique, il est extrêmement rare de travailler sur des ensembles flous non normalisés. [Web 4]

Définition 4

Le support de A est l'ensemble des éléments de X appartenant au moins un peu à A. Autrement dit, c'est l'ensemble $\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\}$. [Web 4]

Définition 5

Le noyau de A est l'ensemble des éléments de X appartenant totalement à A. Autrement dit, c'est l'ensemble $\text{noy}(A) = \{x \in X \mid \mu_A(x) = 1\}$. Par construction, $\text{noy}(A) \subseteq \text{supp}(A)$. [Web 4]

Définition 6

Une α -coupe de A est le sous-ensemble classique des éléments ayant un degré d'appartenance supérieur ou égale à α : $\alpha\text{-coupe}(A) = \{x \in X \mid \mu_A(x) \geq \alpha\}$ [Web 4]

3.4. Les variables linguistiques

La fonction d'appartenance faisant le lien entre la logique floue et la variable linguistique que nous allons définir à présent permettra de définir des systèmes flous en langage naturel.

Définition

Soit V une variable (qualité du service, montant du pourboire, etc.), X la plage de valeurs de la variable et TV un ensemble fini ou infini de sous-ensembles flous.

Une variable linguistique correspond au triplet (V, X, TV) . [Web 4]

3.5. Les opérateurs flous

Afin de pouvoir manipuler aisément les ensembles flous, nous redéfinissons les opérateurs de la théorie des ensembles classiques afin de les adapter aux fonctions d'appartenance propres à la logique floue permettant des valeurs strictement entre 0 et 1. Contrairement aux définitions des propriétés des ensembles flous qui sont toujours les mêmes, la définition des opérateurs sur les ensembles flous est choisie, à l'instar des fonctions d'appartenance. Voici les deux ensembles d'opérateurs pour le complément (NON), l'intersection (ET) et l'union (OU) utilisés le plus couramment :

Dénomination	Intersection ET : $\mu_{A \cap B}(x)$	Réunion OU : $\mu_{A \cup B}(x)$	Complément NON : $\mu_{\bar{A}}(x)$
Opérateurs de Zadeh MIN/MAX	$\min(\mu_A(x), \mu_B(x))$	$\max(\mu_A(x), \mu_B(x))$	$1 - \mu_A(x)$
Probabiliste PROD/PROBOR	$\mu_A(x) \times \mu_B(x)$	$\mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x)$	$1 - \mu_A(x)$

Tableau 1: Les opérateurs flous [Web 4]

Avec les définitions usuelles des opérateurs flous, nous retrouvons toujours les propriétés de commutativité, distributivité et associativité des opérateurs classiques. Cependant, nous relevons deux exceptions notables :

En logique floue, le principe du tiers exclu est contredit : $A \sqcup \bar{A} \neq X$, autrement dit $\mu_{A \sqcup \bar{A}}(x) \neq 1$.

En logique floue, un élément peut appartenir à A et non A en même temps :

$A \neq \emptyset$, autrement dit $\mu_A(x) \neq 0$. Notons que ces éléments correspondent à l'ensemble $\text{supp}(A) - \text{noy}(A)$. [Web 4]

3.6. Le raisonnement en logique floue

En logique classique, les raisonnements sont de la forme :

$$\left\{ \begin{array}{l} \text{Si } p \text{ alors } q \\ P \text{ vrai alors } q \text{ vrai} \end{array} \right.$$

En logique floue, le raisonnement flou, également appelé raisonnement approximatif, se base sur des règles floues qui sont exprimées en langage naturel en utilisant les variables linguistiques dont nous avons donné la définition précédemment. Une règle floue aura cette forme :

Si $x \in A$ et $y \in B$ alors $z \in C$, avec A, B et C des ensembles flous.

Voici les deux définitions de l'implication floue les plus couramment utilisées :

Nom	Valeur de vérité
Mamdani	$\min(f_a(x), f_b(x))$
Larsen	$f_a(x) \times f_b(x)$

Tableau 2: Définitions de l'implication floue [Web 4]

3.7. Entrepôt de Données Flou

3.7.1. Concept du Data Warehouse Flou

Les concepts flous peuvent être intégrés, dans une structure de table méta sans affecter le noyau d'un entrepôt de données. Le modèle de **FASEL** proposé est plus souple, car il permet d'intégrer et de définir les concepts flous sans qu'il soit nécessaire de concevoir le cœur d'un entrepôt de données. En utilisant ces données floues approche de l'entrepôt, il est possible d'extraire et d'analyser les données simultanément dans un classique pointu et d'une manière floue.

3.7.2. Définition de base et les méta tables floues

Pour l'intégration des concepts flous dans un entrepôt de données, il faut d'abord analyser quels éléments dans l'entrepôt de données doivent être classés flous. L'élément peut être un fait dans la table de faits ou d'un attribut d'une dimension. Un élément qui doit être classé flou est appelé l'attribut cible et la gamme des instances de cet élément de valeur est appelé le domaine de l'attribut. [Fasel, 2012]

Définition 01(Domaine d'attribut). Un ensemble de valeurs possibles ou la gamme des valeurs possibles qu'un attribut de dimension ou d'un fait peut avoir est appelé domaine d'un attribut ou univers du discours. Domaine Dom d'un attribut A est représenté par « Dom_A ». [Fasel, 2012]

Définition 02(L'attribut cible). Un attribut de dimension ou un fait qui est requis pour être classé flou est appelé un attribut cible (TA). Sous la classification floue, les instances de TA sont classées sur un ensemble (S) qui est représenté par une variable linguistique. La linguistique de la variable est constituée d'un

ensemble de termes non numériques appelés termes linguistiques $S = T_1, \dots, T_k$. Les termes linguistiques d'une variable linguistique sont capturés dans un attribut appelé classe attribut d'appartenance. [Fasel, 2012]

Définition 03 (Attribut de l'appartenance de classe). Une classe d'appartenance d'attribut (CMA) pour un attribut cible TA, représenté par CMA_{TA} , est un attribut qui possède un ensemble de linguistique termes T_1, \dots, T_k à laquelle l'attribut cible peut appartenir. En d'autres termes, pour toutes les valeurs possibles d'un attribut cible (domaine des données d'attribut Dom_{TA}), il y'a un rapport correspondant à une valeur CMA. Les valeurs de CMA sont les valeurs de l'ensemble S.

Toutes les valeurs de Dom_{TA} appartiennent à un certain degré flou pour une valeur CMA. Le degré d'appartenance à une valeur de CMA est appelé **degré d'appartenance** et ce degré d'appartenance définit la relation d'une instance TA à une valeur CMA. [Fasel, 2012]

Définition 04 (degré d'appartenance). Le degré d'appartenance $MD \in [0, 1]$ est le degré auquel les valeurs d'un attribut cible TA sont en lien avec le plan linguistique T_1, \dots, T_k , respectivement avec les valeurs de CMA.

MD est calculé en utilisant une fonction d'appartenance. [Fasel, 2012]

Définition 05 (fonction d'appartenance). Une fonction de l'appartenance d'une classe d'appartenance d'attribut CMA est une fonction $\mu (TA)$ qui est utilisé pour calculer le degré d'appartenance MD d'une TA à une classe d'appartenance d'attribut CMA : $\mu : TA \rightarrow [0, 1]$. [Fasel, 2012]

Les degrés d'appartenance générés par des fonctions d'appartenance sont capturés comme des attributs degrés d'appartenance dans le modèle d'entrepôt de données floue. Un attribut degré d'appartenance est défini en tant que :

Définition 06 (Attribut Degré d'appartenance). Un attribut degré d'appartenance MDA d'un attribut cible TA, est un attribut qui dispose d'un ensemble de degrés d'appartenance de l'attribut cible TA. La valeur d'un degré d'appartenance est calculée par une fonction d'appartenance et est représenté par $\mu_t(\text{TA}) = \text{MD}$, où MD est le degré d'appartenance de TA pour le terme linguistique t dans les CMA. [Fasel, 2012]

Un attribut qui doit être manipulé flou est étendu avec deux tables méta. La première table méta contient une description de la notion floue et la deuxième table méta contient les degrés d'appartenance de chaque instance à l'égard de la classe d'appartenance des attributs. Les deux tableaux sont définis comme suit :

Définition 07 (Table de classification floue). Une table qui se compose de termes linguistiques et leurs identificateurs uniques est appelé table de classification floue FCT. Qui est une table à deux attributs qui se compose d'un attribut d'identité et un attribut de classe d'appartenance, où l'attribut de l'identité est un identifiant unique des valeurs de la table. [Fasel, 2012]

$$\text{FCT}_{\text{TA}} = \{\text{identifiant}, \text{CMA}_{\text{TA}}\}$$

Définition 08 (Table d'appartenance floue). Une table qui stocke les valeurs représentant la mesure dans laquelle une valeur est liée à un terme linguistique est appelée table d'appartenance floue FMT. Qui est une table avec quatre attributs : l'attribut identifiant de la table, l'identifiant de l'attribut cible TA, l'identifiant de l'attribut classe d'appartenance A dans CMA_{TA} dans la table de classification floue FCT_{TA} et l'attribut degré d'appartenance MDA_{TA} pour TA. [Fasel, 2012]

$$\text{FMT}_{\text{TA}} = \{\text{identifiant}, \text{identifiant de TA}, \text{identifiant de } \text{CMA}_{\text{TA}}, \text{MDA}_{\text{TA}}\}$$

3.7.3. Modèle d'un entrepôt de données flou

Le modèle d'entrepôt de données flou est une combinaison de quatre types de tables. Ce sont des tables de dimensions, tables de faits, des tables d'appartenance floue et tables de classification floues.

Définition (Modèle d'un entrepôt de données flou). Un modèle d'entrepôt de données flou est un ensemble de tables et il est représenté par **FDW**. [Fasel, 2012]

$FDW = \{Dim, Fact, FCT_{TA}, FMT_{TA}\}$ Où,

$Dim = \{\text{un ensemble de catégorie attributs, le niveau de la catégorie des attributs}\}$

$Fact = \{\text{un ensemble de mesures}\}$

$TA = \{T A_1, T A_2, \dots, T A_n\}$, n est le nombre d'attributs qui sont classés flou.

Il est à noter que l'ensemble des attributs cibles est un sous-ensemble de l'ensemble de la dimension et l'ensemble de faits. Formellement, TA est un sous-ensemble de $Dim \cup Fact$ (c.-à-d. $\forall T A_i \in Dim \cup Fact : 1 \leq i \leq n$).

Pour chaque attribut $T A_1, T A_2, \dots, T A_n$:

$FCT_{TA_i} = \{Identifiant, CMA_{TA_i}\}$ Où $1 \leq i \leq n$.

$FMT_{TA_i} = \{Identifiant, Identifiant FCT, Identifiant TA_i, MDA_{TA_i}\}$ où $1 \leq i \leq n$.

3.8. Conclusion

Nous avons abordé dans ce chapitre la notion générale de la logique floue et son utilisation dans le monde informatique. Elle s'affirme comme une technique opérationnelle utilisée à côté d'autres techniques avancées.

CHAPITRE IV

RECOMMANDATION DES

REQUETES

"The goal is to turn data into information, and information into insight".

-Carly Fiorina-

4. La Recommandation des Requêtes

4.1. Introduction

Les systèmes d'accès à l'information ont pour objectif de fournir des résultats répondant aux besoins et attentes des utilisateurs et ne nécessitant aucun effort ou traitements supplémentaires pour être examinés et exploités. En ce sens, deux problèmes distincts sont fréquemment soulevés. Le premier est comment gérer les requêtes retournant un résultat vide. Si aucun résultat ne répond pleinement à la requête soumise par l'utilisateur, l'enjeu est alors de fournir des réponses alternatives (recommandations). Le deuxième problème concerne les requêtes qui retournent un ensemble de résultats volumineux pour être exploitées efficacement et facilement par l'utilisateur. Parmi l'ensemble volumineux des résultats retournés, l'utilisateur n'est généralement intéressé que par le sous ensemble des résultats qu'il juge les plus pertinents. Pour extraire ce sous ensemble, l'utilisateur doit parcourir la totalité des résultats retournés, ce qui constitue pour lui une tâche fastidieuse et ardue.

La solution consiste donc à faire coopérer et gérer l'interactivité entre le système avec l'utilisateur. Ce dernier peut exprimer ses préférences relatives aux résultats qu'il souhaite obtenir. Le système, en se basant sur les préférences utilisateur, va réduire l'ensemble des résultats obtenus et ne retourne que ceux qui sont pertinents. [Jerbi, 2012]

4.2. Définition de la recommandation des requêtes

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 [Jerbi, 2012]

La recommandation de requête fournit en général deux fonctionnalités :

- 1) l'assistance à la définition de requête par la proposition de parties de requête,
- 2) la proposition de requêtes complètes afin de faciliter l'exploration de l'espace multidimensionnel.

Ainsi, la prise en compte de l'utilisateur dans un système d'information décisionnel a pour objectif principal l'amélioration de l'extraction des informations en fonction de la perception et des intérêts de cet utilisateur. Cette extraction implique l'utilisation des techniques de personnalisation et de recommandation en se basant sur les préférences des utilisateurs, de ses usages précédents ou historiques, de ses contraintes, etc. Donc, l'objectif recherché par l'utilisateur est qu'il ne sera pas inondé d'informations sans relation avec ses attentes mais au contraire il aura des informations correspondantes au mieux à ses centres d'intérêt. Il existe deux types d'approches pour gérer la recommandation dans un contexte décisionnel, à savoir la personnalisation et la recommandation. [Khemiri, 2015]

4.3. Types des recommandations des requêtes

4.3.1. *La personnalisation*

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.

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, son historique de navigation etc.). Le terme personnalisation est alors utilisé pour la prise en compte du profil utilisateur.

A la lumière de ce qui précède, nous pouvons dire que la personnalisation a pour objectif de faciliter l'expression des besoins des utilisateurs et de rendre l'information sélectionnée exploitable à l'utilisateur. Par conséquent, la fonction de personnalisation d'un système peut être résumée en deux phases : (i) définition du profil et (ii) exploitation du profil afin de fournir des réponses pertinentes à l'utilisateur. Le système doit définir le profil de l'utilisateur d'une manière implicite ou explicite puis l'exploiter que ce soit par adaptation du système ou par recommandation de nouvelles informations à l'utilisateur. [Jerbi, 2012]

4.3.2. *La recommandation*

Pour parler de la recommandation [Khemiri, 2015], il faut préciser que la personnalisation et la recommandation sont très liées tel que la personnalisation évoque les préférences personnelles tandis que la recommandation évoque des suggestions sans recours obligatoirement aux préférences utilisateur. Par

ailleurs, les recommandations peuvent être proposées en fonction du comportement général de l'utilisateur.

En effet, un système de recommandation a pour objectif de fournir à un utilisateur un contenu pertinent en fonction de ses préférences. Ce dernier peut ainsi réduire son temps de recherche et reçoit également des suggestions de la part du système auxquelles il n'aurait pas prêté attention.

Ainsi, les systèmes de recommandation sont des systèmes d'information de filtrage visant à représenter les éléments d'information qui peuvent intéresser l'utilisateur. Trois approches de recommandation sont présentées dans la littérature :

- *Approche de recommandation cognitive ou basée sur le contenu* (content based recommendation) qui repose sur le contenu seulement ;
- *Approche de filtrage collaboratif* (collaborative filtering recommendation) tente de trouver des liens entre les utilisateurs via leurs intérêts respectifs ;
- *Approche de recommandation hybride* qui combine les deux.

4.3.2.1. Recommandation Cognitive ou basée sur le contenu

Les systèmes de recommandation basés sur le contenu s'appuient sur des évaluations effectuées par un utilisateur sur un ensemble d'objets (documents, livres, films, etc.). L'objectif est alors de comprendre les motivations l'ayant conduit à juger comme pertinent ou non un objet donné. Le système peut alors proposer à l'utilisateur un choix parmi de nouveaux objets jugés proches des objets qu'il a précédemment appréciés.

Les systèmes de recommandation basés sur le contenu comparent les caractéristiques des objets afin de recommander des objets similaires à ceux déjà appréciés par l'utilisateur.

4.3.2.2. Recommandation Collaborative

La notion de filtrage collaboratif est à la base de la recommandation, les méthodes de filtrage par le contenu étant plutôt liées aux systèmes de recherche d'informations personnalisés. Elle se fonde non plus sur le même utilisateur mais cherche à rapprocher l'utilisateur courant avec un ensemble d'utilisateurs existants. Le filtrage collaboratif se base sur l'idée que les personnes à la recherche d'information devraient se servir de ce que d'autres ont déjà trouvé et évalué.

4.3.3.3. Recommandation Hybride

Les systèmes de filtrage hybrides combinent les deux types de filtrage : le filtrage basé sur le contenu et le filtrage collaboratif. Ainsi, la recommandation personnalisée utilise les données comportementales de l'utilisateur pour personnaliser en temps réel le contenu fourni. Les systèmes de recommandation hybride fournissent des suggestions en adéquation avec le profil utilisateur. Par exemple, cette technique est utilisée en e-commerce pour proposer une offre de produits unique et spécifique à l'internaute. Cette technique consiste à filtrer les recommandations par les préférences personnelles pour avoir des recommandations personnalisées. Les approches hybrides utilisent des algorithmes alliant des comparaisons entre profils d'objets et profils d'utilisateurs.

Le processus de prise de compte de l'utilisateur se présente comme suit :

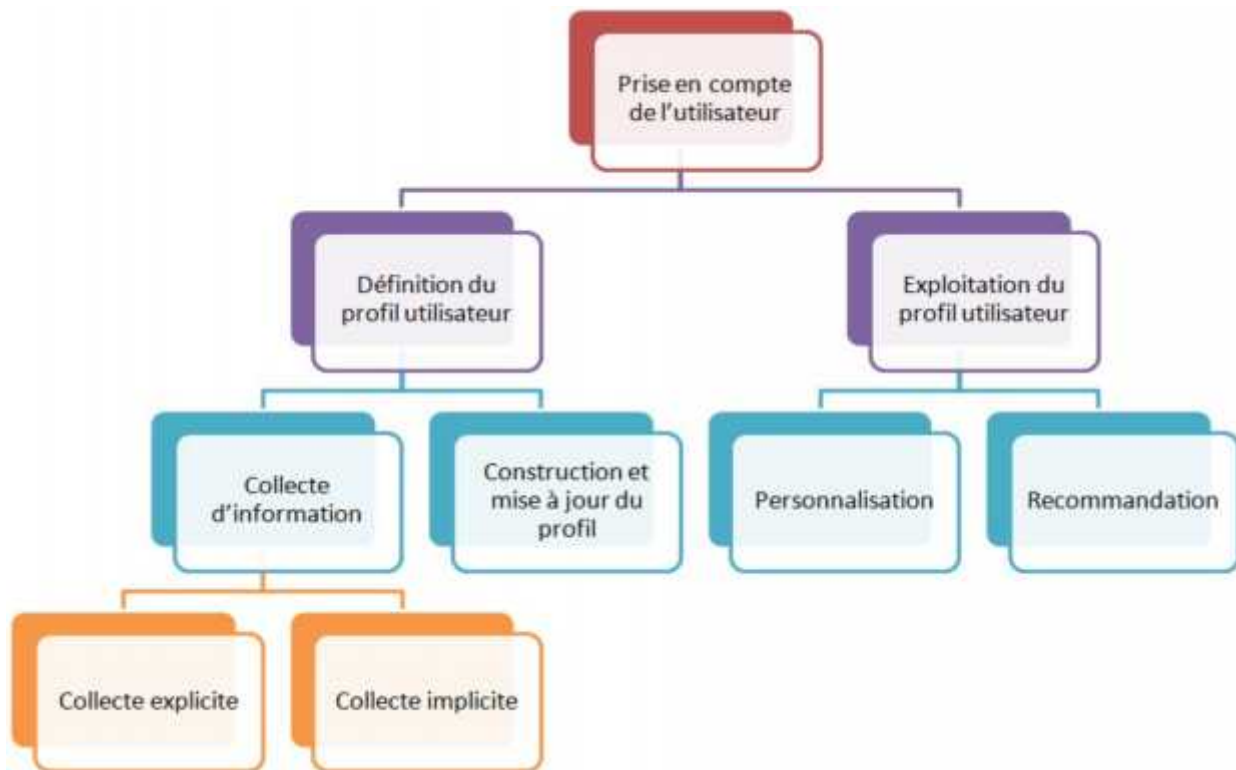


Figure 4: Le processus de prise de compte de l'utilisateur

4.4. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur les systèmes de recommandations des requêtes en général, partant de leurs définitions, leurs types et modèles.

Nous allons aborder dans le chapitre suivant, les importants travaux liés aux systèmes de recommandations, leur principe de base, leur avantages et inconvénients.

CHAPITRE V

TRAVAUX CONNEXES

"I don't spend my time pontificating about high concept things, I spend my time solving engineering and manufacturing problems".

-Elon Musk-

5. Travaux Connexes

5.1. Introduction

Nous considérons ici les travaux les plus significatifs qui répondent au mieux à notre problématique de recherche et à notre projet, à savoir la recommandation des requêtes décisionnelles dans des entrepôts de données relationnels.

5.2. Travaux de Bellatreche [Bellatreche, 2005&2006]

Principe : Les travaux de Bellatreche permettent de personnaliser la requête ainsi que la visualisation du résultat.

Approche : La personnalisation est centrée sur l'utilisateur. Elle consiste à étendre la requête de l'utilisateur par des prédicats issus du profil. Cette approche s'applique à une requête ponctuelle sans considérer les requêtes précédentes de l'analyse courante. La modélisation de l'analyse OLAP n'est pas étudiée.

Algorithme : Il prend en entrée la requête Q de l'utilisateur, le profil et une contrainte de visualisation. Il génère une requête $Q' \sqsupseteq Q$ au sens de l'inclusion syntaxique des requêtes MDX.

Système : L'implantation de cette approche est effectuée en dehors du moteur de requête OLAP puisque la requête étendue est envoyée au moteur de requête et est exécutée à la place de la première.

Avantages & Inconvénients : L'avantage majeur de cette approche est le non accès aux tables fait durant le processus de personnalisation. Cependant, l'inconvénient est le chargement en mémoire des tables dimension durant ce processus, ce qui pose un problème de performance.

5.3. Travaux de Ravat [Ravat, 2007&2008]

Principe : Les travaux de Ravat se situent dans le cadre de la personnalisation des requêtes par expansion.

Approche : Elle n'est pas basée sur un modèle d'analyse OLAP malgré la proposition d'une algèbre d'opérateurs OLAP. La personnalisation d'une requête est indépendante des requêtes déjà lancées au cours de l'analyse courante.

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

Système : Il est demandé à l'utilisateur de définir les règles et de spécifier manuellement un seuil de personnalisation. L'implantation de cette approche est effectuée au niveau applicatif par l'ajout d'un module permettant d'analyser la requête et de l'enrichir en fonction des règles.

Avantages & Inconvénients : L'inconvénient majeur de cette approche réside dans la subjectivité de la précision du seuil de la requête qui déterminera les attributs de dimension à afficher. L'approche ne permet pas par exemple de focaliser l'analyse sur l'année en cours. Bien que cette approche permette de recommander des attributs de dimension, elle ne représente pas une approche d'assistance à la formulation de requête. En effet, elle ne permet pas de traiter des requêtes incomplètes.

5.4. Travaux de Giacometti [Giacometti, 2008&2009]

Principe : Les travaux de Giacometti permet de recommander la requête suivante durant une analyse OLAP.

Approche : Une analyse est considérée comme une succession de requêtes MDX, où chaque requête est représentée par un ensemble de références. Elle est basée sur l'exploitation de l'historique des requêtes de tous les utilisateurs. Il s'agit d'une approche *centrée historique* qui sélectionne d'une façon *descendante* une requête à recommander à partir de l'ensemble des requêtes du log.

Algorithme : Il prend en entrée la session courante et un log des analyses effectuées sur la BDM. Il génère un ensemble de requêtes à recommander.

Système : l'algorithme de génération de recommandation est implanté au niveau applicatif. Cependant, cet algorithme nécessite l'appariement de l'analyse courante avec toutes les analyses du log, ce qui induit un temps de calcul important.

Avantages & Inconvénients : L'inconvénient majeur de cette approche est l'absence de la prise en compte de l'utilisateur lors de la génération des recommandations. L'ensemble des requêtes candidates est le même quel que soit l'utilisateur. Par ailleurs, cette approche se limite à la recommandation de requêtes suivant des scénarios d'analyse précédents et ne permet pas de recommander de nouvelles requêtes.

5.5. Travaux de Golfarelli [Golfarelli, 2009]

Principe : Les travaux de Golfarelli traitent la problématique des résultats volumineux ou vides des requêtes OLAP qui sont inspirés des travaux qui se basent sur le modèle BMO (« Best Matches Only ») d'exécution de requêtes où seulement les tuples du résultat qui ne sont pas dominés par d'autres sont renvoyés.

Approche : La requête est exécutée sans personnalisation, puis les préférences sont exploitées pour déterminer les tuples du résultat qui sont meilleurs que tous les autres.

Algorithme : Il prend en entrée une requête, un ensemble de préférences et l'instance de l'entrepôt de données. Il produit en sortie l'ensemble des tuples non dominés qui représente un sous-ensemble du résultat généré sans personnalisation.

Système : l'implantation impose l'utilisation du langage MDX pour renvoyer des n-uplets avec différents niveaux d'agrégation. Cette implantation a nécessité l'extension du moteur de requête afin de prendre en compte le modèle BMO.

Avantages & Inconvénients : Les avantages sont : la réduction du volume important du résultat et l'expressivité du modèle de préférences. Les principaux inconvénients sont l'effort manuel et cognitif de formulation des préférences et la nécessité d'exécuter plusieurs requêtes intermédiaires afin de répondre à une seule requête utilisateur.

5.6. Travaux de Garrigós [Garrigos, 2009]

Principe : le mécanisme de personnalisation selon Garrigós débute depuis l'étape de conception de l'entrepôt.

Approche : Dans la première étape, un profil de l'utilisateur est défini ainsi qu'un ensemble de règles précisant les actions de personnalisation à effectuer en ligne. Dans la seconde et au moment de l'interrogation de l'entrepôt, se fait la génération d'une vue du schéma en fonction du profil de l'utilisateur courant afin de faciliter la tâche de formulation de requête. Il s'agit d'une approche d'assistance à la définition de requête.

Algorithme : Le principe est de générer une partie du schéma de l'entrepôt (approche descendante) à partir du profil de l'utilisateur et d'un ensemble de règles. L'union entre les règles et le profil permet de sélectionner celles dont les conditions de déclenchement sont satisfaites.

Système : Garrigós n'a pas donné d'indications sur l'implantation et la performance de son approche. Il faut noter qu'aucune extension du moteur de requête n'est obligatoire puisque la personnalisation concerne la consultation du schéma. Cependant, un langage de définition des règles est utilisé.

Avantages & Inconvénients : L'inconvénient principal est que l'approche proposée touche seulement le niveau conceptuel mais pas le modèle d'implantation ou de stockage du profil ou de l'algorithme de son exploitation pour la personnalisation.

5.7. Travaux de H. Jerbi [Jerbi, 2012]

Principe : elle se base sur le modèle d'analyse OLAP indépendant des choix d'implantation ainsi que du langage de requête et des structures de visualisation des données.

Approche : elle permet d'assister l'utilisateur à formuler une requête OLAP, ou de personnaliser des requêtes précédentes et de renvoyer, en plus du résultat de requêtes, des recommandations soit par anticipation ou par alternative.

Algorithme : en entrée la requête Q , le profil utilisateur U et après analyse de Q et U on détermine les préférences candidates homogènes selon un score puis on restitue en sortie le résultat de la requête Q' .

Système : l'implantation se fait grâce au schéma en constellation à travers des faits et des dimensions via une analyse OLAP en utilisant les graphes.

Avantages & Inconvénients : comme inconvénient, on cite le problème d'optimisation des algorithmes utilisés afin de limiter le nombre d'accès à la méta base plus la saturation de la mémoire cache à cause des systèmes OLAP. Pour les avantages, il assiste le décideur dans la navigation OLAP.

5.8. Travaux de R.Khemiri [Khemiri, 2015]

Principe : Guider l'utilisateur à construire la requête de façon interactive.

Approche : elle se base sur les techniques de datamining qui consistent à rechercher les ensembles fréquents extraits des requêtes précédentes via un fichier LOG en sélectionnant les attributs/mesures intéressantes.

Algorithme : utilisation de l'algorithme Apriori qui consiste à rechercher les règles d'associations des ensembles (Itemsets) fréquents par élagage des ensembles peu fréquents en utilisant les arbres de décisions. En entrée la requête, matrice attribut/mesure et en sortie les itemsets fréquents des requêtes.

Système : Système FIMIOQR (Frequent Itemsets Mining for Interactive OLAP Query Recommendation) en Java dans l'environnement Netbeans

Avantages & Inconvénients : l'avantage majeur c'est l'interactivité et l'assistance dans la construction des requêtes, ce qui rend la tâche plus facile même pour un non informaticien. Comme inconvénient c'est que le système FIMIOQR prend en compte seulement les clauses SQL suivantes : Select, Where, Group By, Having, Order By (ne traite pas certains cas, les requêtes imbriquées par exemple).

5.9. Synthèse des travaux de recommandation des requêtes

Nous dressons ici un tableau comparatif des travaux examinés précédemment. Pour faciliter la lecture et l'interprétation du tableau ci-dessous on utilise l'expression « N.D. » qui signifie qu'aucun élément n'a été défini par rapport au critère énoncé. De ce qui précède, nous pouvons constater les insuffisances suivantes :

Approche : Inexistence d'une approche de personnalisation générique et absence d'une approche mixte qui intègre les deux fonctionnalités de personnalisation de requête et de recommandation pour les analyses OLAP.

Algorithme : Aucun travail ne considère les préférences pour générer les recommandations,

Systeme : A l'exception de [Bellatreche, 2005], tous les travaux supportent un seul langage de requête et sont dépendants de la structure de visualisation.

5.10. Tableau Comparatif des travaux existants

Travaux	Type de personnalisation/ Recommandation (*)		Moment de la recommandation			Objet d'exploitation	
	Cognitif	Collaboratif	Avant	Pendant	Après	Profil	Fichier log
Bellatreche (2005, 2006)	X		X			X	
Ravat (2007, 2008)	X		X	X		X	
Giacometti (2008, 2009)		X	X		X	X	
Golfarelli (2009)	X		X		X	X	
Garrigós (2009)		X		X		X	
H. Jerbi (2012)	X				X	X	
R.Khemiri (2015)	X			X	X		X

Tableau 3: Tableau Comparatif des travaux existants

(*) : En général, les approches de personnalisation et de la recommandation s'utilisent en même temps (recommandation hybride).

5.11. Conclusion

Nous avons abordé dans ce chapitre un état de l'art sur les systèmes de recommandation des requêtes dans les entrepôts de données par la présentation de leurs principes de base, leurs approches de conception, leurs algorithmes et système d'implémentation et enfin les avantages et les inconvénients de chaque travail de recherche.

A la fin, nous avons fait un tableau qui récapitule les différents travaux étudiés à travers quelques indicateurs des systèmes de recommandation.

Partie II

ETUDE PRATIQUE

CHAPITRE VI: CONCEPTION

CHAPITRE VII : REALISATION & IMPLEMENTATION

CHAPITRE VI

CONCEPTION

“No matter how good the team or how efficient the methodology, if we’re not solving the right problem, the project fails”

-Woody Williams-

6. Conception

6.1. Introduction

Pour répondre aux besoins des utilisateurs à interroger efficacement un entrepôt de données en vue d'une prise de décision afin d'améliorer le processus d'activité d'une part et bien maîtriser le budget d'autre part. Nous sommes devant un problème complexe vis-à-vis le nombre énorme des résultats des requêtes à exploiter car l'utilisateur n'est pas forcément un informaticien ou un expert d'un domaine donné pour bien préciser ses besoins donc nous traitons ici un problème d'ambiguïté, d'imprécision ou flou.

En d'autres termes, L'utilisateur peut alors restreindre ses critères de recherche mais prend alors le risque de n'obtenir que peu voir aucune réponse ou alors il les élargit et prend le risque d'avoir un nombre important de réponses sans aucune préférence entre celles-ci.

Notre tâche consiste à réaliser un système qui gère l'imprécision de quelques termes flous dans le langage MDX. En outre, il permet la recommandation efficace des requêtes aux décideurs en affichant les parties intéressantes suivant leurs profils pour avoir une meilleure prise de décision. Cela est possible à l'aide de l'union de deux notions théoriques : personnalisation / recommandation de filtrage collaboratif

En d'autres termes, la source de personnalisation/recommandation peut être sous forme de préférences utilisateur après la soumission des requêtes floues.

Afin de bien comprendre la solution proposée, nous illustrons ci-dessous les principales notions et termes utilisés dans notre approche de conception.

6.2. Interrogation des Requêtes

6.2.1. Langage d'Interrogation Flou

Le langage SQL s'est imposé comme un standard d'interrogation dans la plupart des Systèmes de Gestion de Base de Données Relationnelles (SGBDR) à l'instar de MySQL, qui est un SGBD très efficace mais il lui manque de la flexibilité, tant sur le plan syntaxique que sémantique. Sur ce dernier point, nous proposons une extension de MDX, qui est l'alternative de SQL OLAP afin de nous offrir une flexibilité dans l'interrogation de notre entrepôt.

Cette flexibilité concerne notamment la possibilité d'introduire des conditions vagues dans les requêtes qui apparaissent indispensables lorsque l'utilisateur n'est pas en mesure de traduire de manière booléenne (en « tout ou rien ») ses attentes et contraintes, celles-ci étant elles-mêmes de nature imprécise. La flexibilité introduite dans ce langage d'interrogation traduit également une notion de *préférence* permettant une distinction qualitative des éléments sélectionnés et ainsi d'ordonner les réponses par leur degré de satisfaction vis-à-vis des préférences de l'utilisateur. En fonction d'un nombre attendu de résultats, ces préférences sont également exploitées pour suggérer des réponses indirectes.

La théorie des ensembles flous est utilisée comme un cadre formel pour étendre les opérateurs ensemblistes (intersection, union, différence) et relationnels (sélection, projection, jointure, division) utilisés dans SQL dans des relations graduelles ou floues.

Notre système doit permettre l'interrogation de l'entrepôt BAC avec la prise en compte de quelques termes flous. Plus précisément il sera capable au final de réaliser les mêmes opérations que celles possibles pour un SGBD classique, tout en introduisant en plus la possibilité d'interroger ceux-ci à partir

de questions imprécises (ou floues). L'avantage d'ajouter de l'imprécision est d'éviter une trop haute sensibilité au choix des paramètres d'une part et de permettre aussi de filtrer les résultats selon leur degré de correspondance à une requête posée au système ou d'introduire un critère de préférence. Ceci n'est pas possible avec un système classique basé sur la logique booléenne qui permet uniquement de répondre oui ou non à une question.

Soit par exemple un attribut résultat qui décrit l'admission d'un candidat au BAC par ses propriétés 'Admis' ou 'Non admis'. Ces deux qualificatifs correspondent à des notions vagues et subjectives modélisées dans le système par des ensembles flous, chacun étant caractérisé par une fonction d'appartenance. Cette fonction d'appartenance qualifie par un score normalisé dans l'intervalle $[0,1]$ l'appartenance d'un tuple à ce prédicat. Ainsi, contrairement à SQL ou MDX qui repose sur une logique booléenne, un tuple appartiendra de manière graduelle à l'ensemble des réponses. Ce score d'appartenance offre directement une information utile pour le classement qualitatif des réponses vis-à-vis de la requête flexible. Outre, la prise en compte de prédicats flous, le système introduit des modificateurs qui expriment des adverbes de la langue naturelle, tels que : *élevé, moyen, faible*, relatifs aux taux de succès des résultats du BAC.

Le résultat d'une requête ici n'est plus une relation classique, mais une relation floue où chaque tuple est associé à un degré de satisfaction des conditions spécifiées dans la requête MDX. Pour résumer, nous pouvons dire que l'utilisation de termes flous est plus souple et plus proche du raisonnement humain. Nous ne cherchons plus à savoir si des éléments satisfont ou pas une condition, mais à quel degré ils la satisfont. Nous allons maintenant voir comment modéliser un terme flou.

6.2.2. Modélisation d'un terme flou

Un terme flou correspond à un adjectif de la langue naturelle. Pour modéliser un terme flou, une fonction trapézoïdale est utilisée. Il reste pour l'utilisateur du terme flou à définir les paramètres caractérisant cette fonction.

Exemple : Soit le terme flou “admis” appliqué à un candidat. Une définition de ce terme peut être qu'un candidat est admis si son résultat est supérieur à 9.50, puis de plus en plus admis entre 9.50 et 10 et admis au-delà.

6.2.3. Requêtes MDX Floues

Nous présentons ici la partie floue dans la clause **SELECT ON ROWS** dans le langage **MDX**, comme suit :

```
SELECT<Mesures> On Columns;  
<Members pouvant être floues>On Rows  
FROM<Cube de données>  
WHERE<conditions > ou Slicer
```

Dans notre cas d'étude, nous avons opté pour le langage MDX au lieu du SQL standard car il offre plusieurs avantages, à savoir : temps d'exécution optimisé, facilité dans la navigation du cube de données, ...

6.3. Serveur OLAP MONDRIAN

6.3.1. Introduction

Mondrian est un moteur **OLAP** (**O**n**L**ine **A**nalytical **P**rocessing) écrit en Java par **Julian Hyde** en 2001 qui permet la conception, la publication et le requêtage de cubes multidimensionnels.

Mondrian permet l'exécution de requêtes en langage MDX seulement sur des entrepôts de données s'appuyant sur des SGBDR comme MySQL, PostgreSQL, Oracle ou Microsoft SQL Server d'où sa caractérisation de « **ROLAP** » (Relational OLAP). En matière de **ROLAP**, **Mondrian est la référence Open Source**.

Mondrian permet d'accéder aux résultats dans un format multidimensionnel (cube de données) compréhensible par une API de présentation côté client, le plus souvent utilisé par exemple en mode Web est : **JPivot**.

6.3.2. Architecture de Mondrian

Pour bien comprendre les fonctionnalités et les caractéristiques de Mondrian, il faut maîtriser quelques concepts de base de l'analyse multidimensionnelle et les termes associés : notion de cubes, modélisation en étoile et en flocon, tables de faits et des dimensions, mesures, hiérarchies, niveaux, membres, ...

Mondrian s'appuie sur une modélisation OLAP standard et peut donc se connecter à n'importe quel entrepôt de données conçu dans les règles de l'art de la Business Intelligence.

Il est intéressant de noter que Mondrian est le composant OLAP utilisé par la plupart des suites de BI Open Source notamment **Pentaho**, **JasperServer** et **SpagoBI**.

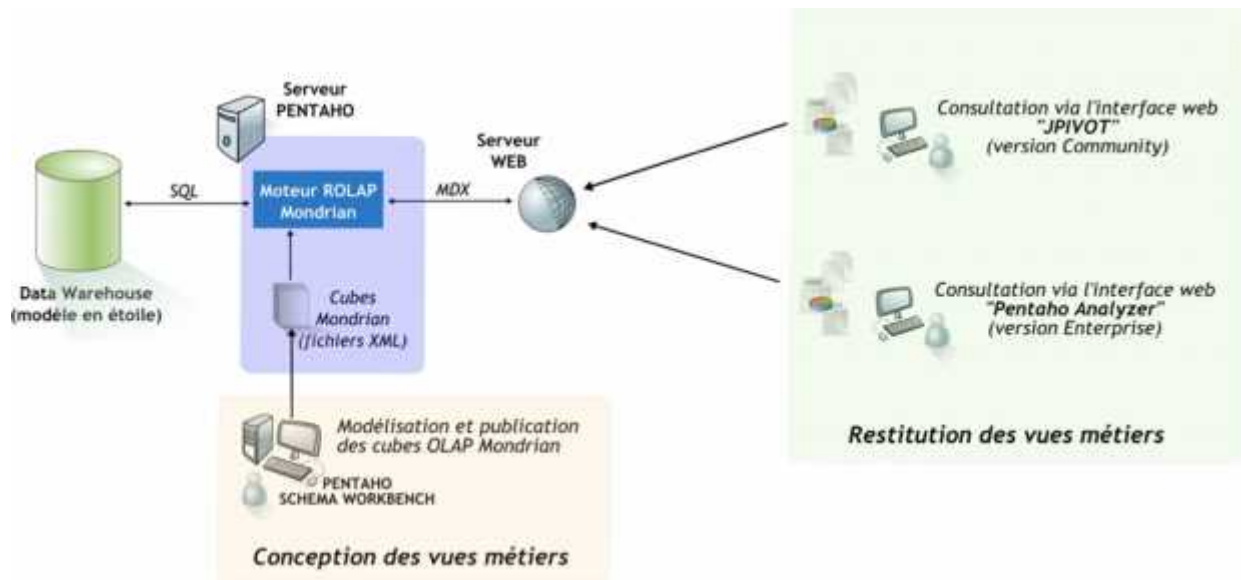


Figure 5: Architecture BI

Les éléments de l'architecture présentée ci-dessus sont les suivants :

Datawarehouse avec stockage des données au format OLAP. Les schémas en étoile et en flocons sont gérés par Mondrian, ainsi que de nombreux autres concepts propres à l'OLAP ;

Serveur web J2EE Pentaho ou Apache TOMCAT. Ce dernier embarque le moteur ROLAP Mondrian qui permet d'effectuer des requêtes multidimensionnelles (langage MDX) sur des données stockées dans un SGBD relationnel (interrogation SQL) ;

Pour la partie conception, les cubes sont modélisés avec « *Pentaho Schema Workbench* », un client riche open source (en xml) de modélisation et de publication de schémas Mondrian sur un serveur Apache ;

Pour la partie restitution, les vues métiers sont accessibles en client léger via un navigateur web (Internet Explorer, Firefox, Google Chrome, ...) afin d'effectuer

des analyses avancées en utilisant le plus célèbre client OLAP en open source qui est : **JPivot**, qui reste un outil de requêtage puissant et stable.

Mondrian s'appuie sur des **schémas XML** pour la définition des cubes.

Un **schéma Mondrian** permet donc de définir le modèle logique ainsi que le mapping sur le modèle physique :

- ✓ Le modèle logique décrit les cubes, les dimensions, les hiérarchies, les niveaux et les membres sur lesquels vont s'appuyer les requêtes MDX ;
- ✓ Le modèle physique correspond à la source de données sur laquelle s'appuie le modèle logique (le modèle en étoile et/ou flocon).

Les schémas Mondrian sont créés par les concepteurs et analystes avec l'outil « Pentaho Schema Workbench » dont une description est donnée dans le paragraphe suivant.

6.3.3. Pentaho Schema Workbench

Pentaho Schema Workbench est un client riche open source (écrit en Java) qui permet de créer des schémas de cubes sans avoir à connaître la syntaxe XML de Mondrian. Cet outil présente une interface avancée qui permet d'effectuer les actions suivantes :

- Connexion au Datawarehouse (via JDBC)
- Création de schémas et cubes Mondrian
- Création des mesures, dimensions et hiérarchies d'un cube
- Jointures pour les tables floconnées
- Tables d'agrégation à utiliser (le cas échéant)
- Création de membres calculés, de dimensions partagées, de cubes virtuels

- Définition des rôles (gestion de la sécurité d'accès à l'intérieur d'un cube)
- Publication des cubes sur le serveur Pentaho

6.3.4. Interface de requêtage web (JPivot)

JPivot est sans aucun doute l'interface web de requêtage open source pour Mondrian la plus utilisée.

Pour donner un léger aperçu sur l'utilisation de JPivot, nous illustrons ci-après l'interrogation de la base démo donnée avec Mondrian **FoodMart** en présentant les outils et étapes nécessaires à l'installation de l'OLAP SERVER Mondrian.

6.3.5. Installation de Mondrian

Pour pouvoir utiliser Mondrian, nous devons installer les outils suivants :

- Serveur d'application Apache TomCat 7.0.65
- Java Kit Développement : jdk-8u73.exe
- Mondrian 3.4.1.zip (qui contient mondrian.war, le fichier à redéployer sur le serveur)
- SGBDR : Mysql server 5.5
- Le pilote de connection de la BDD : mysql-connector-java-5.1.6-bin.jar (à copier dans le dossier : *C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib*)
- La base exemple Foodmart : foodmart_mysql.sql

Après l'installation de ces outils selon l'ordre précité sous Win 7 64 bits par exemple, nous pouvons exécuter Mondrian via la saisie dans le navigateur

Web : <http://localhost:8080/mondrian>, et ainsi s'affiche la page principale comme suit :

Mondrian examples

- [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](#)
- [XMLA for Analytics tests](#)

Other links

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

Figure 6: la page principale mondrian

Pour pouvoir interroger la base FoodMart, nous devons changer les paramètres des fichiers JSP suivants : `mondrian.jsp`, `fourheir.jsp`, `arrow.jsp`, `color.jsp`, qui se trouvent dans le dossier : « *C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\mondrian\WEB-INF\queries* » comme suit :

```
jdbcDriver="sun.jdbc.odbc.JdbcOdbcDriver"
```

```
jdbcUrl="jdbc:odbc:MondrianFoodMart"
```

```
catalogUri="/WEB-INF/queries/FoodMart.xml">
```

Par :

```
jdbcDriver="com.mysql.jdbc.Driver"
```

```
jdbcUrl="jdbc:mysql://localhost/Foodmart?user=foodmart&password=foodmart"
```

```
catalogUri="/WEB-INF/queries/FoodMart.xml">
```

Ainsi que le fichier web.xml comme suit :

```
Provider=mondrian; Jdbc=jdbc:odbc:MondrianFoodMart;Catalog=/WEB-INF/queries/FoodMart.xml;
```

```
JdbcDrivers=sun.jdbc.odbc.JdbcOdbcDriver;Par:
```

```
Provider=mondrian; Jdbc=jdbc:mysql://localhost/foodmart?user=foodmart;password=foodmart;
```

```
Catalog=/WEB-INF/queries/FoodMart.xml;JdbcDrivers=com.mysql.jdbc.Driver;
```

Après le lancement de Mondrian (<http://localhost:8080/mondrian>), nous choisissons le 1^{er} lien (**JPivot pivot table**) qui correspond à la requête MDX du fichier *mondrian.jsp* suivante :

```
select
```

```
  {[Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store Sales]} on columns,
```

```
  {([Promotion Media].[All Media], [Product].[All Products])} ON rows
```

```
from Sales
```

```
where ([Time].[1997])
```

Et après quelques secondes, le résultat suivant est affiché et via le menu de Mondrian en haut nous pouvons facilement manipuler et analyser notre cube OLAP en utilisant les opérations de Slice, Dice, Drill Down, Roll Up, ... etc

Test Query uses Mondrian OLAP

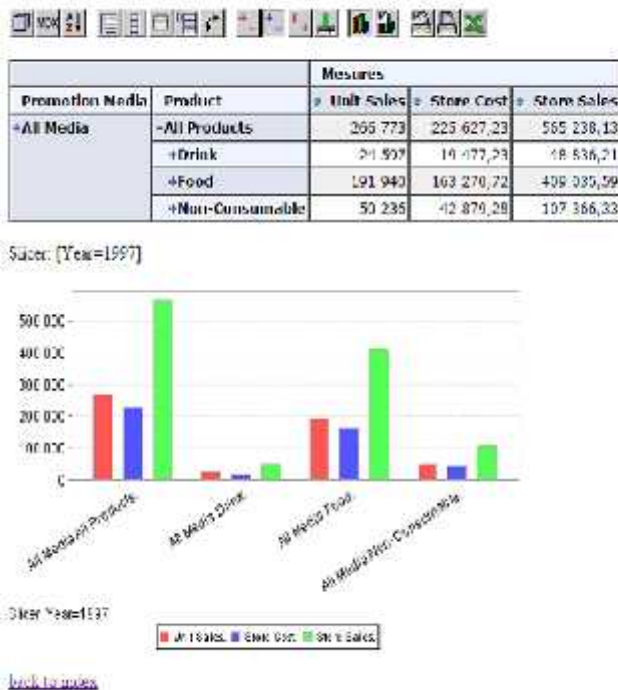


Figure 7: Résultat de requête test

C'est le même principe pour les autres liens des requêtes MDX.

6.4. Le Langage MDX

6.4.1. Définition

MDX (**M**ulti**D**imensional **eX**pressions) est le langage de requête dédié aux bases de données multidimensionnelles ou pour les SGBDR OLAP. C'est l'équivalent du SQL aux BDR. Il a son propre vocabulaire, lui permettant de manipuler les dimensions, les hiérarchies, les niveaux et les membres.

Dans ce qui suit, nous présentons les termes courants, les fonctions de base et avancées, et quelques exemples liés au MDX.

6.4.2. Les concepts

Membre : un membre est une instance d'un niveau. Par exemple « *Bien* » serait un membre du niveau « *DesMention* » sur une dimension « *Mention* ». Il est désigné soit par son nom, soit par son nom unique, valeurs entre crochets. Si l'on reprend l'exemple précédent, le nom du membre « *Bien* » est [*Bien*], et son nom unique serait [*Mention*].[*DesMention*].[*Bien*]. Dans ce dernier cas, on retrouve respectivement le nom de la dimension, le niveau, et enfin le membre.

N-uplet (Tuple) : une cellule résultant d'un croisement de dimensions. Il est reconnaissable par les parenthèses qui l'entourent. Chaque membre est séparé par une virgule. Par exemple, le tuple ([*Bien*].[*2015*]) croisera les données du membre de la dimension *Mention* qui est « *Bien* » et du membre de la dimension *année* « *2015* ».

Set : un ensemble ordonné de *n*-uplets préfixé et suffixé par des crochets. Les éléments qui le composent sont séparés par une virgule. Voici un exemple :
 {([*Bien*].[*2015*]), ([*Mesure*].[*Taux*],[*Lycee*].[*Commune*]) }

6.4.3. Construction d'une requête MDX

Une requête MDX est construite pour répondre à une question multidimensionnelle. Elle précise donc les axes sur lesquels les éléments (membres,tuples,sets) sont définis. Le format d'une requête MDX est :

```
SELECT elementsDeLaxeColonne ON COLUMNS,
      elementsDeLaxeLigne ON ROWS
FROM nomDuCube
WHERE elementsDuSlicer
```

Notez qu'au lieu de COLUMNS et de ROWS, nous pouvons faire appel à Axis (0) et Axis (1). Le *slicer* est le pseudo axe permettant de filtrer les données.

6.4.4. Fonctions

a. Sur les membres

Il est possible d'appliquer des fonctions sur les membres, niveaux et dimensions. Ceux-ci permettent, une nouvelle fois, de parcourir les éléments les composant.

- *members* : permet de récupérer les membres du niveau en cours. Ne s'applique que sur une dimension ou un niveau ;
- *children* : permet de récupérer les membres sous-jacents au membre en cours. Ne s'applique que sur un membre ;
- *currentMember* : récupère le membre en cours. S'applique sur les trois types d'éléments ;
- *parent* : retourne l'élément hiérarchiquement supérieur ;
- *nextMember* : retourne l'élément suivant, dans la même hiérarchie. Par exemple : `[Annee].[2013].nextMember` renverra sans doute `[Annee].[2014]`. Ne s'applique que sur un membre ;
- *previousMember* : retourne l'élément précédent, dans la même hiérarchie. Ne s'applique que sur un membre ;
- *firstSibling* : retourne le premier enfant du parent du membre courant. Par exemple : `[Annee].[2015].[Aout].firstSibling` renverra `[Annee].[2015].[Janvier]`. Ne s'applique que sur un membre ;
- *lastSibling* : retourne le dernier enfant du parent du membre courant. Ne s'applique que sur un membre ;

- *firstChild* : retourne le premier enfant du membre courant. Par exemple: `[Annee].[2015].firstChild` renverra `[Annee].[2015].[Janvier]`. Ne s'applique que sur un membre ;
- *lastChild* : retourne le dernier enfant du membre courant. Ne s'applique que sur un membre.

Il en existe bien d'autres sur ces éléments (*cousin*, *item*, *lag*, etc).

b. Sur les sets

Les sets ont eux aussi leurs fonctions.

- *head* : retourne les éléments de tête du set ;
- *tail* : retourne les éléments de pied du set ;
- *subset* : retourne un sous-ensemble du set ;
- *topcount* : retourne les x éléments du set qui ont la plus grande valeur ;
- *order* : retourne le set trié selon la méthode passée en paramètre ;
- *filter* : retourne un set filtré des éléments satisfaisant la condition énoncée ;
- *union* : retourne un set résultat de l'union des deux sets passés ;
- *intersect* : retourne un set résultat de l'intersection des deux sets passés ;
- *crossjoin* : tableau croisé des sets. Il s'agit du produit cartésien des membres des sets qu'il contient ;
- *hierarchize* : hiérarchisation des sets. Implique que les sets soient de la même dimension.

Exemple : `Hierarchize({[Lycee].[Commune]}, {[Lycee].[DesLycee]})` ramènera le résultat suivant :

Ain Defla
Lycee Ibn Khadoune
LyceeMalek Ben Nabie
Khemis
Lycee Emir Aek
LyceeBougerra Si Mhamed
LyceeKouidri Mohamed

6.4.5. Exemple de fonctions avancées MDX

Les problèmes commencent à apparaître au moment où l'on veut croiser les données de plusieurs dimensions. Le recours à la fonction CrossJoin est alors nécessaire.

SELECT

CrossJoin({[Annee].members},

{[Lycee].[Commune].members}) ON COLUMNS,

{[Specialite].members } ON ROWS

FROM [CubeBacDW]

6.5. Conclusion

Pendant ce chapitre nous avons pu recenser les différents outils nécessaires à la réalisation et à l'implémentation de la solution proposée. Nous avons donné un aperçu global sur l'intégration des termes flous dans l'interrogation des requêtes ainsi que sur l'utilisation du langage MDX et le serveur OLAP Mondrian.

CHAPITRE VII

REALISATION &

IMPLEMENTATION

“To provide a simple scientific solution to complex design problems is the job of an Engineer”

-Bhupesh B. Patil-

“Your most unhappy customers are your greatest source of learning”

-Bill Gates-

7. Réalisation & Implémentation

7.1. Introduction

Après la définition des outils nécessaires dans le chapitre conception, nous présentons dans ce chapitre comment réaliser notre solution et comment l'implémenter.

7.2. Modèle du Data Warehouse BAC

Nous avons opté pour un schéma en étoile afin d'implémenter notre DW BAC, ce qui correspond mieux à nos besoins d'analyses, qui sont dans notre cas : **le taux de succès BAC par Lycée** selon différentes approches, à savoir :

- Approche Multidimensionnelle Relationnelle (ROLAP MDX QUERY),
- Approche Multidimensionnelle Floue (FUZZY MDX QUERY),
- Approche Multidimensionnelle basée sur la Recommandation (RECOMMENDED MDX QUERY).

Le choix de ce modèle en étoile est basé sur les avantages suivants :

- Facilité de navigation,
- Performances d'accès élevées : nombre de jointures limité,
- Gestion facile des agrégats.

Notre modèle du DW BAC se présente comme suit :

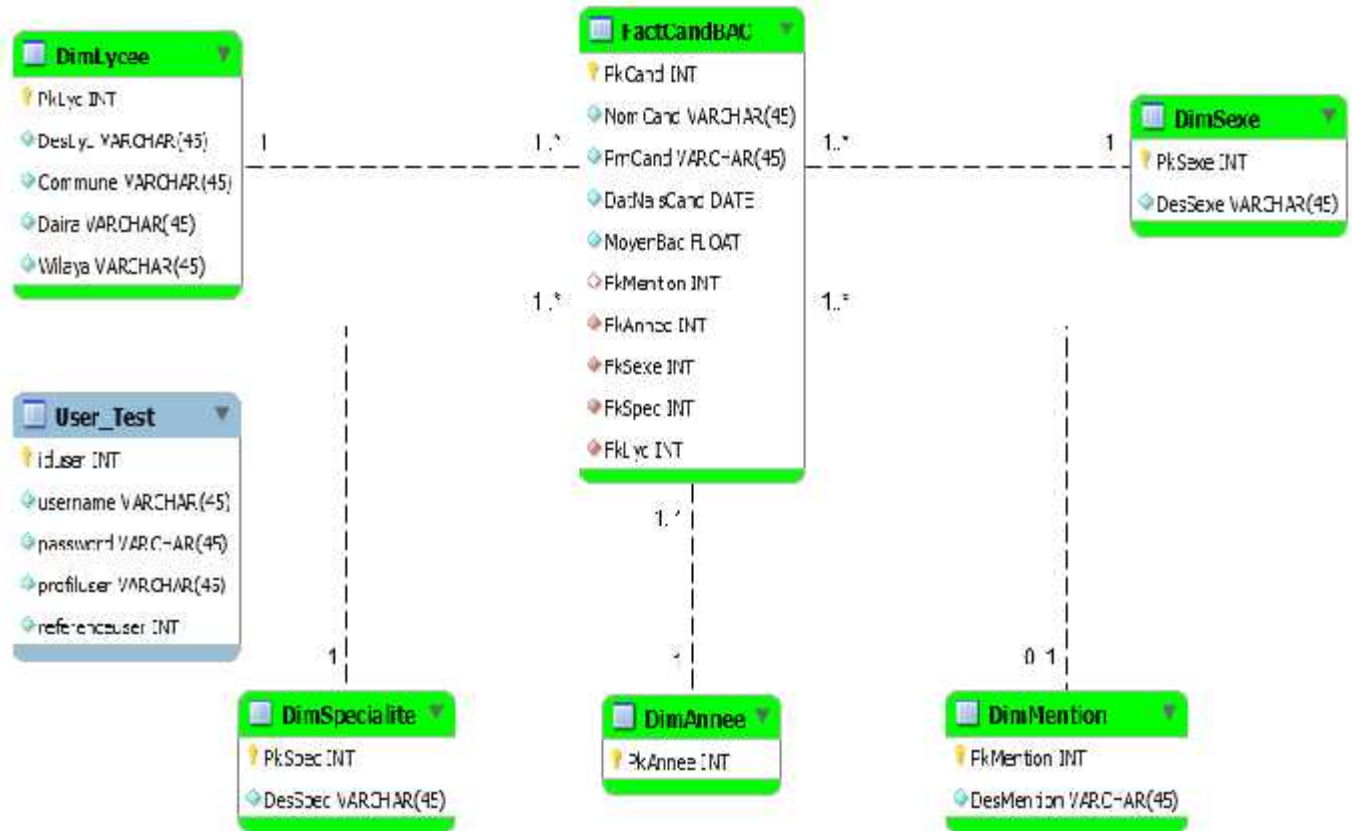


Figure 8: Modèle du DW BAC

Dans ce modèle, nous avons une seule table de fait « **FactCandBAC** » et cinq tables de dimensions : « **DimLycee** », « **DimSpecialite** », « **DimAnnee** », « **DimMention** », « **DimSexe** ». En plus une méta table « **User_Test** » qui gère la navigation dans le cube selon le profil utilisateur (Elle gère essentiellement la partie recommandation).

La table de fait contient comme mesures : « **NomCand** », « **PrnCand** », « **DatNaisCand** » qui peuvent être utilisés pour d'autres sujets d'analyse mais les mesures importantes dans notre cas d'étude sont : « **MoyenBAC** » et « **PkCand** ». Pour lesquelles, nous faisons des agrégations par **Sum** et **Count** afin de pouvoir calculer notre mesure Taux de succès du BAC selon les axes

« Lycee », « Specialite », « sexe », « mention » ou même sur l'axe « année » qui est utilisé ici comme un Slicer. Le Taux de succès du BAC se calcule comme suit :

Taux de succès=(Nombre de Candidats Réussis/Nombre de Candidats Inscrits)*100

Après la phase d'authentification de décideur à travers le renseignement du son nom d'utilisateur et son mot passe et si tout va bien, le menu principal de l'application «*BI Analysis for BAC Succes* » s'affiche.

Chaque lien dans le menu principal représente une requête MDX qui s'exécute sur le serveur OLAP Mondrian.

7.3. Présentation de la solution

Afin d'éviter la génération des résultats vides ou des ensembles importants de résultat suite à l'interrogation des entrepôts de données relationnels via l'utilisation du langage SQL, nous proposons ici une solution qui évite ces deux problèmes d'une part et permettre à l'utilisateur de naviguer d'une façon personnalisée dans une partie de cube qui l'intéresse via le langage MDX d'autre part, en se basant sur les préférences utilisateurs.

Pour réaliser cela, nous avons créé une méta table nommée « **User_Test** », qui doit être isolée par rapport à notre entrepôt de données (Table de fait et tables des dimensions).

Notre démarche d'implémentation de la solution proposée commence tout d'abord par le chargement du DATA WEREHOUSE via un ETL d'une façon manuelle en utilisant des fonctions avancées de Microsoft Excel.

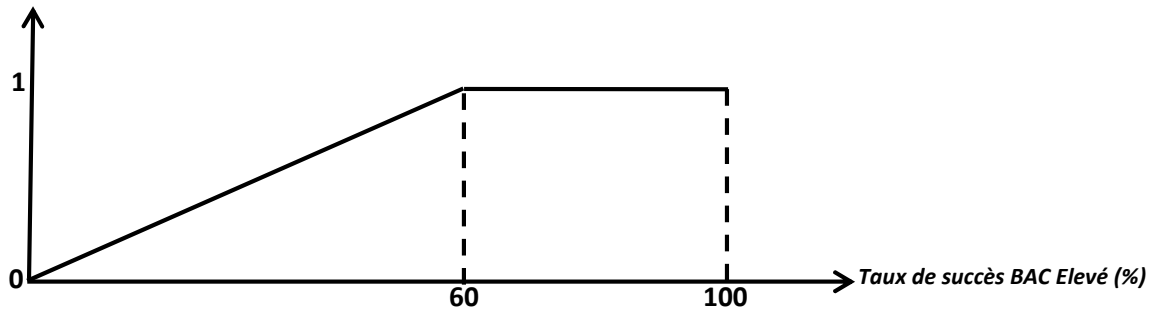
Après cela, nous allons construire le cube de données OLAP à analyser par l'outil Open Source PENTAHO SCHEMA WORKBENCH, en créant la table des faits, les dimensions, les hiérarchies, les niveaux, les membres et les mesures par lesquelles nous pouvons analyser les besoins du décideur. Dans notre cas d'étude, la mesure à analyser est : **Le taux de succès du BAC par Lycée** avec ou sans d'autres dimensions (Sexe, Spécialité, Année, Mention).

Nous avons aussi intégré dans notre solution quelques termes flous. Dans notre cas, le langage MDX peut interpréter trois genres de termes flous pour notre mesure, à savoir : Taux de succès **ELEVE, MOYEN, FAIBLE**.

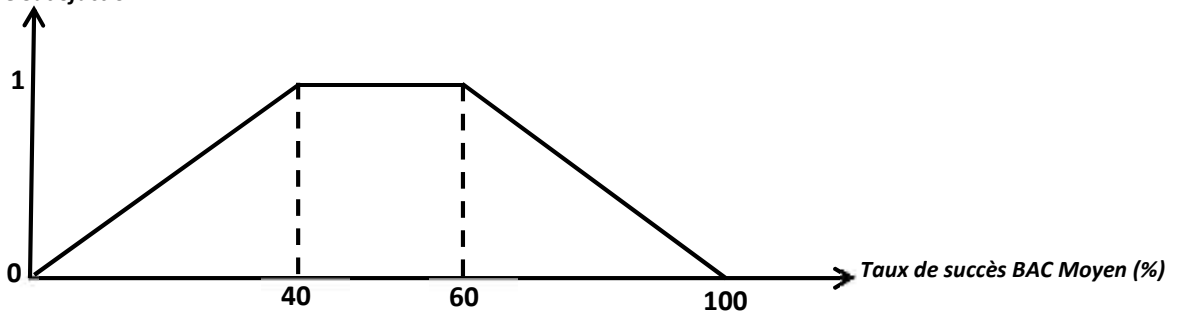
Le but souhaité des requêtes à terme flou ou les requêtes flexibles est de classer les résultats obtenus selon un degré de satisfaction D et la requête MDX nous affiche les n meilleurs (préférés) éléments de l'ensemble d'éléments possible qui satisfait suffisamment les prédicats graduels (Elevé, Moyen, Faible). Dans notre cas la requête MDX affiche les n tuples dont le taux de succès BAC soit « Elevé » si et seulement si le taux est strictement supérieur à 0.6 (60%), et entre 0.4 et 0.6 pour le prédicat « Moyen », et strictement inférieur à 0.4 (40%) pour le prédicat « Faible ».

A la lumière de ce qui précède, nous définissons les trois fonctions d'appartenance : $f_{\text{elevé}}(x)$, $f_{\text{moyen}}(x)$, $f_{\text{faible}}(x)$ relatives aux trois prédicats flous 'Elevé', 'Moyen', 'Faible' avec leur degré de satisfaction $D_{\text{elevé}}$, D_{moyen} et D_{faible} respectivement comme il est illustré dans les figures suivantes :

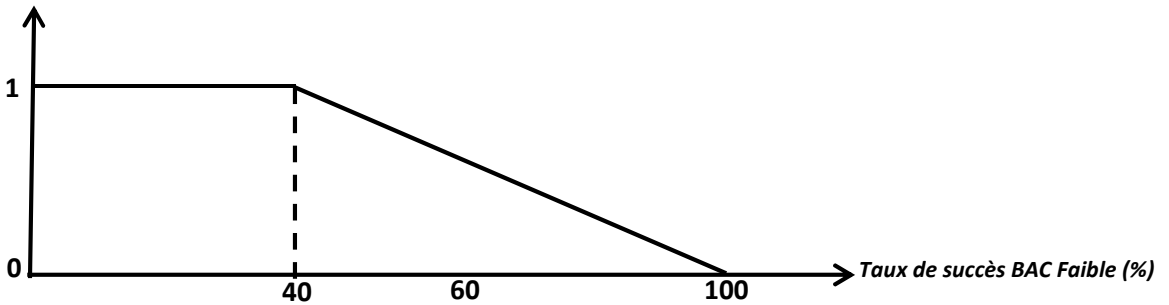
Degré de Satisfaction



Degré de Satisfaction



Degré de Satisfaction



Ainsi, le taux de succès BAC peut être modélisé par l'ensemble flou :

$$E = \{(Elevé > 0.6), (0.4 > Moyen \geq 0.6), (Faible \leq 0.4)\}.$$

Et les fonctions d'appartenance peuvent être modélisées comme suit :

$$f_{elevé}(x) = [0, 1] \text{ si } x \in [60, 100]$$

$$f_{moyen}(x) = [0, 1] \text{ si } x \in [40, 60]$$

$$f_{faible}(x) = [0, 1] \text{ si } x \in [0, 40]$$

En outre, notre approche développée ici est différente des travaux réalisés auparavant dans la littérature des systèmes de recommandations des requêtes dans un contexte décisionnel dans le sens où notre solution permet de naviguer dans des sous cubes adaptés à chaque type de profil utilisateur et d'en faire des analyses OLAP en utilisant le langage MDX. Sur la base du profil utilisateur, la navigation multidimensionnelle est personnalisée. Nous recommandons au décideur une vue matérialisée adéquate à ses préférences. Nous utilisons dans notre solution la recommandation de filtrage collaboratif selon trois groupes utilisateurs (**ADMIN, DIRECTEUR WILAYA, DIRECTEUR LYCEE**) et nous classons chaque nouvel utilisateur dans un groupe adéquat suivant deux paramètres : sa **fonction** et sa **responsabilité géographique**. Ainsi, notre méta table «**User_Test** » a la structure suivante : **iduser, username, password, profiluser**(fonction), **referenceuser**(responsabilité géographique).

La gestion des préférences utilisateurs s'effectue d'une manière implicite par la vérification des préférences de l'utilisateur courant par rapport aux trois groupes (classes) d'utilisateurs prédéfinis dans la méta table et grâce à un algorithme de recommandation (voir figure ci-dessous) qui permet de modifier dynamiquement le schéma XML du cube OLAP suivant le profil à chaque accès utilisateur, nous obtiendrons comme résultat d'analyse une vue matérialisée du cube exploitable par l'utilisateur pour lui aider dans son processus de prise de décision. Nous illustrons ci-après l'algorithme de recommandation à base de profil utilisateur par le diagramme suivant :

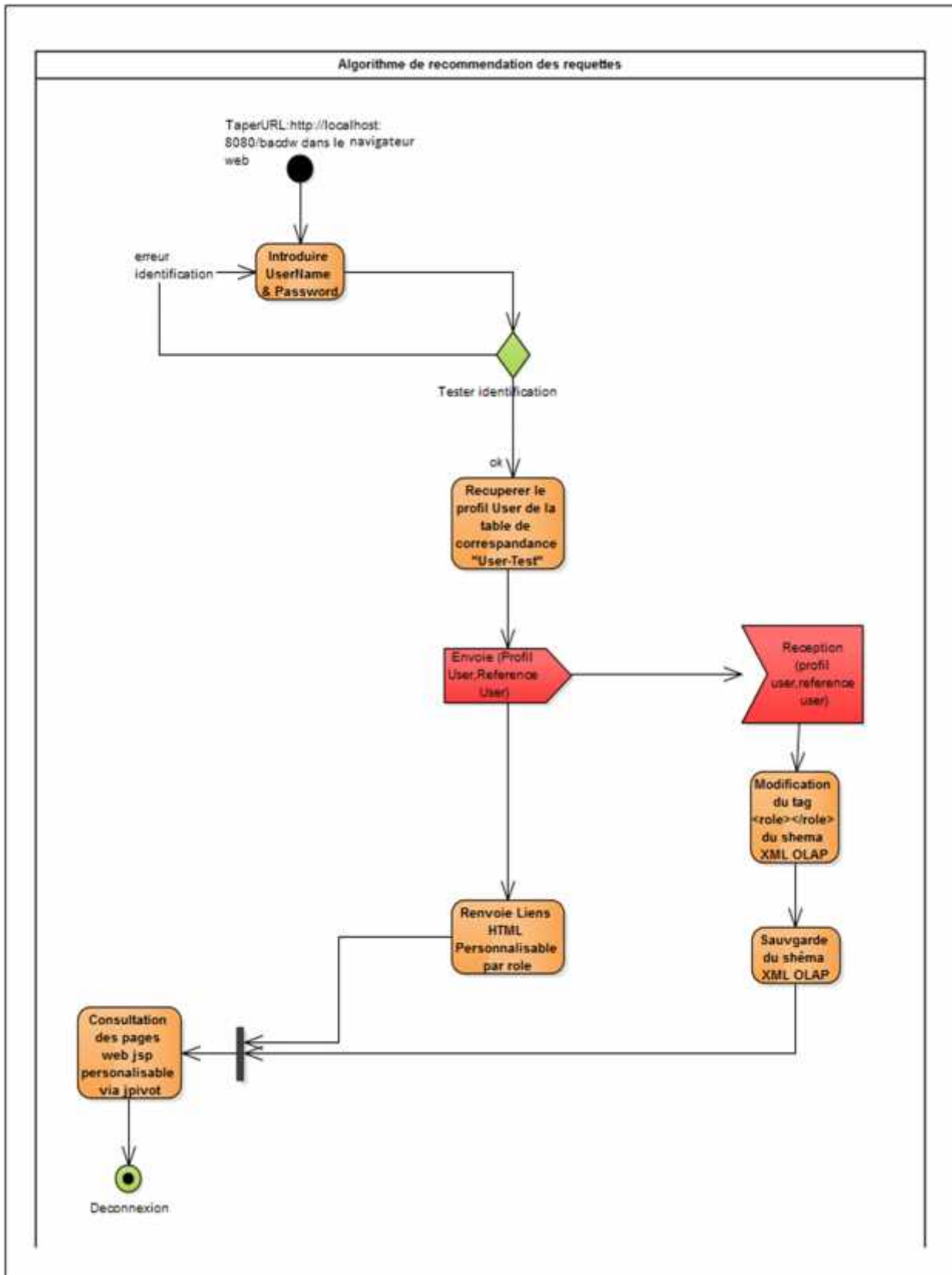


Figure 9 : Algorithme de recommandation à base de profil utilisateur

7.4. Architecture Logicielle Utilisée

L'architecture logicielle utilisée pour implémenter notre solution est composée essentiellement de logiciels « open-source ». Nous avons utilisé le SGBD MySQL pour stocker la base de données relationnelle de l'entrepôt qui va être interrogée via le langage MDX par un client OLAP JPIVOT, lui qui utilise le serveur OLAP MONDRIAN et tous les deux sont écrit en Java.

Les cubes OLAP MONDRIAN sont définis selon un schéma multidimensionnel spécifique (Etoile) dans un fichier XML préalablement élaboré. Ce cube MONDRIAN peut être interrogé grâce à un client spécifique, ici nous avons utilisé un qui est intégré, c'est JPIVOT.

7.5. Fonctionnement Général

Une fois le cube MONDRIAN spécifié dans un fichier en XML, il est rempli par les données de l'entrepôt, le client JPIVOT et le serveur MONDRIAN fonctionnent ainsi :

1. Le client JPIVOT transmet une requête au format MDX écrite par l'utilisateur dans un fichier JSP au serveur MONDRIAN,
2. Après validation du format de la requête MDX, MONDRIAN utilise le schéma XML pour transformer la requête MDX en requêtes SQL et passe la main à la couche de requêtage SQL,
3. La couche de requêtage SQL de MONDRIAN accède à la base de données relationnelle physique de l'entrepôt et récupère les données résultantes des requêtes SQL exécutées,

4. Les résultats sont consolidés et transmis au client JPIVOT pour être présentés à l'utilisateur sous forme de tableaux croisés dynamiques ou des graphiques pour mieux répondre aux besoins d'analyses.

Dans notre exemple d'étude relatif à analyser un entrepôt des données des résultats du BAC des 05 dernières années de la wilaya de **AIN DEFLA** peut être généralisé et ainsi utilisé par toutes les wilayas d'Algérie et supervisé par le ministre de l'éducation nationale par exemple, son conseiller ou son chef de cabinet qui aura le profil « ADMIN ».

Nous définissons dans notre logiciel 03 types de profils comme suit :

ADMIN (Administrateur ou Super utilisateur),

DIRECTEUR DE WILAYA,

DIRECTEUR DE LYCEE dans la wilaya,

Cette approche permet la recommandation ainsi que la sécurisation et la confidentialité des données du DW, car par exemple un directeur d'un lycée de la wilaya de CHLEF ne peut pas afficher des données relatives à un autre lycée dans la même Wilaya ou dans une autre Wilaya.

Le schéma de fonctionnement général de la solution proposée se présente comme suit :

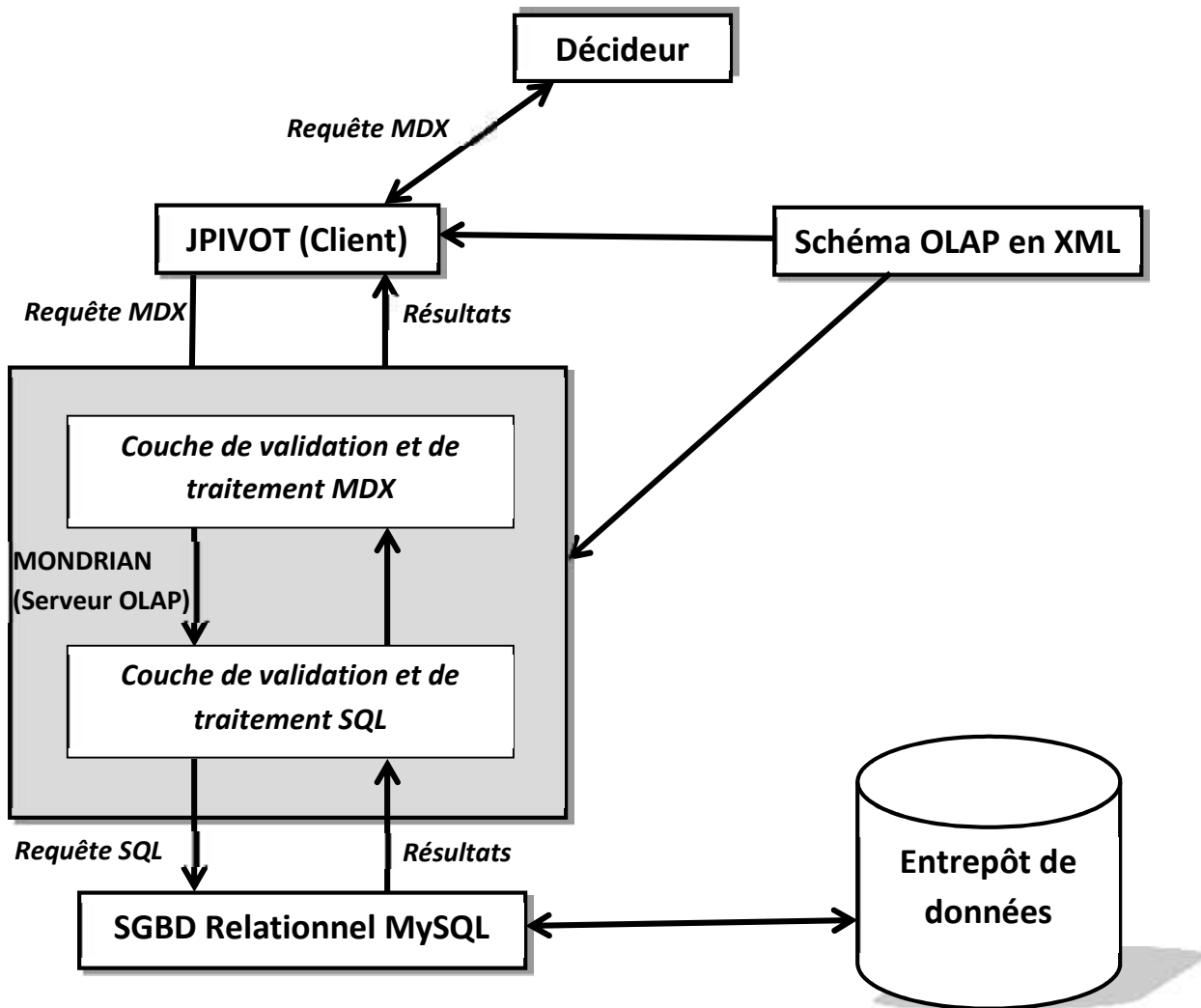
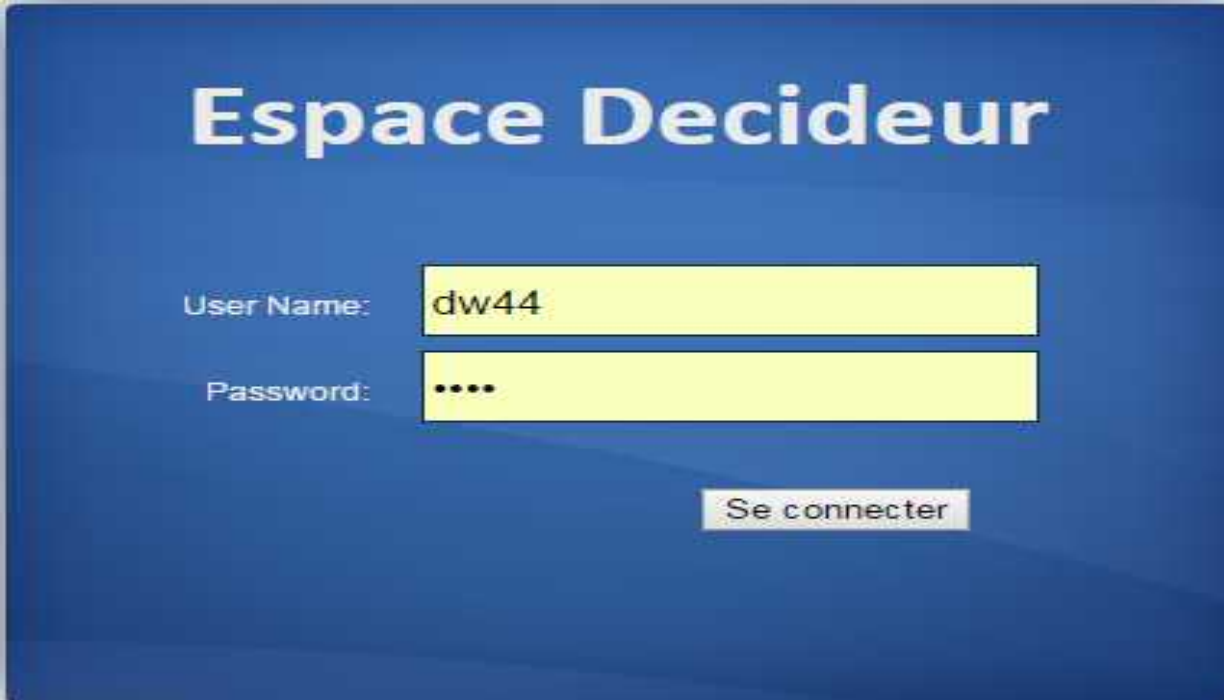


Figure 10: Architecture de la solution utilisée

7.6. Présentation des résultats

Nous présentons dans ce qui suit une démonstration avec des imprimés écrans des différentes fonctionnalités du logiciel réalisé «*BI Analysis for BAC Succes*» pour le **profil directeur de wilaya (profiluser=dw)**.

La première étape est de s'identifier à travers la boîte de dialogue suivante :



The image shows a login dialog box with a blue background. At the top, the text 'Espace Decideur' is displayed in large white letters. Below this, there are two input fields. The first is labeled 'User Name:' and contains the text 'dw44'. The second is labeled 'Password:' and contains four dots. Below the input fields, there is a button labeled 'Se connecter'.

Figure 11: Boîte d'identification

Après la vérification de la validité des informations saisies (Username et password) dans la méta table « user_test », le menu principal s'affiche.

Pour chaque type de profil (admin, dw, dl), nous avons une page principale qui contient l'ensemble des requêtes MDX (ROLAP, FUZZY, RECOMMANDÉ) mais seulement le dernier type de requête (Recommandé)

MDX Query) qui constitue notre objectif d'étude qui diffère entre les trois profils existants. Pour le profil (**dw**) avec la référence (44), le menu principal est :



Figure 12: Le menu principal de logiciel « BI Analysis for BAC Succes »

Nous présentons dans ce qui suit le contenu et le résultat de chaque requête MDX selon le profil utilisateur comme suite :

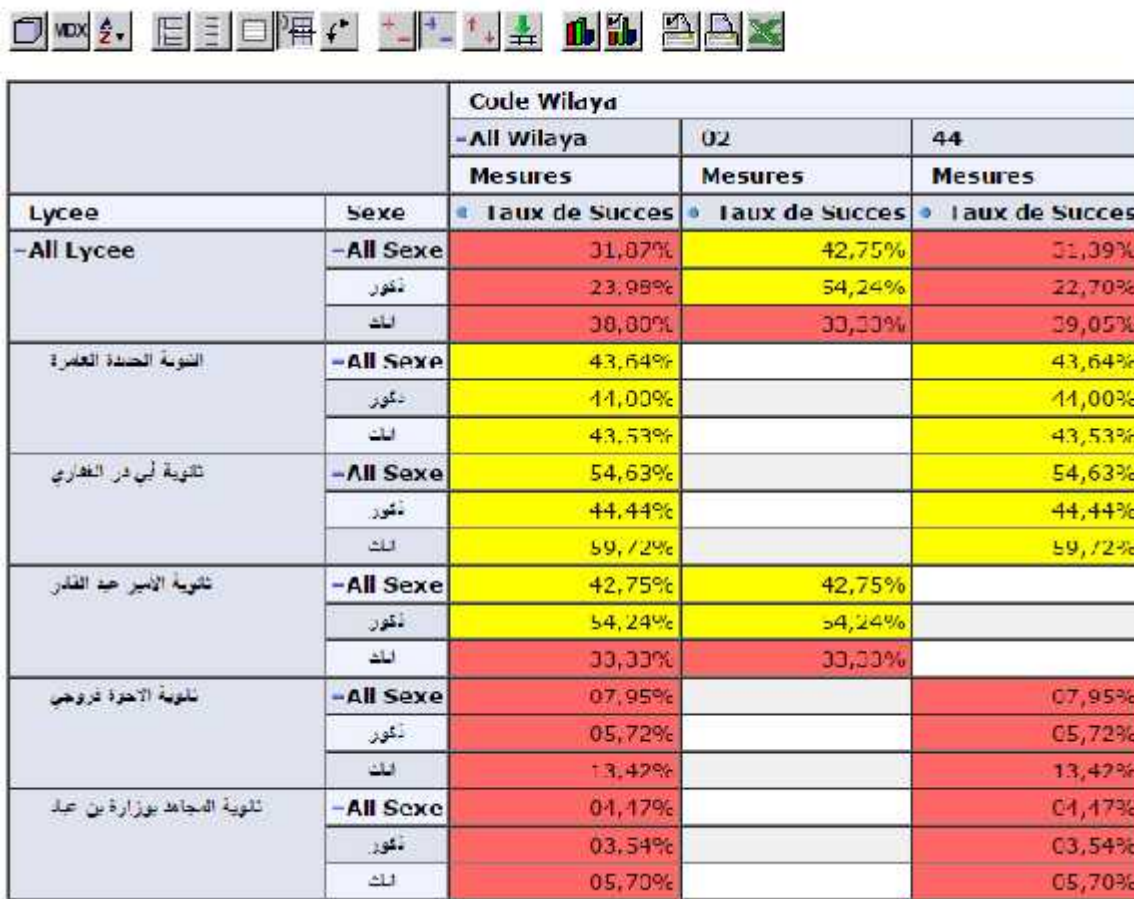
7.6.1. **ROLAP MDX QUERY : Taux de succès BAC /Lycée**

a. **Requête MDX :**

```
SELECT {[Measures].[Taux de Succes]} ON COLUMNS,
{([DesWilaya].[All DesWilaya], [Etablissement.Lycees].[All Lycee], [Sexe].[All Sexe])} ON ROWS
FROM CubeBacDW
WHERE [Annee].[2015]
```

b. **Résultat :**

ROLAP MDX QUERY: Taux de Succes BAC/Lycees et Sexe



		Code Wilaya		
		-All Wilaya	02	44
		Mesures	Mesures	Mesures
Lycee	Sexe	Taux de Succes	Taux de Succes	Taux de Succes
-All Lycee	-All Sexe	31,07%	42,75%	31,39%
	ذكور	23,98%	54,24%	22,70%
	اناث	38,00%	33,33%	39,05%
الثوية الصدة العامرة	-All Sexe	43,64%		43,64%
	ذكور	44,00%		44,00%
	اناث	43,53%		43,53%
ثلوية لى در الفغارى	-All Sexe	54,63%		54,63%
	ذكور	44,44%		44,44%
	اناث	59,72%		59,72%
ثلوية الامير عبد القادر	-All Sexe	42,75%	42,75%	
	ذكور	54,24%	54,24%	
	اناث	33,33%	33,33%	
ثلوية الاحود قروچى	-All Sexe	07,95%		07,95%
	ذكور	05,72%		05,72%
	اناث	13,42%		13,42%
ثلوية المجاهد بوزاردين عماد	-All Sexe	04,17%		04,17%
	ذكور	03,54%		03,54%
	اناث	05,70%		05,70%

Figure 13: ROLAP MDX QUERY : Taux de succès BAC /Lycée

7.6.2. FUZZY MDX QUERY : Taux de succès du BAC Elevé/Lycee

a. Requête MDX :

```
SELECT {[Measures].[Taux de Succes]} ON COLUMNS,
FILTER([Etablissement.Lycee].[DesLyc].Members, ([Measures].[Eval] = "Eleve")) ON ROWS
FROM CubeBacDW
WHERE [Annee].[2015]
```

b. Résultat :

FUZZY MDX QUERY:Taux de Succes BAC ELEVE /Lycee

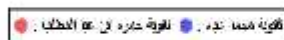


	Mesures
Lycee	Taux de Succes
تأوية حمزة ابن عبد الملك	70,00%
تأوية محمد حمزة	51,17%

Slicer: [Annee=2015]



Slicer: Année=2015



[back to index](#)

Figure 14: FUZZY MDX QUERY : Taux de succès du BAC Elevé/Lycee

7.6.3. FUZZY MDX QUERY : Taux de succès du BAC Moyen/Lycee

a. Requête MDX :

```
SELECT {[Measures].[Taux de Succes]} ON COLUMNS,
FILTER([Etablissement.Lycee].[DesLyc].Members, ([Measures].[Eval] = "Moyen")) ON ROWS
FROM CubeBacDW
WHERE [Annee].[2015]
```

b. Résultat :

FUZZY MDX QUERY:Taux de Succes BAC MOYEN /Lycee



	Mesures
Lycee	Taux de Succes
التربية الجديدة العنبرة	42,64%
ثانوية بني ذر القفاري	54,63%
ثانوية الأمير محمد القادر	42,75%
ثانوية امارة أجدد	60,00%
ثانوية ترويس مهنعلي	43,03%
ثانوية قويدري محمد	40,07%
ثانوية محمد بوراس	58,90%
ثانوية مزاح علي	46,10%

Slicer: [Annee--2015]



Slicer: Année=2015

Figure 15: FUZZY MDX QUERY : Taux de succès du BAC Moyen/Lycee

7.6.4. FUZZY MDX QUERY : Taux de succès du BAC Faible/Lycee

a. Requête MDX :

```
SELECT {[Measures].[Taux de Succes]} ON COLUMNS,
FILTER([Etablissement.Lycee],[DesLyc].Members, ([Measures].[Eval] = "Faible")) ON ROWS
FROM CubeBacDW
WHERE [Annee].[2015]
```

b. Résultat :

FUZZY MDX QUERY: Taux de Succes BAC FAIBLE /Lycee



Lycee	Mesures
	► Taux de Succes
ثانوية الإمامة تروحي	07,95%
ثانوية المعاهد بوزارة بن حيد	04,47%
ثانوية المعاهد محي محذوب عبد القادر	18,10%

Slicer: [Annee-2015]



[back to index](#)

Figure 16: FUZZY MDX QUERY : Taux de succès du BAC Faible/Lycee

7.6.5. RECOMMANDED MDX QUERY Avec Profil : ADMIN

a. Requête MDX :

```
SELECT {(DESWILAYA).[ALL DESWILAYA],[Measures].[Taux de Succes]} ON COLUMNS,
(ETABLISSEMENT).[ALL LYCEE],[SEXE].[ALL SEXE] ON ROWS
FROM CubeBacDw
WHERE ([Annee].[2015])
```

b. Résultat :

RECOMMANDED MDX QUERY:Taux de Succes BAC pour ADMIN



		Wilaya		
		-All Des Wilaya	من النظر	النسبة
		Measures	Measures	Measures
Lycee	Sexe	Taux de Succes	Taux de Succes	Taux de Succes
-All Lycee	-All Sexe	31,87%	31,39%	42,75%
	ذكور	23,96%	22,70%	54,24%
	انك	38,80%	30,05%	33,33%
لتقوية الطليحة العائرة	-All Sexe	43,64%	43,64%	
	ذكور	44,00%	44,00%	
	انك	43,53%	43,53%	
تقوية أي نر العفاري	-All Sexe	54,63%	54,63%	
	ذكور	44,44%	44,44%	
	انك	59,72%	59,72%	
تقوية الأمير عبد القادر	-All Sexe	42,75%		42,75%
	ذكور	51,21%		51,21%
	انك	33,33%		33,33%
تقوية الاخيرة فريحي	-All Sexe	07,95%	07,95%	
	ذكور	05,72%	05,72%	
	انك	13,42%	13,42%	
تقوية المعاهد بوراوين عبد	-All Sexe	04,47%	04,47%	
	ذكور	03,54%	03,54%	
	انك	05,70%	05,70%	
تقوية المعاهد حسي بنظير عبد القادر	-All Sexe	18,16%	18,16%	
	ذكور	12,68%	12,68%	

Figure 17: RECOMMANDED MDX QUERY Avec Profil : ADMIN

7.6.6. RECOMMANDÉ MDX QUERY Avec Profil : Directeur WILAYA 44

a. Requête MDX :

```
SELECT ([Wilaya], [Measures].[Taux de Succes]) ON COLUMNS,
([Etablissement],[SEXE]) ON ROWS
FROM CubeBacDw
WHERE ([Annee].[2015])
```

b. Résultat :

RECOMMENDED MDX QUERY: Taux de Succes BAC/Lycees pour Directeur WILAYA

		Code Wilaya
		44
		Mesures
Lycee	Sexe	Taux de Succes
-All Lycee	-All Sexe	31,39%
	ذكور	22,70%
	انثى	39,05%
التربية الجيدة العفوية	-All Sexe	13,51%
	ذكور	44,00%
	انثى	43,53%
تأهوية من طرف النظاري	-All Sexe	54,03%
	ذكور	44,44%
	انثى	59,72%
تأهوية لإختوة تروحي	-All Sexe	07,95%
	ذكور	05,72%
	انثى	13,42%
تأهوية لمخاطبة بوزارة بن عبد	-All Sexe	04,47%
	ذكور	03,54%
	انثى	05,70%
تأهوية لمخاطبة من مطرب عبد القادر	-All Sexe	18,16%
	ذكور	17,68%
	انثى	24,72%
تأهوية امبارك أحمد	-All Sexe	50,00%
	ذكور	54,41%
	انثى	50,45%

Figure 18: RECOMMENDED MDX QUERY Avec Profil : Directeur Wilaya 44

7.6.7. RECOMMANDED MDX QUERY Avec Profil : Directeur LYCEE 2444

a. Requête MDX :

```
SELECT ([CODLYCEE].[Measures].[Taux de Succes]) ON COLUMNS,
CROSSJOIN ({[ETABLISSEMENT].[DESLYC].MEMBERS}, {[SEXE].[DESSEXE].MEMBERS}) ON
ROWS
FROM CubeBacDw
WHERE ([Annee].[2015])
```

b. Résultat :

RECOMMANDED MDX QUERY: Taux de Succes BAC pour Directeur LYCEE



		Cod Lycee
		2444
		Mesures
Lycee	Sexe	Taux de Succes
تقوية حمزة ابن عبد الحبيب	ذكور	85,13%
	اناث	56,67%

Slicer: [Annee=2015]

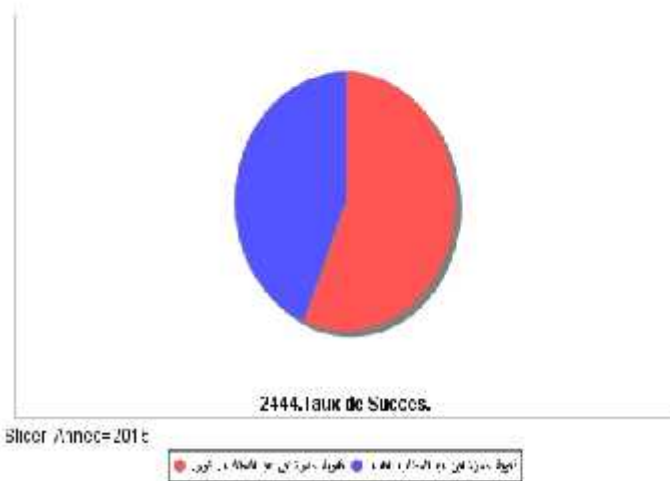


Figure 19: RECOMMANDED MDX QUERY Avec Profil : Directeur Lycee 2444

7.7. Indicateurs de performance du système

L'analyse des performances du système nous conduit à faire un test benchmark entre le langage MDX et le langage SQL en comparant leurs temps d'exécution car les tests pratiques ont prouvé que :

- Le **SGBDR MySql** permet déjà la scalabilité de n'importe qu'elle système ainsi que sa remonté en charge et surtout pour notre cas avec l'augmentation annuelle du nombre des tuples du cube de données BAC D,
- Le serveur web **Apache Tomcat** gère efficacement les accès simultanés multi utilisateurs à distance dans une architecture client-serveur,
- Le **temps de réponse** des différentes requêtes testées dans notre logiciel est strictement inférieur à *1 seconde* pour 10 000 tuples existants dans la base BACDW, ce qui rend le temps d'attente invisible pour l'utilisateur.

Pour analyser la performance et approuver la puissance du langage MDX par rapport à SQL dans notre cas d'étude, nous avons fait le test benchmark entre MDX Query et SQL Query.

7.7.1. Test Benchmark (MDX Query VS SQL Query)

Nous avons testé la requête MDX suivant et sa équivalente en SQL.

✓ **Requête MDX:**

```
SELECT ([Wilaya], [Measures].[Taux de Succes]) ON COLUMNS,  
([Etablissement],[SEXE]) ON ROWS  
FROM CubeBacDw  
WHERE ([Annee].[2015])
```

✓ Requête SQL équivalente:

SELECT DesLyc,DesSexe,

(select COUNT(p.fklyc) from factcandbac p where ((p.fklyc=l.pklyc) and (p.fksexe=s.pksexe))
group by p.fklyc,p.fksexe) as Nbrinscrit,

(select COUNT(p.fklyc) from factcandbac p where ((p.fklyc=l.pklyc and moyenbac>=10) and
(p.fksexe=s.pksexe)) group by p.fklyc,p.fksexe) as NbrReussitBac,

(select COUNT(p.fklyc) from factcandbac p where ((p.fklyc=l.pklyc and moyenbac>=10) and
(p.fksexe=s.pksexe)) group by p.fklyc,p.fksexe)/

(select COUNT(p.fklyc) from factcandbac p where ((p.fklyc=l.pklyc) and (p.fksexe=s.pksexe))
group by p.fklyc,p.fksexe) as TauxReussitBacParSexeLycee

FROM factcandbac p

RIGHT OUTER JOIN dimlycee l on l.pklyc=p.fklyc

RIGHT OUTER JOIN dimsexe s on s.pksexe=p.fksexe

WHERE (fkannee=2015 and l.codwil=44)

GROUP BY p.fklyc,p.fksexe

✓ Résultats du Test Benchmark:

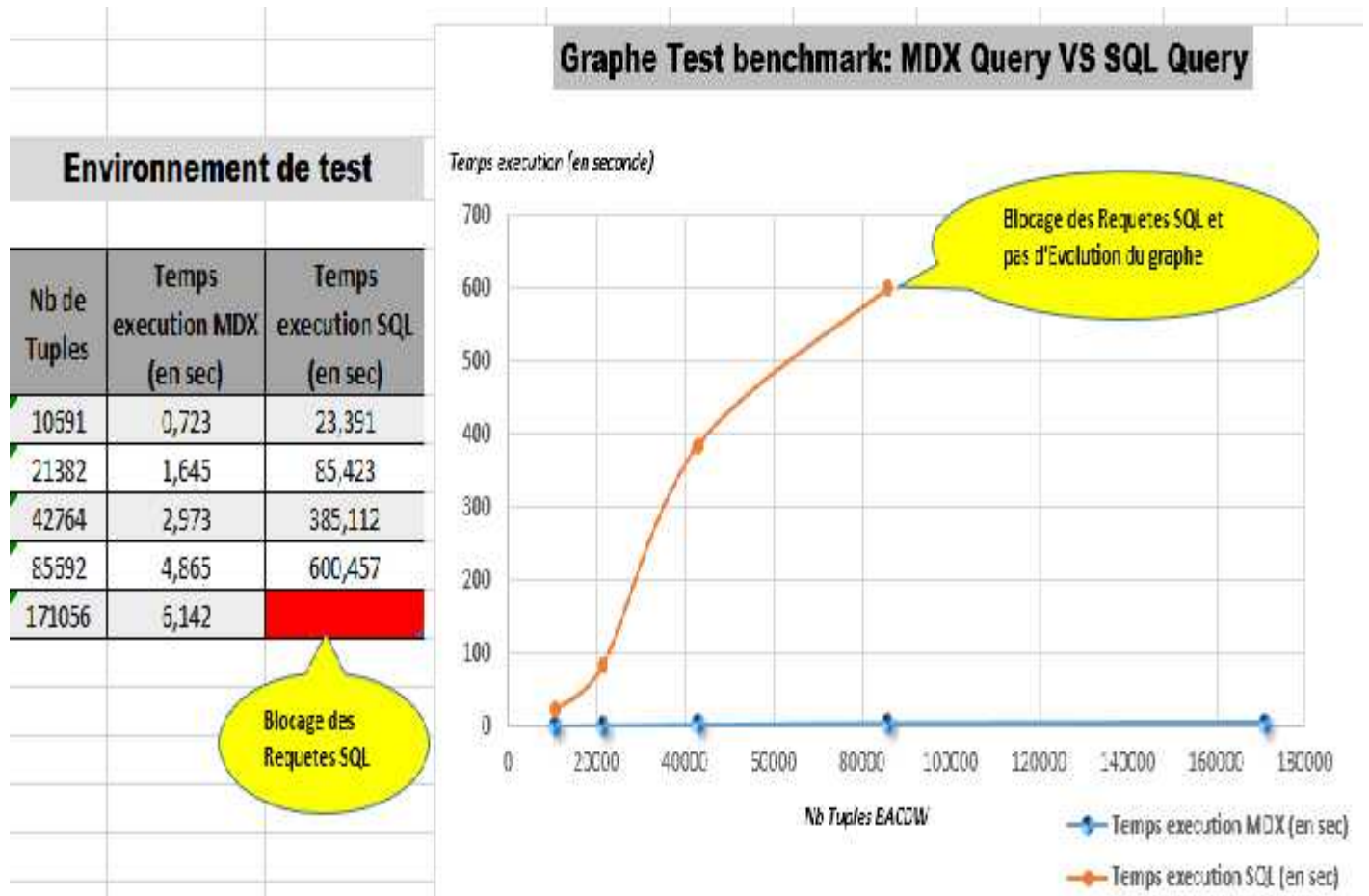


Figure 20 : Test de Benchmarking (MDX Query VS SQL Query)

7.8. Conclusion

Dans ce chapitre nous avons implémenter notre propre solution BI via l'utilisation du langage MDX en interrogeant le cube de données BAC avec la prise en compte du profil utilisateur.

Notre travail se distingue des travaux existants dans le domaine car nous avons utilisé le langage MDX Query dans la recommandation des requêtes, ce qui offre une réelle originalité à notre méthode dans le cas de prise de compte des préférences utilisateurs dans la recommandation des requêtes décisionnelles.

Conclusion & Perspectives

“If you think you are worth what you know, you are very wrong. Your Knowledge today does not have much value beyond a couple of years. Your value is what you can learn and how easily you can adapt to the changes this profession brings so often”.

-Jose M.Aguilar-

Conclusion & Perspectives

Notre travail s'inscrit dans le domaine des technologies BI. Cette technologie permet d'avoir une vision détaillée sur les indicateurs de performance à analyser d'une façon multidimensionnelle dans une entreprise ou dans une administration, comme dans notre cas d'étude.

Nous avons constaté que peu de travaux de recherches se sont intéressés à la personnalisation des analyses OLAP. De même l'absence des modèles génériques qui se basent sur les préférences du décideur.

A la lumière de cela, nous avons proposé une approche globale des analyses OLAP des résultats du BAC basé sur la personnalisation et la recommandation des requêtes. Cette approche se base sur le profil de l'utilisateur et elle prend en considération l'utilisation de quelques termes flous dans le langage MDX pour interroger l'entrepôt de données.

Notre projet permet de répondre aux besoins des décideurs intéressés par l'analyses des résultats du BAC en leur offrant trois types d'analyses :

- ✓ La première est standard en utilisant le ROLAP MDX QUERY,
- ✓ La seconde par l'introduction de la notion de logique floue dans les requêtes MDX, qui est le FUZZY MDX QUERY,
- ✓ Le dernier type, qui est notre objectif : RECOMMENDED MDX QUERY ou nous avons présenté un algorithme de recommandation

qui permet de modifier le schéma du cube dynamiquement suivant les préférences de l'utilisateur (*Personnalisation*) et ainsi lui recommander des vues du cube qui l'intéressent (*Recommandation de filtrage collaboratif*).

En effet, notre solution proposée, relative à la recommandation des requêtes, est une combinaison de la personnalisation et de la recommandation de filtrage collaboratif, d'où la complémentarité des actions de l'une à l'autre.

Néanmoins, notre travail dégage des perspectives qui peuvent être étudiées par d'autres étudiants ultérieurement :

- ✓ Tester le logiciel réalisé avec les résultats du BAC de toutes les wilayas, en définissant les différents profils de recommandation des requêtes ;
- ✓ Développer la partie de FUZZY MDX QUERY en intégrant d'autres prédicats flous et ainsi développer une réelle couche floue au langage MDX ;
- ✓ Intégrer d'autres éléments liés au profil utilisateur dans notre modèle et faire une extension du profil à travers la gestion des historiques de navigation et la table de correspondance pourrait être ignorée ;
- ✓ Gestion de la mémoire cache du système par l'intégration des algorithmes d'optimisation des requêtes MDX.

Références Bibliographiques

Références Bibliographiques

1. Bibliographie

- 1- [NEGRE, 2016]
« Entrepôt de données » d'Elsa NEGRE, support de cours, Univ Paris Dauphine, 2015-2016.
- 2- [NEGRE, 2008]
« Quand la recommandation rencontre la personnalisation » d'Elsa NEGRE, rapport de publication, Univ Tours France
- 3- [Khemiri, 2015]
« Vers l'OLAP Collaboratif pour la recommandation des analyses en ligne personnalisée » de Rym KHEMIRI, Thèse Doctorat, Univ Lyon2 France, 23 Septembre 2015
- 4- [Laurent, 2002]
« Base de données multidimensionnelles floues et leur utilisation pour la fouille de données » de Anne LAURENT, Thèse Doctorat, Univ Paris 6 France, 20 Sept 2002
- 5- [BARBET & GUILLARD, 2007]
« Refonte du prototype d'interrogation floue ISQLF » de BARBET Aude & GUILLARD Florian, Rapport de projet, ENSSAT France, Mars 2007
- 6- [PILGRIM, 2011]
« Projet PostgreSQLF » de l'équipe PILGRIM, Rapport de projet, IRISA France, 2011
- 7- [BARTEL, 2006]
« Interrogation floue de bases de données : extension de iSQLf » de BARTEL Samuel, Rapport de projet, ENSSAT France, 2006
- 8- [Fasel, 2012]
« Concept and Implementation of a Fuzzy Data Warehouse » de Daniel FASEL, Thèse Doctorat, Univ Fribourg Swiss, 30 Mai 2012
- 9- [Belaid & Bensoula, 2015]
« Conception et réalisation d'un système décisionnel pour la gestion des ressources humaines de la société KAHKARIB » de B.Belaid&A.Bensoula, Mémoire Ingénieur, ESI Algérie, 22 Juin 2015
- 10- [Espinasse, 2015/2016]
« Supports de cours et travaux pratiques sur les DW pour cycle Ingénieur » de B.Espinasse, Ecole Poly. Univ Marseille, 2015/2016
- 11- [Aouiche, 2007/2008]
« Supports de cours et travaux pratiques sur Mondrian et MDX pour cycle Ingénieur » de K.Aouiche, Univ Lyon2, 2007/2008

- 12- [**Bellatreche, 2005a**]
« Techniques d'optimisation des requêtes dans les Data Werhouses » de
LadjelBellatreche, rapport de publication, ENSMA France
- 13- [**Bentayeb, 2005**]
« Personnalisation dans les entrepôts de données : bilan et perspectives », de
F.Bentayeb, rapport de publication, Univ Lyon2
- 14- [**Fabre, 2005**]
« Modèle d'entrepôt de données à base de règles », de C. Fabre, rapport de publication,
Univ Lyon2
- 15- [**Boussaid, 2005**]
« Personnalisation de l'information et gestion des profils utilisateurs : une approche
fondée sur les ontologies », de I.Boussaid, mémoire Magister en Informatique, Univ
Constantine Algérie, 18 Avril 2005
- 16- [**Kostadinov, 2012**]
« Data personalization and user profiles management”, de DimitreKostadinov, rapport
de publication, Univ Versailles France
- 17- [**Boussaid, 2012**]
« Profilisation et contextualisation des systèmes OLAP : Aperçu de l'état de l'art », de
Omar Boussaid, rapport de publication, Univ Lyon2 France
- 18- [**Ghozzi, 2012**]
« Matérialisation des vues dans un modèle multidimensionnel contraint », de
FaizaGhozzi, rapport de publication, Univ Toulouse France
- 19- [**Khemiri & Bentayeb & Boussaid, 2009**]
« Recommandation interactive de requêtes décisionnelles », de
R.Khemiri&F.Bentayeb&O.Boussaid, rapport de publication, Univ Lyon2 France
- 20- [**Jerbi, 2012**]
« Personnalisation d'analyses décisionnelles sur les données multidimensionnelles »,
de H. Jerbi, Thèse doctorat, Univ Toulouse France, 20 Janvier 2012
- 21- [**Briksi, 2008**]
« Etude de la relaxation de requêtes dans un contexte flexible », de Amine
BriksiNigassa, mémoire de magister, Univ Tlemcen Algérie,
- 22- [**Zadeh, 1965**]
« Fuzzy Sets », de L. Zadeh, 1965
- 23- [**Béchet, 2009**]
« Etat de l'art sur les systèmes de recommandation », de N.Béchet, rapport de projet,
INRIA France
- 24- [**Negre, 2009**]
« Les systèmes de recommandations », d'Elsa Negre, rapport de présentation, Univ
Paris Dauphine France
- 25- [**Chouder, 2007**]
« Entrepôt distribué de données », de L.Chouder, mémoire de magister, ex INI
Algérie, 23 Janv 2007

- 26- [Fabre & Laurent, 2009]**
 « Vers davantage de flexibilité et d'expressivité dans les hiérarchies contextuelles des entrepôts de données », de C.Fabre&A.Laurent, rapport de projet, Univ Lyon2 France
- 27- [Laurent, 2009]**
 « FUB et FUB MINER : deux systèmes pour la représentation, la manipulation et la fouille de données multidimensionnelles floues », de A.Laurent, rapport de publication, Univ Paris 6 France
- 28- [Tchenar & Rahali, 2014]**
 « Une application médicale pour la recommandation contextuelle des documents », de M.I.Tchenar&Y.Rahali, mémoire de Master, Univ Tlemcen Algérien, 24 Juin 2014
- 29- [Teste, 2009]**
 « Modélisation et manipulation des systèmes OLAP : de l'intégration des documents à l'utilisateur », de O. Teste, thèse doctorat, Univ Toulouse France, 07 Décembre 2009
- 30- [Bosc & Pivert, 2008]**
 « SQLf : A Relational Database Language of Fuzzy Querying », de P.Bosc&O.Pivert, rapport de projet
- 31- [Inmon, 1996]**
 « Building the Data Warehouse », Second Edition, W.B. Inmon, John Wiley & Sons USA, 1996
- 32- [Kimball, 2004]**
 « The Data Warehouse ETL Toolkit », 3ème édition, R.Kimball, Wiley Publishing, 2004
- 33- [Bellatreche, 2005]**
 « A personalization framework for OLAP queries », Intl. Workshop on Data Warehousing and OLAP (DOLAP), Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D. (2005).
- 34- [Bellatreche, 2006]**
 « Personalization of MDX Queries ». Journées Bases de Données Avancées (BDA), Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H. (2006).
- 35- [Garrigós, 2009]**
 « A Conceptual Modeling Approach for OLAP Personalization ». Intl. Conf. on Conceptual Modeling (ER), Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009).
- 36- [Giacometti, 2009]**
 « Recommending Multidimensional Queries ». Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, Giacometti, A, Marcel, P., Negre, E. (2009).
- 37- [Giacometti, 2008]**
 « A Framework for Recommending OLAP Queries ». Intl. Workshop on Data Warehousing and OLAP (DOLAP), Giacometti, A., Marcel, P., Negre, E.(2008)
- 38- [Golfarelli, 2009]**
 « Expressing OLAP Preferences ». Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, Golfarelli, M. et Rizzi, S. (2009).

- 39- [Ravat, 2007] « Querying Multidimensional Databases ». Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, Ravat F., Teste O., Tournier R., Zurfluh G. (2007).
- 40- [Ravat, 2008] « Personalization and OLAP Databases ». Annals of Information Systems, Vol. 3, Numéro spécial "New Trends in Data Warehousing and Data Analysis", Ravat F., Teste O. (2008).
- 41- [Fasel, 2012] « Concept and Implementation of a Fuzzy Data Warehouse » Daniel Fasel 30.05.2012.

2. Webographie

- **[WEB 1]** « Article online - L'informatique décisionnel – BI - Valat Didier»
[Online]. Disponible : <http://perso.univ-lyon1.fr/haytham.elghazel/BI/presentation.html>
[Consulté le : 17-Dec-2015].

- **[WEB 2]** « Article online - Le BI C'est quoi ? - Alexandre Le Clainche-Consultant informatique décisionnelle. »
[Online]. Disponible : <http://www.alexandre-le-clainche.com/business-intelligence/la-bi-cest-quoi/comment-page-1/>
[Consulté le : 19-Dec-2015].

- **[WEB 3]** « Article online - Data Warehousing Concepts ».
[Online]. Disponible : http://docs.oracle.com/cd/B28359_01/server.111/b28313/concept.htm
[Consulté le : 25-Dec-2015].

- **[WEB 4]** « Article online - Introduction à la logique floue ».
[Online]. Disponible : <http://franck-dermoncourt.developpez.com/tutoriels/algo/introduction-logique-floue/>
[Consulté le : 26-Dec-2015].