

MA-004 - 203 - 2

Ministère de l'enseignement supérieur et de la recherche scientifique
Université Saad DAHLAB de Blida



Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Domaine: Mathématique et Informatique

Filière: Informatique

Spécialité: Ingénierie des logiciels

Sujet :

Proposition d'une Structure d'un cube de données OLAP adapté aux environnements mobiles

Réalisé par :

M^{elle} BEHA Fatma Zahra

M^{elle} LABDANI Amina

Encadré par :

M^{me} N. REZOUG

Année Universitaire
2012 – 2013

Résumé :

Les applications d'aide à la décision telles que OLAP sont traditionnellement conçues pour les environnements informatiques de bureau. L'adaptation des applications OLAP aux contextes d'utilisation mobile (ex : Smartphones) pose certains problèmes dus à la nature et aux contraintes de ces environnements.

Des techniques de compression ont été proposées afin de répondre aux contraintes de mobilité en termes de capacité mémoire. Ils s'intéressent à minimiser la taille des données sans perte d'information et permettent aussi une analyse OLAP hors-ligne à partir des dispositifs mobiles.

Ce mémoire vise à apporter une solution, basée sur la compression des cubes de données et une architecture orientée services (SOA), pour l'adaptation des cubes de données OLAP aux dispositifs mobiles.

La compression est basé sur la personnalisation selon les choix de l'utilisateur (les données préférés), en trouvant finalement un cube de données compressé composé de données agrégées et pertinentes.

Le cube compressé peut être stocké à l'intérieur du dispositif mobile afin d'une analyse OLAP même hors-ligne, pour l'aide à la décision en tout lieu et à tout moment.

- Mots clés :

Entrepôt de données, OLAP, Mobile OLAP,MDX, XMLA, personnalisation des requêtes

Remerciement

Avant toute chose, nous tenons à remercier le Dieu le tout puissant de nous avoir donné la volonté, la patience et le courage de poursuivre et d'achever ce travail dans les bonnes conditions.

En second lieu Nous tenons à remercier chaleureusement et affectueusement tous ceux et toutes celles qui de près ou de loin ont contribué à la réalisation de notre projet.

Nos vifs remerciements accompagnés de toute notre gratitude vont tout d'abord à notre promotrice M^m REZOUG pour nous avoir proposé ce sujet, pour les conseils qu'il n'a cessé de nous prodiguer, ses encouragements et surtout pour la confiance qu'il nous a accordée pour la réalisation de ce projet, Nous prions Allah de lui rendre grâce pour avoir fait de notre travail avec elle, un réel honneur et grand plaisir.

Nous tenons également à remercier Mr Khalef BOUSSEBAT pour son aide précieuse, ces conseils avisés, ces idées riches et ces encouragements, Nous prions Allah de lui rendre grâce pour avoir aidé nous, un réel honneur et grand plaisir.

Nous tenons à remercier aussi Mlle Hadjer KERROUR pour son soutien et son aide, puisse Dieu l'aider à atteindre tous leurs buts.

Par ailleurs ; nous rendons un vibrant hommage à l'ensemble du corps professoral du département d'informatique de l'université Saâd DAHLAB de Blida qui ont contribué activement et vaillamment à notre formation.

Aussi à l'ensemble du personnel de la bibliothèque centrale qui se sont montrés très professionnels et serviables et nous ont fournis des conditions de travaux optimales.

A tous ceux et à toutes celles dont les noms n'apparaissent pas sur cette page, qu'ils demeurent convaincus, que nous ne les avons point oubliés et qu'ils soient assurés de notre profonde gratitude.

Merci

Dédicace

À mes parents pour leur amour inestimable, leur croyance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.

À mon grand-père Djelloul, mes grand-mères Malika, zoulikha pour ses douceurs et ses gentilleses.

À mes frères Sidali pour leur tendresse, Mohamed Rayane, ma plus grande source de bonheur, j'espère que la vie lui réserve le meilleur.

À mes tantes et mes oncles pour leurs précieux encouragements.

À la mémoire de mon deuxième père ; chère oncle Tayeb

À mon binôme Amina ainsi que toute sa famille

À mes chères amies Hadjer, Romayla, Melyara, Amina .

À mes cousins et mes cousines surtout à Soumia, Meriem, Hadjer, Imene , Manel, et Takwa

À tous mes camarades de la promotion master GL2011/2012

À tous ceux qui, de loin ou de près, n'ont cessé de m'apporter leur soutien tout au long de mes études,

À tous ce qui veille les nuits pour faire jaillir la lumière de savoir

Je dédie ce mémoire

Fatma Zehra

Dédicace

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents qui sont toute ma vie et tout ce que j'ai de plus cher au monde, en témoignage de ma reconnaissance infinie pour les nombreux sacrifices qu'ils n'ont cessé de déployer pour moi et dont je serais à jamais redevable.

Que dieu les garde et leur procure la santé et le bonheur.

A mes frères et bien aimés Morad ,Rachid, Omar, Adel , Particulièrement à mon très cher frère Rjad pour les conseils et les soutiens qu'il n'a cessé de me prodiguer durant toutes mes années d'études.

A mes très chères sœurs Khadidja , Hassina , Nassima .

A mes chers neveux Abdellah, Abderrahmane, Mohamed et ma chère nièce Ikram .

Puisse Dieu les protéger

A tous les miens dont mes oncles, mes tentes, mes cousins et cousines et a toute ma grande famille

A tous mes amis et tous ceux qui me sont chers

A mon amie et binôme Fatma Zahra ainsi que toute sa famille.

A tous mes camarades de la promotion GL 2011/2012

A tous mes enseignants du primaire à l'université,

A tous ceux qui, de loin ou de près, n'ont cessé de m'apporter leur soutien tout au long de mes études,

A tous ce qui veille les nuits pour faire jaillir la lumière de savoir

Et à toi

Amina

SOMMAIRE

Sommaire

INTRODUCTION GENERALE

1. Contexte du travail	I
2. Problématique	II
3. Objectif	II
4. Organisation du mémoire	III

PARTIE I : ETAT DE L'ART

CHAPITRE I : Les entrepôts de données et les systèmes OLAP

1. Introduction	5
2. Le système décisionnel.....	5
2.1 Définition	5
2.2 Architecture d'un système décisionnel	6
3. Entrepôt de données (Data Warehouse)	7
3.1 Définition	7
3.2 Le magasin de données	8
3.3 Les métadonnées	8
4. Modélisation multidimensionnelle d'un Data Warehouse	8
5. Schémas de modélisation multidimensionnelle	9
5.1 Schéma en Etoile	9
5.2 Schémas en flocon de neige	9
5.3 Schéma en constellation de faits	10
6. OLAP (On-Line Analytical Processing)	11
6.1 Définition	11
6.2 Les 12 règles d'OLAP	11
6.3 OLTP versus OLAP	12
6.4 Concepts OLAP	13
7. Architecture fonctionnelle d'un système OLAP	16
8. Manipulation des données multidimensionnelles	17
8.1 Les opérations classiques	17
8.2 Les opérations de restructuration	18
8.3 Les opérations de granularité	18
9. Les systèmes OLAP	19
9.1 Les systèmes ROLAP	19
9.2 Les systèmes MOLAP	20
9.3 Les systèmes HOLAP	21
10. Conclusion	21

SOMMAIRE

CHAPITRE II : Mobile OLAP

1. Introduction	22
2. Mobilité en informatique	22
3. Les dispositifs mobiles	22
4. Architecture logicielle des plateformes mobiles	23
4.1 Application autonome	23
4.2 Application autonome avec synchronisation	23
4.3 Client lourd	24
4.4 Client léger	25
5. Mobile OLAP	26
6. Les contraintes des systèmes utilisant Mobile OLAP	27
6.1 Conditions de conception d'interface utilisateurs pour OLAP mobile	27
6.2 Les conditions de connexion dans un système Mobile OLAP	28
6.3 Contraintes de performances	28
7. Les travaux existants dans le Mobile OLAP	29
7.1 Implémentation d'OLAP dans un environnement mobile	29
7.2 Techniques de compression des cubes de données	30
8. Conclusion	34

CHAPITRE III : La personnalisation dans les entrepôts de données

1. Introduction	35
2. La notion de personnalisation	35
2.1 Définition	35
2.2 Personnalisation et recommandation	36
3. La notion du profil	37
3.1 Définitions	37
3.2 Le contenu du profil	37
3.3 Modélisation du profil utilisateur	37
3.4 Construction du profil	38
3.5 Exploitation de profil	39
4. Les systèmes d'accès personnalisé à l'information	41
5. La notion de préférences	41
5.1 Niveau des préférences	42
5.2 Formulation de préférences	42
5.3 Approches qualitatives Vs Approches quantitatives	42
5.4 Contextualisation	43
5.5 Exploitation de préférences	43
5.5.1 Expression des préférences	44
5.5.2 Sélection des préférences	44
6. Personnalisation des systèmes OLAP	45
6.1 Personnalisation du schéma OLAP	46
6.2 Personnalisation de l'interrogation des données	47
6.2.1 Personnalisation de requêtes	47
6.2.2 Recommandation de requêtes	47
6.3 Personnalisation de la visualisation des données	49
6.4 Personnalisation de la prise de décision	51
6.5 Intégration du profil et contexte de l'utilisateur dans OLAP mobile	51
7. Conclusion	54

SOMMAIRE

PARTIE II : LA DEMARCHE DE DEVELOPPEMENT

CHAPITRE IV : Architecture pour un système OLAP Mobile

1. Introduction	55
2. Architecture du Système proposé	57
2.1 Présentation	57
2.2 Les Composants de système	59
3. Le fonctionnement de Système	61
4. La Communication dans le Système	63
4.1 Communication Mobile OLAP - Serveur d'application	63
4.1.1 Architecture orientée services(SOA)	63
4.1.2 Avantages d'une SOA sur les plateformes mobiles	64
4.1.3 Présentation web service	65
4.1.4 Exploitation du service web dans notre Système	65
4.2 Communication Serveur d'application - Serveur OLAP	67
4.2.1 Présentation	67
4.2.2 Exploitation de Service XMLA dans notre Système	68
5. Les Pseudos Algorithmes pour la compression des cubes de données OLAP	69
5.1 Formalisation du problème	69
5.2 Pseudo Algorithmes pour le client Mobile	70
5.3 Pseudo Algorithme pour le Serveur d'application	71
6. Conclusion	72

CHAPITRE V : Réalisation

1. Introduction	73
2. Présentation de l'entrepôt utilisé	73
3. Architecture technique du système	76
4. Les environnements de développement	77
4.1 Matériel utilisé	77
4.2 Les langages utilisés	77
4.3 Les outils utilisés	79
4.4 Les API utilisées	81
5. Les outils du Web Services	83
5.1 Apache CXF 2.7	83
5.1.1 JAX-WS	83
5.1.2 Apache Axis 1.6	83
5.1.3 Protocole SOAP	84
6. Application cube OLAP compressé via les web services	85
6.1 Le scénario de L'application	85
7. Conclusion	93
CONCLUSION GENERALE	94
ANNEXE	96
BIBLIOGRAPHIE	98

LISTE DES FIGURES

Liste des figures

<i>Figure 1.1: Architecture d'un data warehouse</i>	6
<i>Figure 1.2: Schéma en étoile de l'entrepôt « ventes »</i>	9
<i>Figure 1.3: Schéma en flocon de l'entrepôt « ventes »</i>	10
<i>Figure 1.4: Schéma en constellation de l'entrepôt « ventes »</i>	10
<i>Figure 1.5 : le cube multidimensionnel vente</i>	15
<i>Figure 1.6 : Architecture fonctionnelle d'un système OLAP</i>	16
<i>Figure 1.7: Exemples de Clients OLAP</i>	17
<i>Figure 1.8 : Architecture ROLAP</i>	20
<i>Figure 1.9: Architecture MOLAP</i>	20
<i>Figure 1.10: Architecture HOLAP</i>	21
<i>Figure 2.1 : Application mobile autonome</i>	23
<i>Figure 2.2 : Architecture avec client mobile lourd</i>	24
<i>Figure 2.3: Architecture avec client mobile léger</i>	25
<i>Figure 2.4: mobile OLAP</i>	26
<i>Figure 2.5: une simple architecture du mobile OLAP</i>	29
<i>Figure 2.6: Architecture FCLOS</i>	31
<i>Figure 2.7: extraction et compression de la vue OLAP bidimensionnelle dans Hand-OLAP</i>	32
<i>Figure 3.1: L'intégration du profil la phase d'exécution de la requête</i>	39
<i>Figure 3.2: L'intégration du profil la phase d'exécution de la requête et la visualisation des résultats</i>	40
<i>Figure 3.3 : L'intégration du profil la phase de présentation des résultats</i>	40
<i>Figure 3.4: Architecture fonctionnelle d'un système d'accès personnalisé à l'information</i>	41
<i>Figure 4.2 : Architecture global du système</i>	58
<i>Figure 4.3: Le Schéma de Fonctionnement du Système</i>	61
<i>Figure 5.1 : Schéma en Etoile du cube Sales</i>	75
<i>Figure 5.2 : Architecture technique du système</i>	76

LISTE DES FIGURES

<i>Figure 5.3 : Schéma API de JAVA pour OLAP</i>	81
<i>Figure 5.4 : Le Pilote XMLA</i>	82
<i>Figure 5.5 : Requêtes et réponses HTTP</i>	84
<i>Figure 5.6 : Entrer le domaine d'analyse</i>	85
<i>Figure 5.7 : Le menu d'application du service Personnalisation</i>	86
<i>Figure 5.8 : La liste des Dimensions</i>	87
<i>Figure 5.9 : La liste des niveaux</i>	87
<i>Figure 5.10 : La liste des mesures</i>	88
<i>Figure 5.11 : Cube de données non compressé</i>	89
<i>Figure 5.12 : La sélection des dimensions</i>	90
<i>Figure 5.13 : La sélection des niveaux (A)</i>	90
<i>Figure 5.14 : La sélection des niveaux (B)</i>	91
<i>Figure 5.15 : La sélection des unités mesurée</i>	91
<i>Figure 5.16 : Le résultat d'algorithme de compression</i>	92
<i>Figure 5.17 : Cube de données compressé après la personnalisation</i>	93

LISTE DES TABLEAUX

Tableau 1.1 : Comparaison des processus OLTP et OLAP	13
Tableau 2.1 : Synthèse des travaux de compression dans mobile OLAP.....	33
Tableau 3.1 : Synthèse des travaux sur la personnalisation de l'interrogation des bases de données multidimensionnelles	53
Tableau 5.1 : Les dimensions et les niveaux de cube Sales	74
Tableau 5.2: les unités mesurées de cube Sales	74

Liste des Abréviations

BI	<i>Business Intelligence</i>
OLAP	<i>On-Line Analysis Processing</i>
SI	<i>Système d'Information</i>
SID	<i>Système d'Information Décisionnel</i>
SGBD	<i>Système de Gestion de Base de Données</i>
SGBDR	<i>Système de Gestion de Base de Données Relational</i>
PC	<i>Personal Computer</i>
OLTP	<i>On-Line Transactional Processing</i>
MOLAP	<i>Multidimensional On-Line Analysis Processing</i>
ROLAP	<i>Relational On-Line Analysis Processing</i>
HOLAP	<i>Hybrid On-Line Analysis Processing</i>
SOLAP	<i>Spatial On-Line Analysis Processing</i>
SIG	<i>Systèmes d'Information Géographiques</i>
MDX	<i>Multidimensional Data Expression</i>
PDA	<i>Personal Digital Assistant</i>
SQL	<i>Structured Query Language.</i>
RI	<i>Recherche d'Information</i>
BDD	<i>Base De Données.</i>
BDM	<i>Base de Données Multidimensionnelle.</i>
XML	<i>eXtensible Markup Language</i>
WSDL	<i>Web Services Description Language</i>
SOAP	<i>Simple Object Access Protocol</i>
UDDI	<i>Universal Description, Discovery and Integration</i>
SOA	<i>Services Oriented Architecture</i>
XMLA	<i>XML for Analysis</i>
API	<i>Application Programming Interface</i>
ETL	<i>Extract, Transform, Load</i>
JDBC	<i>Java Data Base Connectivity</i>

*I*NTRODUCTION

*G*ENERALE

1. Contexte du travail :

Depuis le milieu des années 1990, une catégorie de systèmes d'aide à la décision existe sur le marché : il s'agit des outils OLAP, pour (*On-Line Analytical Processing*), permettant d'effectuer une analyse exploratoire des données de manière rapide, interactive et conviviale. Ces logiciels font partie du domaine appelé Business Intelligence (BI), soit « informatique décisionnelle », et découlent du besoin des entreprises et organisations d'appuyer les processus de prise de décision sur un éventail de données et d'analyses les plus complètes et globales que possible.

D'un autre côté, l'informatique mobile a fait des avancées considérables au cours de la dernière décennie : on peut le constater avec l'ubiquité des appareils tels que les téléphones mobiles, ordinateurs portatifs et de poche (PDA) pouvant se connecter aux réseaux sans fils, qu'il s'agisse de téléphonie cellulaire ou de réseaux WiFi. Ainsi, les multiples services offerts via Internet sont désormais accessibles en tout lieu et en tout temps, au creux de la main. En parallèle, l'évolution du commerce électronique et autres systèmes répartis sur Internet ont donné naissance à un paradigme architectural ainsi qu'une série de protocoles de communication : il s'agit des architectures orientées services et des services Web [1].

Les applications OLAP actuelles sont conçues pour les environnements informatiques de bureau. Afin d'aider à la prise de décision en situation de mobilité, il serait souhaitable d'offrir des versions mobiles à ces applications.

Le mixage des technologies entrepôt de données et dispositifs mobiles est connu par l'entrepôt de données ubiquitaire ou par abus de langage Mobile OLAP [2].

Une des contraintes de Mobile OLAP est l'espace de mémoire de traitement et de stockage réduit dans les dispositifs mobiles, ce qui a mener à chercher des techniques de compression visent à réduire la taille des données provenant des entrepôts de données.

2. Problématique :

Lorsqu'un utilisateur nomade lance une analyse OLAP sur son dispositif, son système devrait pouvoir travailler hors-ligne avec assez de données agrégées pour prendre les bonnes décisions. Et dès qu'il y a une reconnexion, le système devrait utiliser une technique de synchronisation pour savoir qu'elle partie (segment) des données n'a pas encore été sauvegardée sur le dispositif mobile [2].

Les applications OLAP possèdent de gigantesques de données d'analyse. Par conséquent la taille des réponses est souvent volumineuse, et génère une surcharge informationnelle qui dérouté ou décourage l'utilisateur dans son exploration ou sa navigation. Afin de stocker ces réponses dans les dispositifs mobiles qui exigent des contraintes tel que l'espace de mémoire de traitement , et de stockage réduit, cela à mener à chercher des techniques de compression permettant la réduction de la taille des données provenant des entrepôts de données . Ces techniques visent à réduire la taille des cubes de données sans perte d'informations.

La personnalisation des données OLAP peut apporter des réponses satisfaisantes à ce problème car elle permet d'une part à discriminer les analystes en fonction de leurs centres d'intérêt (domaines d'analyses. Par exemple : achat, vente, location, etc..), de leurs préférences, et à leur délivrer des résultats pertinents et d'autre part ;elle répond implicitement à la contrainte capacité mémoire du dispositif mobile vue que la taille des résultats pertinents est réduit, donc on peut les sauvegardés à l'intérieur des dispositifs mobiles afin de les analysés d'une manière autonome par l'utilisateur(décideur,analyste,etc...).

3. Objectif :

L'idée principale de la compression et de son implémentation dans les environnements mobiles est de fournir des cubes OLAP réduits dans leurs tailles afin de les stockés à l'intérieur des dispositifs mobiles dont le but est réaliser une analyse OLAP hors-ligne [2].

L'objectif de notre travail est de réaliser un système pour Mobile OLAP plus particulièrement proposer une structure d'un cube de données OLAP compressé adapté à l'environnement mobile et de :

- Interroger les cubes de données d'une manière personnalisée et cela par la prise en compte des préférences d'un utilisateur dans la reformulation de ses requêtes ensuite les exécuter sur l'entrepôt de données.
- Offrir au client OLAP Mobile un espace d'analyse personnalisé (propre à lui) qui va lui faciliter ses analyses.
- Fournir au client OLAP Mobile un cube compressé implicite, en écartant les données inutiles, et pertinent en fonction de ses caractéristiques spécifiques (préférences exprimés par des poids).

4. Organisation du mémoire :

Pour présenter notre travail et le domaine dans lequel il s'inscrit, nous avons organisé ce mémoire en deux parties.

Partie I : Etat De L'art qui possédera trois chapitres :

- **Chapitre I: *Les entrepôts de données et les systèmes OLAP*** : nous allons présenter l'architecture globale d'un système décisionnel basé sur l'approche « Entrepôt de données » Nous présenterons successivement les concepts de base de l'entrepôt de données, la modélisation multidimensionnelle, les outils d'analyse en ligne, le concept OLAP et la navigation dans les données et les systèmes OLAP.

INTRODUCTION GÉNÉRALE

- **Chapitre II : *OLAP Mobile*** : Ce chapitre est introduit par la relation de l'informatique et la mobilité, les dispositifs mobiles, définit ensuite les contraintes de systèmes utilisant Mobile OLAP, et la partie la plus importante est la présentation des travaux de recherche existants dans Mobile OLAP.
- **Chapitre III: *La personnalisation dans les entrepôts de données*** : dans lequel nous présenterons les notions de la personnalisation, la notion du profil et de préférence, puis nous dresserons un panorama des travaux réalisés sur la personnalisation dans les systèmes OLAP en général.

Partie II : La démarche de développement : possédera deux chapitres :

- **Chapitre IV : *Architecture pour un système OLAP Mobile***: nous allons présenter dans ce chapitre notre architecture du système global, expliquer les différents composants du système ainsi que les différentes interactions entre eux, et montrer les pseudos algorithmes proposés côté client et serveur.
- **Chapitre V : *Réalisation*** : Cette section décrit l'environnement et les outils de réalisation de notre prototype.

Enfin, une conclusion à ce mémoire synthétisera nos principales contributions et donnera quelques perspectives à notre travail.

PARTIE I:

ETAT DE L'ART

CHAPITRE I :

Les entrepôts de données et les systèmes OLAP

1. Introduction :

La Business Intelligence (BI), également intelligence d'affaires ou informatique décisionnelle, La notion de BI englobe les solutions informatiques dont le but est de consolider les informations disponibles au sein des bases de données de l'entreprise. En effet, les entreprises, se trouvent confrontées à des environnements de plus en plus complexes et compétitifs dans lesquels le pilotage implique des choix qui doivent être faits dans des temps très courts tout en prenant en compte un volume d'informations toujours plus important [3].

Alors un système d'aide à la décision est indispensable. Ce système se fait en premier temps par la construction d'un entrepôt de données, qui est une base de données avec des contraintes et structures spécifiques, ensuite l'application de deux techniques OLAP (On-Line Analysis Processing) et l'analyse statistique [4]. On va voir tous ces concepts en détails par la suite.

Dans ce chapitre nous allons parler brièvement sur le système décisionnel dans la première section. La deuxième section est consacrée à l'entrepôt de données et la troisième section nous allons détailler l'outil d'analyse en ligne OLAP.

2. Le système décisionnel :

Il existe deux grandes familles de Systèmes d'Information (SI). On distingue en effet : Les SI opérationnels et Les SI décisionnels.

Les premiers, les SI opérationnels, sont utilisés pour la gestion du quotidien. Souvent, ils sont associés à des progiciels ou des applications développées pour répondre à une problématique métier [3]. Les seconds, les systèmes d'information décisionnels ont été définis comme suit :

2.1 Définition :

Un système d'information décisionnel (SID) est un ensemble de données organisées de façon spécifique, facilement accessible et appropriées à la prise de décision ou encore une représentation intelligente de ces données au travers d'outils spécialisés. La finalité d'un système décisionnel est le pilotage de l'entreprise [3].

2.2 Architecture d'un système décisionnel :

L'architecture type d'un système décisionnel peut être représentée en quatre niveaux. La figure 1.1, nous montre cette architecture [3].

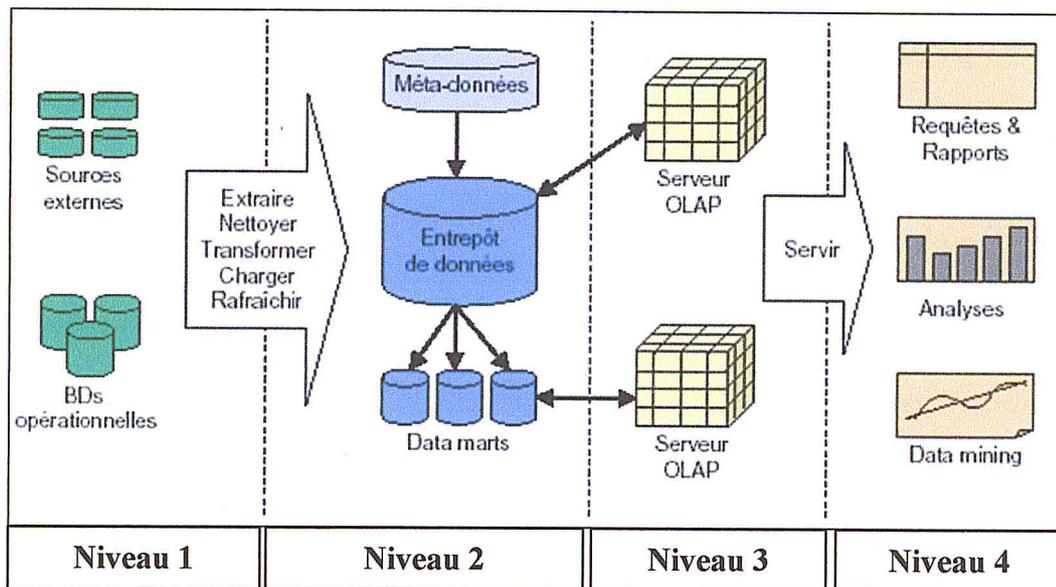


Figure 1.1: architecture d'un data warehouse [3].

- **Le premier niveau :** est celui des sources de données, celui des systèmes source, à savoir le système de gestion de l'entreprise contenant les bases de données opérationnelles ainsi que des sources externes.
- **Le deuxième niveau :** concerne la récupération, la transformation des données, puis l'alimentation d'un entrepôt de données.
- **Le troisième niveau :** est celui de l'exploitation de l'entrepôt en question, celui-ci étant organisé de telle manière que l'on puisse, par fonction de l'entreprise, récupérer l'information mais avec une capacité d'agrégation des données par métier. Les data marts, considérés comme des bases de données métier, sont en fait extraites des entrepôts de données.
- **Le quatrième et dernier niveau :** fournit à l'utilisateur final le moyen de composer sa propre analyse et restitution des données.

3. Entrepôt de données (*Data Warehouse*) :

Le concept d'entrepôt de données a été proposé par W. H. Inmon en 1990 pour répondre à des besoins d'analyse pour les décideurs que les systèmes opérationnels ne pouvaient pas fournir.

3.1 Définition :

Dans son ouvrage de référence "*Building the Data Warehouse*" [5], W.H. Inmon définit l'entrepôt de données comme "*une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse*". Cette définition englobe différents termes que nous explicitons [6] :

- « *Intégrées* » : Les données de l'entrepôt proviennent de différentes sources éventuellement hétérogènes. L'intégration consiste à résoudre les problèmes d'hétérogénéité des systèmes de stockage, des modèles de données, de sémantique de données.
- « *orientées sujet* » : Les données de l'entrepôt peuvent être réorganisées autour de thèmes tels que le patient, les diagnostics, les médicaments...
- « *non volatiles* » : Les données de l'entrepôt sont essentiellement utilisées en mode de consultation. Elles ne sont pas modifiées par les utilisateurs.
- « *historisées* » : La prise en compte de l'évolution des données est essentielle pour la prise de décision qui par exemple utilise des techniques de prédiction en s'appuyant sur les évolutions passées pour prévoir les évolutions futures.
- « *résumées* » : Les informations issues des sources de données doivent être agrégées et réorganisées afin de faciliter le processus de prise de décision.
- « *disponibles pour l'interrogation et l'analyse* » : Les utilisateurs doivent pouvoir consulter les données réorganisées de l'entrepôt en fonction de leur droit d'accès au travers d'outils interactifs d'aide à la manipulation et l'analyse.

3.2 Le magasin de données :

« Le magasin de données ou Data Mart est défini comme un sous ensemble logique d'un entrepôt de données » [7]. Il représente un projet réalisable. On considère souvent le magasin de données comme la réduction de l'entrepôt de données à un seul processus ou à un groupe de processus ciblant un groupe métier spécifique [4].

3.3 Les métadonnées :

Les métadonnées sont toutes les informations de l'environnement de l'entrepôt de données qui ne constituent pas les données. Proprement dites, sont les « données sur les données » [4].

4. Modélisation multidimensionnelle d'un Data Warehouse :

La modélisation multidimensionnelle consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse. Conceptuellement, cette modélisation multidimensionnelle a donné naissance aux concepts de *fait* et de *dimension* [6].

- **Concept de faits** : Le fait modélise le sujet de l'analyse. Un fait est formé de *mesures* correspondant aux informations de l'activité analysée. Les mesures d'un fait sont numériques et généralement valorisées de manière continue [6]. Les mesures sont stockées dans des tables de faits qui contiennent les valeurs des mesures et les clés vers les tables de *dimensions* [8].

- **Concept de dimension** : Une dimension modélise une perspective de l'analyse. Une dimension se compose de paramètres (attributs) correspondant aux informations faisant varier les mesures de l'activité [6].

5. Schémas de modélisation multidimensionnelle :

Trois types de schémas sont fréquemment rencontrés, le schéma en étoile, le schéma en flocon et le schéma en constellation de faits.

5.1 Schéma en Etoile :

Dans un schéma en étoile, une table centrale de faits contenant les faits à analyser, référence les tables de dimensions par des clefs étrangères (voir la figure 1.2). Chaque dimension est décrite par une seule table (feuille de l'arbre de tables) dont les attributs représentent les diverses granularités possibles [9].

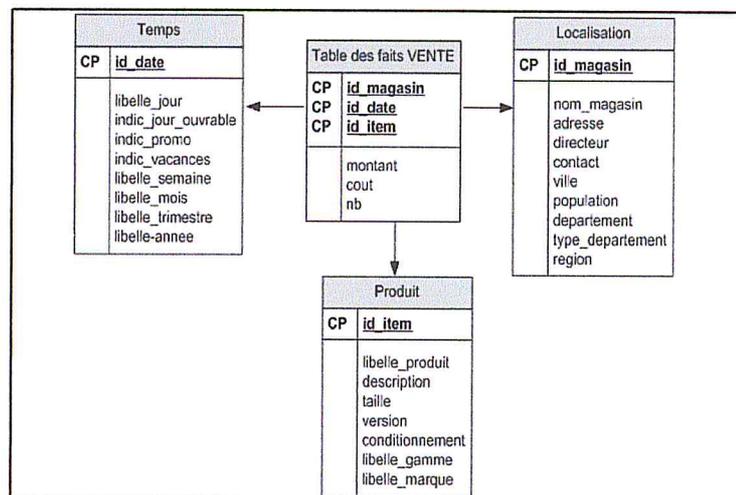


Figure 1.2: Schéma en étoile de l'entrepôt « ventes » [10].

5.2 Schémas en flocon de neige:

Dans un schéma en flocon, cette même table de faits, référence les tables de dimensions de premier niveau, au même titre que le schéma en étoile. La différence réside dans le fait que les dimensions sont décrites par une succession de tables (à l'aide de clefs étrangères) représentant la granularité de l'information (voir la figure 1.3). Ce schéma évite les redondances d'information mais nécessite des jointures lors des agrégats de ces dimensions [9].

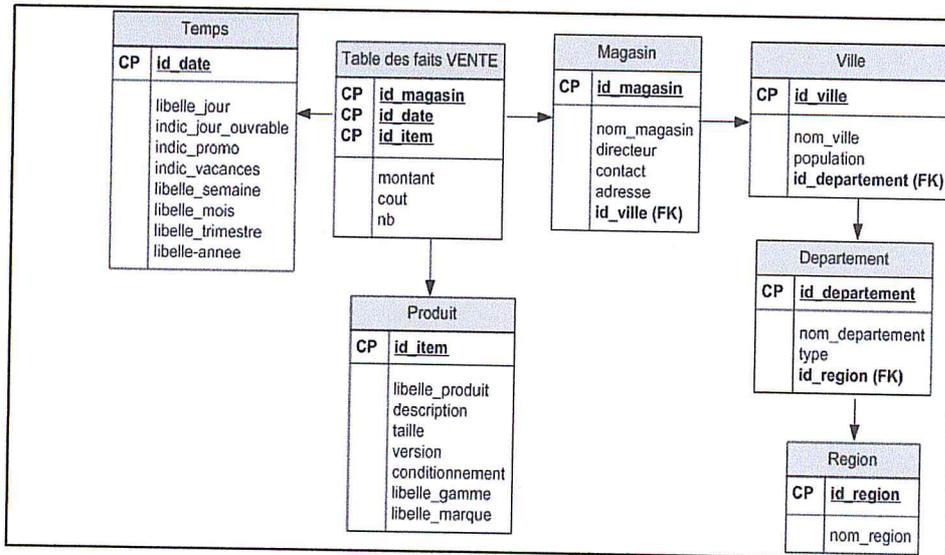


Figure 1.3: Schéma en flocon de l'entrepôt « ventes » [10].

5.3 Schéma en constellation de faits :

Il s'agit de fusionner plusieurs modèles en étoile qui utilisent des dimensions communes (voir la figure 1.4). Un modèle en constellation comprend donc plusieurs faits et de dimensions communes ou non [3].

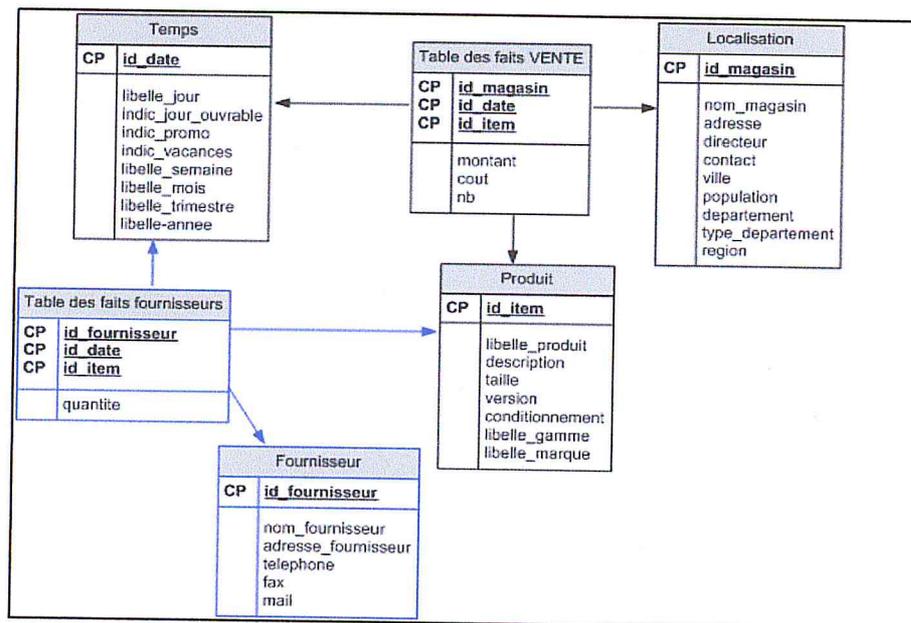


Figure 1.4: Schéma en constellation de l'entrepôt « ventes » [10].

6. OLAP (On-Line Analytical Processing) :

Les bases de données opérationnelles sont caractérisées par une activité constante constituée de modifications et d'interrogations fréquentes des données par de nombreux utilisateurs. Elles assurent le stockage, la consultation et les mises à jour des données détaillées en temps réel. Les données sont gérées par des processus transactionnels en ligne, connus sous le nom d'OLTP (On Line Transactional Processing), afin d'assurer la cohérence de l'information (en évitant la redondance) et optimiser les requêtes sur les données détaillées

Pour ce qui concerne les entrepôts de données, l'analyse des données se fait par l'écriture de requêtes produisant des tableaux de bord synthétiques, la production de tableaux croisés dynamiques et la production de tableaux de données pour des méthodes de data mining (fouille de données). Ces applications utilisent des processus d'analyse en ligne OLAP (On-Line Analytical Processing) [11].

6.1 Définition :

« OLAP est une activité globale de requêtage et de présentation de données textuelles et numériques contenues dans l'entrepôt de données ; style d'interrogation et de présentation spécifiquement dimensionnelle » [12]. La technologie OLAP est non relationnelle et presque toujours basée sur un cube de données multidimensionnel explicite. Les bases de données OLAP sont également connues sous le nom de bases de données multidimensionnelles [4].

6.2 Les 12 règles d'OLAP :

Afin de formaliser le concept OLAP, fin 1993, E. Codd publie un article intitulé « Providing OLAP to User Analysts » aux Etats-Unis, dans lequel il définit douze règles que tout système de pilotage multidimensionnel devrait respecter. Ces règles sont les suivantes [13] :

1. **La vue multidimensionnelle** : le modèle OLAP est multidimensionnel par nature,
2. **La transparence** : l'emplacement physique du serveur OLAP est transparent pour l'utilisateur,
3. **L'accessibilité** : l'utilisateur OLAP dispose de l'accessibilité à toutes les données nécessaires à ses analyses,
4. **La stabilité** : la performance des rapports reste stable indépendamment du nombre de dimensions,
5. **Architecture client-serveur** : le serveur OLAP s'intègre dans une architecture client-serveur.
6. **Le dimensionnement** : le dimensionnement est générique afin de ne pas fausser les analyses,
7. **La gestion complète** : le serveur OLAP assure la gestion des données clairsemées,
8. **Les multi-utilisateurs** : le serveur OLAP offre un support multi-utilisateurs (gestion des mises à jour, intégrité, sécurité),
9. **L'inter-dimension** : le serveur OLAP permet la réalisation d'opérations inter dimensions sans restriction,
10. **L'aspect intuitif** : le serveur OLAP permet une manipulation intuitive des données,
11. **La flexibilité** : la souplesse de l'édition des rapports est intrinsèque au modèle,
12. **L'analyse sans limites** : le nombre de dimensions et de niveaux d'agrégation possibles est suffisant pour autoriser les analyses les plus poussées.

6.3 OLTP versus OLAP :

Les processus transactionnels en ligne OLTP, sont utilisés par les entreprises afin de gérer les informations contenues dans leurs systèmes opérationnels. Les systèmes OLTP ne peuvent répondre aux besoins spécifiques des entreprises pour analyser l'information et supporter efficacement leurs processus d'aide à la décision.

CHAPITRE I : Les entrepôts de données et les systèmes OLAP

Effectivement, contrairement aux applications transactionnelles OLTP, les applications OLAP sont réellement orientées utilisateur, dans leur choix et leur développement. Si les choix technologiques restent un enjeu important, le service rendu et l'efficacité apportée aux utilisateurs dans leur métier quotidien le sont tout autant [3].

Le tableau suivant montre la différence entre les systèmes OLTP et OLAP :

Caractéristiques	Processus OLTP	Processus OLAP
Objectifs	Interrogation, Mise à jour	Analyse, Décision
Nature de données	Individuelles	Multidimensionnelles, agrégées, orientées utilisateur
Fraîcheur de données	Récentes, dynamique	Historiques, statiques
Traitement	Simple	Complexe, semi-automatique
Utilisateurs	Tout type	Décideurs
Temps de réponse	Rapide	Moins rapide

Tableau 1.1 : Comparaison des processus OLTP et OLAP [3].

6.4 Concepts OLAP :

Certains concepts et termes se rapportant aux éléments constituant un cube OLAP seront utilisés tout au long de ce mémoire. Nous définissons ces termes [1]:

- *Cube* : Structure de données destinées à l'analyse, organisées de manière multidimensionnelle et hiérarchique. Un cube est constitué de cellules adressées par des membres disposés sur les dimensions. Un cube n'est pas limité à trois dimensions ; il peut en compter un nombre arbitraire. Les données d'un cube sont généralement puisées à même l'entrepôt de données.

CHAPITRE I : Les entrepôts de données et les systèmes OLAP

- **Dimension** : Correspond aux différents axes d'analyse offerts par le cube. Les dimensions sont des catégories décrivant les thèmes sous lesquels les faits se présentent. Le temps, les catégories de produit, les employés d'une organisation, sont tous des exemples de thèmes pouvant correspondre à une dimension d'un cube
- **Membre** : Les instances d'entités de la réalité dans les dimensions se nomment membres. Il s'agit en autres termes de points sur l'axe d'analyse que constitue une dimension. Par exemple, une dimension « Limites administratives » peut comporter les membres « Canada », « États-Unis » et « France » .
- **Niveau** : Une dimension peut se diviser en plusieurs niveaux, pour modéliser des phénomènes à différentes échelles ou niveaux de détail. Par exemple, une dimension représentant un découpage administratif peut se diviser en niveaux « Pays », « Province », « Région économique » et « Division de recensement » (comté), en allant du plus général au plus détaillé. Les membres appartiennent à un niveau spécifique et peuvent comporter des relations N:1 vers les membres du niveau supérieur .Ces relations sont ce qui permet d'agréger les faits, autrement dit faire le calcul d'une valeur sommaire à partir des valeurs détaillées.
- **Hierarchie** : Certaines dimensions peuvent comporter plus d'un ensemble de niveaux permettant de décrire un thème d'analyse. Par exemple, on peut décrire le temps selon un découpage jour-semaine ou encore jour-mois-année. On parle alors de hiérarchies multiples ou alternatives. Une dimension peut ainsi comporter plus d'une hiérarchie, offrant des chemins différents pour remonter vers les membres de niveau plus général. En règle générale, le niveau le plus détaillé est commun à toutes les hiérarchies à l'intérieur d'une dimension, puisque ce sont les membres de ce niveau qui définissent le grain de la dimension et par conséquent l'adressage des faits détaillés.
- **Fait** : Il s'agit d'une unité d'information contenant les valeurs des mesures, correspondant au croisement d'un membre de chaque dimension. Les liens vers les membres impliqués font également partie du fait .
- **Mesure** : Une mesure se définit comme étant une variable dans le cube OLAP représentant une réalité observée (mesurée), pour laquelle chaque fait comporte une

CHAPITRE I : Les entrepôts de données et les systèmes OLAP

valeur. Nous tenons ainsi à distinguer le concept de mesure, qui est une variable, de son instance dans un fait, soit les valeurs données à une mesure pour un fait. Les types de données des mesures sont dans la plupart des cas des nombres scalaires (e.g. unités vendues, montant des ventes, population, température) mais il peut également s'agir d'objets plus complexes (chaînes de caractères, vecteurs, positions géographiques).

- **Cellule** : Une cellule d'un cube est définie comme le contenant de la valeur d'une mesure pour un fait donné. En contraste avec le terme mesure, une cellule fait référence à un emplacement précis dans le cube ; on peut ainsi dire qu'il s'agit d'une mesure positionnée dans l'hypercube, au croisement des membres d'un fait.

La figure suivante (figure 1.5) montre un exemple d'un cube de données multidimensionnel et ses différents composants :

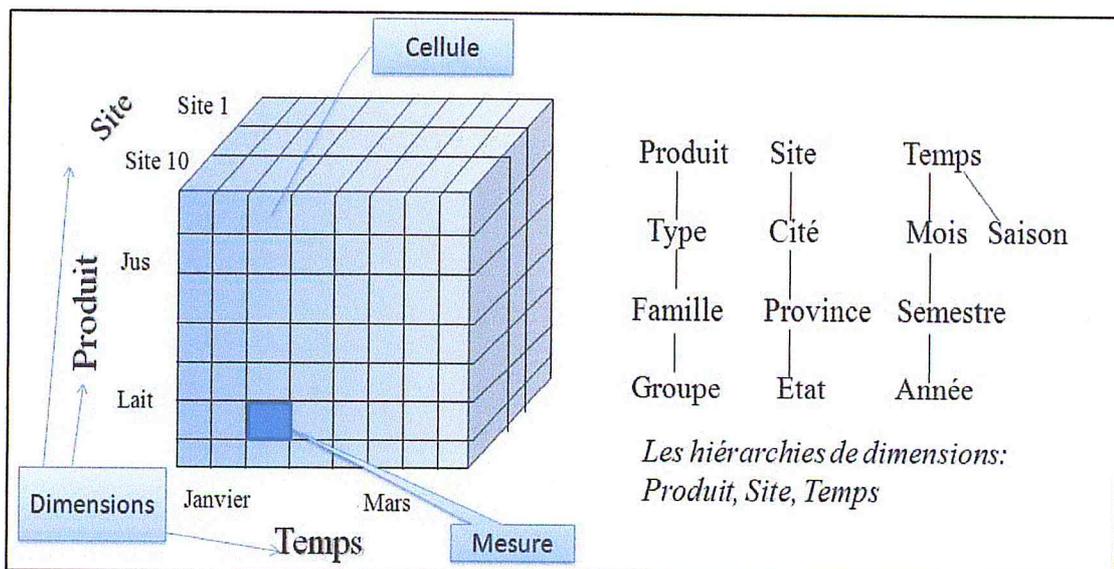


Figure 1.5 : le cube multidimensionnel vente [8].

7. Architecture fonctionnelle d'un système OLAP

Les systèmes décisionnels s'appuient sur une architecture trois tiers telle qu'illustrée par la (figure 1.6) [10] :

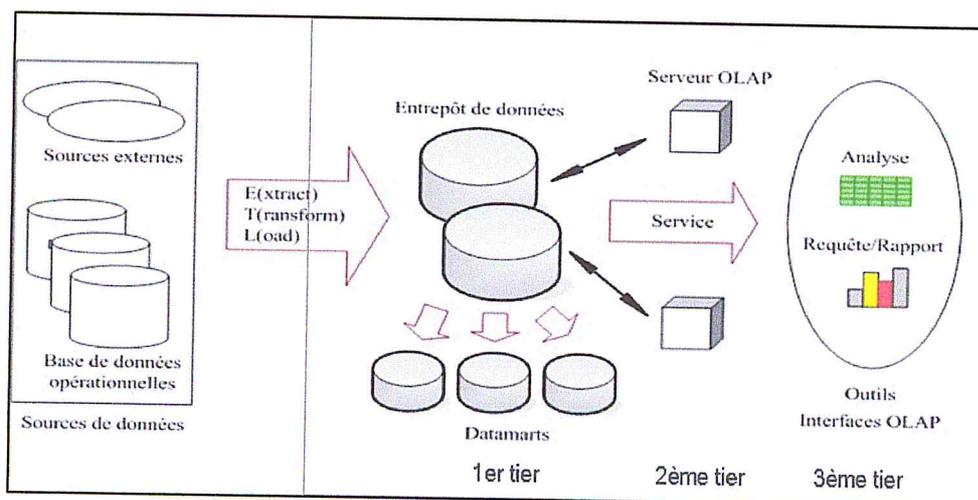


Figure 1.6 : Architecture fonctionnelle d'un système OLAP [10]

Le premier tiers est un SGBD. Les données pertinentes pour l'analyse sont extraites des bases de données transactionnelles, nettoyées et transformées avec les outils ETL et intégrées dans un entrepôt de données maintenu par le SGBD.

Le SGBD contient aussi un ensemble de métadonnées concernant les sources de données, les mécanismes d'accès, les procédures de nettoyage et d'alimentation, les utilisateurs, etc. Le deuxième niveau de l'architecture est un serveur OLAP, tels que par exemple Mondrian, DB2 Olap Server, Oracle OLAP ou Microsoft Analysis Services. Le serveur OLAP génère et maintient l'hypercube. Il fournit une vue multidimensionnelle des données et implémente l'ensemble des opérateurs OLAP (Roll-Up, Drill-Down, etc.) permettant de les analyser. Le dernier niveau est un client OLAP, par exemple Microstrategy, Cognos, JPivot etc. Le client offre une interface utilisateur composée d'outils de reporting, d'analyse interactive, et parfois de fouille de données. Son rôle est de rendre l'information multidimensionnelle « visible » (figure 1.7), en d'autres termes, de permettre de découvrir des connaissances grâce à la seule visualisation et interaction avec les données.



Figure 1.7: Exemples de Clients OLAP [10]

8. Manipulation des données multidimensionnelles :

Les utilisateurs ont besoin d'outils de synthèse et de manipulation de données qui soient efficaces et conviviaux pour la prise de décision. La catégorisation des opérations OLAP en trois types : classiques, de granularité et de restructuration. Les sections suivantes sont consacrées à la description des opérations de manipulation dans le cadre de l'analyse en ligne des données multidimensionnelles [3].

8.1 Les opérations classiques:

Dans cette section nous présentons les opérations dites classiques, celles qui étendent les opérateurs de l'algèbre relationnelle. On peut retenir comme opération classique : la sélection, la projection, le produit cartésien, la jointure, les opérations ensemblistes (union, intersection et différence), la suppression d'une mesure, l'ajout d'une mesure calculée et le renommage [3].

- **Slice** : Consiste à couper une tranche du cube en faisant une sélection sur les cellules par des prédicats selon une dimension.
- **Dice** : Consiste à extraire un sous-cube en faisant une sélection sur les dimensions. Cela revient à faire une restriction sur les dimensions et non plus par rapport à un critère sur la mesure.

8.2 Les opérations de restructuration :

Les opérations agissant sur la structure multidimensionnelle visent à changer le point de vue des données observées [6].

- La **rotation** (*rotate*) consiste à effectuer une rotation de l'hypercube, de manière à présenter une face différente.
- La **permutation** (*switch*) consiste à inverser deux dimensions, de manière à permuter deux tranches de l'hypercube.
- La **division** (*split*) consiste à représenter chaque tranche de l'hypercube en passant à une représentation tabulaire. Plus généralement, cette opération permet de réduire le nombre de dimensions.
- L'**emboîtement** (*nest*) permet d'imbriquer les positions (valeurs) d'un paramètre d'une dimension avec un autre paramètre. L'intérêt de cette opération est qu'elle permet de représenter de manière bidimensionnelle toutes les informations de l'hypercube quelque soit le nombre de dimensions.
- L'**enfouissement** (*push*) consiste à combiner les positions (valeurs) d'un paramètre d'une dimension aux mesures du fait et donc de transformer un paramètre en mesure.

8.3 Les opérations de granularité :

Les opérations agissant sur la granularité d'observation des données caractérisent la hiérarchie de navigation entre les différents niveaux. Elles correspondent aux opérations suivantes [6] :

- Le **forage vers le haut** (*drill-up* ou *roll-up* ou *scale-up*) consiste à représenter les données de l'hypercube à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension. Une fonction d'agrégation (somme, moyenne...) en paramètre de l'opération indique comment sont calculées les valeurs du niveau supérieur à partir de celles du niveau inférieur.
- Le **forage vers le bas** (*drill-down* ou *roll-down* ou *scale-down*) consiste à représenter les données de l'hypercube à un niveau de granularité inférieur, donc sous une forme plus détaillée.

Le forage vers le haut ou vers le bas nécessite des informations non contenues dans l'hypercube pour passer d'une représentation initiale à une représentation de granularité différente. L'augmentation nécessite de connaître la fonction d'agrégation utilisée tandis que l'affinement nécessite de connaître les données au niveau inférieur.

9. Les systèmes OLAP :

En fonction de la manière dont les données sont stockées, on distingue deux approches fondamentales pour construire un système basé sur le modèle multidimensionnel :

L'approche ROLAP (Relational OLAP), où l'on utilise un système de gestion de bases de données relationnel pour stocker les données, et l'approche MOLAP (Multidimensional OLAP), où les données sont physiquement matérialisées de manière multidimensionnelle. Il existe une autre approche nommée HOLAP (Hybrid OLAP), où l'on couple les deux premières approches.

9.1 Les systèmes ROLAP :

Les systèmes de type ROLAP (Relational On Line Analytical Processing) utilisent un SGBD relationnel pour stocker les données de l'entrepôt. Ils représentent une interface multidimensionnelle pour le SGBD relationnel (figure 1.8). Le moteur OLAP est un élément supplémentaire qui fournit une vision multidimensionnelle de l'entrepôt, des calculs de données dérivées et des agrégations à différents niveaux. Il est aussi responsable de la génération des requêtes SQL mieux adaptées au schéma relationnel, qui profitent des vues matérialisées existantes pour exécuter efficacement ces requêtes. Les mesures (par exemple les quantités vendues) sont stockées dans une table qu'on appelle la table des faits. Pour chaque dimension du modèle multidimensionnel, il existe une table qu'on appelle la table de dimension (comme Produit, Temps, Client) avec tous les niveaux d'agrégation et les propriétés de chaque niveau. Ces systèmes peuvent stocker de grands volumes de données, mais ils peuvent présenter un temps de réponse élevé [14].

Les principaux avantages de ces systèmes sont :

- une facilité d'intégration dans les SGBDs relationnels existants.
- une bonne efficacité pour stocker les données multidimensionnelles.

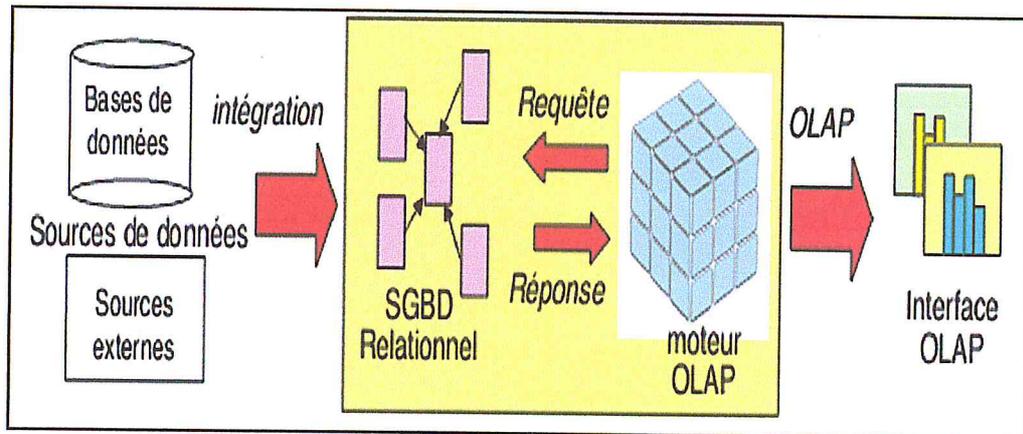


Figure 1.8 : Architecture ROLAP [15]

9.2 Les systèmes MOLAP :

Les systèmes de type MOLAP (stockent les données dans un SGBD multidimensionnel sous la forme d'un tableau multidimensionnel (multidimensionnel array) (voir figure 1.9). Chaque dimension de ce tableau est associée à une dimension du cube. Seules les valeurs de données correspondant aux données de chaque cellule sont stockées. Ces systèmes demandent un pré-calcul de toutes les agrégations possibles. En conséquence, ils sont plus performants que les systèmes traditionnels, mais difficiles à mettre à jour et à gérer.

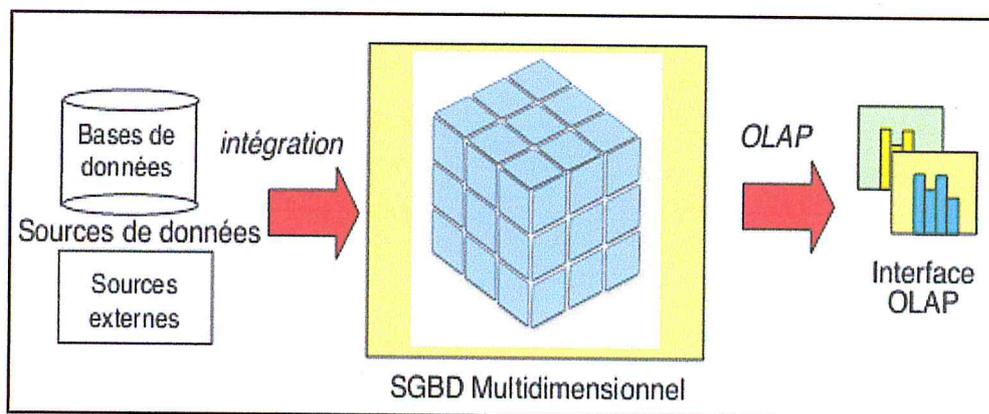


Figure 1.9: Architecture MOLAP [15]

CHAPITRE I : Les entrepôts de données et les systèmes OLAP

Les systèmes MOLAP apparaissent comme une solution acceptable pour le stockage et l'analyse d'un entrepôt lorsque la quantité estimée des données d'un entrepôt ne dépasse pas quelques gigaoctets et lorsque le modèle multidimensionnel évolue peu. Mais lorsque les données sont éparées, ces systèmes sont consommateurs d'espace et des techniques de compression doivent être utilisées [14].

9.3 Les systèmes HOLAP :

Un système HOLAP est un système qui supporte et intègre un stockage des données multidimensionnelles et relationnelles d'une manière équivalente pour profiter des caractéristiques de correspondance et des techniques d'optimisation. La figure suivante (figure 1.10) montre une architecture des systèmes HOLAP [4].

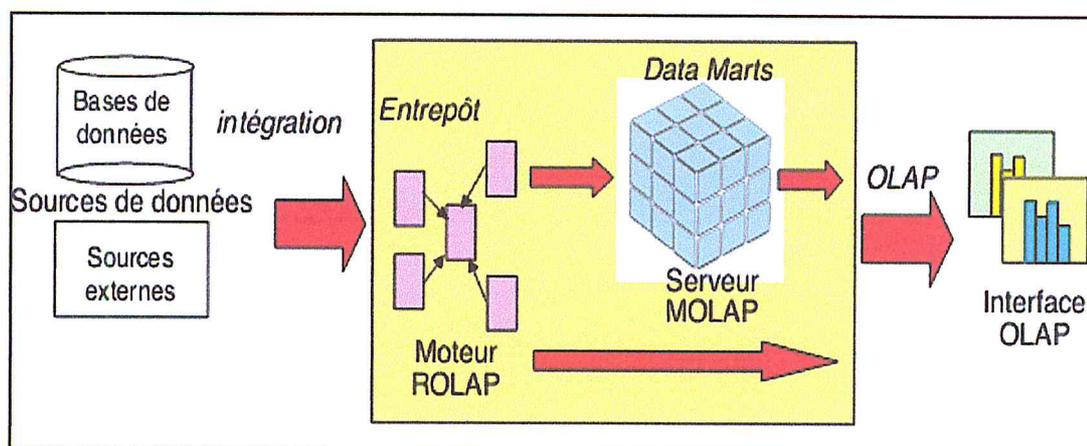


Figure 1.10: Architecture HOLAP [15]

10. Conclusion :

Dans ce chapitre, nous avons évoqué les principaux concepts liés aux entrepôts de données, que ce soit au niveau de la modélisation, de la stratégie de conception, de l'exploitation, etc. Nous avons vu, également les opérations de manipulation de données, les différents systèmes OLAP.

Cette étude a une relation directe à la prochaine étape qui focalise sur le mobile OLAP, ce dernier est une nouvelle technologie qui rassemble deux technologies, les entrepôts de données et le concept mobile.

CHAPITRE II :

OLAP Mobile

1. Introduction :

Le traitement analytique en ligne (OLAP) est une tendance de technologie de base de données, basée sur la vue des données multidimensionnelle. Beaucoup d'applications et des environnements de développement existent offrant l'analyse OLAP aux réalisateurs, aux analystes et aux utilisateurs. Par conséquent l'utilisation des dispositifs mobiles est devenue indispensable dans tous les aspects de la vie réelle [16].

Dans ce chapitre nous allons parler premièrement sur la relation de l'informatique et la mobilité et les différents types et architectures logiciels de dispositifs mobiles. Ensuite nous allons focaliser sur le mobile OLAP, les contraintes de mobilités, les différentes travaux existants et terminer par une conclusion.

2. Mobilité en informatique :

Avec l'avènement des réseaux de télécommunication sans-fil, qu'il s'agisse de téléphonie cellulaire numérique (GSM, GPRS,) ou de réseaux informatiques Wifi, l'informatique est devenue nomade et ubiquitaire. La miniaturisation des composantes électroniques a également permis la production d'appareil informatique mobile de plus en plus petits et puissants : on n'a qu'à penser aux téléphones mobiles évolués (Smartphones, tels que les Blackberry, Apple iPhone, etc.), aux appareils de divertissement audio-visuels portatifs et aux assistants personnels numériques (PDA). Ces développements technologiques font en sorte que l'information est accessible en tout lieu (selon la couverture des réseaux sans-fil) et en tout temps [1].

3. Les dispositifs mobiles :

Les dispositifs mobiles sont un environnement faisant une référence au concept de mobilité, c'est à dire tout un environnement ayant un moyen de communication entre eux via une technologie sans fil, On trouve plusieurs types de dispositifs mobiles, les assistants numériques personnels (PDA), les téléphones cellulaires, les ordinateurs portables, les tablettes et les Smartphones [17].

4. Architecture logicielle des plateformes mobiles :

Les contraintes des environnements informatiques mobiles ont amené le développement d'architectures adaptées à la conception de systèmes logiciels pour ces environnements. De manière simplifiée, on peut énoncer quatre grandes catégories d'architectures [1]:

4.1 Application autonome :

Il s'agit d'une simple application mobile qui ne dépend pas de ressources distantes (Voir la figure 2.1). L'application est installée sur l'appareil mobile, et le stockage de toutes données nécessaires pour son fonctionnement est local. Exemples: calculatrice, bloc-notes, jeux individuels, logiciel de navigation par GPS (avec cartes pré-chargées sur l'appareil).

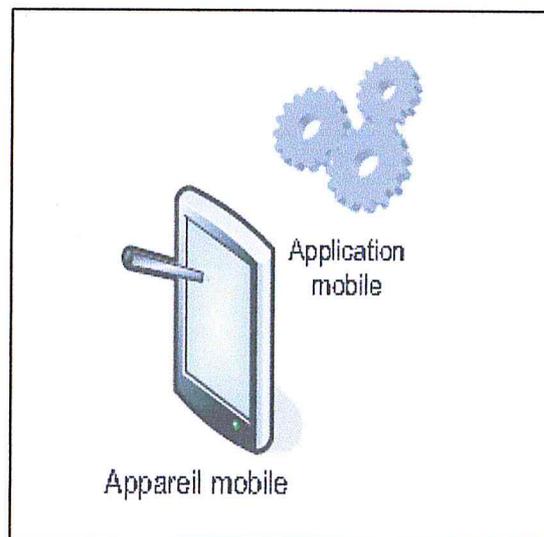


Figure 2.1 : Application mobile autonome [1].

4.2 Application autonome avec synchronisation :

Pour ce type d'application, le fonctionnement est similaire à une application autonome normale (logiciel installé sur l'appareil mobile), mais une synchronisation périodique permet l'échange de données avec un autre système, dit « station de base ». La synchronisation peut s'effectuer en connectant directement l'appareil à un ordinateur de bureau (e.g. par câble USB) ou encore par un lien sans-fil

(ex :Bluetooth). Un logiciel installé sur la station de base prend en charge l'échange des données avec l'appareil mobile. Les logiciels et protocoles de synchronisation sont généralement propriétaires et spécifiques à chaque fabricant d'appareil mobile, et sont limités quant aux applications avec lesquelles ils supportent l'échange de données. Exemple : agenda personnel (synchronisation des tâches, des contacts et des rendez-vous avec un logiciel de bureau) .

4.3 Client lourd :

Il s'agit d'une application installée sur l'appareil mobile, et destinée à interagir avec des ressources à distance, par réseau sans-fil (Voir la figure 2.2). Cette architecture, que l'on retrouve également dans les environnements informatiques de bureau, est aussi souvent nommée client-serveur : le client se connecte via le réseau à un serveur, qui traite les requêtes et retourne les données au client. Ce type d'architecture est généralement dépendant d'une connectivité réseau persistante (pour toute la durée d'utilisation de l'application) .

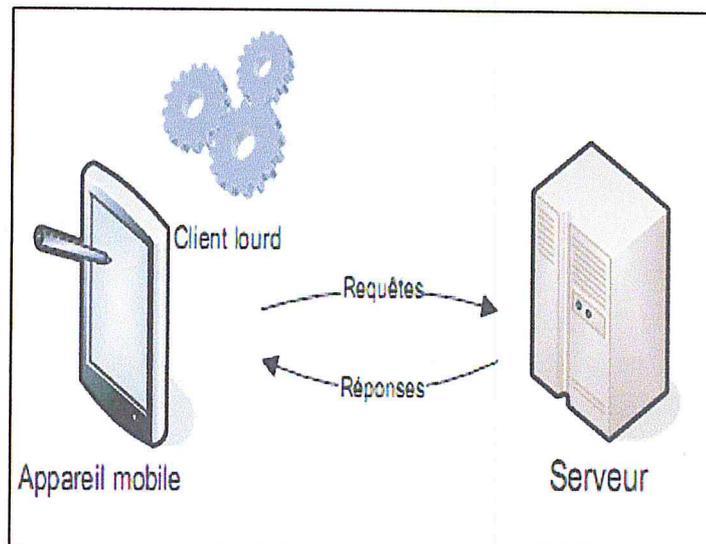


Figure 2.2 : Architecture avec client mobile lourd [1].

4.4 Client léger :

Contrairement aux clients lourds, l'architecture avec client léger (voir la figure 2.3) n'impose pas l'installation d'une application sur l'appareil mobile. L'application proprement dite s'exécute sur un serveur, et l'affichage et l'interaction avec l'utilisateur se font dans un navigateur Web. Le déploiement et la mise à jour de l'application en est donc simplifiée, puisqu'il n'est pas nécessaire de la réinstaller sur chaque appareil mobile pour ce faire. De plus en plus, les applications avec client léger s'exécutant dans un navigateur gagnent en interactivité et en ergonomie, grâce à la technologie dite Ajax, permettant d'éviter le rechargement complet de la page Web lors de chaque interaction de l'utilisateur. Ce type d'architecture a l'avantage d'être indépendante du type d'appareil ciblé (puisque'elle ne requiert qu'un navigateur Web pour fonctionner), et de ne pas nécessiter l'installation d'un logiciel sur l'appareil.

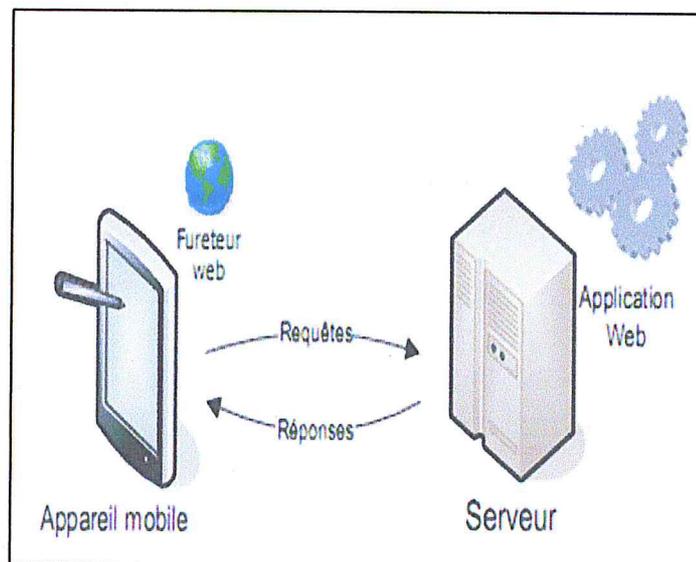


Figure 2.3: Architecture avec client mobile léger [1].

5. Mobile OLAP:

Avec la croissance récente gigantesque du marché mobile et de la disponibilité de nombreux dispositifs mobiles à d'énormes nombres d'utilisateurs finaux, le besoin pour les applications fonctionnant avec ces dispositifs est devenu de plus en plus pressant. Les applications OLAP ont été aussi affectées en cette tendance, d'autant plus que les Systèmes d'aide à la décision sont essentiels à des dirigeants (directeurs, administrateurs, etc.) et aux exécutifs (gouvernements, administrations, directions, etc...), qui généralement utilisent les PDAs ou les téléphones mobiles avec des techniques ayant des capacités avancées [16]. Le mixage des technologies entrepôt de données et dispositifs mobiles est connu par l'entrepôt de données ubiquitaire ou par abus de langage Mobile OLAP [2] (voir la figure 2.4).

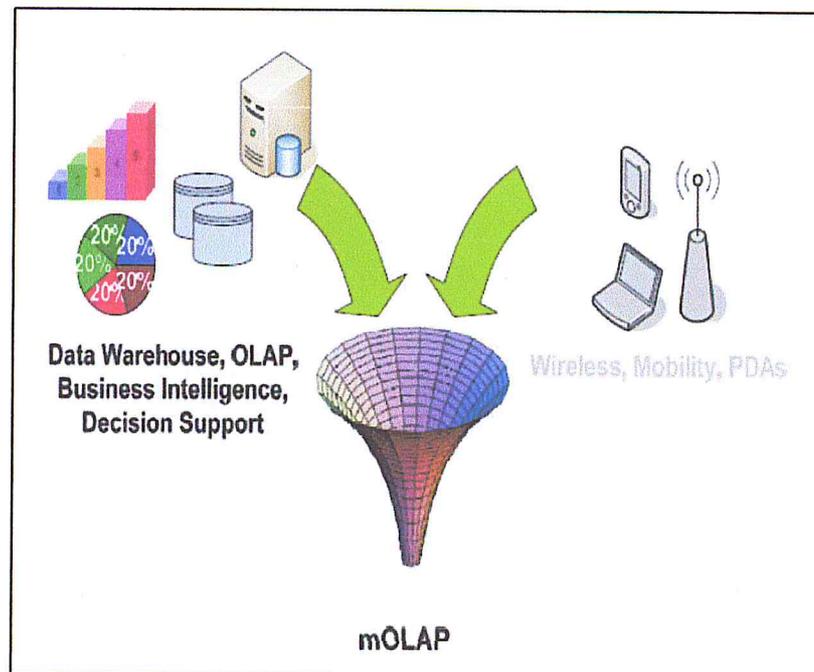


Figure 2.4: mobile OLAP [18].

6. Les contraintes des systèmes utilisant Mobile OLAP :

Dans cette section nous présenterons un certain nombre de conditions pour promouvoir les systèmes OLAP dans les environnements mobiles, en se basant sur tous les aspects impliqués dans le problème (mobilité). De ces conditions générales les plus importantes sont soigneusement choisies pour former la base pour une architecture du système OLAP adaptée à l'environnement mobile [16].

6.1 Conditions de conception d'interface utilisateurs pour OLAP mobile :

Les interfaces utilisateurs et la conception des présentations visuelles pour les dispositifs mobiles sont sujettes à un certain nombre des contraintes en comparaison par rapport à ceux des ordinateurs de bureau. Celles-ci incluent :

a - Petite taille d'affichage: La petite taille d'affichage des dispositifs mobiles est une contrainte centrale à tenir en compte, d'autant plus que les écrans de présentation d'OLAP incorporent habituellement une masse de données à présenter en utilisant un graphique habituellement tabulaire ou autre.

b. - Résolution limitée d'écran: Les résolutions typiques des dispositifs mobiles s'étendent de 100*80 Pixel pour les téléphones portables et 240*320 Pixel pour les PDAs imposant de ce fait le soin spécifique à prendre par les concepteurs d'interfaces pour étendre l'utilisation des systèmes OLAP dans les environnements mobiles.

c. Nombre limité de couleurs disponibles : Le nombre de couleur est limité à une centaine, alors si une analyse OLAP est faite à base de couleur ça serait un handicap. Ainsi le codage d'informations, la différenciation et la séparation des résultats à base de couleur sont d'autres conditions à prendre en charge par le concepteur du design de l'interface.

Encore plus les dispositifs mobiles ne possèdent pas de clavier typique, ni d'indicateur de dispositif 2D, et s'ils existent leur rentabilité est nettement inférieur aux ordinateurs de bureaux. Et enfin, généralement les dispositifs mobiles ne disposent ni de clavier ni de souris [2].

6.2 Les conditions de connexion dans un système Mobile OLAP :

Les environnements mobiles souffrent de problème de connexion et de déconnexion et de faible bande passante. En effet, lorsqu' un utilisateur nomade lance une analyse OLAP sur son dispositif, son système devrait pouvoir travailler hors ligne avec assez de données agrégées pour prendre les bonnes décisions. Et dès qu'il y a une reconnexion, le système devrait utiliser une technique de synchronisation pour savoir quelle partie (segment) des données n'a pas encore été sauvegardée sur le dispositif mobile [2].

6.3 Contraintes de performances :

Pour OLAP mobile, un ensemble de conditions hardware de base sont requises [16] :

a. - Capacité de traitement : les dispositifs mobiles sont en retard en comparaison avec les systèmes de bureau sur lesquels OLAP a eu lieu. Néanmoins, les outils d'entrée OLAP fonctionnant sur les dispositifs mobiles exigent une certaine capacité de traitement minimum pour exécuter des requêtes simples ou toute autre analyse ad hoc sur les données agrégées et stockées localement, avec des techniques de présentation et de visualisation convenablement conçues et mises en application pour les dispositifs mobiles.

b. Stockage locale : La mémoire est un autre problème importante quand des volumes de données énormes doivent être traités afin d'une analyse OLAP. Vu que l'entrepôt de données nécessite une large capacité mémoire. En raison de limitation de dispositifs mobiles; étant donné qu'un utilisateur exécutant l'analyse OLAP n'a pas besoin vraiment d'accéder aux ensembles de données détaillés et des tables de fait du l'entrepôt de données, les données fortement agrégées ou résumées peuvent être stockées dans les dispositifs mobiles pour un balayage hors ligne ; afin d'une lecture plus rapide.

Pour toutes ces contraintes, plusieurs recherches ont été entreprises, ces derniers sont devenus des challenges pour une conception d'un entrepôt de données mobile pouvant être analysé localement sur un dispositif mobile. Dans la section suivante nous dressons un bilan sur un ensemble de recherches qui visent à mixer les deux technologies les dispositifs mobiles et les entrepôts de données.

7. Les travaux existants dans le Mobile OLAP :

Il existe peu de travaux dans ce contexte, la plupart focalisent la compression des cubes de données sans perte d'information, et implémentation d'OLAP dans les environnements mobiles. On citera dans ce qui suit deux catégories des recherches menées dans ce domaine :

7.1 Implémentation d'OLAP dans un environnement mobile :

Nous citerons quelques travaux dans ce contexte comme se suit :

- **Travaux d'A.S.Maniatis :**

A.S. Maniatis [16] a introduit le terme de mobile OLAP, a décrit les fonctions et les particularités de chaque technologie (environnement mobile, système OLAP) ; il a exposé un système prototype utilisant le système OLAP et un type de dispositif mobile. Il est constitué de trois modules (serveur OLAP traditionnel, serveur d'application middleware, et les applications mobiles finales). (Voir la figure 2.5) ; Il a proposé aussi un modèle de représentation de cube de données CPM constitué d'une couche logique permettant la formulation de cube et d'une couche de représentation de cube sur un écran 2D. Ce modèle a permis de séparer entre l'extraction des cubes et leur représentation.

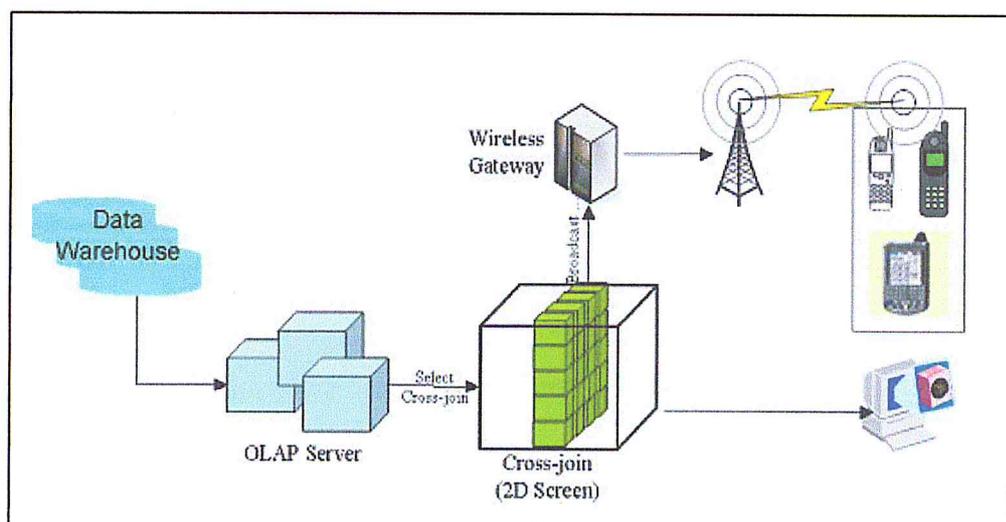


Figure 2.5:une simple architecture du mobile OLAP [16].

Les dispositifs mobiles utilisent les données provenant du serveur OLAP qui sont fortement agrégées et résumées. Les données sont sauvegardées localement si les applications mobiles ont besoins plus de données il faut formuler une requête au serveur OLAP via le middleware qui se charge de préparer les données à l'utilisation sous format 2D [16].

- **Travaux de Sharaf et al:**

Sharaf et al [19] proposent un algorithme d'ordonnancement (scheduling) nommé **STOBS**, ce dernier exploite la technique des tables sommaires ou résumées et la possibilité du client mobile à exécuter les requêtes simples. Il permet de maximiser l'agrégation des données partagées par les clients (en utilisant les sémantiques communes entre les tuples) et réduit la taille de diffusion. Il propage les données en mode de diffusion, pour cela les auteurs ont présenté un protocole permettant une diffusion des tables résumées. Il a essayé de mettre en évidence que la diffusion est mieux qu'une transmission point à point [19].

7.2 Techniques de compression des cubes de données :

Une des contraintes est l'espace de mémoire de traitement et de stockage réduites dans les dispositifs mobiles, ce qui à mener à chercher des techniques permettant la réduction de la taille des données provenant des entrepôts de données. Ces techniques visent à réduire la taille des cubes de données sans perte d'informations.

- **Travaux de Wang et al, 2002:**

Wang et al [20] ont proposé une nouvelle approche pour réduire la taille du cube de données et de se fait de son temps de calcul. La technique explore les propriétés des tuples simples de sorte que le nombre de tuples dans un cube soit condensé en un seul tuple sans qu'il y ait perte d'informations en utilisant un nouvel opérateur Cube By (Group by multidimensionnel). Une simple opération de décompression permet de décompresser le tuple représentatif en un ensemble de tuples qu'il cache. Un schéma de condensation nommé **BST** (base single tuple) est introduit. Un algorithme heuristique est utilisé pour calculer le plus petit BST [20].

CHAPITRE II : OLAP Mobile

- **Travaux de Lakshmanan et al 2002/2003 :**

Lakshmanan et al [21] proposent la méthode **Quotient Cube** pour la compression d'un cube de données en résumant son contenu sémantique et en le structurant sous forme de partitions de classes. La meilleure partition n'est pas seulement celle qui permet de réduire la taille du cube mais aussi celle qui permet de conserver une structure de treillis valide donnant la possibilité de naviguer avec les opérations d'agrégation (Roll-Up) et de spécification (Drill-Down) dans le cube réduit. Malheureusement, la technique des Quotient Cube fournit des structures peu compactes. De plus, ces structures ne sont pas adaptées aux mises à jour des données.

Lakshmanan et al [22] proposent une nouvelle version améliorée **QC-Tree (Quotient Cube Tree)** qui pallie les limites de la technique des Quotient Cube. QC-Tree permet de rechercher les structures compactes de données dans un cube, d'extraire et de construire les cubes intéressants à partir des données mises à jour.

- **Travaux de Michalarias et al :**

Michalarias et al [23] ont proposé une structure physique de cube de données fortement compressée (**m-dwarf**) qui ne fait pas perdre les informations sémantiques et qui est désigné à être utilisé dans les applications mobiles. **m-dwarf** peut être intégré dans l'architecture **FCLOS** [23] (voir la figure 2.6).

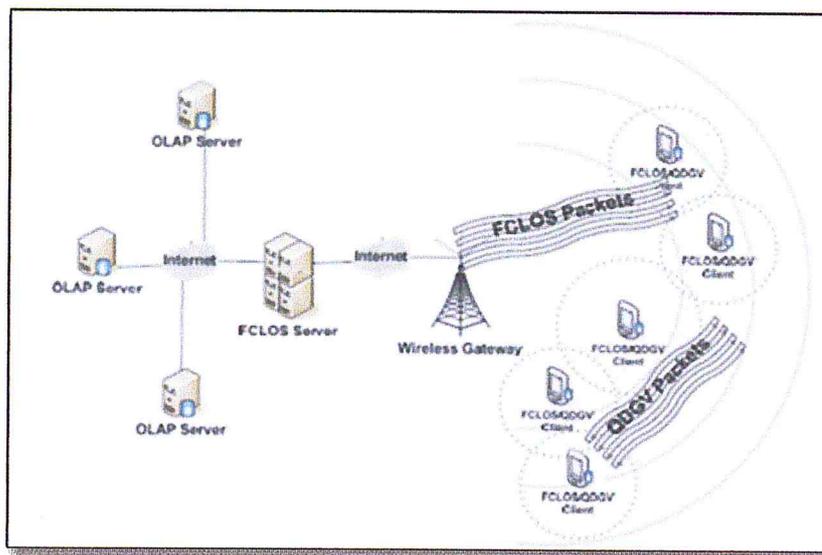


Figure 2.6: Architecture FCLOS [18].

FCLOS est un algorithme de scheduling explicitement conçu pour la diffusion des données agrégées dans des infrastructures de réseau dynamique (mobile, sans fil, PDA...). Fclos-md est extension de Fclos qui au lieu de diffuser des tables résumés, diffuse des m-dwarfs [24].

- **Travaux de Cuzzocrea et al ,2008 :**

Cuzzocrea et al [25] ont proposé une technique de compression efficace des cubes de données et une architecture d'un système (basée sur cette compression), nommé **Hand-OLAP** qui permet aux utilisateurs mobiles utilisant des dispositifs mobiles, en appartenant à un réseau sans fil d'extraire et voire les informations venant d'un serveur OLAP lointain distribué sur un réseau câblé (voir figure 2.7) [25]. L'idée dont le système est fondé sur : plutôt que d'interroger les cubes de données multidimensionnels originaux (étant constamment raccordé à WLAN), cela peut être plus convenable pour les utilisateurs mobiles de télécharger et maintenir à l'intérieur de leur dispositif mobile une vue OLAP bidimensionnelle compressée de données qu'ils veulent traiter, et interroger hors ligne. **Hand-OLAP** adopte une représentation partagée de la vue OLAP compressé. La principale caractéristique de Hand-OLAP fournit ainsi des réponses approximatives et d'une manière rapide aux requêtes OLAP [25].

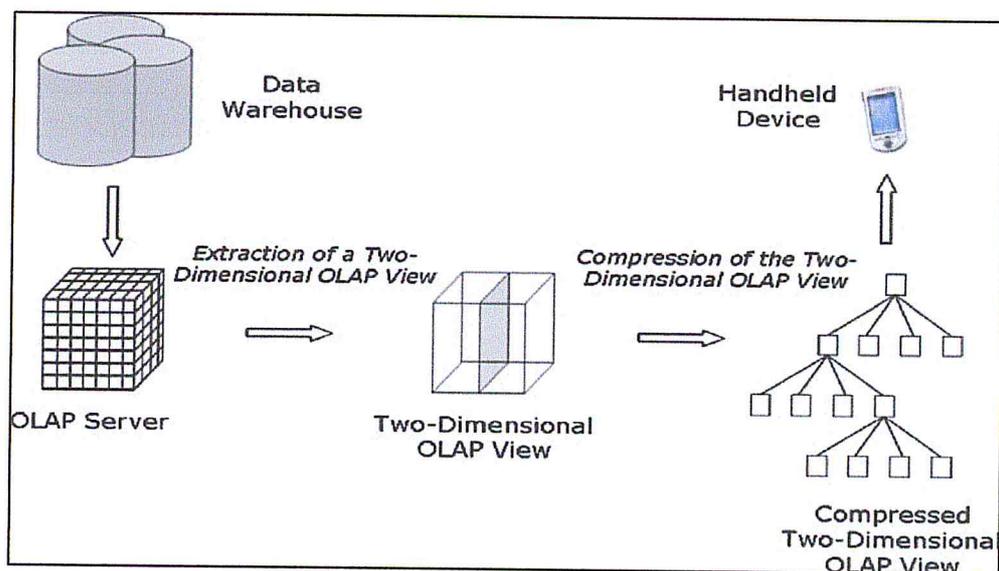


Figure 2.7 : Figure 2.7:extraction et compression de la vue OLAP bidimensionnelle dans Hand-OLAP [25]

CHAPITRE II : OLAP Mobile

Le concept de client en mode connecté et déconnecté est abordé : lorsque la connexion au réseau sans-fil n'est plus disponible, la vue compressée stockée localement sur l'appareil mobile est utilisée pour répondre aux requêtes de l'utilisateur. Toutefois, les auteurs de [1] sont d'avis que le fait de limiter les clients OLAP mobiles à seulement deux dimensions, à des niveaux agrégés et à des données approximatives peut être trop restrictif pour certaines applications, par exemple dans une situation d'urgence où l'accès à des données locales détaillées et précises peut s'avérer nécessaire. Ils ont constaté qu'il est préférable d'offrir la possibilité du choix des dimensions à inclure, le niveau de détail du forage et la compression ou non des cubes aux décideurs.

Les méthodes citées ci-dessus (compression) permettent de réduire la taille des données d'un cube de données OLAP mais elles souffrent de quelques problèmes suivants :

- Ces méthodes retournent des résultats approximatifs et non détaillés que veut voire exactement l'utilisateur.
- elles peuvent causer aussi une perte d'information.
- elles peuvent retourner des résultats non sollicités par l'utilisateur, cela augmente la taille du cube reçu avec des données non pertinentes.

Le tableau (Tableau 2.1) suivant représente les inconvénients des travaux de compression qui cités précédemment :

Travaux	Perte d'information	Résultats approximatifs	Données non pertinents
Travaux de Wang et al [20]		x	x
Lakshmanan et al [21],[22]	x		x
Michalarias et al [23], [24]		x	x
Cuzzocrea et al [25]	x		x

Tableau 2.1 : Synthèse des travaux de compression dans mobile OLAP

Il est préférable que ces données expriment les besoins exactes de l'utilisateur pour aide à la décision.

Nous allons présenter dans la partie suivante (le chapitre 4) une technique permettant de pallier à ces problèmes ; c'est la technique de compression par personnalisation des données.

8. Conclusion :

Dans ce chapitre nous avons d'abord présenté quelques concepts liés à l'informatique mobile, nous nous sommes focalisés aux systèmes OLAP mobile. Nous avons évoqué alors les différentes contraintes des systèmes utilisant Mobile OLAP, et invoqué les travaux existants faites dans ce contexte (technique de compression des cubes, intégration des préférences des utilisateurs dans les requêtes, présentation d'un modèle personnalisé de l'utilisateur).

Le prochain chapitre nous allons parler sur la notion de personnalisation et surtout la personnalisation dans les entrepôts de données.

CHAPITRE III :

La personnalisation
dans les entrepôts de
données

1. Introduction :

La personnalisation constitue un enjeu capital pour l'industrie informatique. En effet, le problème de personnalisation se pose dès lors qu'un service est offert en réponse à des besoins de l'utilisateur. Par exemple, dans le domaine de l'interaction Homme-Machine, l'adaptation de l'interface aux caractéristiques de l'utilisateur (novice ou expert, personne handicapée ou valide, jeune ou âgé, ...) constitue un facteur clé du succès ou du rejet d'un système. Plus particulièrement, dans le domaine des systèmes d'informations, la personnalisation a été abordée par différentes communautés scientifiques telles qu'en bases de données [26], [27], [28], [29] et en recherche d'information [30], [31]. Elle a été définie comme solution à deux problèmes mutuels. D'une part, l'augmentation de l'information disponible suite à l'essor des technologies de l'information et l'avènement d'Internet engendre des résultats massifs comportant des données sans intérêt. D'autre part, le manque d'expressivité des langages d'interrogation des données, tels que SQL en bases de données et les requêtes sous forme de mots clés en recherche d'information, rend la tâche de description des informations recherchées difficile. Ainsi, la personnalisation a pour objectif, d'une part, de faciliter l'expression des besoins de l'utilisateur et, d'autre part, de sélectionner des informations pertinentes.

2. La notion de personnalisation

La personnalisation de l'information est une dimension qui permet la mise en œuvre d'un système centré utilisateur non dans le sens d'un utilisateur générique mais d'un utilisateur spécifique [32].

2.1 Définition :

La personnalisation de l'information se définit, entre autres, par un ensemble de préférences individuelles représentées par des couples (attribut, valeur), par des ordonnancements de critères ou par des règles sémantiques spécifiques à chaque utilisateur ou communauté d'utilisateurs. Ces modes de spécification servent à décrire le centre d'intérêt de l'utilisateur, le niveau de qualité des données qu'il désire ou des modalités de présentation de ces données [27].

2.2 Personnalisation et recommandation

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

La recommandation est un service de personnalisation qui consiste à proposer à l'utilisateur des éléments vis-à-vis de ses préférences ou en servant à l'expérience des autres utilisateurs.

Les deux processus de recommandation et de personnalisation de l'information diffèrent sur le contenu de la réponse :

- La personnalisation permet de réduire le contenu du résultat qui consiste en un sous ensemble de la réponse à la requête initial.
- La recommandation permet de compléter le résultat personnalisé .elle consiste à enrichir le contenu de résultat avec de nouvelles informations qui ne sont pas demandées par la requête initial de l'utilisateur .ainsi, elle garantit une réponse non vide qui est possible dans la personnalisation.

Quel que soit le domaine technologique, la personnalisation de l'information peut être exploitée selon deux modes de gestion : en recommandation ou en interrogation.

- Les systèmes de recommandation [33] exploitent les profils des utilisateurs ou de communautés d'utilisateurs pour disséminer des offres ciblées sur les centres d'intérêt et les préférences de ces derniers.
- La personnalisation en interrogation [33] consiste à adapter l'évaluation d'une requête par rapport aux caractéristiques et aux préférences de l'utilisateur qui l'a émise. Dans ce contexte, le système réagit à une demande spécifique de l'utilisateur en transformant sa requête afin de la rendre plus précise, en personnalisant l'affichage des résultats

3. La notion du profil :

3.1 Définitions :

1. Un profil utilisateur est une collection d'informations sur l'utilisateur. Cette collection peut être vue comme un ensemble de caractéristiques avec des valeurs associées contenant par exemple ce que l'utilisateur préfère, ce qu'il est capable de faire...etc. On doit également prendre en compte l'historique des actions de l'utilisateur, voir leur évolution dans le temps [8].
2. Un profil utilisateur regroupe l'ensemble des connaissances nécessaires à une évaluation efficace des requêtes et à une production d'une information pertinente adaptée à chaque utilisateur [34].

3.2 Le contenu du profil :

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

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

3.3 Modélisation du profil utilisateur :

L'introduction de la dimension utilisateur dans un processus d'accès à l'information, mérite, voire nécessite une réflexion sur la modélisation de l'entité *utilisateur*. Le système d'accès personnalisé à l'information, est un processus caractérisé par trois phases :

1. définition d'une représentation des unités d'information caractérisant l'utilisateur du système. Elle correspond à la définition de la structure du profil utilisateur.
2. l'instanciation de cette représentation au cours d'une activité d'accès à l'information pour un utilisateur particulier. Elle regroupe des techniques d'acquisition des données utilisateur ainsi que des approches de construction pour agencer les informations collectées selon la structure représentative définie lors de l'étape précédente .

3. l'évolution du profil au cours du temps. Elle nécessite la mise en place de stratégies de mise à jour du contenu informationnel du profil.

3.4 Construction du profil :

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

1. l'acquisition et la collecte des données utilisateur ;
2. puis la construction proprement dite du profil [35].

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

a) *L'acquisition explicite :*

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

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

b) *L'acquisition implicite (apprentissage) :*

L'acquisition implicite consiste à collecter les informations décrivant l'utilisateur, en observant les dimensions et les membres fréquemment sollicités et en considérant les caractéristiques de l'environnement à partir duquel il intervient (les capacités et les limites du dispositif utilisé lors de ces interactions). Et ce, en se basant sur l'historique de ses interactions avec le système. Le principe est que l'utilisateur doit avoir le contrôle de son profil à tout moment afin de pouvoir invalider les mises à jour incorrectes du profil. Après une telle invalidation, le gestionnaire doit prendre en compte et modifier la manière (le processus) de gestion et de mise à jour du profil.

3.5 Exploitation de profil :

Le contenu du profil d'un utilisateur peut être utilisé à différents moments de cycle de vie de la requête. Il peut servir pour enrichir le contenu de la requête, ou pour optimiser son exécution ou encore être utilisé pour adapter les résultats selon les modalités de présentation.

Les figures : 3.1, 3.2, 3.3, illustrent les principales phases du processus d'accès personnalisé à l'information selon le cycle de vie de la requête où l'on prend en compte le profil utilisateur, à savoir :

1. **A la phase d'exécution de la requête :** lors de cette phase le contenu informationnel du profil peut être exploité directement dans le processus d'exécution de la requête. L'intégration du profil lors de la phase d'exécution de la requête est beaucoup plus utilisée dans les systèmes de recherche d'information (RI).

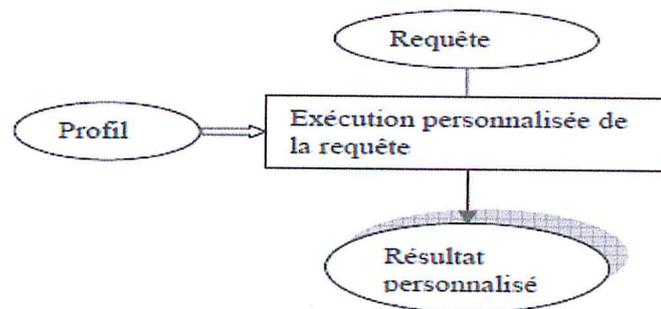


Figure 3.1: L'intégration du profil lors de la phase d'exécution de la requête[6]

2. **A la phase de présentation des résultats :** La personnalisation à ce stade du processus d'accès à l'information offre une solution en réordonnant les résultats pour ne présenter à l'utilisateur que les informations pertinentes en réponse à son besoin en information. Ce besoin est formulé en conjuguant les informations données par l'utilisateur (la requête soumise) et celles extraites de son profil. Ainsi, la restitution des résultats s'effectue en fonction de la notion de pertinence personnelle de l'utilisateur. Dans le contexte des entrepôts de données, la personnalisation consiste à exploiter les informations du profil utilisateur sur l'aspect visualisation qui est primordial dans le contexte de l'analyse en ligne.

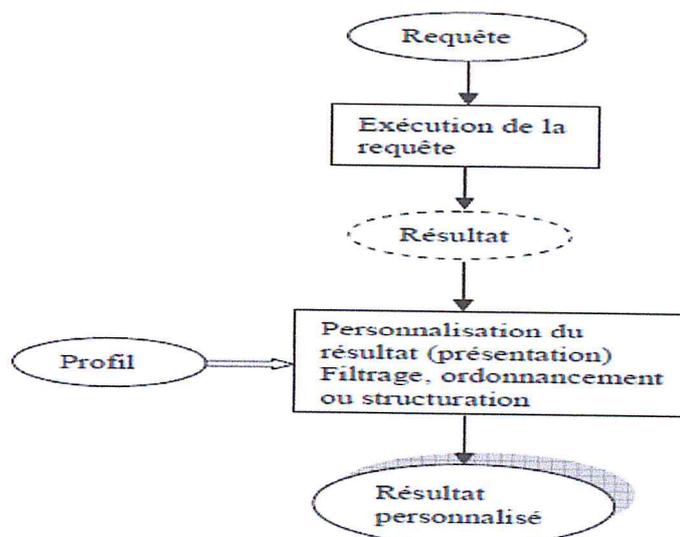


Figure 3.2: L'intégration du profil lors de la phase d'exécution de la requête et la visualisation des résultats [8]

3. A la phase d'affinement de la requête : Les requêtes utilisateur sont assurément une source évidente importante pour l'identification des besoins en information de l'utilisateur. Néanmoins, les utilisateurs soumettent souvent des requêtes très courtes et ambiguës. L'objectif de la personnalisation à ce stade du cycle de vie de la requête est de clarifier le besoin en information de l'utilisateur en se basant sur son profil. Ainsi, la reformulation de requête dans ce cadre intègre les composantes informationnelles issues du profil de l'utilisateur pour identifier, enrichir et cibler son intention.

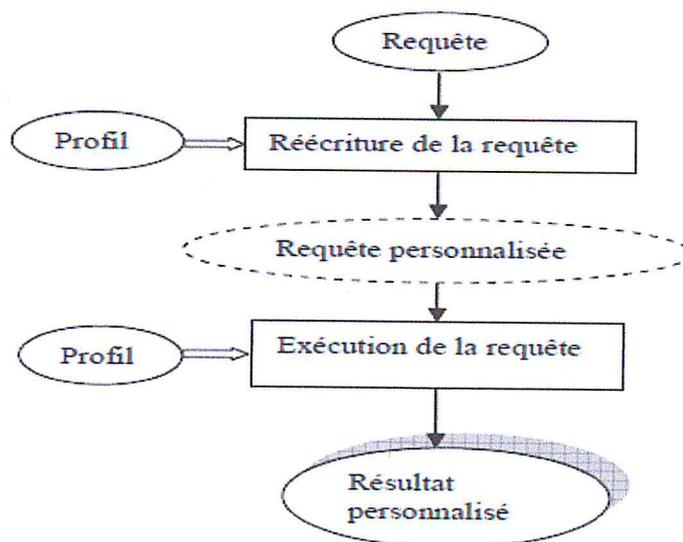


Figure 3.3 : L'intégration du profil lors de la phase de présentation des résultats [8]

4. Les systèmes d'accès personnalisé à l'information :

Un système d'accès personnalisé à l'information est un système qui intègre l'utilisateur, en tant que structure informationnelle, tout au long de la chaîne d'accès à l'information.

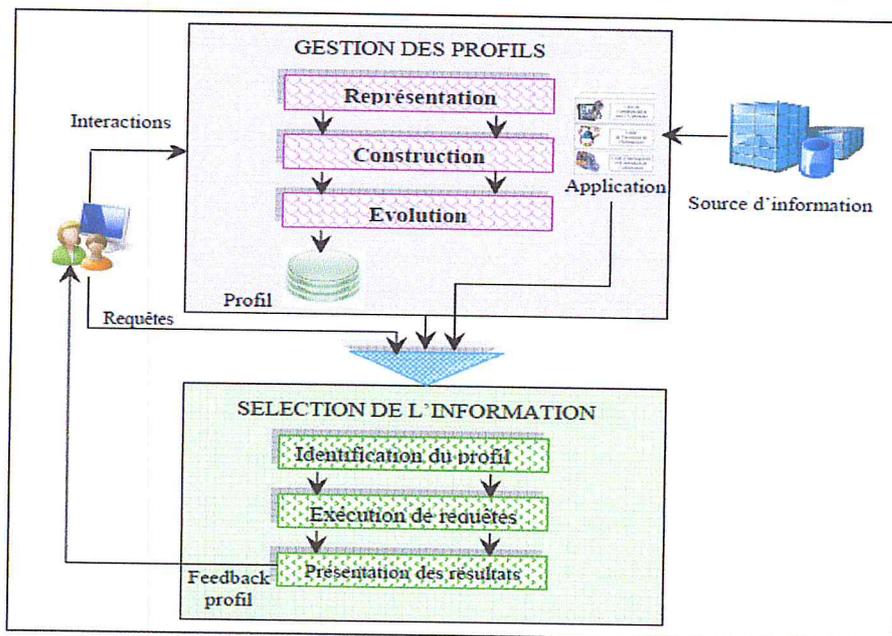


Figure 3.4: Architecture fonctionnelle d'un système d'accès personnalisé à l'information [36]

Cette architecture (figure 3.4) est centrée autour de l'utilisateur, sur la base de cette architecture nous dégagons principalement deux fonctions fondamentales qui sont la gestion des profils et la sélection de l'information.

5. La notion de préférences :

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

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

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

5.1 Niveau des préférences :

Les préférences dans les BDM doivent spécifier le chemin des données que l'utilisateur désire analyser [37]. Ainsi, les modèles des préférences OLAP ont porté sur le schéma ainsi que les valeurs de la BDM.

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

- Les préférences sur les dimensions [33] décrivent l'ensemble des dimensions pertinentes pour l'analyse d'un fait.
- Les préférences sur les paramètres [38], [39] spécifient les niveaux de granularité préférés au long d'une dimension

5.2 Formulation de préférences :

Les travaux de modélisation des préférences OLAP se sont basés sur des approches initialement définies pour les bases de données. Les préférences sont formulées dans ces approches selon une approche quantitative [40], [26], [29] ou qualitative [41], [42].

- ✓ *L'approche qualitative*, les préférences sur un ensemble sont formulées par des relations d'ordre entre ses éléments.
- ✓ *L'approche quantitative*, permet d'exprimer les préférences d'une façon indirecte par l'utilisation de fonctions de score qui associent un nombre réel à chaque n-uplet du résultat de la requête. Dans ce cas, un n-uplet₁ est préféré à un n-uplet₂ si son score est supérieur à celui de t₂.

5.3 Approches qualitatives Vs Approches quantitatives :

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

5.4 Contextualisation :

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

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

Le contexte est toute information susceptible de caractériser la situation d'une entité. Une entité est une personne, un lieu ou un objet qui est considéré pertinent pour l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application [43]. Il existe deux catégories de contextes : les contextes internes et les contextes externes.

- Un *contexte externe* : définit une situation externe à la base de données. Il est typiquement défini par des attributs spécifiques appelés paramètres de contexte [44]. L'accompagnement de l'utilisateur, son emplacement et la date courante sont des exemples de paramètres de contexte.
- Un *contexte interne* : est décrit par des attributs ou des conditions sur les valeurs de la base de données ou de la requête. Les contextes internes dans [45], [42] sont définis par des conditions sur la présence de valeurs spécifiques d'attributs de la base de données.

Exemple : Considérons une préférence de l'utilisateur pour les publications en entrepôts de données. Un contexte interne associé à cette préférence peut être « l'année de publication est supérieure à 2000 ». Un exemple de contexte externe pour cette préférence est « en réunion de travail ».

5.5 Exploitation de préférences :

Durant un processus de personnalisation basé sur les préférences, différents problèmes se posent par rapport au traitement des préférences : Comment exprimer les préférences ? Comment sélectionner celles qui sont pertinentes ? Comment intégrer ces préférences dans la requête ? Quels impacts auront-elles sur l'exécution de la requête ?

5.5.1 Expression des préférences :

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

Les préférences quantitatives sont formulées dans [39] par des règles ECA (Evènement-Condition-Action) qui associent un poids à un attribut de dimension lorsqu'une opération OLAP est invoquée. Cette action est contrainte par la satisfaction d'une condition.

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

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

Golfarelli et Rizzi [38] ont défini une algèbre de formulation de préférences qualitatives simples (POS, NEG, CONTAIN, ...). Cette algèbre est enrichie par deux opérateurs de composition de préférences : Pareto (\otimes) pour indiquer le même degré d'importance entre deux préférences et Priorisation (\triangleright) pour définir un ordre de priorité entre préférences. Ces différents opérateurs sont ensuite intégrés dans une requête MDX à l'aide de clause PREFERRING.

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

5.5.2 Sélection des préférences :

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

- **Méthode 1 :** elle est centrée sur l'applicabilité de la préférence. Une préférence **P** est applicable à une requête **Q** si l'exécution de **Q** combinée

conjonctivement avec P ne renvoie pas un résultat vide [37] ou si le score de P est supérieur à un seuil exprimé dans Q [31].

- *Méthode 2* : elle a été proposée en bases de données où les préférences sont sélectionnées si leurs contextes appartiennent avec le contexte de la requête.
- Si une préférence P est rattachée à un contexte interne C , la sélection de P dépend d'une confrontation entre le contexte C et les tuples de la base de données [46] ou les attributs de la requête [45].
- Dans le cas de contexte externe, le contexte courant de l'utilisateur $CC(U)$ est d'abord détecté. Une préférence est sélectionnée si son contexte apparie avec $CC(U)$. Selon [47], [44].l'appariement de contexte revient à déterminer les contextes qui sont égaux ou plus généraux que $CC(U)$. Par exemple, si l'utilisateur est localisé à Toulouse au moment de la requête, les préférences qui sont associées à la localisation France sont sélectionnées.

6. Personnalisation des systèmes OLAP :

La personnalisation dans le cadre des entrepôts de données présente un réel intérêt dans un contexte où les analyses et la prise de décision sont réalisées par l'utilisateur lui-même. Ce dernier a sa propre connaissance des données visualisées lors de la navigation et donc des besoins qui peuvent lui être propre, d'où cet intérêt pour la personnalisation. Autrement dit, l'utilisateur est en interaction directe avec le système au niveau de l'analyse des données, en particulier dans le contexte de la navigation.

Les travaux sur la personnalisation dans la communauté des bases de données multidimensionnelles ont porté sur les différents niveaux de l'architecture des systèmes OLAP. Nous présentons dans la suite les travaux de personnalisation par niveau tel que :

- la personnalisation du schéma multidimensionnel.
- la personnalisation de l'interrogation des données.
- la personnalisation de la visualisation des données.
- la personnalisation de la prise de décision.

6.1 Personnalisation du schéma OLAP :

Après la conception du schéma de la BDM, les besoins de l'utilisateur peuvent évoluer dans le temps [48], [49], [50]. Les travaux sur la personnalisation du schéma multidimensionnel répondent à la problématique suivante :

Comment adapter le schéma d'une base de données multidimensionnelle aux besoins évolutifs de chaque utilisateur ?

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

- *L'évolution du schéma* : elle est centrée sur la mise à jour des hiérarchies des dimensions (l'ajout ou la suppression d'un niveau de granularité et l'ajout ou la suppression d'une instance de dimension).
- *La gestion de vues du schéma* : elle permet de générer une vue personnalisée du schéma en fonction du profil de l'utilisateur courant [48].

❖ *Travaux de Favre, 2007 :*

Afin d'enrichir les possibilités d'analyse d'un entrepôt de données, *Favre* [49] propose une approche permettant la création de niveaux d'analyse supplémentaires dans les hiérarchies de dimension ou définissant de nouvelles hiérarchies de dimension a été faite. Cette approche permet aux utilisateurs d'intégrer leurs propres connaissances sur la façon d'agréger les données sous la forme de règles de type « si-alors ». Supposons qu'un utilisateur veuille analyser les ventes du produit « *food* », non pas par « *year* » ni par « *day* » et non plus par « *month* », mais par semaine, information qui n'est pas présente dans l'entrepôt. L'utilisateur peut alors exprimer ses connaissances sur les types de Time, afin de créer un niveau « *semaine* », correspondant à une nouvelle hiérarchie de la dimension « *Time* ». L'utilisateur pourra ainsi réaliser des analyses des ventes du produit « *food* » par semaine .

❖ *Travaux de Garrigós et al, 2009 :*

Le mécanisme de personnalisation selon *Garrigós et al* [48] débute depuis l'étape de conception de la BDM. Durant cette étape, un profil de l'utilisateur est défini ainsi qu'un ensemble de règles ECA (Evènement, Condition, Action) précisant les actions de personnalisation à effectuer en ligne. La deuxième étape de ce mécanisme est effectuée au moment de l'interrogation de la BDM. Elle permet de

générer une vue du schéma en fonction du profil de l'utilisateur courant afin de faciliter la tâche de formulation de requête.

En ce qui concerne l'*algorithme*, les auteurs proposent de générer une partie du schéma de la BDM (approche descendante) à partir du profil de l'utilisateur et d'un ensemble de règles. Contrairement à [51] où l'ensemble des règles ECA constitue le profil de l'utilisateur, les règles proposées dans ce travail permettent de mettre à jour le profil et d'adapter le schéma de la BDM en fonction du profil.

6.2 Personnalisation de l'interrogation des données :

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

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

Deux catégories de travaux peuvent être distinguées:

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

6.2.1 Personnalisation de requêtes :

La personnalisation de requête est un mécanisme effectué avant ou après l'évaluation de la requête afin de changer la requête ou l'ordre du résultat. L'objectif de cette approche est de restituer les données les plus pertinentes pour chaque utilisateur.

❖ *Travaux de Ravat et Teste, 2008 :*

Ravat et Teste [39] proposent une solution pour la personnalisation de la navigation OLAP en exploitant des préférences exprimées par des poids sur les éléments du schéma de la base de données multidimensionnelles, des poids reflétant l'intérêt de l'utilisateur.

Dans ce cas, l'utilisateur assigne des poids aux concepts multidimensionnels afin d'obtenir directement les analyses désirées, évitant ainsi des opérations de navigation.

Un système basé sur des règles ECA (Event, Condition, Action) permet de générer des tables multidimensionnelles contenant uniquement les données identifiées comme pertinentes en fonction des poids. Cette solution quantitative permet de simplifier l'expression des requêtes d'analyse.

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

❖ *Travaux de Golfarelli et al 2009, 2011 :*

Golfarelli et Rizzi (2009) [38], Golfarelli et al (2011) [37] traitent la problématique des résultats volumineux ou vides des requêtes OLAP. Inspirés des travaux de [41] en bases de données, ils se basent sur le modèle BMO (« Best Matches Only ») d'exécution de requêtes où seulement les tuples du résultat qui ne sont pas dominés par d'autres sont renvoyés.

Dans cette 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. Plus précisément, l'exécution de la requête suit une démarche non proactive qui évalue d'une manière incrémentale des requêtes enrichies par des combinaisons de préférences jusqu'à trouver une requête qui renvoie un résultat non vide.

6.2.2 Recommandation de 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. Elle répond à la problématique suivante :

Comment guider l'utilisateur dans l'exploration de la base de données multidimensionnelle afin de l'aider dans son processus de prise de décision?

La recommandation de requête fournit deux fonctionnalités :

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

❖ *Travaux de Giacometti et al 2008,2009 :*

Giacometti et al (2008,2009) [52],[53] proposent un système de recommandation d'analyses multidimensionnelles en se basant sur la navigation qu'effectue un utilisateur donné par rapport aux navigations réalisées par les autres utilisateurs.

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

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 .

6.3 Personnalisation de la visualisation des données :

La personnalisation de la visualisation des données est l'action d'adapter l'interface de visualisation des données en fonction d'un modèle de l'utilisateur ou de critères externes tels que le type et la taille du dispositif utilisé.

Le problème du volume souvent important du résultat des requêtes OLAP a fait l'objet d'étude de la personnalisation de la visualisation dans les bases de données multidimensionnelles [54]. Les travaux dans cet axe ont porté sur :

- La personnalisation de la structure d'affichage du résultat d'une requête, en définissant par exemple la disposition des données sur les axes d'une table multidimensionnelle [55].
- La mise en valeur de certains indicateurs décisionnels, par exemple les chemins de navigation les plus visités [48].

❖ *Travaux de Bellatreche et al, 2005,2006 :*

Les travaux de Bellatreche et al (2005,2006) [33], [55] se concentrent sur la personnalisation des réponses aux requêtes utilisateur interrogeant les données du cube, c.-à-d. des visualisations. Dans cette approche, la visualisation est sous forme d'une table croisée.

L'objectif de ces travaux est de pouvoir fournir à l'utilisateur un résultat focalisé sur son centre d'intérêt, tout en prenant en compte des contraintes de visualisation.

Dans cette approche, la personnalisation consiste à transformer la requête utilisateur avant l'accès au cube de données. Dans cette méthode, l'utilisateur est interrogé afin de donner ses préférences et une contrainte de visualisation. Ces préférences utilisateurs qui classent les membres, les dimensions et la contrainte de visualisation qui contrôlent les résultats sont stockées dans le profil utilisateur. Et une définition du profil utilisateur dans le contexte OLAP est proposée.

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

Les préférences utilisateur définissent une relation de pré-ordre total pour l'ensemble des membres de toutes les dimensions. Cette relation de pré-ordre total sur les membres utilisée pour définir une relation monotone de pré-ordre total sur les cubes. Par exemple, pour les membres m_1 est plus intéressant que m_2 veut dire que $m_1 \leq m_2$. Pour les cubes si le cube C_1 est le sous cube de C_2 ($C_1 \subseteq C_2$), alors C_1 moins intéressant que C_2 et $C_1 \leq C_2$.

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

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. Par ailleurs, ces travaux se limitent au filtrage des valeurs des attributs affichés par la requête. De plus, seules les préférences sous forme de prédicats égalitaires sont prises en compte. Les préférences du type « Nombre de publications > 10 » ne sont pas supportées. Ceci représente une limite pour la personnalisation des valeurs des attributs numériques, notamment les mesures.

6.4 Personnalisation de la prise de décision :

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

Une première approche permet d'aider l'utilisateur à mieux interpréter les données multidimensionnelles afin de prendre une décision pertinente.

❖ Travaux de Cabanac et al ,2007 :

Les travaux de *Cabanac et al* [56] proposent d'intégrer des annotations sur le schéma et sur les valeurs d'une BDM afin de conserver les commentaires, les réflexions et les explications formulés lors des analyses. Ces annotations sont restituées par le système OLAP conjointement aux éléments auxquels elles sont rattachées [57]. Elles sont utilisées à des fins personnelles (réutilisation des réflexions précédentes) ou collectives (confrontation de différentes interprétations) [56].

Une deuxième approche permet d'aller au-delà de l'assistance à la prise de décision par l'automatisation de la prise de décision [58]. Des règles d'analyse sont définies par l'utilisateur afin de traduire le processus de prise de décision manuel. Une action, correspondant à une décision (par exemple changer le prix d'un article) est effectuée suite à un événement au niveau des bases de données sources (par exemple la diminution de la quantité en stock).

6.5 Intégration du profil et contexte de l'utilisateur dans OLAP mobile :

On citera ci-après quelques-unes des recherches qui ont ouvert la porte à la personnalisation d'OLAP en général et mobile OLAP en particulier [2].

- **Travaux de Li et al :**

Li et al [59] ont proposé un cube graduel qui réduit la taille des données transmises, fournit une stratégie de transmission personnalisée et donne la possibilité aux utilisateurs finaux de visualiser les données hors ligne, il a essayé d'utiliser la notion du profil dans le choix du contour (limites ou bornes) du cube à transmettre. Le profil est décrit ici par les précisions décrites par l'utilisateur, ils permettent de définir les limites des bornes du cube à traiter. Ce profil peut être généré par plusieurs types d'algorithmes (aléatoire, optimal, heuristique), l'heuristique décrit le mieux le profil puisqu'il procure une meilleure qualité et plus de performance dans la description du profil.

- **Travaux de Rizzi et al :**

Le profil ainsi décrit ne reflète pas réellement les caractéristiques des utilisateurs cet axe (profil) n'a été touché que dernièrement ; les chercheurs qui ont apporté un plus à la personnalisation sont : *Rizzi et al* [60] ont proposé un agenda de recherche pour les préférences OLAP, ils ont mentionné que l'expression des préférences dans les requêtes des BDMs est naturelle, et de là les réponses seraient plus pertinentes et non bruitées ou vides.

- **Travaux de Jerbi et al, 2008 :**

Il s'agit non plus d'exploiter des poids, mais plutôt des ordres (représentation qualitative des préférences), ce qui rend la tâche plus aisée pour l'utilisateur. En outre, ces ordres ne sont pas exprimés de façon absolue, mais par rapport à un contexte d'analyse donné. Ceci permet de prendre en compte le fait que les préférences peuvent varier d'un contexte d'analyse à un autre. *Jerbi et al* [61] poursuivent ces travaux en présentant un environnement OLAP intégrant des mécanismes de recommandation contextuelle des requêtes.

CHAPITRE III: Personnalisation dans les entrepôts de données

Les méthodes citées ci-dessus permettent d'avoir des données pertinentes qui sont très importantes pour les décideurs mais ne prennent pas en considération le volume énorme de ces données.

Nous allons dresser une comparaison des différents travaux étudiés sur les différents types de personnalisation des systèmes OLAP dans le tableau suivant :

Travaux	Personnalisation de requête	Recommandation de requête	Catégorie	Entrée	Sortie	Timing
<i>Favre, 2007 [49]</i>	X		centrée utilisateur	-Règles de type « Si_ Alors » -schéma BDM(S)	schéma(S')	Avant l'évaluation de la requête
<i>Garrigós et al, 2009 [48]</i>	X		centrée utilisateur	-schéma BDM(S) -profil -règles ECA	schéma(S')	Durant conception, Durant définition de requête
<i>Ravat et Teste, 2008[39]</i>	X		Centrée utilisateur	-requête Q -profil comporte règle ECA -schéma BDM	Requête Q'	Avant l'évaluation de la requête
<i>Golfarelli et al 2009, 2011[37] [38]</i>	X		Centrée utilisateur	-requête Q -préférences -instance BDM	tuples	Après l'évaluation de requête
<i>Giacometti et al 2008,2009 [52],[53]</i>		X	Centrée historique	-schéma BDM -instance BDM -log d'analyses -analyse courante -requête Q	Requête Q'	Après l'évaluation de la requête
<i>Bellatrech et al 2005,2006[33], [55]</i>	X		Centrée utilisateur	-requête Q -profil -contraintes -instance BDM	$Q' \subseteq Q$	Avant l'évaluation de requête
<i>Cabanac et al ,2007 [56]</i>		X	Centrée historique	schéma BDM -instance BDM -log d'analyses -analyse courante -requête Q	schéma(S')	Durant conception, Durant définition de requête

Li et al [59]	X		centrée utilisateur	-schéma BDM(S) -profil -règles ECA	schéma(S')	Durant conception, Durant définition de requête
Rizzi et al [60]	X		Centrée utilisateur	-requête Q -profil -schéma BDM	Requête Q'	Avant l'évaluation de la requête
Jerbi et al, 2008 [61]		X	Centrée historique	-schéma BDM - contraintes -log d'analyses -analyse courante -requête Q	Requête Q'	Après l'évaluation de la requête

Tableau 3.1 : Synthèse des travaux sur la personnalisation de l'interrogation des bases de données multidimensionnelles

La personnalisation est l'une des techniques de compression qui peut résoudre les problèmes que nous avons mentionnés précédemment, Nous nous intéressons sur la personnalisation des requêtes quantitatives.

7 Conclusion

Dans ce chapitre, nous avons évoqué les différents concepts liés à la personnalisation concernant la notion du profil, la notion des préférences, la différence entre personnalisation et recommandation, les différents types de personnalisation des systèmes OLAP. Et pour chaque type de personnalisation nous avons fait un rapide survol des différents travaux effectués.

Par la suite, nous allons attaquer le noyau de notre thème qui est la proposition d'un prototype de système mobile OLAP dont nous allons proposer notre méthode qui est la technique de compression des cubes de données par la personnalisation des données.

PARTIE II :

*La Démarche De
Développement*

CHAPITRE IV :

Architecture pour un système OLAP Mobile

1. Introduction

Afin de répondre aux visions des techniques de compression des cubes OLAP qui sollicitent d'avoir des résultats sans perte d'information ; la personnalisation des données peut être considéré comme une technique de compression ; apporte une solution à cette contrainte car elle peut mettre à disposition des utilisateurs (analystes, décideurs) des cubes OLAP implicitement compressés avec un taux de perte d'information diminué en plus qu'elle offre des résultats pertinents (selon les choix de l'utilisateur) .

Pourquoi le choix de la technique de personnalisation ?

D'après ce qu'on a cité précédemment dans la partie ETAT DE L'ART ; chapitre OLAP Mobile, il existe pas mal de travaux qui s'intéressent sur la compression des cubes OLAP.

Des auteurs ont proposé des techniques de compression efficaces comme **Alfredo Cuzzocrea et al [25]** qui définissent Hand-OLAP ; une technique fournit des partitions d'un cube de données situé sur un serveur OLAP distant, sa principale caractéristique c'est qu'elle retourne des réponses approximatives (en utilisant des opérateurs d'agrégations : SUM, AVG) aux requêtes OLAP et son architecture est adaptée aux environnements mobiles.

Et **Wang et al [20]**, ses travaux concentrent sur la structure du cube OLAP. Ils proposent une technique qui permet de réduire la taille d'un cube en utilisant l'opérateur d'agrégation d'OLAP **Cube By** qui permet de condenser le nombre des tuples en un seul tuple (**pour plus de détail voir chapitre OLAP Mobile**).

Ces techniques visent à réduire la taille des cubes de données sans perte d'informations. Elles permettent de retourner des réponses approximatives et non détaillées et l'aspect utilisateur n'a pas été respecté (le centre d'intérêt sur les données à analyser est négligeable).

CHAPITRE IV : Architecture pour un système OLAP Mobile

Des travaux de recherches ont été dévoilés sur l'accès personnalisé OLAP comme **Ravat et Teste [39]** qui proposent une solution pour la personnalisation de la navigation OLAP en exploitant des préférences exprimées par des poids sur les éléments du schéma de la base de données multidimensionnelles, des poids reflétant l'intérêt de l'utilisateur.

Cette technique permet de générer des tables multidimensionnelles contenant uniquement les données identifiées comme pertinentes et détaillées en fonction des poids assignés par l'utilisateur.

Pour cela, on peut conclure que la technique de personnalisation des requêtes OLAP a apporté une amélioration aux différentes techniques de compression des cubes OLAP car elle permet d'une part à discriminer les analystes en fonction de leurs centres d'intérêt (domaines d'analyses. Par exemple : achat, vente, location, etc.), de leurs préférences, et à leur délivrer des résultats pertinents et d'autre part ; elle sert à réduire implicitement la taille des cubes résultants de la requête multidimensionnelle personnalisée.

Dans ce chapitre nous allons proposer notre solution informatique qui est un système pour Mobile OLAP, ce système est basé sur la technique de personnalisation des requêtes utilisateurs afin d'obtenir un cube de données compressé adapté aux environnements mobiles.

2. Architecture du Système proposé :

Notre architecture de système est basé sur L'architecture 3-tiers ce que nous présentons par la suite.

2.1 Présentation :

Les systèmes OLAP Mobile, se sont des systèmes OLAP, dans lequel un décideur peut analyser leurs données via les dispositifs mobiles. Ces systèmes (OLAP mobile) offrant une représentation multidimensionnelle des données que les analystes explorent interactivement leurs données à partir d'un terminal mobile. Cette approche a connu un développement important et incontournable grâce à sa capacité de permettre un accès à distance et interactif aux données analysées. Le décideur se trouve confronté à un espace multidimensionnel, souvent très vaste, sur lequel il doit opérer un nombre important de manipulations afin d'obtenir un résultat le plus proche possible de ses besoins.

Le système que nous allons proposer, est un système OLAP mobile, destiné à un groupe d'analystes (Comité directeur, le département commercial central) où à partir de leurs dispositifs mobiles peuvent analyser leurs données et prendre des décisions sur l'état transversale de leurs entreprises à distance.

L'objectif principal est de retourner au client mobile (analyste) un cube de données compressé adapter au leur environnement mobile, un cube de données dont sa taille est compatible avec la capacité mémoire d'un terminal mobile où nous utilisons la technique de personnalisation des requêtes pour la compression de ces cubes.

La figure suivante (Figure 4.2) montre une description générale sur notre architecture de système proposé qui est basé sur une architecture orienté service.

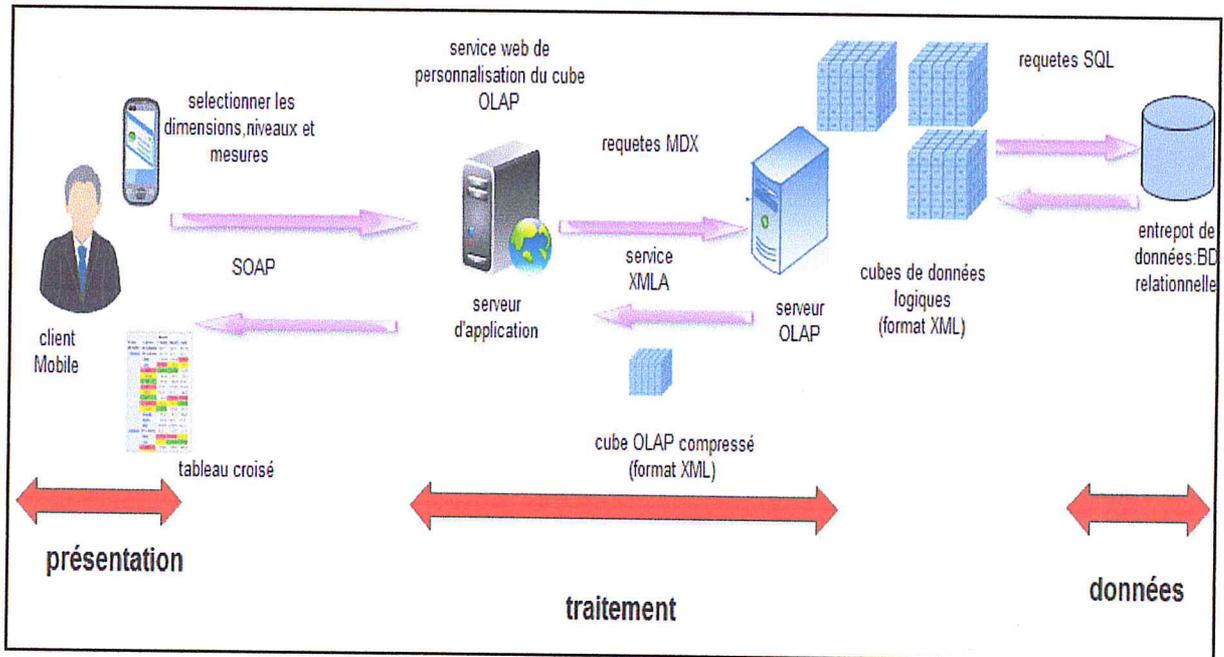


Figure 4.1 : Architecture globale du système

Notre architecture est divisée en trois couches essentielles : *présentation*, *traitements* et *stockage (données)*, le détail de ces couches est présenté dans ce qui suit :

🚦 Couche de Présentation :

La couche présentation est la partie de notre système qui doit être présentée à l'analyste sur son environnement mobile. Elle est composée des fenêtres d'authentification utilisateur, des listes de métadonnées (les noms des cubes, les dimensions, les niveaux, les unités mesurés) qu'un utilisateur peut choisir ce que veut visualiser dans le cube compressé et l'affichage de ces données sous forme d'un tableau croisé dynamique multidimensionnels.

🚦 Couche de Traitements :

Cette couche comporte la partie de traitement des données dans notre système, c'est le noyau du système, Elle regroupe le serveur OLAP et le serveur d'application. Le rôle de cette couche est gérer, traiter, et contrôler les données de l'analyste. Cette couche possède aussi le processus de compression des cubes de données OLAP via la technique de personnalisation sans perte d'information et selon les

choix de l'utilisateur, ce processus permet de résoudre les problématiques de compression des cubes OLAP adaptés aux environnements mobiles (perte d'information, et capacité mémoire du dispositif mobile).

Le processus de compression est réalisé par un service web qui permet de traiter la demande de l'utilisateur (analyste, décideur) sur les métadonnées des cubes OLAP, interroger le serveur OLAP par la génération des requêtes multidimensionnelles personnalisées et retourner les cubes compressés vers les clients mobiles .

Couche de Données :

Cette couche comporte les entrepôts de données de l'analyste, ces entrepôts sont stockés dans un serveur de données (SGBD).

Les serveurs OLAP qui existent dans la couche traitement peuvent accéder et récupérer ces entrepôts pour construire des cubes multidimensionnels (tables d'agrégat) pour l'analyse.

2.2 Les Composants de système :

Notre système proposé est composée par des clients mobiles et des serveurs, chaque serveur est nommé selon le service qu'il offre à l'analyste tel qu'un serveur d'application, un serveur de données et un serveur OLAP. Chaque composant du système à un rôle important, nous détaillons chacun par la suite :

Le Client mobile (dispositif mobile) :

Ce sont des terminaux mobiles pour les analystes qui peuvent analyser leurs données, ces dispositifs possèdent d'une application analytique, cette application fournit une communication sans fil pour récupérer et afficher les cubes personnalisés.

Cette application est responsable à l'envoi au serveur d'application les choix préférés de l'analyste et leurs poids tel que le nom du cube à analyser, ses dimensions, ses niveaux, ses membres, et ses mesures.

✚ Le Serveur d'application :

Le serveur d'application joue le rôle d'intermédiaire entre le client mobile et le serveur OLAP, son rôle est de gérer tous les requêtes envoyées par les clients mobiles, traiter leurs choix préférés en fonction des poids, transformer ces choix aux requêtes MDX personnalisées pour récupérer le cube compressé à partir du serveur OLAP et l'envoyer au client mobile.

✚ Le Serveur OLAP :

Un serveur OLAP est un serveur analytique pour la manipulation de grande capacité de données des décideurs, spécifiquement conçu pour soutenir et agir sur les structures des données multidimensionnelles. Les structures multidimensionnelles sont disposées de telle sorte que chaque élément de données est situé et accessible sur la base de l'intersection des éléments de dimensions définies par les décideurs.

La conception du serveur et de la structure des données sont optimisés pour une récupération rapide ad-hoc de l'information dans n'importe quelle orientation, ainsi que pour la vitesse, le calcul et la transformation des données brutes en fonction des relations des formules flexible ,Le serveur OLAP peut soit organiser physiquement les informations multidimensionnelle transformés pour offrir des temps de réponse rapide et constante aux utilisateurs finaux, ou il peut remplir ses structures de données en temps réel à partir de bases de données relationnelles ou autre, ou d'offrir un choix à la fois.

✚ Le Serveur de données :

Le système de gestion de bases de données relationnel (SGBDR) est le point central d'un entrepôt de données décisionnelles. Il sert principalement à stocker les données de l'entrepôt, et à les récupérer grâce à un langage de requête tel que SQL.

3. Le fonctionnement de Système :

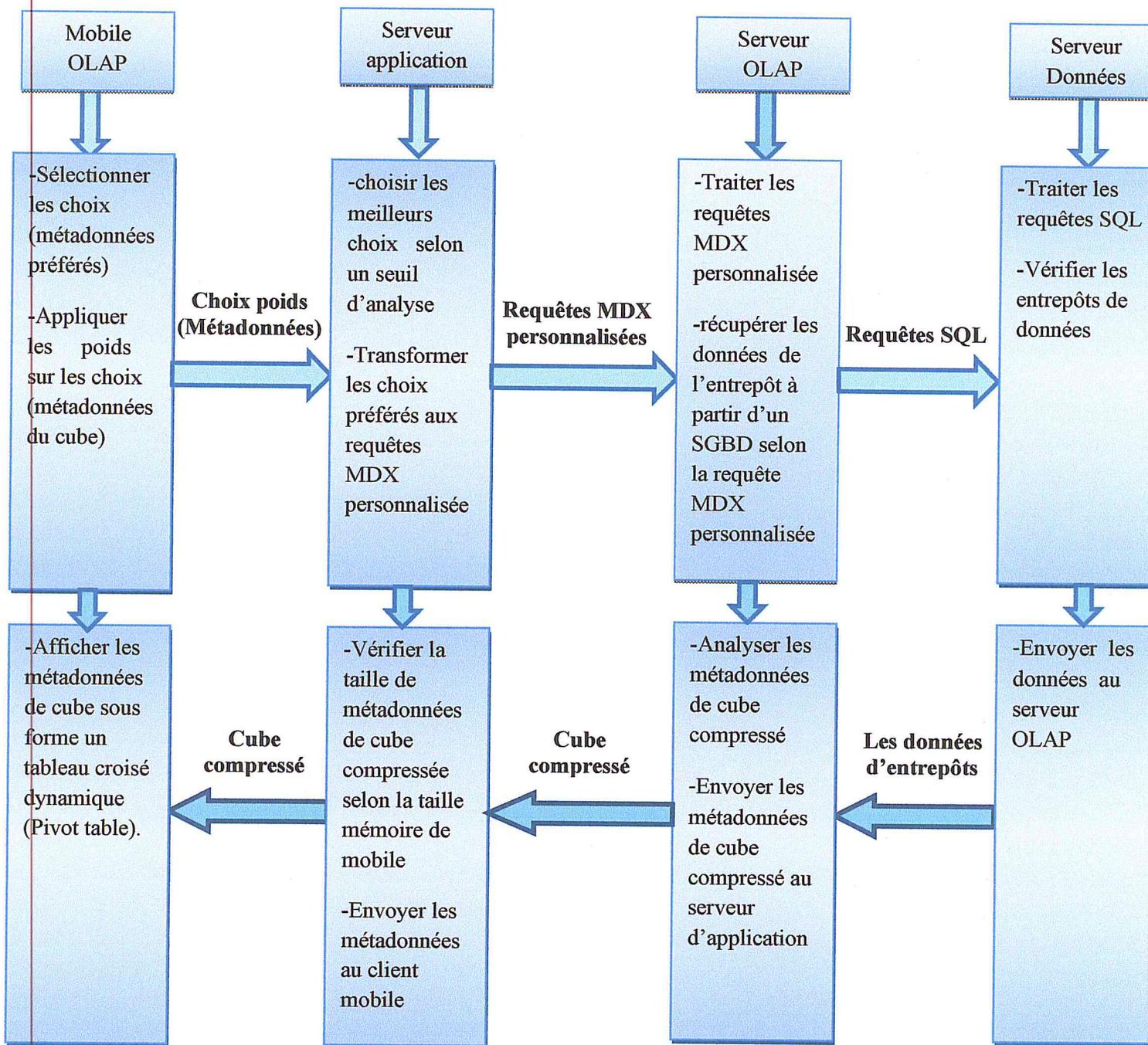


Figure 4.2:Le Schéma de Fonctionnement du Système

A chaque composant de système nous attribuons des tâches spécifiques, (la figure 4.3) définit les tâches principaux associées à chaque composant de système.

CHAPITRE IV : Architecture pour un système OLAP Mobile

Afin d'obtenir un cube compressé nous appliquons la méthode personnalisation des requêtes qui permet de retourner un cube OLAP contenant uniquement les dimensions, les niveaux, et les mesures préférée selon les choix du client mobile (décideur). Le fonctionnement du système peut être réalisé comme suit:

1. Lorsqu'un décideur (analyste) veut consulter leurs données à distance sur un terminal mobile pour prendre des décisions là où il se situe, il établit une connexion avec le serveur d'application, et demande de faire une analyse.
2. Le serveur d'application envoie au décideur la liste des métadonnées du cube qui veut analyser (dimensions, niveaux, et mesures).
3. Le décideur sélectionne les choix préférés et attribue des poids à chaque métadonnée sélectionné (exemple : le décideur choisit les dimensions Produit, Temps, Client, Promotion, il donne respectivement les valeurs : 1, 2 4 ,3), puis envoie ces choix avec les poids appliqués au serveur d'application, avec la contrainte de stockage (taille de stockage réservé pour stocker le cube OLAP sur l'environnement Mobile),
4. Un programme identifie les choix préférés parmi celles choisies par le décideur selon un critère d'analyse qui est le seuil, ce dernier est choisi selon le nombre d'objets de cube (dimensions, niveaux, mesures), en général, le seuil est calculé par choisir **un tiers le nombre de métadonnées choisies**. par exemple si un utilisateur choisit dix dimensions, alors la réponse est égal à la partie entière de $10 * 1/3 = 3$; le système retourne trois dimensions du plus pertinent au moins pertinents selon les valeurs de leurs poids.

Ces choix seront ensuite reformulés sous forme d'une requête MDX personnalisée.

5. La requête MDX est envoyé au serveur OLAP qui se trouve à distance, afin d'extraire les données sélectionnées (de l'entrepôt de données) stockées dans le serveur de données.
6. Le nouveau cube (compressé) est envoyé vers le serveur d'application, ce dernier vérifie la taille des données reçues par rapport à la taille de stockage réservé pour le cube sur le mobile. Si la taille du cube compressé est supérieure à la taille mémoire du mobile alors il sera divisé en partitions, et la

partition qui sera envoyé au client mobile possède les données (dimensions et membres) de la plus pertinente au moins pertinente selon les valeurs du poids des métadonnées préférés.

7. Enfin, le cube compressé est représenté dans un tableau multidimensionnel croisé.

4. La Communication dans le Système :

4.1 Communication Mobile OLAP - Serveur d'application :

Chaque décideur possède un dispositif mobile pour communiquer avec le serveur à distance et analyser leurs données, le premier serveur qui réagit avec le dispositif mobile c'est le serveur d'application, pour cela nous utilisons la méthode de communication **web service** pour connecter le mobile OLAP avec le serveur d'application. Il offre l'interface d'appel des méthodes du processus de la personnalisation (sélectionner domaine d'analyse, sélectionner métadonnées préférés, etc...) qui s'exécute à distance sur le serveur d'application pour un client mobile.

Notre architecture est basé sur l'architecture orientée service car elle peut être enrichie avec d'autres services supplémentaires (par exemple service authentification).

4.1.1 Architecture orientée services(SOA) :

Une architecture orientée services est une architecture logicielle s'appuyant sur un ensemble de services.

Les éléments fondamentaux de SOA sont [1] :

- Le service
- La description du service: (Interface)
- La publication: (annuaire de service)
- La découverte: (recherche)
- L'invocation

On distingue trois rôles distincts dans une architecture orientée services [1] :

- **Le fournisseur de service:** crée le service Web, puis publie son interface ainsi que les informations d'accès au service dans un annuaire de services Web.
- **L'annuaire de service :** rend disponible l'interface du service ainsi que ses informations d'accès, pour n'importe quel demandeur potentiel de service
- **Le consommateur de service :** accède à l'annuaire de service pour effectuer une recherche afin de trouver les services désirés. Ensuite il se lie au fournisseur pour invoquer le service.

4.1.2 Avantages d'une SOA sur les plateformes mobiles :

Les architectures orientées services ainsi que leur réalisation sous forme de services Web ont comme avantage de [1] :

- favoriser la réutilisation de composants logiciels (services).
- faciliter l'intégration de systèmes hétérogènes alors :
 - toute plateforme mobile, peu importe le matériel ou le système d'exploitation, a la capacité d'interagir de manière uniforme avec les services Web.
 - leur support des communications asynchrones les rendent également bien adaptés aux réseaux sans-fil qui ne peuvent garantir des connexions persistantes : chaque requête étant indépendante l'une de l'autre, le client peut contacter le service au moment opportun, lorsque la connectivité réseau est disponible.
- Possibilité de la répartition des rôles :
 - il est possible de déléguer à un service les traitements plus complexes (consommateurs de temps processeur), au lieu de les réaliser à même l'appareil mobile.
 - un puissant serveur qui héberge le service arrivera à exécuter ce traitement en moins de temps que l'appareil mobile, qui comporte un processeur plus lent et une mémoire vive limitée.

4.1.3 Présentation web service :

Un **service Web** est un composant logiciel identifié par une **URL**, dont les interfaces publiques sont définies et appelées en exploitant **XML**, plus particulièrement un service web est tout service qui peut être [62]:

- disponible sur Internet
- découvert par d'autres systèmes logiciels
- Utiliser un système de message basé sur **XML**
- N'est pas lié à **aucun système d'exploitation** particulier
- N'est pas lié à **aucun langage de programmation**.

5.1.4 Exploitation du service web dans notre Système :

Les Services Web constituent une infrastructure technologique distribuée faite pour assembler des systèmes hétérogènes en une seule et même fonction à travers un réseau.

On a choisi l'exploitation du service web car il présente les avantages suivants [1] :

- Les données peuvent être présentées uniquement sur le serveur distant (par exemple un catalogue produit, un classement en temps réel, etc...) ;
- Le serveur distant peut disposer d'une puissance de calcul ou de capacité de stockage dont l'utilisateur local ne dispose pas ;
- L'application distante peut être utilisée simultanément par un grand nombre d'utilisateurs et sa mise à jour intervient qu'à un seul endroit.
- Les services web représentent un mécanisme de communication entre applications distantes à travers le réseau internet indépendant de tout langage de programmation et de toute plate-forme d'exécution.
- Les services web utilisent le protocole HTTP comme moyen de transport. Ainsi, les communications s'effectuent sur un support universel, maîtrisé et généralement non filtré par les pare feux.

CHAPITRE IV : Architecture pour un système OLAP Mobile

Dans notre cas, nous proposons trois services web principaux :

➤ Service authentification :

Il permet de l'utilisateur qui accède au système de s'authentifier d'abord pour introduire son nom d'utilisateur et son mot de passe, il possède une seule méthode qui est :

- ❖ Opération *authentifier* (nom utilisateur, mot de passe) : cette opération permet de gérer les droits d'accès.

➤ Service consultation métadonnées :

Il offre à l'utilisateur les listes des métadonnées qui composent le cube choisit, pour qu'il puisse sélectionner les métadonnées qu'il veut visualiser selon ses choix. il possède trois opérations :

- ❖ Opération *existe cube* (identifiantCube) : elle vérifie l'existence du cube choisit. Si le cube n'existe pas, un message d'erreur s'affichera.
- ❖ Opération *liste_Dimensions* (identifiantCube,) : elle retourne la liste des dimensions qui existent dans le cube que veut analyser ainsi que nombres de niveaux pour chaque dimension.
- ❖ Opération *liste_Niveaux* (identifiantCube, nomDimension) : elle retourne la liste des niveaux par dimension dans le cube que veut analyser.
- ❖ Opération *liste_unités_mesurés*(identifiantCube) :elle retourne la liste des unités mesurés qui existent dans le cube que veut analyser.

➤ Service Personnalisation :

Il offre une interface d'appel des méthodes du processus de la personnalisation qui s'exécutent à distance sur le serveur d'application pour un client Mobile. Le service Web que nous avons proposé peut effectuer l'opération suivante :

- Opération *Récupérer_cube_compressé* (liste Dimensions Préférés, liste Niveaux Préférés, liste Unités Mesurés Préférés, liste Poids Dim, liste Poids Niv, liste Poids Unités Mesurés) :

Cette opération est le noyau de service du Personnalisation, elle permet de récupérer le cube compressé. Elle s'appuie premièrement par *Traiter métadonnées préférés()*, elle prend comme entrée la *liste_métadonnées_préférés* reçues depuis le client Mobile et *Vérifier seuil d'analyse()* (le calcul du seuil est varié selon le nombre de métadonnées qui doit être affiché à l'utilisateur) ensuite elle appelle l'opération *ordonner poids()* et *générer la requête MDX personnalisé()* correspondante ensuite fait appelle à *Récupérer données serveur OLAP()* qui permet d'établir une connexion avec le serveur OLAP via le service XMLA afin d'obtenir les métadonnées voulus (les membres et les mesures), et enfin l'opération *Envoyer données client mobile ()* qui permet d'envoyer le cube compressé au client mobile après vérification de la taille de stockage du cube sur le mobile.

L'algorithme « **Pseudo Algorithme pour le Serveur d'application : paragraphe 6.3** » s'exécute dans cette opération

4.2 Communication Serveur d'application - Serveur OLAP :

Afin d'interroger les cubes situées sur le serveur OLAP distant, ce dernier doit se connecter avec le serveur d'application via le service XMLA, ce qui nous présentons par la suite :

4.2.1 Présentation :

XMLA est un service basée sur XML ou protocole de communication utilisé dans le domaine du traitement analytique en ligne (OLAP) et Business Intelligence (BI). Ce service a été créé dans le but de standardiser la communication d'une application client et un fournisseur de données analytiques via une connexion Internet [63]. La communication entre la source de la clientèle et des données s'effectue à l'aide de messages XML à travers le Simple Object Access Protocol standard (SOAP). Pour interroger les bases de données multidimensionnelles, XMLA utilise MDX, un langage commun d'interrogation multidimensionnelle. Pour plus de détails, (reportez-vous à la requête MDX chapitre Réalisation).

5.2.2 Exploitation de Service XMLA dans notre Système :

Le service XMLA possède les avantages suivants [1] :

- XMLA permet aux applications clientes de communiquer avec les serveurs OLAP ou des sources multidimensionnelles, qui sont basés sur Internet
- la résolution des problèmes de dépendance client-serveur.
- les utilisateurs peuvent déployer des applications à partir de n'importe quel endroit, et peut les partager au sein des organisations ou entre les entreprises, à travers le Web.
- XMLA supporte les protocoles Web standard (XML, HTTP ou SOAP) et de diverses plates-formes standard (par Microsoft, SAS, SAP et Hyperion, et récemment Oracle), et ne nécessite pas la connaissance des interfaces spécifiques et des API ou des installations spéciales.
- XMLA minimise le nombre de demandes et d'allers retours client au serveur, permettant ainsi aux sources de données d'améliorer l'évolutivité et la robustesse.

Dans notre système, le service XMLA propose un mécanisme d'interrogation des cubes de données situés sur le serveur OLAP via le protocole HTTP, plus particulièrement il permet de connecter le serveur OLAP distant avec le serveur d'application afin de récupérer les métadonnées à partir des sources de données multidimensionnelles.

5. Les Pseudos Algorithmes pour la compression des cubes de données OLAP :

Notre objectif est l'utilisation de la méthode de personnalisation pour la compression des cubes de données adaptés aux environnements mobiles afin d'essayer de résoudre les problématiques qui existent dans les techniques de compression des cubes OLAP.

Pour parvenir à la mise en œuvre de notre système, nous formalisons le problème de personnalisation traité :

5.1 Formalisation du problème :

Entrées :

Cube non compressé

Sélection objetscube (dimensions, membres, mesures)

Taille mémoire Mobile.

Poids sur les dimensions, niveaux, et mesures.

Programme :

TransformerMétadonnéesPertinentesMDXpersonnalisée() ;

Sortie :

Cube OLAP compressé

5.2 Pseudo Algorithmes pour le client Mobile :

Pour afficher le tableau croisé du cube compressé sur le mobile, il suffit de récupérer les métadonnées (dimensions, niveaux, mesures et membres), attribuer des poids aux métadonnées sélectionnés (dimensions, niveaux, mesures), et les envoyer vers le serveur d'application (métadonnées sélectionnés + poids) avec la taille de stockage sur le mobile réservé pour le cube OLAP compressé reçu.

Entrées :

NomCube, listeDimension, listeNiveaux, listeMesure

Programme :

Connecter serveur () ;
Récupérer métadonnées (NomCube) ;
Sélectionner dimensions préférées (ListeDimension);
Ajouter poids dimensions (poids) ;
Sélectionner niveaux préférés (ListeNiveaux) ;
Ajouter poids niveaux (poids) ;
Sélectionner mesures préférés(ListeMesure) ;
Ajouter poids mesures (poids) ;
Envoyer taille mémoire mobile () ;
Envoyer métadonnées préférés serveur () ;
Envoyer poids métadonnées préférés () ;
Calculer taille de stockage mobile () ;
Envoyer taille de stockage mobile () ;

Sortie :

Métadonnées préférés
Poids
Tableau croisé du cube compressé
T_stock_cube_mobile

5.3 Pseudo Algorithme pour le Serveur d'application :

Lors de l'arrivée des métadonnées préférés avec leurs poids au serveur, un programme commence à traiter ces derniers en les comparant par rapport à un seuil d'analyse (selon poids et nombre dimensions), une requête MDX est formulée et envoyée vers le serveur OLAP afin d'extraire toutes les données pertinentes dont les valeurs poids de leurs métadonnées vérifient la condition du seuil d'analyse.

Après la réception des données pertinentes, il calcule la taille du résultat pertinent puis vérifie la taille du stockage réservé sur mobile par rapport à cette dernière (nombre d'enregistrements) obtenu :

- Si la taille du résultat est supérieure à la taille du mobile, alors il sera divisé en partitions et celle la plus convenable sera envoyée vers le client mobile.
- Sinon il sera envoyé directement.

Ce programme est exprimé par le pseudo algorithme suivant :

Entrée :

T_stock_cube_mobile // taille de stockage de cube sur le mobile
Métadonnées préférés
Métadonnées _serveur OLAP

Programme :

Traiter métadonnées préférés (Métadonnées, poids) ;
Vérifier seuil d'analyse (poids, nbrDimensions) ;
Ordonner poids (poidsDimension, poidsNiveaux, poidsMesures) ;
Transférer requête MDX personnalisé (Métadonnées préférés) ;
Récupérer données serveur OLAP (requêteMDXpersonnalisée) ;
Calculer taille données reçus () ;
Vérifier taille stockage (T_stock_cube_mobile, taille donnée reçus) ;
Envoyer données client mobile (Métadonnées _serveur OLAP) ;

Sortie :

Données du cube compressé

6. Conclusion :

Dans ce chapitre, nous avons vu une description sur notre proposition, nous avons présenté l'architecture de notre système, le service web responsable de la compression ainsi nous avons proposé les deux algorithmes client mobile et serveur d'application pour la sélection des métadonnées dont le but est la compression des cubes OLAP avec personnalisation.

CHAPITRE V :

Réalisation

1. Introduction :

Au cours du chapitre précédent, nous avons conçu et détaillé notre système ainsi que sa logique de fonctionnement. Nous avons décrit le module de la personnalisation des requêtes.

Afin de concrétiser ce système, nous passons à la dernière partie du développement où nous allons présenter la réalisation de notre système.

2. Présentation de l'entrepôt utilisé :

Les données à manipuler sont celles de « *FoodMart* », un entrepôt de données de Microsoft. Cet entrepôt contient **07 cubes de données** : **Warehouse, Store, HR, Sales, Sales Ragged, Sales 2, Warehouse and Sales.**

Dans nos expérimentations, nous avons exploité le cube « *Sales* ». Les deux tableaux suivants (5.1 et 5.2) résument les dimensions, les niveaux, et les mesures du cube « *Sales* » :

CHAPITRE V : Réalisation

Nom dimension	niveau	description
Customers	Country, State or Province, City, Name	Geographical hierarchy for registered customers of our stores
Education Level	Education Level	Education level of customer, such as "Graduate Degree" or "High School Degree."
Gender	Gender	Customer gender: "M" or "F"
Marital Status	Marital Status	Customer marital status: "S" or "M"
Product	Product Family Product Department Product Category Product Subcategory Brand Name Product Name	The products that are on sale in the FoodMart stores.
Promotion Media	Media Type	The media used for a promotion, such as Daily Paper, Radio, or Television.
Promotions	Promotion Name	Identifies promotion that triggered the sale.
Store	Store Country Store State Store City Store Name	Geographical hierarchy for different stores in the chain (country, state,city).
Store Size in SQFT	Store Square Feet	Area occupied by store, in square feet.
Store Type	Store Type	Type of store, such as "Deluxe Supermarket" or "Small Grocery."
Time	Years, Quarters, Months	Time period when the sale was made.
Yearly Income	Yearly Income	Income of customer.

Tableau 5.1 : Les dimensions et les niveaux de cube Sales [8]

Nom mesure	Description
Unit Sales	Number of units sold
Store Cost	Cost of goods sold.
Store Sales	Value of sales transactions
Store Sales	Number of sales transactions
Store Sales Net	Value of sales transactions less cost of goods sold
Sales Average	Store sales/sales count. (This is a calculated measure.)

Tableau 5.2: les unités mesurées de cube Sales [8]

CHAPITRE V : Réalisation

Les préférences (les choix) de l'utilisateur sont exprimées sur 5 dimensions chaque dimension contient au moins un seul niveau.

La figure suivante (figure 5.1) montre le schéma en étoile de cube « Sales », cette figure détermine une description générale sur les dimensions, leurs niveaux ainsi que les unités mesurés.

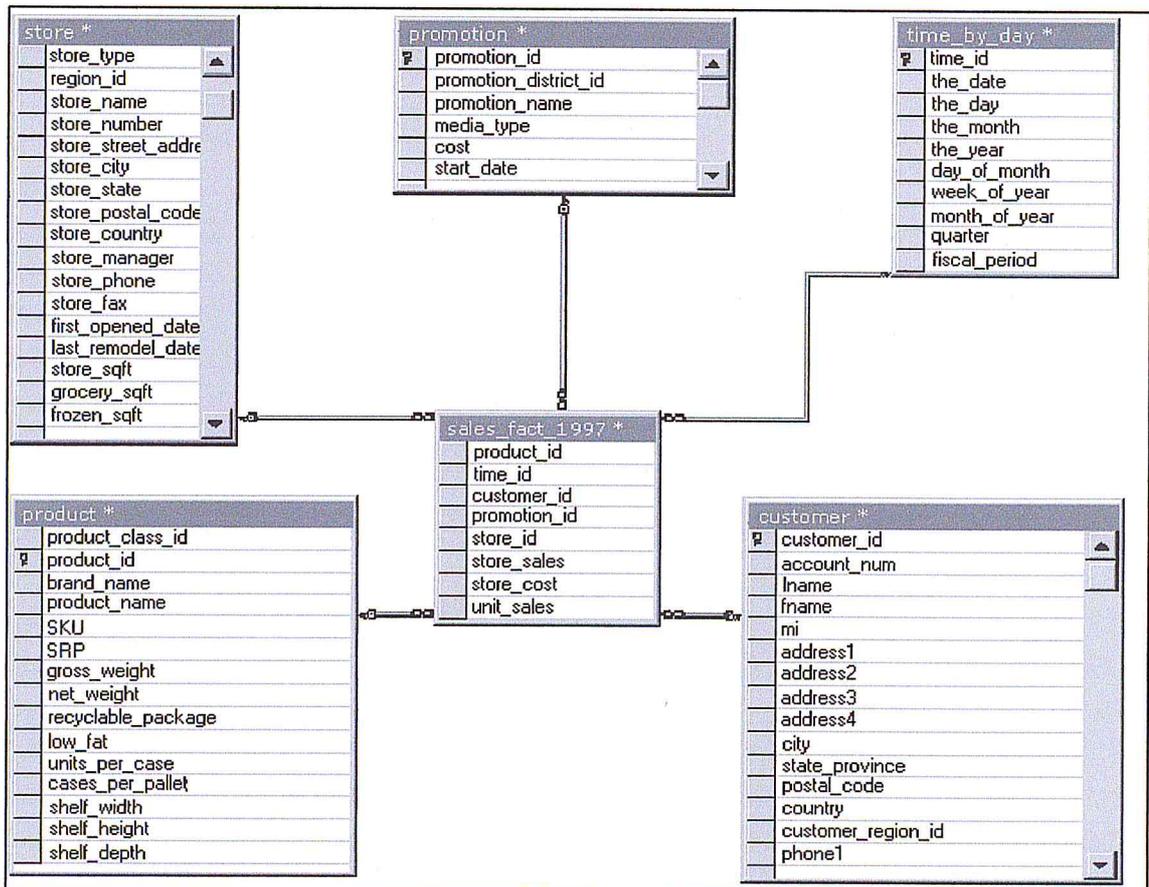


Figure 5.1 : Schéma en Etoile du cube Sales [64].

3. Architecture technique du système :

Notre système est implémenté sur une architecture technique (figure 5.2) de type trois – tiers, où nous utilisons Le serveur de données Microsoft office Access, le serveur d'application possède deux serveurs HTTP (Apache), un pour déployer le serveur OLAP Mondrian, et l'autre pour déployer les méthodes du service Personnalisation ; ces méthodes sont appelés et exécutés à distance au niveau du serveur d'application et les résultats seront affichés au niveau du client Mobile.

Les algorithmes que nous avons proposés dans le chapitre précédent sont testés au niveau du serveur d'application. Ces algorithmes utilisent le service XMLA pour la connexion avec le serveur OLAP.

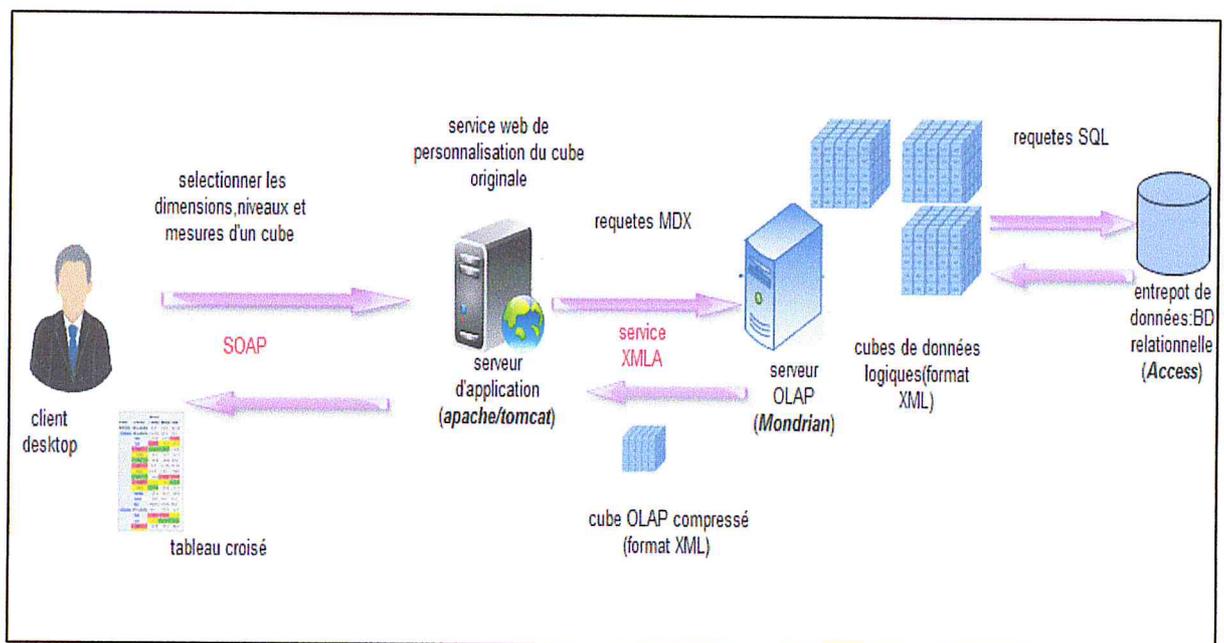


Figure 5.2 : Architecture technique du système

4. Les environnements de développement :

4.1 Matériel utilisé :

Nous avons développé notre application sur des machines dotées du système d'exploitation Windows 7(32 bit) et ayant des processeurs Intel Celeron® Dual Core. La fréquence d'horloge du CPU est de 2.3 GHz et la mémoire installée (RAM) est de 2.00 Go.

4.2 Les langages utilisés :

4.2.1 Langage JAVA :

Java est un langage de programmation informatique orienté objet créé par James Gosling Patrick Naughton de Sun Microsystems. (Aujourd'hui Oracle). Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Nous utilisons, ce langage pour le développement de l'application OLAP [65].

Le choix approprié :

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

4.2.2 Langage MDX :

MDX est un langage de requêtes pour les bases de données multidimensionnelles, de la même manière que SQL est utilisé pour les requêtes sur les bases de données relationnelles. Dans son approche, MDX est proche du SQL sur son aspect select et where même si la similarité ne va pas plus loin. Le but des expressions multidimensionnelles MDX est de rendre aisé et intuitif l'accès aux données de différentes dimensions [66].

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

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

4.2.2.1 La Clause Cross Join :

La fonction Crossjoin retourne le produit croisé de deux ou plusieurs ensembles spécifiés. L'ordre des tuples dans le jeu en résulte dépend de l'ordre des ensembles à assembler et l'ordre de leurs membres. Par exemple, lorsque le premier ensemble est constitué de $\{x_1, x_2, \dots, x_n\}$, et le second ensemble se compose de $\{y_1, y_2, \dots, y_n\}$, le produit vectoriel de ces ensembles est [67] :

$$\{(x_1, y_1), (x_1, y_2), \dots, (x_1, y_n), (x_2, y_1), (x_2, y_2), \dots, (x_2, y_n), \dots, (x_n, y_1), (x_n, y_2), \dots, (x_n, y_n)\}$$

Exmple :

```
select {[Measures].[Unit Sales], [Measures].[Store Cost],  
[Measures].[Store Sales]} ON COLUMNS,  
Crossjoin(Hierarchize(Union({[Promotion Media].[All Media]},  
[Promotion Media].[All Media].Children)), {[Product].[All Products]})  
ON ROWS  
from [Sales]  
where [Time].[1997]
```

4.3 Les outils utilisés :

4.3.1 Eclipse IDE :

Eclipse IDE est un environnement de développement intégré libre, le terme Eclipse désigne également le projet correspondant, lancé par IBM. Il est extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java à l'aide de la bibliothèque graphique SWT, de IBM . Pour notre application nous utilisons eclipse-JEE Kepler (Eclipse IDE for Java EE Développeurs) qui contient tous les outils indispensable pour créer des applications Java et Java Enterprise Edition (Java EE) .

4.3.2 Serveur OLAP Mondrian :

Mondrian est un moteur OLAP (Online Analytical Processing) écrit en Java par Julian Hyde qui permet la conception, la publication et le requêtage de cubes multidimensionnels. Il permet l'exécution de requêtes en langage MDX sur des entrepôts de données s'appuyant sur des SGBDR, 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 compréhensible par une API de présentation côté client, le plus souvent en mode Web, avec par exemple **JPivot**, **Pentaho Analyzer**, **Pentaho Analysis Tool**, **GeoAnalysis Tool (G.A.T)** .

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 [68]**.

4.3.3 Serveur Web Apache Tomcat 7.0 :

Tomcat est un serveur d'application Java permettant d'exécuter des servlets et des pages serveur Java (JSP). Il est développé sous licence open-source par la fondation Apache. Il peut être utilisé ou couplé avec un serveur Web (dont Apache), et porté sur n'importe quel système sur lequel une machine virtuelle Java est installée [69].

Pour notre cas nous avons utilisé le serveur Apache Tomcat pour déployer le serveur Mondrian.

4.3.3 Le serveur de données Access :

De nombreuses applications OLAP font usage d'un entrepôt de données décisionnelles pour le stockage des cubes. Ceux-ci sont généralement basés sur des systèmes de gestion de bases de données relationnels tel que : ACCESS, MySQL, etc... ; Les données des cubes sont organisées selon un modèle multidimensionnel (schémas en étoile, en flocon ou parent- enfant), dont les principes diffèrent des modèles relationnels normalisés des systèmes

Transactionnels Le peuplement et la mise à jour des entrepôts de données décisionnelles sont faits depuis diverses sources de données opérationnelles. Des outils nommés ETL (*Extract, Transform, Load*), qui permettant l'extraction, la transformation et le chargement de données.

4.4 Les API utilisées :

4.4.1 OLAP for java (Olap4j) :

Olap4j a été conçu pour être agnostique plate-forme Java, de sorte que vous avez la liberté de choisir la plate-forme que vous préférez. La principale distribution d'Olap4j est compatible avec Java versions 5 et 6.

Olap4j est une API OLAP, il est également une spécification pour l'implémentation des bases de données d'OLAP [70], [71].

La figure suivante (figure 5.3) représente le schéma de l'API de java pour OLAP (OLAP4j) :

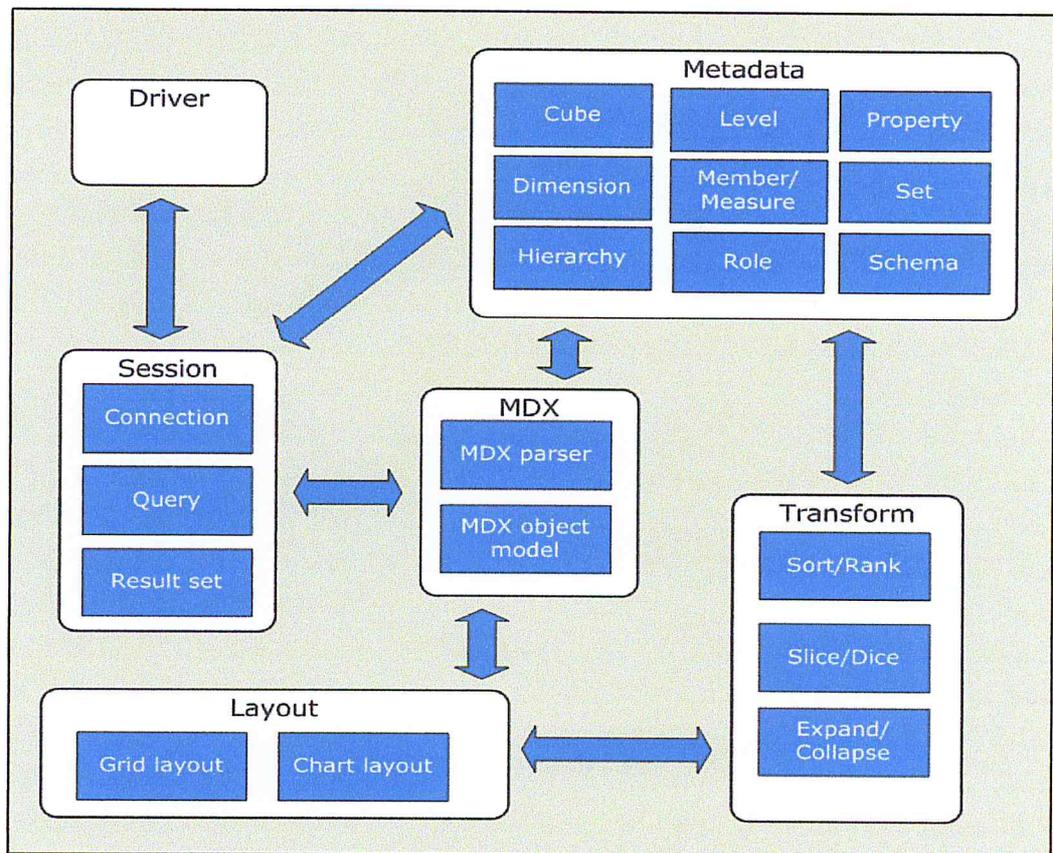


Figure 5.3 : Schéma API de JAVA pour OLAP [71].

4.4.2 Le pilote XMLA Dans OLAP for JAVA 1.0 :

La communication avec un serveur distant OLAP sont effectuées via le protocole de transport HTTP. Il est très simple, des moyens bien connus sont utilisés pour la communication entre les clients et les serveurs. La spécification XMLA exploite la puissance de HTTP, il suffise de dire que c'est le moyen utilisé par le pilote XMLA. XMLA utilise également l'authentification HTTP comme moyen de sécurité. Afin d'établir une connexion à un serveur distant qui n'est pas instancié par l'objet de connexion lui-même, olap4j contient un driver d'implémentation intégré XMLA [72].

La figure suivante (figure 5.3) représente le pilote XMLA.

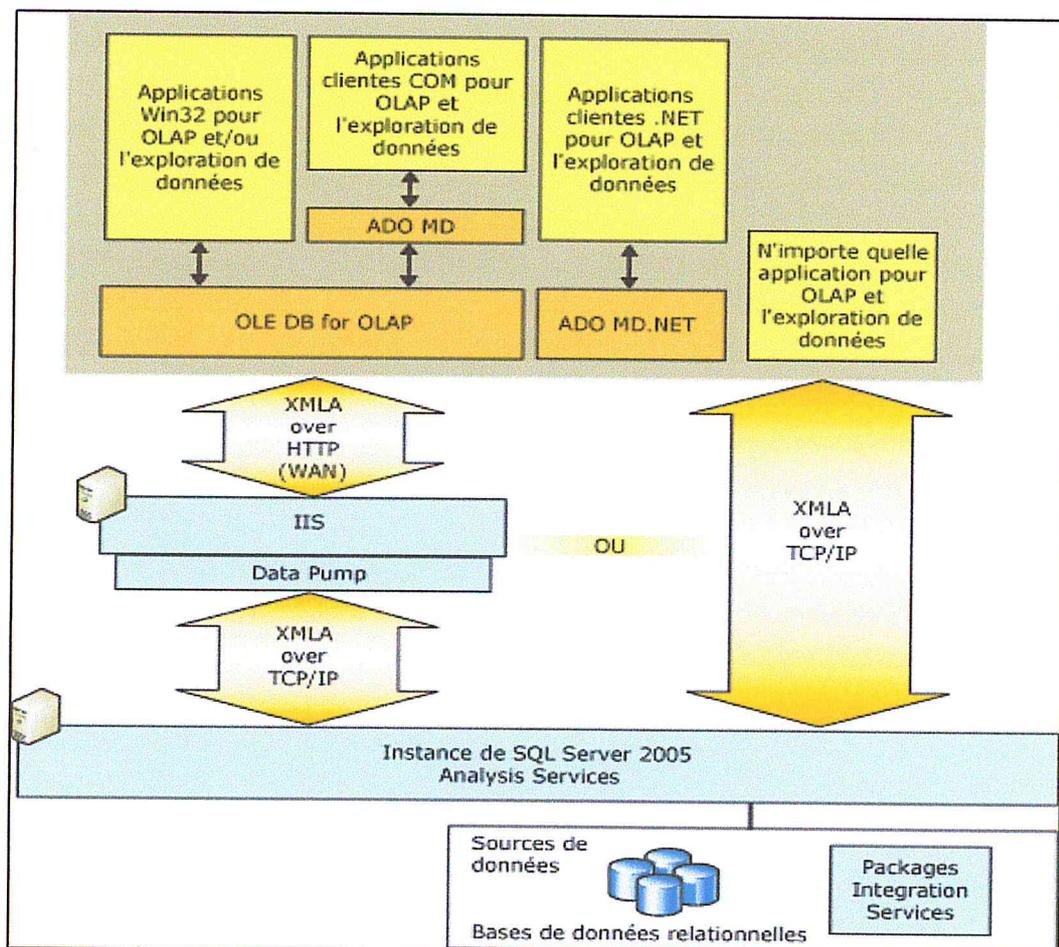


Figure 5.4 : Le Pilote XMLA [72].

5. Les outils du Web Services :

5.1 Apache CXF 2.7 :

Apache CXF est un cadre de services open source. CXF vous aide à construire et développer des services utilisant des API de programmation d'interface, comme JAX-WS et JAX-RS. Ces services peuvent parler une variété de protocoles tels que SOAP, XML / HTTP, REST HTTP, ou CORBA et le travail sur une variété de transports tels que HTTP, JMS ou JBI. Il comprend également un «simple interface» qui permet de créer des clients et des terminaux sans annotations. CXF supporte à la fois premier contrat de développement avec WSDL et le code de premier développement à partir de Java [73].

5.1.1 JAX-WS :

API Java pour les services Web XML (JAX-WS) fait partie d'un ensemble de technologies Java utilisées pour développer des services Web. JAX-WS est également utilisé pour créer des services Web et les clients correspondants qui communiquent en utilisant XML basé sur le protocole SOAP pour envoyer des messages ou utiliser appels de procédure distante pour échanger des données entre le client et le fournisseur de service [74].

5.1.2 Apache Axis 1.6:

Axis est un Framework créé par Apache Software Foundation , qui vise à faciliter le développement de services Web en technologie SOAP.

Axis offre notamment :

- Une API pour développer des services web SOAP RPC ou à base de messages SOAP.
- La sérialisation/désérialisation automatique d'objets Java dans des messages SOAP.
- Le support de différentes couches de transport : HTTP, FTP, SMTP, POP et IMAP.
- Des outils pour déployer, tester et monitorer des web-services.

5.1.3 Protocole SOAP :

SOAP est un protocole d'invocation de méthodes sur des services distants. Basé sur XML, SOAP a pour principal objectif d'assurer la communication entre machines. Le protocole permet d'appeler une méthode RPC et d'envoyer des messages aux machines distantes via http (figure 5.5). Ce protocole est très bien adapté à l'utilisation des services Web, car il permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers [75].

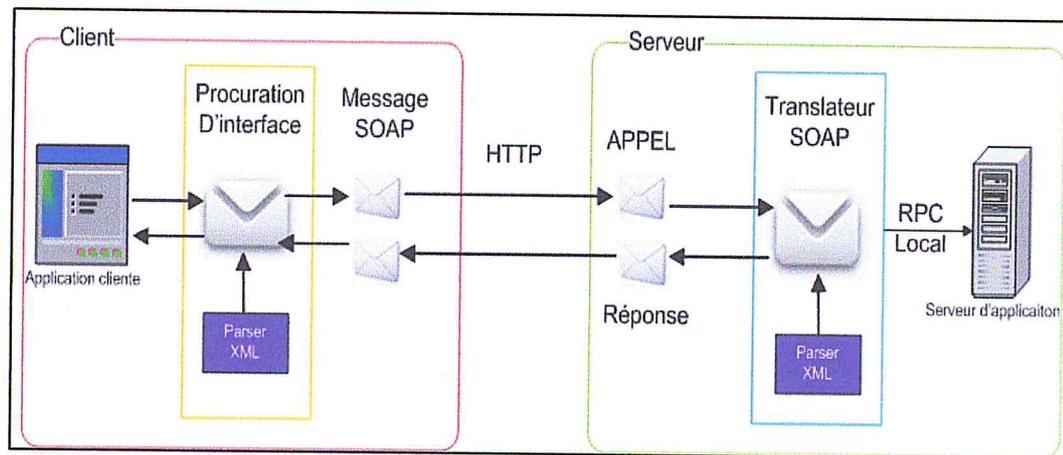


Figure 5.5 : Requêtes et réponses HTTP [75].

SOAP est bien plus populaire et utilisé que XML-RPC. C'est une recommandation du W3C. D'après cette recommandation, SOAP est destiné à être un protocole léger dont le but est d'échanger des informations structurées dans un environnement décentralisé et distribué. Une des volontés du W3C vis-à-vis de SOAP est de ne pas réinventer une nouvelle technologie. SOAP a été construit pour pouvoir être aisément porté sur toutes les plates-formes et les technologies existantes.

6. Application cube OLAP compressé via les web services :

Nous représenterons les interfaces de notre application afin de donner une vue générale sur les fonctionnalités de cette dernière.

6.1 Le scénario de L'application :

Comme nous avons mentionné précédemment, nous exploitons le cube « Sales » Lorsqu'un utilisateur lance l'application, une fenêtre sera affichée (Figure 5.6), il faut entrer l'identifiant du cube à analyser puis cliquer sur le bouton (*Analyser*).si le cube n'existe pas sur le serveur OLAP, un message « ce cube n'existe pas » s'affichera. Cette tâche est l'exécution de l'opération exist_cube() du service web consultation métadonnées

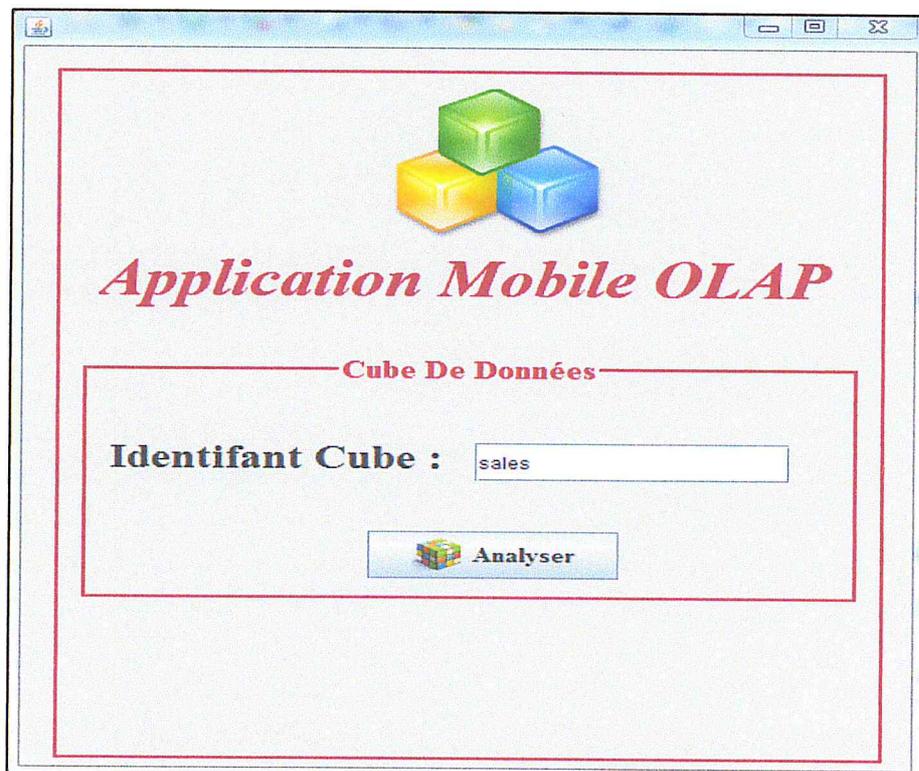


Figure 5.6 : Entrer le domaine d'analyse

CHAPITRE V : Réalisation

L'utilisateur peut visualiser la liste des métadonnées qui compose le cube choisit le système affiche à l'utilisateur (analyste) la liste des dimensions (**figure 5.8**), la liste des niveaux (**figure 5.9**) et la liste des mesures (**figure 5.10**).

Ces tâches sont l'exécution successive des opérations *liste_Dimensions()*, *liste_Niveaux()*, *liste_unitésMesurés()* du service web consultation métadonnées que nous avons mentionné dans le chapitre précédent.

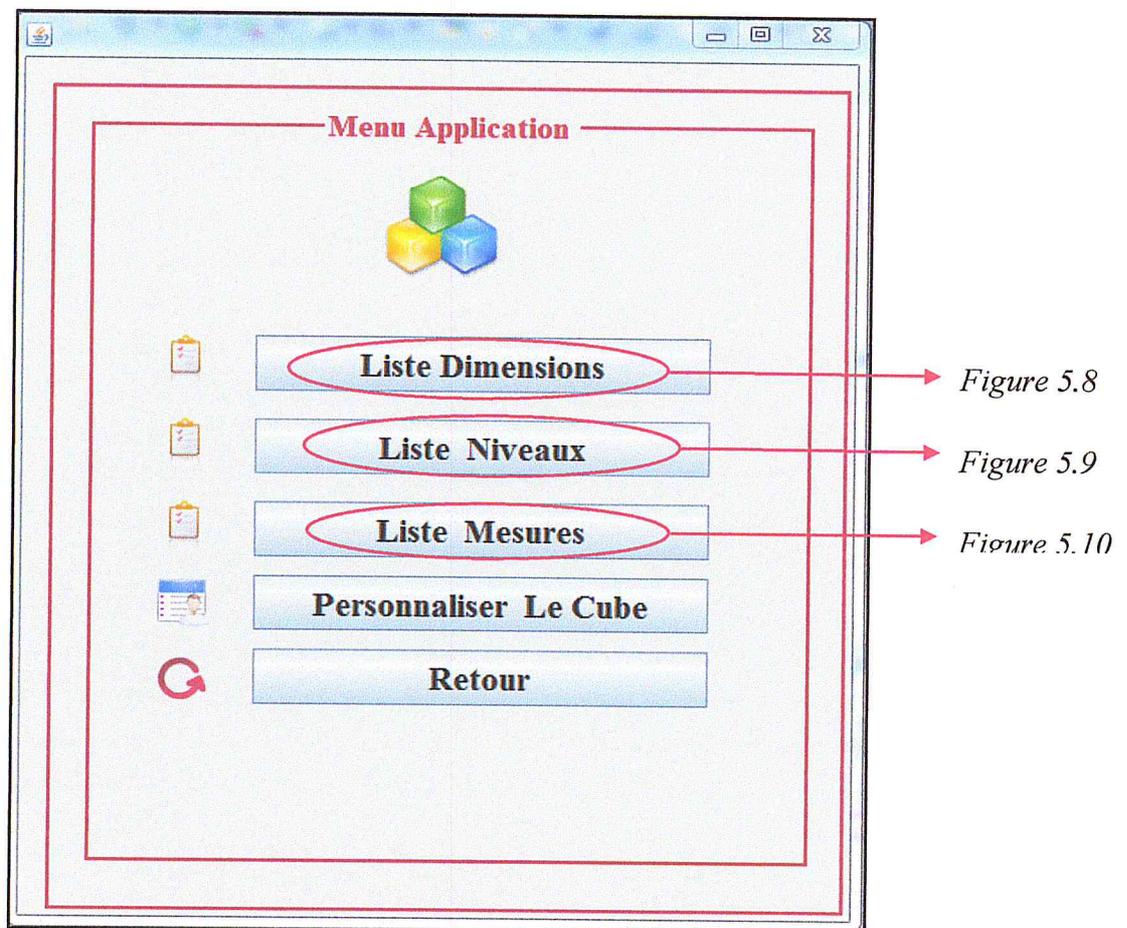
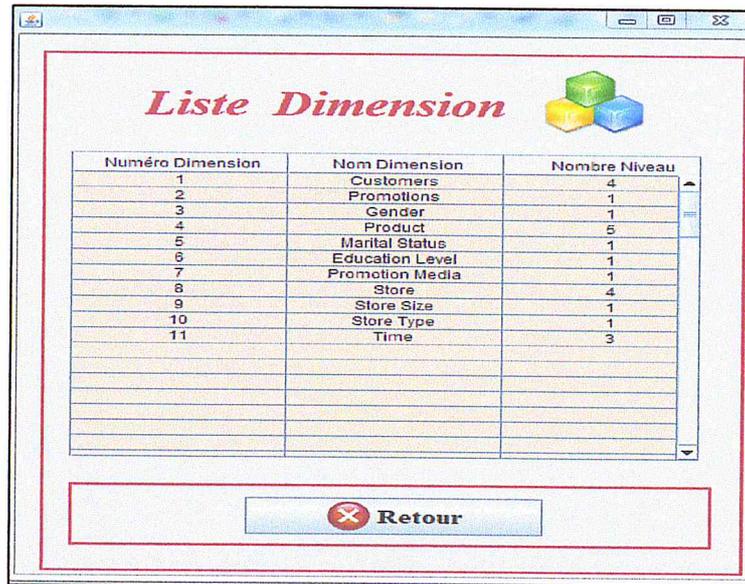


Figure 5.7 : Le menu d'application du service Personnalisation

CHAPITRE V : Réalisation

- ❖ La liste des dimensions disponibles : on obtient ce tableau en cliquant sur Liste Dimensions de la figure 5.7

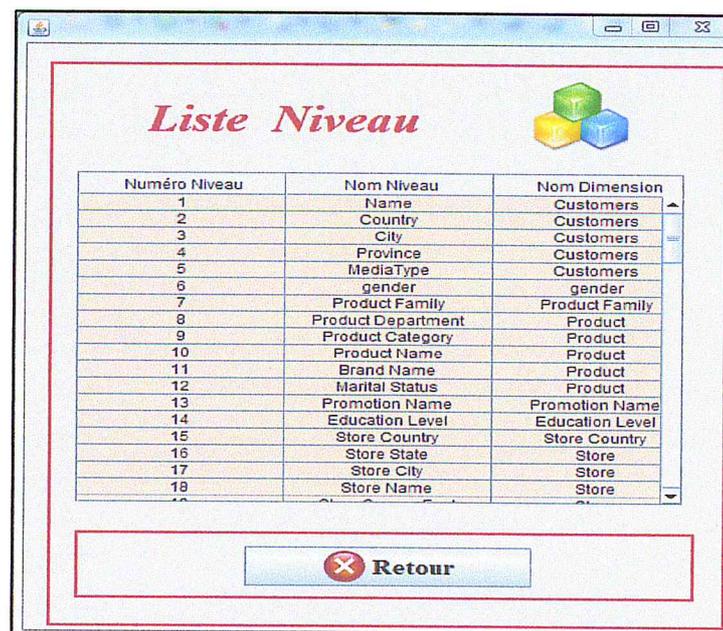


The screenshot shows a window titled "Liste Dimension" with a 3D cube icon. It contains a table with three columns: "Numéro Dimension", "Nom Dimension", and "Nombre Niveau". Below the table is a "Retour" button.

Numéro Dimension	Nom Dimension	Nombre Niveau
1	Customers	4
2	Promotions	1
3	Gender	1
4	Product	5
5	Marital Status	1
6	Education Level	1
7	Promotion Media	1
8	Store	4
9	Store Size	1
10	Store Type	1
11	Time	3

Figure 5.8 : La liste des Dimensions

- ❖ La liste des niveaux disponibles: on obtient ce tableau en cliquant sur Liste Niveaux de la figure 5.7 .



The screenshot shows a window titled "Liste Niveau" with a 3D cube icon. It contains a table with three columns: "Numéro Niveau", "Nom Niveau", and "Nom Dimension". Below the table is a "Retour" button.

Numéro Niveau	Nom Niveau	Nom Dimension
1	Name	Customers
2	Country	Customers
3	City	Customers
4	Province	Customers
5	MediaType	Customers
6	gender	gender
7	Product Family	Product Family
8	Product Department	Product
9	Product Category	Product
10	Product Name	Product
11	Brand Name	Product
12	Marital Status	Product
13	Promotion Name	Promotion Name
14	Education Level	Education Level
15	Store Country	Store Country
16	Store State	Store
17	Store City	Store
18	Store Name	Store

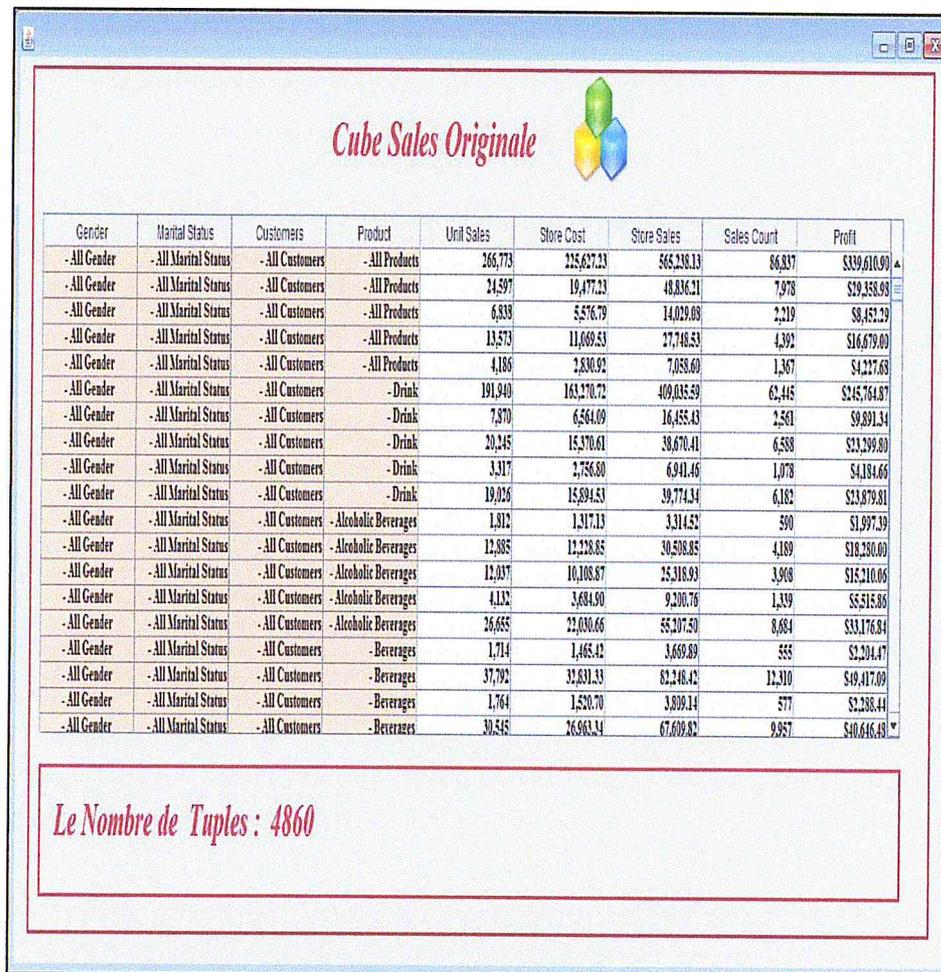
Figure 5.9 : La liste des niveaux

CHAPITRE V : Réalisation

Afin de montrer la compression des données avec la méthode de personnalisation des requêtes nous choisissons un exemple de la sélection des données et poids appliquées à ces sélections les figures (5.12, 5.13, 5.14, 5.15) déterminent l'exemple de test qui sera présenté par la suite , mais avant la sélection des choix de l'utilisateur nous présentons le cube de données originale non compressé (Figure 5.11) ce cube comporte tous les métadonnées ,en remarquant que le nombre de tuples est énorme .

❖ Cube de données non compressé :

Cube Sales Originale



Gender	Marital Status	Customers	Product	Unit Sales	Store Cost	Store Sales	Sales Count	Profit
- All Gender	- All Marital Status	- All Customers	- All Products	266,773	225,627.23	565,228.13	86,837	\$39,610.90
- All Gender	- All Marital Status	- All Customers	- All Products	24,597	19,477.23	48,836.21	7,978	\$29,358.98
- All Gender	- All Marital Status	- All Customers	- All Products	6,838	5,576.79	14,029.68	2,219	\$8,452.29
- All Gender	- All Marital Status	- All Customers	- All Products	13,573	11,669.53	27,748.53	4,392	\$16,679.00
- All Gender	- All Marital Status	- All Customers	- All Products	4,186	2,830.92	7,058.60	1,367	\$4,227.68
- All Gender	- All Marital Status	- All Customers	- Drink	191,940	163,270.72	409,035.59	62,445	\$245,764.87
- All Gender	- All Marital Status	- All Customers	- Drink	7,370	6,564.09	16,455.43	2,561	\$9,891.34
- All Gender	- All Marital Status	- All Customers	- Drink	20,245	15,370.61	38,670.41	6,588	\$23,299.80
- All Gender	- All Marital Status	- All Customers	- Drink	3,317	2,756.50	6,941.46	1,078	\$4,184.66
- All Gender	- All Marital Status	- All Customers	- Drink	19,926	15,894.53	39,774.34	6,182	\$23,879.81
- All Gender	- All Marital Status	- All Customers	- Alcoholic Beverages	1,812	1,317.13	3,314.52	590	\$1,997.39
- All Gender	- All Marital Status	- All Customers	- Alcoholic Beverages	12,985	12,228.85	30,528.85	4,189	\$18,280.00
- All Gender	- All Marital Status	- All Customers	- Alcoholic Beverages	12,037	10,108.87	25,318.93	3,968	\$15,210.06
- All Gender	- All Marital Status	- All Customers	- Alcoholic Beverages	4,132	3,684.50	9,200.76	1,339	\$5,515.86
- All Gender	- All Marital Status	- All Customers	- Alcoholic Beverages	26,655	22,000.66	55,207.50	8,684	\$33,176.84
- All Gender	- All Marital Status	- All Customers	- Beverages	1,714	1,465.42	3,669.89	555	\$2,204.47
- All Gender	- All Marital Status	- All Customers	- Beverages	37,792	32,831.33	82,248.42	12,310	\$49,417.09
- All Gender	- All Marital Status	- All Customers	- Beverages	1,764	1,520.70	3,829.14	577	\$2,288.44
- All Gender	- All Marital Status	- All Customers	- Beverages	30,545	26,663.34	67,629.82	9,957	\$40,645.48

Le Nombre de Tuples : 4860

Figure 5.11 : Cube de données non compressé

❖ La sélection des dimensions :

Personnaliser Les Dimensions

<input checked="" type="checkbox"/> Customers	2
<input checked="" type="checkbox"/> Gender	4
<input type="checkbox"/> Product	poids
<input checked="" type="checkbox"/> Marital Status	1
<input checked="" type="checkbox"/> Promotions	3

Suivant **Retour**

Figure 5.12 : La sélection des dimensions

❖ La sélection des niveaux :

Personnaliser Les Niveaux

*** Dimension Customers :**

<input checked="" type="checkbox"/> Name	1
<input type="checkbox"/> Country	poids
<input checked="" type="checkbox"/> City	2
<input checked="" type="checkbox"/> Province	3

*** Dimension Gender :**

<input checked="" type="checkbox"/> Gender	1
--	---

*** Dimension Marital Status :**

<input checked="" type="checkbox"/> Marital Status	1
--	---

Suiv... **Retour**

Figure 5.13 : La sélection des niveaux (A)

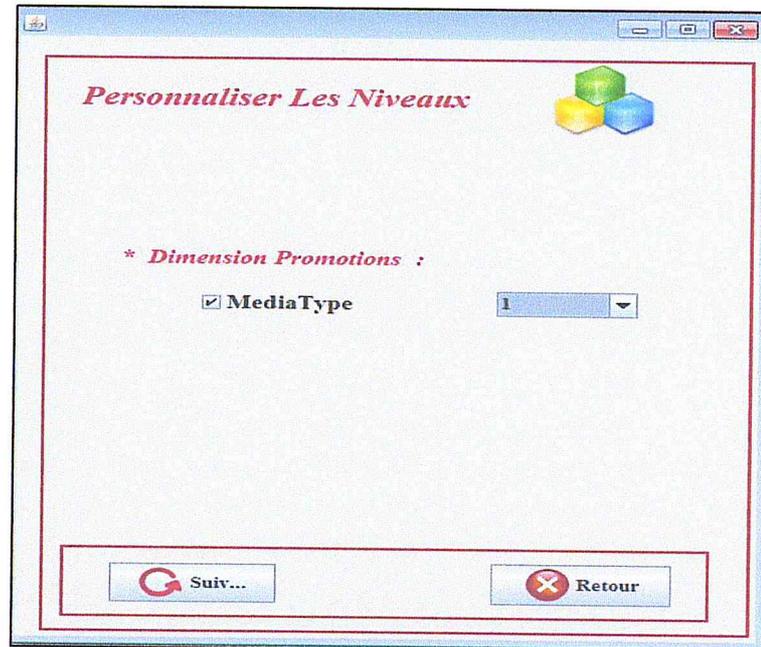


Figure 5.14 : La sélection des niveaux (B)

❖ La sélection des unités mesurées :

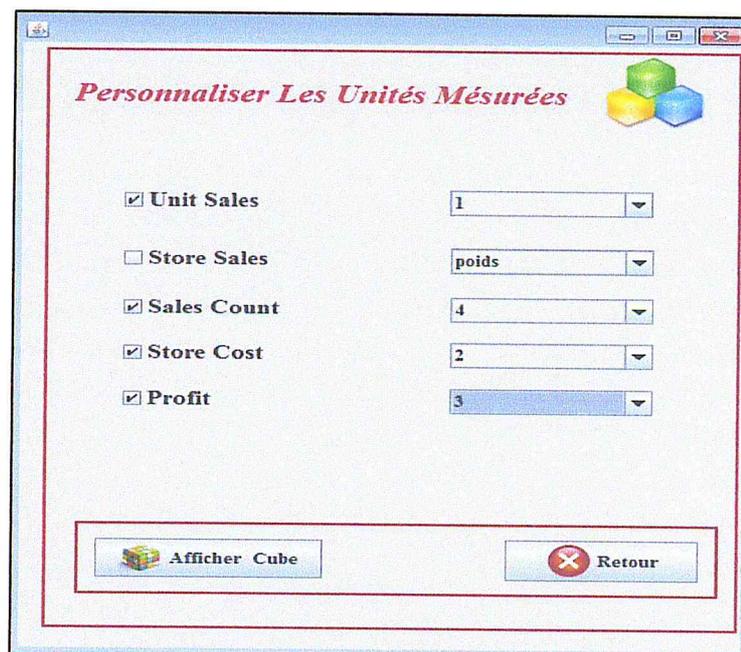


Figure 5.15 : La sélection des unités mesurée

CHAPITRE V : Réalisation

Après la sélection de l'utilisateur, Le programme de notre application exécute l'algorithme que nous proposons dans le chapitre précédant (Génération de la requête MDX personnalisé selon les choix préférés de l'utilisateur), la figure suivante (**figure 5.16**) montre le résultat d'exécution :

```
***** le résultat d'algorithme de compression*****
les choix préférés se sont :
** les dimension préférés :
la valeur da dimension : Marital Status leur poids est : 1
la valeur da dimension : Customers leur poids est : 2
la valeur da dimension : Pronotions leur poids est : 3
la valeur da dimension : Gender leur poids est : 4
*****
** les dimension préférés :
la valeur da niveau : Male ,leur poids est : 1 ,de dimension : Customers
la valeur da niveau : Gender ,leur poids est : 1 ,de dimension : Gender
la valeur da niveau : Marital Status ,leur poids est : 1 ,de dimension : Marital Status
la valeur da niveau : MediaType ,leur poids est : 1 ,de dimension : Pronotions
la valeur da niveau : City ,leur poids est : 2 ,de dimension : Customers
la valeur da niveau : Province ,leur poids est : 3 ,de dimension : Customers
*****
** les unités mesurés préférés :
la valeur da mesure : Unit Sales leur poids est : 1
la valeur da mesure : Store Cost leur poids est : 2
la valeur da mesure : Profit leur poids est : 3
la valeur da mesure : Sales Count leur poids est : 4
** Générer une requete MDX personnalisé :
la requete mdx personnalisé:
select {[Measures].[Unit Sales],[Measures].[Store Cost],[Measures].[Profit]} ON COLUMNS, Hierarchize(Union(Union(Crossjoin({[Marital Status].[All Marital Status]}), Union(Cross
from [Sales]
where [Time].[1997]
Driver loaded
connexion established
le nombres de mebres est :5616
le nombres de mesures est:1872
afficher le nbr de tuple
1872
```

Figure 5.16 : Le résultat d'algorithme de compression

Après l'exécution de notre algorithme, le cube de données compressé sera affiché à l'utilisateur, ce cube possède uniquement les données pertinentes celles parmi choisi par l'utilisateur (**figure 5.17**). On remarque que le nombre de tuples a été diminué.

Cube Vente Compressé

Marital Sta...	Customers	Promotions	Unit Sales	Store Cost	Profit
- All Marita...	- All Custo...	- All Prom...	266,773	225,627.23	\$339,610...
- All Marita...	- All Custo...	- All Prom...	901	758.79	\$1,154.07
- All Marita...	- All Custo...	- All Prom...	2,081	1,689.79	\$2,551.74
- All Marita...	- All Custo...	- Bag Stuff...	1,789	1,482.10	\$2,195.37
- All Marita...	- All Custo...	- Bag Stuff...	932	766.60	\$1,183.22
- All Marita...	- All Custo...	- Bag Stuff...	700	609.48	\$900.59
- All Marita...	- All Custo...	- Best Savi...	921	793.25	\$1,215.40
- All Marita...	- All Custo...	- Best Savi...	4,792	3,955.96	\$5,865.75
- All Marita...	- All Custo...	- Best Savi...			
- All Marita...	- All Custo...	- Big Promo	1,219	1,045.75	\$1,562.96
- All Marita...	- All Custo...	- Big Promo	781	665.25	\$1,013.63
- All Marita...	- All Custo...	- Big Promo	1,652	1,384.72	\$2,076.45
- All Marita...	- All Custo...	- Big Time...	1,959	1,614.17	\$2,444.80
- All Marita...	- All Custo...	- Big Time...	843	747.34	\$1,110.90
- All Marita...	- All Custo...	- Big Time...			
- All Marita...	- All Custo...	- Big Time...	1,638	1,344.21	\$2,041.18
- All Marita...	- All Custo...	- Big Time...	689	546.42	\$821.89
- All Marita...	- All Custo...	- Big Time...	1,607	1,349.63	\$2,013.46

Le Nombre de Tuple: 1872

Figure 5.17 : Cube de données compressé après la personnalisation

7. Conclusion :

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

A decorative purple border with rounded corners and a double-line effect, enclosing the text.

*C*ONCLUSION
*G*ENERALE

CONCLUSION GÉNÉRALE

- ❖ Enfin, le déploiement du système. Une base de données commercial de l'entrepôt « *Foodmart* » est utilisée pour notre test, elle possède sept cubes ; nous avons utilisé le cube « *Sales* ».

Notre système est implémenté sur : *Microsoft office ACCESS* comme serveur de données et *Mondrian* pour serveur des cubes OLAP.

La finalité de ce mémoire s'articule autour de concept de réduire la taille des cubes OLAP existants, permettant de sélectionner un sous-ensemble des données correspondant au contexte d'analyse d'un utilisateur mobile. La taille réduite des cubes personnalisés rend possible leur transmission sur des réseaux sans-fil dont la bande passante est limitée, ainsi que leur stockage sur un appareil mobile.

Durant le déroulement de notre projet, nous avons acquis une bonne expérience dans le domaine des SI, précisément les SI décisionnels basés sur Le Business Intelligence, le Data Waterhouse, le Data Mart, l'OLAP, sont des nouveaux concepts que nous avons tenus.

Ce mémoire est la première itération du projet. Il pose une base de réflexion concrète qui peut toujours être améliorée, corrigée ou complétée.

- Perspectives :

- ➡ Client OLAP Mobile : exécution et test notre application sur le vrais terminale mobile (androïde, iPhone)
- ➡ Améliorer la communication entre le client mobile et le serveur d'application (transmission des données volumineux [cube de données]).

CONCLUSION GÉNÉRALE

Conclusion général :

Le travail accompli au cours de notre projet de recherche a permis de mieux positionner la faisabilité et les critères technologiques à considérer pour amener une analyse OLAP à fonctionner dans les environnements informatiques mobiles, afin d'améliorer la prise de décision, tant locale que mobile . L'objectif général du notre projet est de de mettre en œuvre une architecture système pour la compression des cubes OLAP à l'aide de la technique de personnalisation des requêtes, orienté vers les environnements mobiles.

Notre architecture de système est sous forme d'une architecture tiers divisés en trois parties : une partie présentation, une partie traitement et une partie données.

Pour atteindre notre objectif, un certain nombre d'étapes ont été franchies :

- ❖ Premièrement, nous avons identifié les contraintes des environnements informatiques mobiles qui font en sorte que l'analyse OLAP de bureau actuelles doivent être modifiées et adaptées pour pouvoir être portées sur des plateformes mobiles. Ces contraintes se rapportent aux interfaces utilisateurs et aux méthodes d'entrée, aux capacités de calcul, de stockage et d'alimentation électrique, aux liens de communication et à l'interopérabilité entre systèmes. Dans le cadre de ce projet de recherche, nous nous sommes concentrés sur ce qui est nécessaire pour adapter les cubes de données OLAP face à ces contraintes.
- ❖ Deuxièmement, nous avons utilisé la technique de personnalisation des requêtes basées sur des préférences pondérés appliqués sur les métadonnées d'un cube OLAP déjà généré sur le serveur OLAP.

Deux pseudos algorithmes ont été proposés : le premier coté client OLAP Mobile permet de sélectionner les métadonnées préférés, calculer la taille de stockage réservé pour le cube OLAP reçu sur le mobile, le second permet d'extraire les données pertinentes à partir du cube original.

A decorative purple frame with a double-line border and rounded corners, enclosing the text.

*A*NNEXE

1. L'architecture trois-tiers :

Une architecture trois-tiers est un modèle d'architecture d'applications à trois niveaux, Elle permet de distinguer rigoureusement la couche « présentation », la couche « métier » et la couche « données » d'une application (**Figure 4.1**). Elle permet une évolution de l'un de ces trois tiers de façon, relativement, indépendant des deux autres.

Cette notion de couches n'émet aucune hypothèse quant à la distribution physique de l'application.

Les informaticiens sont ainsi affranchis de la localisation du code, que celui-ci soit dans un autre processus ou sur une autre machine. Mais cette séparation en couches a aussi d'autres objectifs :

- **L'évolutivité** : ce découpage offre la garantie de la meilleure évolutivité possible pour chaque couche.
- **La mise à jour** : d'une couche n'affecte pas ou peu les autres couches, dans la mesure de ses relations avec les couches voisines. Un changement dans la partie métier n'oblige par exemple pas à redéployer tous les clients.
- **La sécurité** : les données sont traitées par le tiers métier et ne sont pas visibles directement par les clients. Pour répondre à des exigences de sécurité (par exemple celles demandées par des applications prenant en charge des achats ou transactions bancaires depuis le domicile), on peut intégrer des niveaux d'isolation entre les couches comme des protocoles d'authentification (qui peuvent représenter eux-mêmes un niveau de l'architecture).
- **L'interopérabilité** : l'uniformisation des interfaces entre les couches, et/ou la mise en place de protocoles standardisés d'échange permet de faire communiquer des services réalisés avec des moyens différents (langage de programmation, protocoles, ...) et sur des supports hétérogènes (système d'exploitation, matériel, ...).

ANNEXE

- **La réutilisabilité** : les entreprises ont besoin d'accéder et de distribuer des informations de leurs systèmes d'application antérieurs à Internet. Cette architecture permet, grâce à des protocoles de communication, d'utiliser l'existant. D'autre part, grâce à la séparation logique, l'entreprise pourra réutiliser des composants dans d'autres applications.

- **La maîtrise de la montée en charge** : on peut notamment modifier le déploiement physique selon les besoins d'exploitation et utiliser des « serveurs d'application » qui optimisent l'utilisation des ressources pour de nombreux utilisateurs.

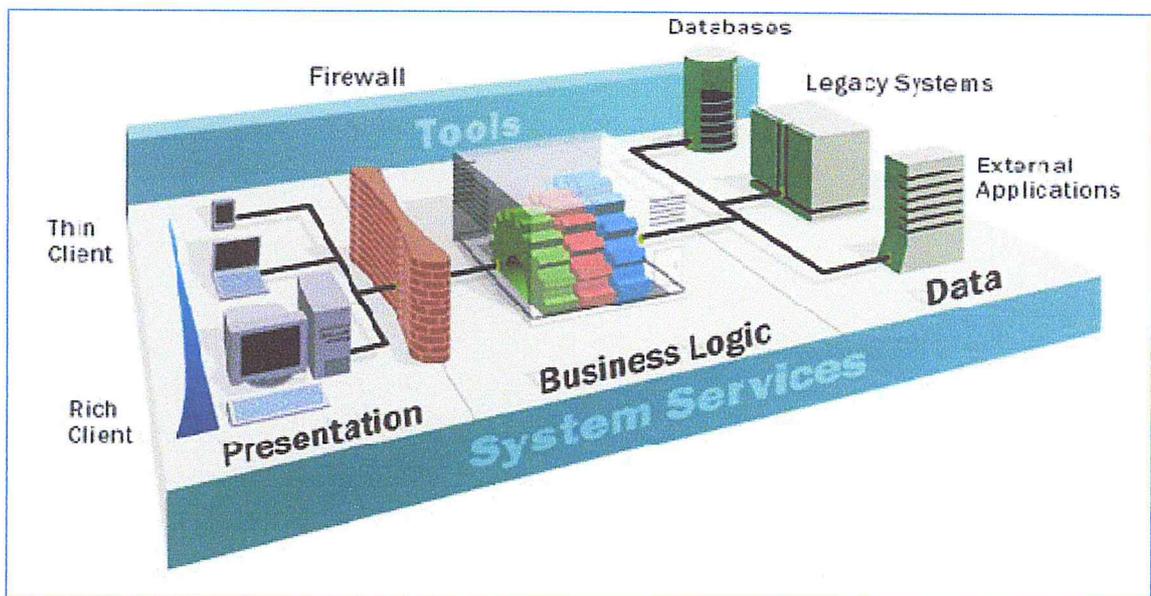


Figure 4.1: Architecture trois-tiers

Références

Bibliographique

Bibliographie

- [1] : ETIENNE Dubé, Conception et Développement d'un Service Web de Constitution de Mini Cubes SOLAP pour Clients Mobiles, l'Université Laval QUÉBEC (2008) , Article , pages 2- 30.
- [2] : Omar Boussaid , Fahima Nader , Nachida Rezoug ,Implémentation d'Olap dans les environnements mobiles : aperçu de l'état de l'art. Article pages 1-7.
- [3] : Lamiaa NAOUM, Un modèle multidimensionnel pour un processus d'analyse en ligne de résumés flous ,thèse de doctorat, UNIVERSITÉ DE NANTE ECOLE POLYTECHNIQUE (novembre 2006), pages 7- 17 et 31 - 36 .
- [4] : BOUZIDI EL Hocine , SADKI Omar , Conception et implémentation d'un entrepôt de données et application des techniques du Data Mining pour l'analyse des échecs/succès des étudiants de l'ESI mémoire fin d'étude (2010), pages 6 – 39 .
- [5] : W. Bill Inmon, « Building the data warehouse », Edition Wiley Computer Publishing, USA 2002.
- [6] : Olivier TESTE, Modélisation et manipulation d'entrepôts de données complexes et historisées. Thèse de Doctorat, Université Paul Sabatier de Toulouse (2000).
Pages 7-23
- [7] : Ralph Kimball: « Entrepôt de données », International Thomson Publishing France, 1997.
- [8] : Mme Boucetha Lila née Bousnina, Personnalisation des requêtes OLAP, mémoire de magistère, Université M'hamed Bougara de Boumerdes (2010)
pages 7-14 et 50 – 60.
- [9] : J.-F. Desnos, Entrepôt de données – Introduction pages 4 - 6 .
<http://imss-www.upmf-grenoble.fr/prevert/SpecialiteIHS/ED/IntroductionED.pdf>
(mise à jour novembre 2013).
- [10] : Pascal Wehrle, Modèle multidimensionnel et OLAP sur architecture de grille , L'institut National des Sciences Appliquées de Lyon thèse de doctorat (2009). Pages 18 - 22 .
- [11] : Alio Boly, FONCTIONS D'OUBLI ET RÉSUMÉS DANS LES ENTREPÔTS DE DONNÉES, Thèse de doctorat de l'Ecole Nationale Supérieur de Télécommunication de Paris. (décembre 2006) , pages 13- 15.

REFERENCES BIBLIOGRAPHIQUES

- [12] : Ralph Kimball: « Concevoir et déployer un data warehouse » Edition Eyrolles, 2000, France.
- [13] : E. F. Codd. Providing OLAP (On-Line Analytical Processing) to user analysts: An IT mandate. In Technical Reports. IBM, 1993.
- [14] : Ladjel Bellatreche, Techniques d'optimisation des requêtes dans les data warehouses , France (2003) Article, pages 4- 5.
- [15] : Bernard ESPINASSE, Entrepôts de données : Systèmes OLAP : ROLAP, MOLAP et HOLAP (5), l'Université d'Aix-Marseille (AMU), Ecole Polytechnique Universitaire de Marseille, Septembre 2013, pages 13- 61.
- Andreas S. Maniatis, The Case for Mobile OLAP, National Technical University of Athens, Department of Electrical and Computer Engineering, Knowledge and Database Systems Laboratory, 2004. , Article, pages 3- 7.
- [16] : O. FOUIAL, «Découverte et fourniture de services adaptatifs dans les environnements mobiles», Thèse de doctorat en Informatique et Réseaux de l'Ecole Nationale Supérieure des Télécommunications ENST, Paris, France, 30 Avril, 2004.
- [17] : Ilias Michalarias, Multidimensional Data Management in Mobile Environments, Berlin Brandenburg Graduate School On Distributed Information Systems Freie University Berlin, 2010.
- [18] : M.SHARAF and P. K. CHRYSANTHIS On-Demand Data Broadcasting for Mobile Decision Making Mobile Networks and Applications 9, pages 703–714, 2004.
- [19] : J. F. J. X. Y. Wei Wang, Hongjun Lu. Condensed cube: An efficient approach to reducing data cube size. ICDE, 2002.
- [20] : Laks V.S. Lakashmanan, Jian Pei, et Jiawei Han. Quotient Cube : How to Summarize the Semantics of a Data Cube. In Proceedings of International Conference of Very Large Data Bases, VLDB'02, 2002, article Pages 778 - 789.
- [21] : Laks V.S. Lakshmanan, Jian Pei, et Yan Zhao. QC-Trees: An Efficient Summary Structure for Semantic OLAP. In ACM Press, editor, Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, 2003, Article, Pages 64-75.
- [22] : I. Michalarias and H.-J. Lenz. Dissemination of multidimensional data using broadcast clusters. In Distributed Computing and Internet Technology, volume 3816 of LNCS, pages 573–584. Springer, 2005.
- [23] :

REFERENCES BIBLIOGRAPHIQUES

- [24] : Ilias Michalarias, Arkadiy Omelchenko, "Compressed Aggregations for mobile OLAP Dissemination," *dexa*, pp.609-614, 18th International Conference on Database and Expert Systems Applications (DEXA 2007) .
- [25] : Alfredo Cuzzocrea Filippo Furfaro ·Domenico Saccà, Enabling OLAP in mobile environments via intelligent data cube compression techniques,2008 , Article , pages 99- 101.
- [26] : Koutrika, G. Ioannidis, Y. E. (2004). Personalization of queries in database systems. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 597–608.
- [27] : Mokrane Bouzeghoub, Dimitre Kostadinov. Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de définition de profils. In Actes de la seconde édition de la Conférence en Recherche d'Information et Applications (CORIA), pages 201 - 218, Grenoble. (France, 2005).
- [28] : Koutrika,G.; Ioannidis, Y. Constrained Optimalities in Query personalization*. Date de publication : (2005) , Pages: 73 – 84 ISBN:1-59593-060-4 ACM Press New York, NY, USA.
- [29] : Koutrika, G., Ioannidis, Y. E. (2005a). Personalized queries under a generalized preference model. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 841–852.
- [30] : Chirita, P.-A., Firan, C.S., Nejdl, W. (2007). Personalized query expansion for the Web. Intl. Conf. on Research and Development in Information Retrieval, pages 7- 14.
- [31] : Liu, F.,Yu, C., Andmeng,W. (2004). Personalized Web search for improving retrieval effectiveness. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 16, No. 1, pages 28–40.
- [32] : Lynda Tamine-Lechani, Nesrine Zemirli, Wahiba Bahsoun. Approche statistique pour la définition du profil d'un utilisateur de système de recherche d'information. *Information – Interaction - Intelligence*, 7(1), Laboratoire PRiSM, Université de Versailles (2007).article pages 1 – 16.
- [33] : Arnaud Giacometti, Patrick Marcel, Elsa Negre A Framework for Recommending OLAP Queries. In 11th ACM International Workshop on Data Warehousing and OLAP (DOLAP 08), Napa Valley, California, USA, pages 73–80. Laboratoire d'Informatique Université François Rabelais de Tours – France (2008).
- [34] : Mokrane Bouzeghoub,Dimitre Kostadinov. Une approche multidimensionnelle pour la personnalisation de l'information. « INRIA Roocquencourt et laboratoire PRiSM, université de versailles 45, avenue des Etats-Unis,78035 Versailles ». (2004).

REFERENCES BIBLIOGRAPHIQUES

- [35] : Lynda Tamine, Wahiba Bahsoun. Définition d'un profil multidimensionnel de l'utilisateur : Vers une technique basée sur l'interaction entre dimensions. Programme ACI Masses de Données, projet MD-33 (<http://apmd.prism.uvsq.fr/>). Laboratoire IRIT (Equipe SIG) Université Paul Sabatier (2008). pages 1 -12.
- [36] : Lynda Lechani Tamine, Mohand Boughanem. Accès personnalisé à l'information : vers la définition d'un profil utilisateur multidimensionnel. In International Symposium On Programming Systems, pages 20 - 28. USTHB. (2005).
- [37] : Golfarelli, M., Rizzi, S., Biondi, P. (2011). myOLAP: An Approach to Express and Evaluate OLAP Preferences. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 23, No. 7, pages 1050–1064.
- [38] : Golfarelli, M. et Rizzi, S. (2009). Expressing OLAP Preferences. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 83–91.
- [39] : Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.
- [40] : Agrawal, R. and Wimmers, E. L. A framework for expressing and combining preferences. In Chen, W., Naughton, J. F., and Bernstein, P. A., editors, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA, pages 297–306. ACM. (2000).
- [41] : Kießling, W. (2002). Foundations of preferences in database systems. Intl. Conf. on Very Large Data Bases (VLDB), pages 311–322.
- [42] : Chomicki, J. (2003). Preference formulas in relational queries. ACM Trans. Database Syst. 28, 4, pages 427–466.
- [43] : Dey, A. K. (2001). Understanding and using context. Personal and Ubiquitous Computing, Vol. 5, No. 1, pages 4–7.
- [44] : Stefanidis, K., Pitoura, E., Vassiliadis, P. (2007). Adding context to preferences. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 846–855.
- [45] : Agrawal, R., Rantzaou, R., and Terzi, E. (2006). Context-sensitive ranking. ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD), pages 383–394.
- [46] : Stefanidis, K., Drosou, M., Pitoura, E. (2009). "You May Also Like" Results in Relational Databases. Intl. Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB), VLDB Workshops.

REFERENCES BIBLIOGRAPHIQUES

- [47] : Bunningen, A. H., Feng, L., and Apers, P. M. G. (2006). A context-aware preference model for database querying in an ambient intelligent environment. Intl. Conf. on Database and Expert Systems Applications (DEXA), pages 33 – 43.
- [48] : Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009). A Conceptual Modeling Approach for OLAP Personalization. Intl. Conf. on Conceptual Modeling (ER), pages 401–414.
- [49] : Cecile Favre. Évolution de schémas dans les entrepôts de données : mise à jour de hiérarchies de dimension pour la personnalisation des analyses. Thèse de doctorat, Université Lumière Lyon 2 École Doctorale Informatique et Information pour la Société (2007) , pages 71 – 133 .
- [50] : Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. (1999). Maintaining Data Cubes under Dimension Updates. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 346 –355.
- [51] : Ravat F., Teste O., Zurfluh G. (2007a). Personnalisation de bases de données multidimensionnelles. Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID), pages 121–136.
- [52] : Giacometti, A., Marcel, P., Negre, E. (2008). A Framework for Recommending OLAP Queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 73–80.
- [53] : Giacometti, A, Marcel, P., Negre, E. (2009). Recommending Multidimensional Queries. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 453–466.
- [54] : Stolte, C. (2003). Query, Analysis, and Visualization of Multidimensional Databases, Thèse de doctorat, Université de Stanford (Etats-Unis), Juin 2003.
- [55] : Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D. (2005). A personalization framework for OLAP queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 9–18.
- [56] : Cabanac, G., Chevalier, M., Ravat, F., Teste, O. (2007). An Annotation Management System for Multidimensional Databases. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), pages 89 – 98.
- [57] : Jerbi, H. (2007). Mémoire d'expertises décisionnelles à base d'annotations. Mémoire Master 2 Recherche, Université Paul Sabatier, Toulouse III, Juin 2007.
- [58] : Thalhammer, T., Schrefl, M., Mohania, M. (2001). Active DataWarehouses. Complementing OLAP with Analysis Rules, Data and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 39, No. 3, pages 241–269.

REFERENCES BIBLIOGRAPHIQUES

- [59] : A. K. Dey. Providing Architectural Support for Building Context-Aware Applications. PhD thesis, College of Computing, Georgia Institute of technology, December 2000.
- [60] : Rizzi, S.: OLAP preferences: a research agenda. In: DOLAP, pages 99–100 (2007).
- [61] : Jerbi, H., F. Ravat, O. Teste, et G. Zurfluh Management of Context- Aware Preferences in Multidimensional Databases. In ICDIM 08, pages 669-675. (2008).
- [62] : <http://searchsoa.techtarget.com/definition/Web-services>, 09/09/2013.
- [63] : <http://news.xmlforanalysis.com/olap/> 09/09/2013.
- [64] : <http://technet.microsoft.com/en-us/library/cc966471.aspx>,09/09/2013.
- [65] : <http://www.istantic.com/hv2/programmation/Java/Generalites/Generalites.htm>
05/07/2013
- [66] : <http://www.dominiqueverriere.com/2011/01/decouverte-du-langage-mdx-partie-1.html>,
09/09/2013
- [67] : <http://technet.microsoft.com/en-us/library/ms144816.aspx>, 09/09/2013 .
- [68] : <http://www.osbi.fr/cubes-olap-avec-mondrian/>09/09/2013.
- [69] : <http://tomcat.apache.org/>09/09/2013.
- [70] : <http://www.olap4j.org/>,09/09/2013
- [71] : http://www.olap4j.org/olap4j_fs.html, 09/09/2013
- [72] : <http://technet.microsoft.com/fr-fr/library/ms174518.aspx>, 09/09/2013
- [73] : <http://atatorus.developpez.com/tutoriels/webservice/soap-apachecxf-serveur-client/>,
09/09/2013
- [74] : <http://mbaron.developpez.com/soa/jaxws/>,09/09/2013
- [75] : http://www-igm.univ-mlv.fr/~dr/XPOSE2005/rouvio_WebServices/soap.html,
09/09/2013