

la décision (SID)1.23System d'aide à la décision (SID)section.1.2 t et le multidimensionnelle 1.35L'entrepôt et le multidimensionnellesection.1.3

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ Ammar TELIDJI - LAGHOUAT

Faculté des Sciences

Département de Mathématique et Informatique



MÉMOIRE DE MASTER

Domaine: Mathématique et Informatique

Filière: Informatique

Option: Système d'Information & Décision

Thème

**Langage d'interrogation graphique des bases de données
multidimensionnelles temporelles**

Présenté par:

SEKKOUTI Khaled

BOUSBAA Yacine

Soutenu devant le jury composé de :

Mr T.BENDOUMA	Président	U. Amar Telidji, Laghouat
Mr Y.GUELLOUMA	Examineur	U. Amar Telidji, Laghouat
Mr L.AMEUR	Examineur	U. Amar Telidji, Laghouat
Mr L.CHELLAMA	Encadreur	U. Amar Telidji, Laghouat

Année //2014-2015

Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut... Tous les mots ne sauraient exprimer la gratitude, L'amour, le respect, la reconnaissance... Aussi, c'est tout simplement que

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, sa tendresse, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude. Ce travail est dédié à mon père nadir SEKKOUTI, décédé trop tôt, qui m'a toujours poussé et motivé dans mes études. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'un fils qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde ! Mon frère yacine et mes soeurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité. À MES AMIS DE TOUJOURS : HARROUZI, BACHIR, BRAHIM, YACINE, YACINE BOUSBA, DAHMANE, ... En souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble. Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère. À ma fiancée, reçois à travers de ce travail tout mon respect, ma gratitude et Ma profonde reconnaissance. À tous les membres de ma famille, petits et grands Veuillez trouver dans ce modeste travail l'expression de mon affection. À TOUTES LES PERSONNES QUI ONT PARTICIPÉ À L'ÉLABORATION DE CE TRAVAIL À TOUS CEUX QUE J'AI OMIS DE CITER À tous ceux qui, par un mot, m'ont donné la force de continuer ...

Khaled SEKKOUTI

Dédicace

*Je dédie ce modeste travail à mes très chers parents qui m'ont
toujours apporté leur soutien, leur dévouement, leurs
encouragements, et mon appris à ne pas baisser les bras dans la
vie. Que dieu vous bénisse et vous garde ,
ainsi mes chères frères Ramzi et Mouhamad et ma chère sœur
Amel, et à toute ma famille.*

*À tous mes chers amis Nouredine.E, Mohammed.S,
Mohammed.D, Mounir.A ,Mohammed Amine.B, Youcef.B,
Bachir.F, Mohammed.B, Elhadj.B, Tefik.S; Pour tous les
instants inoubliables que j'ai passés, et que je vais les passés
avec vous.*

*À mon binôme Khaled SEKKOUTI que je remercie pour tous
ses efforts.*

*À tous ceux qui m'aiment et ceux que j'aime,
À toi ...*

Yacine BOUSBA

Remerciement

En préambule à ce mémoire, Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail. En second lieu, nous tenons à remercier notre encadreur Mr CHELLAMA Laradj , pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité. Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions. Enfin, nous souhaitons adresser nos remerciements les plus sincères aux personnes proches de notre cœur , mes amis mes proches qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce modeste travail. Merci à tous et à toutes.

Resumé

Notre travail est inclus dans le contexte des systèmes décisionnels et plus précieusement, dans le domaine des bases de données multidimensionnelles. Ce travail consiste à définir un prototype d'interrogation graphique de bases de données multidimensionnel permettant la gestion de l'évolution temporelle. Après avoir présenté les travaux sur les modèles et les langages de manipulation multidimensionnels temporels, nous présentons le modèle multidimensionnel de notre travail.

Mots clé: Base de données multidimensionnelles ,évolution temporelle, langage graphique.

Abstract

Our work is included in the context of decision systems more carefully, in the field of labor multidimensionnelles. Ce databases is to define a graphical query language of multidimensional databases for management of the time course. After presenting the work on models and temporal multidimensional manipulation languages, we present the multidimensional model of our work. .

Keywords: Multidimensional database, temporal evolution, graphic query language.

ملخص

يُندرج عملنا في إطار نظم صنع القرار بالضبط في مجال قواعد البيانات متعددة الأبعاد. هذا العمل يتضمن إنشاء نموذج للاستفسار عن قواعد بيانات متعددة الأبعاد لإدارة التطور الزمني بعد عرض الأعمال على نماذج ولغات التلاعب متعددة الأبعاد الزمانية، نقوم بتقديم نموذج متعدد الأبعاد الخاص بعملنا.

الكلمات المفتاحية: قاعدة بيانات متعددة الأبعاد، التطور الزمني، رسومية لغة الاستعلام.

Tableau des Matières

Remerciement	iii
Resumé	iv
Liste des Abréviations	xi
Introduction	1
1 Généralité multidimensionnelle	3
1.1 Introduction	3
1.2 System d'aide à la décision (SID)	3
1.2.1 Définition	4
1.2.2 Architecture	4
1.3 L'entrepôt et le multidimensionnelle	5
1.3.1 Entrepôt et Magasins de données	5
1.3.2 Le Multidimensionnel	8
1.4 Les données temporelles	17
1.4.1 Définition	17
1.4.2 Quelque approche temporelle	18
1.5 Classification des requêtes	22
1.6 Domaines d'application	23
1.7 Logiciels	23
1.7.1 Mondrian	23
1.7.2 Palo	24
1.7.3 FreeAnalysis	24
1.8 conclusion	25

2	<i>Etat de l'art</i>	26
2.1	Introduction	26
2.2	Langage d'interrogation	27
2.2.1	OQL (Object Query Language)	27
2.2.2	SQL (sigle de StructuredQueryLanguage)	28
2.2.3	MDX	28
2.3	Les langages d'interrogation temporels	30
2.3.1	langages pour les données multidimensionnelles	31
2.4	langages graphiques	32
2.4.1	Langage FRED	32
2.4.2	Langage COMET	33
2.4.3	Langage MVDW	34
2.4.4	Langage GRAPHIC-OLAP	36
2.4.5	Langage XML-GOLAP	37
2.4.6	Langage GOLAP Personnel	39
2.4.7	Langage Docware	41
2.4.8	COGNOS PowerPlay	42
2.4.9	CRYSTAL Analysis	42
2.5	conclusion	43
3	<i>Implémentation</i>	44
3.1	Introduction	44
3.2	Conception	44
3.2.1	Langages	45
3.2.2	Environnement de développement	48
3.3	Structure de l'outil	50
3.4	Conclusion	52
	Conclusion	54
	Bibliographie	55

Tableau des Figures

1.1	Architecture des systèmes décisionnels	4
1.2	Table de fait Ventes	9
1.3	Table de dimension temps	10
1.4	Représentation multidimensionnelle (cube)	10
1.5	EXEMPLE D'UN SCHEMA EN ETOILE	11
1.6	Schéma en constellation	11
1.7	Exemple Drill _{Up} Drill – down	13
1.8	Slice	14
1.9	Exemple Dice	14
1.10	Exemple Pivot	15
2.1	Architecture de systèmes d'aide à la décision Fred	32
2.2	L'outil d'administration COMET	34
2.3	Architecture de langage GOLAP	37
2.4	Architecture de langage XML GOLAP	39
2.5	Architecture de langage GOLAP Personnel	40
2.6	Architecture de langage DocWare	41
3.1	Fonctionnement du JavaScript	47
3.2	Fonctionnement du langage PHP	49
3.3	La Conception utiliser	50
3.4	L'aspet temps	51
3.5	Résultat et Opération OLAP	51
3.6	Résultat par Switch	52
3.7	Résultat par rollUP	53

3.8	Résultat graphique	53
-----	------------------------------	----

liste des Tableaux

1.1	Comparaison entre SGBD et DW	8
1.2	Modification de la valeur d'un attribut	18
1.3	Modification <i>n_uplets</i>	19
2.1	Comparaison entre MDX et SQL	29

Liste des Abréviations

BDM: base de données multidimensionnelle

BPM: Business Process Management

CSS: Cascading Style Sheet

DOCWARE: DOCument WAREhouse

DW: Data warehouse

DWQ: Data Warehouse Quality

ED: Entrepôts de donnée

ETL: Extract, Transform, Load

GPL: GNU General Public License

HTML: HyperTextMarkupLanguage

Hybrid OLAP: Hybrid online analytical processing

IBM: International Business Machines Corporation

JS: JavaScript

LCD: Le langage de Contrôle des Données

LDD: Le langage de Définition des Données

LMD: Le langage de Manipulation des Données

MDX: Multi Dimensional eXpression

MOLAP: Multidimensional online analytical processing

MPL: Mozilla Public License

MVDW: Multi-version data warehouse

ODMG: Object Database Management Group

OLAP: Online analytical processing

OLTP: Online transaction processing

OQL: Object Query Language

PHP: Hypertext Preprocessor

QBE: Query By Example

RATP: Régie autonome des transports parisiens

ROLAP: Relational online analytical processing

SGBD: Système de gestion de base de données

SID: System d'aide à la décision

SPJ: Sélection - Projection - Jointure

SQL: Sigle de StructuredQueryLanguage

TM: Table multidimensionnelle

UML: Unified Modeling Language

W3C: World Wide Web Consortium

XDE: EXtended environnement de développement

XSL: Extensible Stylesheetlanguage

Introduction

Le pilotage stratégique de l'entreprise est de plus en plus complexe et la quantité de paramètres à prendre en compte n'arrête pas de changer. Les nouvelles technologies de l'information apportent des solutions efficaces et des outils performants à ces problématiques avec l'accroissement du potentiel des machines.

Les Systèmes d'Information (SI) permettent aux entreprises de centraliser les nombreuses informations, d'extraire les informations décisives et de piloter efficacement la stratégie.

L'informatique décisionnelle a pour objectif d'élaborer des systèmes d'analyse de données dédiés au soutien et à l'amélioration des processus décisionnels des organisations. Ces systèmes OLAP (On-Line Analytical Processing) sont généralement constitués de bases de données multidimensionnelles, communément appelées magasins de données (datamarts).

Le modèle multidimensionnel est le plus utilisé pour la modélisation de ces Datamarts. Il est l'un des nouveaux développements remarquables de la conception des bases de données car il étend de façon considérable les possibilités d'analyse de grands ensembles de données multidimensionnels.

Malgré cela, la structure statique de ce modèle ne prise pas en compte l'évolution des sources de données et des besoins des décideurs. C'est pour cela que de nouveaux modèles dits multidimensionnels temporels ont été définis, qui intègrent l'aspect temporel dans le modèle multidimensionnel.

L'interrogation multidimensionnelle des données représente aussi un axe de recherche important, visant à développer des outils d'interrogation graphique simples, pour permettre une utilisation plus facile de ces bases de données décisionnelles, dédiées aux décideurs souvent non familiers avec l'informatique.

Dans le contexte décisionnel, notre travail consiste à proposer un prototype graphique d'interrogation des bases de données qui soit simple et interactif.

Notre document est organisé comme suit :

Chapitre 1: Généralité

Ce chapitre permet de récapituler et de donner des généralité sur les base des données multidimensionnelle et la notion temporel.

Chapitre 2: Etat de l'art

Ce chapitre permet de récapituler et de donner une idée sur les travaux réalisés sur le sujet de langage graphique et langage d'interrogation.

Chapitre 3: Implémentation

Ce chapitre permet de présenter notre conception et de lister tous les outils utilisés pour l'implémentation et de donner une idée sur notre outil .

Conclusion

Ce chapitre permet de récapituler les chapitres précédents et de spécifier les perspectives qui peuvent faire l'objet d'une autre recherche.

Chapter 1

Généralité multidimensionnelle

1.1 Introduction

Le changement effectuer aux informations et le croisement des technologies de l'information a augmenté le volume des informations disponibles de manière considérable. Les données collectées en des avantages puissants pour éclairer des résultats passés, actuelles et surtout futures, les entreprises font face à un volume croissant de données qui transitent au sein de leur système d'information et cette masse d'informations rend difficile leur exploitation.

Pour une bon exploitation de ces donnée il est naturel des chercher des actus qui donne une meilleur exploitation de ces données. Ils sont apparus pour gérer de très gros volumes de données issues de sources hétérogènes, les systèmes d'aide à la décision ont été mis en place au sein des entreprises. Ces systèmes permettent un traitement synthétique de l'information pour faciliter les prises de décisions.

1.2 System d'aide à la décision (SID)

De nos jours, les décideurs ont besoin d'une vision synthétique et globale des informations circulant dans l'entrepris afin de guider et adapter leur prise de décision. Pour faciliter ce processus, ils emploient des systèmes d'aide à la décision qui permettent aux décideurs d'avoir une vision globale sur les activités d'une entreprise par un accès rapide et interactif à un ensemble de vues des données organisées pour refléter l'aspect multidimensionnel des

données de l'entreprise.

1.2.1 Définition

Un système d'aide à la décision est l'ensemble des outils informatiques (matériels et logiciels) qui permettent l'analyse des données opérationnelles issues du système d'information des entreprises. Ces données sont transformées en une vision orientée décideur puis analysées au moyen de manipulations et restitutions adaptées [1].

1.2.2 Architecture

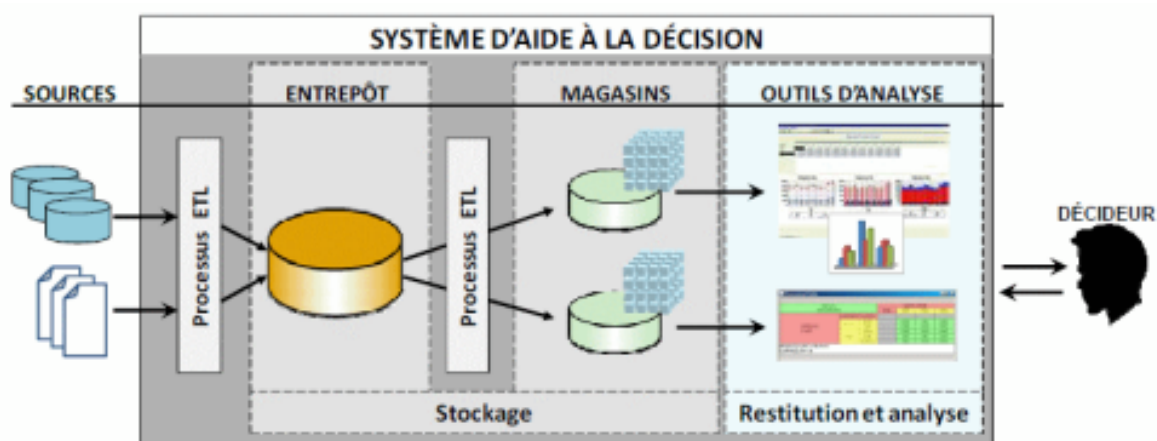


Figure 1.1: Architecture des systèmes décisionnels [1]

Comme présent la Figure 1.1, trois catégories d'outils sont employées :

- * Les outils d'extraction, de transformation et de chargement ou ETL (Extract, Transform, Load) des données opérationnelles pour alimenter et rafraîchir les données contenues dans le système d'aide à la décision, peuvent être internes (bases de production) ou externes (Internet, bases des partenaires) à l'entreprise.
- * Les outils de stockage et de traitement des données décisionnelles, extraites des données opérationnelles (l'entrepôt de données, Les magasins de données).
- * Les outils de restitution et d'analyse des données décisionnelles sous une forme adaptée aux décideurs qui permettent de manipuler et visualiser les données au travers d'interfaces interactives et fonctionnelles dédiées à des décideurs souvent non informaticiens (directeurs, chefs de services, etc.).

1.3 L'entrepôt et le multidimensionnelle

1.3.1 Entrepôt et Magasins de données

Au cours de ces dernières années, les entrepôts de données ont joué un rôle essentiel dans le domaine de l'informatique décisionnelle en soutenant et en améliorant les processus décisionnels des organisations.

Entrepôt de donnée

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

«L'entrepôt est une collection de données orientées sujet, intégrées, non volatiles et historiques, organisées pour le support d'un processus d'aide à la décision» [W.H. Inmon (1996)].

- **Donnée orientées sujet:** les données d'un entrepôt s'organisent par sujets ou thèmes. Cette organisation permet de rassembler toutes les données, pertinentes à un sujet et nécessaires aux besoins d'analyse, qui se trouvent répandues à travers les structures fonctionnelles d'une entreprise.
- **Donnée intégrée:** les données d'un entrepôt sont le résultat de l'intégration de données en provenance de multiples sources ; ainsi, toutes les données nécessaires pour réaliser une analyse particulière se trouvent dans l'entrepôt. L'intégration est le résultat d'un processus qui peut devenir très complexe du à l'hétérogénéité des sources.
- **Donnée non volatile:** les données chargées dans l'entrepôt sont surtout utilisées en interrogation et ne peuvent pas être modifiées, sauf dans certains cas de rafraîchissement.
- **Donnée historiques:** les données d'un entrepôt représentent l'activité d'une entreprise pendant une longue période où il est important de gérer les différentes

valeurs qu'une donnée prises au cours du temps. Cette caractéristique donne la possibilité de suivre une donnée dans le temps pour analyser ses variations.

Le DW se structure en quatre classes de données:

- **Les données détaillées** : Elles reflètent des événements les plus récents. Les données provenant des systèmes de production sont intégrées à ce niveau.
- **Les données agrégées** : Elles correspondent à des éléments d'analyse représentatifs des besoins des utilisateurs. Ce sont donc des données déjà traitées par le système et représentant un premier résultat d'analyse et de synthèse des données contenues dans les systèmes de production. Elles doivent être facilement accessibles et compréhensibles.
- **Les données historisées** : Chaque nouvelle insertion dans le datawarehouse ne détruit pas les anciennes valeurs mais crée une nouvelle insertion.
- **Les méta-données** : Il s'agit « de données sur les données ». Elles décrivent les règles ou processus attachés aux données du système. Les méta-données permettront notamment de connaître :
 - * quelles sont les données entreposées, leur format, leur signification, leur degrés d'exactitude.
 - * les processus de récupération/extraction dans les bases sources.
 - * la date du dernier chargement du datawarehouse.
 - * l'historique des données sources et de celles du datawarehouse.

Magasins de données (Data Marts)

Un magasin de données est un extrait de l'entrepôt conforme à des besoins d'analyse particuliers et organise selon un modèle adapté aux outils d'analyse et d'interrogation décisionnelle (un modèle en cube ou multidimensionnel). Le magasin est généralement stocké au sein d'une base de données multidimensionnelle (BDM).

- Se concentre sur 1 sujet d'analyse (ex: les ventes OU les livraisons, mais pas les deux).

- Sert à faire des analyses simples et spécialisées (ex: les fluctuations des produit par type).
- Nombre de sources limitées, provenant la plupart du temps d'un même département.
- Même processus de conception que les entrepôts de données, mais demande moins de ressources.
- Ils sont orientés sujet.
- Sont de petits entrepôts nécessitant une infrastructure plus légère et sont mis en œuvre plus rapidement.

Les données extraites sont adaptées pour l'aide à la décision (usage particulier, recherche de corrélation, logiciel de statistiques,...).

ED vs SGBD

	SGBD	ED
Utilisateur	Nombreux(Employés)	Peu(Analystes)
Donnée	Alphanumériques/ Détaillées/ Atomiques/ Dynamiques	Numériques/ Résumées/ Agrégées/ Statiques
Requête	Prédéfinies	Un seul utilisation
Accès	Peu de données (courantes)	Beaucoup d'informations (historiées)
Temps d'exaction	Court	Long
But	Dépend de l'application	Prend des décisions
Comparaison de données	Non : Archivage ou mise à jour des données	Oui : Données non volatiles, données historiées
Opérations réalisées sur les données	Consultation, mais surtout mise à jour et ajout de données	Consultation de données uniquement

Table 1.1: Comparaison entre SGBD et DW

1.3.2 Le Multidimensionnel

Un modèle multidimensionnel est la base des entrepôts de données et de l'analyse multidimensionnelle, il permet de stocker les données dans un modèle plus adapté au besoin de décideur. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse [1].

Ce modèle de données permet d'étudier les données sous plusieurs perspectives. Son axe d'analyse facilite l'accès aux données. Il est plus facilement compréhensible, même pour les personnes qui ne sont pas expertes en informatique.

Modélisation conceptuelle :

Au niveau conceptuel, il s'agit de présenter les données sous plusieurs dimensions selon les besoins des décideurs, cette modélisation se fait soit par cube ou concept de fait et dimensions. [2]

1. Concept de fait et dimension

- **Fait**

- * modélise le sujet de l'analyse.
- * formé par des mesures correspondant aux informations de l'activité analysée.
- * ces mesures sont numériques.

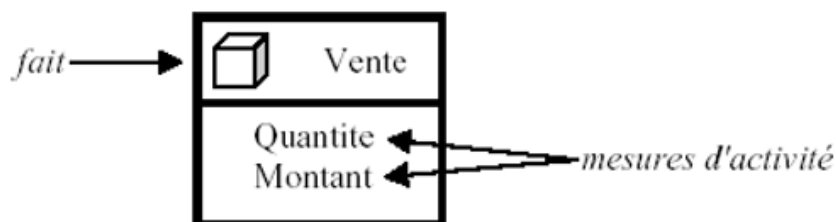


Figure 1.2: table de fait Ventes [3]

- **Concept dimension**

Le sujet analysé, le fait, est analysé suivant différentes perspectives ou axes caractérisant ses mesures de l'activité. Le fait est analysé selon différents axes appelés dimensions. Une dimension modélise donc une perspective d'analyse selon laquelle sont visualisées les mesures du fait. Une dimension est composée de paramètres qui représentent les niveaux de visualisation des mesures d'analyse. Par exemple le fait précédant (vente) peut être analysé sur plusieurs perspectives temps, catégorie.

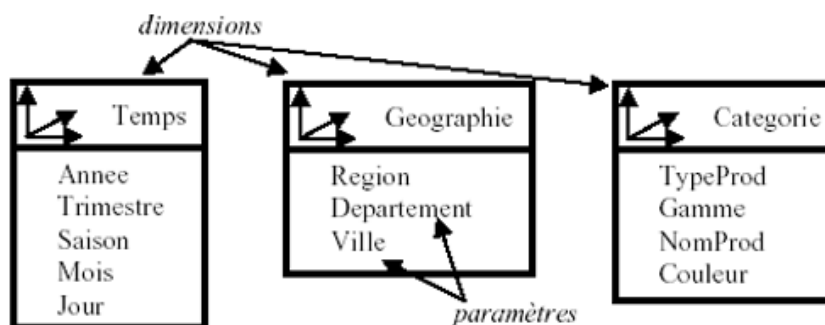


Figure 1.3: table de dimension temps [3]

2. Les cube

Un cube de données (hypercube) est une vue multidimensionnelle où chaque cellule représente un fait selon des axes d'analyse. Ces derniers correspondent aux dimensions du cube. Le fait est observé par une mesure ou plus, à laquelle est associée une fonction d'agrégation (SUM, AVG, MAX, MIN, COUNT, ...).

Il organise les données en une ou plusieurs dimensions qui déterminent une mesure d'intérêt. Une dimension spécifie la manière dont on observe les données pour les analyser, alors qu'une mesure est un objet d'analyse. Chaque dimension est formée par un ensemble d'attributs et chaque attribut peut prendre différentes valeurs.

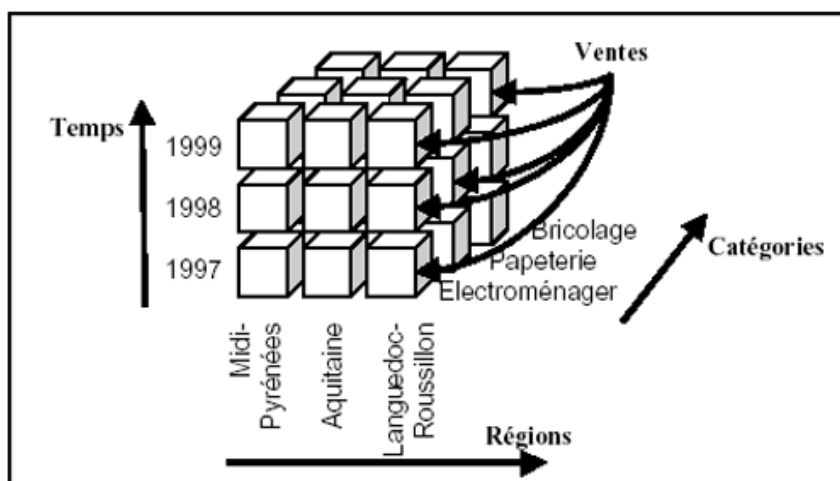


Figure 1.4: Représentation multidimensionnelle (cube) [3]

Représentation ED

1. En étoile

Dans un tel schéma, les mesures sont regroupées dans un seul fait relié à plusieurs dimensions regroupant les paramètres de l'analyse.

L'avantage de cette modélisation est de formaliser une hiérarchie au sein d'une dimension. Par contre, la modélisation en flocon induit une dénormalisation des dimensions générant une plus grande complexité en termes de lisibilité et de gestion.

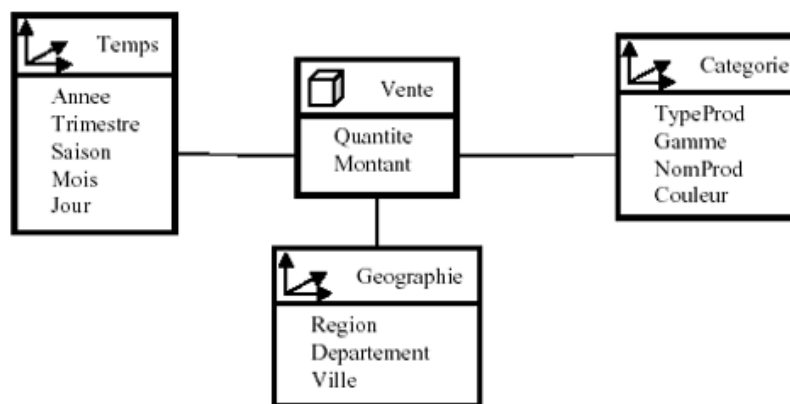


Figure 1.5: EXEMPLE D'UN SCHEMA EN ETOILE [3]

2. En constellation

Ce schéma est une extension du schéma en étoile. Il consiste à fusionner plusieurs schémas en étoile qui utilisent des dimensions communes.

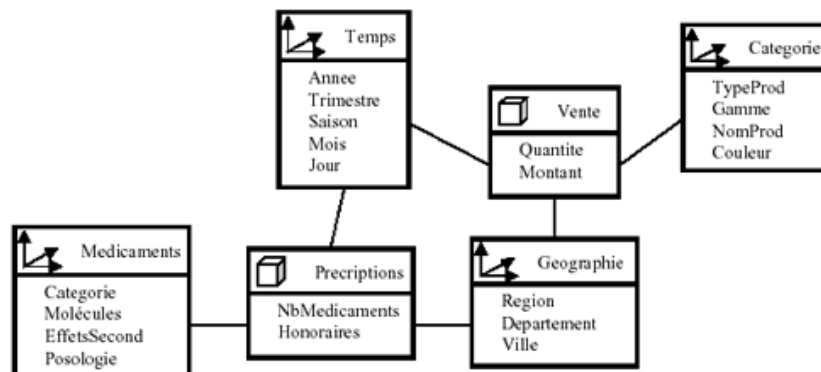


Figure 1.6: Schéma en constellation [3]

Modélisation logique

Il existe des approches pour construire des systèmes basés sur le modèle multidimensionnel. L'approche MOLAP (Multidimensional OLAP) implémente le cube de données dans un tableau multidimensionnel. Par contre, l'approche ROLAP (Relational OLAP) utilise un SGBD relationnel pour gérer et stocker le cube de données.

1. Approche MOLAP

Les systèmes de type MOLAP (Multidimensional OLAP) stockent les données dans un SGBD multidimensionnel sous la forme d'un tableau multidimensionnel, le serveur MOLAP extrait les données de l'hypercube et les présente directement au module client. Chaque dimension du tableau est associée à une dimension du cube.

2. Approche ROLAP

Les systèmes de type ROLAP utilisent un SGBD relationnel pour stocker les données de l'entrepôt. Dans ces systèmes, la table de fait est traduite en une relation et chaque dimension en une ou plusieurs relations selon qu'elle soit normalisée ou pas. Si les tables de dimension sont normalisées, chaque dimension est traduite par plusieurs tables, et le modèle résultant est dit en flocon (snowflake).. Le moteur ROLAP 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 de requêtes SQL mieux adaptées au schéma relationnel, qui profitent des vues matérialisées existantes pour exécuter efficacement ces requêtes.

3. Approche Hybrid OLAP

L'implémentation H-OLAP ("Hybrid OLAP" ou OLAP hybride) constitue un croisement entre ROLAP et MOLAP. Un serveur H-OLAP accède à deux bases de données différentes, multidimensionnelle qui contient les données agrégées et relationnelle, contient les données détaillées. L'implémentation H-OLAP hérite des caractéristiques des implémentations R-OLAP et M-OLAP selon le type de données manipulées par les requêtes, R-OLAP dans le cas de données détaillées et M-OLAP dans le cas de données agrégées. Cette implémentation est intéressante lorsque la majeure partie des requêtes manipulent des données agrégées (l'accès aux données dans

M-OLAP est peu coûteux) et que les données détaillées représentent un volume important (les SGBDR permettent de traiter un volume considérable de données). En général, dans cette approche, les données consultées fréquemment sont matérialisées sous forme multidimensionnelle pour un accès rapide, et les autres données sont laissées dans la base relationnelle pour être extraites directement lors des requêtes.

Opération sur les cubes

Opération Drill down | Drill Up

Les opérateurs de forage vers le haut et de forage vers le bas permettent soit de généraliser l'analyse, soit de l'affiner en modifiant le paramètre utilisé pour définir les valeurs d'une arête du cube. En effet, les dimensions sont associées à des hiérarchies ; ces deux opérateurs permettent, respectivement, de « monter » ou de « descendre » dans une hiérarchie (de la plus résumée jusqu'à la plus détaillée).

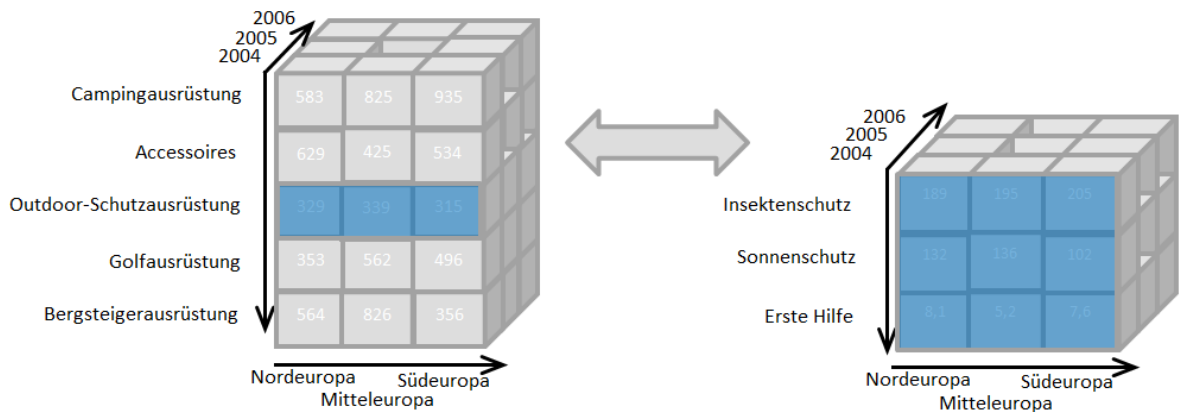


Figure 1.7: Example $Drill_{Up}Drill_{Down}$ (source : wikipedia)

Opération Slice

L'opérateur Slice permet de sélectionner un sous ensemble du cube, selon une ou plusieurs valeurs d'une dimension particulière, la création d'un nouveau cube à une dimension moins.

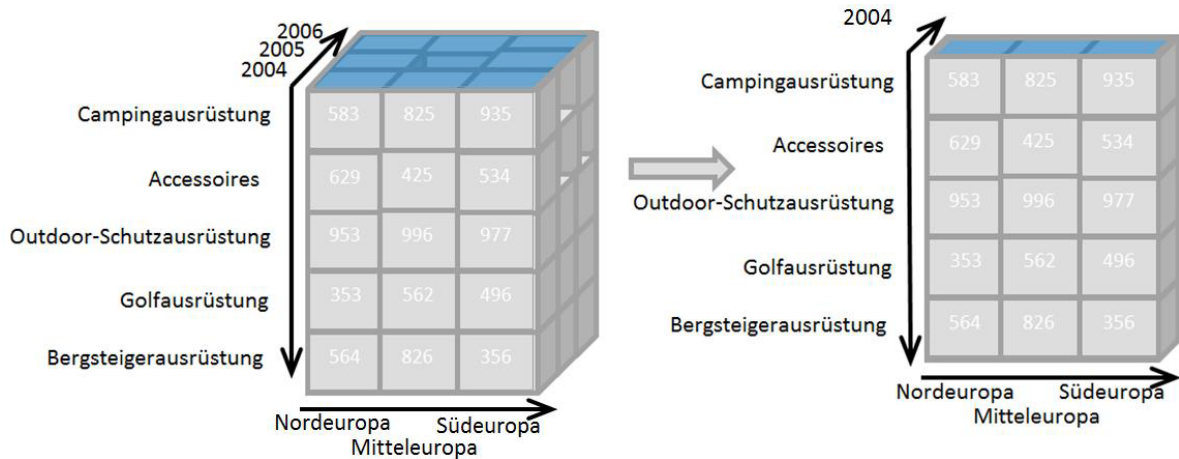


Figure 1.8: Slice (source : wikipedia)

Opération Dice

Permet de faire une projection du cube Le nouveau cube montre les chiffres d'un nombre limité de catégories de produits de vente.

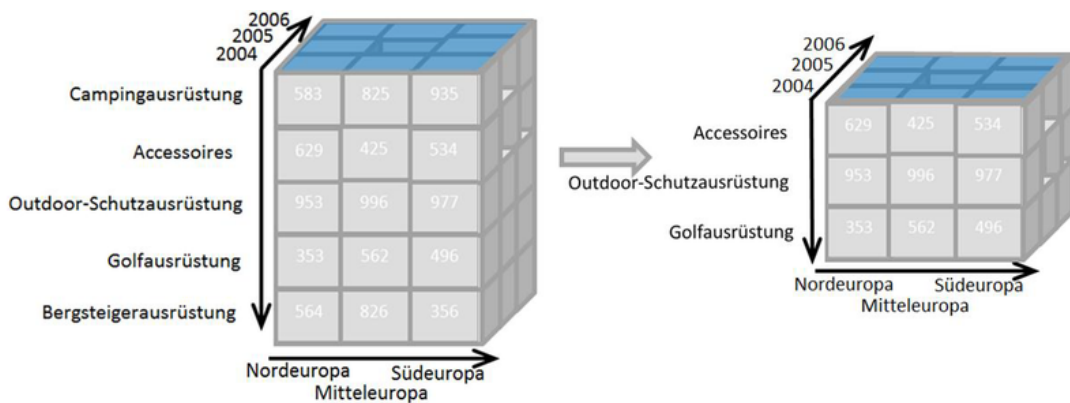


Figure 1.9: Example Dice source : wikipedia

Opération Pivot

permet à un analyste pour faire tourner le cube dans l'espace pour voir ses différents visages. Par exemple, les villes pourraient être disposés verticalement et horizontalement produits lors de la visualisation des données pour un trimestre donné.

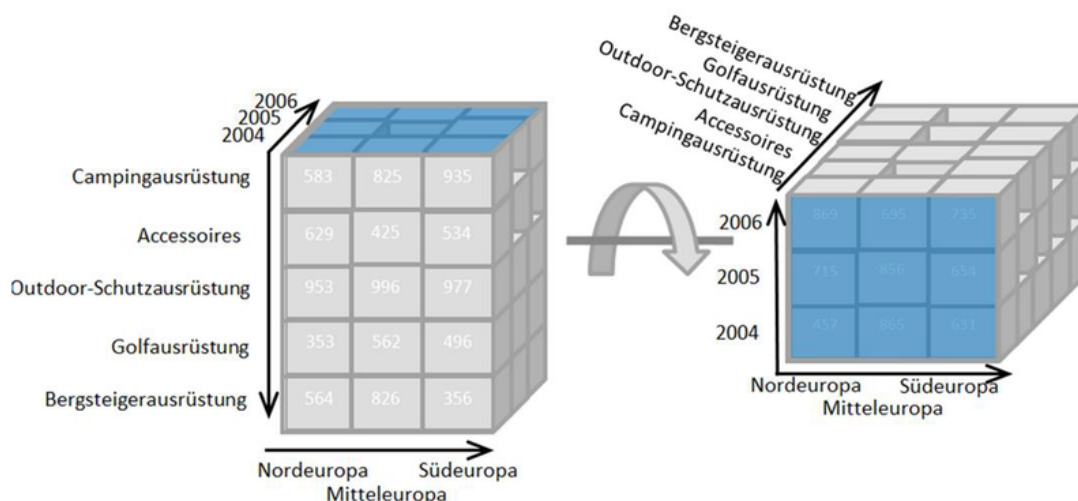


Figure 1.10: Example Pivot source : wikipedia

Il existe autre opérations :

- * Opération Roll Up : Implique le calcul de l'ensemble des relations de données pour une ou plusieurs dimensions. Pour ce faire, une relation ou une formule de calcul peuvent être définis.
- * Des opérations de transformation permettent l'ajout d'attributs de dimension en tant qu'indicateur d'analyse (push) ou de convertir un indicateur d'analyse en parametre (pull).
- * Les opérations d'ordonnancement permettent de changer l'ordre des valeurs (positions) des paramètres des dimensions (switch) ou de réordonner les paramètres d'une hiérarchie (nest). Par généralisation, cette dernière permet d'imbriquer un attribut dans une autre hiérarchie.
- * Certains auteurs proposent aussi l'emploi des opérations binaires ensemblistes (union, différence et intersection) qui nécessitent une très forte compatibilité entre les deux structures multidimensionnelles manipulées. Certains travaux ont aussi proposé la

notion de jointure inspirée de la jointure relationnelle, mais d'un intérêt limité dans un environnement multidimensionnel.

Règles de base OLAP

- **Transparence** : l'utilisateur doit pouvoir accéder à la base, sans se préoccuper de l'emplacement du serveur.
- **Accessibilité** : les données doivent toutes être accessibles, sans ambiguïté.
- **Manipulation des données** : La navigation doit pouvoir s'effectuer simplement et intuitivement à travers des interfaces graphiques.
- **Souplesse d'affichage** : le système doit pouvoir retourner les résultats des requêtes sous une forme graphique.
- **Multidimensionalité** : le système doit permettre une manipulation multidimensionnelle de l'ensemble des données de l'entrepôt.
- **Client-serveur** : architecture du système.
- **Multi-utilisateur** : l'accès et les recherches simultanés de la base doivent être possibles.
- **Accès stable** : le nombre de dimensions et/ou le nombre de niveaux d'agrégation doivent pouvoir changer sans perturber le fonctionnement de l'entrepôt.
- **Gestion des matrices creuses** : Les cellules " vides " d'un hypercube doivent être gérées de façon efficace afin de limiter les capacités de stockage nécessaire et les temps d'accès aux données de l'hypercube.
- **Croisement des dimensions** : le système permet d'effectuer des opérations entre et dans les dimensions.
- **Analyse sans limite** : le nombre de dimensions et de niveaux d'agrégation permettent des analyses complexes.

1.4 Les données temporelles

Habituellement, dans les bases de données, l'évolution d'une entité du monde réel se traduit au niveau de l'instance qui la représente par une modification de la valeur (cf. (Date, 82), concept de base de données opérationnelle). L'entité reste décrite par le même schéma, seule la valeur de l'instance correspondante change. Les données sources sont issues essentiellement des systèmes opérationnels (OLTP), ces données sont peu ou pas historiées : Certaines évolutions ne sont pas conservées, la période de conservation des données temporelles est courte et les dates sont stockées sous forme d'un simple attribut. Dans le contexte des entrepôts de données, de nombreux problèmes liés à la représentation du temps se posent, et il est nécessaire de concevoir des moyens permettant de manipuler simplement le temps.

1.4.1 Définition

« Une base de données temporelle est une base de données avec des aspects de temps intégrés, c'est-à-dire un modèle de données temporel et une version temporelle du langage structuré de requêtes (StructuredQueryLanguage-SQL). Plus précisément, les aspects temporels contiennent habituellement le temps valide et le temps transaction. Ces attributs marchent ensemble pour former une donnée bitemporale, offrent les concepts et les fonctionnalités permettant aux applications de dater les informations et de gérer l'histoire de leurs évolutions. »

Le temps valide dénote la période de temps durant laquelle un fait est vrai par rapport à la réalité. Le temps transaction est la période de temps pendant laquelle un fait est stocké dans la base de données. Dans les bases de données temporelles, les évolutions des valeurs sont toutes conservées. Ceci induit l'inconvénient majeur de générer des volumes de données très importants, altérant rapidement le fonctionnement de la base. Des données temporelles doivent alors être soit supprimées de la base ou bien archivées. La donnée bitemporale combine à la fois le temps valide et le temps transaction.

L'utilisation d'un SGBD temporel permet des schémas plus simples des requêtes plus simples, des codes des applications moins complexes, plus faciles à produire, à corriger, à maintenir, de meilleures performances.

1.4.2 Quelque approche temporelle

Différentes solutions ont été proposées pour représenter l'évolution temporelle. Une première solution est celle des bases de données temporelles (ou historiques), les différentes valeurs prises par un attribut sont conservées, soit systématiquement pour les bases de données historiques, soit selon une fréquence temporelle prédéfinie ou définie par l'utilisateur. Une autre solution est l'approche gestion de versions. Pour chaque entité du monde réel représentée, on conserve différentes valeurs prises par l'instance qui la décrit. Ces valeurs sont conservées sur décision de l'utilisateur ou en fonction de règles qu'il définit.

Approche Jensen

Affecter un intervalle de temps à chaque modification de la valeur d'un attribut de type [valeur,date]. Cet nouvel état contient une valeur pour chacun des attributs. D'un état à l'autre, les attributs n'ayant pas été modifiés ont la même valeur. Pour chaque n-uplet, chaque valeur prise par un attribut est conservée et associée à la date, ou la période à laquelle elle correspond. Une entité est donc décrite, pour chaque attribut, par un ensemble de couples (valeur, date). La période associée à la valeur d'un attribut est en général la période de validité de la valeur associée. La période de validité d'une valeur est l'intervalle pendant lequel le fait décrit est vrai dans le monde réel modélisé [4].

Matricule	Nom	Valid time	Address	valid times
1000	Durand	(1/80 - a)	Paris	(1/80 -a)
1000	Durand	(1/80 - a)	Lyon	(7/82 -a)

Table 1.2: Modification de la valeur d'un attribut [4]

Approche Hubert

Associer un intervalle de temps aux n-uplets. Chaque changement de la valeur est conservé. La valeur qu'il possédait précédemment est associée à un intervalle de temps borné et la nouvelle valeur à un intervalle de temps non borné, Chaque valeur prise par un n-uplet est associée à une date ou une période. Un n-uplet représente donc une même entité du monde réel durant une période donnée du monde réel modélisé. L'intégration du temps dans une relation est spécifiée lors de la définition du schéma. [4]

Matricule	Nom	Address	From	TO
1000	Durand	Paris	1-1-80 10:00AM	7-31-82 5:00PM
1000	Durand	Lyon	7-31-82 5:00PM	forever

Table 1.3: Modification n_{uplets} [4]

Une autre approche qui vise à permettre l'historisation de versions de structure et la comparaison des données. L'approche proposé est basé sur l'ajout d'intervalles de validité est crucial à la prise en compte des évolutions (rajouter des extensions temporelles, sous forme d'intervalles de validité délimités par des dates de début et de fin de validité, aux éléments du schéma multidimensionnel). Spécifiquement l'ajoute d'une table de fait, construite à partir d'un schéma multidimensionnel temporel.

Des approche Multidimensionnel

« Un schéma multidimensionnel temporel, est représenté par un ensemble de dimensions temporelles, une dimension Temps, un ensemble de relations de mapping et une table de fait. Et une version de structure est une version d'un schéma multidimensionnel temporel TDM valide pendant un intervalle de temps. » Il existe deux types des évolutions au niveau multidimensionnel :

- Evolution des données : insertion/mise à jour/suppression d'enregistrements.
- Evolution des schémas : le schéma multidimensionnel peut évoluer en réponse à de nouveaux besoins d'analyse ou encore à l'évolution naturelle des éléments analysés.

Approche Mathurin, Maryvonne, Yvan et Tchounikine

Selon Mathurin Body, Maryvonne Miquel, Yvan Bédard et Anne Tchounikine la dernière version ne permet pas de voir l'évolution des données circuler pour l'analyse, il peut même conduire à des fausses conclusions car il peut être besoin des données n'est pas pris en compte. Donc ils proposent une approche qui permettre l'historisation de versions de structure et la comparaison des données.[5]

L'approche proposer est basée sur une table de fait multi version construite à partir d'un schéma multidimensionnel temporel. Le modèle est basé sur la redéfinition des tables de fait et de dimensions par l'ajout d'intervalles de validité. Cette approche est essentiellement bâtie sur une unique table de faits multi-versions.

Pour implémenter le modèle proposé dans les outils OLAP existants, les auteurs ont choisi une architecture constituée de trois parties :

- Un entrepôt de données temporel : qui contient le schéma multidimensionnel temporel et ses méta-données, y compris les relations de mapping.
- Un entrepôt de données multi-versions : dans lequel se trouve la dimension « mode de représentation temporel » et la table de fait multi-versions.
- Un cube OLAP : construit à partir de l'entrepôt de données multi-versions, en utilisant des agrégations, et qui permet aux requêtes d'intégrer les concepts de modes de représentation.

Mathurin Body, Maryvonne Miquel, Yvan Bédard et Anne Tchounikine présentent plusieurs adaptations à leur approche pour une meilleure implémentation de ce modèle conceptuel (l'associer à un schéma en étoile).

Approche Chamoni et Stock

Selon les travaux de Peter Chamoni, Steffen Stock proposent l'ajout d'intervalles de validité est crucial à la prise en compte des évolutions, rajouter des extensions temporelles, sous forme d'intervalles de validité délimités par des dates de début et de fin de validité, aux éléments du schéma multidimensionnel. [6]

Approche Kimball

Ralph Kimball propose trois façons de gérer les évolutions. La première ne garde pas trace de l'historique des évolutions, la seconde crée un nouvel enregistrement pour la donnée modifiée, et la troisième consiste à ajouter une colonne pour garder trace de l'ancienne valeur ainsi que la date du changement.

Approche gestion des versions

Les modèles de gestion de versions sont une solution permettant de gérer les évolutions au niveau des bases de données en gardant les versions antérieures et postérieures aux évolutions. [7]

Concernant la gestion des évolutions d'instances, trois grandes approches se sont distinguées . La première concerne les modèles de gestion de versions d'attributs et s'inspire des études effectuées dans les bases de données temporelles. La seconde approche propose la gestion des versions d'objets et la conservation des instances de l'objet et non des attributs séparément. Quant à la dernière approche, elle préconise la gestion de versions de contextes. Le principe étant de regrouper des versions d'entités selon une signification précise.

L'évolution du schéma de la base de données a également été abordée par l'approche des versions . Cela consiste à conserver les schémas, avant et après modification, en sachant que toute modification donne lieu à une nouvelle version et que toutes les versions sont conservées.

Dans les bases de données temporelles, les évolutions des valeurs sont toutes conservées. Ceci induit l'inconvénient majeur de générer des volumes de données très importants, altérant rapidement le fonctionnement de la base. Des données temporelles doivent alors être soit supprimées de la base ou bien archivées.

1.5 Classification des requêtes

Pour poser sa requête, l'utilisateur peut spécifier une version d'étoile de son choix, ou bien, travailler, par défaut, sur la version en cours (la plus récente). Une requête qui ne porte pas sur un fait, mais seulement sur une dimension, est appelée requête simple par opposition à requête multidimensionnelle. Une requête qui ne porte que sur une seule version est appelée « requête mono-version ».

Toutefois, pour répondre à une requête, il est parfois nécessaire d'interroger d'autres versions en plus de celle choisie, auquel cas la requête est considérée comme étant temporelle multi-versions.

1. **Requête simple:** Une requête simple est une requête qui porte sur les différentes versions d'une dimension. La réponse à ces requêtes nécessite seulement l'interrogation des dimensions.
2. **Requête multidimensionnelle :** Les requêtes multidimensionnelles peuvent être classées en deux catégories, requêtes mono-version et requêtes multi-versions.
 - **Requête mono-version :** Une requête mono-version est une requête qui ne concerne qu'une seule version d'étoile. La réponse à une telle requête ne nécessite l'interrogation que de la version d'étoile sur laquelle l'utilisateur a posé sa requête.
 - **Requête multi-versions :** Une requête est qualifiée de multi-versions lorsqu'elle interroge plusieurs dimensions. Ce type correspond aux requêtes impliquant une période de temps plus large que l'intervalle de validité de la version du schéma interrogée. Toutes les dimensions valides durant cette période, induites par des évolutions de structures du schéma, sont interrogées. Dans ce cas, le système se charge de retrouver les versions à interroger, selon l'intervalle de temps spécifié dans la requête. L'utilisateur peut aussi préciser explicitement l'ensemble des versions qu'il veut interroger.

1.6 Domaines d'application

- * Domaine bancaire.
- * Domaine des télécommunications
- * Domaines de l'assurance , de pharmacie et de santé, ...

1.7 Logiciels

Les outils OLAP se concentrent sur l'analyse multidimensionnelle en proposant des objets graphiques (tableaux croisés,...). Ces outils utilisent les données de l'entrepôt ou des magasins, mais ne permettent pas leur construction.

1.7.1 Mondrian

Mondrian est un serveur OLAP écrit en Java. Il autorise une analyse interactive très large des données stockées dans une base de données SQL sans avoir à écrire de code SQL.

- Type de Cube : R-OLAP.
- Point fort : la capacité.
- Projet âgé de 6 ans.
- Très bonne documentation.
- Distribué sous Licence CPL.
- Sponsorisé par Pentaho, 1 administrateur et 22 développeurs ont clairement étaient identifiés.
- Taux de fréquentation et visibilité Internet très bon.
- Taille de la communauté difficile à déterminer car reliée directement à la suite décisionnelle Pentaho.
- Pas d'interface graphique.
- faible niveau de packaging.

- Ne dispose pas d'une traduction Française.
- Plutôt bon.

1.7.2 Palo

Palo est un serveur multidimensionnel Le système opère en temps réel et supporte la consolidation hiérarchique comme de nombreux outils de Business Intelligence.

- Type de Cube : M-OLAP.
- Point fort : la performance.
- Projet âgé de 2 ans.
- Très mauvaise documentation.
- Distribué sous Licence GPL.
- Sponsorisé par Jedox AG, 1 administrateur et 11 développeurs ont clairement étaient identifiés.
- Taux de fréquentation est assez faible.
- Taille visibilité Internet relativement bonne.
- Interface graphique.
- Bon niveau de packaging.
- Ne dispose pas d'une traduction Française.
- Difficile à déterminer du fait de son jeune âge.

1.7.3 FreeAnalysis

La plate-forme FreeAnalysis fournit aussi bien un accès Java qu'un accès Web aux données multidimensionnelles, tout en fournissant un grand nombre d'applications pour gérer les projets OLAP, de la définition du Cube à son déploiement.

- Type de Cube : R-OLAP et M-OLAP.

- Projet âgé de 3 ans.
- Très mauvaise documentation.
- Distribué sous Licence MPL 1.1.
- Sponsorisé par BPM Conseil, aucun contributeur clairement identifié.
- Taille, taux de fréquentation et visibilité Internet assez faible.
- interface graphique.
- Faible niveau de packaging.
- Dispose d'une traduction Française.
- Très bon taux d'activité.

1.8 conclusion

Nous avons défini, à travers cette première chapitre du mémoire, les principaux concepts qui nous intéressent dans le cadre de notre projet. Le premier chapitre nous a permis de présenter le contexte de nos travaux qui est le système décisionnel, ainsi que les concepts de magasin de données et de modélisation multidimensionnelle et la notion temporel.

Chapter 2

Etat de l'art

2.1 Introduction

Typiquement, les visualisations attendent des données sous une forme spécifique. Les données dans la source de données peuvent ne pas correspondre exactement à cette structure. Par exemple, la source de données peut avoir plus de deux colonnes, ou l'ordre des colonnes peut ne pas correspondre à l'ordre attendu.

Le langage de requête fournit la capacité d'envoyer manipulation de données et des demandes de mise en forme de la source de données, et faire en sorte que la structure et le contenu de données renvoyé correspond à la structure attendue. L'idée initiale de langage de requête est introduite par Ted Codd dans son article séminal sur le modèle relationnel, c'est l'idée de la différence entre les données et leur implémentation : le langage de requête doit exprimer le quoi et non le comment, c'est à dire ce que je cherche et non la façon dont il doit être calculé.

Codd introduit donc un langage, basé sur le calcul des prédicats pour interroger une base de données relationnelle (donc sous forme de tables). Il introduit aussi une autre façon d'interroger, via une algèbre et montre l'équivalence entre les deux. Le projet de Codd, alors employé d'IBM deviendra un projet de recherche à forte visibilité, System R. Le projet de recherche donnera lieu à un produit, DB2.

2.2 Langage d'interrogation

Langage de commande spécialisé pour l'interrogation de base de données et permettant à l'utilisateur de formuler des questions et de recevoir des réponses de la part du système interrogé. Les langages d'interrogation utilisent des syntaxes et des modes de dialogue différents (sélection de termes dans une liste de termes proposés, mots simples combinés, utilisation d'opérateurs de recherche, langage naturel, etc.).

Ci-dessous, nous passons en revue sur ces langages:

2.2.1 OQL (Object Query Language)

Langage d'interrogation de bases de données objets proposé par l'ODMG, basé sur des requêtes SELECT proches de celles de SQL. Le langage est construit de manière fonctionnelle. Une question est une expression fonctionnelle qui s'applique sur un littéral, un objet ou une collection d'objets, et retourne un littéral, un objet ou une collection d'objets.

Les expressions fonctionnelles peuvent être composées afin de constituer des requêtes plus complexes. Le langage est aussi fortement typé, chaque expression ayant un type qui peut être dérivé de la structure de l'expression et du schéma de la base. La correction de la composition des types doit être vérifiée par le compilateur OQL. Du point de vue du typage, OQL est peu permissif, et ne fait guère de conversions automatiques de types, à la différence de SQL. Syntaxiquement, beaucoup de questions sont correctes, mais erronées du point de vue typage. OQL n'est pas seulement un langage d'interrogation, mais bien un langage de requêtes avec mises à jour. En effet, il est possible de créer des objets et d'invoquer des méthodes mettant à jour la base.

OQL permet aussi la navigation via les objets liés de la base, mais seulement avec des expressions de chemins mono values.

Quelle est la liste de tous les rapports achetés par John ?

```
(Q) SELECT R.ID FROM R IN RAPPORT, P IN R.ESTACHETE
      WHERE P.NOM = "JOHN"
      (R) BAG<ID: STRING>
```

2.2.2 SQL (sigle de StructuredQueryLanguage)

SQL est un langage commercial il permet d'interroger des données relationnelles [ISO99]. SQL est devenu un standard des bases de données relationnelles (en 1987, normalisation de ce langage par ANSI). SQL s'utilise sous deux formes, soit d'une manière interactive, soit à l'intérieur d'un langage hôte (C, fortran, cobol...).

SQL ne comporte qu'une vingtaine d'instructions, il est dit procédural (l'accès aux données se fait par leur contenu et non par leur chemin). SQL a 4 fonctions nécessaires :

1. La description de donnée : permet de définir ou supprimer une base de données ou une relation (Verbes CREATE, DROP).
2. Le contrôle de l'accès au donnée (Verbes GRANT , REVOKE).
3. La manipulation des données : suppression, modification, insertion dans une relation (Verbes DELETE, UPDATE, INCERT).
4. L'interrogation des données (SELECT). Version complet dans les gros systèmes MySQL, Oracle, D2B, INFORMIX, Et version moins complet dans les SGBD Micro comme ACCESS, OpenOffice,....

Une requête SQL a la syntaxe suivante :

```
SELECT <liste de champs>
FROM <liste de tables>
WHERE <condition>
```

2.2.3 MDX

MDX : est l'acronyme de Multi Dimensional eXpression, est un langage de requêtes OLAP pour les bases de données multidimensionnelles. A été inventé par Mosha Pasumansky au sein de Microsoft a été présenté pour la première fois en 1997 comme un volet de la spécification OLE DB for OLAP (ODBO).version commerciale Microsoft OLAP Services 7.0, Microsoft Analysis Services en1998 dernière version de la spécification OLE DB for OLAP (ODBO) en 1999.

MDX peut être présenté comme une extension de SQL : structures de requêtes et mots clés similaires, mais avec de grandes différences. MDX est fait pour naviguer dans les

bases multidimensionnelles et pour définir des requêtes sur tous leurs objets (dimensions, hiérarchies, niveaux, membres et cellules) afin d'obtenir (simplement) une représentation sous forme de tableaux croisés

Comparaison entre MDX et SQL

Multidimensionnel (MDX)	Relationnel (SQL)
Cube	Table
Dimension	Plusieurs colonnes liées
Niveau(Levels)	Colonne (chaîne de caractère ou valeur numérique)
Mesure(Mesure)	Colonne (discrète ou numérique)
Membre de dimension (Dimension membre)	Valeur dans une colonne et une ligne particulière de la table

Table 2.1: Comparaison entre MDX et SQL

Structure générale d'une requête MDX

```
SELECT [<axis_specification > [, <specification_des_axes > ...]]
      FROM [<specification_du_cube >]
      WHERE [<specification_de_filtre >]
```

- FROM : spécifier la source de donnée (cube).
- SELECT : les résultats attendu par le requête.
 - * nb quelconque de dimensions pour former les résultats de la requête.
 - * terme d'axe pour éviter confusion avec les dimensions du cube.
 - * pas de signification particulière pour les rows et les columns, mais il faut définir chaque axe : axe1 définit l'axe horizontal et axe2 définit l'axe vertical.

Quelques définitions

Membre: c'est une instance d'un niveau d'une dimension, spécifié entre crochets [...].

Tuple: c'est une suite de plusieurs membres entre parenthèses séparés par une virgule.

Cellule: est l'intersection des dimensions d'un cube de données.

Set: c'est un ensemble ordonné de tuples.

2.3 Les langages d'interrogation temporels

De nombreux travaux ont traité de l'extension des langages pour prendre en compte les aspects temporels. De nombreuses extensions ont été apportées aux langages relationnels notamment au langage SQL. Le langage TSQL2 est en cours de normalisation.

Les langages temporels sont essentiellement basés sur l'ajout d'une clause temporelle (clause WHEN) au niveau des requêtes. Des conditions de sélection basées sur l'utilisation d'opérateurs temporels sont spécifiées dans cette clause. Les conditions temporelles portent sur la comparaison de valeurs de types temporelles et plus particulièrement celle de la période de validité des n-uplets. La période de validité des n uplets est également utilisée au niveau de l'affichage du résultat d'une requête.

Exemple:

Question 1 : "A quel âge Durand était-il stagiaire ?"

```
      Select Age
      From R1, R3
      Where R1.Matricule = R3.Matricule and R1.Nom = "Durand"
            and R3.Employ = "stagier"
      When R1.Interval overlap R3.Interval
```

La clause When permet dans ces deux requêtes de mettre en correspondance dans R1 et R3 les n-uplets qui décrivent une personne pour la même période c'est-à-dire les n-uplets pour lesquels les périodes de validité se chevauchent (overlap).

2.3.1 langages pour les données multidimensionnelles

La représentation multidimensionnelle des données induit de nouveaux langages constitués de nouvelles opérations liées à l'aspect "analyse". Ces opérations peuvent être classées selon le type de langage.

Types de langages

Les langages dédiés aux bases de données multidimensionnelles sont classés en trois types [8] à savoir :

- **Le langage de Définition des Données (LDD)** : ce langage définit les opérateurs nécessaires à la définition d'un schéma multidimensionnel (définition des faits, dimensions et hiérarchies) ainsi qu'à la modification de ses éléments ou encore leur suppression.
- **Le langage de Manipulation des Données (LMD)** : ce langage définit les opérations nécessaires à la mise à jour ainsi qu'à la consultation des données.
- **Le langage de Contrôle des Données (LCD)** : Ce langage définit les opérations nécessaires à l'affectation et à la suppression des droits des utilisateurs sur les éléments du schéma multidimensionnel. Les langages graphiques présentent l'avantage de visualiser le schéma de la base avec les différents liens entre les classes, tandis que les langages tabulaires présentent un apport visuel limité et que les langages iconiques font mal apparaître les différents liens entre les classes.

2.4 langages graphiques

Quelques travaux de recherche, ainsi que des outils commerciaux, ont proposés des interfaces graphiques dédiées au multidimensionnel qui facilite au utilisateur la manipulation des BDM et prendre des bon décisions à partir resultatats d'analyse.

2.4.1 Langage FRED

Une approche se distingue en proposant un processus permettant la construction du système d'aide à la décision. Ce processus se décompose en trois phases permettant d'isoler les différents problèmes à résoudre : une phase d'intégration des sources de données, une phase de construction de l'entrepôt et une phase de réorganisation des données pour construire les magasins. [3]

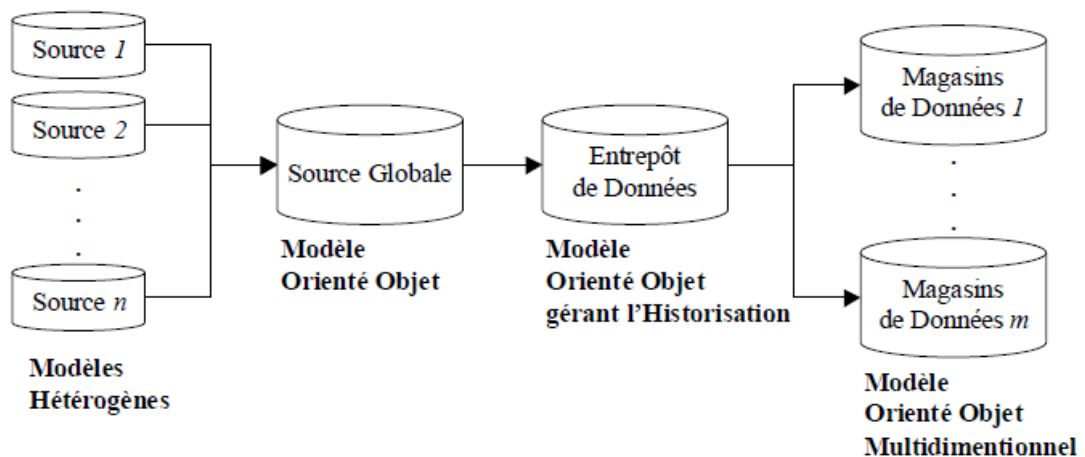


Figure 2.1: Architecture de systèmes d'aide à la décision Fred [3]

Une approche qui proposant des interfaces graphiques qui aident l'administrateur lors du processus de construction de l'entrepôt et des magasins de données.

Il présente une interface de construction de l'entrepôt de données et de construction des magasins de donnée, la 1er basée sur un modèle conceptuel orienté objet et la 2eme permettant de réorganiser les données de l'entrepôt selon un modèle orienté objet multidimensionnel.

2.4.2 Langage COMET

Johann Eder et Christian Koncilia proposent ce langage dans l'article un modèle de données multidimensionnel temporel appelé «COMET» qui permet de supporter l'évolution des données et schémas multidimensionnels. [9] Les principaux objectifs de ce modèle sont :

1. Entrepôt de données temporel : le modèle COMET permet de garder trace des modifications au niveau instances (données) et au niveau schéma aussi.
2. Constellation de faits : le modèle COMET prend en compte les schémas en constellation, il permet le partage de dimensions.
3. Agrégation proportionnelle : le modèle supporte les agrégations proportionnelles pour permettre une agrégation correcte des instances d'attributs non disjoints (ont plusieurs pères). Par exemple, la 5ème semaine appartient au mois Janvier et Février.
4. Dimensionnalité générique : toutes les dimensions sont considérées équivalentes que se soit dans leur structure ou leurs possibilités opérationnelles, et sont traitées de la même façon. Pour implémenter le modèle COMET, les auteurs utilisent une approche d'implantation d'entrepôt de données temporel qu'ils appellent : « approche indirecte ».

Selon les auteurs le modèle et le prototype d'entrepôt de données temporel présentés supporte les requêtes qui concernent une version de structure particulière mais aussi les requêtes qui portent sur plusieurs versions de structure. Cependant, dans ce dernier cas, des fonctions de conversion doivent être appliquées et engendrent des transformations excessives des données.

Le prototype COMET consiste en trois parties :

- l'entrepôt de données temporel (contient les informations concernant les versions de structures, les mesures et les fonctions de transformation) implémenté avec Oracle 8.1.
- Le transformateur (transpose toutes les valeurs de cellules requises, de toutes les

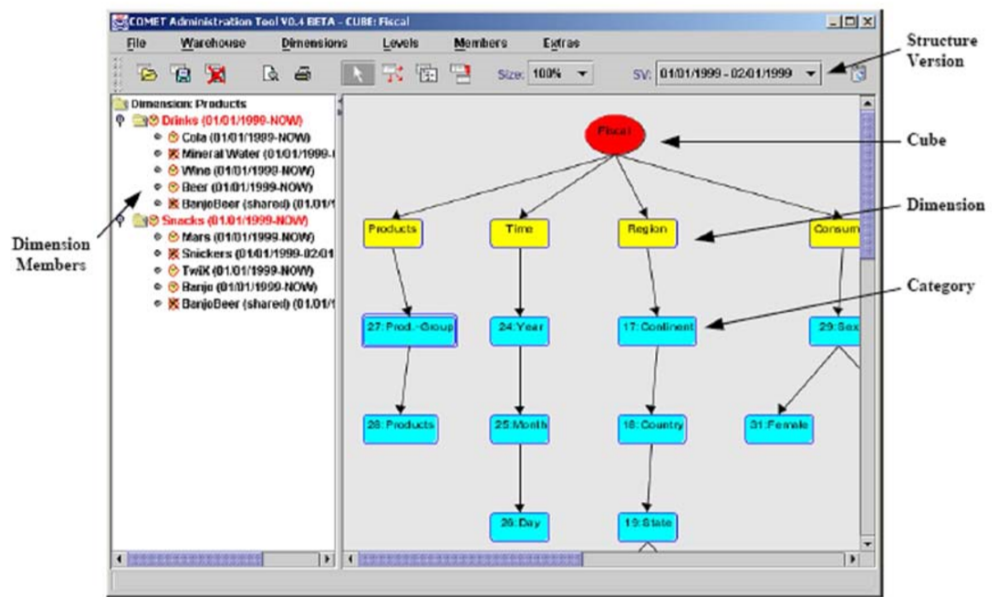


Figure 2.2: L'outil d'administration COMET [9]

versions de structure requises, dans la version de structure choisie, en utilisant les fonctions de transformations définies).

- L'outil d'administration (permet de définir, et de modifier, le schéma et la structure de l'entrepôt de données et d'importer les données des cellules vers l'entrepôt de données temporel).

2.4.3 Langage MVDW

Ce langage est proposé par Morzy et Wrembel ; ce langage proposée vise à gérer les évolutions du schéma et des données par historisations de versions. A chaque changement, un administrateur crée une version de l'entrepôt de données et lui associe un intervalle de temps donné. L'historisation permet de gérer les différents schémas de l'entrepôt et les données qui lui sont associées, et de répondre aux requêtes qui portent sur plusieurs versions.[10]

Un entrepôt de données multi-versions MVDW est composé de l'ensemble de ses versions. Chaque version est composée d'une version de schéma S_i et une version d'instance I_i qui garde les données décrites par S_i . La structure des informations contenues dans l'entrepôt de données est constitué de schémas de dimension et schémas de fait. Alors

qu'une instance de l'entrepôt et constituée d'instances de dimension (instances des niveaux hiérarchiques de la dimension) et d'instances de fait (les valeurs du fait).

Le prototype proposé, les requêtes sont traitées en deux étapes.

- La requête est décomposée en un ensemble de requêtes partielles indépendantes, chacune pour une version.
- Les résultats partiels sont intégrés dans un ensemble commun de données, si possible.

Pour répondre à de telles requêtes, les auteurs proposent une extension du langage SQL traditionnel.

- Interrogation de la version en cours : par défaut, les utilisateurs utilisent les versions mono-version adressées à la version réelle en cours (la dernière). Dans ce cas, du point de vue utilisateur, aucune extension du langage SQL n'est requise. Cependant, du point de vue implantation, il faut retrouver toutes les données qui appartiennent à la version.
- Interrogation d'un ensemble de versions réelles : les requêtes multi-versions dans l'ensemble des versions réelles sont définies en spécifiant l'intervalle de temps qui intéresse l'utilisateur. A cette fin, la commande `SELECT` a été étendue, avec la clause : version de la date « date de début » jusqu'à « date fin », pour sélectionner toutes les versions concernées suivant leurs dates de début et de fin de validité.
- Interrogation d'un ensemble de versions alternatives : comme les versions alternatives ne sont pas ordonnées linéairement dans le temps, l'utilisateur doit préciser explicitement l'ensemble des versions qui l'intéressent. Pour cela, la commande `SELECT` a été étendue avec la clause : version alternative (ver-id / ver-name, ...).
- Interrogation d'une seule version : les requêtes mono-version pour une version réelle ou alternative, sont traitées comme des cas particuliers des deux cas d'interrogation précédents, en spécifiant dans la clause version la date pour laquelle la version est valide ou alors l'identifiant et le nom de la version, selon le cas.
- Fusions de résultats de requêtes partielles : dans quelques cas, il est possible de regrouper les résultats des requêtes partielles (en un seul cube). Cette fusion est

définie en incluant la fusion (merge) dans la clause ver-id / ver-name, où ver-id et ver-name désignent l'identifiant et le nom de la version dont le schéma sera utilisé comme schéma de destination pour tous les résultats des requêtes partielles obtenues.

2.4.4 Langage GRAPHIC-OLAP

Graphic-OLAP est un outil graphique développé en 2001, il repose sur un ensemble d'interfaces utilisateurs permettant de définir, de manipuler, de visualiser et de consulter des bases de données multidimensionnelles structurées en constellation. Il intègre essentiellement deux modes de fonctionnement : textuel au travers du langage OLAP-SQL, et Graphique au travers d'un graphe de visualisation de la constellation. Graphic-OLAP repose sur deux langages : langage assertionnel OLAP-SQL et langage graphique.

Langage graphique: L'outil GRAPHIC-OLAP offre une vision conceptuelle (schéma) de la BDM au travers de diverses interfaces : arborescente (représentation classique), graphique (représentation conforme aux formalismes de notre modèle en constellation), et hyperbolique (représentation dédiée aux grandes constellations). [11]

Architecture

L'architecture de GRAPHIC-OLAP décrite à la figure suivante comporte trois niveaux.

Interface: GRAPHIC-OLAP propose plusieurs type d'interface pour communiquer avec la base de données multidimensionnelle. Le visualiseur interactif qui permet d'afficher la base de données multidimensionnelle en schéma constellation en plus l'interface éditeur de commande OLAP-SQL qui donne la main à l'utilisateur pour écrire ces ordres assertionnels et d'obtenir des réponses. En fin l'interface résultant qui permet d'afficher les résultats d'une commande d'interrogation d'une base de données multidimensionnelles.

Traitements: dans ce niveau il existe 4 composants analyseur, ECOUTEUR EVENEMENT, MOTEUR OLAP/SQL et GENERATEUR GRAPHIQUE. Tous les commandes OLAP-SQL de définition (LDD), de manipulation (LMD) et d'interrogation (LID) reçoivent de l'utilisateur doit être analysée lexicalement et syntaxiquement pour être validée. Après la validation les commandes, ils sont pris en charge par le moteur OLAP/SQL chargé des traductions vers le SGBD relationnel de stockage. L'écouteur d'événements permet de capturer les réactions graphiques de l'utilisateur pour permettre au moteur OLAP/SQL

la traduction des manipulations graphiques effectuées. Les ordres d'interrogations sont calculés et leurs résultats sont mis en forme par le générateur graphique dans une table multidimensionnelle.

Stockage: regroupe deux bases de données. La base de données ROLAP qui stocke les données des faits et des dimensions tandis la Metabase décrit les structures multidimensionnelles de la constellation.

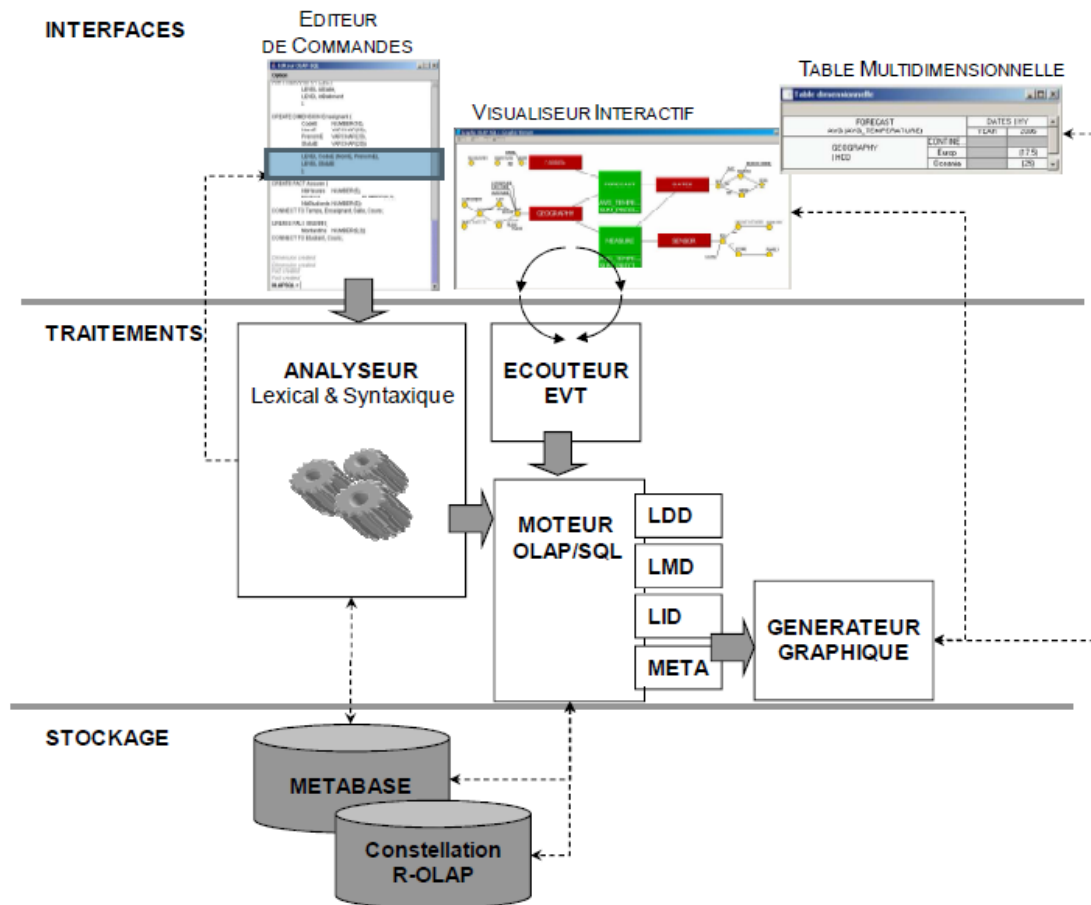


Figure 2.3: architecture de langage GOLAP [11]

2.4.5 Langage XML-GOLAP

XML-GOLAP est un outil graphique développé en 2009, il permet à l'utilisateur de définir, de manipuler, de visualiser et de consulter documents XML au sein d'une base de données multidimensionnelles. [12]

Architecture

Interface: offre un éditeur de commandes OLAP-SQL permettant de saisir des requêtes d'interrogation intégrant les fonctions d'agrégation textuelle qui offrent en sortie une table multidimensionnelle qui contient les données de l'analyse souhaitée.

Traitement: Chaque interrogation OLAP-SQL (LID) est analysée par l'analyseur lexical et syntaxique, puis est prise en charge par le moteur OLAP/SQL effectuant les traductions vers le SGBD relationnel de stockage. L'agrégateur textuel effectue des interactions entre R-OLAP, les documents entreposés et l'ontologie de domaine stockée sous forme de fichiers XML. Une fois la requête d'interrogation valide est calculée, le générateur graphique met les résultats dans une table multidimensionnelle.

Stockage: regroupe deux bases de données. La base de données ROLAP qui stocke les documents XML. Le Metabase qui regroupe la description des structures multidimensionnelles et le système de gestion de fichiers qui contient l'ontologie utilisée pour l'agrégation de texte.

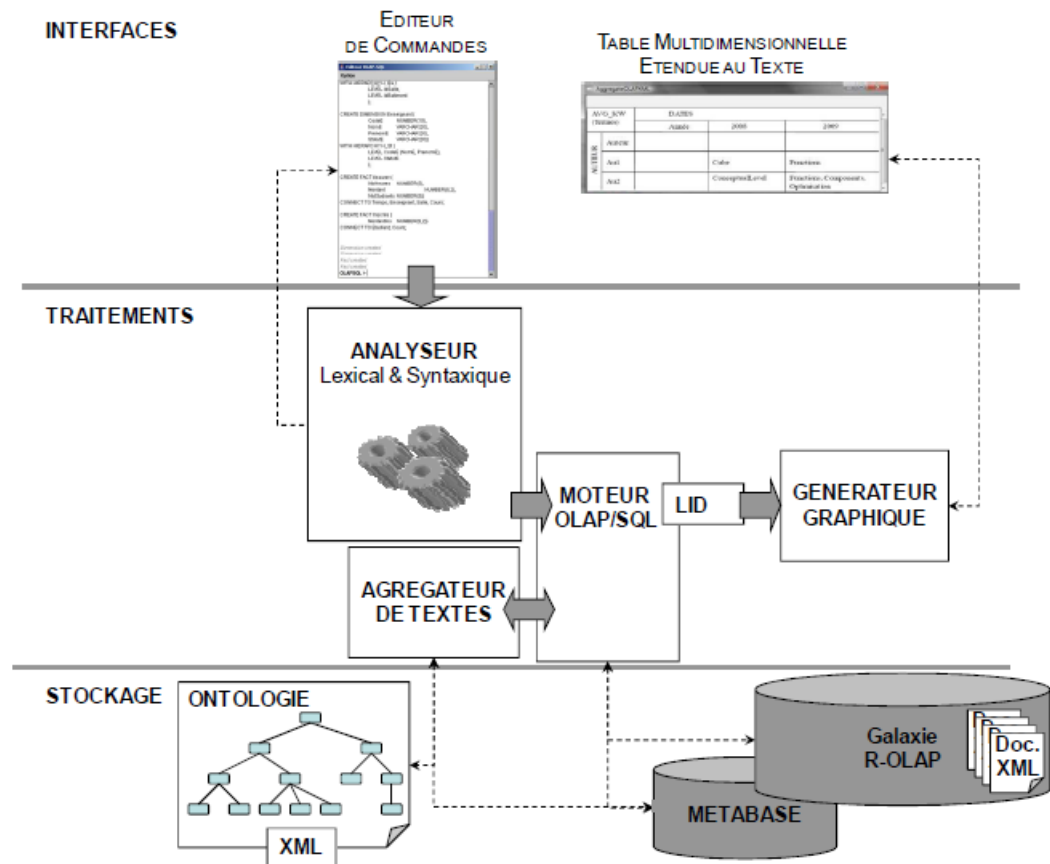


Figure 2.4: Architecture de langage XML GOLAP [12]

2.4.6 Langage GOLAP Personnel

Personnel-GOLAP est un outil graphique développé en 2009, il permet personnaliser une base de données multidimensionnelle en intégrant à la fois l'expertise décisionnelle et les préférences de l'utilisateur. [13][14]

Architecture

Interface: L'éditeur de commandes permet de saisir des requêtes d'interrogation, saisir les règles de performance associé à la base de données multidimensionnelle. Ces règles prouvent être quantitative ou qualitative. Parallèlement, l'éditeur d'annotation permet de créer, modifier, supprimer et modifier des annotations ancrées dans la constellation. Enfin, le system utilise les requête d'interrogation et les règle de performance et les annotations pour produis un tables multidimensionnelles personnalisées qui réponde au besoin.

TRAITEMENTS: regroupe cinq composants. Chaque commande doit être analysée

par l'analyseur lexical et syntaxique pour être validée. Les commande valider doit être chargé par le moteur OLAP/SQL pour être traduisiez vers le SGBD relationnel. Pour les annotations exprimer par l'utilisateur doit gérer par le moteur MED. Une fois la requête d'interrogation valide est calculé, le générateur graphique mis les résultats dans une table multidimensionnelle personnalisées intégrant les annotations associées au contexte.

STOCKAGE: regroupe deux bases de données. La Metabase regroupe la description des structures multidimensionnelles de la constellation ainsi que les préférences contextuelles exprimées par l'utilisateur. La base ROLAP contient les données décisionnelles de la constellation enrichies d'annotations de l'utilisateur.

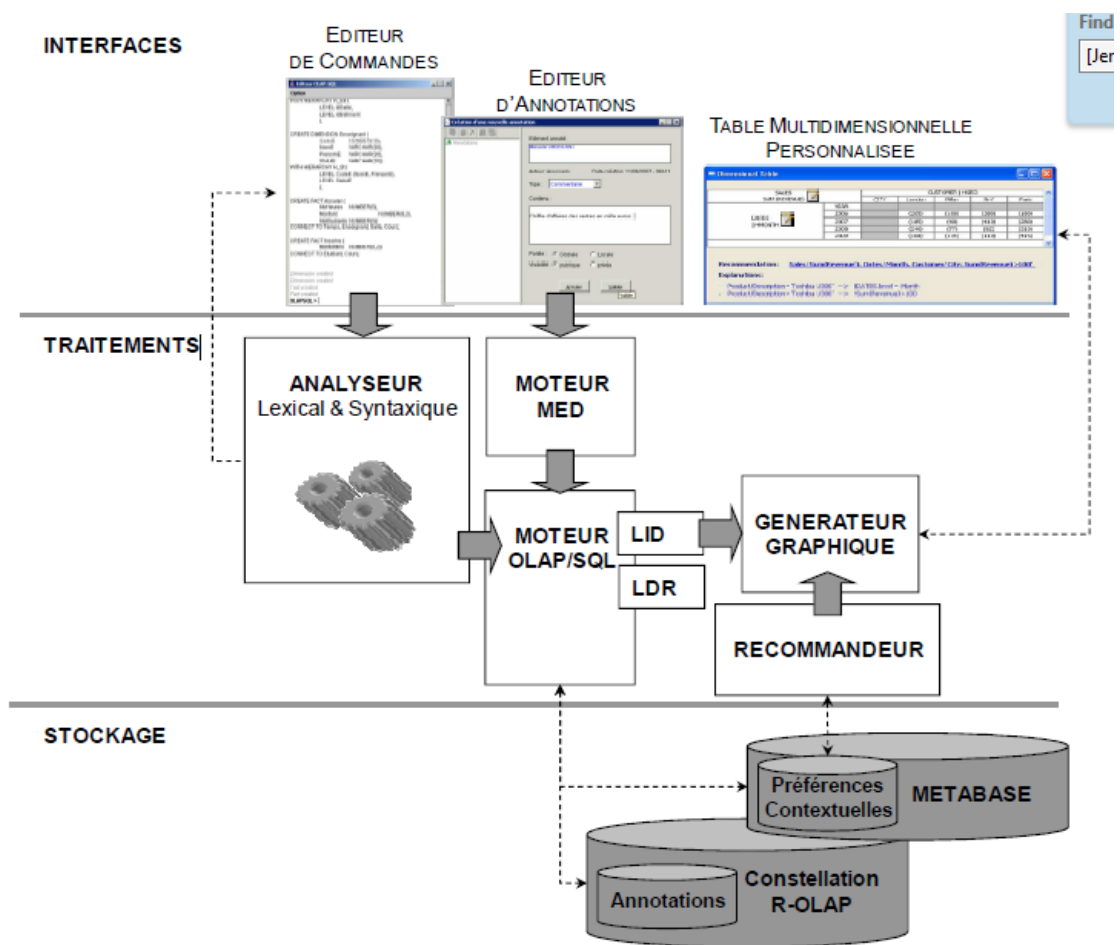


Figure 2.5: Architecture de langage GOLAP Personnel [13]

2.4.7 Langage Docware

Ce langage permet de l'intégration et l'analyse des documents textuels intitulé DOCument WAREhouse (DOCWARE). DOCWARE support une construction graphique incrémental des entrepôts de document a partir de document filtrées et sélectionner. Assiste le concepteur dans l'élaboration des magazines de documents. [15]

Architecture

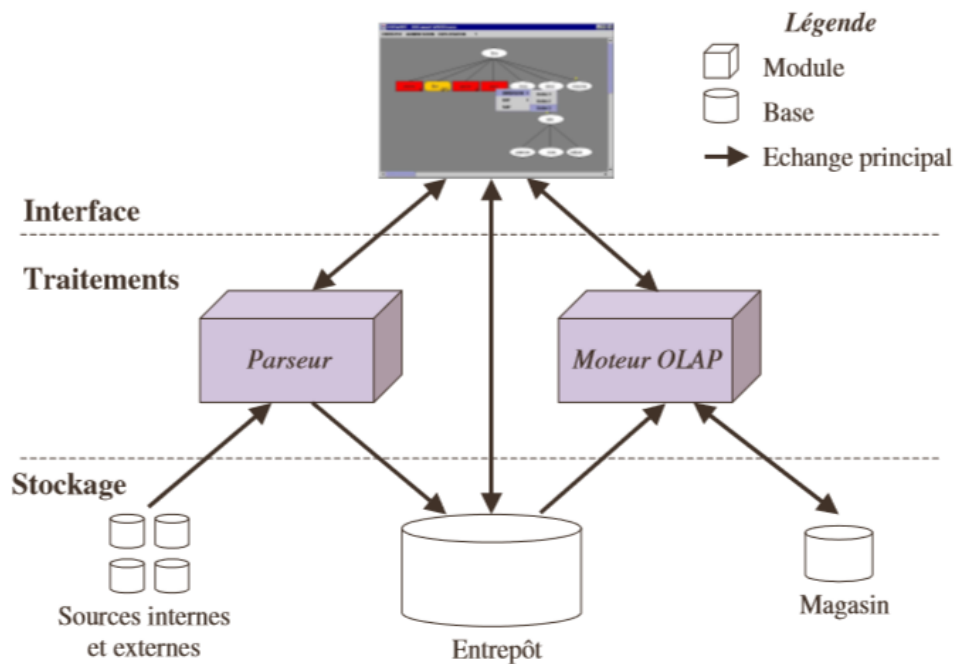


Figure 2.6: Architecture de langage DocWare [15]

Une interface graphique: qui permet d'intégrer un document dans un entrepôt documentaire, de visualiser les structures logiques de cet entrepôt et les manipuler afin de créer des schémas multidimensionnels.

Traitement: qui regroupe deux modules. Le parseur permet l'intégration automatique des documents XML dans l'entrepôt. Il repose sur l'extraction de structure et de contenu, comparaison de structure arborescents et insertion du contenu du document. Le modèle analyse permet de construire un schéma multidimensionnel à partir des structures logique contenues dans l'entrepôt.

Stockage: qui permet de sauvegarder les documents dans un SGBD Oracle.

Moteur OLAP

Ce module permet de construire un schéma multidimensionnel et affiche la table multidimensionnelle résultante. Pour la construction du schéma multidimensionnel. Le système affiche les structures logiques génériques ou la liste des documents de l'entrepôt. Le concepteur sélectionne un des composants affichés et l'effectue à une structure multidimensionnelle (fait, dimension) via des menus contextuels.

Quant aux outils commerciaux, ils s'améliorent d'année en année. Nous pouvons citer:

2.4.8 COGNOS PowerPlay

Outil d'analyse OLAP permet d'analyser les cubes de données qui sont représentés par un schéma en étoile. Les cubes sont représentés sous forme arborescente fondée par IBM. Cet outil est classé le 1er mondial du « business intelligence », son interface est présentée sous la forme d'un tableur muni de fonctions très évoluées.

Les requêtes se font par glisser-déplacer des éléments (mesures, dimensions, hiérarchies, paramètres et attributs faibles) vers la fenêtre de présentation des résultats (un tableur), en les positionnant en lignes ou en colonnes.

2.4.9 CRYSTAL Analysis

Un outil pour visualiser les données conceptuelles, arborescence et a visualisé un seul fait à la fois développé par Seagate Software. Il contient une extension pour Excel Il a été publié sous le nom Analyse Seagate comme une application libre écrite en Java publiée en 1999.

A un accès facile aux données de l'entreprise et les exploiter, réponses fiables aux n'importe quelles questions de gestion, création de rapports ad hoc et manipulation de données illimitées, renforcement de la visibilité sur vos données grâce à des options de mise en forme des données, à des calculs personnalisés, à des formules, à des graphiques, à un suivi des modifications des données.

Des fonctionnalités avancées de sélection sont mises en avant mais elles requièrent toutes de maîtriser un langage de requête textuel (SQL en général).

RATIONAL Rose XDE

Rational Rose XDE, un «eXtended environnement de développement" pour les développeurs de logiciels, intègre avec Microsoft Visual Studio .fonder par IBM en2006. Rose XDE, un éditeur de schémas UML très connu est édité par Rational Software, racheté par IBM. Mais comme UML étant une méthode sur le paradigme objet, la modélisation multidimensionnelle via XDE n'est donc pas indépendante de l'architecture sous-jacente et impose le support du paradigme objet.

2.5 conclusion

ces logiciels bien que puissants et faciles d'utilisations, souffrent de l'absence de visualisation globale du schéma multidimensionnel car il se limite à la présentation d'un unique schéma.

De plus, le système de visualisation par arborescence est absolument inutilisable en cas de schéma large, imposant un parcours lent et fastidieux.

Il existe aussi des outils issus de l'industrie, qui se concentrent essentiellement sur les aspects de l'interrogation.

Cependant, même si les requêteurs graphiques sont simples et facilitent l'accès aux informations, ils restent limités pour un processus d'analyse complexe.

Les outils spécifiques OLAP offrent de nombreuses possibilités de manipulation multidimensionnelle mais ils sont difficiles d'accès pour les non informaticiens.

Chapter 3

Implémentation

3.1 Introduction

Notre approche sert à offrir aux utilisateurs et aux ingénieurs une plateforme qui permet aux utilisateurs d'interroger une BDM pour exprimer ces besoins et prend des décisions. Donc cette plateforme joue un rôle majeur dans le processus de système d'aide à la décision dans le but de fiabiliser et d'atteindre les bons résultats.

3.2 Conception

Pour réaliser notre projet on a utilisé une base de données trouvée sur web. Cette base de données donne des statistiques sur la ligue majeure de baseball. Elle contient les statistiques de 1871 à 2012. Cette base de données contient des tangages, des frappeurs, et les statistiques de mise en service pour Major League Baseball, de 1871 à 2012. Il comprend les données de ces deux ligues actuelles (américaine et nationale), les quatre autres "grands" ligues (American Association, Union Association, joueurs de la Ligue, et Ligue fédérale), et l'Association nationale de 1871 à 1875.

Cette base de données a été créée par Sean Lahman, pionnier de l'effort de faire des statistiques de baseball librement à la disposition du grand public. Ce qui a commencé comme un effort d'un homme en 1994 a considérablement augmenté, et maintenant une équipe de chercheurs ont rassemblé leurs efforts pour faire de cette la plus grande et la plus précise. Source de statistiques de baseball disponibles n'importe où.

Cette base de données ne peut jamais prendre la place d'un bon livre de référence comme L'Encyclopédie de baseball. Mais il permettra aux gens font pour le genre des requêtes et des analyses que ces sources traditionnelles ne permettent pas.

3.2.1 Langages

HTML (**HyperTextMarkupLanguage**)

HTML est un langage à balisage qui décrit la structure logique d'un document hypertexte, Né en 1989 sous l'impulsion de TimBernersLee inventeur du Web. Il est basé sur SGML(StructuredMarkupLanguage), qui est une vieille norme utilisée pour la description de documents [16].

Ce langage contient des commandes implémentées par des balises pour marquer les différents types de texte (titres, paragraphe, listes...), inclure des images, des formulaires, des liens ...etc.

CSS(**Cascading Style Sheet**)

Dès 1994, devant les relatives faiblesses du Html en matière de présentation, l'idée d'adjoindre des styles au Html comme dans les logiciels de traitement de texte fait son apparition. En 1995, le W3C (WWWConsortium) à réaliser une première ébauche sur les feuilles de style qui est publier en novembre 1995, ce document devient une recommandation officielle le 17 décembre 1996 sous le nom de "Feuilles de style en cascade niveau1" (Cascading Style Sheets, level 1) CSS1 [17].

Ces feuilles de style permet de séparer les données et la mise en forme pour:

- Améliorer l'accessibilité.
- Faciliter la maintenance.
- Faciliter le portage pour un autre site.
- Simplifier et aérer le code.
- Uniformisation du code entre les pages du même site.

JavaScript

Le JavaScript est un langage de programmation de script orienté objet, il a été créé par la société Netscape apparue en 1996 puis normalisé en Juin 1997 par l'ECMA (European Computer Manufacturers Associations) [18].

Ce langage permet au développeur de faire :

- Petites applications : calculettes, éditions de texte, jeux, ...
- Aspect graphiques : modification d'images, gestion de fenêtres, modification locale de la page HTML, ...
- Test de validité des champs de saisies : vérifier si une zone de saisie a le bon format, ...

Le script est un programme dont le code source est inclus dans un document HTML. Ce programme est interprété et s'exécute sur la machine du client lorsque le document est chargé ou lors d'une action de l'utilisateur.

La dernière version du langage est le JavaScript 2.

a) Fonctionnement du JavaScript :

La figure suivante illustre le fonctionnement du JavaScript [14] :

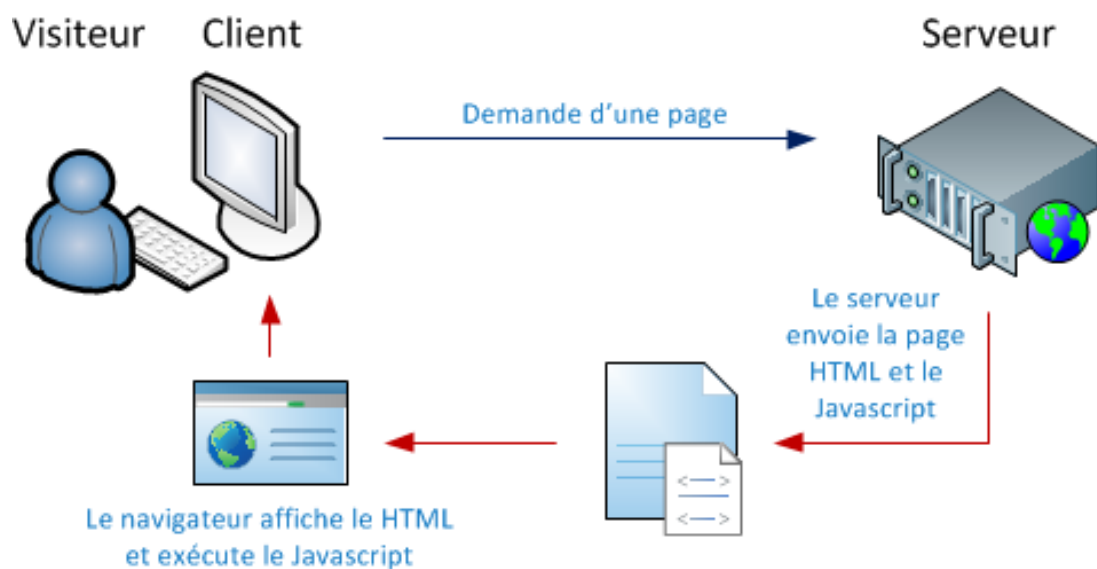


Figure 3.1: Fonctionnement du JavaScript [14]

b) Les différents niveaux d'insertion du code JavaScript:

Le code JavaScript est inséré comme suit [13] :

- Soit utiliser la balise `< SCRIPT>`.
- Soit utiliser un fichier externe `< SCRIPT SRC="fichier.js">`.
- Soit dans un attribut de balise.
- Soit en gestion d'événements

PHP

Le langage PHP a été conçu en 1994 par Rasmus Lerdorf. Ce langage de programmation permet essentiellement de construire des sites Web dynamiques et particulièrement lorsqu'ils sont reliés à une base de données.

Le plus important avantage du langage PHP est le support d'un grand nombre de bases de données comme : InterBase PostgreSQL, dBase, FrontBase, Direct MS-SQL, MySQL, IBM, ODBC, Oracle ...

Le langage PHP inclus le support des services utilisant les protocoles tels que IMAP, SNMP, NNTP, POP3 et http [19].

a) Fonctionnement du PHP:

La figure suivante nous montre le fonctionnement du système lorsque le serveur a reçu une demande par un client ou un navigateur [20].

3.2.2 Environnement de développement**WAMPSEVER**

Le WAMPSEVER est un Environnement de développement qui permet de construire des applications web dynamiques et qu'il offre les outils nécessaires pour les manipuler [21].

- Le PHPMyadmin comme un serveur MySQL.

- L'Apache comme un serveur web.
- L'interpréteur de script (PHP).

Manipulation:

Une fois l'installation du Wampserver est terminée, lancer son exécution. L'icône du Wampserver va apparaitre sur la barre des tâches sous forme d'un demi-cercle blanc.

Remarque: si la couleur du demi-cercle est rouge ou jaune, cela veut dire que l'un des deux serveurs (Apache,MySQL) a du mal démarrer, ou les deux, et il faut les redémarrer.

- Le serveur MySQL: MySQL est un serveur de base de données SQL très rapide, multi-threadé, multi-utilisateur et robuste. Le serveur MySQL est destiné aux missions stratégiques et aux systèmes de production à forte charge, ainsi qu'à l'intégration dans des logiciels déployés à grande échelle.
- Le serveur web(Apache):

En donnant l'adresse du 'LocalHost' qu'est une adresse IP ou un nom de domain (localhost .ex) au navigateur, puis le serveur web va lancer sa page d'accueil suivante:

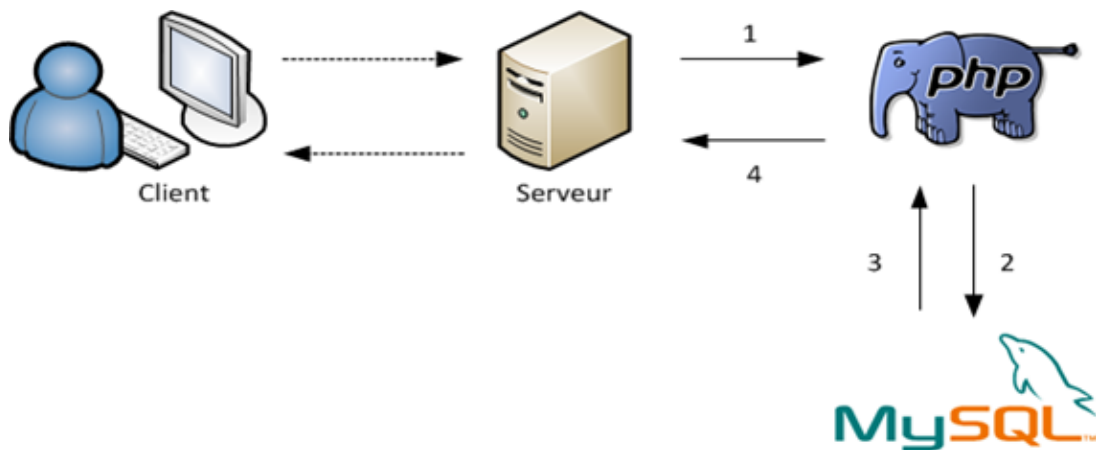


Figure 3.2: Fonctionnement du langage PHP [20]

3.3 Structure de l'outil

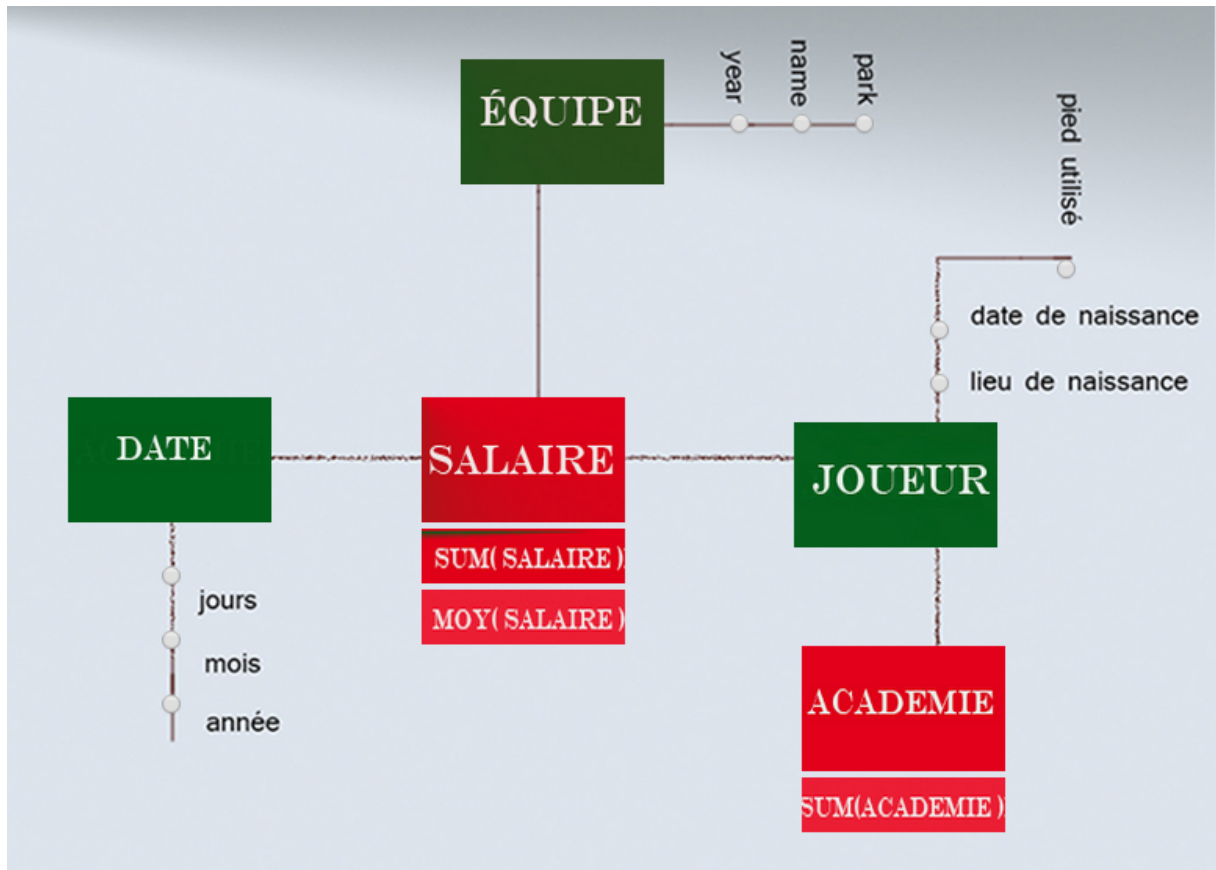


Figure 3.3: La page d'accueil

Explication:

La première page qui s'affiche à l'utilisateur est la page d'accueil, qui permet à l'utilisateur de connaître le schéma de base de donnée en schéma constellation et tout les fonctions prédéfini qui peut utiliser dans cette base de donnée multidimensionnelle.

Dans le schéma constellation les table de fait son présenter avec la couleur rouge et les table de dimensions son présenter avec le vert.

Dans cette page l'utilisateur doit choisir la fonction qui voulez l'utiliser pour interroger la base de donnée multidimensionnel. Après l'utilisateur va donc sélectionner des éléments de ce graphe pour interrogation,il doit choisir sur chaque table de dimension un mesures sur quel base l'interrogation de notre base de donnée. (figure 3.13)

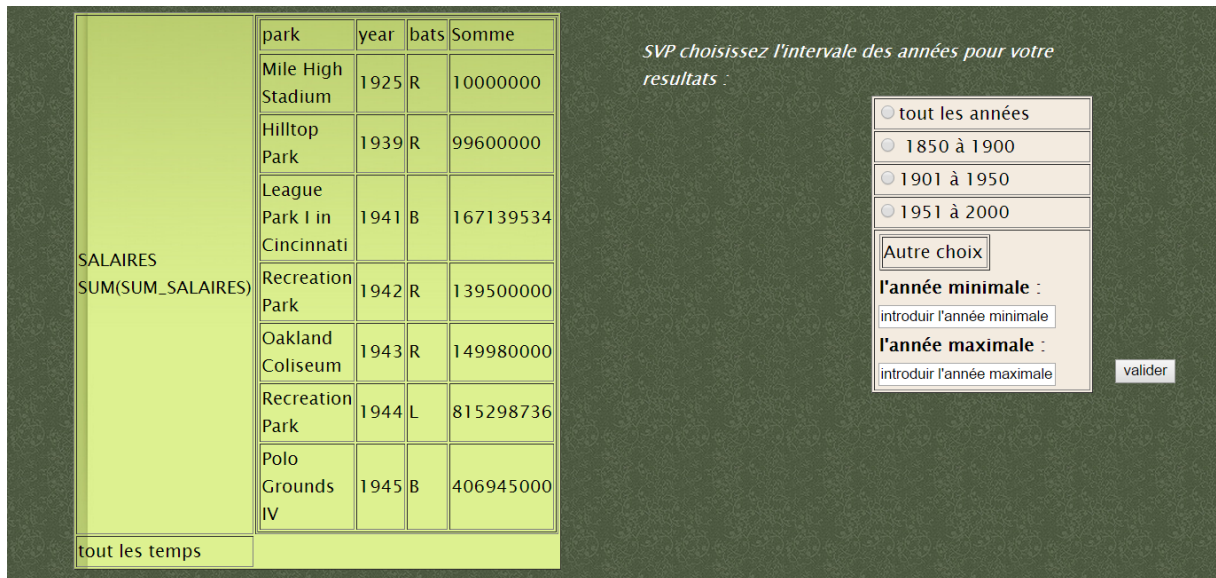


Figure 3.4: L'aspect temps

Explication:

Après le choix des mesures et la fonction d'interrogation et la validation, le prototype affiche une fenêtre qui contient tous les résultats selon la mesure choisie, groupée par Année/Mois ou Jours. Pour utiliser l'aspect temporel, la fenêtre affiche des intervalles de temps. Pour cela, l'utilisateur doit choisir un intervalle de temps qu'il fixe ou un autre intervalle défini par l'utilisateur (figure 3.14).



Figure 3.5: Résultat et Opération OLAP

Explication: Après le choix de l'intervalle de temps notre prototype dans les résultat corresponde a l'intervalle choisir et les mesures et la fonction choisir dans un table. Ansi notre prototype permet a l'utilisateur d'applique quelque opération OLAP sur la table résultant (figure 3.15).

- * RoulUp qui permet de réduire le détail des données visualisées .
- * FRotate qui permet de changer le fait de l'analyse courante.
- * Switch qui permet une permutation des valeurs des paramètres d'une des dimensions.

La figure 3.13|3.14 affiche les résultats après l'application des operation OLAP choisir par l'utilisateur. Ansi notre prototype donne les résultats en forme graphique pour plus de précision (figure 3.15).

	park	year	bats	SUM	1901-> to ->1950
SALAIRES SUM(SUM_SALAIRES)	Mile High Stadium	1925	R	10000000	
	Hilltop Park	1939	R	99600000	
	League Park I in Cincinnati	1941	B	167139534	
	Recreation Park	1942	R	139500000	
	Oakland Coliseum	1943	R	149980000	
	Recreation Park	1944	L	815298736	
	Polo Grounds IV	1945	B	406945000	

	park	year	bats	SUM	1901-> to ->1950
SALAIRES SUM(SUM_SALAIRES)	23rd Street Grounds	1990	R	242466400	
	Arlington Stadium	1985	L	17654483430	
	Arlington Stadium	1966	R	111918498647	
	Atlanta-Fulton County Stadium	1989	L	1352583309	
	Atlanta-Fulton County Stadium	1977	R	172765242506	
	Bank One Ballpark	1987	R	5200257511	
	Bennett Park	1988	R	4179726688	

Figure 3.6: Résultat par Switch

3.4 Conclusion

Nous avons présenté, dans ce chapitre la conception utiliser et notre prototype. Le prototype permet de manipuler une base de données multidimensionnelle ,l'interface graphique fournie permet de manipuler et d'interroger une base de données multidimensionnelle . Jusque-là, on a terminé avec la partie implémentation.

SALAIRES SUM(SUM_SALAIRES)	park	year	bats	Somme	1901 to 1950
	Mile High Stadium	1925	R	10000000	
	Hilltop Park	1939	R	99600000	
	League Park I in Cincinnati	1941	B	167139534	
	Recreation Park	1942	R	139500000	
	Oakland Coliseum	1943	R	149980000	
	Recreation Park	1944	L	815298736	
	Polo Grounds IV	1945	B	406945000	

SALAIRES SUM(SUM_SALAIRES)	year	park	Somme	1901-> to->1950
	1925	Mile High Stadium	10000000	
	1939	Hilltop Park	99600000	
	1941	League Park I in Cincinnati	167139534	
	1942	Recreation Park	139500000	
	1943	Oakland Coliseum	149980000	
	1944	Recreation Park	815298736	
	1945	Polo Grounds IV	406945000	

Apres RollUp ----->

Figure 3.7: Résultat par roullUp

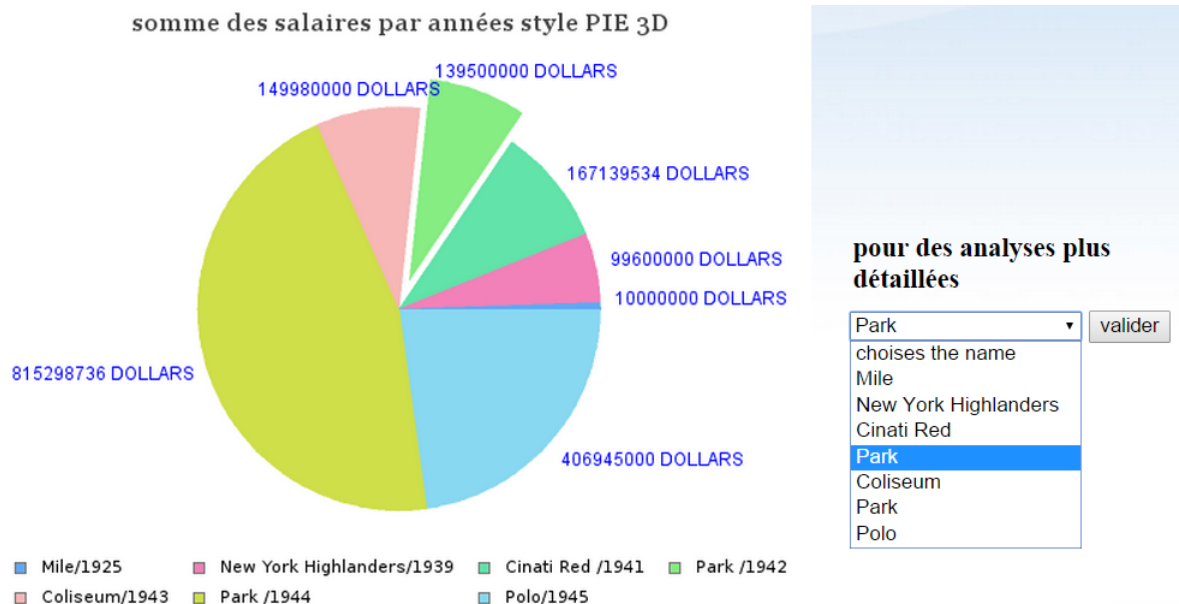


Figure 3.8: Résultat graphique

Conclusion

Ce travail consiste à concevoir un langage graphique qui permet l'interrogation de bases de données multidimensionnelles conçues selon un modèle permettant la gestion de l'évolution temporelle des données. Un langage d'interrogation d'une base de données conçue selon ce modèle, devrait donc pouvoir répondre à des requêtes temporelles. Notre apport, par ce travail, a été de concevoir un tel langage, qui permet d'interroger une base de données multidimensionnelle graphiquement.

Le travail accompli, ne constitue qu'une brique dans la gestion de bases de données multidimensionnelles temporelles et de nombreuses perspectives peuvent être envisagées, d'une part pour améliorer notre langage. Comme premières perspectives, il s'agit de :

- Améliorer l'interface de visualisation du prototype en proposant une représentation graphique qui permet de visualiser les différents schéma multidimensionnel.
- Définir des commandes graphiques qui permettent une manipulation des opérations OLAP sur les tables résultats des requêtes sans revenir à la représentation graphique du schéma conceptuel.
- Améliorer le prototype pour être compatible avec n'importe quelle base de données multidimensionnelle.

Ces différents points représentent des perspectives qui peuvent faire l'objet d'une autre étude.

Bibliography

- [1] Olivier TESTE. Modélisation et manipulation d'entrepôts de données complexes et historisées. 2001.
- [2] R. KIMBALL. The data warehouse toolkit. 1996.
- [3] Olivier TESTE Frédéric BRET. Construction graphique d'entrepôts et de magasins de données. 2008.
- [4] G.HUBERT. les versions dans les bases de données orientées objet :modélisation et manipulation. 1997.
- [5] M.MIQUEL A.TCHOUNKINE M.BODY, Y.BEDARD. A multidimensional and multiversion structure for olap. 2002.
- [6] S.STOCK P.CHAMONI. Temporal structures in data warehousing.
- [7] Leïla Benakezouh. Etude et conception de bases de données multidimensionnelles temporelles. 2006.
- [8] R. Tournier G. Zurfluh F. Ravat, O. Teste. Algebraic and graphic languages for olap manipulations. 2008.
- [9] E. Annoni. Conception et developpement d'un langage assertionnel pour les bases de donnees multidimensionnelles. 2003.
- [10] R.WREMBEL T.MORZY. On querying versions of multiversion data warehouse. 2003.
- [11] E. Annoni. Conception et développement d'un langage assertionnel pour les bases de données multidimensionnelles. 2003.
- [12] F. Abdelhedi. Etude et développement d'un mécanisme d'agrégation pour l'analyse olap de documents xml. 2009.
- [13] H. Jerbi. Mémoire d'expertises décisionnelles à base d'annotations. 2007.

- [14] F. Atigui. Personnalisation de requêtes dans une base de données multidimensionnelle. 2009.
- [15] K.Khrouf. Entropots de documents: de l'alimentation a l'exploitation. 1997.
- [16] Luc VAN LANCKER. Html 4 maîtrisez le code source. 2008.
- [17] Luc VAN LANCKER. Des css au dhtml java script appliqué aux feuilles de style. 2006.
- [18] Emmanuel GUTIERREZ. Javascript des fondamentaux aux concepts avancés. 2008.
- [19] Egon Schmid Jim Winstead Lars Torben Wilson Rasmus Lerdorf Zeev Suraski Andrei Zmievski Jouni Ahto Damien Seguy Stig Saether Bakken, Alexander Aulbach. Manuel php. 2001.
- [20] Jean-Marie Defrance. Formation elephorm apprendre php et mysql .
- [21] Fabien Duchateau. Lif4 programmationweb cours css. 2012.