

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaia
Faculté des Sciences exactes
Département d'informatique



En vue de l'obtention du diplôme de Master en Administration et sécurité des réseaux

Mémoire de fin De cycle

**Thème : Etude et Configuration d'un entrepôt de données
cas hôpital Khelil Amrane**

Réalisé par :

M^r OUHAB Abdallah
M^r RAMTANI Tarik

Encadré par :
M^r SEBAA Abderrzak

Examiné par
Pr. *M^r* TARI Abdelkamel
Ex. *M^{me}* TAHAKOURT Zineb
Ex. *M^r* SIDER Abderrahmane

Promotion : 2013/2014

Remerciements

*Nous remercions **DIEU** tout puissant de nous avoir donné la force, la santé, le courage et la patience de pouvoir accomplir ce travail.*

Un grand merci à toutes nos familles surtout nos parents pour leur encouragement et leur suivi avec patience du déroulement de notre projet.

*Nos remerciements s'étendent à notre promoteur **M^r SEBAA Abderrzak** pour sa patience, sa disponibilité et surtout pour ses judicieux conseils, qui ont contribué à alimenter nos réflexions.*

*Nos sincères remerciements s'adressent à **M^r BENBOURNAN Massinissa** pour sa disponibilité et ces efforts qui nous ont aidés à réaliser notre travail, aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

Nous tenons à remercier chacun des membres du jury pour nous avoir fait l'honneur d'examiner et dévaluer notre travail et de l'enrichir par leurs propositions.

Dédicaces

A nos chers parents qui nous ont donné un magnifique modèle de labeur et de persévérance.

A nos frères et sœurs à qui nous souhaitons un avenir prospère.

A toutes nos familles sans exceptions.

A tous nos amis et collègues.

A tous ceux qui nous ont aidés.

A tout le personnel du département informatique.

A tous ceux que nous avons omis.

Nous dédions cet humble travail.

TABLE DES MATIÈRES

Thème : Etude et Configuration d'un entrepôt de données	1
cas hôpital Khelil Amrane	1
1 Concepts de base sur les BD et les ED	12
1.1 Base de données	12
1.1.1 Historique	12
1.1.2 Définition	13
1.1.3 Objectifs d'une base de données	14
1.2 Système de Gestion de Base de Données (SGBD)	14
1.2.1 Historique	14
1.2.2 Definition	15
1.2.3 Objectifs d'un SGBD	16
1.2.4 Les différents modèles de bases de données	16
1.3 Les entrepôts de données	17
1.3.1 Historique	17
1.3.2 Définition	17
Non volatiles	17
Orientées sujet	18
Intégrées	19
Historisées	19
1.3.3 Différence entre les Bases de données et les Entrepôt de données	19
1.3.4 Architecture d'un entrepôt de données	20
Les sources de données	21
Stockage	21
Serveur OLAP	22
Outils de front end	22
Table de fait	23
Table de dimension	23

Le cube de données	23
Datamart	24
OLAP (On-Line Analytical Processing)	25
2 Outils d'implémentation d'ED	35
2.1 SQL Server	35
2.1.1 Historique	35
2.1.2 Définition	36
2.1.3 Les fonctionnalités principales de Microsoft SQL Server	36
2.1.4 Les avantages de Microsoft SQL Server	37
2.1.5 Les inconvénients de Microsoft SQL Server	37
2.2 Oracle	37
2.2.1 Historique	38
2.2.2 Définition	39
2.2.3 Les fonctionnalités principales d'Oracle	39
2.2.4 Les avantages d'Oracle	39
2.2.5 Les inconvénients d'Oracle	40
2.3 Pentaho	40
2.3.1 Historique	40
2.3.2 Définition	41
2.3.3 Les fonctionnalités principales de Pentaho	41
2.3.4 Les avantages de Pentaho	42
2.3.5 Les inconvenants de Pentaho	43
2.4 Talend open studio	43
2.4.1 Historique	43
2.4.2 Définition	43
2.4.3 Les fonctionnalités principales de Talend Open Studio	44
2.4.4 Les avantages de Talend Open Studio	44
2.4.5 Les inconvénients de Talend Open Studio	45
3 Comparaison entre les Outils d'ED	46
3.1 Les paramètres de comparaisons	46
3.1.1 paramètres de forme	46
3.1.2 Capacité	47
3.1.3 Les étapes d'implémentation d'un ED assuré par l'outil	47
3.1.4 Type d'index supporté	47
3.1.5 Fonctions UDF(User Defined Function) supportées	48
3.1.6 Procédures stockées	48
3.1.7 Triggers	48
3.1.8 Partitionnement des tables	48
3.1.9 Protection de la force brute	49

3.1.10 Transaction	49
3.1.11 Cursors	49
3.1.12 Vues	49
3.1.13 Contrainte d'intégrité	49
3.1.14 Outils d'administration et de gestion disponible	49
3.1.15 Interface graphique et l'ergonomie	49
3.1.16 Métadonnée	50
3.2 Comparaison	50
3.3 Discutions des résultats	58
4 Conception	59
4.1 Présentation d'organisme d'accueil	59
4.1.1 Historique de CHU de Bejaïa	59
4.1.2 Mission de CHU de Béjaïa	60
4.1.3 Structures de CHU de Béjaïa	60
Hôpital Khelil Amrane	60
Hôpital Frantz Fanon	61
Hôpital Targua Ouzemour	62
Centre de wilaya de transfusion sanguine (CWTS)	62
4.1.4 Organigramme administratif du CHU de Béjaïa	63
4.2 Les étapes de conception de l'entrepôt de données	65
4.2.1 Spécification des besoins	65
4.2.2 Concevoir le modèle multidimensionnelle	66
Les Dimensions	66
Les mesure	69
Table de faits	69
La Modélisation	70
4.2.3 Le cube OLAP	72
Représentation d'un cube de données	73
Les opérations OLAP	73
4.2.4 Représentation graphique	75
5 Implémentation	76
5.1 Les étapes de création de l'entrepôt de données	76
5.2 Exécution des requêtes T-SQL	77
5.2.1 Création de la base de données de l'entrepôt	78
5.2.2 Créeer et peupler les tables des dimensions	78
Création des tables des dimensions	78
peupler les tables de dimensions	78
5.2.3 Créeer et peupler de la table de fait	78
5.3 Création de cube OLAP	80

5.3.1	Démarrer le Projet Analyses Services	80
5.3.2	création de la source de données	80
5.3.3	Création des Vues de la source de données	82
5.3.4	Création de Cube	84
5.3.5	Modification les Dimensions	84
5.3.6	Création d'attribut Hiérarchie Dans la dimension Temps	85
5.3.7	Déploiement de Cube	85
	Etape 01 :	86
	Etape 02 :	86
	Etape 03 :	87
5.3.8	Traitements de cube	87
	Etape 01 :	87
	Etape 02 :	87
	Etape 03 :	88
5.3.9	Parcourir le Cube d'analyse	88
5.4	Exemples des résultats trouvés	90
5.4.1	Exemple avec une dimension	90
5.4.2	Exemple avec deux dimensions	90
5.4.3	Exemple avec trois dimensions	91
	Références	97

TABLE DES FIGURES

1.1	Non volatilité des données [1.24]	18
1.2	Données orienté sujet dans un ED [1.23].	18
1.3	Architecture d'un entrepôt de données [1.25]	21
1.4	Exemple de table de fait	23
1.5	Tables de dimension	23
1.6	schéma générale d'un Cube de données	24
1.7	Un cube de données original.	26
1.8	L'opération Pivot.	27
1.9	L'opération Slice	27
1.10	L'opération Dice	28
1.11	L'opération Roll up	28
1.12	L'opération Drill down	29
1.13	Le schéma en étoile	31
1.14	Le schéma en flocon de neige	32
1.15	Le schéma en constellation	33
4.1	Organigramme administratif du CHU de Béjaïa	64
4.2	Les étapes d'implémentation de notre entrepôt de données	65
4.3	Table de dimension " Maladie "	67
4.4	Table de dimension " Temps "	67
4.5	Table de dimension " Région "	68
4.6	Table de dimension " Patient "	68
4.7	Table de dimension " Direction "	69
4.8	Table de fait	70
4.9	Schéma en étoile	71
4.10	Schéma en flocon de neige	71
4.11	Schéma en constellation	72
4.12	Le cube original	73

4.13 L'opération Pivot	74
4.14 L'opération Slice	74
4.15 L'opération Dice	74
4.16 L'opération Roll up	74
5.1 Les étapes de la réalisation de notre entrepôt de données	77
5.2 Démarrer Projet Analysis Services	80
5.3 Nouvelle source de données	81
5.4 New	81
5.5 Création d'une nouvelle connexion	81
5.6 Hériter d'options	82
5.7 Nouvelle source de données View	83
5.8 déplacement la table de faits	83
5.9 Ajout des tables associées	83
5.10 Nouveau Cube	84
5.11 Ajout aux Attributs volet sur le côté gauche.	85
5.12 Création d'attribut Hiérarchie Dans la dimension Temps	85
5.13 Déploiement Propriétés Première	86
5.14 Exécuter pour traiter le Cube	87
5.15 Traitement terminé	88
5.16 Parcourir le Cube d'analyse	89
5.17 exemple des résultats	89
5.18 Exemples de résultats avec une dimension	90
5.19 le nombre de malades pendant des périodes de temps différentes et pour des maladies différentes	91
5.20 Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 10ans et 20ans.	91
5.21 Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 21ans et 40ans.	92
5.22 Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 41ans et 60ans.	92
5.23 l'interface principale.	93
5.24 Interface commune.	93
5.25 le rapport final.	94

LISTE DES TABLEAUX

1.1	Comparaison entre les Bases de données et les Entrepôts de données [1.24]	20
1.2	comparaison entre ROLAP et MOLAP [1.27]	34

Introduction générale

Un des capitaux les plus importants d'une organisation sont les données qu'elle traite. Les applications qui manipulent ces données n'ont cessé de connaître des progrès permanents en termes de traitements, de stockage et de performances. Grâce aux bases de données exécutées des transactions journalières.

Ces applications utilisent des processus qui permettent de gérer des données variées et d'effectuer des traitements et transactions sur ces données .

Cependant, la plupart des entreprises disposent d'une masse d'information considérable, distribuées sur des systèmes opérationnels, ainsi, ces systèmes paraissent peu adaptés pour servir de support à la prise de décision. Nous nous posons donc la question suivante : comment procéder à une intégration décisionnelle réussite des données éparpillées et hétérogènes dans les BDD opérationnelles ?

Face à cette inadéquation, l'informatique décisionnelle ou encore la Business intelligence (BI) est ainsi apparue durant les années 70, permettant d'exploiter de grandes quantités de données, manipulées dans un but d'analyse, afin d'en tirer les informations pertinentes et de définir des indications et des tableaux de bord qui facilitent la prise de décision.

Cependant, pour rendre cette analyse possible, les entreprises avaient besoin de nouvelle donnée agrégées, consolidées, historique et synthétisées selon plusieurs axes. Le besoin d'une nouvelle architecture, qui stocke et traite ce type de données s'est émergée du domaine de la BI, ce qui a donné naissance à l'architecture d'entrepôts de données (Datawarehouse) et les datamarts (Magasin de données). Ces dernières sont des structures qui intègrent des données pour l'analyse et la représentation d'information pertinente.

Toutefois, les Entrepôts de données ou les datamarts peuvent se révéler des outils audacieux et complexes pouvant mener l'entreprise à sa perte si les informations dont elle dispose ne sont pas mises à jour, par conséquent l'entreprise prendra des décisions en fonction des données erronées. C'est pourquoi l'intégration des données est décisive dans un projet décisionnel. En effet, les données agrégées et intégrées jouent un rôle central par rapport à leurs analyses et à leurs présentations. Selon Ralph Kimball, un des spécialistes dans ce domaine, l'intégration des données représente l'étape la plus critique d'un projet décisionnel.

Problématique La problématique posée dans notre mémoire se situe dans le cadre du développement d'un système décisionnel pour le suivi des maladies dans l'hôpital Khellil Amrane. Ce système intégrera des différentes sources de données (BDD relationnelles), basées sur une structure ciblée qui est le Datamart (magasin de données). Ainsi, notre objectif est de proposer une solution décisionnelle permettant de donner des statistiques sur le nombre de malades par rapport

a plusieurs paramètres comme : le nom de la maladie, la catégorie d'âge des malades, la région etc.

Méthodologie adaptée Afin de réaliser l'entrepôt de données demandé, nous avons effectué une étude sur les différents outils d'implémentations d'entrepôt de données (SQL Server, Oracle, Pentaho, Talend Open Studio), suivie par une comparaison entre ces outils, après avoir analysé les résultats, nous avons constaté que le SQL Server est l'outil le plus adapté pour notre application. Une fois l'outil d'implémentation choisi, nous avons spécifié notre méthodologie de conception et d'implémentation de notre système décisionnel. En effet, nous avons élaboré notre approche en trois phases principales :

- **Identification des besoins** : dans cette phase nous devons appréhender les facteurs clés qui conduisent l'hôpital Khellil Amrane à définir de manière efficace ses besoins et les traduire pour les intégrer lors de la phase de conception.
- **Conception du système** : cette phase comprend la conception de la BDD décisionnelle, à travers des concepts multidimensionnels, et la conception de l'interface utilisateur à savoir le schéma en étoile, en flocon de neige et en constellation.
- **Réalisation du système** : dans cette phase nous allons détailler l'implémentation de l'architecture technique du système décisionnel.

Organisation du mémoire

Afin de répondre à la problématique posée, nous avons présenté ce document en cinq chapitres :

- **Le chapitre 01 " Concepts de base sur les BD et les ED "** : évoque tous les concepts de base sur les bases de données et les entrepôts de données, à savoir, leurs historiques, définitions et architectures etc.
- **Le chapitre 02 " Outils d'implémentation d'ED "** : présenté les différents outils d'implémentation d'entrepôt de données les plus utilisées, à savoir, leurs historiques, définitions, les fonctionnalités principales, les avantages et les inconvénients.
- **Le chapitre 03 " Comparaison entre les outils d'ED "** : effectuer une comparaison entre les quatre (04) outils d'implémentation d'entrepôt de données les plus utilisés : le SQL Server, Oracle, Pentaho et Talend Open Studio, pour extraire à la fin l'outil le plus adapté pour notre application.
- **Le chapitre 04 " Conception "** : présenter d'abord l'organisme d'accueil, à savoir, l'historique, la mission et la structure de CHU de Bejaia, pour finir avec la conception de notre projet.
- **Le chapitre 05 " Implémentation "** : décrit l'implémentation de l'architecture décisionnelle, ainsi que les détails techniques de chaque partie qui compose notre solution décisionnelle.

CHAPITRE 1

Concepts de base sur les BD et les ED

Introduction

Dès le début de l'informatique, on a voulu construire des systèmes pour effectuer des calculs (équations différentielles, calcul matriciel, ...etc). Aujourd'hui, la tendance actuelle est la gestion de grandes quantités d'informations. Cela revient à stocker et manipuler des données. Ces données peuvent être de natures diverses et les opérations plus ou moins compliquées.

1.1 Base de données

1.1.1 Historique

Les disques durs, mémoire de masse de grande capacité, ont été inventés en 1956. L'invention du disque dur a permis d'utiliser les ordinateurs pour collecter, classer et stocker de grandes quantités d'informations.

Le terme database (base de données) est apparu en 1964 pour désigner une collection d'informations partagées par différents utilisateurs d'un système d'informations militaire [1.1].

Les premières bases de données hiérarchiques sont apparues au début des années 1960. Les informations étaient découpées en deux niveaux de hiérarchie : un niveau contenait les informations qui sont identiques sur plusieurs enregistrements de la base de données. Le découpage a ensuite été étendu pour prendre la forme d'un diagramme en arbre [1.1].

En 1965, Charles Bachman conçoit l'architecture Ansi/Sparc¹ encore utilisée de nos jours. En

¹L'architecture fondamentale sur laquelle reposent les SGBD modernes

1969, il crée le modèle de données réseau au sein du consortium CODASYL² pour des applications informatiques pour lesquelles le modèle hiérarchique ne convient pas [1.1]. Charles Bachman a reçu le prix Turing en 1973 pour ses « contributions exceptionnelles à la technologie des bases de données ».

En 1968 Dick Pick crée un système d'exploitation contenant un système de gestion de base de données multivaluée (SGBDR MV).

En 1970, Edgar F. Codd note dans sa thèse mathématique sur l'algèbre relationnelle qu'un ensemble d'entités est comparable à une famille définissant une relation en mathématique et que les jointures sont des produits cartésiens. Cette thèse est à l'origine des bases de données relationnelles. Edgar F. Codd a reçu le prix Turing en 1981[1.1].

Le modèle entité-association a été inventé par Peter Chen en 1975; il est destiné à clarifier l'organisation des données dans les bases de données relationnelles [1.2],[1.3].

Dans le modèle relationnel, la relation désigne l'ensemble des informations d'une table, tandis que l'association, du modèle entité-association, désigne le lien logique qui existe entre deux tables contenant des informations connexes.

Les premières bases de données étaient calquées sur la présentation des cartes perforées³ : répartis en lignes et colonnes de largeur fixe. Une telle répartition permet difficilement de stocker des objets de programmation ; en particulier, elles ne permettent pas l'héritage entre les entités, caractéristique de la programmation orientée objet.

Apparues dans les années 1990, les bases de données objet-relationnel utilisent un modèle de données relationnel tout en permettant le stockage des objets. Dans ces bases de données les associations d'héritage des objets s'ajoutent aux associations entre les entités du modèle relationnel [1.3].

1.1.2 Définition

Une base de données est une collection de données persistantes utilisées par le système d'application d'une entreprise quelconque. Plus précisément, on appelle base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche et consultions de données). [1.4]. [1.5].

²Un consortium formé en 1959 pour guider l'élaboration d'une norme langage de programmation qui pourrait être utilisé sur de nombreux ordinateurs.

³Un morceau de papier rigide qui contient des informations représentées par la présence ou l'absence de trou dans une position donnée.

1.1.3 Objectifs d'une base de données

La base de données a beaucoup d'objectifs parmi lesquels nous pouvons citer :

- Eviter les redondances et les incohérences des données qui entraînaient fatallement une approche où les données seraient reparties dans des différents fichiers sans connexion entre eux ;
- Offrir un langage de haut niveau pour la définition et la manipulation des données ;
- Contrôler l'intégrité entre plusieurs utilisateurs et la confidentialité des données ;
- Assurer l'indépendance entre les données et les traitements. [1.6]

1.2 Système de Gestion de Base de Données (SGBD)

1.2.1 Historique

Jusqu'en 1960 les informations étaient enregistrées dans des fichiers manipulées par les logiciels applicatifs [1.7]. L'idée des bases de données a été lancée en 1960 dans le cadre du programme Apollo. Le but était de créer un dispositif informatique destiné à enregistrer les nombreuses informations en rapport avec le programme spatial, en vue de se poser sur la lune avant la fin de la décennie [1.8]. C'est dans ce but que IBM, conjointement avec Rockwell met sur le marché le logiciel Information Management System (IMS). Avec ce SGBD, les informations sont enregistrées dans des bases de données organisées de manière hiérarchique [1.7].

À la même époque, General Electric, avec l'aide de Charles Bachman met sur le marché le logiciel " Integrated Data Store ". Avec ce SGBD les informations sont enregistrées dans des bases de données organisées selon un modèle réseau, ce qui permet d'enregistrer des informations ayant une organisation plus complexe que le modèle hiérarchique [1.7]. En 1965, Dick PICK développe le système d'exploitation qui comporte un SGBD et le langage Databasic de Charles Bachman[1.9]. En 2002 la technologie de Pick est utilisée dans des produits contemporains tels que JBase⁴ [1.10]. [1.11].

En 1967, le consortium CODASYL forme un groupe de travail qui travaille à la normalisation de deux langages informatiques en rapport avec les bases de données : le DML⁵ et le DDL⁶ [1.7].

Les organisations hiérarchiques et réseau des années 1960 manquaient d'indépendance vis-à-vis du format des fichiers, ils rendaient complexe la manipulation des données et il leur manquait une base théorique. En 1970 Edgar Frank Codd, employé de IBM publie le livre " A relational model

⁴Une suite de composants logiciels permettant de développer et administrer des bases de données.

⁵Data Définition Langage

⁶Data Manipulation Langage

of data for large shared data banks ", un ouvrage qui présente les fondations théoriques de l'organisation relationnelle [1.7]. Sur la base des travaux d'E.F.Codd, IBM développe le SGBD System R, qui sera mis sur le marché à la fin des années 1970. Il est destiné à démontrer la faisabilité d'un SGBD relationnel. Le langage informatique propre à ce SGBD est le " Structured Query Language " (SQL), défini par IBM et destiné à la manipulation des bases de données relationnelles [1.8].

Charles Bachman reçoit le prix Turing en 1973 pour ces contributions à la technologie des bases de données et Edgar Frank Codd reçoit le prix Turing en 1981 pour les mêmes raisons. En 1978, ANSI⁷ publie la description de l'architecture Ansi/Sparc qui sert de modèle de référence en rapport avec l'indépendance des données des SGBD [1.7]. Les deux SGBD ténors du marché de 2010 que sont IBM DB V2 et Oracle Database ont été mis sur le marché en 1979 et sont tous deux basés sur le modèle relationnel. La même année le langage SQL est normalisé par ISO[1.7].

Les moteurs de recherche et les datawarehouse⁸ sont des applications informatiques apparues dans les années 1990, qui ont influencé le marché des SGBD. Les moteurs de recherche ont nécessité le traitement d'informations non structurées et écrites en langage naturel. Et les datawarehouse ont nécessité la collecte et la consolidation de très grandes quantités d'informations en vue de réaliser des tableaux de synthèse [1.12].

Les modèles d'organisation orienté objet et objet-relationnel sont apparus dans les années 1990[1.7]. Les premiers SGBD objet-relationnel ont été Postgres, Informix et Oracle Database en 1995. Le standard relatif au langage SQL a été modifié en 1999 pour pouvoir s'appliquer à ce type de SGBD [1.13].

1.2.2 Definition

Le système de gestion de base de données est une suite de programmes qui manipule la structure de la base de données et dirige l'accès aux données qui y sont stockées. Une base de données est composée d'une collection de fichiers ; le seul moyen d'accéder aux données est par le SGBD, qui sert alors d'intermédiaire entre la base de données et ses usagers. Celui-ci reçoit des demandes de manipulation du contenu et effectue les opérations nécessaires sur les fichiers. Il cache la complexité des opérations et offre une vue synthétique sur le contenu. Le SGBD permet en outre à plusieurs usagers de manipuler simultanément le contenu, et peut offrir différentes vues sur un même ensemble de données [1.14].

⁷Agence Nationale de la Sécurité Informatique

⁸entrepôt de données

1.2.3 Objectifs d'un SGBD

Le Système de Gestion de Base de Données a beaucoup d'objectifs parmi lesquels nous pouvons citer :

- Possibilité de modifier l'organisation physique (accès) sans modifier les programmes.
- Modification du schéma conceptuel sans modification des programmes.
- Manipulations des données par des utilisateurs qui n'ont pas la connaissance de l'organisation de la base et qui disposent de langages évolués "naturels".
- Possibilités pour les utilisateurs avertis de manipuler les données à partir de langages hôtes (Pascal, Fortran, C, Java...). Efficacité et rapidité au niveau des accès sur les supports.
- L'administrateur de la base définit les structures de données, de stockage et de contrôle.
- Le SGBD permet d'éviter qu'une même donnée apparaisse plusieurs fois dans la base pour éviter les problèmes de mise à jour.
- Cohérence des données lors des mises à jour (les règles de contraintes d'intégrité sont définies par l'administrateur).
- Plusieurs applications simultanées sur les données.
- Contrôle des droits d'accès.
- Reprise sur panne [1.15]. [1.16]. [1.17].

1.2.4 Les différents modèles de bases de données

Il existe cinq modèles de SGBD, différenciés selon la représentation des données qu'ils contiennent et qui sont :

Le modèle hiérarchique : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD

Le modèle réseau : comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant.

Le modèle relationnel : les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations.

Le modèle déductif : les données sont représentées sous forme de table, mais leur manipulation se fait par calcul de prédictats.

Le modèle objet : les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées classes présentant des données membres. Les champs sont des instances de ces classes [1.7].

1.3 Les entrepôts de données

Les entrepôts de données deviennent aujourd’hui des outils incontournables dans plusieurs domaines (universités, compagnies aériennes, banques, compagnies d’assurances etc). Le concept d’entrepôt de données a été introduit pour décrire une plateforme logique facilitant l’accès aux diverses données d’une organisation, et permettant l’utilisation de ces données à des fins d’analyse et de prise de décision. [1.18].

1.3.1 Historique

L’histoire de l’entrepôt de donnée commence par la création des deux termes principale de l’entrepôt de donnée : fait et dimension par General Mills et l’Université Dartmouth en 1960.

- 1983 - Teradata introduit dans sa base de données managériale un système exclusivement destiné à la prise de décision.
- 1988 - Barry Devlin et Paul Murphy publient l’article Une architecture pour les systèmes d’information financiers (An architecture for a business and information system) où ils utilisent pour la première fois le terme Datawarehouse.
- 1990 - Red Brick System crée Red Brick Warehouse, un système spécifiquement dédié à la construction de l’entrepôt de données.
- 1991 - Bill Inmon publie Building the Data Warehouse (Construire l’entrepôt de données).
- 1995 -Le Data Warehousing Institute, une organisation à but lucratif destinée à promouvoir le datawarehousing, est fondé.
- 1996 - Ralph Kimball publie The Data Warehouse Toolkit (La boîte à outils de l’entrepôt de données). [1.18]. [1.19]

1.3.2 Définition

Plusieurs définitions ont été données au concept ED⁹. Au risque de ne pas être original, nous retenons la définition de W.H. Inmon, qui, dans son ouvrage de référence "Building the Data Warehouse", décrit un ED comme "Une collection de données orientées sujet, intégrées, non volatiles et historiées, organisées pour supporter un processus d'aide à la décision ". Cette définition englobe les termes clés suivants [1.21] :

Non volatiles

Les données d’un entrepôt sont généralement utilisées en mode consultation. Elles peuvent être interrogées mais ne sont ni modifiées, ni supprimées (sauf dans les cas de rafraîchissement de l’entrepôt). Ceci permet de conserver la traçabilité des informations afin de pouvoir effectuer des analyses sur une longue période (Figure1.1) [1.24].

⁹entrepôts de données

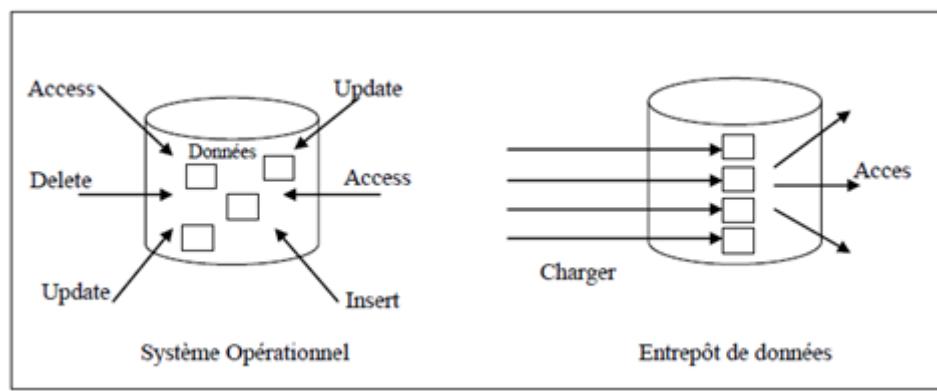


FIG. 1.1 – Non volatilité des données [1.24]

Orientées sujet

Les données des systèmes d'informations sont organisées selon les applications d'une entreprise. Ces données sont orientées applications. Les données d'un ED sont organisées par thèmes ou par sujets (Exemple : Production, Vente, Marketing etc). Cette organisation permet de rassembler toutes les informations relatives à un thème précis afin de faciliter la prise de décision. Ces données sont orientées sujet Figure 1.2 [1.23].

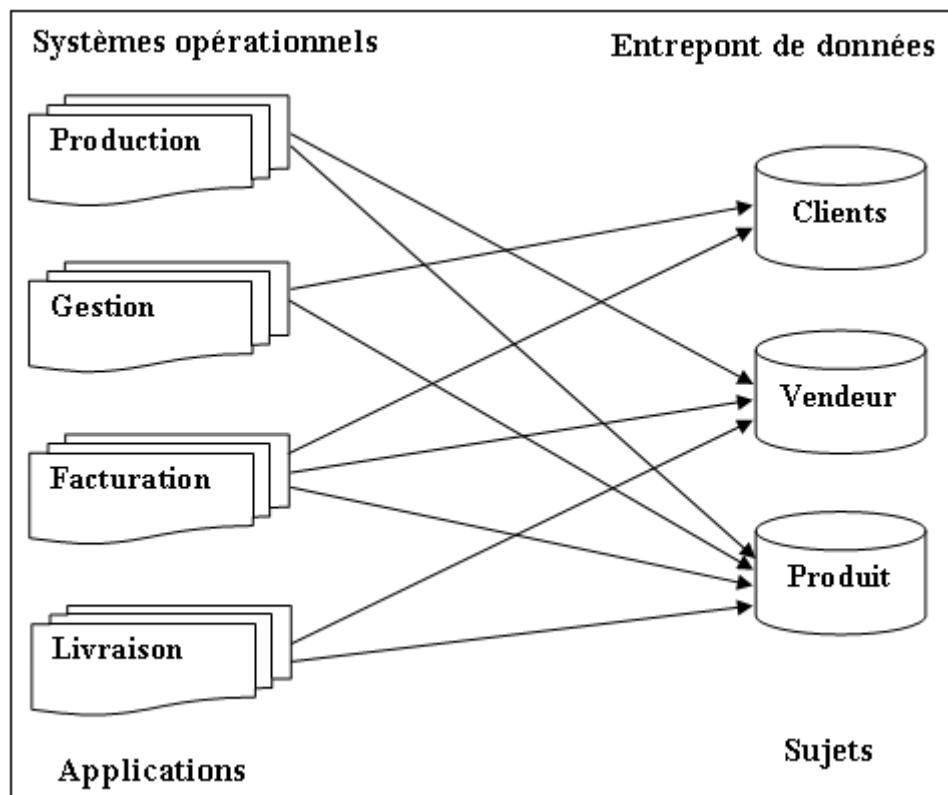


FIG. 1.2 – Données orienté sujet dans un ED [1.23].

Intégrées

Les données d'un ED proviennent de différentes sources hétérogènes. Cette hétérogénéité donne lieu à des conflits syntaxiques et sémantiques (conflits de représentation, de noms, de contexte et de mesure) entre les données des sources. L'intégration des données permet d'éliminer l'ensemble de ces conflits afin d'avoir une représentation uniforme et cohérente des données lors de leur chargement au niveau de l'ED. [1.24].

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. [1.22].

La construction d'un entrepôt revient à faire correspondre les besoins des utilisateurs avec la réalité des informations disponibles. Nous devons d'abord identifier et analyser les sources de données, ce qui nous permet de proposer les mécanismes adaptés selon les caractéristiques des informations. Ensuite, nous devons organiser l'ensemble de données à l'intérieur de l'entrepôt. Pour cela, nous devons d'abord structurer ces informations en considérant leur granularité. Ceci nous permet d'aboutir à la conception d'un schéma multidimensionnel qui permet de répondre aux besoins des utilisateurs. [1.23].

1.3.3 Différence entre les Bases de données et les Entrepôt de données

L'objectif premier d'un ED est de stocker les données pertinentes aux besoins de prise de décision. Contrairement aux bases de données opérationnelles qui sont conçues pour supporter des opérations journalières, un entrepôt est conçu pour supporter des opérations d'analyse utile à la prise de décision. Le tableau suivant (Tableau 1.1) récapitule les principales différences entre les BDDs opérationnelles et les EDs. [1.24]

Caractéristiques	Base de données	Entrepôts de données
But	Exécution d'un processus métier	Exécution d'un processus métier
Usage	Support à la gestion courante	Support à la prise de décision
Principe de Conception	Troisième forme normale	Conception multidimensionnelle
Données	Actuelles, brutes	Historiques, agrégées
Opérations	Lecture et écriture	Lecture et rafraîchissement
Utilisateurs	Employé	Analyste et décideur
Taille	Des giga-octets	Plutôt des téra-octets

TAB. 1.1 – Comparaison entre les Bases de données et les Entrepôts de données [1.24]

1.3.4 Architecture d'un entrepôt de données

Le processus de construction d'un ED, comme illustré dans la Figure 1.3, peut être structuré en quatre axes :

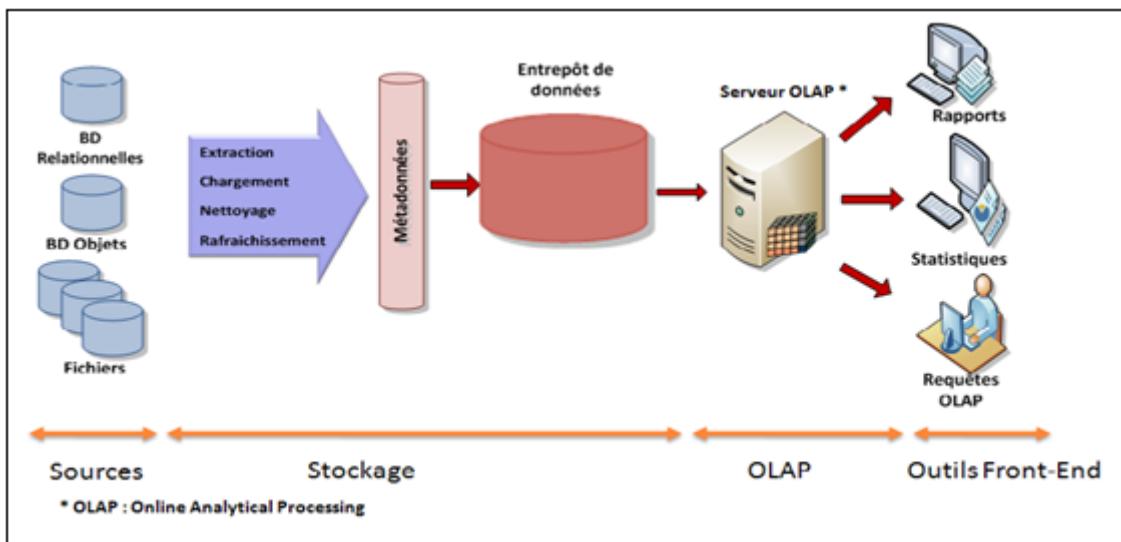


FIG. 1.3 – Architecture d'un entrepôt de données [1.25]

Les sources de données

L'entrepôt de données stocke des données qui proviennent de différentes sources d'informations hétérogènes et distribuées. Ces sources peuvent être des bases de données, des fichiers de données, des sources externes à l'entreprise etc. [1.25]

Stockage

Avant de pouvoir être stockées, les données des sources doivent d'abord être nettoyées. Le processus de nettoyage consiste à sélectionner et à épurer les données pour éliminer toute erreur et réconcilier les différences sémantiques entre ces données. Une fois nettoyées, ces données, seront intégrées dans l'entrepôt. Le processus de rafraîchissement consiste à propager vers l'entrepôt, les changements effectués sur les données des sources par le processus ETL¹⁰ [1.25].

Cette étape consiste à effectuer les transformations nécessaires au chargement des données des sources au niveau du schéma logique de l'entrepôt. Ceci se fait en trois étapes :

- L'extraction : consiste à récupérer les données à partir des systèmes sources. Cette étape nécessite de gérer la synchronisation des processus d'extraction afin d'assurer l'intégrité des données chargées.
- La transformation : consiste en une série de règles permettant le formatage des données extraites selon le schéma cible de l'entrepôt, comme par exemple assigner de la sémantique aux données sources et associer les champs sources aux champs cibles.
- Le chargement : permet l'alimentation de l'entrepôt par les données en respectant les contraintes

¹⁰Extract Transform Load

du SGBD cible. Cette étape doit être validée afin de détecter et corriger toute erreur ayant survenue durant le chargement.

Les données concernant la création, la gestion, et l'usage de l'entrepôt sont stockées dans un répertoire indépendant de l'entrepôt. Ces données sont appelées " métadonnées ". Les méta-données peuvent contenir des informations sur les sources et leurs contenus, le schéma de l'entrepôt, les règles de rafraîchissement, les profils et groupes d'usagers.

Un ED peut comporter plusieurs magasins de données (data-marts). Les magasins sont des extraits de l'entrepôt consacrés à un type d'utilisateurs et répondant à un besoin spécifique. Ils sont dédiés aux analyses décisionnelles de type OLAP ¹¹ [1.25].

Serveur OLAP

Un serveur OLAP permet d'accéder à l'entrepôt, il convertit les requêtes des clients en requêtes d'accès à l'ED et fournit des vues multidimensionnelles des données à des outils d'aide à la décision [1.25].

Outils de front end

Ces outils formatent les données, conformément aux besoins des utilisateurs, sous différentes formes : tableaux, courbes, rapports, statistiques etc [1.25] subsection La modélisation d'un ED Les modèles de conception des systèmes transactionnels OLTP ne sont pas adaptés aux systèmes OLAP dont les requêtes sont souvent très complexes, utilisent beaucoup de jointure, demandent beaucoup de temps de calcul et sont de nature ad hoc.

Pour ce type d'environnement OLAP, une nouvelle approche de modélisation a été proposée : la modélisation multidimensionnelle. Popularisée par Ralph Kimball dans les années 90, cette modélisation est aujourd'hui reconnue comme la modélisation la plus appropriée aux besoins d'analyse et de prise de décision [1.26].

La modélisation multidimensionnelle est une technique de conceptualisation et de visualisation des modèles de données. Elle offre une structuration et une organisation des données facilitant leur analyse. Elle a pour principal objectif d'avoir une vision multidimensionnelle des données. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives d'analyse. Cette modélisation est intéressante d'une part parce qu'elle représente une modélisation assez proche de la manière de penser des analystes, et d'autre part, elle facilite aux utilisateurs la compréhension des données. Le modèle multidimensionnel renferme les deux concepts fondamentaux de faites de dimension.

¹¹On-Line Analytical Processing

Table de fait

Un fait représente un sujet d'analyse. Il est constitué de plusieurs mesures relatives au sujet traité. Ces mesures sont numériques et généralement valorisées de façon continue, en général les faits sont implicitement représentés par la combinaison des valeurs des dimensions. La Figure1.4 illustre un exemple de tables de fait.

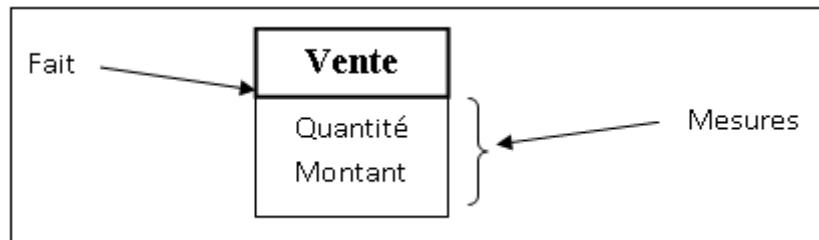


FIG. 1.4 – Exemple de table de fait

Table de dimension

Concept essentiel des bases de données multidimensionnelles, la dimension est le critère suivant lequel on souhaite évaluer, quantifier et qualifier le fait, elle peut être utilisée pour la sélection de données selon le niveau de précision désiré. Aussi, elle peut être affinée, décomposée en hiérarchies, afin de permettre à l'utilisateur d'examiner ses indicateurs à différents niveaux de détail. La Figure1.5 illustre un exemple de tables de dimensions :

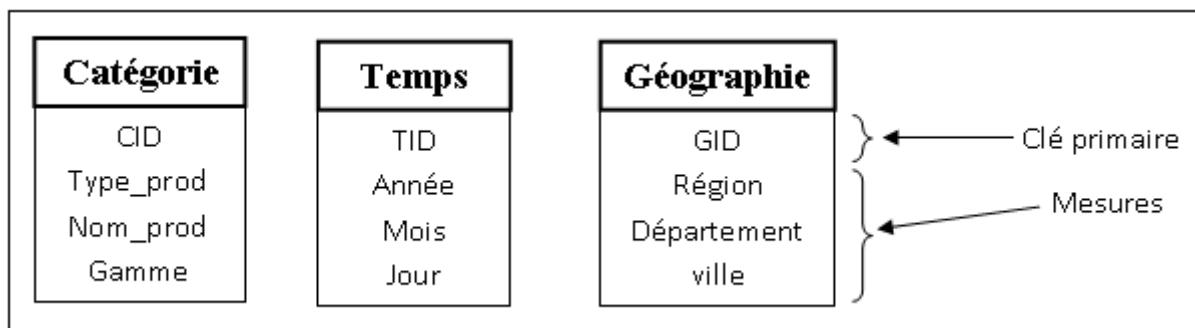


FIG. 1.5 – Tables de dimension

Le cube de données

C'est une représentation multidimensionnelle d'un fait et de ses dimensions. Comme le montre la Figure1.6, par exemple, dans le cas de ventes de produits dans des pays dans le temps ; le fait est les ventes et les dimensions sont : pays, produit et temps. Pour chaque combinaison des trois

dimensions (pays, produit, temps), on peut accéder à la mesure numérique associée au fait ventes (cellule non vide). Les interrogations s'interprètent souvent comme l'extraction d'un plan, d'une droite de ce cube (par exemple, lister les ventes d'un produit X), ou l'agrégation de données le long d'un plan ou d'une droite (par exemple, total des ventes de produit X).

Théoriquement, un cube peut contenir un nombre infini de dimensions. Mais dans la pratique, la

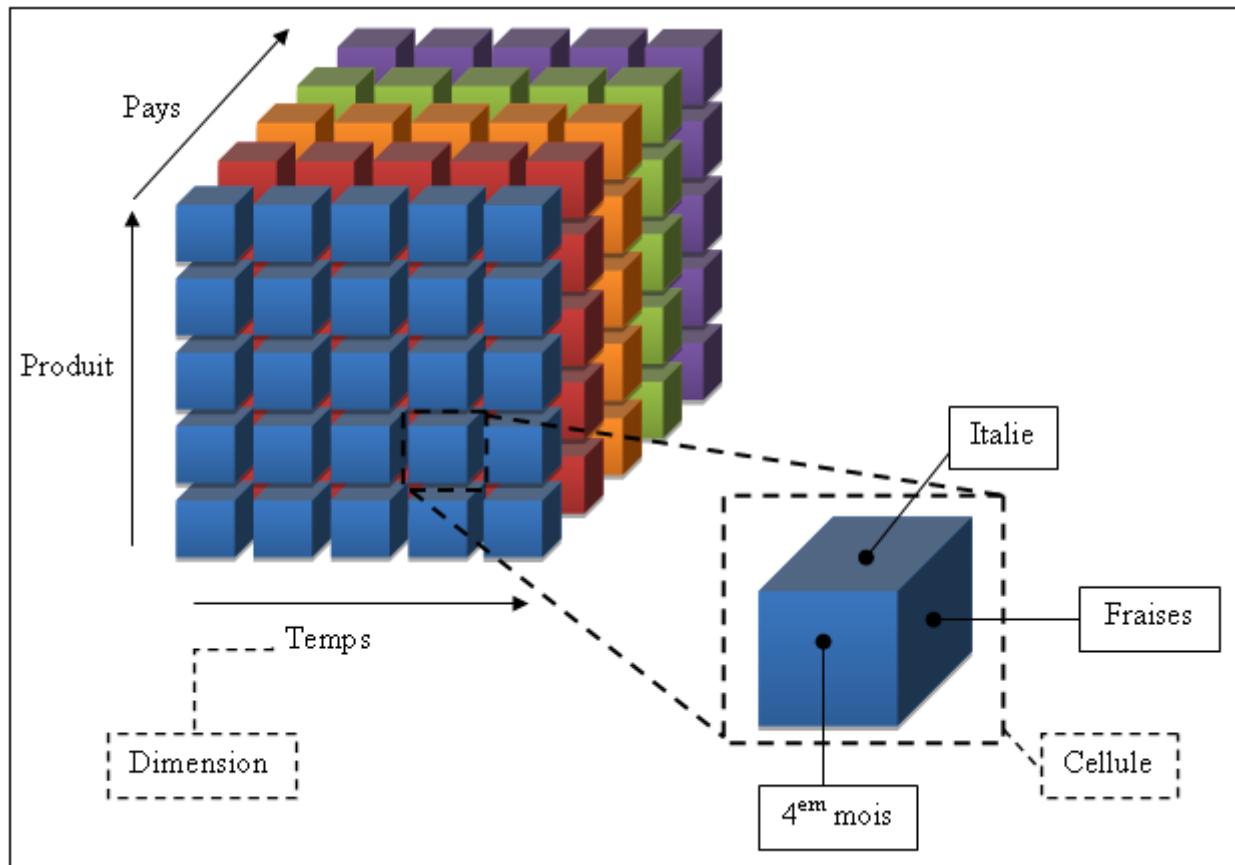


FIG. 1.6 – schéma générale d'un Cube de données

plupart des cubes contient quatre à douze dimensions. Des problèmes de performance sont observés au-delà de cet intervalle.

Datamart

le Datamart est le magasin de données de l'entrepôt. Il permet de distribuer dans une vue métier et agrégée tout ou partie des données stockées dans le Datawarehouse. Il permet ainsi une exploitation optimale des données en les structurant pour des outils ou applications métiers (cube, table individus variables, tables événements etc). [1.30]

OLAP (On-Line Analytical Processing)

Les données opérationnelles constituent la source principale d'un système d'information décisionnel. Les systèmes décisionnels complets reposent sur la technologie OLAP, conçue pour répondre aux besoins d'analyse des applications de gestion.

• 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 multidimensionnelle devrait respecter. Ces règles sont les suivantes.

1. Multi dimensionnalité :

Permet d'avoir une vision multidimensionnelle des données (ce qui n'est pas le cas avec une table - unidimensionnel)

2. Transparence :

L'utilisateur ne doit pas se rendre compte de la provenance des données si celles-ci proviennent de sources hétérogènes (système homogène à l'analyste) ; ces sources peuvent être un fichier Excel, une base de données de production ou même un fichier texte !

3. Accessibilité :

OLAP est décrit comme un middleware qui se place entre les sources de données hétérogènes et un front-end (sous la forme d'un datawarehouse).

4. Stabilité :

Les performances ne doivent pas être diminuées lors de l'augmentation du nombre de dimension ou de la taille de la base de données, mais proportionnelles à la taille des réponses retournées.

5. Client-Serveur :

Il est essentiel que le produit soit Client-Serveur mais aussi que les composants serveurs d'un produit OLAP intègrent facilement ses différents clients.

6. Dimensionnement :

Chaque dimension doit être équivalente par rapport à sa structure et ses capacités opérationnelles.

7. Gestion complète :

Le système OLAP ajuste automatiquement son schéma physique pour s'adapter au type du modèle et au volume des données (plus on dispose de place plus on peut agréger).

8. Multi-Utilisateurs :

Les outils OLAP doivent fournir des accès concurrents, l'intégrité et la sécurité (ce n'est pas une auberge espagnole).

9. Inter Dimension :

Les calculs doivent être possibles à travers toutes les dimensions (les agrégats doivent être faits dans toutes les dimensions).

10. Intuitive Data Manipulation :

La manipulation des données se fait directement à travers les cellules d'une feuille de calcul, sans recourir aux menus ou aux actions multiples. Il doit permettre l'analyse intuitive dans plusieurs dimensions au final.

11. Flexibilité :

Lors de la création de rapports, les dimensions peuvent être présentées de n'importe quelle manière.

12. Analyse sans limites :

Dimensions et niveaux d'agrégations illimités.

- **Les fonctionnalités d'OLAP**

Plusieurs opérations OLAP peuvent être effectuées sur cette structure appelées opérations de restructuration qui sont : pivot, Slice, Dice et d'autres opérations sont liées à la granularité permettant ainsi la hiérarchisation des données. Ces opérations sont Roll up et Drill down. La Figure 1.7 montre un cube de données original.

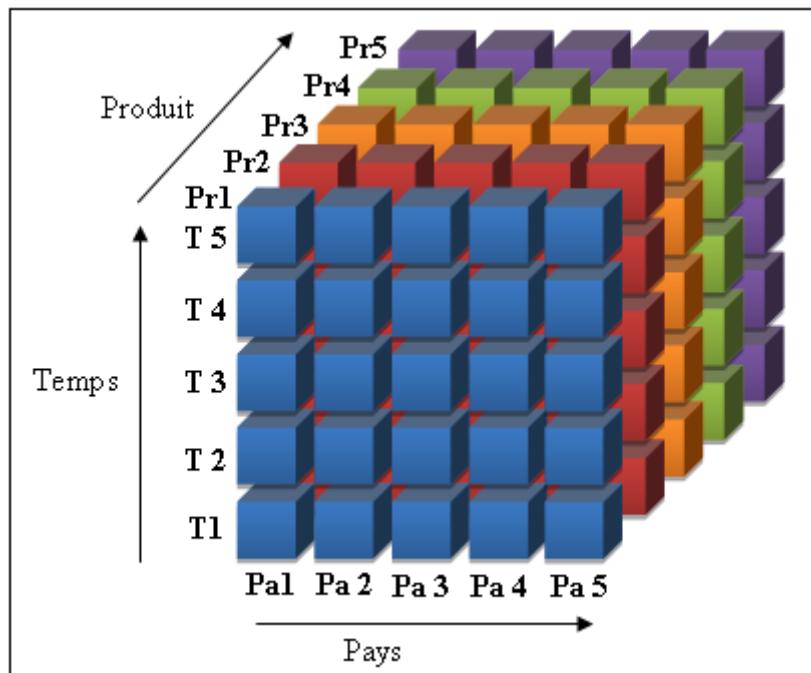


FIG. 1.7 – Un cube de données original.

1. L'opération Pivot

Elle permet d'effectuer une rotation du cube autour d'un des axes de manière à permettre une représentation d'un ensemble de faces différent, Comme le montre la (Figure 1.8).

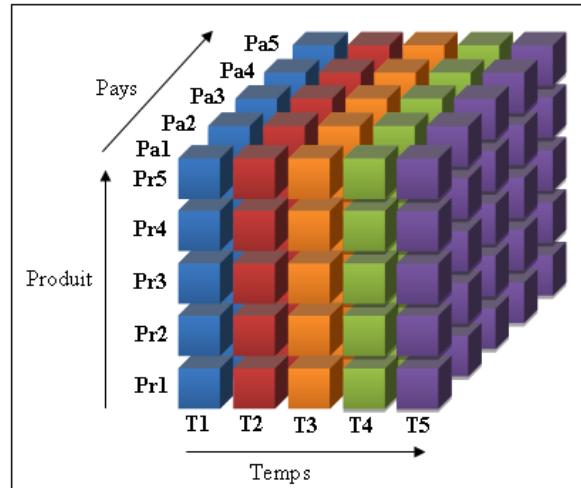


FIG. 1.8 – L'opération Pivot.

2. L'opération Slice

L'application de cette opération sur le cube original, donne une tranche de ce dernier sur une valeur particulière d'une dimension donnée, Comme le montre la (Figure 1.9).

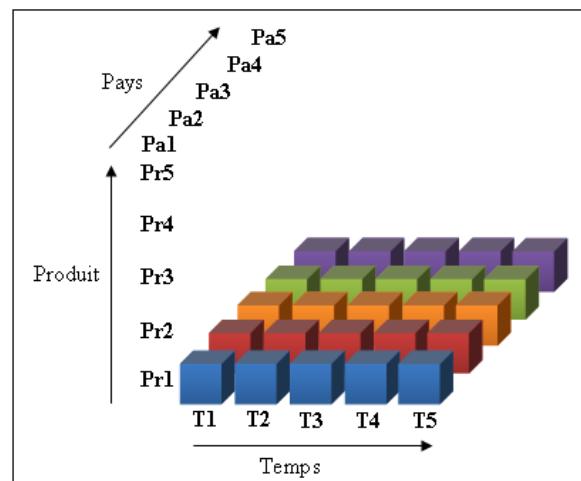


FIG. 1.9 – L'opération Slice

3. L'opération Dice

Dans ce cas, on définit un sous-cube de l'espace original. En d'autre terme, en spécifions un intervalle de valeurs sur une ou plusieurs dimensions. L'utilisateur peut retirer les blocs significatifs des données agrégés, Comme le montre la (Figure 1.10).

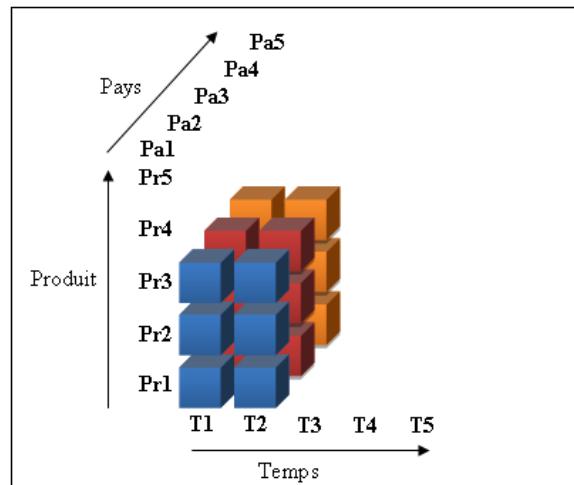


FIG. 1.10 – L'opération Dice

4. L'opération Roll up

Elle permet de visualiser les données de manière résumée (en allant d'un niveau particulier de la hiérarchie vers un niveau plus général). Comme le montre la (Figure 1.11).

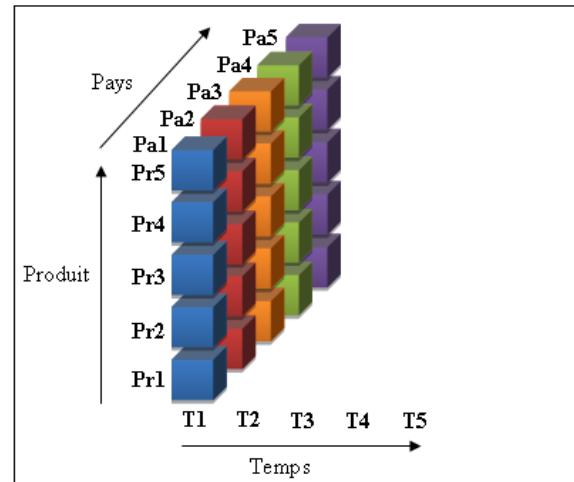


FIG. 1.11 – L'opération Roll up

5. L'opération Drill down

Permet de naviguer vers des données d'un niveau inférieur et donc plus détaillé, Comme le montre la (Figure 1.12).

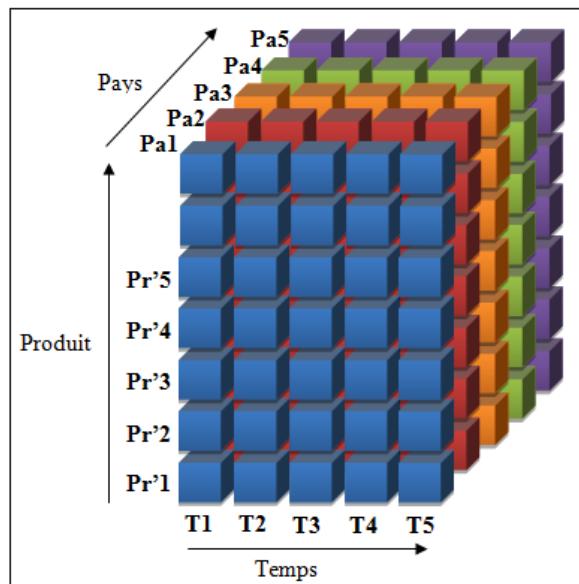


FIG. 1.12 – L'opération Drill down

- Les implémentations d'un cube multidimensionnel

Il existe deux manières principales pour construire un système basé sur un modèle multidimensionnel, selon la manière dont le cube est stocké : l'approche MOLAP et l'approche ROLAP [1.27].

1. L'approche MOLAP

Les systèmes MOLAP (Multidimensional On-Line Analytical Processing) implémentent le cube sous forme d'un tableau multidimensionnel (SGBD multidimensionnel). Chaque dimension du tableau représente une dimension du cube. Les données de chaque cellule sont stockées.

Le principal avantage d'un système MOLAP est sa performance en temps d'accès (l'accès aux données est direct). Outre le fait que ce système ne présente aucun standard, ses principaux inconvénients sont :

- Le besoin de redéfinir des opérations pour manipuler les structures multidimensionnelles.
- Difficulté de la mise à jour et de la gestion du modèle.
- Consommation de l'espace lorsque les données sont éparses, ce qui nécessite l'utilisation des techniques de compression.

2. L'approche ROLAP

Les systèmes ROLAP (RelationalOn-Line Analytical Processing) stockent les données du cube en utilisant un SGBD relationnel. Chaque dimension du cube est représentée sous forme d'une table appelée table de dimension. Chaque fait est représenté par une table de fait. Les mesures sont stockées dans les tables de faits qui contiennent les valeurs des mesures et les clés vers les tables de dimensions.

Le principal inconvénient de ces systèmes est qu'ils peuvent présenter un temps de réponse aux requêtes élevé. Leurs principaux avantages sont :

- Exploitation des capacités d'un standard bien établi et maîtrisé le relationnel, ce qui facilite leur intégration dans les SGBD relationnels existants.
 - Stockage de grandes quantités de données.
- Trois schémas sont utilisés pour modéliser les systèmes ROLAP :

- (a) Le schéma en étoile
- (b) Le schéma en flocon de neige
- (c) Le schéma en constellation [1.29]

1. Le schéma en étoile

Chaque dimension du cube est représentée par une table de dimension et les mesures par une table de faits qui référence les tables de dimension en utilisant une clé étrangère pour chacune d'elles. La table de faits est normalisée, les tables de dimension sont généralement dénormalisées. Cette représentation facilite l'analyse selon différentes perspectives. Les requêtes généralement appliquées sur ce schéma sont appelées "requêtes de jointure en étoile", et ont les propriétés suivantes :

- Il y a des jointures multiples entre la table des faits et les tables de dimension.
- Il n'y a pas de jointure entre les tables de dimensions.
- Chaque table de dimension impliquée dans une opération de jointure à plusieurs prédictats de sélection sur ses attributs descriptifs [1.27] [1.28]

La Figure1.13 présente un exemple d'un schéma en étoile.

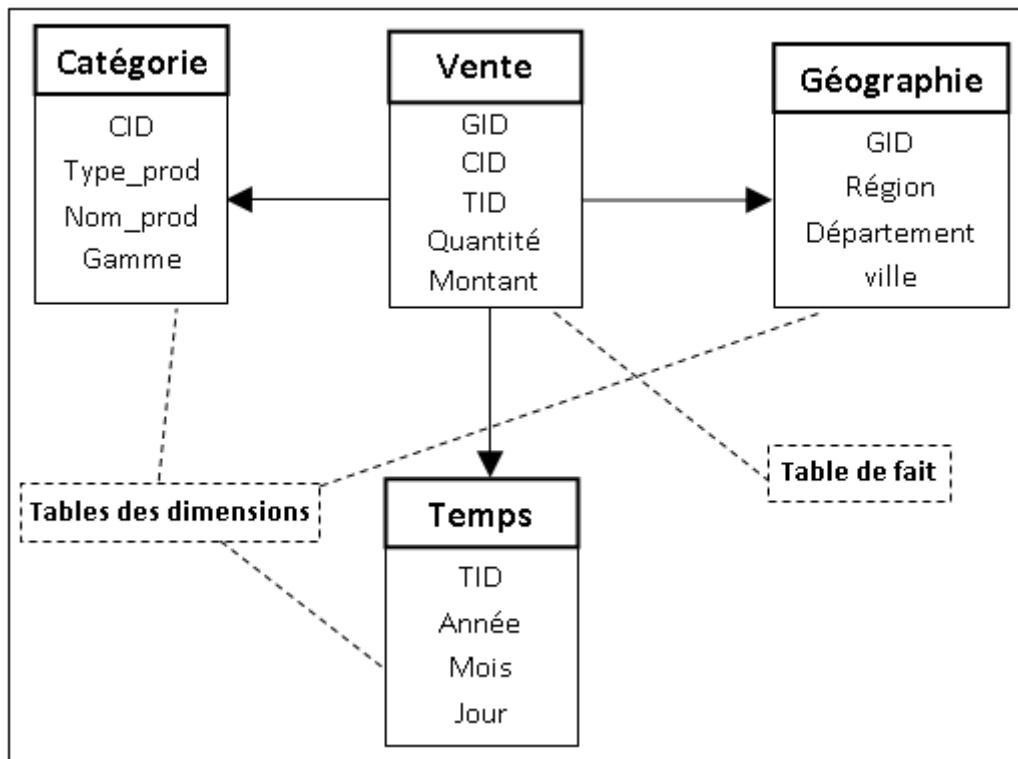


FIG. 1.13 – Le schéma en étoile

2. Le schéma en flocon de neige

Le schéma en flocon de neige est une extension du schéma en étoile. Dans un schéma en étoile, les informations associées à une hiérarchie de dimension, sont représentées dans une seule table, même si les différents niveaux de la hiérarchie ont des propriétés différentes. Le schéma en flocon est le résultat de la décomposition d'une ou plusieurs dimensions en plusieurs niveaux formant une hiérarchie. Les tables de dimensions sont ainsi éclatées en plusieurs tables, ce qui peut être vu comme une normalisation des tables de dimensions. La table de faits reste inchangée. Ce type de schéma offre une meilleure visualisation et compréhension des données, mais peut altérer les performances de l'entrepôt lors de son utilisation. En effet, une requête nécessitera plusieurs jointures ce qui augmente son temps de réponse. La Figure 1.14 présente un exemple d'un schéma en flocon de neige.

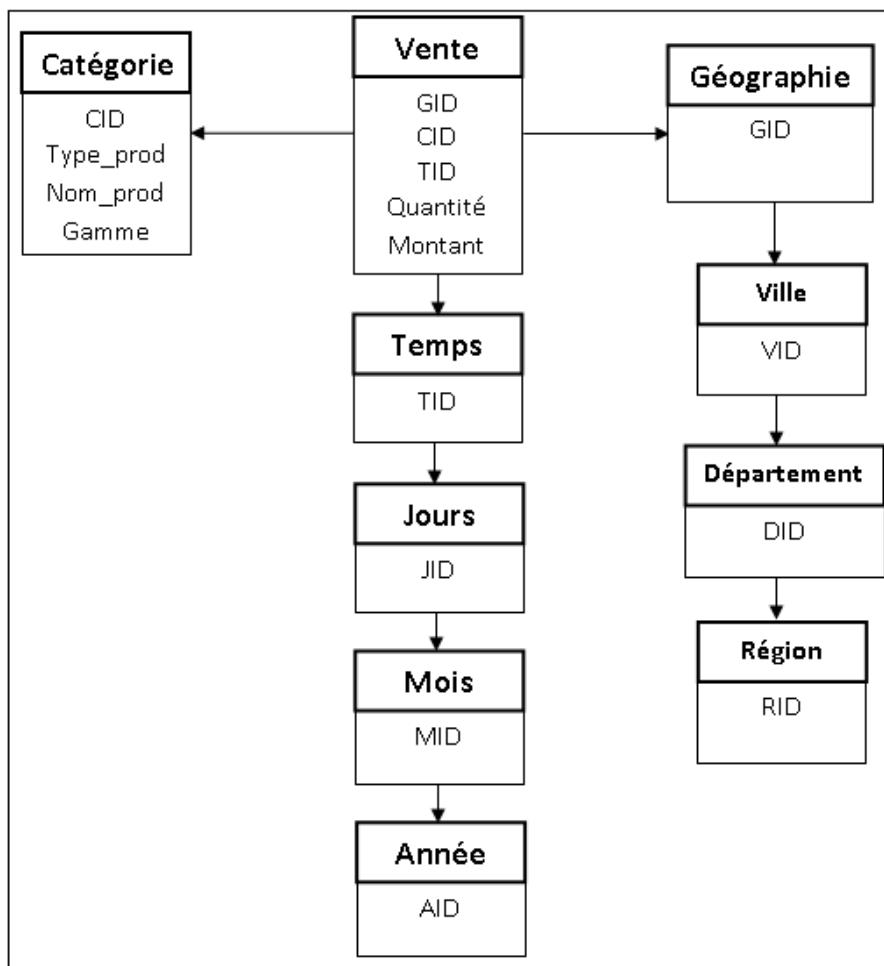


FIG. 1.14 – Le schéma en flocon de neige

3. Les schémas en constellations

Sont des schémas où plusieurs modèles dimensionnels se partagent les mêmes dimensions, c'est-à-dire les tables de faits ont des tables de dimensions en commun. Les tables de dimensions partagées par plusieurs tables de fait doivent être exactement les mêmes. La Figure 1.15 présente un exemple d'un schéma en constellation.

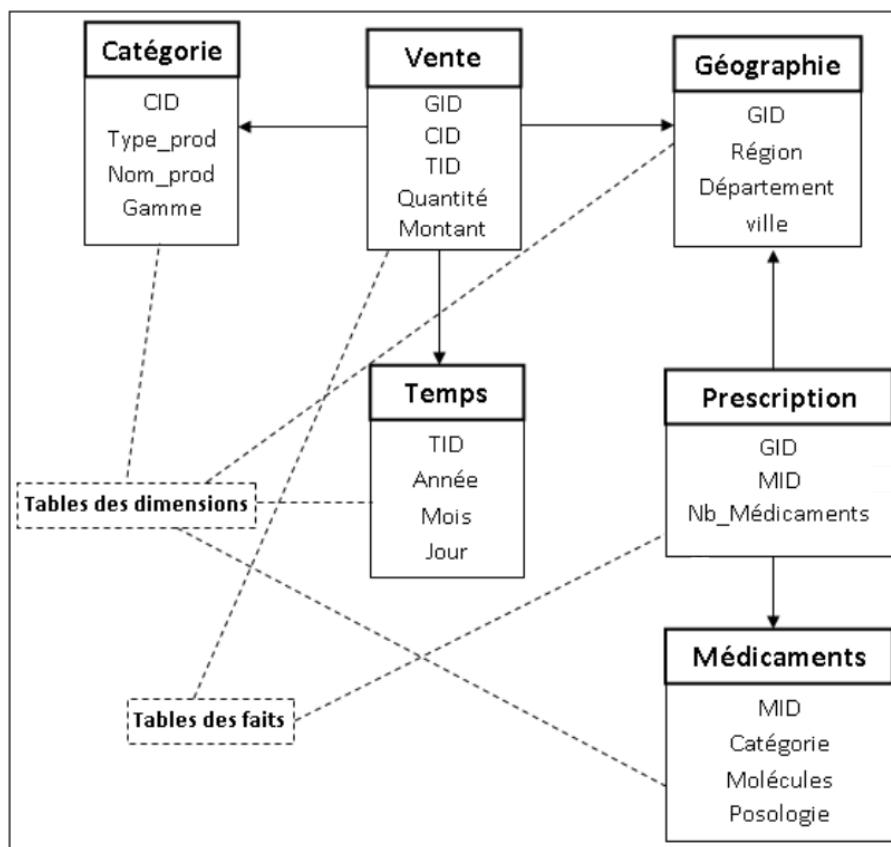


FIG. 1.15 – Le schéma en constellation

3. L'approche HOLAP

Les systèmes HOLAP (HybridOn-Line Analytical Processing) sont des systèmes où les données fréquemment utilisées (généralement les données agrégées) sont maintenues par un SGBD multidimensionnel, et les données non fréquemment utilisées dans un SGBD relationnel. Ceci afin de bénéficier des avantages des deux systèmes cités précédemment. La séparation des données doit être transparente à l'utilisateur final.

Mendelzon propose une comparaison entre les deux approches ROLAP et MOLAP :

Stockage	Avantages	Inconvénients
ROLAP	Technologie familiale	Lenteur
ROLAP	capacité d'expansion élevée	-
ROLAP	Ouvert	-
MOLAP	Modèle multidimensionnel	Technologie non prouvée
MOLAP	Traitement de requête spécialisé	Non scalable
MOLAP	Techniques d'indexation spécialisées	-

TAB. 1.2 – comparaison entre ROLAP et MOLAP [1.27]

Conclusion

Nous avons présenté dans ce chapitre quelques concepts de base sur les bases de données, à savoir l'historique, la définition, l'architecture etc. Et quelques concepts d'entrepôts de données, à savoir l'architectures, la modélisation, le cube de données etc.

Dans le chapitre suivant nous décrirons plus en détail les quelques outils d'implémentation d'entrepôt de données.

CHAPITRE 2

Outils d'implémentation d'ED

Introduction

Plus une entreprise prend de l'envergure, plus il lui est difficile de prendre des bonnes décisions. Il est ainsi indispensable de faire l'acquisition d'un logiciel de gestion d'entreposage avec lequel il lui sera aisément d'effectuer toutes les opérations de gestion de stock dans les meilleures conditions. Dans ce chapitre, nous allons présenter les outils d'entreposage les plus utilisés, à savoir leurs utilités, fonctionnalités et avantages.

2.1 SQL Server

Le SQL server est un système de gestion de base de données et aussi un outil d'entreposage de données utilisé pour la mise en place d'applications de gestion et décisionnelles ainsi la mise en place de data warehouses complexes et optimisés (de dix à quelques centaines de To de volume) [2.1].

2.1.1 Historique

Bien qu'il ait été initialement développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

En 1994, le partenariat entre les deux sociétés ayant été rompu, Microsoft a sorti la version 6.0 puis 6.5 seul, sur la plate-forme Windows NT.

Microsoft a continué de commercialiser le moteur de base de données sous le nom de SQL Server. Tandis que Sybase, pour éviter toute confusion, a renommé Sybase SQL Server en Sybase Adaptive Server Enterprise.

Microsoft SQL Server fait désormais partie de la stratégie technique de Microsoft en matière de

base de données. Le moteur MSDE¹, qui est la base de SQL Server, doit à terme remplacer le moteur Jet (celui qui gère les bases Access) dans les applications telles que Exchange et Active Directory. [2.1]

La version 2005 de SQL Server est sortie le 3 novembre 2005 en même temps que Visual Studio 2005. La prise en charge de Windows Vista et de Windows Server 2008 n'a été ajoutée qu'à partir du Service Pack 2 (SP2). Actuellement le Service Pack 3 est disponible.

La version 2008 de SQL Server (nom de code Katmai) est disponible depuis août 2008. Elle est actuellement au niveau de service pack 3. Elle est disponible en 9 langues, dont le français.

La version 2012 de SQL Server est disponible depuis avril 2012 [2.2].

2.1.2 Définition

SQL Server est un système de gestion de base de données relationnelle développé par Microsoft principalement conçu pour rivaliser contre des concurrents Oracle Database (DB) et MySQL. Comme tous les grands SGBDR, SQL Server prend en charge la norme ANSI SQL², le langage SQL standard. Toutefois, SQL Server contient également des Transacts-SQL, son propre implémentation SQL. SQL Server Management Studio (SSMS) (auparavant connue sous le nom de l'entreprise) est l'outil de l'interface principale de SQL Server, et il prend en charge les environnements 32 bits et 64 bits. SQL Server est parfois appelé MSSQL et Microsoft SQL Server. [2.2]

2.1.3 Les fonctionnalités principales de Microsoft SQL Server

SQL Server est offert en plusieurs éditions avec différents ensemble de fonctionnalités et options de tarification pour répondre à une variété de besoins des utilisateurs, y compris les suivantes :

- Enterprise : Conçu pour les grandes entreprises ayant des besoins complexes de données, entrepôts de données et bases de données Web-enabled. A toutes les fonctionnalités de SQL Server, et le prix de la licence est le plus cher.
- Standard : ciblée vers les petites et moyennes organisations. Prend également en charge le commerce électronique et de l'entreposage de données. [2.1]
- Groupe de travail : Pour les petites organisations. Peuvent être utilisés comme base de données dorsale pour les serveurs Web de petite taille ou des succursales.
- Express : gratuit pour la distribution. Il a le plus petit nombre de caractéristiques et des limites de taille et utilisateurs base de données. Peut être utilisée comme un remplacement pour une base de données d'accès.

En générale les fonctionnalités de Microsoft SQL Serveur se résument comme suit :

- Gestion de bases de données relationnelles.
- Gestion et déploiement centralisé de plusieurs instances et applications depuis un seul point de contrôle.
- Optimisation de stockage des bases de données volumineuses (tables et indexes partitionnées, compression de données, etc).

¹(Microsoft SQL Server Desktop Engine) est une édition limitée mais gratuite de Microsoft SQL Server.

²est devenu une norme de l' American National Standards Institute (ANSI) en 1986

- Prise en charge des données géographiques.
- Gestion de la haute disponibilité.
- Ordonnanceur intégré (SQL Agent).
- Service de notification.
- Gestion de la réPLICATION.
- Prise en charge de la virtualisation [2.2].

2.1.4 Les avantages de Microsoft SQL Server

Les avantages d'SQL Server se résument comme suit :

- SQL Server intègre par défaut des outils de gestion, d'administration et de développement de bases de données.
- Gestion avancée de la sécurité en offrant deux modes d'authentification (Authentification Windows et Authentification SQL Server).
- Prise en compte des spécificités des projets décisionnels (Parallélisations de Datawarehouses etc).
- mise en oeuvre et administration par des interfaces graphiques intuitives.
- Coût relativement moins cher par rapport aux autres SGBD du marché. [2.2] [2.3]

2.1.5 Les inconvénients de Microsoft SQL Server

Lorsque vous modifiez les données d'une table indexée, SQL Server doit mettre à jour les index associés ainsi que La gestion des index demande du temps et des ressources et dans le cas des petites tables, les index présentent peu d'avantages car il peut être plus " coûteux " de parcourir les pages d'index que d'analyser la table dans son intégralité. [2.3]

2.2 Oracle

Oracle est un système de gestion de base de données et aussi un outil d'entreposage de données, il présente une nouvelle architecture multipropriétaires facilitant le déploiement et la gestion de Clouds³ de base de données. Les innovations apportées par ce produit, tel qu'Oracle Multi-tenant⁴, qui permet de consolider rapidement plusieurs bases de données, ainsi que l'optimisation automatique des données avec HeatMap⁵, qui permet de compresser et de hiérarchiser les données avec une plus grande densité, maximisent l'efficacité et la flexibilité des ressources. Ces nouveautés uniques, combinées aux améliorations majeures apportées à la disponibilité, la sécurité et la prise en charge des Big Data, font d'Oracle Database la plateforme idéale pour réaliser des déploiements de Cloud privé et public [2.4].

³Désigne l'utilisation de serveurs distants (en général accessibles par Internet)

⁴Architecture logicielle de base de données

⁵Une carte de chaleur est une représentation graphique de données

2.2.1 Historique

Software Development Laboratories a été créé en 1977. En 1979, SDL(Simple DirectMedia Layer) change de nom en devenant Relational Software Inc (RSI) et introduit son produit Oracle version 2 comme base de données relationnelle. La version 2 ne supportait pas les transactions mais implémentait les fonctionnalités SQL basiques de requête et jointure. Il n'y a jamais eu de version 1, pour des raisons de marketing, la première version a été la version 2. Celle-ci fonctionnait uniquement sur les systèmes DigitalVAX/VMS⁶.

En 1983, RSI devient Oracle Corporation pour être plus représentative de son produit phare. La version 3 d'Oracle, entièrement réécrite en langage de programmation C, est publiée. Celle-ci supportait les transactions grâce aux fonctionnalités de commit⁷ et rollback⁸. C'est aussi à partir de cette version que la plate-forme Unix est supportée. [2.5]

En 1984, la version 4 d'Oracle apparaît, supportant la cohérence en lecture (readconsistency).

Début 1985, Oracle commence à intégrer le modèle client-serveur, avec l'arrivée des réseaux au milieu des années 1980. La version 5 d'Oracle supporte donc les requêtes distribuées.

En 1988, Oracle met sur le marché son ERP⁹ - Oracle Financial basé sur la base de données relationnelle Oracle. [2.6] Oracle version 6 supporte le PL/SQL, le verrouillage de lignes (row-levellocking) et les sauvegardes à chaud (hot backups, lorsque la base de données est ouverte).

En 1992, la version 7 d'Oracle supporte les contraintes d'intégrité, les procédures stockées et les déclencheurs (triggers). En 1995, acquisition d'un puissant moteur multidimensionnel, commercialisé sous le nom d'Oracle Express.

En 1997, la version 8 introduit le développement orienté objet et les applications multimédia.

En 1999, la version 8i est publiée dans le but d'affiner ses applications avec Internet (le i fait référence à Internet). La base de données comporte nativement une machine virtuelle Java.

En 2001, Oracle 9i ajoute 400 nouvelles fonctionnalités et permet de lire et d'écrire des documents XML. [2.5]

À partir de la version 9i, intégration du moteur OLAP au sein d'Oracle : le moteur Oracle express est dorénavant référencé au sein de l'option Oracle OLAP. Les données multidimensionnelles sont accessibles à partir du langage SQL.

En 2003, la version 10g est publiée, version qui supporte les expressions rationnelles. Le g signifie " grid " ; un des atouts marketing de la 10g est en effet qu'elle supporte le " gridcomputing ".

En 2005, vers la fin novembre, une version complètement gratuite est publiée, la " Oracle Database 10g Express Edition ". Apparaît cette même année, Oracle Database 10.2.0.1 aussi connue sous le nom d'Oracle Database 10g Release 2 (10gR2).

En 2007, sortie de la version 11 : Oracle Database 11g, pour Linux et Windows.

Septembre 2009, sortie de Oracle Database 11g Release 2.

1 juillet 2013, sortie d'Oracle Database 12c. [2.6]

⁶Un jeu d'instructions d'architecture

⁷Validation de transaction qui fait référence à la commande synonyme

⁸Sert à annuler une transaction pour une session ouverte.

⁹Entreprise Resource Planning est une Planification des ressources de l'entreprise

2.2.2 Définition

Oracle est un système de gestion de base de données relationnel (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates. [2.7] Oracle est écrit en langage C et est disponible sur de nombreuses plates-formes matérielles (plus d'une centaine) dont :

- AIX (IBM)
- Solaris (Sun)
- HP/UX (Hewlett Packard)
- Windows NT (Microsoft)

Oracle depuis la version 8.0.5 est disponible sous Linux [2.8]

2.2.3 Les fonctionnalités principales d'Oracle

Sa fonction principale est de gérer d'une façon intégrée l'ensemble de données d'une entreprise et de les rendre accessibles à un nombre important d'utilisateurs en garantissant leur sécurité, leur cohérence et leur intégrité.

En générale les fonctionnalités d'oracle se résument comme suit :

- La définition et la manipulation des données.
- La cohérence des données.
- La confidentialité des données.
- L'intégrité des données.
- La sauvegarde et la restauration des données.
- La gestion des accès concurrents [2.6].

2.2.4 Les avantages d'Oracle

Les avantages d'Oracle se résument comme suit :

- Offre une sécurité de premier plan, une réelle disponibilité et évolutivité grâce à Oracle Database 11g, qui a été amélioré pour permettre de tirer parti des serveurs de stockage Oracle Exadata.
- Possibilité de choisir entre une installation automatique ou paramétrier son installation à 100%.
- Gestion entièrement automatique de la mémoire.
- Gestion avancée de la compression des données.
- Très performant sur des gros volumes de données[2.8].

2.2.5 Les inconvénients d'Oracle

- Incapacité de mettre en oeuvre un traitement récursif.
- Incompatibilité et complexité.
- Fonctionnalité limitée.
- Coût [2.7] [2.8].

2.3 Pentaho

Pentaho est un outil d'entrepôts de données utilisé pour la migration de données d'une base à une autre, l'alimentation d'un datawarehouse et de datamarts.

2.3.1 Historique

Pentaho est né de la volonté d'un changement positif, marquant une rupture sur le marché de l'analyse d'entreprise dominé par des méga-fournisseurs bureaucratiques offrant des produits extrêmement chers et lourds, basés sur des plates-formes technologiques dépassées, et qui se concentrent désormais uniquement sur l'intégration avec le reste de leurs suites logicielles au détriment de l'innovation de leurs fonctionnalités BI(Business Intelligence). [2.9]

Cinq spécialistes chevronnés du secteur ont cherché une solution à cette énigme. Aujourd'hui, avec la même équipe fondatrice qui montre le chemin, Pentaho a ouvert la voie de la future Business Intelligence et de l'analyse grâce à une suite complète de fonctionnalités pour l'intégration et l'accès aux données, la recherche, l'analyse et la visualisation. Ces fonctionnalités flexibles peuvent être déployées sur site, dans le nuage de serveurs ou intégrées dans d'autres applications, et permettent d'accéder virtuellement à toutes les sources de données, des tableurs aux données en masse.

Pentaho est maintenant le premier fournisseur indépendant d'analyse d'entreprise pour des milliers d'entreprises mondiales et fournisseurs de logiciels avec un coût jusqu'à 90 La Business intelligence connaît des changements considérables qui transforment radicalement la façon dont le secteur et les clients conçoivent l'analyse. La croissance exponentielle des données en masse, stimulée par la grande prolifération des données, est le moteur clé de ce changement. La tendance à l'adoption de l'informatique en nuage dans les entreprises exerce également une pression sur les fonctionnalités d'analyse d'entreprise, les incitant à incorporer toutes les données pertinentes à partir de diverses sources de données, afin de permettre aux utilisateurs finaux de prendre des décisions plus globales et plus approfondies, au moment approprié. De plus, la tendance à la consumérisation des technologies de l'information entraîne toujours plus d'attente en termes de disponibilité et de facilité d'utilisation des applications. Elles doivent pouvoir être utilisées partout, à tout moment et de la façon dont les utilisateurs le souhaitent. [2.10]

Cela se traduit par de nouvelles exigences des clients en matière d'analyse d'entreprise. Ces der-

niers souhaitent que les nouveaux outils et processus puissent collecter rapidement et facilement tout type de données et stocker, gérer, manipuler, agréger, analyser et intégrer toutes ces données de manière à ce qu'elles engendrent un impact positif pour l'entreprise. Pentaho construit le futur de l'analyse d'entreprise. Pentaho lie étroitement l'intégration de données à l'analyse d'entreprise dans une plate-forme moderne qui permet aux services informatiques et aux utilisateurs d'accéder, de rechercher et d'analyser facilement toutes les données ayant un impact sur les résultats de l'entreprise. [2.11]

2.3.2 Définition

Pentaho est une solution d'informatique décisionnelle open source entièrement développée en Java. Elle porte sur toute la chaîne décisionnelle et utilise différents outils et composants :

- Pour la collecte et l'intégration : les outils d'ETL (Extract-Transform-Load) Kettle ou Mondrian.
- Pour la diffusion : un serveur d'application JBoss ou TOMCAT.
- Pour la présentation : JFreeReport, BIRT ou encore JasperReport. [2.12]

2.3.3 Les fonctionnalités principales de Pentaho

Pentaho se décrit comme " le premier grand fournisseur de solutions décisionnelles à proposer des fonctions pour le Big Data ". Il offre ses services sur le secteur de l'informatique décisionnelle open source depuis 2004.

Son activité principale est la description de l'exploitation des données d'une entreprise pour, ainsi, faciliter la prise de décision. Cette exploitation de données repose sur des informations provenant de sources multiples, concernant l'entreprise ou les acteurs externes au marché économique qui l'entourent. [2.14]

Grâce aux outils d'analytique, ces données sont modélisées afin d'élaborer des tableaux de bord permettant leur lecture. Sur son site, Pentaho explique qu'il permet " aux services informatiques et aux utilisateurs métiers d'accéder, de visualiser et d'explorer facilement toutes les données ayant un impact sur les résultats ". Pentaho est donc une suite logicielle décisionnelle complète qui permet la création et la diffusion de documents décisionnels à un grand nombre de destinataires par via une interface web.

Pentaho est entièrement open source, et propose une souscription pour des services éditeurs haut de gamme, incluant un support illimité et des différentes garanties. [2.15]

En effet, Pentaho comprend tous les éléments nécessaires pour un projet décisionnel :

- Un moteur de reporting, avec Pentaho Report Designer, un logiciel graphique de conception de rapports
- Un outil de conception de metamodels, avec Pentaho Metadata Designer, outil de définition métier d'une base de données relationnelle, permettant de requêter sur cette base sans connaissance technique.
- Un moteur d'analyse multidimensionnelle, avec Pentaho Analysis (Mondrian), un serveur OLAP utilisé par toutes les suites décisionnelles concurrentes mais maintenu par Pentaho. Il permet d'analyser un très grand volume de données, toujours très simplement pour l'utilisateur.
- Un accès aux documents décisionnels via une console web pour les utilisateurs, ergonomique, multi-onglets et sécurisé.
- Une console d'administration et de supervision, gérant entre autres la diffusion en masse des rapports par email.
- Un outil d'alimentation et de transformation, avec Pentaho Data Integration (ex Kettle), ETL très simple à prendre en main. Un outil de datamining, avec Pentaho Weka.
- Pentaho dispose également d'un client Ms-Excel pour Mondrian et d'un connecteur avec Google Maps, ...etc[2.16].

Pentaho est une solution complète d'ETL comprenant :

- Bibliothèque constituée de 50 objets de mapping.
- Données avancées pour le datawarehousing.
- Exécution et Entreprise-classe scalability.
- Connecteur de SAP ¹⁰.

L'intégration de données de Pentaho est ouverte, repose sur une architecture normalisée et est ajustable à n'importe quel environnement ou solution de BI [2.18].

2.3.4 Les avantages de Pentaho

Les avantages de Pentaho se résument comme suit :

- Pentaho Data Integration est gratuit et Open Source.
- Un large appui de point d'émission de données.
- Réutilisation facile des questions et des composants de transformation avec les modèles.
- Exécution groupée des transformations d'ETL.
- Entièrement intégré avec la suite open source de Pentaho.
- Entièrement basé sur du Java (des scripts personnalisés sous forme de script java).

¹⁰Progiciel de gestion d'entreprise

2.3.5 Les inconvenants de Pentaho

- Il ne dispose pas de plateforme d'exécution et de contrôle des traitements fournis. Ceux-ci doivent être lancés en ligne de commande (outil fourni) au travers de schedulers.
- Manque de gestionnaire de traitements pour gérer le changement d'environnement (test/production).

2.4 Talend open studio

Talend open studio est un outil d'entrepôts de données utilisé pour l'échange Inter-Application (EAI), la migration de vieux applicatifs vers de nouveaux, le contrôle de l'information et l'alimentation du décisionnel.

2.4.1 Historique

Crée en 2005, Talend est une société anonyme¹¹ ayant des bureaux notamment aux États-Unis, en France, en Angleterre, en Allemagne, en Chine et au Japon. Talend est l'un des membres fondateurs d'OW2¹² et de l'Open Solutions Alliance¹³. Il a été élu au Conseil d'Administration de AFDEL (Association Française Des Éditeurs de Logiciel). En novembre 2010, Silver Lake Partners a annoncé investir 34 millions dans Talend. Les différents versions de Talend Open Studio sont les suivantes :

- La version v 0.5.1(25/10/2006) : Distribution de la version française de Talend Open Studio Guide Utilisateur.
- La version v 0.5.5(21/01/2007) : Suite à la distribution de Talend Open Studio version 1.1.0, mise à niveau des informations générales du guide utilisateur version française avec celles du guide version anglaise.
- La version v 0.7(05/03/2007) : Mise à niveau des composants. Version 0.7 du Guide Utilisateur anglais. [2.19] [2.20]

2.4.2 Définition

Talend Open Studio est un ensemble puissant et flexible de produits Open source pour développer, tester, déployer et administrer des projets de gestion de données et d'intégration d'applications. Talend propose la seule plateforme unifiée simplifiant la gestion de données et l'intégration d'applications en fournissant un environnement unifié pour gérer le cycle de vie complet à travers les frontières de l'entreprise. La productivité des développeurs est considérablement améliorée grâce à un environnement graphique simple à utiliser, basé Eclipse, combinant l'intégration de

¹¹peut être connue sans être identifiée ou au contraire être identifiée sans être connue

¹²Association (loi de 1901) internationale à but non lucratif dédiée au développement d'intergiciels libres de qualité industrielle.

¹³Open Solutions : Grandes entreprises dédiées à la production de solutions de logiciels libres de classe entreprise de travailler ensemble.

données, la qualité de données, le MDM¹⁴, l'intégration d'applications et le Big Data.

Les produits Talend abaissent considérablement les barrières d'adoption pour les entreprises qui souhaitent des solutions packagées puissantes pour relever les défis opérationnels tels que le nettoyage de données, la gestion de données maître, et le déploiement d'un Enterprise Service Bus. Tirant parti des technologies Apache et les étendant, les solutions Talend d'ESB Open source et de SOA Open source permettent aux entreprises de construire des architectures flexibles et de hautes performances intégrant et convertissant en services des applications distribuées.

Tous les produits Talend Open Studio sont 100% Open source. Il est possible de les télécharger et de les utiliser gratuitement. La Communauté Talend fournit un ensemble de services d'apprentissage et de support, notamment un forum de discussion, des tutoriels, le suivi des bogues, le code source et les add-ons des produits. [2.21]

2.4.3 Les fonctionnalités principales de Talend Open Studio

- Business Modeler : permet de modéliser les besoins métiers d'un point de vue non technique mais fonctionnel (nécessite tout de même une excellente vision de l'ensemble du projet)
- Déploiement et maintenance : Exportation des jobs sous forme de fichier .jar, .exe et .sh.
- Centralisation des métadonnées (description des données).
- Génération d'une documentation de qualité au format HTML.
- Réutilisation du code : routines.
- Réutilisation des requêtes SQL génériques : Modèles SQL.
- Environnement de développement entièrement modulable.
- Gestion des logs et erreurs.
- Versionning de jobs.
- Définition de contextes d'exécution (développement, test, production) [2.21].

2.4.4 Les avantages de Talend Open Studio

Les avantages de Talend Open Studio se résument comme suit :

- Le nombre de composants est proprement impressionnant, et la forge constitue un réservoir à fonctionnalités supplémentaires .
- Son approche permet de se passer quasiment totalement de connaissances en Java. Dès qu'on souhaitera écrire des expressions, il faudra bien évidemment connaître la base, mais on est très loin de connaissances approfondies, et un grand débutant Java pourra tout-à-fait être productif avec Talend sans difficultés .
- Le choix de Java comme langage de génération de code permet de s'assurer un script final

¹⁴(Master Data Management) : gestion des données de référence, une discipline informatique qui concerne les données de référence partagées

multi-plateforme. Un script créé, testé, et compilé sous Windows s'exécutera parfaitement sous un environnement Linux .

- Il est gratuit, et son code source est disponible, mais il est en même temps supporté par un éditeur et dispose de déclinaisons payantes avec un meilleur support et des fonctionnalités inédites. Cela signifie que quel que soit votre engagement et votre budget, il pourra vous convenir. [2.22].

2.4.5 Les inconvénients de Talend Open Studio

- Des bugs sont parfois présents. Du chemin a été fait, et dans la version 4.x ils sont maintenant rares, mais encore dans la version précédente il existait des bugs bloquant carrément le déploiement.
- L'ergonomie n'est pas toujours au top, et c'est un euphémisme. La version 5 semble toutefois aller dans le bon sens sur ce point.
- Le mode "clicodrome" a ses limites : difficile de versionner les projets, difficile d'expliquer son problème sur un forum sans passer par des screenshots¹⁵ (ce qui rend compliqué l'accès au support sur le forum), et le nombre de clics pour réaliser un job compliqué devient vite incalculable.
- L'effet Eclipse : l'éditeur est très lourd, Pour arranger ça, on peut limiter le nombre de composants dans la palette. [2.23]

Conclusion

Dans ce chapitre nous avons présenté les différents outils d'implémentation d'entrepôt de données les plus utilisées à savoir SQL Server, Oracle, Pentaho et Talend Open Studio, en spécifiant : leurs historiques, définitions, fonctionnalités principales, avantages et inconvénients.

Dans le chapitre suivant nous entamerons une comparaison entre ces outils pour mieux choisir l'outil le plus adapté pour notre application.

¹⁵Une capture d'écran

CHAPITRE 3

Comparaison entre les Outils d'ED

Introduction

Vue l'existence de plusieurs outils d'entreposage de données, certains libres (pentaho ou Talend Open Studio pour l'ETL) et d'autres des SGDB commerciaux (SQLServer, Oracle), dans ce chapitre, on va faire une comparaison entre ces outils pour choisir l'outil le plus adapté pour notre application.

3.1 Les paramètres de comparaisons

Afin de mieux choisir un outil d'implémentation d'entrepôt de donnée, nous allons discuter les paramètres les plus importants à prendre en considération dans cette comparaison.

3.1.1 paramètres de forme

Parmi les paramètres de comparaison, nous avons commencé par les paramètres de forme qui nous donne une vue générale sur l'outil :

- Licence : contrat par lequel le titulaire des droits d'auteur sur un outil définit avec son cocontractant (exploitant ou utilisateur) les conditions dans lesquelles cette outil peut être utilisé.
 - Propriétaire : désigne toute licence qui n'est pas une licence libre.
 - Licence publique générale limitée GNU Version 2.1 (LGPLv2.1) : est une licence utilisée par certains logiciels libres. Pour les logiciels du projet GNU, nous utilisons uniquement des licences compatibles avec la GNU GPL.
- Editeur : Un éditeur de logiciel est une entreprise qui assure la conception, le développement et la commercialisation de progiciels (logiciels standards). Elle peut confier la mise en œuvre, l'intégration, et la personnalisation ("customisation" en anglais) à des sociétés de services et d'ingénierie informatique.

- Langage utilisé pour son développement.

3.1.2 Capacité

Chaque outil supporte un certain nombre de caractéristiques de donnée, pour cela nous avons choisi les paramètres suivant :

- Taille maximale de base de données.
- Taille maximale de table.
- Taille maximale d'une ligne.
- Nombre maximum de colonnes par table.
- Taille maximale de la valeur binaire.
- Taille maximale de la valeur de caractère.
- Taille maximale de valeur numérique.
- Taille maximale de nom de la colonne.

3.1.3 Les étapes d'implémentation d'un ED assuré par l'outil

Les phases d'un projet d'entrepôt de données sont similaires à ceux de la plupart des projets de base de données, ils commencent par l'identification des besoins et se terminent par l'exécution du script T-SQL, le cube OLAP et la phase destiné seulement pour l'entrepôt de données.

1. Identifier et collecter les exigences.
2. Concevoir le modèle dimensionnel.
3. Exécuter des requêtes T-SQL pour créer et peupler les dimensions et les tables de faits.
4. Création d'un cube OLAP [3.1].

3.1.4 Type d'index supporté

Un index est une structure de données utilisée et entretenue par le système de gestion de base de données (SGBD) pour lui permettre de retrouver rapidement les données. L'utilisation d'un index simplifie et accélère les opérations de recherche, de tri, de jointure ou d'agrégation effectuées par le SGBD.

L'index placé sur une table va permettre au SGBD d'accéder très rapidement aux enregistrements, selon la valeur d'un ou plusieurs champs. les différents types d'index sont les suivants :

- Cluster tables : trient et stockent les lignes de données de la table ou la vue en fonction de leurs valeurs de clé.
- Expression : permet de référencer ou extraire les éléments sélectionnés d'une matrice ou d'un vecteur.
- Bitmap : est une structure de données, en particulier un tableau de données binaires. - Full-text : recueille, analyse et stocke les données pour faciliter rapide et précise la recherche d'information .
- Partial : aussi connu comme index filtré est un index qui a un certain état qui lui est appliquée afin qu'il comprend un sous-ensemble des lignes de la table. Cela permet à l'indice de rester faible,

même si la table peut être relativement grande, et d'avoir une sélectivité extrême.

- Spatial : utilisée par les bases de données pour optimiser les calculs impliquant des positionnements ou des distances. [3.2]

3.1.5 Fonctions UDF(User Defined Function) supportées

Une fonction définie par l'utilisateur fournit un mécanisme pour étendre la fonctionnalité du serveur de base de données en ajoutant une fonction qui peut être évaluée dans les instructions SQL. Une fonction scalaire retourne une seule valeur (ou NULL), tandis qu'une fonction de table renvoie un tableau (relationnelle) comprenant zéro ou plusieurs lignes, chaque ligne avec une ou plusieurs colonnes. [3.3]

3.1.6 Procédures stockées

Une procédure stockée est un ensemble d'instructions SQL précompilées, stockées dans une base de données et exécutées sur demande par le SGBD qui manipule la base de données. Les procédures stockées peuvent être lancées par un utilisateur, un administrateur ou encore de façon automatique par un événement déclencheur ("trigger"). [3.3]

3.1.7 Triggers

Un trigger est un programme qui se déclenche automatiquement suite à un événement, ou bien un événement qui provoque l'exécution d'un algorithme. [3.5]

3.1.8 Partitionnement des tables

Une partition est une division logique d'une table stockée en plusieurs parties indépendantes. Le partitionnement des tables est généralement effectué pour améliorer la gestion, la performance ou la disponibilité. Chaque partition se retrouve sur des serveurs ou des disques différents. Cela permet également d'obtenir une capacité de base de données supérieure à la taille maximale des disques durs ou d'effectuer des requêtes en parallèle sur plusieurs partitions.

Les différentes catégories de partitionnements sont les suivant :

- Range : définit les partitions par tranche supérieure exclusive (toutes les données < 0).
- Hash : la partition de stockage est calculée dynamiquement par un calcul de type hash code, ce mode impose un nombre de partitions $2n$.
- Composite : méthode de partitionnement hybride. Les données sont d'abord partitionnées by range. Ensuite, chaque partition sera sous-partitionnée soit by hash ou by list. Il n'est pas nécessaire d'avoir exactement le même nombre de sous-partitions par partition : par exemple, une partition peut être constituée de 4 sous-partitions alors qu'une autre sera composée de 5 sous-partitions.
- List : définit une valeur par partition (utilisé lorsque la liste de valeur pour le champ considéré est faible). [3.7]

3.1.9 Protection de la force brute

C'est une protection contre l'attaque par force brute qui est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Cette attaque consiste à tester, une à une, toutes les combinaisons possibles et elle permet de casser tout mot de passe en un temps fini indépendamment de la protection utilisée. [3.3]

3.1.10 Transaction

Une transaction telle qu'une réservation, un achat ou un paiement est mise en oeuvre via une suite d'opérations qui font passer la base de données d'un état A (antérieur à la transaction) à un état B postérieure des mécanismes permettent d'obtenir que cette suite soit à la fois atomique, cohérente, isolée et durable. [3.4]

3.1.11 Cursors

Le terme curseur est employé comme synonyme de pointeur (de données). [3.3]

3.1.12 Vues

Une vue est une table virtuelle calculée à partir des tables de base par une requête. elle permet d'améliorer le temps de réponse des requêtes. [3.6]

3.1.13 Contrainte d'intégrité

Une contrainte d'intégrité permet de garantir la cohérence des données lors des mises à jour de la base. En effet, les données d'une base ne sont pas indépendantes, mais obéissant à des règles sémantiques, après chaque mise à jour, le SGBD contrôle qu'aucune contrainte d'intégrité n'est violée. [3.8]

3.1.14 Outils d'administration et de gestion disponible

C'est l'ensemble d'outils qui permet de donner l'accès aux noms d'utilisateur et aux mots de passe pour accorder des priviléges d'accès au système de gestion de base de données. [3.3]

3.1.15 Interface graphique et l'ergonomie

Une interface graphique est un dispositif de dialogue homme-machine, dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage, le plus souvent une souris. [3.3]

3.1.16 Métadonnée

La métadonnée représente de l'information structurée décrivant, expliquant et localisant la ressource et en facilitant la recherche, l'usage et la gestion. [3.3]

3.2 Comparaison

Dans la suite, un tableau de comparaison des quatre(4) outils (SQL Server, Oracle, Pentaho, Talend Open Studio) sur les différents paramètres expliquer précédemment :

points de comparaison	Oracle	SQL Serveur	Talend Open Studio	Pentaho
Adresse URL	www.oracle.com	www.microsoft.com	www.talend.com	www.pentaho.com
Année de sortie	1979 [3.10]	1989 [3.10]	2006 [3.9]	2000 [3.9]
Licence	Propriétaire [3.10]	Propriétaire [3.10]	Licence publique générale limitée GNU Version 2.1 (LGPLv2.1) [3.9]	GNU Lesser General Public License Version 2.1 (LGPLv2.1)[3.9]
Editeur	Oracle Corporation[3.16]	Microsoft [3.3]	Talend - France [3.9]	Pentaho - Etats-Unis [3.9]
taille maximale de base de données	Illimité [3.22]	5242585 TB ($T = 10^{12}$) [3.23]	512 TB [3.24]	Illimité [3.25]

taille maximale de table	4GB*taille de bloc [3.22]	5242585 TB [3.23]	512 TB [3.24]	16 EB (E= 10^{18}) [3.25]
taille maximale d'une ligne	8 KB [3.22]	Illimité [3.23]	32677 B [3.24]	65530 B [3.25]
nombre maximum de colonnes par table	1000 [3.22]	30000 [3.23]	1012 [3.24]	65135 [3.25]
la taille maximale de la valeur binaire	Illimité [3.22]	2GB [3.23]	2GB [3.24]	4 GB [3.25]
La taille maximale de la valeur de caractère	4000B [3.22]	2 GB [3.23]	32 GB [3.24]	4 GB [3.25]

taille maximale de valeur numérique	126 B [3.22]	126 B [3.23]	64 B [3.24]	64 B [3.25]
taille maximale de nom de la colonne	30 [3.22]	128 [3.23]	128 [3.24]	128 [3.25]
Les étapes de création d'un entrepôt de données	Toutes les étapes assurer	Toutes les étapes assurer	Toutes les étapes assurer	Toutes les étapes assurer
Types d'index accepté	Cluster tables , Expression , Bitmap , Full-text , Partial[3.10]	Expression, Full-text, Spatial[3.10]	Expression, Full-text, Bitmap [3.3]	Full-text[3.3]
Triggers, functions, procédures	Oui	Oui	Oui	Oui

partitionnement des tables	Range, Hash , Composite , List[3.18]	Range[3.19]	Range, Hash, Composite, List[3.20]	Range, Hash, Composite, List[3.21]
protection de la force brute	OUI	OUI	NON	NON
Transaction	OUI	OUI	OUI	OUI
Curseurs	OUI	OUI	OUI	OUI
Vues	OUI	OUI	OUI	OUI
Fonction (UDF)	OUI	OUI	NON	NON

Contrainte d'intégrités	OUI	OUI	OUI
Outils d'administration et de gestion disponible	Enterprise Manager [3.10]	SQL Advantage, SQL Server management Studio[3.10]	Talend Administration Center[3.15] Pentaho User Console pour le côté client (PUC) et PentahoAdministration-Console (PAC) pour le côté serveur [3.17]
utilisateurs concerné	entreprises, groupes de travail, particuliers, relationnel, spatial, distribué [3.3]	entreprises, groupes de travail, particuliers, relationnel, distribué [3.3]	entreprises, groupes de travail, particuliers, relationnel, distribué [3.3]
Plate-forme	Windows, Unix et Linux.[3.10]	Windows, Unix et OS/2 [3.10]	Windows, Unix et Linux. [3.9]

interface graphique	GUI (Graphical User Interface) et SQL [3.3]	GUI (Graphical User Interface) et SQL [3.3]	Outil graphique basé sur Eclipse [3.9]
	Outils <ul style="list-style-type: none"> • Les outils d'administration • Les outils de développement • Les outils de communication • Les outils de génie logiciel • Les outils d'aide à la décision [3.13] 	<ul style="list-style-type: none"> • Moteur de base de données • Les Services d'analyse • Services Rapports • Services d'intégration • Services des données • Services de la qualité des données • Outils d'administration • Composants de connectivité [3.14] 	<p>Outils plus petit par rapport aux autres outils d'entrepôt, mais très orientés vers l'intégration de données. Pour les actions similaires (par exemple, les tables de base de données de lecture), une seule étape (pas un de chaque fabriquant), et le comportement en fonction de la base de données définie par la connexion.</p> <p>[3.9]</p>

Métadonnées	Sous le SGBD ORACLE, l'instruction COMMENT ON TABLE... permet de poser des métadonnées sur une table et les vues (ALL TAB COMMENTS) ou (USER TAB COMMENTS) permettent de visualiser les métadonnées niveau table.[3.11]	Utilise des vue de schémas d'informations pour l'obtention de métadonnées. Les vues de schémas d'informations fournissent une vue interne indépendante des tables système des métadonnées SQL Server sont conformes à la norme ISO pour (INFORMATION SCHEMA) [3.12]	Métadonnées complet qui comprennent des liens vers des bases de données et les objets (tables, vues). Les métadonnées sont stockées dans le centre de l'espace de travail et ce n'est pas nécessaire de relire de la source ou du système de destination. [3.9]	Les métadonnées sont limitée à des connexions de base de données, les métadonnées peuvent être partagées par différentes transformations et informations (tables decatalogue / champs) ou des fichiers spécifications (structure) sont stockées dans les étapes et ne peuvent pas être réutilisés.
-------------	---	---	---	--

3.3 Discutions des résultats

Apres avoir analysé les paramètres des outils dans le tableau, nous avons constaté que les quatre(4) outils assurent les fonctionnalités suivantes : les fonctions (UDF) supportées, les procédures stockées, la transaction, les curseurs, les vues, les contraintes d'intégrité, et pour les paramètres de fond ils ont des caractéristiques similaires.

La différence entre eux réside dans gratuite ou non du l'outil (SQL Server et Oracle sont payant, Pentaho et Talend Open Studio sont des outils licence libre).

L'interface graphique des deux outils SQL Server et Oracle est plus conviviale par rapport aux deux autres outils (Pentaho et Talend Open Studio).

Nous avons noté aussi que les outils d'administration offerte par Oracle et SQL Server sont plus simples pour l'utilisateur, par contre Pentaho et Talend Open Studio demande plus de connaissances.

Pour le coté sécurité SQL Server et Oracle sont plus avancés que les deux autre (Protection de la force brute).

Notre choix est porté sur SQL Server que nous avons déjà utilisé et qui est disponible et simple à utiliser. Néanmoins, nous pouvons recommander Oracle pour la réalisation des entrepôts de données plus puissant, Pentaho est recommandé pour les entreprises avec des petits budgets.

Conclusion

Au cours de ce chapitre, nous avons effectué une comparaison entre les outils d'implémentation d'entrepôt de donnée les plus utilisées : SQL Server, Oracle, Pentaho et Talend Open Studio, et après avoir analysé les résultats, nous avons constaté que le SQL Server est l'outil le plus adapté pour notre application.

Dans le chapitre suivant, nous décrivons en détail la conception de notre système décisionnel.

CHAPITRE 4

Conception

Introduction

Après le choix de l'outil d'implémentation, notre objectif est concevoir un entrepôt de donnée pour l'hôpital Khelil Amrane. Dans ce chapitre, on va présenter d'abord l'organisme d'accueil, à savoir l'historique, la mission et la structure de CHU de Bejaia, ensuite on va entamer la conception de notre projet.

4.1 Présentation d'organisme d'accueil

Actuellement l'hôpital Khalil Amrane de Béjaia est en phase de transition, ce dernier qui fut un établissement public hospitalier (EPH) deviendra un centre hospitalo-universitaire (CHU). Cette transition va engendrer un changement total administratif et fonctionnel.

4.1.1 Historique de CHU de Bejaïa

Le secteur sanitaire de Bejaïa comprend plusieurs structures de santé, parmi celles-ci il y a l'hôpital Khalil Amrane. Le secteur sanitaire de Béjaia s'étale sur une superficie de 460,65 KM². Il assure une couverture sanitaire aux 240.258 habitants des sept (07) communes suivantes : Béjaia, ouedghir, tichy, Tala hamza, Boukhlifa, Aokas, et Tizi-nberber. Le secteur est géré par la direction de l'hôpital Khelil Amrane, situé au chef-lieu de la commune de Béjaia.

L'établissement public à caractère administratif, le secteur sanitaire de Béjaia est doté d'un budget de fonctionnement et d'une autonomie de gestion.

Jusqu'en 1991, date de l'inauguration et de l'entrée en fonction de l'hôpital Khelil Amrane, le secteur sanitaire de Béjaia n'était doté que de deux hôpitaux : Aokas et Frantz fanon, hérités de la période coloniale.

En 2011, l'hôpital Khelil Amrane est devenu le siège du centre Hospitalo-universitaire (CHU) de Béjaïa. La création de ce dernier fait suite à la création de la faculté de médecine¹. Le secteur sanitaire dispose des structures suivantes :

- 01 hôpital chef-lieu.
- 02 hôpitaux annexes.
- 01 polyclinique avec maternité.
- 01 maternité urbaine.
- 03 polycliniques.
- 01 laboratoire d'hygiène.
- 04 centres de santé avec maternité.
- 01 clinique dentaire.
- 01 maison de diabétique.
- 01 centre de contrôle sanitaire aux frontières.
- 01 unité de dépistage et de suivi (UDS).
- 01 centre de transfusion sanguine (CTS).
- 30 unités de soins.
- 01 école paramédicale.

4.1.2 Mission de CHU de Béjaïa

Conformément à l'article 5 du décret exécutif N°97-466 du 02 décembre 1997 fixant les règles de création, d'organisation et de fonctionnement des hôpitaux, l'hôpital a pour mission de prendre en charge, de manière intégrée et hiérarchisée, les besoins sanitaires de la population. Plus concrètement, il a pour tâche de :

- Assurer l'organisation et la programmation de la distribution des soins.
- Mettre en oeuvre les activités de prévention, de soins, de réadaptation médicale et d'hospitalisation.
- Assurer les activités liées à la santé reproductive et à la planification familiale.
- Appliquer les programmes nationaux et locaux de santé et de population.
- Contribuer au recyclage et au perfectionnement du personnel des services de santé.

4.1.3 Structures de CHU de Béjaïa

Hôpital Khelil Amrane

Le secteur sanitaire de Bejaïa comprend plusieurs structures de santé, parmi lesquelles il y a l'hôpital Khelil Amrane.

Le secteur sanitaire de Bejaïa couvre sur une superficie de 460,65 Km. Il assure une couverture sanitaire aux 240.258 habitants des sept (07) communes suivantes : Bejaia, Oued-Ghir, Tichy, Tala

¹Décret exécutif n°09-319 du 6 octobre 2009 complétant la liste des centres hospitalo-universitaire

hamza, Boukhlifa, Aokas et Tizi-N'berber.

Le secteur sanitaire est géré par la direction de l'hôpital Khelil Amrane, situé au chef-lieu de la commune de Bejaïa. Il est doté d'un budget de fonctionnement et d'une autonomie de gestion.

Jusqu'en 1991, date de l'inauguration et de l'entrée en fonction de l'EPH Khelil Amrane, le secteur sanitaire de Bejaïa n'était doté que de deux hôpitaux : Aokas et Frantz Fanon, hérités de la période coloniale.

En 2011, au niveau de l'hôpital Khelil Amrane " la direction générale " est devenue le siège du centre Hospitalo-universitaire (CHU) de Bejaïa. La création de ce dernier est venue à la suite de l'inauguration de la faculté de médecine et de sciences de soins infirmières.

Hôpital Frantz Fanon

Suivant un acte public en date de 07/08/1878 Mr et Mme Troncy ont fait donation à la commune de Bougie d'un immeuble situé à la rue de trêve place dit du train, pour servir à la construction d'un hôpital civil dans la localité. La commission de surveillance de l'ambulance de bougie a été autorisé à accepter cette donation par arrêt de Mr le préfet de Constantine en date de 28/09/1879 par décret du 08/03/1883 de Mr le Président de la République a fait concession gratuite à titre de donation à l'hôpital civil de bougie (Frantz Fanon actuellement) des immeubles domaniaux situés à Bougie.

Le dit immeuble (offert par Mr et Mme Troncy) mis en vente le 09/11/1889 a été adjugé à Mr le général Surney pour la somme de 40200 Franc qui ont été employé concurremment avec les fonds de subvention du gouvernement général à la construction de l'hôpital actuel.

Par décret présidentiel du 03/06/1893, il est fait concession gratuite à l'hôpital civil de Bougie d'un terrain domanial de la contenance de 6354.23 ancien Franc.

D'après les archives du répertoire des malades hospitalisés on n'a déduit que l'hôpital a commencé ses activités en janvier 1896.

A sa construction, il a été nommé " Hôpital Civil de Bougie ".

Vers les années 50, il a été nommé " Hôpital Régional de Bougie " d'après les archives, à cette époque, il a été composé des Services suivants :

- Service Troncy pasteur de médecine " Femme et Homme ".
- Service Gérard de chirurgie " Homme ".
- Service Perrusset de chirurgie " Homme ".
- Service Rambert de chirurgie " Femme " et maternité.
- Service Gerard bis ophtalmologie et ORL.

- Service curie phtisiologie et isolement " Femme".
- Service bloc phtisiologie et isolement " Homme ".
- Service le Grain militaire.
- Service aliènes de psychiatrie.
- Service Généraux.
- Service radiologie.
- Camion d'ophtalmologie.
- Laboratoire d'analyse.
- Divers services administratifs.

Quelques années après l'indépendance, il a pris le nom du " Secteur Sanitaire de Bejaia ". Après l'inauguration de l'Hôpital Khelil Amrane en 1991 il devient " Hôpital Frantz Fanon " du nom du célèbre psychiatre Français, connue pour son engagement en faveur de l'indépendance de l'Algérie durant la guerre de libération (1954-1962).

Hôpital Targua Ouzemour

La structure de l'hôpital Mère et enfant de Bejaïa est à l'origine d'une structure de la CNAS, versée à la santé et qui a été réaménagée en clinique mère et enfant qui faisait partie du secteur sanitaire de Bejaïa en 1991.

Le décret exécutif n° 07/140 du 2 Jourad-el-aoual 1428 correspondant au 19 mai 2007, modifié et complété, portant création, organisation et fonctionnements des établissement publics hospitaliers et des établissements publics de santé de proximité à permis de mettre en place en 2008 l'EHS est née.

Centre de wilaya de transfusion sanguine (CWTS)

Inauguré le 4 juillet 2000 avec la participation de l 'Agence Nationale du Sang, la Direction de la Santé et du Secteur Sanitaire de Bejaia.

Situé au centre-ville avec accès facile et stratégique aux donneurs de sang pour la collecte de sang total.

Il couvre les besoins de santé de la Wilaya de Bejaia et les villes environnantes, dont la population est estimée au 31/12/2011 à 935 200 habitants.

Le CHU est chargé, en relation avec l'Université de Bejaïa de la formation supérieure en sciences médicales, en sciences de soins infirmières et concerné des missions :

D'exploration, de prévention, de diagnostic de soins, de formation, d'études et de recherche.

En 2011, au niveau de l'hôpital Khelil Amrane " la direction générale " est devenue le siège du centre Hospitalo-universitaire (CHU) de Bejaïa. La création de ce dernier est venue à la suite de l'inauguration de la faculté de médecine et de sciences de soins infirmières.

Le CHU est placé sous la double tutelle :

- Administrative du ministre chargé de la santé.
- Pédagogique est assurée par le ministre chargé de l'enseignement supérieur.

Assurant la distribution des produits sanguins labiles aux différentes structures de soins publiques et privés de la wilaya :

CHU est composé :

- Hôpital Khelil Amrane.
- Hôpital Frantz Fanon.
- Hôpital targua ouzemmour (clinique mère- enfant).
- BS DE L'EPH d'Aokas.
- Postes de Transfusion Sanguine : Akbou, Sidi-Aich ; Kherrata.
- Banque de Sang d 'Amizour.
- Banque de sang d'Aokas.
- 5 Cliniques privées :
 - Rameau de l'olivier. (Chirurgie et hémodialyse).
 - Les L-ilas.
 - Dr Mohdeb. (Hémodialyse).
 - médico-chirurgicale Dr Belaid.
 - médico-chirurgicale e du Dr Benmerad.

4.1.4 Organigramme administratif du CHU de Béjaïa

L'organigramme administratif du CHU de Béjaïa est résumé sur la figure suivante : (Figure 4.1).

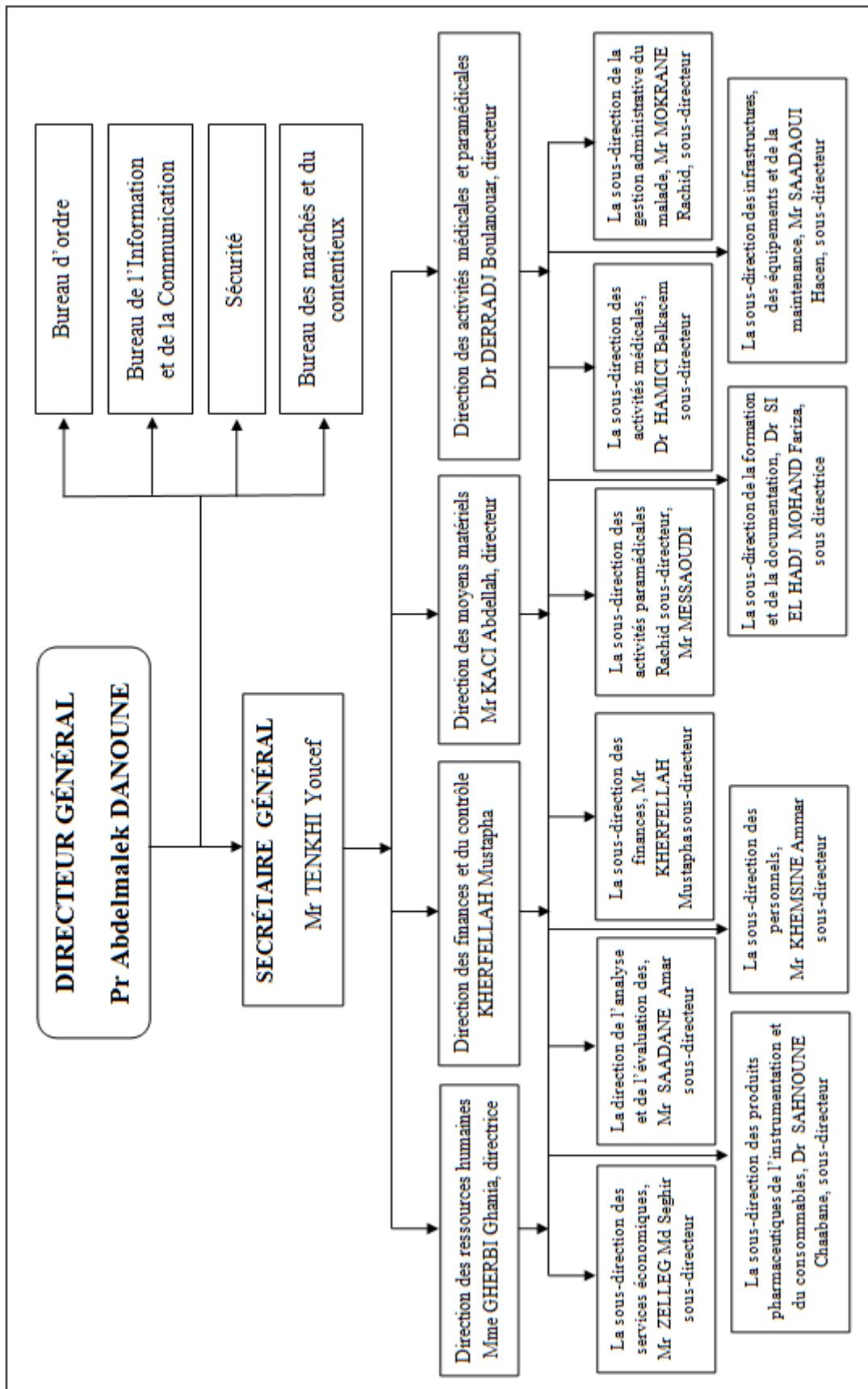


FIG. 4.1 – Organigramme administratif du CHU de Béjaïa

4.2 Les étapes de conception de l'entrepôt de données

Pour la conception de notre entrepôt de données, nous avons suivi les quartes étapes majeurs montrées dans la (Figure 4.2), c'est à dire : la spécification des besoins, puis la conception du modèle multidimensionnelle et la création du cube OLAP, enfin, la représentation graphique.

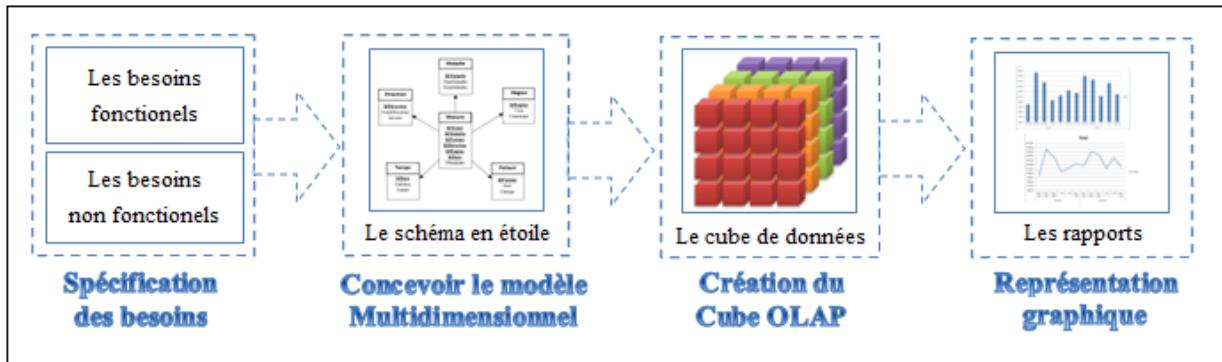


FIG. 4.2 – Les étapes d'implémentation de notre entrepôt de données

4.2.1 Spécification des besoins

Le but essentiel de l'entrepôt de données est de donner une vue générale et concise sur un domaine quelconque, dans notre cas il nous permettra de donner une meilleure vision sur le nombre de malades par rapport à plusieurs paramètres : temps, régions, type de maladie, catégorie d'âge etc.

Après avoir effectué un stage au niveau de L'hôpital Khellil Amrane de Bejaia, nous avons sélectionné les besoins suivants :

Les besoins fonctionnels

Consiste à avoir des informations décisionnelles sur le nombre de malades par rapport aux paramètres suivants :

1. Le nom ou le type de chaque maladie dans des différentes périodes de temps.
2. Le nom ou le type de chaque maladie dans des différentes régions.
3. Le nom ou le type de chaque maladie par rapport aux tranches d'âge des malades.
4. Le nom ou le type de chaque maladie dans les services et directions concernées.
5. Le nom ou le type de chaque maladie dans des différentes périodes et régions.
6. Le nom ou le type de chaque maladie dans des différentes régions et différentes tranches d'âge.
7. les catégories d'âge des malades dans des différentes périodes de temps.

8. les différentes régions dans des différents périodes de temps.
9. les catégories d'âge des malades dans des différents périodes de temps.
10. Le nom ou bien le type de chaque maladie dans des différentes régions et pour des catégories d'âge et des périodes de temps différentes.

Les besoins non fonctionnels

A part les besoins fondamentaux, notre implémentation doit répondre aux critères suivants :

1. La rapidité du traitement : En effet, vu le nombre important des données que nous pouvons analyser, il est impératif que la durée d'exécution des traitements s'approche le plus possible du temps réel.
2. La performance : notre application doit être avant tout performante à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.
3. La sécurité : l'application doit être protégée avec une clé ou un mot de passe avec un système de protection contre la force brute.

4.2.2 Concevoir le modèle multidimensionnelle

Nous devons concevoir un modèle multidimensionnelle pour répondre aux exigences des utilisateurs qui doivent répondre aux besoins d'affaires et contient des informations qui peuvent être facilement accessible. La Conception du modèle doit être facilement extensible en fonction des besoins futurs. Ce modèle prend en charge la conception d'un cube OLAP pour fournir les résultats d'une requête "instantanés" pour les analystes.

Les Dimensions

Les tables de dimension contiennent des descriptions textuelles sur les sujets de l'établissement médical ou l'hôpital Khellil Amrane. Pour pouvoir repenser aux besoins de l'établissement nous avons proposé les cinq (05) tables de dimensions suivantes : Maladie, Temps, Région, patient, direction.

- **La table de la dimension "Maladies"** : elle contient la clé primaire " IdMaladie " et les attributs " TypeMaladie " et " NomMaladie"

Les attributs avec les données utilisées :

- TypeMaladie : Chronique, Cancer, Yeux.
- NomMaladie : Epilepsie, Alergie, Paumon, Sang, Infection, Glandome.

La figure 4.3 suivante montre la table de dimension Maladie

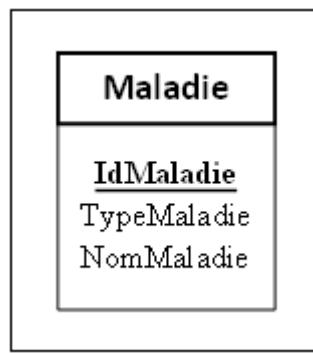


FIG. 4.3 – Table de dimension " Maladie "

- **La dimension " Temps " :** Elle contient la clé primaire " IdDate " et les attributs :" CatMois " et " Année ".

Les attributs avec les données utilisées :

- CatMois : entre Janvier et Mars, entre Avril et Juin, entre Juillet et Septembre, entre Octobre et Décembre.
- Année :2012, 2013.

La figure 4.4 suivante montre la table de dimension Temps

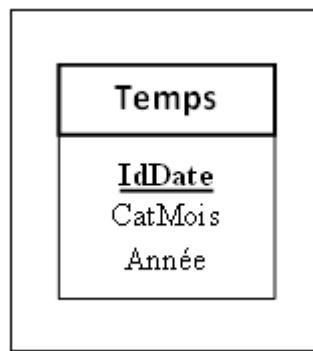


FIG. 4.4 – Table de dimension " Temps "

- **La dimension " Région " :** Elle contient la clé primaire " IdRegion " et les attributs " Ville " et " Commune ".

Les attributs avec les données utilisées :

- Ville : Béjaïa, Autres.
- Commune : Béjaïa, Akbou, Ferraoun, Kendira, Sedouk, Autres.

La figure 4.5 suivante montre la table de dimension Région

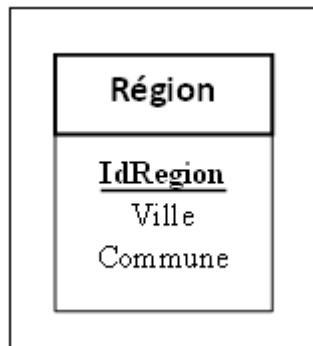


FIG. 4.5 – Table de dimension " Région "

- **La dimension " Patient " :** Elle contient la clé primaire " IdPatient " et les attributs " Sexe " et " CatAge ".

Les attributs avec les données utilisées :

- Sexe : Homme, Femme.
- CatAge : moins de 10ans, entre 10ans et 20ans, entre 21ans et 40ans, entre 41ans et 60ans, entre 61ans et 70ans, plus de 70 ans.

La figure 4.6 suivante montre la table de dimension Patient

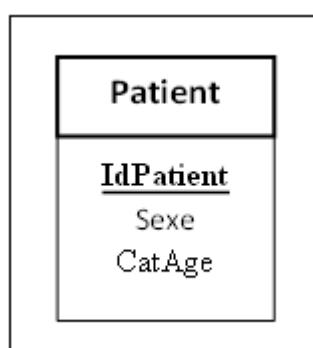


FIG. 4.6 – Table de dimension " Patient "

- **La dimension " Direction "** : Elle contient la clé primaire " IdDirection " et les attributs " NomDirection " et " Service ".

Les attributs avec les données utilisées :

- NomDirection : Khelil Amrane, Frantz Fanon, Targua Ouzemmour, BS de L'EPH Aoukas.
- Service : service de neurologie et maladies neuromusculaires, service des maladies Infectieuses et Tropicales.

La figure 4.7 suivante montre la table de dimension Direction

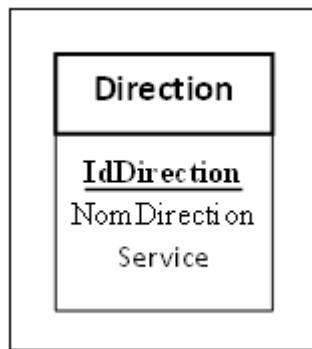


FIG. 4.7 – Table de dimension " Direction "

Les mesure

Diminuer le nombre des malades et le but essentiel dans chaque secteur qui travaille pour la santé, c'est pour cela qu'on a choisi le "nombre de malades" comme mesure pour notre table de fait pour avoir une vue générale ou bien détaillée, sur le nombre des malades par rapport à la région, temps, catégorie d'âge etc. Qui permettra ensuite au personnel médical de faire une analyse sur les résultats trouvés et prendre des bonnes décisions.

Table de faits

Contient principalement trois (03) paramètres essentiels : la clé primaire " IdMesur ", les clés étrangères qui sont les clés primaires des différentes dimensions : " IdMaladie, IdDate, IdRegion, IdPatient, IdDirection " et à la fin la mesure : " le nombre de malades ", la figure 4.8 suivante montre la table de fait réalisée

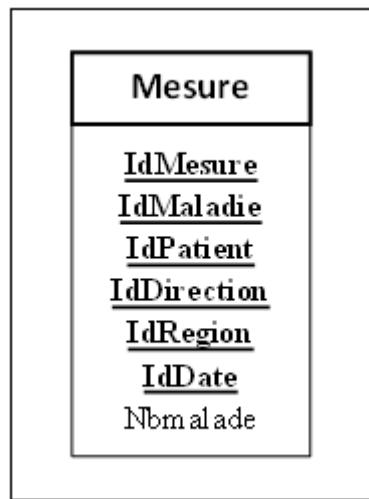


FIG. 4.8 – Table de fait

La Modélisation

Les trois schémas essentiels utilisés pour modéliser un entrepôt de données sont :

1. Le schéma en étoile
2. Le schéma en flocon de neige
3. Le schéma en constellation

Dans notre projet nous avons utilisé le schéma étoile en raison de modèle d'attribut hiérarchique, il permet une analyse et des performances rapides en interrogeant les données.

- Schéma en étoile

Le schéma ressemble à une étoile, avec des points de rayonnement d'un centre. Le centre de l'étoile se compose de la table de fait " mesure " et les points de l'étoile sont les tables de dimensions " Maladie, Temps, Région, patient, direction " comme le montre le schéma 4.9 suivant

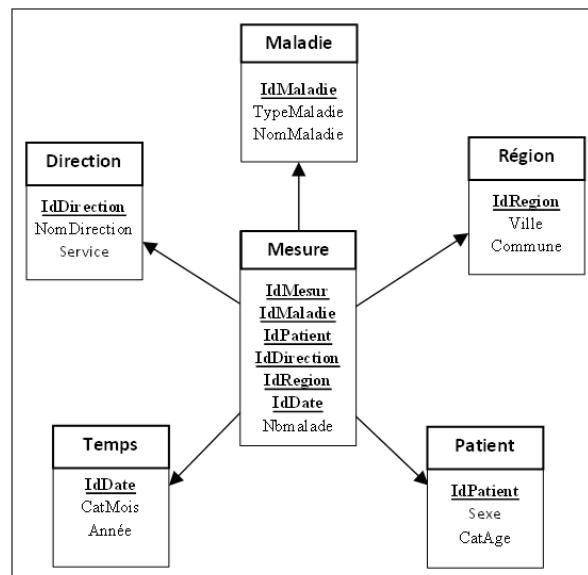


FIG. 4.9 – Schéma en étoile

- Le schéma en flocon de neige

Le schéma est le résultat de la décomposition d'une ou plusieurs dimensions en plusieurs niveaux formant une hiérarchie qui offre une meilleure visualisation et compréhension des données comme le montre le schéma 4.10 suivant

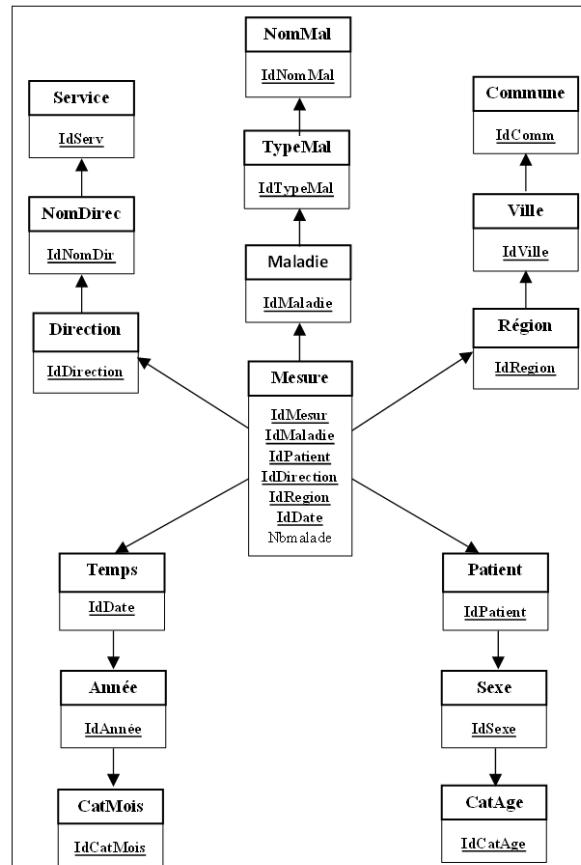


FIG. 4.10 – Schéma en flocon de neige

- Le schéma en constellation

Sont des schémas où plusieurs modèles dimensionnels se partagent les mêmes dimensions, c'est-à-dire les tables de faits ont des tables de dimensions en commun. Les tables de dimensions partagées par plusieurs tables de fait doivent être exactement les mêmes. La figure 4.11 suivante présente un exemple d'un schéma en constellation.

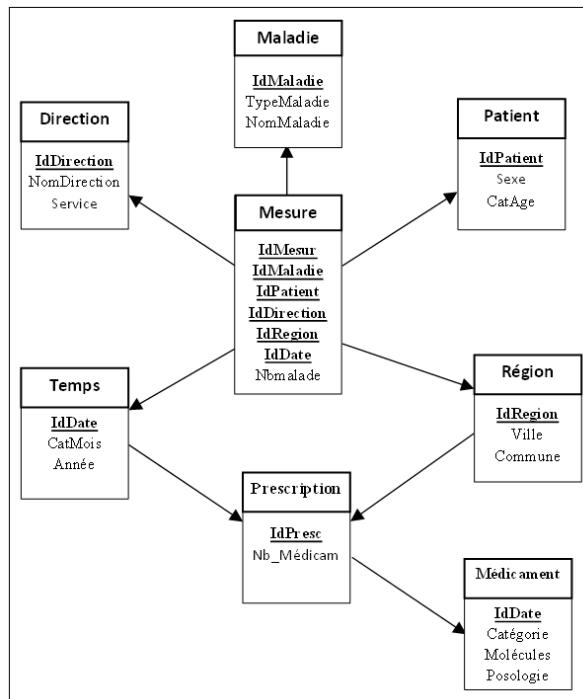


FIG. 4.11 – Schéma en constellation

4.2.3 Le cube OLAP

Afin de faciliter l'analyse, il est indispensable de recourir à la modélisation multidimensionnelle des données. Grâce au concept OLAP, nous pouvons transformer les données stockées dans des bases de données relationnelles en informations médicales pertinentes et faciles à exploiter. Nous utiliserons pour la conception de notre base de données multidimensionnelle la modélisation ROLAP, en exploitant les codes T-SQL " Transact Structured Query Language ".

Un cube de donnée est identifié par 03 dimensions ou bien attributs d'une dimension. Puisque nous avons 05 dimensions et chaque dimension contient 2 attributs alors on peut avoir plusieurs cubes en fonction des dimensions, les différents cubes que nous pouvons avoir en combinaison des 05 dimensions sont les suivants :

Cube 1 : Maladie, Temps et Région.

Cube 2 : Maladie, Temps et Patient.

Cube 3 : Maladie, Temps et Direction.

Cube 4 : Maladie, Région et Patient.

Cube 5 : Maladie, Région et Direction.

Cube 6 : Maladie, Patient et Direction.

Cube 7 : Temps, Région et Patient.

Cube 8 : Temps, Région et Direction.

Cube 9 : Temps, Patient et Direction.

Cube 10 : Région, Patient et Direction.

A partir de chacun des cubes au-dessus nous pouvons extraire plusieurs cubes en combinaison des attributs de chaque dimension, si on prend par exemple le cube1, les combinaisons qu'on peut avoir sont les suivantes :

Cube 1.1 : Maladie, TypeMaladie, Temps.CatMois et Région.Ville.

Cube 1.2 : Maladie, TypeMaladie, Temps.CatMois et Région.Commune.

Cube 1.3 : Maladie, TypeMaladie, Temps.Année et Région.Ville.

Cube 1.4 : Maladie, TypeMaladie, Temps.Année et Région.Commune.

Cube 1.5 : Maladie, NomMaladie, Temps.CatMois et Région.Ville.

Cube 1.6 : Maladie, NomMaladie, Temps.CatMois et Région.Commune.

Cube 1.7 : Maladie, NomMaladie, Temps.Année et Région.Ville.

Cube 1.8 : Maladie, NomMaladie, Temps.Année et Région.Commune.

En tout on peut avoir 80 cubes différents en combinaison des dimensions avec leur attributs.

Représentation d'un cube de données

Pour la représentation du cube de données nous avons choisi le Cube1.6 comme exemple, la représentation du cube correspondant est la suivante (Figure 4.12)

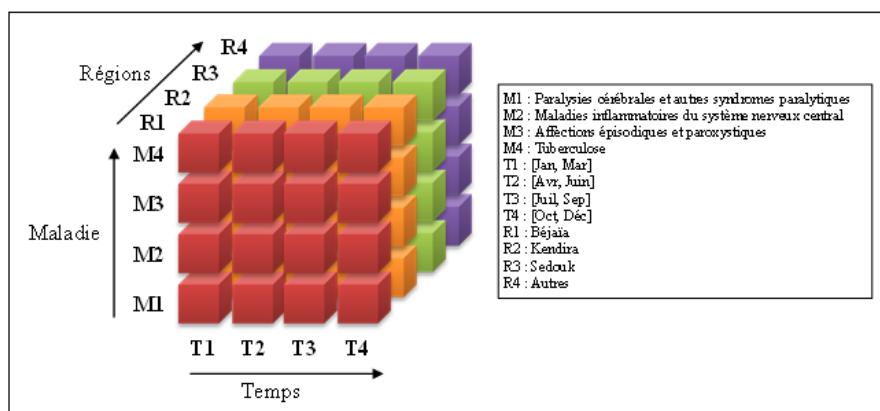


FIG. 4.12 – Le cube original

Les opérations OLAP

Les opérations principales OLAP qu'on peut utiliser pour rependre aux différents besoins sont : pivot, Slice, Dice et Roll up, les figures suivantes (figure 4.13, 4.14, 4.15 et 4.16) montrent un exemple de chaque opération effectuée au schéma présenté précédemment (figure 4.13)

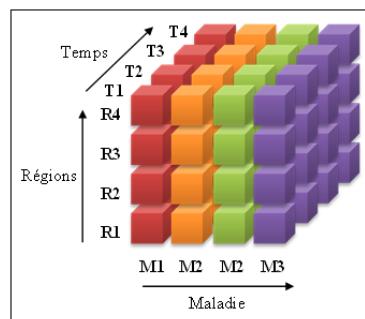


FIG. 4.13 – L'opération Pivot

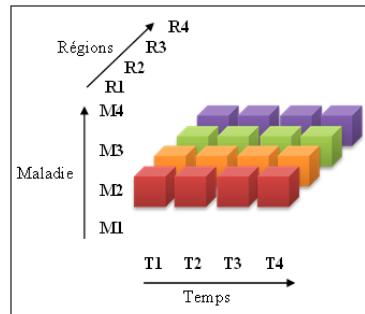


FIG. 4.14 – L'opération Slice

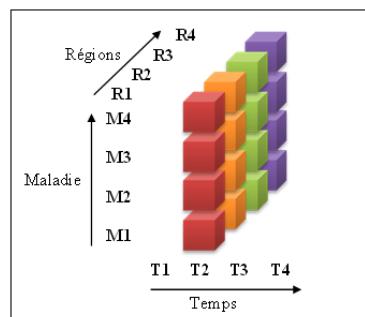


FIG. 4.15 – L'opération Dice

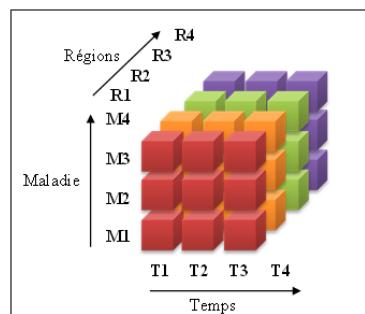


FIG. 4.16 – L'opération Roll up

4.2.4 Représentation graphique

Nous avons choisi d'utiliser pour l'outil graphique Excel, car il est facile à utiliser, au même temps très riche par des fonctionnalités graphiques, en plus, on trouve pas des difficultés pour faire une connexion entre notre cube de données et l'outil Excel, vu que les deux outils sont développés par Microsoft.

Dans le chapitre suivant, nous allons montrer quelques représentations graphiques.

Conclusion

Au cours de ce chapitre, nous avons commencé d'abord par la présentation d'organisme d'accueil, en spécifiant l'historique et les différentes structures de l'hôpital Khelil Amrane ensuite nous avons entamé la conception de notre projet, à savoir, l'identification et la collection des exigences ou bien des besoins, concevoir le modèle dimensionnel et enfin la création du cube OLAP.

Dans le prochain chapitre, nous allons présenter le détail de l'implémentation de chaque partie de notre solution décisionnelle.

CHAPITRE 5

Implémentation

Introduction

Nous consacrons ce chapitre à la partie réalisation, ce processus débute par l'exécution des requêtes T-SQL afin de créer et peupler les tables de dimensions et la table de fait, suivi par la création de cube OLAP. Nous avons réalisé des interfaces graphiques qui permettent de visionner les résultats sous forme de graphe.

5.1 Les étapes de création de l'entrepôt de données

Pour la création de notre entrepôt de données, nous avons suivi les trois étapes majeures montrées dans la (figure 5.1), commençant par la création de modèle dimensionnel, en suite la création de cube OLAP et a la fin la représentation graphique.

- pour la création de modèle dimensionnel, nous avons commencé par la création de la base de données de l'entrepôt, en suite la création et la peuplissions de la table de fait et les tables de dimensions.
- pour la création de cube OLAP, nous avons établi la source des données et des Vues, crée le cube, puis l'attribut hiérarchique dans les dimensions, déployer le cube pour un eventuel traitement.
- pour la représentation graphique, nous avons effectué une connexion entre notre cube de données et l'outil graphique Excel.

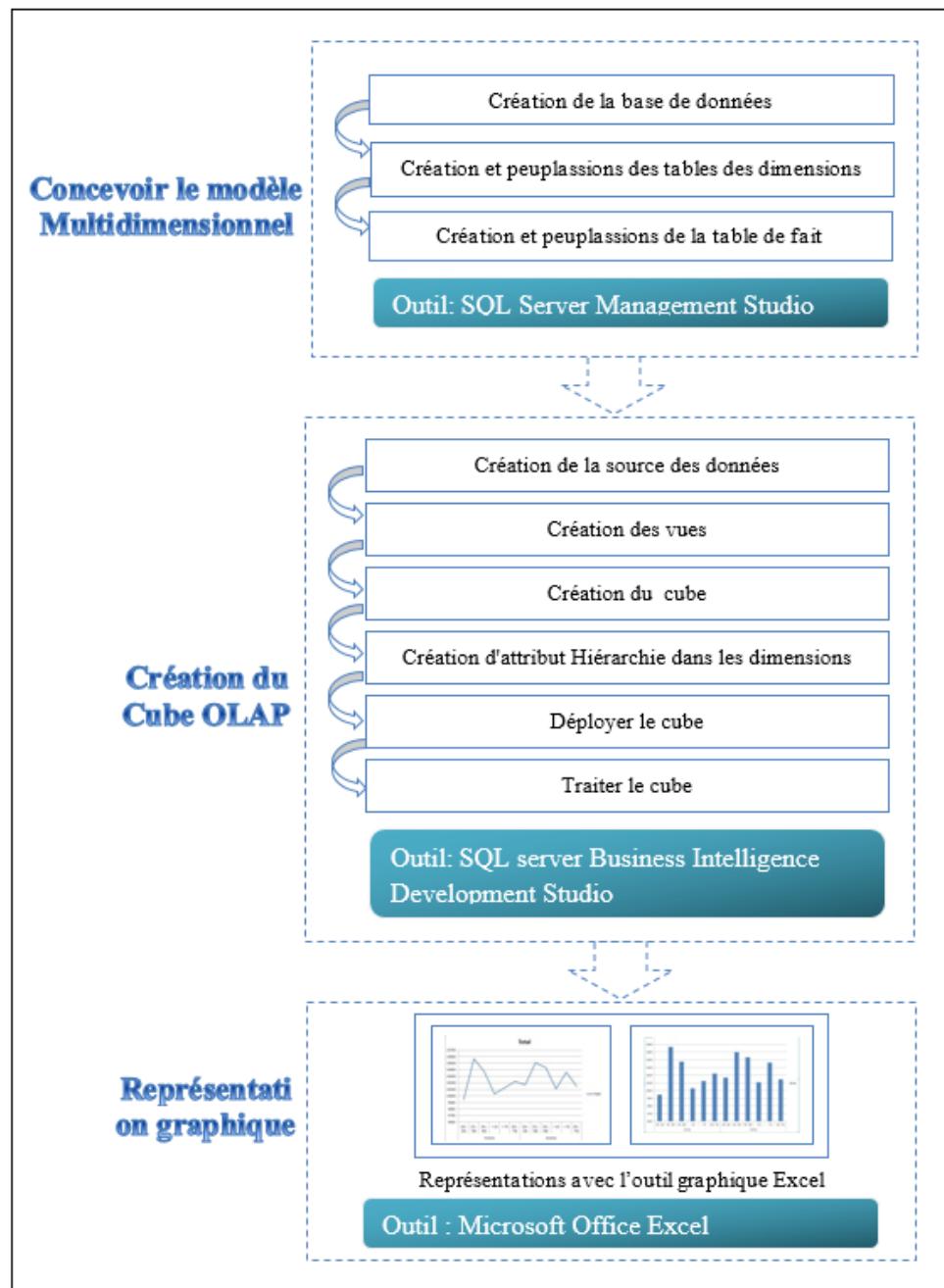


FIG. 5.1 – Les étapes de la réalisation de notre entrepôt de données

5.2 Exécution des requêtes T-SQL

Nous avons utilisé les requêtes T-SQL Script dans SSMS (SQL Server Management Studio) pour créer les tables et les peupler avec des données de l'hôpital, pour cela nous avons suivi les étapes suivantes :

5.2.1 Crédation de la base de données de l'entrepôt

Pour la création de la base de données pour notre entrepôt de données, nous avons utilisé l'instruction suivante :

```
Create database DataWarehouse  
Go  
Use DataWarehouse  
Go
```

5.2.2 Créer et peupler les tables des dimensions

Création des tables des dimensions

Pour la création de la dimension "maladie" pour notre entrepôt de données, nous avons utilisé les instructions suivantes :

```
Use DataWarehouse  
Create table Maladie  
(IdMaladie int primary key identity, TypeMaladie varchar(100), NomMaladie  
varchar(100))  
Go
```

Pour la création des autres dimensions, nous avons suivi les mêmes instructions.

peupler les tables de dimensions

Pour peupler la dimension "Maladie" avec l'échantillon de valeurs, nous avons utilisé les instructions suivantes :

```
Use DataWarehouse  
Insert into Maladie(TypeMaladie, NomMaladie)values  
( 'Chronique', 'Epilepsie'), ( 'Chronique ', 'Alergie ' ), ( 'Cancer', 'Paumon ' ),  
( 'Cancer', 'Sang' ), ( 'Yeux', 'Infection ' ), ( 'Yeux', 'Glancome' )  
Go
```

Pour remplir les autres dimensions, nous avons suivi les mêmes instructions.

5.2.3 Crédier et peupler de la table de fait

Nous avons créé une table de fait pour contenir l'entrée transactionnelle (nombre de malades) avec les colonnes clés étrangères appropriées qui se réfèrent à la colonne de clé primaire de dimensions.

```
Use DataWarehouse  
Create Table Mesure  
(IdMesure bigint primary key identity, IdMaladie int , IdDateint, IdRegionint,  
IdPatient int, IdDirection int, Nbmaladeint)  
Go
```

En suite, nous avons ajouté des relations entre la table de fait et les tables de dimensions.

```
Use DataWarehouse  
ALTER TABLE Mesure ADD CONSTRAINT IdMaladie FOREIGN KEY  
(IdMaladie) REFERENCES Maladie(IdMaladie);  
ALTER TABLE Mesure ADD CONSTRAINT IdDate FOREIGN KEY (Id-  
Date) REFERENCES Temps(IdDate);  
ALTER TABLE Mesure ADD CONSTRAINT IdRegion FOREIGN KEY  
(IdRegion) REFERENCES Region(IdRegion);  
ALTER TABLE Mesure ADD CONSTRAINT IdPatient FOREIGN KEY (Id-  
Patient) REFERENCES Patient(IdPatient);  
ALTER TABLE Mesure ADD CONSTRAINT IdDirection FOREIGN KEY  
(IdDirection) REFERENCES Direction(IdDirection);  
Go
```

A la fin, nous avons peuplé la table de fait avec l'échantillon de valeurs.

```
Use DataWarehouse  
Insert into Mesure (IdMaladie, IdDate, IdRegion, IdPatient,IdDirection, Nb-  
malade)values  
(1,2,2,4,1,8),(1,2,2,4,3,6),(1,2,2,4,5,0),(1,2,2,4,7,7),(1,2,2,5,1,2),(1,2,2,5,3,9),  
(1,2,2,5,5,4),(1,2,2,5,7,1),(1,2,2,6,1,6),(1,2,2,6,3,3)  
Go
```

5.3 Crédation de cube OLAP

Pour la création de cube OLAP dans l'environnement Microsoft BIDS(Business Intelligence Development Studio), nous avons suivi les étapes indiquées ci-dessous après avoir Démarré l'environnement BIDS.

5.3.1 Démarrer le Projet Analyses Services

Pour le démarrage de projet analyses services nous avons cliqué sur "fichier" ensuite "nouveau" et "projet" après nous avons sélectionné "projet analyses services" et à la fin nous avons donné un nom pour le projet (Figure 5.2) .

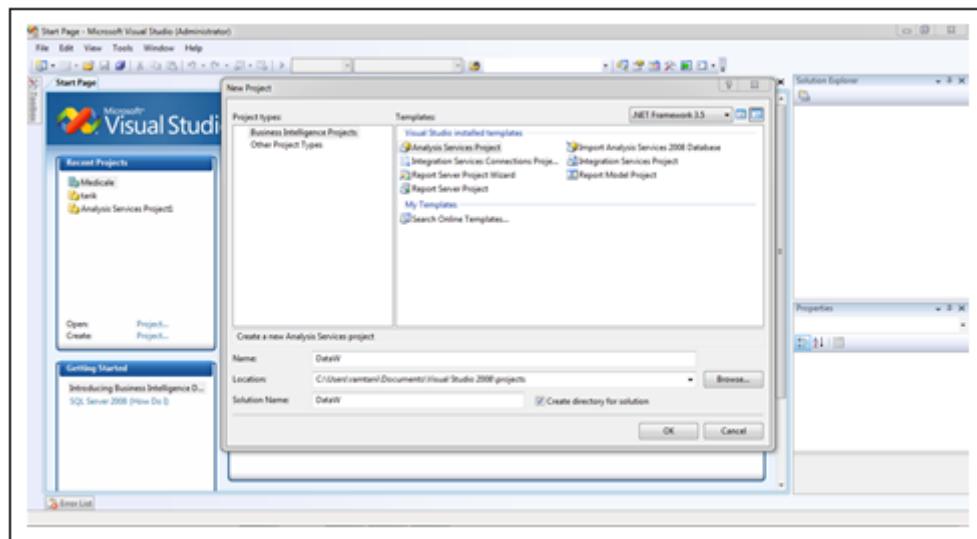


FIG. 5.2 – Démarrer Projet Analysis Services

5.3.2 création de la source de données

Pour la création de la source de données, nous avons suivi les étapes suivantes :

- Nous avons cliqué sur le bouton droit sur "source de données" dans l'explorateur de solutions situé au côté droit de l'interface, la (Figure 5.3) montre l'interface.
- Sélectionner New (Figure 5.4).
- Créer une nouvelle connexion (Figure 5.5).
- Sélectionner la connexion créé dans les connexions de données.
- Sélectionner Hériter d'options (Figure 5.6).
- Attribuer le Nom de la source.

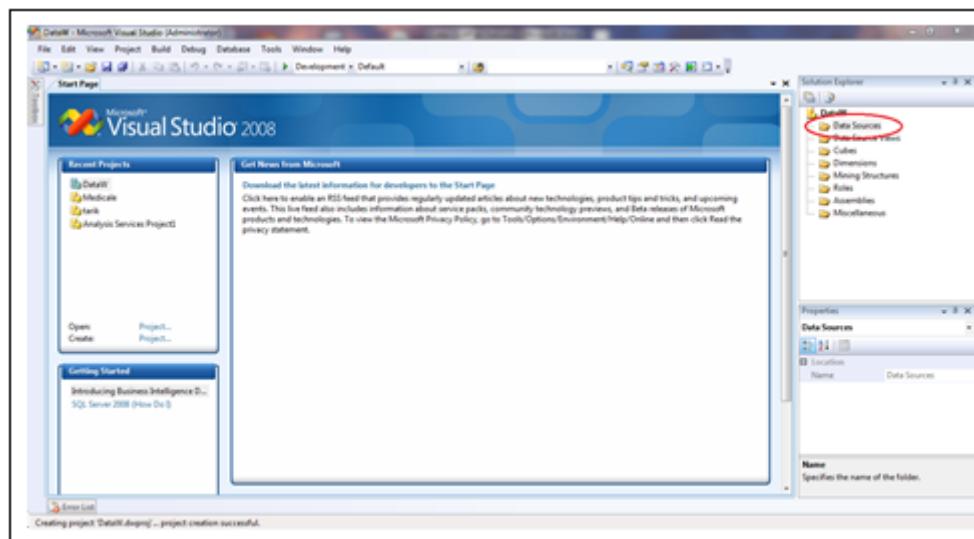


FIG. 5.3 – Nouvelle source de données

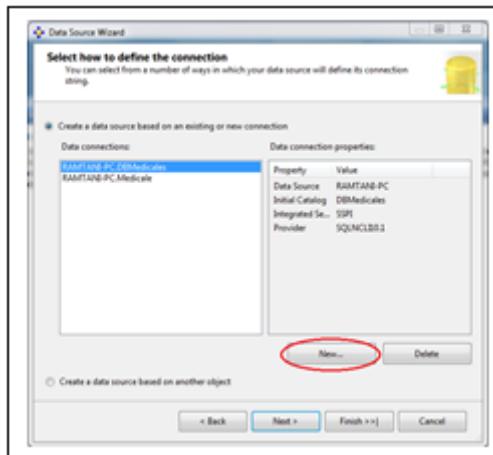


FIG. 5.4 – New

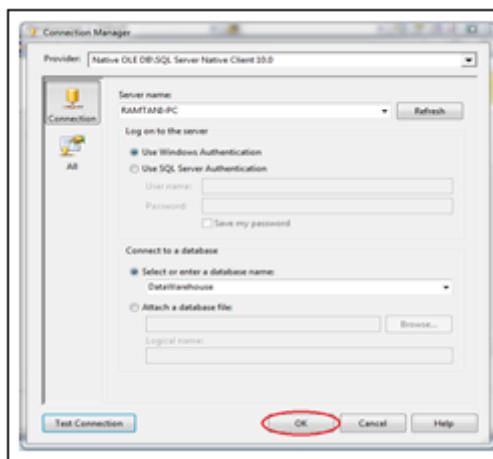


FIG. 5.5 – Crédit d'une nouvelle connexion

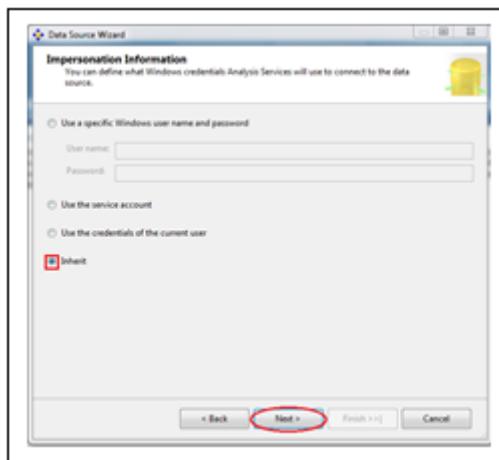


FIG. 5.6 – Hériter d'options

5.3.3 Crédation des Vues de la source de données

Pour la création des Vues de la source de données, nous avons suivi les étapes suivantes :

- Nous avons cliqué sur le bouton droit sur "source de données Vues" dans l'Explorateur de solutions (Figure 5.7).
- Sélectionner la Source de données relationnelles, que nous avons créé précédemment (Data-Warehouse).
- Pour déplacer la table de fait sur le côté droit pour le mettre dans la liste d'objets, nous avons sélectionné la table "Mesure", ensuite nous avons cliqué sur le bouton "fléché" pour déplacer l'objet sélectionné à la droite (Figure 5.8).
- Pour ajouter des dimensions qui sont liées à notre table de fait, nous avons sélectionné la table de fait à la droite et nous avons ajouté des tables associées par un clique sur "ajouté"(Figure 5.9).
- A la fin nous avons affecté un Nom (DataWarehouse).

Maintenant la source de données est prête à utiliser.

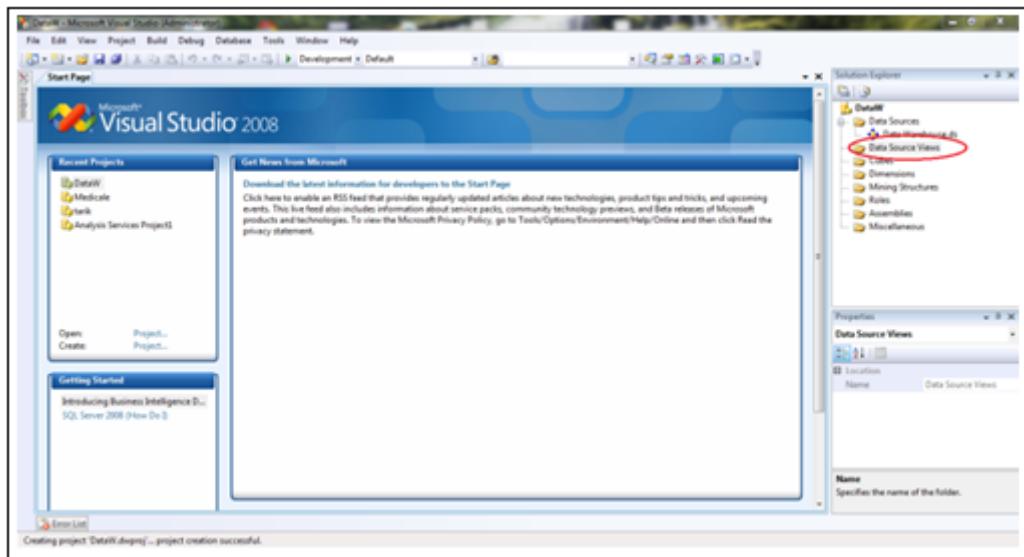


FIG. 5.7 – Nouvelle source de données View

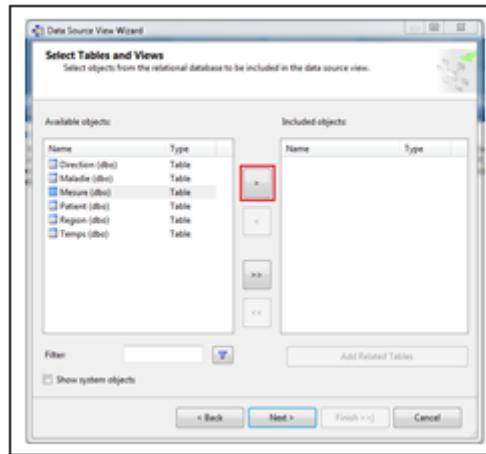


FIG. 5.8 – déplacement la table de faits

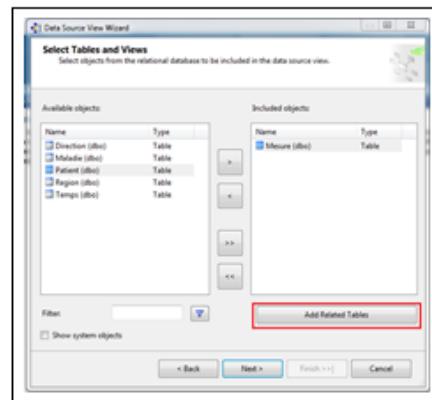


FIG. 5.9 – Ajout des tables associées

5.3.4 Crédation de Cube

Pour la création de Cube, nous avons suivi les étapes suivantes :

- Nous avons cliqué sur le bouton droit sur "Cube" Dans l'Explorateur de solutions (Figure 5.10).
- Sélectionner l'option "utiliser les tableaux existants".
- Sélectionner le nom de la table de fait "Mesure" dans les tableaux de groupe de mesures.
- Choisir les mesures qu'on veut placer dans notre Cube.
- Sélectionner toutes les dimensions, qui sont associés à notre table de fait.
- Affecter un nom pour le cube (DataWarehouse).

Maintenant, notre cube est prêt, nous pouvons voir le cube et les dimensions ajoutés dans notre explorateur de solution nouvellement créé.

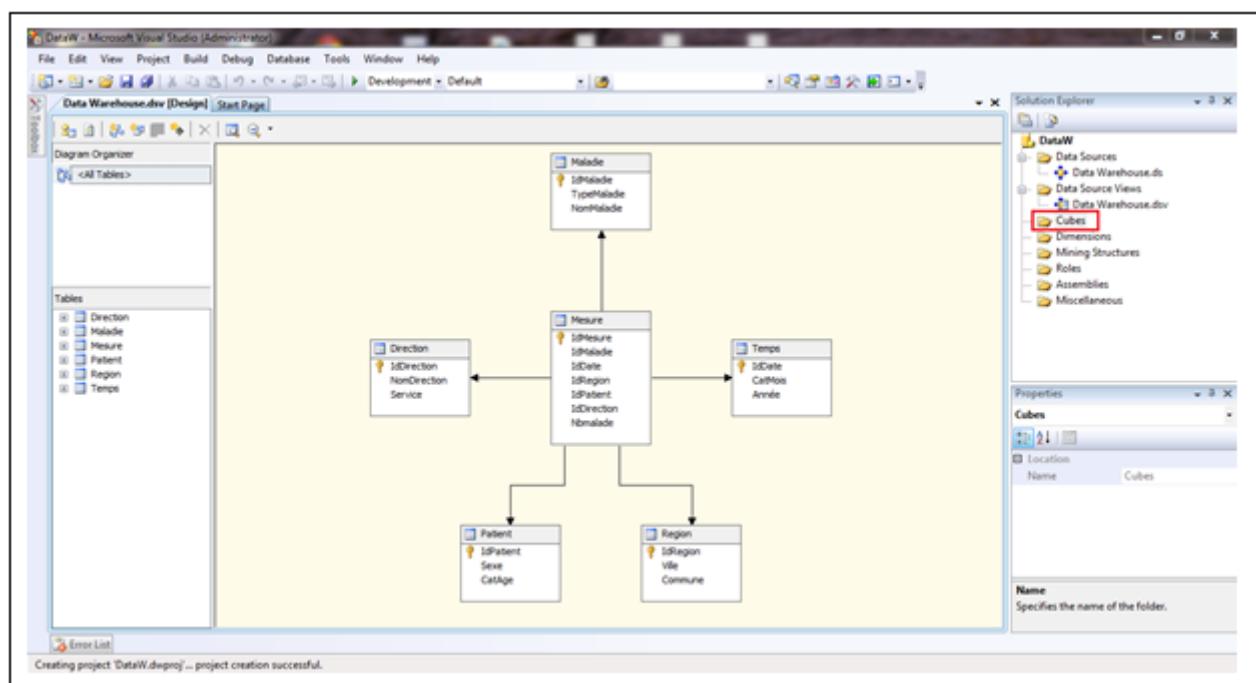


FIG. 5.10 – Nouveau Cube

5.3.5 Modification les Dimensions

dans l'explorateur de solutions, sur la dimension "Maladie", nous avons ajouté aux attributs désirés sur le côté gauche.

Pour les autres dimensions nous avons suivi la même modification (Figure 5.11).

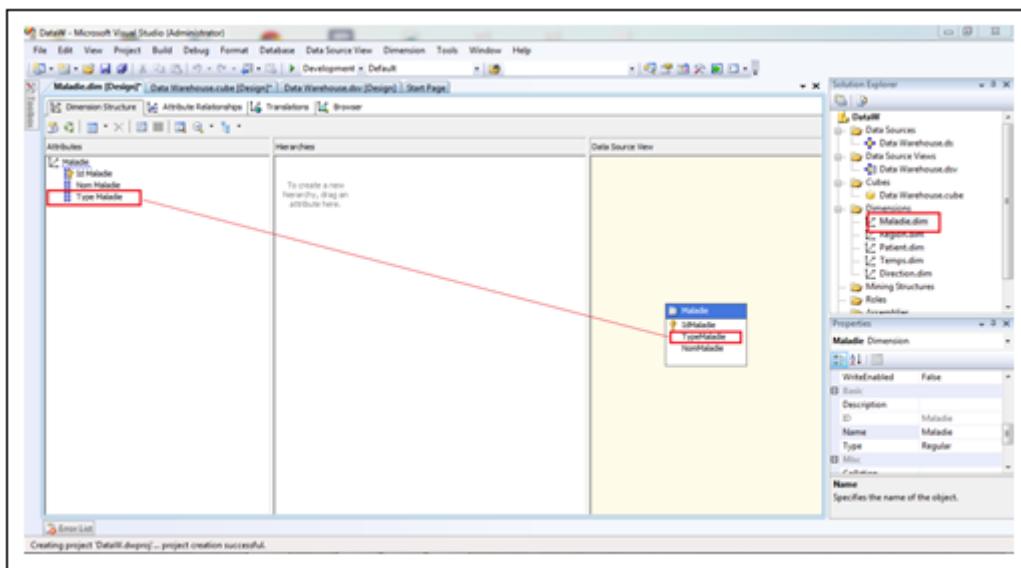


FIG. 5.11 – Ajout aux Attributs volet sur le côté gauche.

5.3.6 Crédation d'attribut Hiérarchie Dans la dimension Temps

Nous avons utilisé l'option "glissé-déplacé" sur les champs dans l'ordre à partir des attributs de fenêtre Hiérarchie (CatMois, Année) (Figure 5.12).

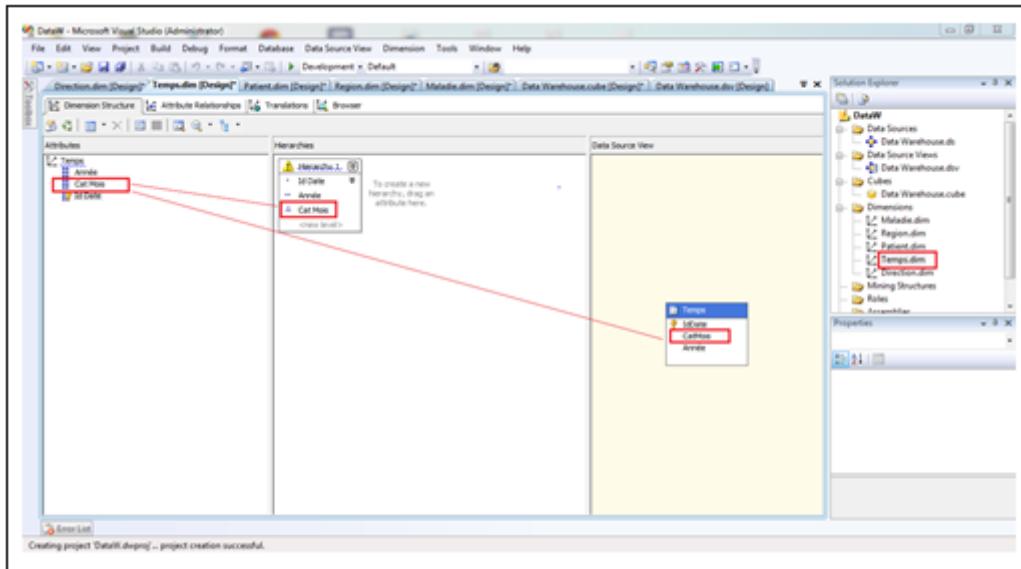


FIG. 5.12 – Crédation d'attribut Hiérarchie Dans la dimension Temps

5.3.7 Déploiement de Cube

Le Déploiement de cube ce fait en 03 étapes suivantes :

Etape 01 :

Dans l'explorateur des solutions, nous avons effectué un clique droit sur le nom du projet "DataW", en suite sélectionné "Propriétés".

Etape 02 :

Dans "Propriétés de configuration" nous avons sélectionné "Déploiement" (Figure 5.13).

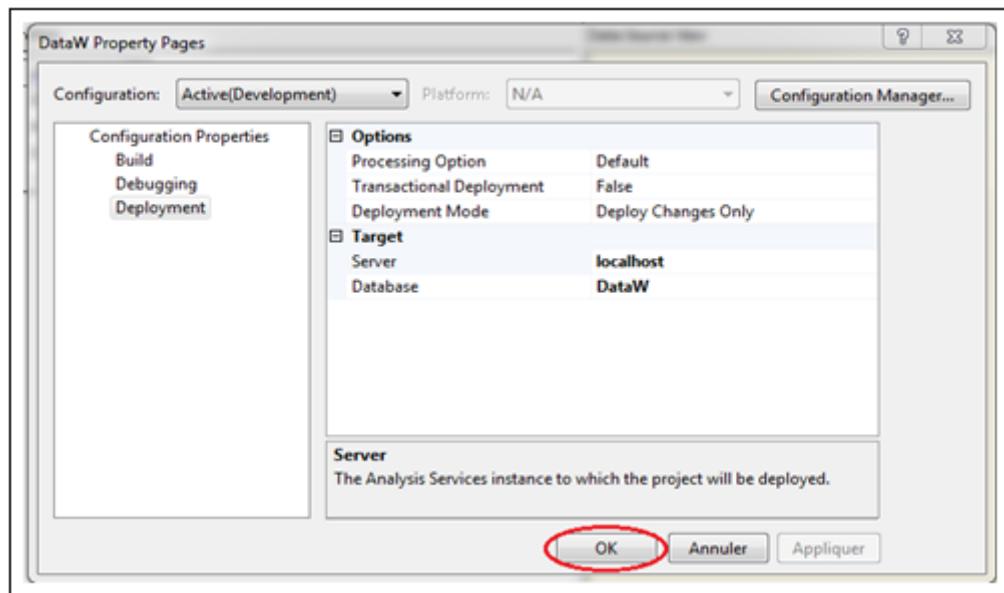


FIG. 5.13 – Déploiement Propriétés Première

Etape 03 :

Une fois que le déploiement est terminé, nous pouvons voir un message "terminé" dans les propriétés de déploiement.

5.3.8 Traitement de cube

Le Traitement de cube ce fait en 03 étapes suivantes :

Etape 01 :

Dans l'Explorateur de solutions, nous avons effectué un clique droit sur le nom du projet "DataW" en suite un clique sur "processus".

Etape 02 :

Exécuter pour traiter le Cube (Figure 5.14)

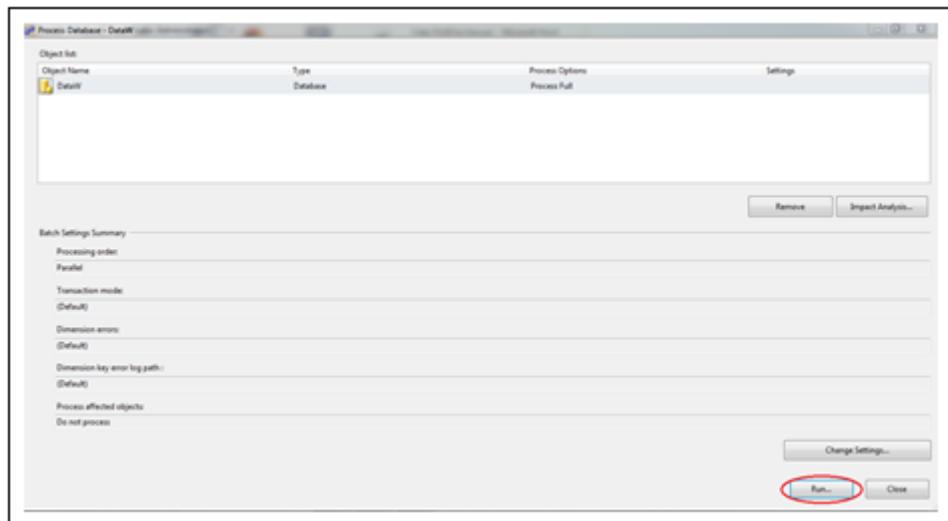


FIG. 5.14 – Exécuter pour traiter le Cube

Etape 03 :

Une fois le traitement est terminé, nous pouvons voir l'état comme "Successeur" Processus (Figure 5.15).

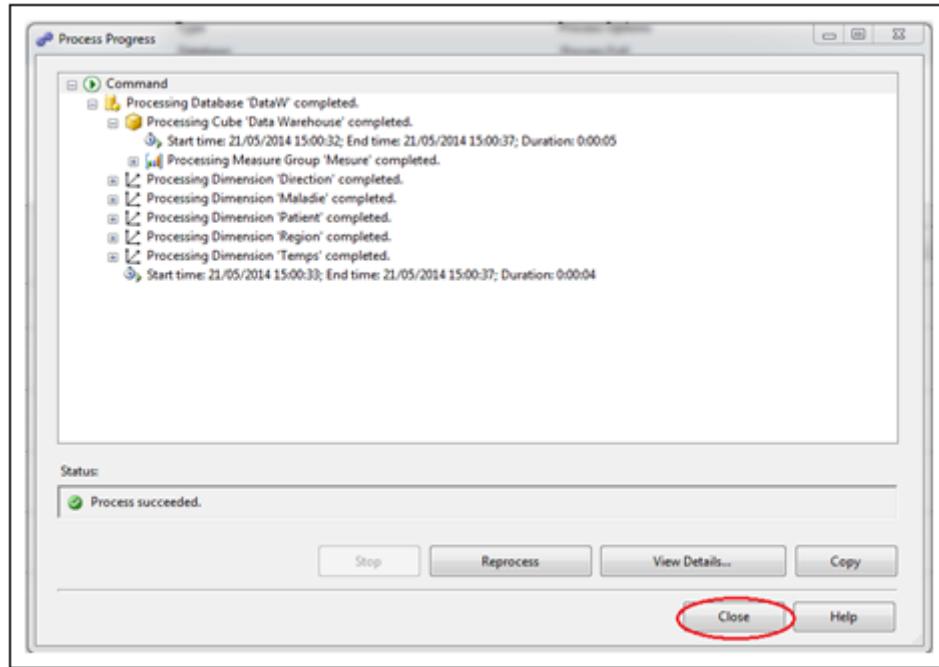


FIG. 5.15 – Traitement terminé

5.3.9 Parcourir le Cube d'analyse

Pour parcourir le Cube d'analyse nous avons entré dans l'explorateur de solutions, en suite un clique droit sur le nom du cube (DataWarehouse), à la fin, un clique sur Parcourir (Figure 5.16) .

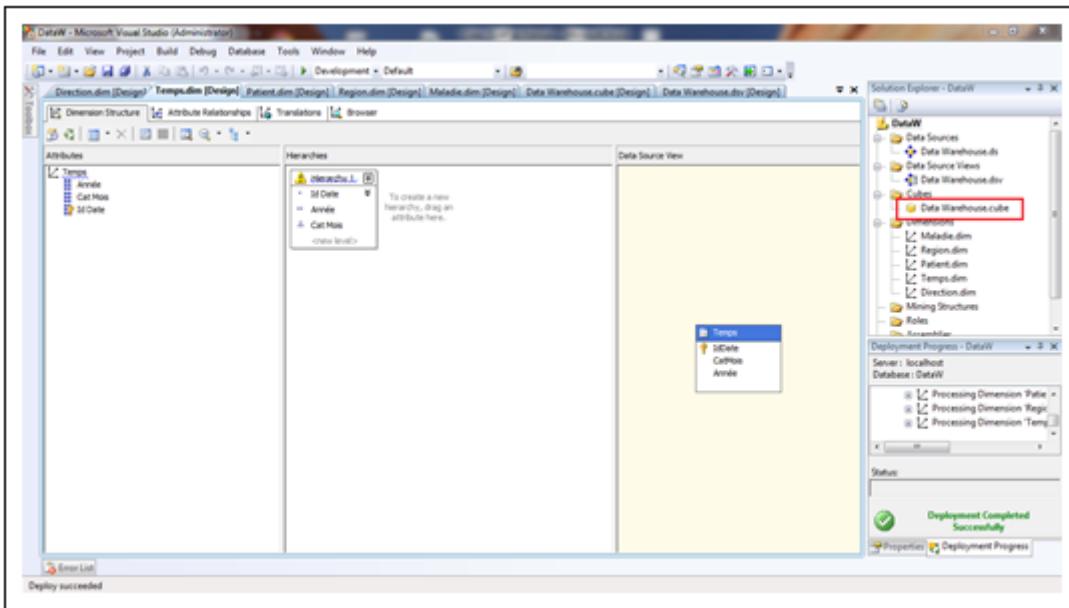


FIG. 5.16 – Parcourir le Cube d'analyse

Ensuite nous avons cliqué sur le bouton droit sur une mesure de la table de fait située sur la gauche dans le groupe de mesure et nous avons sélectionné "add to data area", la même chose pour les attributs de dimensions en spécifiant la forme de l'attribut : " add to row area " pour les lignes et " add to column area" pour les colonnes (Figure 5.17).

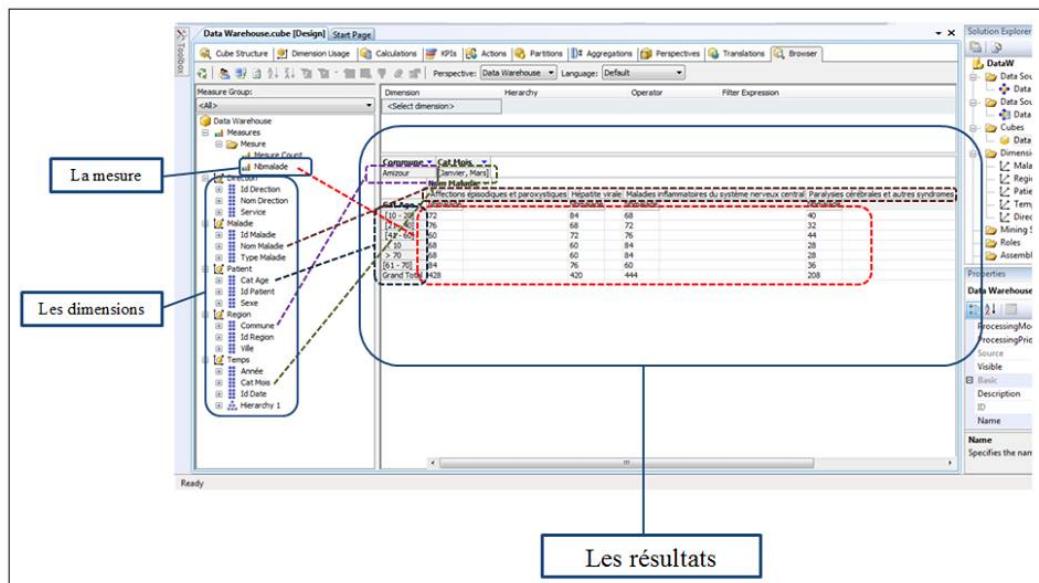


FIG. 5.17 – exemple des résultats

5.4 Exemples des résultats trouvés

5.4.1 Exemple avec une dimension

Pour une représentation avec une dimension, nous avons choisi l'exemple suivant : comparaison le nombre de malade pendant une période de temps. (Figure 5.18)

The screenshot shows a data analysis interface with a sidebar and a main content area.

Left Sidebar:

- <All>
- Data Warehouse
 - Measures
 - Mesure Count
 - Nbmalade
 - Direction
 - Maladie
 - Patient
 - Region
 - Temps
 - Année
 - Members
 - Année
 - Cat Mois
 - Id Date
 - Hierarchy 1

Main Content Area:

- <Select dimension>
- Drop Filter Fields Here
- Cat Mois ▾

[Avril, Juin]	[Janvier, Mars]	[Juillet, Septembre]	[Octobre, Décembre]	Grand Total
Nbmalade	Nbmalade	Nbmalade	Nbmalade	Nbmalade
12581	14251	16548	15556	61425

- Drop Row Fields Here

FIG. 5.18 – Exemples de résultats avec une dimension

5.4.2 Exemple avec deux dimensions

Pour une représentation avec deux dimensions, nous avons choisi l'exemple suivant : le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes. (Figure 5.19)

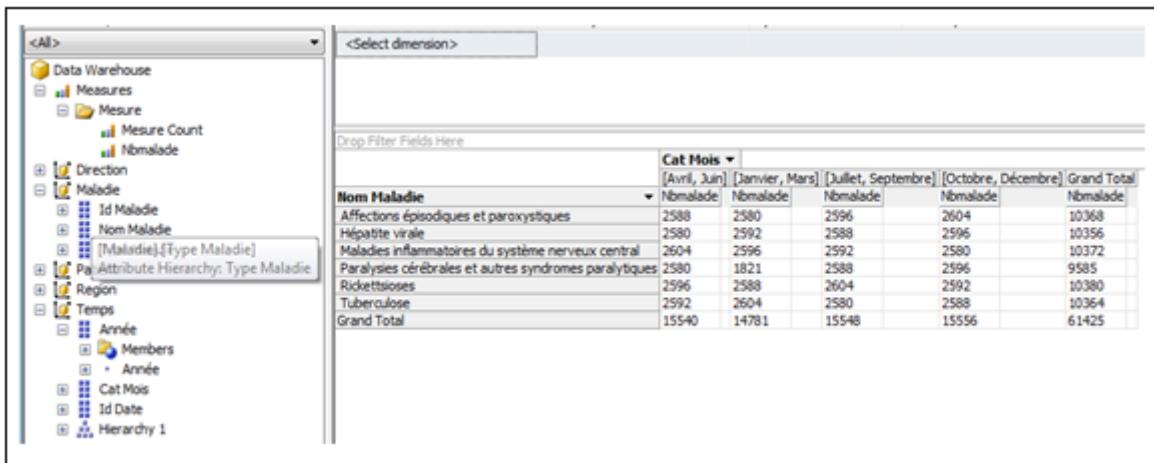


FIG. 5.19 – le nombre de malades pendant des périodes de temps différentes et pour des maladies différentes

5.4.3 Exemple avec trois dimensions

Pour une représentation avec trois dimensions, nous avons choisi l'exemple suivant : Le nombre de malade pendant des périodes et des tranches d'Age différentes. Pour cela nous avons testé 3 fois les noms de maladies et les catégories de mois en changeant à chaque fois la catégorie d'âge. (Figure 5.20 et 5.21 et 5.22)



FIG. 5.20 – Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 10ans et 20ans.

Cat Age : [21 - 40]

Cat Mois : [Avril, Juin] [Janvier, Mars] [Juillet, Septembre] [Octobre, Décembre] Grand Total

Nom Maladie	Nmalade	Nmalade	Nmalade	Nmalade	Nmalade
Affections épisodiques et paroxystiques	436	428	444	432	1740
Hépatite virale	428	420	436	444	1728
Maladies inflammatoires du système nerveux central	432	444	420	428	1724
Paralysies cérébrales et autres syndromes paralytiques	428	298	436	444	1606
Rickettsioses	444	436	432	420	1732
Tuberculose	420	432	428	436	1716
Grand Total	2588	2458	2596	2604	10246

FIG. 5.21 – Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 21ans et 40ans.

Cat Age : [41 - 60]

Cat Mois : [Avril, Juin] [Janvier, Mars] [Juillet, Septembre] [Octobre, Décembre] Grand Total

Nom Maladie	Nmalade	Nmalade	Nmalade	Nmalade	Nmalade
Affections épisodiques et paroxystiques	420	432	428	436	1716
Hépatite virale	432	444	420	428	1724
Maladies inflammatoires du système nerveux central	436	428	444	432	1740
Paralysies cérébrales et autres syndromes paralytiques	432	310	420	428	1590
Rickettsioses	428	420	436	444	1728
Tuberculose	444	436	432	420	1732
Grand Total	2592	2470	2580	2588	10230

FIG. 5.22 – Le nombre de malade pendant des périodes de temps différentes et pour des maladies différentes pour la catégorie d'Age entre 41ans et 60ans.

Représentation graphique

Notre application contient plusieurs interfaces, commençant par l'interface principale illustrée dans la figure (figure 5.23), depuis cette dernière nous pouvons accéder à d'autres interfaces relativement aux mesures que nous voulons analysée (figure 5.24), pour ensuite, imprimer le rapport final. (figure 5.25)



FIG. 5.23 – l'interface principale.

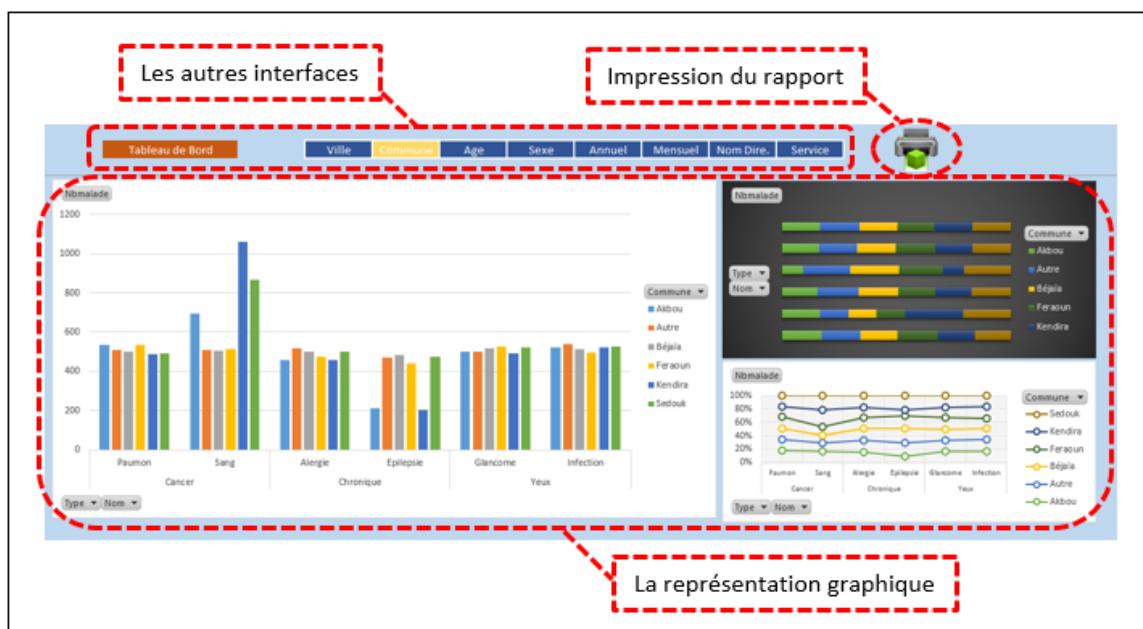


FIG. 5.24 – Interface commune.

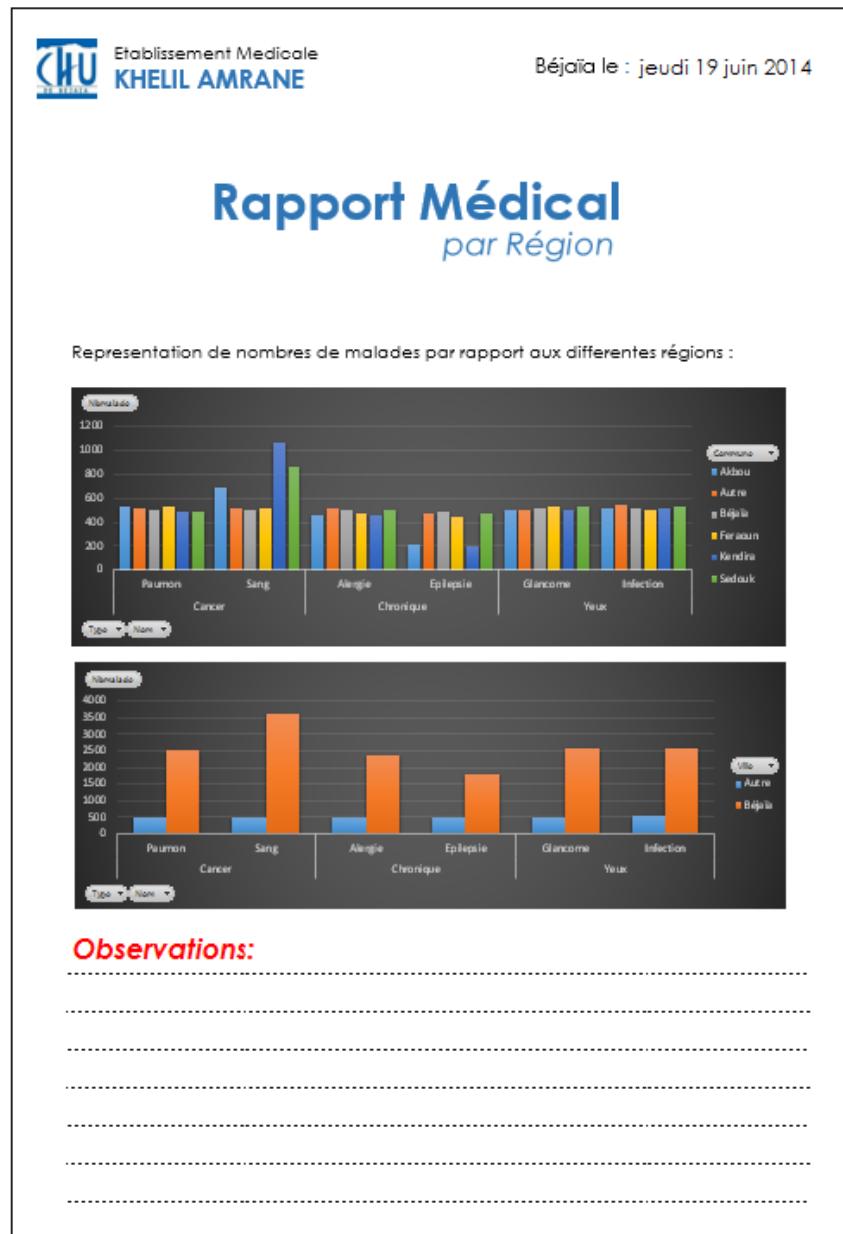


FIG. 5.25 – le rapport final.

Conclusion

Au cours de ce chapitre, nous avons implémenté un entrepôt de donnée pour l'hôpital Khelil Amrane, pour cela nous avons effectué deux(02) étapes majeurs, à savoir les codes T-SQL et la création d'un cube OLAP.

Cette implémentation nous permet d'avoir une vue multidimensionnel sur les données, autrement dit, elle nous permet d'avoir des statistiques importantes par rapport aux différents besoins, ce qui permettra au personnel médicale de faire des analyses et prendre les bonnes décisions.

Conclusion générale

Le travail qui nous a été confié est la construction d'un entrepôt de données " Data warehousing " en utilisant des outils tel que Sql server ou Oracle

Notre démarche est structurée en trois phases principales : études des déférents outils d'implémentation d'entrepôts de données (SQL Server, Oracle, Pentaho et Talend Opens Studio), conception du système et enfin le déploiement.

Dans une première phase, nous avons effectué une étude sur des différents outils d'implémentations d'entrepôt de données (SQL Server, Oracle, Pentaho, Talend Open Studio), suivi par une comparaison entre ces outils, après avoir analysé les résultats, nous avons constaté que le SQL Server est l'outil le plus adapté pour notre application.

La deuxième phase a été consacrée à la conception de notre entrepôts de données. Commençant par la spécification des besoins fonctionnels et non-fonctionnels, puis pour une modélisation multidimensionnelle. Plusieurs sujets intéressants ont été identifiés, faisant correspondre les besoins de l'hôpital Khelil Amrane avec la réalité des informations disponibles.

La dernière phase concerne le déploiement du système, nous avons construit le modèle physique de notre entrepôt. Ce dernier est basé sur le modèle multidimensionnelle. Les cubes OLAP construits sauvegardent l'agrégation des données stockées dans l'entrepôt. En matière d'alimentation, nous avons créé la base de données multidimensionnelle et nous avons également créé une application qui contient des interfaces utilisateurs. Notre système est implémenté sur SQL Server. Ce mémoire présente notre première expérience dans la mise en oeuvre d'un entrepôt de données. Nous avons tout de même rencontré des difficultés qui nous ont empêchés, en quelque sorte, d'atteindre des résultats satisfaisants à savoir : le manque de documentations, le manque de temps et l'absence des informations au niveau de l'hôpital Khelil Amrane.

Durant le déroulement de notre projet, nous avons acquis une bonne expérience dans le domaine des SI, plus précisément les SI décisionnels basés sur le Data Warehousing. Nous avons bénéficié des connaissances concernant certaines outils d'implémentation d'entrepôt de données (Oracle, Pentaho et Talend Opens Studio) et pour le SQL Server des connaissances plus détaillées.

Perspective

Dans le futur on aimeraït avoir un système d'information opérationnelle au niveau de l'hôpital Khelil Amrane, pour pouvoir construire un système décisionnel avec des données réel, et aussi, améliorer la sécurité et utiliser des fonctionnalités plus performante pour des grandes masses de données.

References

Bibliographie

- [1.1]** : David maisons, " DATAWAREHOUSE et DATAMINING ", rapport présenté en vue d'obtenir de l'examen probatoire en système d'information, CENTRE DE VERSAILLES,Edition ,11 décembre 2006.
- [1.2]** : Ladjel Belatreche, " Utilisation des index et de la fragmentation dans la conception logique et physique d'un entrepôt de données ",thèse pour obtenir le garde de Docteur en informatique, université de Clermont ferrand II ,2000.
- [1.3]** : w.Stanislas Kabore, " Mise en place d'un datamart concernant la paie du personnel de l'Etat ", mémoire d'ingénieur de conception en informatique , université POLYTECHNIQUE DE BOBO-DIOULASSO , 2004-2005.
- [1.4]** : Golfarelli M. , " From user requirements to conceptual design in data warehouse design-a survery ",Edition, 2009.
- [1.5]** : Carlos Coronel, Steven Morris et Peter Rob, Database Systems : Design, Implementation, and Management, Edition, Cengage Learning - 2012.
- [1.6]** : Sam Lightstone, Toby J. Teoreyet Tom Nadeau,Physical Database Design, Edition , Morgan Kaufmann - 2007.
- [1.7]** : S. Sumathi, S. Esakkirajan, Fundamentals of Relational Database Management Systems, Edition , Springer - 2007.
- [1.8]** : Thomas M. Connolly - Carolyn E. Begg, Database systems : a practical approach to design, implementation, and management, Edition , Pearson Education - 2005.
- [1.9]** : Ralph Kimball, Laura reeves, Magry Ross, Warren Thornthwaite , "Concevoir et déployer un data Warehouse ", guide de conduite d'un projet ,editions Eyrolles,2000.
- [1.12]** : Ladjel Bellatreche, " La conception physique des datawarehouse ", LISI/ENSMA Téléport2 - 1 , avenue Clément Ader 86960 Futuroscope France ,Edition,2006 .

[1.13] : Patrick O'Neil - Elizabeth O'Neil, Database—principles, programming, and performance, Edition , Morgan Kaufmann - 2001.

[1.14] : Djamel gara, compression dans les entrepôts de données pour l'amélioration des performances, université du Québec à Montréal, mémoire présenté comme exigence partielle de la maîtrise en informatique de gestion, janvier 2013.

[1.15] : Michael R. Groh, Access 2010 Bible, John Wiley and Sons , Edition , 2010.

[1.16] : Latif Al-Hakim, Challenges of managing information quality in service organizations , Edition , Idea Group Inc (IGI) - 2007.

[1.17] : Philip J. Pratt - Joseph J. Adamski, Concepts of Database Management, Edition , Cengage Learning - 2007.

[1.18] : Demmou L. Exploration de problèmes de performance d'un entrepôt de données , Edition , 2011.

[1.19] : Alain Venot, Anita Burgun et Catherine Quantin, Informatique Médicale, e-Santé - Fondements et applications, Springer Science & Business, Livre ,18 janvier 2013.

[1.20] : R.A. Moeller , Distributed data warehousing using Web technology, Edition, 2001.

[1.21] Ralph Kimball,Laura Reeves Margy Ross ,Warren Thorthwaite ,Concevoir et déployer un DW,Livre,2001.

[1.22] : Jef Wijsen, Université de Mons -Hainaut,Février , Data Mining et Data Warehousing, Edition, 2001.

[1.23] : Inmon WH. " Building the data warehouse ". Wiley , Edition , 1992.

[1.24] : A. Doucet and S. Gangarski. Entrepôts de données et Bases de Données multidimensionnelles, Chapitre 12 du Edition : Bases de Données et Internet, Modèles, langages et systèmes. Editions Hermes , 2001.

[1.25] : KHOURI S. thème : Modélisation conceptuelle à base ontologique d'un entrepôt de données (magistère) 2008-2009.

[1.26] : An IT Mandate , Providing OLAP to User-Analysts , Edition , 19 avril 2012.

[1.28] : Attribute Value Reordering for Efficient Hybrid OLAP, Edition , 15 mai 2012.

[2.7] : Glyn Moody, Computerworld UK. "Comment faites-vous une Pentaho ?" 5 mai 2010. ,Edition.

[2.8] : Seth Grimes, InformationWeek. "Open-Source BI Pentaho démarrage fait ses débuts." 16 Juin 2005. Edition.

[2.9] : Glyn Moody, Computerworld UK. "How Do You Make a Pentaho?" , Edition, May 5, 2010.

- [2.10]** : Seth Grimes, InformationWeek. "Open-Source BI Startup Pentaho Makes Debut." June 16, Edition , 2005.
- [2.11]** : MadanSheina, Ovum. "Pentaho BI Suite Enterprise Edition." , Edition , September 15, 2010.
- [2.12]** : Steven Brown, San Francisco Business Times. "Florida's Pentaho hires Quentin Gallivan as CEO in San Francisco.", Edition , October 4, 2011.
- [2.13]** : Madan Sheina, Ovum. "Pentaho BI Suite Enterprise Edition." Septembre 15 2010.Livre .
- [2.14]** : Surya Mukherjee, Ovum. "Pentaho expands coverage for Big Data." ,Edition , March 8, 2012.
- [2.15]** : James Kobielsus, Forrester Research. "The Forrester Wave : Enterprise Hadoop Solutions." , Edition , February 2, 2012.
- [2.16]** : David Menninger, Ventana Research. "Pentaho 4 Unites Enterprise Business intelligence and Data Integration." , Edition , June 22, 2011.
- [2.17]** : Pedro, Alves"Back to basics : Step by step Pentaho + Ctools installation" , Edition , December 15, 2011.
- [2.18]** : Will, Gorman Pentaho Wiki "Pentaho BI Server Marketplace Plugin , Edition , Februay 17, 2012.
- [3.5]** : " Anne-Cécile Caron Licence MIAGE - Bases de Données Les triggers, Edition, 2013-2014".
- [3.6]** : " Georges Gardarin, Bases de données, Edition, 5e tirage 2003".
- [3.7]** : " Daniel Lemire, L'informatique des entrepots de données, Edition , SEMAINE 13 L'exploration des données".
- [3.8]** : Steven Brown, San Francisco Business Times. "engage Pentaho Floride Quentin Gallivan comme chef de la direction de San Francisco." 4 Octobre 2011.Edition.
- [3.10]** : Surya Mukherjee, Ovum. " Pentaho étend la couverture pour Big Data ".Edition 8 mars 2012.

Webographie

[1.10] : <http://www.jbase.com>.

[1.11] : <http://www.temenos.com/Sectors/>.

[1.29] : daniel-lemire.com.

[1.29] : minet.uni-jena.de.

[1.30] : <http://www.softcomputing.com/fr/referentiel-crm/datawarehouse.html>.

[2.1] : <http://technet.microsoft.com/fr-fr/library/ms175609%28v=sql.90%29.aspx>.

[2.2] : <http://www.microsoft.com/france/serveur-cloud/sql/2012/business-intelligence.aspx#analysis>.

[2.3] : <http://www.microsoft.com/france/serveur-cloud/sql/2012/business-intelligence.aspx#reporting>.

[2.4] : <http://technet.microsoft.com/fr-fr/library/ms17560928v=sql.9029.aspx>.

[2.5] : <http://fr.talend.com/open-source-provider/contacts.php>.

[2.19] : <http://fr.talend.com/open-source-provider/contacts.php>.

[2.20] : <http://fr.talend.com/open-source-provider/alliance-initiatives.php>.

[2.21] : <http://www.microsoft.com/france/serveur-cloud/sql/2012/business-intelligence.aspx#analysis>.

[2.22] : <http://www.microsoft.com/france/serveur-cloud/sql/2012/business-intelligence.aspx#reporting>.

[2.23] : <http://fr.talend.com/open-source-provider/alliance-initiatives.php>.

[3.1] : <http://www.codeproject.com/Articles/652108/Create-First-Data-WareHouse>.

[3.2] : <http://www.limsi.fr/Individu/anne/cours>.

[3.3] : <http://www.wikipedia.org/>.

[3.9] : <http://churriwifi.wordpress.com/2010/06/01/comparing-talend-open-studio-and-pentaho-data-integration-kettle/>.

[3.11] : http://blog.developpez.com/zinzineti/p10474/sql-server_2008/sql_server_metadonnees_niveau_table.

[3.12] : <http://msdn.microsoft.com/fr-fr/library/ms186778.aspx>.

[3.13] : http://www.memoireonline.com/04/10/3347/m_Mise-en-place-dun-système-d'information1.html.

[3.14] : [http://msdn.microsoft.com/fr-fr/library/hh995351\(v=sql.15\).aspx](http://msdn.microsoft.com/fr-fr/library/hh995351(v=sql.15).aspx).

[3.15] : <https://help.talend.com/display/HOME/Talend+Enterprise+Data+Integration>.

[3.16] : http://fr.wikiversity.org/wiki/ERP_ORACLE.

[3.17] : http://minitoolkit.net/pentaho/installer_pentaho_windows.html.

[3.18] : http://docs.oracle.com/cd/B28359_01/server.111/b32024/partition.htm.

[3.19] : <http://www.apsql.com/articles/Partition.html>.

[3.20] : http://www.squaredba.com/move-a-range-list-partition-results-ora-14257_301.html.

[3.21] : <http://www.excindex.com/database/615559.html>.

[3.22] : www.oracle.com.

[3.23] : www.microsoft.com.

[3.24] : www.talend.com.

[3.25] : www.pentaho.com.

Résumé

De nos jours, les nouvelles technologies de l'information permettent de concevoir des systèmes d'information particulièrement performants et novateurs. Avec l'apparition des entrepôts de données, tous les utilisateurs peuvent désormais accéder facilement à l'information stratégique, ceci permet à l'entreprise d'être plus réactive dans sa prise de décisions.

Concevoir un Entrepôt de données est reconnu actuellement comme une tâche cruciale sur laquelle peut reposer la réussite d'un projet d'entreposage. L'étude des approches d'intégration nous a permis de constater qu'un entrepôt de données dans sa construction, peut être assimilé à un système d'intégration matérialisé par des concepts multidimensionnels.

Ce mémoire traite la conception et l'implémentation d'un entrepôt de données métier pour le suivi des maladies, dans les quelle le personnel médical puissent analyser les résultats et prendre des décisions.

L'approche retenue pour la réalisation de notre entrepôt de données est divisée en quatre phases principales : dans une première phase, nous avons effectué une étude sur des différents outils d'implementations d'entrepôt de données (SQL Server, Oracle, Pentaho, Talend Open Studio), suivie par une comparaison entre ces outils, ensuite, l'identification des besoins et la conception et à la fin l'implémentation de l'entrepôt.

Abstract

Today, new information technologies are used to design particularly effective and innovative information systems. With the advent of data warehouses, all users can now easily access to strategic information, this allows the company to be more responsive in its decision making.

Designing a Data Warehouse is now recognized as a crucial task that can stand the success of a storage project. The study of integration approaches we found a Data Warehouse in its construction, can be likened to an integration system materialized by multidimensional concepts.

This thesis discusses the design and implementation of a business data warehouse for monitoring disease in which medical personnel can analyze the results and make decisions.

The approach to achieving our data warehouse is divided into four main phases : In the first phase, we conducted a study on the various tools of data warehouse implementations (SQL Server, Oracle, Pentaho, Talend Open Studio), was a comparison between these tools, then the identification of needs and the design and implementation at the end of the warehouse.