

Ministère de l'Enseignement Supérieur et de la Recherche ScientifiqueUniversité Saad Dahlab de BlidaFaculté des sciences de l'ingénieur

Département d'informatique

Mémoire Présenté par :

BENZITOUNI FAYROUZ**KHELFI FODHIL****En vue d'obtenir le diplôme de Master****Domaine** : Mathématique et Informatique. MI**Filière** : Informatique.**Option** : Ingénierie des logiciels.**Titre** :

Conception et Réalisation d'un Système décisionnel, (Data Warehouse NoSQL, Tableau de bord), pour l'activité commerciale et services après-vente d'un concessionnaire automobile.

Promoteur : Melle. N.Boustia**Encadreur** : Mr. K.Boussebat**Organisme d'accueil** : Icon Software**Soutenue le** : 02-07-2012**devant le jury composé de** :

-Mme Ouahrani

Président

-Mme Arkam

Examineur

-Melle Azzouz

Examineur

- promotion 2011/2012 -

Remerciements



Nous remercions dieu pour les heures de courage, de patience, et de sagesse qu'il nous a inspire.

Nous tenons à remercier aussi tous nos enseignants durant nos cinq années d'études qui ont contribué à notre formation, et qui nous ont fait part de leur savoir, particulièrement notre promotrice Melle Boustia,

Nous tenons aussi a présenter nos plus sincères remerciements et notre profonde reconnaissance a notre encadreur, Mr Boussebat Khaled pour son soutien et ses conseils précieux ainsi qu'a toute l'équipe d'Icon software,

Sans oublier nos chères familles qui nous ont soutenues.

Et tous ceux qui ont participé de près ou de loin à la réalisation de ce travail.

MERCI A TOUS

Dédicaces

A mes très chers parents, sans qui je n'aurais jamais vu la lumière du jour, ni devenu ce que je suis. Puissent-ils trouver en ce modeste travail toute la gratitude et la reconnaissance d'un fils dévoué.

A toute ma famille et à mes petits neveux

A mes amis: mohamed, maamer, Azou, Lyas, Amine,

Sidali, Houcem, Lotfi, Abdou,

Nassim , aghilas , hamid , sadek , rahim, manel , hind, nassiba.

A tous mes amis que je n'ai pas pu citer

A mon binôme fayrouz et à toute sa famille.

Je dédie ces moments inoubliables

A tous ceux qui, ne serait-ce qu'une fois, me furent d'une aide aussi petite soit-elle; Qui, de près ou de loin, ont contribué à l'élaboration du présent travail. A tous ceux qui m'aiment et à tous ceux pour qui je compte.

Je dédie le fruit de mon travail

Khelfi fodhil

Dédicaces

A mes très chers parents, sans qui je n'aurais jamais vu la lumière du jour, ni devenu ce que je suis. Puissent-ils trouver en ce modeste travail toute la gratitude et la reconnaissance d'une fille dévouée

A mes chers frères : Samir, Lotfi, Fawzi.

A mes belles sœurs : Hassina, Meriem, Amel

A mes adorables neveux et nièces

A mon fiancé ainsi que tous mes amis

A mon binôme fodal.

A toute ma famille et tous ceux qui m'aiment et à tous ceux pour qui je compte.

Je dédie le fruit de mon travail

Benzitouni Fayrouz

Sommaire :

Introduction générale	1
Problématique et objectifs	3

PARTIE 1: ETAT DE L'ART

CHAPITRE 1 : BUISNESS INTELLIGENCE et SYSTEME DECISIONNEL

Introduction	5
1) Définition Business Intelligence	5
2) Etapes du processus de Business	6
3) Outils de Business Intelligence	7
4) Définition des Systèmes Décisionnel	7
5) Concept de base	8
6) Définition et caractéristiques	9
7) Historique des systèmes décisionnels	10
7.1) L'Infocentre	10
7.2) ENTREPOTS DE DONNEES	11
7.3) Les bases de données multidimensionnelles	13
Conclusion	14

CHAPITRE 2 : DATA WAREHOUSE

Introduction	15
1) Définition	15
2) Composants de base d'un Data Warehouse	17
3) Objectifs d'un Entrepôt de données	18
4) L'alimentation d'un Data Warehouse	19
4.1) Les phases de l'alimentation « E.T.L »	20
4.2) Politiques de l'alimentation	21
4.3) Les outils E .T.L	22
5) des données du Data Warehouse	22
5.1) Le tableau de bord	22
5.1.1) Définition	22

5.1.2) Rôle d'un tableau de bord	23
5.1.3) Exigences d'un tableau de bord	23
5.1.4) Type des tableaux de bord	23
Conclusion	24

CHAPITRE 3 : NOSQL ET MAPREDUCE

Introduction	25
1) Pourquoi le NoSQL ?	25
2) Solutions NoSQL	26
3) Les limites des systèmes de SGBD relationnels face aux usages	29
4) Caractéristiques SGBD NoSQL	32
5) MapReduce	33
5.1) L'étape Map	33
5.2) L'étape Reduce	33
5.3) Exemple	35
5.4) Avantages et Inconvénients	36
Conclusion	37

PARTIE 2 : CONCEPTION DE LA SOLUTION

CHAPITRE 4: ETUDE DE L'EXISTANT ET IDENTIFICATION DES BESOINS

Introduction	38
1) Présentation du domaine	38
2) Système opérationnel	39
3) Aperçu sur la procédure de prise de décision	39
4) Etat du décisionnel	41
5) APPROCHE DEFINITION DES BESOINS	41
5.1) Elaboration de questionnaires	41
5.2) L'ENTRETIEN	42
6) Extraction des besoins	42
Conclusion	46

CHAPITRE 5 : CONCEPTION DES ELEMENTS DU SYSTEME

Introduction	47
1) Modélisation multidimensionnelle	47
1.1) Présentation	47
1.2) Description des activités	48
1.3) Description des dimensions	52
1.4) Description des activités	54
1.5) Schéma de l'entrepôt de données	59
2) Architecture détaillée du système	60
3) Construction du Data Warehouse NoSQL	60
3.1) Liste des collections	61
4) Alimentation du Data Warehouse NoSQL	66
4.1) Demande et Acquisition des données	66
4.2) Transformation	67
4.3) chargement	67
5) Construction du Data Warehouse Classique	67
6) Alimentation du Data Warehouse Classique	71
7) Exemple de comparaison entre Data Warehouse classique et Data Warehouse NoSQL	72
8) MapReduce	74
9) Conception Du Tableau de bord	75
9.1) Diagrammes des cas d'utilisation et de séquence	75
Conclusion	83

PARTIE 3 : MISE EN OEUVRE

CHAPITRE 6 : CHOIX TECHNOLOGIQUES ET ARCHITECTURE PHYSIQUE

Introduction	84
1) OUTILS DE DEVELOPPEMENT	84
1.1) MongoDB	84
1.2) JMongoBrowser	85
1.3) PostgreSQL 9.1	85
1.4) Talend	85
1.5) Pentaho Data Integration (Kettle)	86

1.6) Pentaho Business Analytics	86
2) Architecture physique du système	87
Conclusion	88

CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

Introduction	89
1) La construction du Data Warehouse NoSQL	89
2) La mise en œuvre de l'alimentation	90
2.1) Alimentation avec Talend	90
2.2) Alimentation avec 2F_Loader	93
3) La construction de l'outil de restitution	94
4) Sécurité et habilitations	102
5) Démonstration du système	104
5.1) Connexion au système	104
5.2) Consultation	106
Conclusion	108
Conclusion et perspectives	109

List des figures :

<i>Figure 1 : Etapes du processus Business Intelligence</i>	6
<i>Figure 2 : L'Infocentre</i>	10
<i>Figure 3 : Entrepôt de données</i>	12
<i>Figure 4 : Base Multidimensionnelle</i>	13
<i>Figure 5 : Composant de base d'un Data Warehouse</i>	18
<i>Figure 6 : Processus d'Alimentation</i>	21
<i>Figure 7 : Base de Données Clé-Valeur</i>	27
<i>Figure 8 : Base de Données Orientée colonnes</i>	28
<i>Figure 9 : Base de Données Orientée document</i>	28
<i>Figure 10 : Base de Donnée Orientée graphe</i>	29
<i>Figure 11 : MapReduce</i>	35
<i>Figure 12 : Exemple nombre d'occurrences d'un mot avec MapReduce</i>	36
<i>Figure 13 : Schéma représentatif des étapes d'édition des rapports</i>	40
<i>Figure 14 : Modèle Dimensionnel en flocon de l'activité «vente véhicules neufs»</i>	54
<i>Figure 15 : Modèle Dimensionnel en flocon de l'activité « vente pièces de rechange»</i>	55
<i>Figure 16 : Modèle Dimensionnel en flocon de l'activité «vente pièces ratées»</i>	56
<i>Figure 17 : Modèle Dimensionnel en flocon de l'activité «service après vente »</i>	57
<i>Figure 18 : Modèle Dimensionnel en flocon de l'activité «pilotage des commandes»</i>	58

Figure 19 : Schéma relationnel de la base décisionnelle	59
Figure 20 : Architecture du système	60
Figure 21 : Processus d'alimentation du Data Warehouse NoSQL	66
Figure 22 : schéma technique de l'application du MapReduce	74
Figure 23 : Architecture physique du system	88
Figure 24 : Interface JMongoBrowser	89
Figure 25 : Création d'une connexion dans Talend	90
Figure 26 : Récupération d'un schéma de la base source	91
Figure 27 : Création d'un nouveau Job dans Talend	91
Figure 28 : Job ETL piece	92
Figure 29 : Configuraton du tMap	92
Figure 30 : Processus de 2F_loader	93
Figure 31 : Processus du programme MapReduce	95
Figure 32 : Job vente pièce, client, wilaya avec Pentaho Data Integration.....	96
Figure 33 : Création d'un nouveau tableau de bord	97
Figure 34 : Choix Template du tableau de bord	98
Figure 35 : Choix type tableau de bord	98
Figure 36 : Sélection de la source de données existante	99
Figure 37 : Importer un fichier texte comme source de données	99
Figure 38 : Création de la source de données	100
Figure 39 : Choix des attributs	100
Figure 40 : Configurer chart designer	101

<i>Figure 41 : Tableau de bord créé</i>	101
<i>Figure 42 : Création d'un utilisateur Pentaho</i>	103
<i>Figure 43 : Création d'un rôle Pentaho</i>	103
<i>Figure 44 : Attribution des rôles aux utilisateurs</i>	104
<i>Figure 45 : Interface d'accueil</i>	105
<i>Figure 46 : Message d'erreur</i>	105
<i>Figure 47 : Tableau de bord suivi des chiffres d'affaire des services</i>	106
<i>Figure 48 : Tableau de bord suivi vente véhicules neufs 1</i>	107
<i>Figure 49 : Tableau de bord suivi vente véhicules neufs 2</i>	107
<i>Figure 50 : Partage des rapports</i>	108

Liste des diagrammes :

<i>Diagramme 1 : Diagramme de cas d'utilisation du Tableau de Bord.....</i>	76
<i>Diagramme 2 : Diagramme de séquence du processus de connexion au système (cas normal)</i>	77
<i>Diagramme 3 : Diagramme de séquence : Connexion au système (cas d'erreur)..</i>	77
<i>Diagramme 4 : Diagramme de cas d'utilisation du suivi des ventes pièces de rechange</i>	78
<i>Diagramme 5 : Diagramme de cas d'utilisation du suivi des ventes véhicules neufs</i>	79
<i>Diagramme 6 : Diagramme de cas d'utilisation du suivi service après vente.....</i>	80
<i>Diagramme 7 : Diagramme de séquence création d'un tableau de bord.....</i>	81
<i>Diagramme 8 : Diagramme de séquence remplissage fichier avec MapReduce...</i>	82
<i>Diagramme 9 : Diagramme de séquence remplissage du fichier avec Pentaho Data Integration.....</i>	82

Liste des tableaux :

<i>Tableau 1 : Tableau comparatif entre les systèmes transactionnels et les systèmes décisionnels</i>	9
<i>Tableau 2 : Comparaison entre Infocentre et entrepôt de données</i>	12
<i>Tableau 3 : Tableau comparatif entre les SGBD relationnel et NoSQL</i>	32
<i>Tableau 4 : Tableau résumé des indicateurs par axes d'analyse</i>	52
<i>Tableau 5 : Tableau des dimensions</i>	53
<i>Tableau 6 : Liste des collections du Data Warehouse</i>	67
<i>Tableau 7 : Liste des attributs des tables du Data Warehouse classique</i>	71

Résumé :

Ce mémoire consiste en la conception et la réalisation d'un système décisionnel pour l'activité commerciale et service après vente d'un concessionnaire automobile SID AUTO, le projet englobe la conception détaillée et la mise en œuvre de chaque composant du système décisionnel, en commençant par le Data Warehouse NoSQL, qui a été choisit pour le stockage des données pour le coté dynamique et pour se libérer des conditions du relationnel, pour une analyse rapide et efficace des données l'algorithme MapReduce est le plus adapté pour le calcul parallèle et distribué, vient ensuite l'exploitation des données, ou le tableau de bord a été choisit comme portail de restitution.

Abstract:

This thesis is the design and implementation of a decision support system for business activity and customer service of a car dealership SID AUTO, the project involves the detailed design and implementation of each component of decision-making system, starting with the NoSQL data Warehouse, which was chosen for data storage for the next release for this dynamic and relational conditions, for rapid analysis and efficient data, MapReduce algorithm is best suited for parallel computing and distributed, then comes the use of data, or the dashboard has been chosen as the portal of restitution.

Introduction générale :

C'est dans un environnement fortement complexe et hautement concurrentiel qu'évolue la majeure partie des entreprises. Ce climat de forte concurrence exige de ces entreprises une surveillance très étroite du marché afin de ne pas se laisser distancer par les concurrents et cela en répondant, le plus rapidement possible, aux attentes du marché, de leur clientèle et de leurs partenaires.

Pour se faire, les dirigeants de l'entreprise, quelque en soit d'ailleurs le domaine d'activité, doivent être en mesure de mener à bien les missions qui leur incombent en la matière. Ils devront prendre notamment les décisions les plus opportunes. Ces décisions, qui influenceront grandement sur la stratégie de l'entreprise et donc sur son devenir, ne doivent pas être prise ni à la légère, ni de manière trop hâtive, compte tenu de leur conséquences sur la survie de l'entreprise. Il s'agit de prendre des décisions fondées, basées sur des informations claires, fiables et pertinentes. Le problème est de savoir donc comment identifier et présenter ces informations à qui de droit, sachant par ailleurs que les entreprises croulent d'une part sous une masse considérable de données et que d'autre part les systèmes opérationnels (transactionnelles) s'avèrent limitées, voire inaptes à fournir de telles informations et constituer par la même un support appréciable à la prise de décision.

C'est dans ce contexte que les « systèmes décisionnels » ont vu le jour. Ils offrent aux décideurs des informations de qualité sur lesquelles ils pourront s'appuyer pour arrêter leur choix décisionnels.

Comme toute entreprise pariant sur l'intégration des solutions pour garder le rythme d'expansion de ses activités, SID AUTO a engagé Icon Software pour superviser et veiller à l'évolution de l'activité informatique du concessionnaire automobile.

Parmi les tâches confiées à cette entreprise, le projet que nous allons présenter tout au long de ce mémoire et qui fera l'objet de notre projet de fin d'étude. Il s'agit de la Conception et Réalisation d'un Système décisionnel (Data Warehouse NoSQL, Tableau de bord) pour l'activité commerciale et services après-vente des concessionnaires automobiles.

Pour mener à bien notre tâche, nous avons structuré notre document en trois parties : une première partie pour l'étude bibliographique sur le domaine de notre projet, une deuxième pour la conception du système et la dernière partie pour la mise en œuvre de ce système.

Problématique et objectifs :

Icon Software® est une Société de Services en Ingénierie Informatique (SSII) à forte valeur ajoutée, en plein croissance avec un réseau de compétences important. Grâce à une expertise technologique dans le domaine des technologies de l'information autour des environnements ouverts et standards (*Java, Linux et Android*), du fait de ses expériences d'intégration et de développement propose une offre de services innovants et complémentaires permettant aux entreprises de développer et de déployer en toute sérénité leurs projets informatiques.

L'entreprise **Icon Software®** s'est engagée dans le développement d'un système décisionnel de management des systèmes d'information de ses clients (concessionnaires automobile). Dans ce contexte, Icon Software se trouve dans l'incapacité d'analyser par ses moyens disponibles actuels (logiciels) le volume important de données régulièrement collectées de façon rapide, correcte et efficace, et dans l'incapacité d'exploiter les avantages que comportent les nouvelles méthodes d'analyse et de modélisation de données.

Les principales difficultés auxquelles elle fait face sont :

- Augmentation, complexité progressive des données et besoin d'une vue multiaxe afin de satisfaire les besoins des décideurs.
- Existence de problèmes et dysfonctionnements non justifiés au niveau des structures opérationnelles.
- Pertes dans la rentabilité et l'efficacité marketing des produits à cause de la difficulté de repérer, analyser les produits à forte valeur ajoutée, les caractéristiques du client et l'environnement concurrentiel.
- Non satisfaction des clients (après-vente).

- Risque de perte d'une part du marché dans le cas d'arrivée de nouveaux concurrents.

Le but est de proposer un système décisionnel qui assure un meilleur pilotage des activités commerciales et services après-vente des concessionnaires automobile, ainsi, on va introduire la notion des Data Warehouse NoSQL et l'analyse des données en utilisant l'algorithme MapReduce; donc une meilleure prise de décision.

L'objectif est de concevoir et déployer un nouveau système décisionnel qui se traduit par :

- La spécification des besoins des décideurs pour l'activité commerciale (véhicule, pièces de rechange) et service après-vente,
- La constitution d'un Data Warehouse NoSQL à base de modèle document,
- La définition des nœuds d'analyse (clé, valeur) et implémentation de l'algorithme d'analyse MapReduce,
- La constitution d'un portail de restitution (Tableau de bord) basé sur les technologies web et Java EE,

PARTIE 1 : ETAT DE L'ART

CHAPITRE 1 : BUISNESS INTELLIGENCE ET SYSTEME DECISIONNEL

CHAPITRE 2 : Data Warehouse

CHAPITRE 3 : NoSQL et MapReduce

PARTIE 1: ETAT DE L'ART

Cette partie comprend une étude bibliographique, des définitions ainsi que les concepts liés au domaine des systèmes décisionnels. Elle est structurée en trois chapitres, le premier étant des généralités sur la business intelligence et les systèmes décisionnels, le deuxième chapitre est consacré au Data Warehouse, le troisième présentera le NoSQL ainsi que le MapReduce.

CHAPITRE 1 : BUSINESS INTELLIGENCE ET SYSTEME DECISIONNEL

Introduction :

Business Intelligence ou l'informatique décisionnelle interprète les données complexes de l'entreprise et aide les dirigeants à prendre les meilleures décisions.

Son objectif est de définir les méthodes et outils permettant à une entreprise de mettre en place son projet décisionnel. Ces outils facilitent l'accès aux données globales de l'entreprise.

C'est pour cela, mettre en place un **Système décisionnel** capable de répondre aux besoins de l'entreprise est indispensable.

1) Définition Business Intelligence :

Business Intelligence (BI) est un concept qui fait l'objet de multiples définitions.

Le groupe de travail présidé par «Henri Martre» [MARTRE, 1994], en 1994, retient de **Business Intelligence** la définition suivante : «**Business Intelligence** peut être définie comme l'ensemble des actions de recherche, de traitements et de diffusion (en vue de son exploitation) de l'information utile aux acteurs économiques...».

En effet pour «Bernard Besson» et «Jean-Claude Possin» [BESSON et POSSIN, 1996], il s'agit de : «La capacité d'obtenir des réponses à des questions en découvrant des intelligences entre deux ou plusieurs informations préalablement mémorisées. L'entreprise mettra au service de cette capacité tous les moyens dont elle dispose pour saisir des opportunités ou détecter des menaces».

Business Intelligence désigne donc les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données d'une entreprise en vue d'ouvrir une aide à la décision et de permettre aux décideurs de l'entreprise d'avoir une vue d'ensemble de l'activité traitée.

Ce type d'application utilise en règle générale un entrepôt de données (ou Data Warehouse en anglais) pour stocker des données provenant de plusieurs sources hétérogènes et fait appel à des traitements par lots pour la collecte de ces informations. [BENDAVID, 2010].

Business Intelligence a pour rôle de valoriser l'information en la faisant circuler dans l'entreprise. Cette caractéristique peut s'appuyer sur les réseaux internes de l'entreprise comme l'intranet.

2) Etapes du processus de Business Intelligence:

Business Intelligence a pour objectif de permettre aux décideurs et managers de l'entreprise de disposer d'une information de valeur, à laquelle ils puissent se fier dans le cadre de leurs prises de décision. Pour cela, il s'agit de produire de l'information pertinente et à forte valeur ajoutée. Cette exigence doit se retrouver à travers les différentes phases du Processus suivant :

- Collecte de l'information,
- Traitement,
- Diffusion.

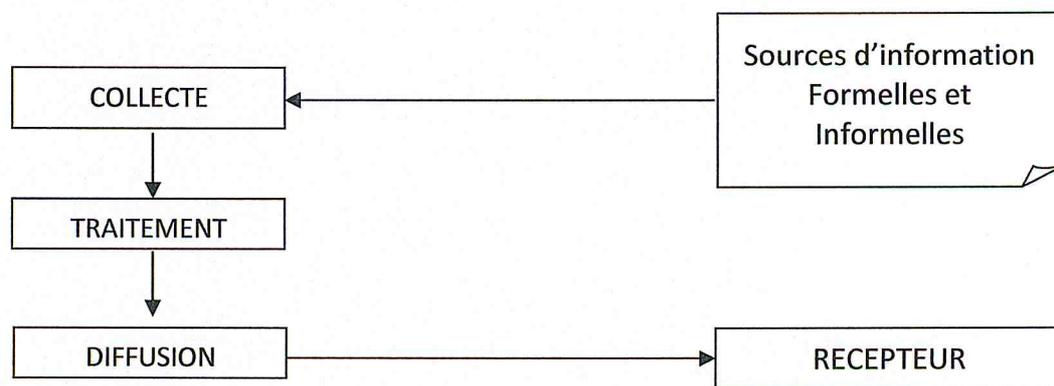


Figure 1 : Etapes du processus Business Intelligence.

- **La collecte ou recherche**, c'est la période de recherche où sont identifiées et exploitées les sources d'informations, ceci dans le cadre d'une planification.
- **L'analyse**, c'est-à-dire le traitement ou **l'exploitation**, c'est au cours de cette étape que les données et les informations passent à l'état de connaissance à travers un processus systématique d'évaluation, d'interprétation et de synthèse destiné à élaborer des conclusions répondant aux besoins de renseignements exprimés.
- **La diffusion** est l'acheminement des renseignements sous une forme appropriée (orale, écrite ou graphique) aux organes ayant exprimés la demande.

3) Outils de Business Intelligence :

Le moteur de recherche [QUESTER, 2004] est au cœur de la plate-forme **Business Intelligence**. Il est un élément d'une chaîne de valeur, qui consiste à collecter et extraire la donnée pour la transformer en information qui va permettre d'optimiser la prise de décision.

On distingue trois catégories d'applications que nous allons développer :

1. Les outils de recherche,
 2. Les outils de traitement,
 3. Les outils de diffusion et de partage.
- Les **outils de recherche** comprennent: les moteurs de recherche, les méta-moteurs et les agents intelligents.
 - Les **outils de traitement** regroupent les applications pour : traiter les données collectées, faciliter leur lecture, les trier, et les classer en catégories pertinentes.
 - Les **outils de diffusion** et de partage permettent de communiquer les bonnes informations aux bons interlocuteurs et de capitaliser les connaissances.

4) Définition des Systèmes:

« Un système d'information est un ensemble organisé de ressources (matérielles, logicielles, personnelles, données, procédures...) permettant d'acquérir, de traiter, de stocker des informations (sous forme de données, textes, images, sons...) dans et entre organisations». [Robert REIX, 2004].

Le choix de l'appellation *système* n'est pas anodin. Il reflète la logique sous-jacente considérant ce dernier comme un ensemble d'entités en interaction entre elles, que l'on pourrait considérer comme autant de maillons formant une chaîne. De ce fait, ce dernier peut être ainsi observé à différents degrés de précision, soit en le considérant comme un système d'information global, soit en accentuant le zoom afin de mettre en valeur deux sous systèmes.

5) Concept de base :

Il y'a deux systèmes d'information : transactionnel et décisionnel.

Le système d'information **transactionnel** : gère les applications quotidiennes et se rapproche à ce titre de la couche opérationnelle. Il est typiquement utilisé par les acteurs métiers afin de répondre à des besoins de simplification et d'automatisation et de gestion.

Le système d'information **décisionnel** : est utilisé pour aider à la prise de décisions de l'entreprise, et à ce titre doit permettre aux décideurs d'avoir un certain recul sur leur entreprise. Il fournit pour cela les informations nécessaires et pertinentes afin de faire les bons choix.

Le décisionnel est donc à l'information de l'entreprise ce que les mathématiques sont à la pensée.

Le tableau suivant résume les différences possibles entre les systèmes décisionnels et transactionnels selon les données et l'usage des systèmes.

Différence	 Systèmes transactionnels	 Systèmes décisionnels
Par les données	Orienté applications métiers	Orienté thèmes et sujets (activités)
	Situation instantanée	Situation historique
	Donnée détaillées et codées non redondantes	Informations agrégées cohérentes souvent avec redondance
	Données changeantes constamment	Informations stables et synchronisées dans le temps
	Pas de référentiel commun	Un référentiel unique
L'usage	Assure l'activité au quotidien	Permet l'analyse et la prise de décision
	Pour les opérationnels	Pour les décideurs
	Mise à jour et requêtes simples	Lecture unique et requêtes complexes transparentes
	Temps de réponse immédiats	Temps de réponse moins critiques
	Faibles volumes à chaque transaction	Large volume manipulé
	Conçu pour la mise à jour	Conçu pour l'extraction
	Usage maîtrisé	Usage aléatoire

Tableau 1 : Tableau comparatif entre les systèmes transactionnels et les systèmes décisionnels.

6) Définition et caractéristiques d'un système décisionnel :

Un **système décisionnel** est un ensemble de données organisées de façon spécifique, facilement accessibles et appropriées à la prise de décision .La finalité d'un système décisionnel est le pilotage de l'entreprise.

Les trois principales caractéristiques d'un système décisionnel sont :

- La capacité de traiter un volume de données important.

- Temps de réponse très élevés.
- Les requêtes plus complexes du point de vue informatique, car elles contiennent de nombreuses opérations de jointure et de regroupement.

7) Historique des systèmes décisionnels :

Lorsque les entreprises ont commencé à comprendre la valeur ajoutée apportée par les outils d'aide à la décision, elles ont immédiatement cherché à en bénéficier. Pour cela elles ont commencé à interroger leurs bases de données opérationnelles. Tout en poursuivant bien entendu leurs opérations quotidiennes, elles ont donc sollicité leurs applications opérationnelles, bien souvent au-delà de ce qu'elles étaient capables de supporter. Et les outils d'administration informatiques n'étaient à l'époque pas aussi perfectionnés que maintenant.

Des situations de tension sont donc apparues en interne dans certaines entreprises, entre les « **opérationnels** » qui demandaient, à juste titre, une disponibilité permanente de leurs applications de production, et les « **analystes** », qui demandaient, également à pouvoir accéder aux données pour réaliser leurs analyses. Mais la différence structurelle entre les requêtes (simples pour les systèmes opérationnels, complexes pour le décisionnel) rendait cette cohabitation presque impossible.

7.1) L'Infocentre :

La première solution trouvée était une solution de bon sens, c'est de **dupliquer les bases de données de production**, c'est-à-dire que chaque jour, chaque semaine, ou chaque mois, une copie des informations de production était réalisée, à l'identique, sur un autre ordinateur, spécialement pour les analystes.

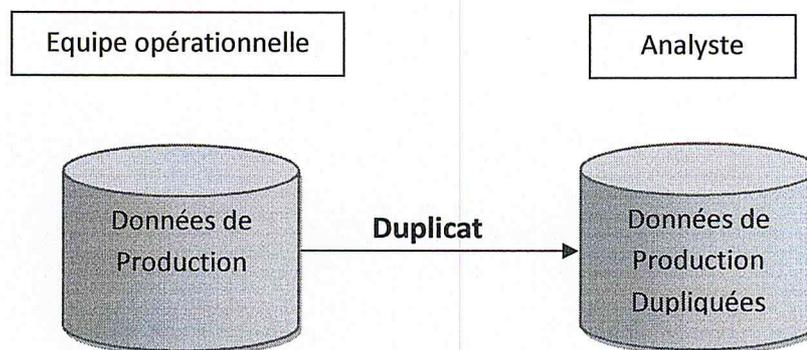


Figure 2 : L'Infocentre.

Cette solution permettait aux équipes opérationnelles de continuer à utiliser leurs applications, sans être nullement perturbés par les requêtes analytiques, et les analystes pouvaient prendre le risque de lancer des requêtes complexes, analysant par exemple le chiffre d'affaires suivant plusieurs dimensions (clients, produits, fournisseurs), sur plusieurs années, mois par mois, sans prendre le risque de bloquer le système opérationnel.

Ce mode de fonctionnement, en doublon total entre les deux systèmes, se révélait très coûteux. Les serveurs, les disques durs, les bases de données, devaient tous être acquis en double, uniquement pour les besoins de l'analyse.

7.2) ENTREPOTS DE DONNEES :

Le concept d'entrepôt de données « *Data Warehouse* » a été formalisé pour la première fois en 1990.

Après plusieurs années d'utilisation des infocentres, et après avoir constaté que les informations traitées dans les applications opérationnelles étaient très différentes de celles interrogées dans les applications ; et que les questions posées par un décideur impliquent fréquemment des informations stockées dans plusieurs applications ou base de données, par exemple lorsque vous calculez la rentabilité de vos clients, vous exportez des données de la gestion commerciale (factures, commandes), de la comptabilité (délais de règlement, impayés), mais également de la gestion de production (coût des produits fabriqués). Le fait de dupliquer ces bases de données dans un infocentre ne simplifie en rien ces extractions. Les services informatiques ont donc imaginés une évolution intelligente de ce mode de stockage, ils ont mis en place, en sortie des bases de production, **un entrepôt de données**. Cet entrepôt uniquement **dédié au stockage des données décisionnelles**, permet de réconcilier les différentes sources initiales de données, et les applications de production.

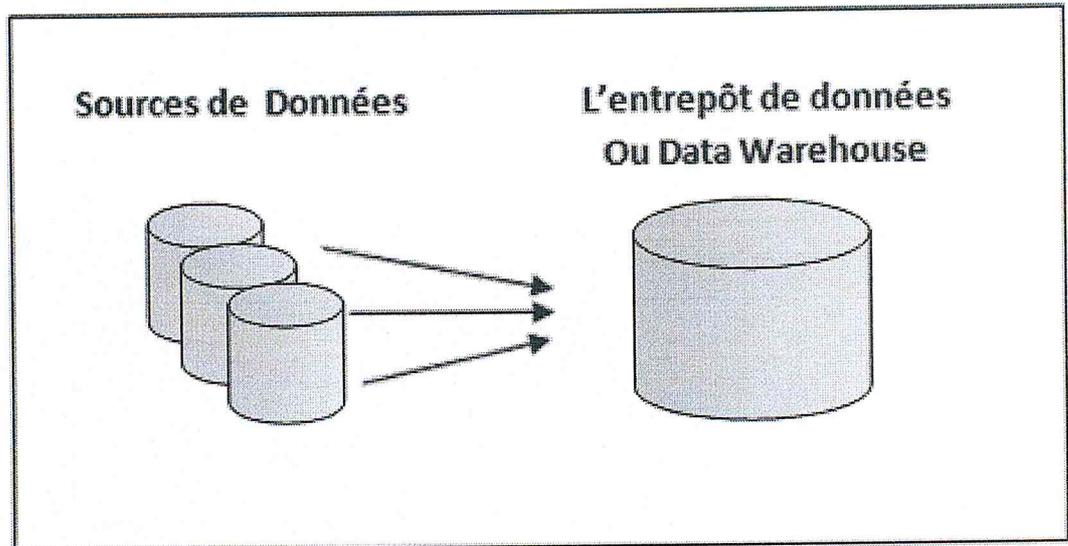


Figure 3 : Entrepôt de données.

Fréquemment construit à partir d'une base de données relationnelle, cet entrepôt de données sert littéralement d'entrepôt. On y verse une copie des données qui serviront, un jour, à l'analyse et à la prise de décision. Cela évite également de conserver un historique trop important dans les bases de production, souvent concernées par le court terme, alors que les analystes ont besoin de recul pour détecter des tendances.

Le tableau suivant donne une comparaison entre l'Infocentre et l'entrepôt de données.

Infocentre	Entrepôt de données
Collection de données	Collection de données
Orientées sujet	Orientées sujet
Intégrée	Intégrée
Volatiles	Non volatiles
Actuelles	Historiées
Organisées pour le support d'un processus de décision ponctuelle	Organisées pour le support d'un processus de d'aide à la décision
Outils	Architecture

Tableau 2 : Comparaison entre Infocentre et entrepôt de données

7.3) Les bases de données multidimensionnelles :

Quand un manager parle de ses résultats, il leur associe toujours une variable. Cette dernière est la dimension. Par exemple, on cherche le chiffre d'affaire par trimestre, la rentabilité par client, la marge par commercial et par région.

Les bases de données relationnelles sont constituées d'un ensemble de tables à deux dimensions. Lorsqu'on lance des **requêtes décisionnelles**, comme par exemple :

« Le calcul de la marge par client, et par gamme de produits, mois par mois depuis un an », le système a besoin d'accéder à de nombreuses tables. Une telle requête peut être très longue à exécuter, voir même parfois bloquer totalement le serveur.

Pour résoudre ce problème, les bases de **données multidimensionnelles** ont été inventées par le Docteur **Edgard Codd**, déjà considéré comme le Père des bases de données relationnelles. Ces bases se présentent sous forme d'un **cube**.

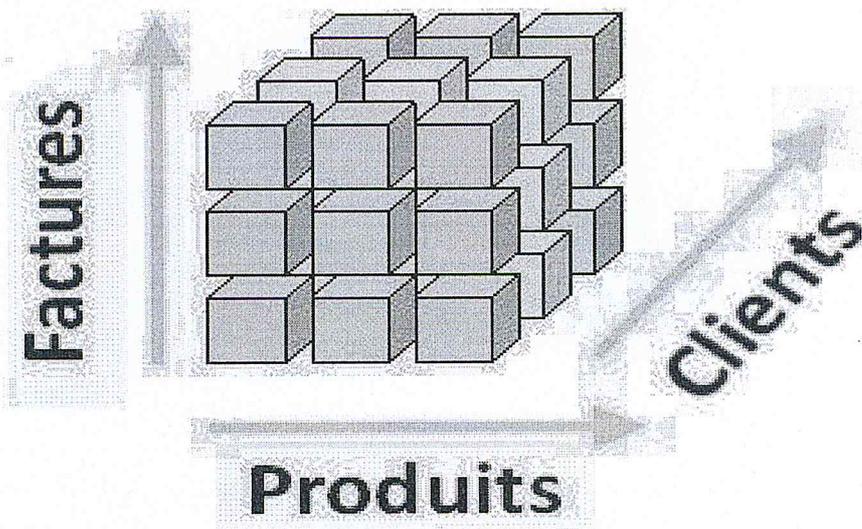


Figure 4 : Base Multidimensionnelle.

Chaque base de données **multidimensionnelle** pré-calculé et stocke toutes les informations, au croisement de chacune des dimensions. Le chiffre d'affaire réalisé par chaque client, sur chaque produit, dans chaque magasin, durant chaque mois, sera

pré-calculé, et conservé dans le cube, au croisement de chacune des dimensions prévues.

Ainsi, lors du lancement d'une requête décisionnelle complexe, le système aura simplement à lire les croisements de chaque colonne du cube, pour extraire des informations qui auront déjà été calculées.

Conclusion :

Le développement des premiers systèmes d'informations s'est concentré sur l'automatisation des processus opérationnels et le bon déroulement de l'activité principale de l'entreprise et l'excellence opérationnelle.

Les besoins d'analyse sont arrivés bien plus tard, ils relèvent plus de l'avantage concurrentiel et le pilotage de l'entreprise que de l'excellence opérationnelle.

Le système décisionnel cherche donc à donner un aperçu global de l'entreprise via des outils d'analyse pour aider les décideurs à prendre des décisions, et cela en organisant l'ensemble de données consolidé à partir des différentes sources dans une base de données unique 'Date Warehouse' qu'on va présenter dans le chapitre suivant.

CHAPITRE 2 : DATA WAREHOUSE

Introduction :

L'information est devenue vitale pour l'entreprise. Toutes les données, qui proviennent du système de production de l'entreprise, des sources externes, vont devoir être organisées, coordonnées, intégrées et enfin stockées pour donner à l'utilisateur une vue intégrée et orientée métier dans un entrepôt de données ou bien «Data Warehouse».

1) Définition :

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

Un entrepôt de données « *Data Warehouse* » est défini de plusieurs façons comme suit :

- Est une collection de données orientées sujet, intégrées, non volatiles et *historisées*, organisées pour le processus de décisions [Inmon, 2002].
- Désigne à la fois la base dans laquelle sont stockées l'ensemble des informations, mais également l'ensemble du système d'information décisionnel [Ghomari, 2000].
- Fait référence à une collection de technologies d'aide à la décision permettant à des (managers, dirigeants et analystes) de prendre des décisions pertinentes et rapides [Chaudhuri, 1997].
- Organise et conserve les données nécessaires aux processus informationnels et analytiques dans une perspective de long terme.

Ce système correspond à une collection de données (agrégées ou atomiques) pour aider les gestionnaires dans leur prise de décision. Les données sont :

❖ Orientées sujets :

L'entrepôt de données est organisé autour des sujets majeurs de l'entreprise, contrairement aux données des systèmes opérationnels qui en général sont organisés

par processus fonctionnels. Les données sont structurées par thèmes. L'intérêt de cette organisation est de disposer de l'ensemble des informations utiles sur un sujet le plus souvent transversal aux structures fonctionnelles et organisationnelles de l'entreprise.

❖ **Données intégrées :**

Les données alimentant l'entrepôt de données proviennent de multiples applications sources hétérogènes. Les données des systèmes de production doivent être converties, reformatées et nettoyées, de façon à avoir une seule vision globale dans l'entrepôt de données.

Intégrer les données consiste à maîtriser leur hétérogénéité, pour donner au contenu de l'entrepôt de données une présentation homogène et pour garantir sa qualité. Cette phase d'intégration est très complexe et représente 60 à 90 % de la charge totale d'un projet.

❖ **Historiées :**

Dans un entrepôt de données, la donnée ne doit jamais être mise à jour en mode « annule et remplace ». Un référentiel temps doit être associé à la donnée afin d'être capable d'identifier une valeur particulière dans le temps.

❖ **Non volatiles :**

La non volatilité des données est en quelque sorte une conséquence de l'historisation. Une même requête effectuée à quelque mois d'intervalle en précisant la date de référence de l'information recherchée donnera le même résultat.

❖ **Résumées :**

Les informations issues des sources de données doivent être agrégées et réorganisées afin de faciliter le processus de prise de décision.

❖ **Disponible pour l'interrogation et l'analyse :**

Les utilisateurs doivent pouvoir consulter les données réorganisées de l'entrepôt en fonction de leur droit d'accès au travers d'outils interactifs d'aide à la manipulation et l'analyse.

2) Composants de base d'un Data Warehouse :

Dans ce paragraphe, on présente les principaux composants de base d'un Data Warehouse, selon «Ralph KIMBALL».

• **Systèmes Sources :**

C'est le système opérationnel d'enregistrement, dont la fonction consiste à capturer les transactions liées à l'activité de l'entreprise [Kim, 1997]. Il s'agit souvent de ce que l'on appelle les applications de gestion.

On appelle, d'une façon générale base de production toutes les sources (qu'il s'agisse de données de production, d'informations internes et d'informations externes quel que soit leur mode de stockage) dont il va falloir extraire des données en vue d'alimenter le Data Warehouse.

• **La Zone de Préparation des Données :**

Elle regroupe l'ensemble des processus qui nettoient, transforment, combinent, archivent, suppriment les doublons. Elle prépare les données sources en vue de leur intégration et de leur exploitation dans le Data Warehouse.

• **Data Warehouse (Base de l'Entrepôt de données) :**

Est le lieu de stockage centralisé des informations utiles pour les décideurs, dans lequel les données hétérogènes des sources sont intégrées et stockées durablement, dans un serveur de présentation.

Celui-ci correspond à la machine cible sur laquelle l'entrepôt de données est stocké et organisé pour répondre en accès direct aux requêtes provenant des utilisateurs, des générateurs d'états ou d'autres applications.

Sur le serveur de présentation, les données sont stockées sous forme dimensionnelle afin de faciliter l'accès aux utilisateurs finaux.

- **Data Marts (Magasins de Données) :**

Un data mart est un magasin de données. C'est un sous ensemble du Data Warehouse qui ne contient que les données d'un métier de l'entreprise alors que le Data Warehouse contient toutes les données décisionnelles de l'entreprise pour tous les métiers.

- **Le Portail de Restitution:**

C'est la partie publique du Data Warehouse. Il représente ce que voient les utilisateurs, les outils avec lesquels ils travaillent. Les services offerts par le portail de restitution sont les services d'accès aux données, les applications de modélisations et le «data Mining».

Les services d'accès aux données comprennent : la navigation dans le Data Warehouse, la gestion des requêtes et la généralisation d'états standards. Les applications de modélisation offrent différent type d'analyses basées sur des modèles financiers, système d'évaluation de clientèles, et offrent aussi une analyse sur l'activité centrale du «data Mining», telles que la catégorisation, la classification et le regroupement par affinité.

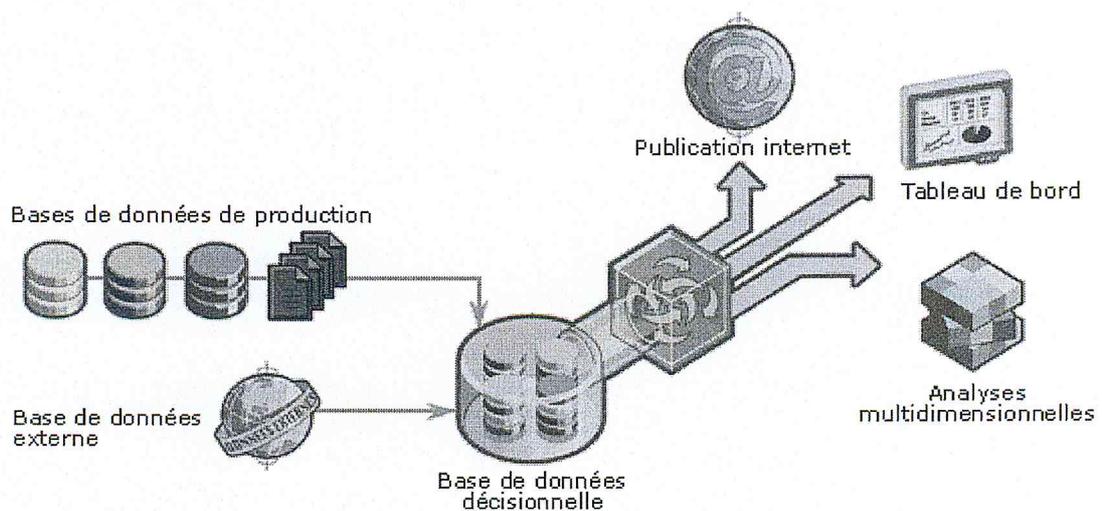


Figure 5 : Composant de base d'un Data Warehouse [BOUSSEBAT, 2003].

3) Objectifs d'un Entrepôt de données :

Voici les objectifs de l'entrepôt de données tels que définis par Ralph Kimball, dans son livre « *Entrepôt de données, Guide pratique du concepteur de Data Warehouse* » [Kimball, 2001].

- Accès aux informations de l'entreprise.
- Les informations d'un entrepôt de données sont cohérentes.
- Les outils de présentation d'informations font partie de l'entrepôt de données.
- Les données publiées sont stockées dans l'entrepôt de données.
- La qualité de l'information d'un entrepôt de données est très importante.

En effet, le principal objectif d'un système d'aide à la décision est précisément de donner à l'utilisateur les moyens d'être autonome, en lui permettant de naviguer dans l'information en fonction des questions qu'il se pose.

4) L'alimentation d'un Data Warehouse:

Une fois le Data Warehouse conçu, pour l'alimenter il faut avoir préalablement identifié les données intéressantes de celles qui ne le sont pas des bases de production. Ces données ainsi réorganisées deviennent des informations utiles pour le décisionnel.

La première phase de la construction d'un Data Warehouse consiste généralement à extraire les données utiles des systèmes opérationnels qui dans de nombreux cas sont hétérogènes, diffusées et complexes.

- Elles sont hétérogènes car bien souvent on rencontre plusieurs SGBD différents.
- Elles sont diffusées car on les trouvera au sein de plusieurs environnements matériels, éventuellement reliés par plusieurs réseaux interconnectés différents.
- Elles sont complexes, car on rencontrera plusieurs modèles logiques et physiques prioritairement orientés vers les traitements complexes.

Cette alimentation (le plus souvent appelée processus ETL «Extract-Transform-Load») se déroule en 3phases qui sont :

- Extraction des données primaires (issues par exemple des systèmes de production)
- Transformation des données

- Le chargement des données traitées dans l'entrepôt de données

Ces trois étapes décrivent une mécanique cyclique qui a pour but de garantir l'alimentation du Data Warehouse en données homogènes, propre et fiables.

4.1) Les phases de l'alimentation « E.T.L » :

Les phases du processus E.T.L représentent la mécanique d'alimentation du Data Warehouse. Ainsi elles se déroulent comme suit :

a) L'extraction des données :

« L'extraction est la première étape du processus d'apport de données à l'entrepôt de données. Extraire, cela veut dire lire et interpréter les données sources et les copier dans la zone de préparation en vue de manipulations ultérieures.» [kimball, 2005].

Elle consiste en :

- Cibler les données
- Appliquer les filtres nécessaires
- Définir la fréquence de chargement

Lors du chargement des données, il faut extraire les nouvelles données ainsi que les changements intervenus sur ces données.

b) La transformation des données :

La transformation est la seconde phase du processus. Cette étape, qui du reste est très importante, assure en réalité plusieurs tâches qui garantissent la fiabilité des données et leurs qualités. Ces tâches sont :

- Consolidation des données
- Correction des données et élimination de toute ambiguïté
- Élimination des données redondantes
- Compléter et renseigner les valeurs manquantes

c) Le chargement des données :

C'est la dernière phase de l'alimentation d'un entrepôt de données, le chargement est une étape indispensable. Elle reste toute fois très délicate et exige une certaine connaissance des structure du système de gestion de la base de données (tables et index) afin d'optimiser au mieux le processus

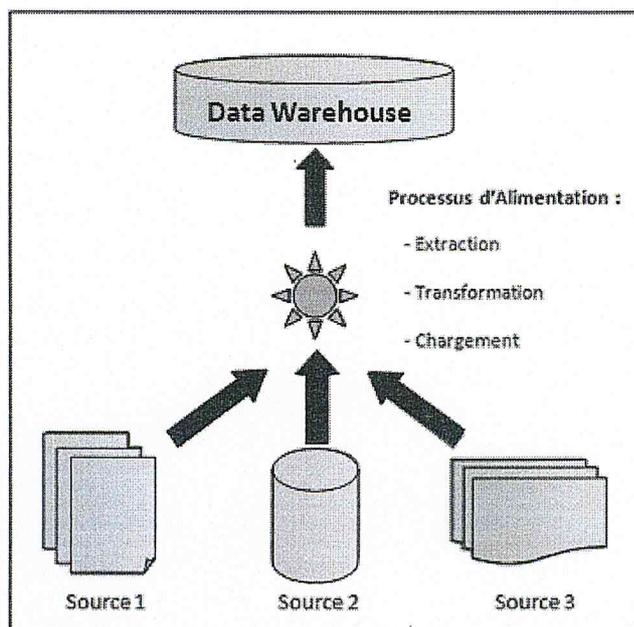


Figure 6 : Processus d'Alimentation.

4.2) Politiques de l'alimentation :

Le processus de l'alimentation peut se faire de différentes manières. Le choix de la politique de chargement dépend des sources : disponibilité et accessibilité.

Ces politiques sont :

- **Push** : dans cette méthode, la logique de chargement est dans le système de production il " pousse " les données vers la zone de préparation quand il en a l'occasion. L'inconvénient est que si le système est occupé, il ne poussera jamais les données.
- **Pull** : contrairement a la méthode précédente, le Pull " tire " les données de la source vers la zone de préparation. L'inconvénient de cette méthode est qu'elle peut surcharger le système s'il est en cours d'utilisation.

- **Push-pull** : c'est la combinaison des deux méthodes. La source prépare les données à envoyer et indique à la zone de préparation qu'elle est prête. La zone de préparation va alors récupérer les données.

4.3) Les outils E.T.L :

Les outils ETL, en français ETC « extraction-transformation-chargement » [kimball, 2005], sont des outils qui garantissent la fiabilité et facilitent le déroulement des trois phases citées précédemment. D'où leur importance dans un projet Data Warehouse.

Le processus E T L doit être sur, correctif, rapide et transparent.

5) Exploitation des données du Data Warehouse :

Une fois que les données se trouvent dans le Data Warehouse, il ne reste plus qu'à les exploiter.

L'utilisateur final doit alors pouvoir interroger les données en ligne, à l'aide d'outils simples et conviviaux, qui leurs permettent de répondre à leurs nouveaux besoins.

Parmi les modes d'exploitation des données du Data Warehouse on a :

5.1) Le tableau de bord :

5.1.1) Définition :

Le tableau de bord est un ensemble d'indicateurs peut nombreux conçus pour permettre aux gestionnaires de prendre connaissance de l'état et de l'évolution des systèmes qu'ils pilotent et d'identifier les tendances qui les influenceront sur un horizon cohérent avec la nature de leurs fonctions, [BOU, 2003].

C'est un outil de **pilotage** qui souligne l'état d'avancement dans lequel se trouve le processus afin de permettre au responsable de mettre en place des actions correctives (*décisions*).

C'est un outil de **contrôle** du système à piloter : il représente l'évolution des activités et mesure l'atteinte des objectifs tout en reflétant des conséquences des décisions prises. C'est un tableau qui présente des caractéristiques d'affichage, des présentations visuelles, synthétiques et d'espace limité.

5.1.2) Rôle d'un tableau de bord :

On peut résumer les rôles d'un tableau de bord dans les points suivants :

- Mesure de la performance
- Outil de diagnostic
- Sources d'informations
- Réducteur d'incertitude
- Stabilisateur d'information
- Facilitateur de communication
- Stimulant pour la réflexion

5.1.3) Exigences d'un tableau de bord :

Afin de bien accomplir son rôle et sa mission le tableau de bord doit répondre à des exigences précises comme :

Etre clair, Etre synthétique, Etre tenu à jour, Donner des informations correctes, Présenter une comparaison chronologique, Présenter une comparaison avec les concurrents, Présenter une comparaison avec une norme, Présenter une comparaison avec le budget.

5.1.4) Type des tableaux de bord :

• Tableau de bord d'activité :

Il s'agit d'un tableau de bord classique, qui présente des indicateurs pour suivre des activités. Il est utile quand le suivi des différentes activités prime sur le suivi d'actions mises en œuvre pour atteindre un objectif.

Le tableau de bord d'activité est en fait un tableau de bord de suivi des activités.

- **Tableau de bord de pilotage :**

Il s'agit d'un tableau de bord pour atteindre un ou plusieurs objectifs avec des plans d'action préalablement défini.

L'utilité d'un tableau de bord de pilotage varie selon le positionnement de son utilisateur au sein de la chaîne hiérarchique.

- **Tableau de bord de projet :**

Il s'agit d'un tableau de bord pour atteindre un objectif généralement matérialisé par un produit physique.

La réalisation de ce type de tableau de bord demande de connaître les phases du projet, les activités qui le composent, puis les tâches par activité.

[Ben mehidi et Mbilinyi, 2009]

Conclusion :

Le Data Warehouse est le cœur du système d'information décisionnel. C'est une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision.

Dans le chapitre suivant nous allons introduire la notion du NoSQL, que nous avons choisit pour la construction de notre Data Warehouse, ainsi que MapReduce que nous utiliserons pour l'analyse des données.

CHAPITRE 3 : NOSQL ET MAPREDUCE

Introduction :

Avec l'essor des grandes plateformes web, le volume de données à gérer par les applications a explosé, et les systèmes de gestion de données traditionnels, relationnels et transactionnels, basés sur le langage SQL, ont montré leurs limites.

C'est dans ce genre de problématiques que les SGBD classiques ne sont plus les outils les plus adaptés. Poussés dans des situations extrêmes, il fallait une nouvelle façon de faire et c'est ainsi qu'en 2009 "NoSQL" est né, qui aurait pu être nommée "NoRel" comme le suggère-lui même l'inventeur du terme NoSQL, Carl Strozzi.

NoSQL signifie "Not Only SQL", littéralement "pas seulement SQL". Ce terme désigne l'ensemble des bases de données qui s'opposent à la notion relationnelle des SGBDR. La définition, "pas seulement SQL", apporte un début de réponse à la question "Est ce que le NoSQL va tuer les bases relationnelles?". En effet, NoSQL ne vient pas remplacer les BD relationnelles mais pour en combler les faiblesses ou compléter les fonctionnalités des SGBDR pour donner des solutions plus intéressantes dans certains contextes. [Aurelien FOUCRET, 2011]

1) Pourquoi le NoSQL ?

Les axes principaux du NoSQL sont: la haute disponibilité et le partitionnement des données, au détriment de la consistance.

Le premier besoin fondamental auquel répond NoSQL est **avoir un schéma de données dynamique** et de la **performance**.

En effet, ces dernières années, les géants du Web comme Google et Amazon ont vu leurs besoins en termes de charge et de volumétrie de données croître de façon exponentielle, Et c'est pour répondre à ces besoins que ses solutions ont vu le jour. Les architectes de ces organisations ont procédé à des compromis sur le caractère ACID (*atomicité, consistance, isolation, durabilité*) des SGBDR.

Ces intelligents compromis sur la notion de relationnel ont permis de dégager les SGBDR de leurs freins à la scalabilité horizontale et à l'évolutivité. Par la suite des entreprises comme Facebook, Twitter ou encore LinkedIn ont fait migré leurs données sur des bases NoSQL. Après avoir constaté que les SGBD relationnels ne sont pas adaptés aux environnements distribués requis par les volumes gigantesques de données et par les trafics importants générés par ces opérateurs.

Cette adoption croissante des bases NoSQL conduit à une multiplication et une amélioration des offres Open Source des moteurs.

Les besoins majeurs identifiés par les fournisseurs de services en ligne et les acteurs du web social sont les suivants :

- Capacité à distribuer les traitements sur un nombre de machines important afin d'être en mesure d'absorber des charges très importantes.
- Capacité à répartir les données entre un nombre important de machines afin d'être en mesure de stocker de très grands volumes de données.
- La distribution de données sur plusieurs data centres (nœud) afin d'assurer une continuité de service en cas d'indisponibilité de service sur un data centre.
- Une architecture qui fonctionne sur du matériel peu spécialisé et donc facilement remplaçable en cas de panne. [Aurelien FOUCRET, 2011]

2) Solutions NoSQL :

On regroupe derrière le NoSQL l'ensemble des technologies de persistance qui se distinguent par une absence de requête et par un relâchement des caractéristiques ACID (Atomicité, Consistance, Isolation, Durabilité), propres aux RDBMS (Relational DataBase Management System SGBD) [Mic, 2009].

La famille des SGBD NoSQL se compose de plusieurs catégories :

❖ Oriente Clé / valeur :

Ce modèle peut être assimilé à une hashmap distribuée. Les données sont, donc, simplement représentées par un couple clé/valeur. La valeur peut être une simple chaîne de caractères, un objet sérialisé...

Cette absence de structure ou de typage ont un impact important sur le requêtage. En effet, toute l'intelligence portée auparavant par les requêtes SQL devra être portée par l'applicatif qui interroge la Base de donnée. Néanmoins, la communication avec la Base de donnée se résumera aux opérations PUT, GET et DELETE.

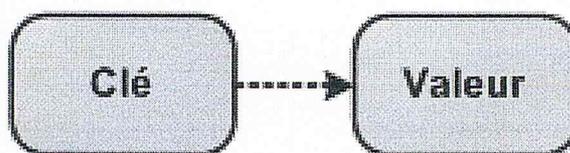


Figure 7 : Base de Données Clé-Valeur [Michaël Figuière, 2010].

❖ Orienté colonne :

Ce modèle ressemble à première vue à une table dans un SGBDR à la différence qu'avec une BD NoSQL orientée colonne, le nombre de colonnes est dynamique. En effet, dans une table relationnelle, le nombre de colonnes est fixé dès la création du schéma de la table et ce nombre reste le même pour tous les enregistrements dans cette table. Par contre, avec ce modèle, le nombre de colonnes peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes ayant des valeurs NULL.

Comme solution, on retrouve principalement HBase (implémentation Open Source du modèle BigTable publié par Google) ainsi que Cassandra (projet Apache qui respecte l'architecture distribuée de Dynamo d'Amazon et le modèle BigTable de Google).

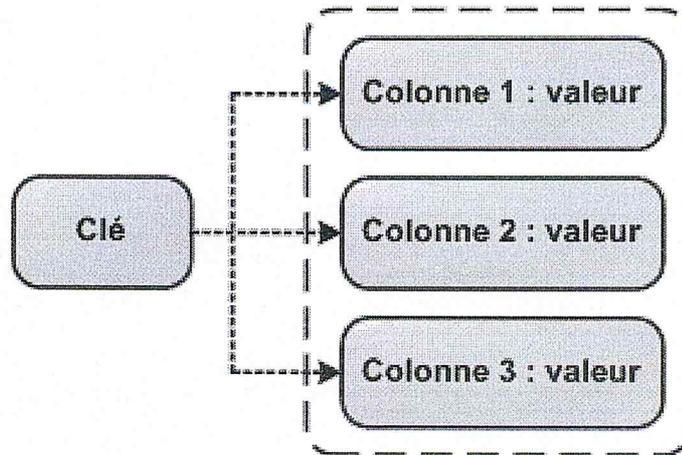


Figure 8 : Base de Données Orientée colonnes [Michaël Figuière, 2010].

❖ **Orienté document :**

Ce modèle se base sur le paradigme clé valeur. La valeur, dans ce cas, est un document de type JSON ou XML. L'avantage est de pouvoir récupérer, via une seule clé, un ensemble d'informations structurées de manière hiérarchique. La même opération dans le monde relationnel impliquerait plusieurs jointures.

Pour ce modèle, les implémentations les plus populaires sont CouchDB d'Apache, RavenDB et MongoDB.

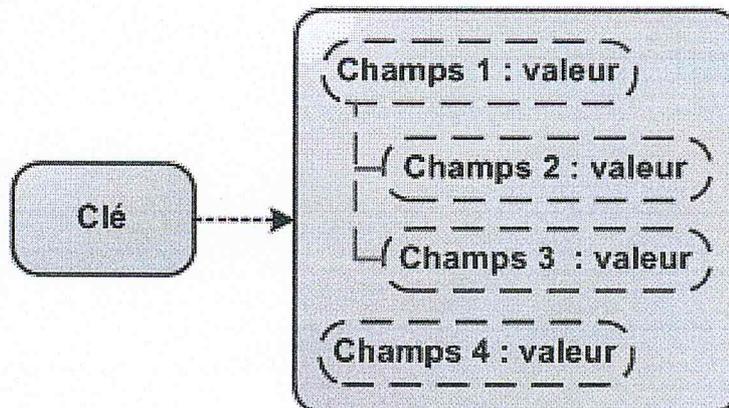


Figure 9 : Base de Données Orientée document [Michaël Figuière, 2010].

❖ **Orienté graphe :**

Ce modèle de représentation des données se base sur la théorie des graphes. Il s'appuie sur la notion de nœuds, de relations et de propriétés qui leur sont rattachées. Ce modèle facilite la représentation du monde réel, ce qui le rend adapté au traitement des données des réseaux sociaux. La principale solution est Neo4J.

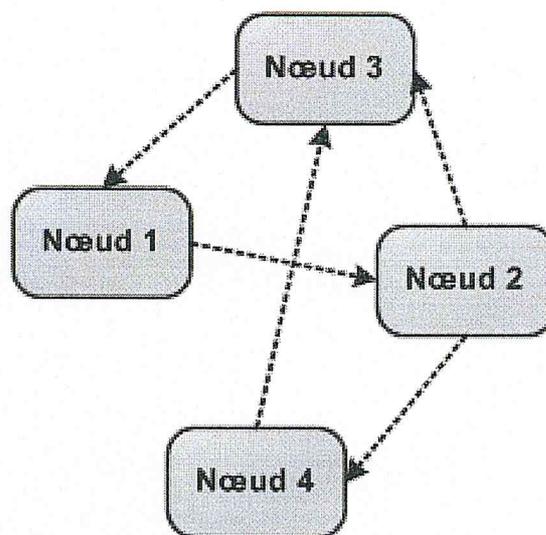


Figure 10 : Base de Donnée Orientée graphe [Michaël Figuière, 2010].

Les deux mouvements “orienté colonne” et “orienté document” découlent bien du système clé valeur et c’est la nature ou la structure de la valeur qui diffère.

3) Les limites des systèmes de SGBD relationnels face aux usages :

Le stockage distribué n’est pas la seule contrainte qui pèse à ce jour sur les systèmes relationnels. Au fur et à mesure du temps, les structures de données manipulées par les systèmes sont devenues de plus en plus complexes avec en contrepartie des moteurs de stockage évoluant peu.

Le principal point faible des modèles relationnels est l’absence de gestion d’objets hétérogènes ainsi que le besoin de déclarer au préalable l’ensemble des champs représentant un objet (absence de l’aspect dynamique).

Pour répondre au besoin des objets hétérogènes non pré déclarés, on a vu apparaître des modélisations complexes sur le plan algorithmique comme le modèle EntityAttributeValue permettant de séparer un objet et ses champs. Ces modèles, quoique répondant au besoin de flexibilité des applications modernes, présentent un inconvénient majeur qui est leur très faible niveau de performance. La majeure partie des outils développés dans le cadre de la mouvance NoSQL **permettent l'utilisation d'objets hétérogènes** apportant comparativement une bien plus grande flexibilité dans les modèles de données ainsi qu'une simplification de la modélisation. [Aurelien FOUCRET, 2011]

Il ya aussi le fait qu'une base de données relationnelle est construite sur les propriétés ACID, ses propriétés bien que nécessaires à la logique du relationnel nuisent fortement aux performances surtout la propriété de cohérence. En effet, la cohérence est très difficile à mettre en place dans le cadre de plusieurs serveurs, car pour que celle-ci soit respectée tous les serveurs doivent être des miroirs les uns des autres, de ce fait deux problèmes apparaissent:

- Le coût en stockage est énorme car chaque donnée est présente sur chaque serveur.
- Le coût d'insertion/modification/suppression est très grand, car on ne peut valider une transaction que si on est certain qu'elle a été effectuée sur tous les serveurs et le système fait patienter l'utilisateur durant ce temps.

[Mathieu ROGER, 2012]

Le tableau suivant illustre une comparaison entre les SGBD relationnel et NoSQL:

SGBD relationnel	NoSQL : non relationnel
<p>Transaction ACID :</p> <ul style="list-style-type: none"> • Atomicité : une transaction doit soit être complètement validée ou complètement annulée. • Cohérence : une transaction ne peut pas laisser la base de données dans un état incohérent. • Isolation : une transaction ne peut voir aucune autre transaction en cours d'exécution. • Durabilité : après que le client ait été informé du succès de la transaction, les résultats de celle-ci ne disparaîtront pas. 	<p>Théorème de CAP:</p> <ul style="list-style-type: none"> • Cohérence : celui-ci est différent de celui des contraintes ACID. Le but est d'avoir la même information entre 2 intervalles de lecture très rapprochés. • Disponibilité : il faut que la lecture des données soit toujours possible. • Panne tolérance : bien gérer les serveurs pour que s'il y en a qui tombe l'information soit toujours récupérable quelque part sur un autre serveur.
<p>mécanisme assurant la cohérence coûteux, protection contre la corruption, index, trigger rendent la scalabilité difficile.</p>	<p>limitation ou inexistence de ces mécanismes (index, trigger...) car le but n'est pas d'être focalisé sur la structure, sur la cohérence des données car c'est une perte de temps et de disponibilité.</p>
<p>notion de table, de clef primaire.</p>	<p>le type de stockage est sous forme (clé, valeur). Chaque clef est unique dans la base de données. Les données ne sont structurées et la cohérence n'est pas vérifiée.</p>
<p>pour interroger la base de données utilisation du langage SQL.</p>	<p>plus de langage SQL. On interroge la base de données grâce à une API (en Java script par exemple) offrant 3 opérations de base : get, set, delete. Pour des requêtes un peu plus complexes on passera par des fonctions MapReduce.</p>

les données sont toujours cohérentes.	il y a plusieurs niveaux de cohérences proposés (Exemple: si tous les commentaires d'un article ne sont pas présents sur la page d'une lecture à l'autre ce n'est pas très grave pour l'utilisateur).
---------------------------------------	---

Tableau 3 : Tableau comparatif entre les SGBD relationnel et NoSQL.

4) Caractéristiques SGBD NoSQL:

Alors que les bases de données relationnelles sont basées sur le concept ACID (Atomicité, Consistance, Isolation, Durabilité), le NoSQL prône l'abandon de ces contraintes. Ne plus répondre au modèle relationnel est la caractéristique principale des bases de données NoSQL.

Les bases de données NoSQL répondent au théorème du CAP d'Eric Brewer qui est plus adapté aux systèmes distribués.

Celui-ci énonce trois grandes propriétés pour les systèmes distribués :

- ✓ C comme Cohérence : tous les nœuds du système voient exactement les mêmes données au même moment.
- ✓ A comme Availability ou Disponibilité : la perte de nœuds n'empêche pas les survivants de continuer à fonctionner correctement.
- ✓ P comme Partition tolérance ou Résistance au partitionnement : aucune panne moins importante qu'une coupure totale du réseau ne doit empêcher le système de répondre correctement (ou encore : en cas de morcellement en sous réseaux, chacun doit pouvoir fonctionner de manière autonome), donc le système est tolérant au partitionnement.

Ce théorème nous apprend aussi qu'il est impossible de respecter ces trois propriétés simultanément, nous ne pouvons en respecter seulement deux à la fois

- Les bases de données relationnelles implémentent les propriétés de Cohérence et de Disponibilité (système AC). Les bases de données

NoSQL sont généralement des systèmes CP (Cohérent et Résistant au partitionnement) ou AP (Disponible et Résistant au partitionnement).

[Mathieu ROGER, 2012]

On interroge les bases de données NoSQL grâce à 3 opérations de base : get, set et delete, pour des requêtes un peu plus complexes on passera par des fonctions MapReduce, ce dernier sera présenté dans ce qui suit.

5) MapReduce :

MapReduce est un Framework de calcul distribué sur de grosses quantités de données, introduit par Google, dans lequel sont effectués des calculs parallèles, de données potentiellement très volumineuses. [Malo Jaffre et Pablo Rauzy, 2010].

MapReduce est une technique de programmation distribuée très utilisée dans le milieu NoSQL et qui vise à produire des requêtes distribuées. Cette technique se décompose en deux grandes étapes :

5.1) L'étape Map :

Dans cette étape le nœud à qui est soumis un problème, le découpe en sous-problèmes, et les délègue à d'autres nœuds (qui peuvent en faire de même récursivement). Les sous-problèmes sont ensuite traités par les différents nœuds à l'aide de la fonction *Map* qui à un couple (clé, valeur) associe un ensemble de nouveaux couples (clé, valeur).

Exemples de fonction Map :

- ✓ A un couple (UserId, User), on assigne le couple (Role, User). A l'issue de l'étape de mapping, on obtient une liste contenant les utilisateurs groupés par rôle.
- ✓ A un couple (UserId, User) on assigne le couple (UserId, User) uniquement si l'email de l'utilisateur se termine par “.fr”.

5.2) L'étape Reduce :

Dans cette étape les nœuds les plus bas font remonter leurs résultats au nœud parent qui les avait sollicités. Celui-ci calcule un résultat partiel à l'aide de la

fonction *Reduce* (réduction) qui associe toutes les valeurs correspondant à la même clé à une unique paire (clé, valeur). Puis il remonte l'information à son tour.

La fonction *Reduce* est appelée sur le résultat de l'étape de mapping et permet d'appliquer une opération sur la liste.

Exemples de fonction *Reduce* :

- ✓ Moyenne des valeurs contenues dans la liste
- ✓ Comptabilisation des différentes clés de la liste
- ✓ Comptabilisation du nombre d'entrées par clé dans la liste

À la fin du processus, le nœud d'origine peut recomposer une réponse au problème qui lui avait été soumis.

Remarque :

L'étape de mapping peut être parallélisée en traitant l'application sur différents nœuds du système pour chaque couple (clé, valeur).

L'étape de réduction n'est pas parallélisée et ne peut être exécutée avant la fin de l'étape de mapping.

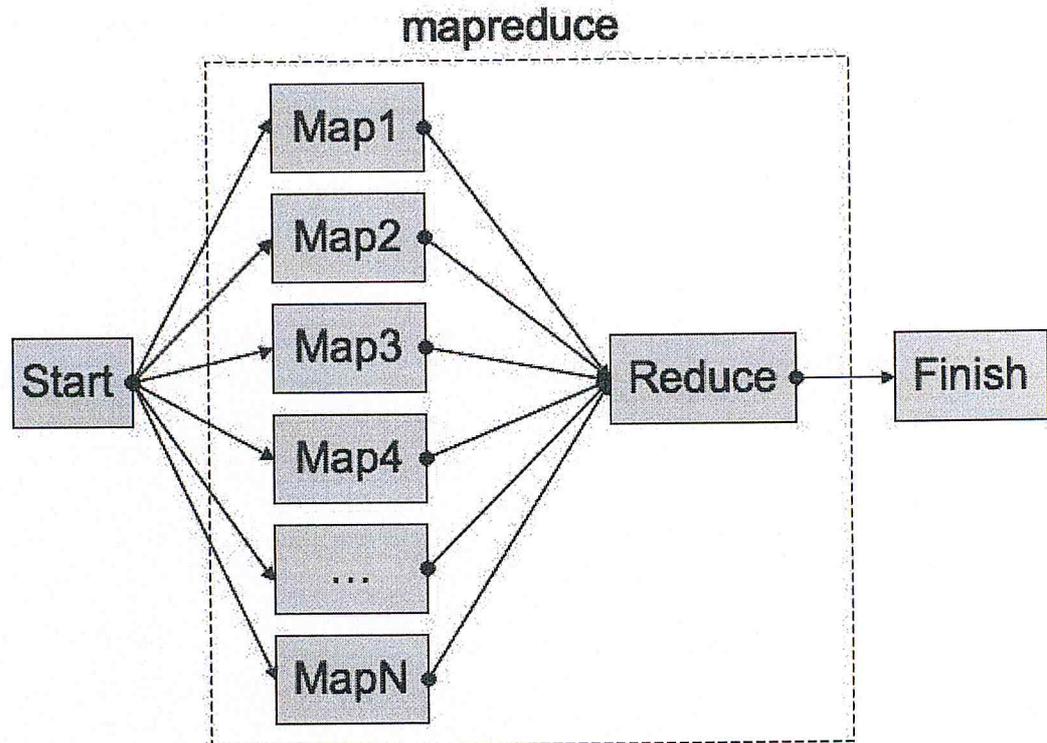


Figure 11 : MapReduce [Web1].

Les bases de données NoSQL proposent diverses implémentations de la technique MapReduce permettant le plus souvent de développer les méthodes Map et Reduce en Javascript ou en Java.

5.3) Exemple :

Pour illustrer l'algorithme MapReduce, considérons un jeu de données constitué des 3 phrases suivantes :

- savoir être et
- savoir faire
- sans faire savoir

Le but de l'illustration est d'appliquer le modèle *MapReduce* afin de sortir le nombre d'occurrences des mots constituant le texte. L'ensemble du processus est schématisé ci-dessous :

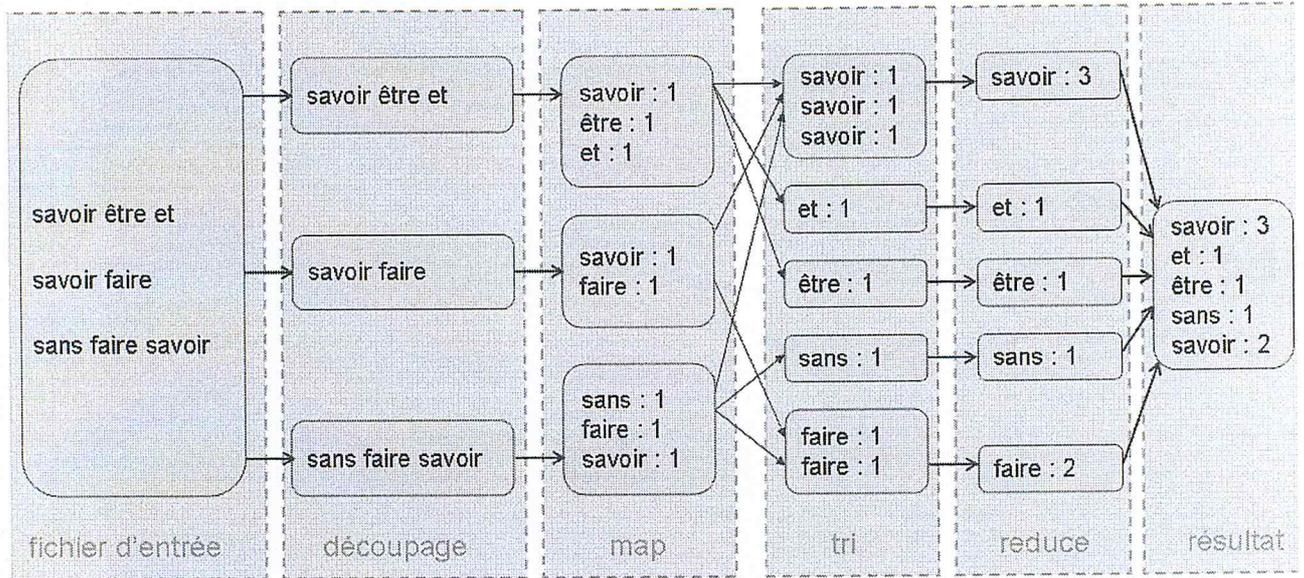


Figure 12 : Exemple nombre d'occurrences d'un mot avec MapReduce.

5.4) Avantages et Inconvénients : [Khaled TANNIR, 2011]

- **Avantages de MapReduce :**

- ✓ Traite de grands volumes de données.
- ✓ Gère des milliers de processeurs.
- ✓ Parallélise et distribue les traitements.
- ✓ Ordonnance des entrées / sorties.
- ✓ Gère la tolérance aux pannes.
- ✓ Surveille les processus.

- **Inconvénients de MapReduce :**

- ✓ Une seule entrée pour les données.
- ✓ Le flux de données en deux étapes le rend très rigide.
- ✓ Le système de fichiers distribués (HDFS) possède une bande passante limitée en entrée / sortie.

- ✓ Les opérations de tris limitent les performances du Framework.

Conclusion :

La technologie NoSQL ne cesse de faire parler d'elle. Le NoSQL s'annonce donc comme une réponse aux limites du relationnel et du monde ACID (Atomicité, Consistance, Isolation, Durabilité), et les géants du web l'ont bien compris, tous abandonnent petit à petit leur BD relationnelle au profit des principes du NoSQL: Google et BigTable, Amazon et SimpleDB, Facebook et Cassandra, LinkedIn et Voldemort...Etc.

Comme la quantité de données à manipuler est de plus en plus grande surtout dans le domaine du web, il fallait donc distribuer ces données pour diminuer la charge sur les nœuds, et aussi distribuer les calculs c'est là que MapReduce intervient.

Cependant, n'oublions pas que, bien que robuste, ces technologies restent jeunes et doivent par conséquent encore évoluer.

Pour ce qui suit nous allons aborder les différentes démarches de la construction du Data Warehouse en allant de sa conception dimensionnelle à son alimentation jusqu'à son exploitation.

PARTIE 2 : CONCEPTION DE LA SOLUTION

**CHAPITRE 4: ETUDE DE L'EXISTANT ET IDENTIFICATION DES
BESOINS**

CHAPITRE 5 : CONCEPTION DES ELEMENTS DU SYSTEME

PARTIE 2 : CONCEPTION DE LA SOLUTION

Dans cette partie du document nous allons aborder la conception de notre projet système décisionnel, cette conception passe par la phase étude de l'existant et l'identification des besoins dans le chapitre 4, la conception et l'alimentation du Data Warehouse NoSQL et le Data Warehouse classique, l'analyse avec MapReduce, et enfin la conception du Tableau de bord.

CHAPITRE 4: ETUDE DE L'EXISTANT ET IDENTIFICATION DES BESOINS

Introduction :

À cause de l'augmentation de la masse de données contenues dans les systèmes opérationnels, on peut se trouver dans l'incapacité de faire des analyses correctes, rapides et efficaces au bon moment de ses données.

Toute entreprise veut exploiter efficacement ses données stockées et fidéliser ses clients en offrant les meilleurs produits et services.

Pour cela, après que l'équipe de l'entreprise Icon Software qui est chargé de superviser l'activité informatique du concessionnaire automobile a fait des études et des analyses sur les bases de données, ils ont décidé qu'il fallait réaliser un système décisionnel à fin de rendre facile la présentation de l'information de l'organisation de manière cohérente et de répondre aux requêtes compliquées des analystes. Ensuite ils ont pensé à utiliser les bases de données NoSQL pour avoir un schéma dynamique et pour ne pas être limité par les règles du modèle relationnel.

Cet entrepôt présentera le cœur du décisionnel de l'entreprise.

1) Présentation du domaine :

Un concessionnaire automobile est le professionnel responsable d'une concession automobile, société qui commercialise l'ensemble des modèles de voitures neuves d'un constructeur de véhicules. Il est le revendeur officiel de la marque automobile, il reçoit ses clients dans sa concession auto, dans laquelle

l'ensemble des voitures neuves commercialisées par le constructeur automobile sont exposées.

Le rôle du concessionnaire automobile est de conseiller ses clients dans l'achat de leur voiture neuve (gamme, modèle, motorisation et équipement). Il est le spécialiste local de la marque de voiture qu'il représente. Au-delà de la vente de voitures neuves, le concessionnaire automobile propose également des prestations d'entretien des véhicules par des mécaniciens spécialistes de la marque, donc le service après-vente, et de vente d'accessoires et de pièces de rechange pour les modèles qu'il commercialise.

2) Système opérationnel :

Dans notre cas d'étude, **SID AUTO** possède trois principales applications : L'application **VenteVN (véhicules Neufs)**, l'application **ventesPiècesRechanges** et l'application **ServiceAprèsVente**.

- **L'application VenteVN (Véhicule neuf)**, est construite autour de:
 - L'enregistrement des opérations des ventes Véhicules neufs.
 - La gestion des commandes et des livraisons.
- **L'application VentesPiècesRechange**, est construite autour de:
 - L'enregistrement des opérations des ventes des pièces de rechange.
 - L'enregistrement des opérations des ventes des pièces de rechange ratées.
- **L'application ServiceAprèsVente**, est construite autour de:
 - L'enregistrement des réparations effectuées.
 - La gestion de la main d'œuvre et des pièces utilisées.

3) Aperçu sur la procédure de prise de décision :

Plus le temps passe, plus le besoin des managers et des décideurs d'avoir une vision claire du fonctionnement de l'entreprise à n'importe quel moment augmente. Pour cela ils utilisent, les rapports générés par les systèmes opérationnels.

Ils procèdent en général comme le schéma ci-dessous le montre:

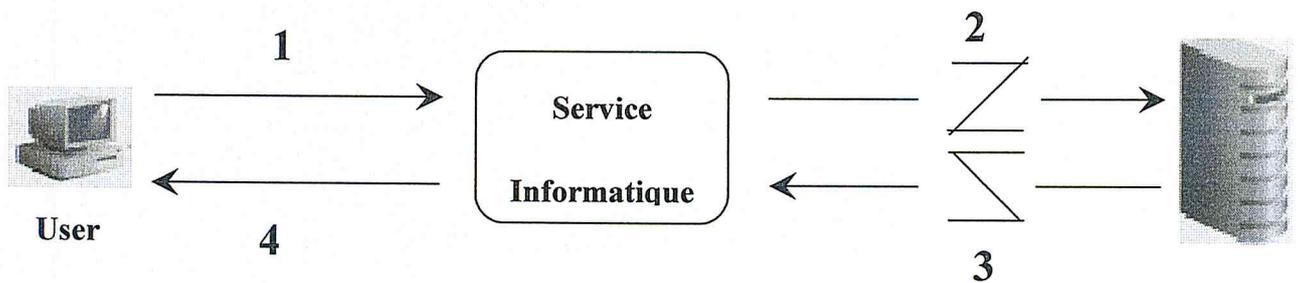


Figure 13 : Schéma représentatif des étapes d'édition des rapports.

Comme nous le montre cette figure, l'édition des rapports passe par quatre phases :

Phase 1 : Demande du rapport

Les décideurs, les analystes et les responsables ont besoin des rapports, ils doivent passer une demande de rapport à l'un des services de la direction en question, à son tour il contrôle la demande et la transmet au service informatique. Ce service est chargé de traiter ces demandes.

Phase 2 : Traitement de la demande

Les membres du service, en recevant la demande du rapport, passent à la détermination de la source de données puis ils préparent la requête et éditent le rapport final.

Phase 3 : Edition des rapports

Après le traitement de la demande, les rapports finaux seront mis dans un fichier qui sera imprimé et envoyé.

Phase 4 : Accès aux rapports

Dans cette phase, le demandeur concerné peut recevoir le rapport demandé.

Mais avec l'augmentation rapide des clients, des commandes et des ventes, l'utilisation de ces rapports commence à perturber le système de production et cause une perte de temps. D'où la solution décisionnelle est meilleure.

4) Etat du décisionnel :

Il est intéressant de signaler que la majorité des concessionnaires automobiles parmi eux le concessionnaire de notre cas pratique, ne disposent d'aucun système d'aide à la décision automatique ou semi-automatique. Aussi, tout processus d'analyse et de prise de décision à tous les niveaux se base essentiellement sur des rapports dont les données sont extraites et consolidés à partir des systèmes transactionnels d'une manière manuelle.

5) APPROCHE DEFINITION DES BESOINS :

Afin de mieux assimiler les besoins on a suivi l'approche de RALPH KIMBALL. L'interview principale doit mettre un questionnaire au point avant le début des entretiens avec les utilisateurs de l'entreprise en ce qui concerne leurs travail ,leurs enjeux et leurs objectifs en essayant de comprendre leurs manières de prendre les décisions actuelles et celles à venir.

5.1) Elaboration de questionnaires : [Ralph, 2000]

Ces questionnaires varient selon la fonction et le niveau hiérarchique de chaque personne interrogée. Dans ce cas nous n'allons pas poser les mêmes questions aux dirigeants et aux analystes.

➤ Questions aux dirigeants :

La rencontre avec les cadres permet de rassembler les informations nécessaires à la compréhension globale de l'entreprise et de ses orientations. Cette perspective nous aide à établir un lien entre les données collectées par ailleurs dans l'organisation.

Les questions de cette partie portent sur :

- Des informations relatives au cadre dirigeant interrogé.
- Des informations permettant de comprendre de façon globale les activités de l'entreprise et ses orientations.

- Des informations permettant de connaître les fonctions et départements de l'organisation qui sont primordiaux, et les rôles qu'ils jouent.
- Des informations concernant les problèmes liés aux prises de décisions.

dirigeant et la manière dont il envisage l'information.

➤ **Questions aux responsables et analystes :**

Dans cette partie nous essayerons de rentrer d'avantage dans le détail.

Elle comprend les questions portant sur :

- Des informations relatives au responsable ou analyste interrogé.
- Des informations sur les types d'analyse qui sont effectuées, et à partir de quelles sources et avec quels outils.
- Des informations sur les problèmes affrontés aujourd'hui.
- Des informations sur les analyses souhaitées au futur.
- Des informations sur les méthodes de diffusion envisagées.
- Des informations sur les types d'analyses prédéfinies (rapports) qui sont déjà effectués, et comment utilisent ils ces informations.

5.2) L'ENTRETIEN :

L'entretien se révèle comme le moyen le plus efficace pour collecter les informations nécessaires à notre travail, il permet entre autre de connaître les impressions de notre interlocuteur .L'interviewe peut en revanche proposer des suggestions et nous éclaircir sur des points pas assez précis dans le questionnaire.

6) Extraction des besoins :

Après avoir cerné les besoins d'analyse au sein de l'entreprise, ils ont été classés comme suit:

❖ **Analyse des Ventes des véhicules neufs :**

➤ **Chiffre d'affaire par client :**

- Par catégorie client (entreprise, ordinaire).
- Par profession.

- Par wilaya/région.
- Chiffre d'affaire par vendeur.
- Chiffre d'affaire par évènement.
- Chiffre d'affaire par rapport au cadre de vente.
- Chiffre d'affaire par période (mois, année ...).
- Chiffre d'affaire par véhicule :
 - Par catégorie (Commercial, touristique).
 - Par modèle de véhicule.
 - Par famille de véhicule (sport, simple).
 - Par options.
 - Par modèle de véhicule et par période.
- Chiffre d'affaire :
 - Par modèle de véhicule et par catégorie du client.
 - Par catégorie de véhicule et catégorie client.
 - Par modèle de véhicule et par évènement.
- Le nombre de véhicules vendus :
 - Par période.
 - Par évènement (salon de l'automobile,..).
 - Par client.
 - Par catégorie client.
 - Par profession.
 - Par wilaya.
 - Par client et par période.
 - Par catégorie client.
 - Par catégorie de véhicule (Commercial, touristique).
 - Par famille de véhicule.
 - Par modèle de véhicule.
 - Par couleur de véhicule.
 - Par option du véhicule.
 - Par vendeur.
 - Par catégorie de véhicule et par catégorie de client.

- Par modèle de véhicule et par période.
- Par modèle de véhicule et par couleur.
- Par modèle véhicule et évènement.
- Par cadre de vente.
- Meilleur client par période.
- Les bénéfices :
 - Par modèle de véhicule.
 - Par famille de véhicule.
 - Par catégorie de véhicule.
 - Par période.
 - Par modèle véhicule et par période.
 - Par évènement.
 - Par vendeur.
 - Par catégorie client.
 - Par wilaya/region.
- ❖ **Analyse de l'activité Services après-vente :**
 - Temps moyen de réparation :
 - Par tache.
 - Temps de réparation par dossier.
 - Temps de travail de chaque mécanicien par période.
 - Les pièces les plus remplacées.
 - Les pièces les plus remplacées par évènement.
 - Chiffre d'affaire par tache.
 - Chiffre d'affaire par évènement.
 - Chiffre d'affaire par pièce (vente pièces/atelier).
 - Les types de réparations (rappel, retour, normal) les plus fréquents.
 - Les bénéfices par tache.
 - Les taches les plus demandées.
 - Les états de réparations les plus constatés par tache
- ❖ **Analyse des Ventes des pièces de rechange et des ventes ratées :**
 - Chiffre d'affaire
 - Par pièce.

- Par période.
- Par client.
- Par client et période.
- Par évènement.
- Par type de vente (atelier, comptoir).
- Par famille de pièces.
- Wilaya/region
- Nombre de vente :
 - Par pièce.
 - Par période.
 - Par type de vente (atelier, comptoir).
 - Par évènement.
 - Par client.
 - Par famille de pièce.
- Ventes ratées :
 - Par pièce.
 - Par période.
- Les bénéfices :
 - Par pièce.
 - Par famille de pièce.
 - Par période.
 - Par évènement.
 - Par type de vente (atelier, comptoir).
 - Par client.
 - Par client et période.
 - Wilaya/region
- ❖ **Analyse des commandes :**
 - Nombre de commandes
 - Par client (catégorie/profession/wilaya).
 - Par période.
 - Par véhicule (model/catégorie/famille/couleur option).

- Par vendeur.
- Nombre de véhicule livré par période.
- Temps moyen de livraison par modèle de véhicule.

Conclusion :

La réussite de notre projet exige une bonne identification des besoins des utilisateurs finaux, les analystes et les décideurs, ce qui implique une bonne conception du Data Warehouse.

Nous avons identifié lors de ce chapitre une liste des besoins des décideurs en utilisant l'approche de l'entretien. C'est à partir de ces besoins que nous construirons les indicateurs et les axes de notre conception dimensionnelle que nous allons présenter dans le chapitre qui suit.

CHAPITRE 5 : CONCEPTION DES ELEMENTS DU SYSTEME

Introduction :

Une fois les besoins collectés et les données auditées, nous voilà prêts à lancer la conception de notre Data Warehouse.

La modélisation dimensionnelle est le nom de la méthode de conception logique souvent associée aux entrepôts de données. Elle vise à présenter les données sous une forme standardisée intuitive et qui permet des accès hautement performants.

Cette méthode diffère de la modélisation entité/relation qui vise à éliminer les données redondantes et apporte de nombreux avantages au niveau du traitement des transactions qui deviennent simple et déterministe.

1) Modélisation multidimensionnelle:

1.1) Présentation :

Notre objectif est de proposer une modélisation multidimensionnelle des données permettant de fournir aux utilisateurs finaux des indicateurs et des états, et d'exploiter à mieux les données stockées au niveau du nouveau système de Data Warehouse.

Dans la modélisation multidimensionnelle on distingue les dimensions, les indicateurs et les faits.

- **Les dimensions** : sont les points de vue, depuis lesquels les “ mesures “ peuvent être observées. Par exemple, dans notre cas: **date, région, véhicule, etc.**
- **Les Indicateurs** : sont les valeurs numériques que l'on compare .Ces valeurs offrent les analyses, en fonction des différentes dimensions.
- **Les faits** : Un fait représente la valeur d'une mesure, (mesurée ou calculée), selon un membre de chacune des dimensions.

1.2) Description des activités:

Il faut rappeler que l'étude de notre Data Warehouse, porte sur l'activité commerciale d'un concessionnaire automobile SID AUTO. Cependant, celle-ci se décompose en quatre sous activités (**Ventes Véhicule neuf, Services après- vente, le pilotage des commandes et Ventes pièces de rechange**).

Le tableau suivant, résume les indicateurs déterminés, par axes d'analyse

Activité	Indicateur	Axes d'analyse
Ventes Véhicules neufs	Chiffre d'affaire	<ul style="list-style-type: none"> • client (catégorie, profession, wilaya/région) • véhicule (modèle, catégorie, famille, option, couleur) • vendeur • cadre de vente • évènement • période (jour, mois, année) • modèle de véhicule et période • modèle de véhicule et catégorie client • catégorie de véhicule et catégorie client • par modèle et par évènement
	Nombre de vente	<ul style="list-style-type: none"> • période (jour, mois, année) • wilaya/région • véhicule (modèle, catégorie, famille, option, couleur) • vendeur • cadre de vente • évènement • client (catégorie, profession, wilaya/région) • client et la période • catégorie du véhicule et catégorie client • modèle du véhicule et la période • modèle du véhicule et par couleur • modèle du véhicule et par évènement
	Meilleur client	<ul style="list-style-type: none"> • période (jour, mois, année)
	Bénéfices	<ul style="list-style-type: none"> • période (jour, mois, année) • véhicule (modèle, catégorie, famille)

		<ul style="list-style-type: none"> • cadre de vente • évènement • vendeur • wilaya/région • modèle du véhicule et la période • catégorie client
Services après-vente	Temps moyen de réparation	<ul style="list-style-type: none"> • tache
	Temps de réparation	<ul style="list-style-type: none"> • dossier
	Chiffre d'affaire	<ul style="list-style-type: none"> • tache • évènement • période (jour, mois, année) • pièce
	Nombre remplacement	<ul style="list-style-type: none"> • pièce • pièce et par évènement
	Nombre de taches	<ul style="list-style-type: none"> • par période (jour, mois, année) • évènement • type de réparation
	Nombre d'heures de travail des mécaniciens	<ul style="list-style-type: none"> • période (jour, mois, année)
	Etats de réparation	<ul style="list-style-type: none"> • tache • mécanicien
	Bénéfices	<ul style="list-style-type: none"> • tache

Ventes pièces de rechange	Chiffre d'affaire	<ul style="list-style-type: none"> • pièce • famille pièces • client • client et période • évènement • type de vente • période (jour, mois, année) • wilaya/région
	nombre pièce vendus	<ul style="list-style-type: none"> • période (jour, mois, année) • type de vente • évènement • client • famille pièces
	Bénéfices	<ul style="list-style-type: none"> • pièce • client • client et période • famille pièces • évènement • type de vente • période (jour, mois, année) • wilaya/région
ventes pièces ratées	Ventes ratées	<ul style="list-style-type: none"> • pièce • période (jour, mois, année)
Pilotage des commandes	Nombre de commandes	<ul style="list-style-type: none"> • véhicule • période (jour, mois, année) • client • vendeur
	Temps moyen de livraison	<ul style="list-style-type: none"> • modèle véhicule

	Nombre de véhicule livré	• période
--	--------------------------	-----------

Tableau 4 : Tableau résumé des indicateurs par axes d'analyse.

1.3) Description des dimensions :

Une dimension contient des membres organisés en hiérarchie : Chacun des membres appartient à un niveau hiérarchique (ou niveau de granularité) particulier. Dans le tableau suivant nous représentons toutes les dimensions déterminées de notre 'Data Warehouse ', après l'étude et les entretiens effectués:

Dimension	Description
Date	Représente le temps, contient le jour, le mois, l'année.
Client	Représente toutes les informations concernant le client
Catégorie_client	Représente la catégorie du client (entreprise, normal)
Profession	Représente la profession du client
Wilaya	Représente la wilaya/région
Région	Représente la région
Véhicule	Représente les informations concernant le véhicule
Catégorie_vehicule	Représente la catégorie du véhicule (commercial, touristique)
Famille_vehicule	Représente la famille du véhicule (sport, simple)
Couleur_vehicule	Représente la couleur du véhicule
Option_vehicule	Représente le type des options (la base, base clim, toute option)
Commande	représente les informations concernant une commande

Vendeur	Représente les agents commerciaux
Evénement	Représente les événements particuliers (le salon de l'automobile, promotion)
Cadre de vente	Représente le cadre de vente (licence moudjahid, ancege, normal, ..)
Pièce	Représente les informations concernant une pièce de rechange
Famille pièce	Représente la famille de la pièce
Type de vente	Représente le type de vente de la pièce (comptoir, atelier)
Livraison	Représente les informations concernant une livraison
Garantie	Représente le type de garantie choisit par client
Dossier	Représente le dossier de la réparation (date, véhicule, client)
Mécanicien	Représente les informations d'un mécanicien
Tache	Représente les informations concernant une tache (désignation, cout, temps)
Type_réparation	Représente le type de la réparation (rappel, retour, normal)
Etat_réparation	Représente l'état de la réparation (bon, défectueux, a revoir)
etat_garantie	Représente l'état de la garantie (sous garantie, hors garantie)

Tableau 5 : Tableau des dimensions.

❖ **Activité < ventes pièces de rechange > :**

Cette activité englobe tous les détails de la vente des pièces de rechanges, dans ce processus il est important de connaître périodiquement le **nombre de pièces vendues**, le **chiffre d'affaire** par rapport aux différents axes (pièce (famille pièce), client (profession, wilaya, catégorie), type de vente (atelier ou comptoir)), **bénéfice** par rapport aux différents axes.

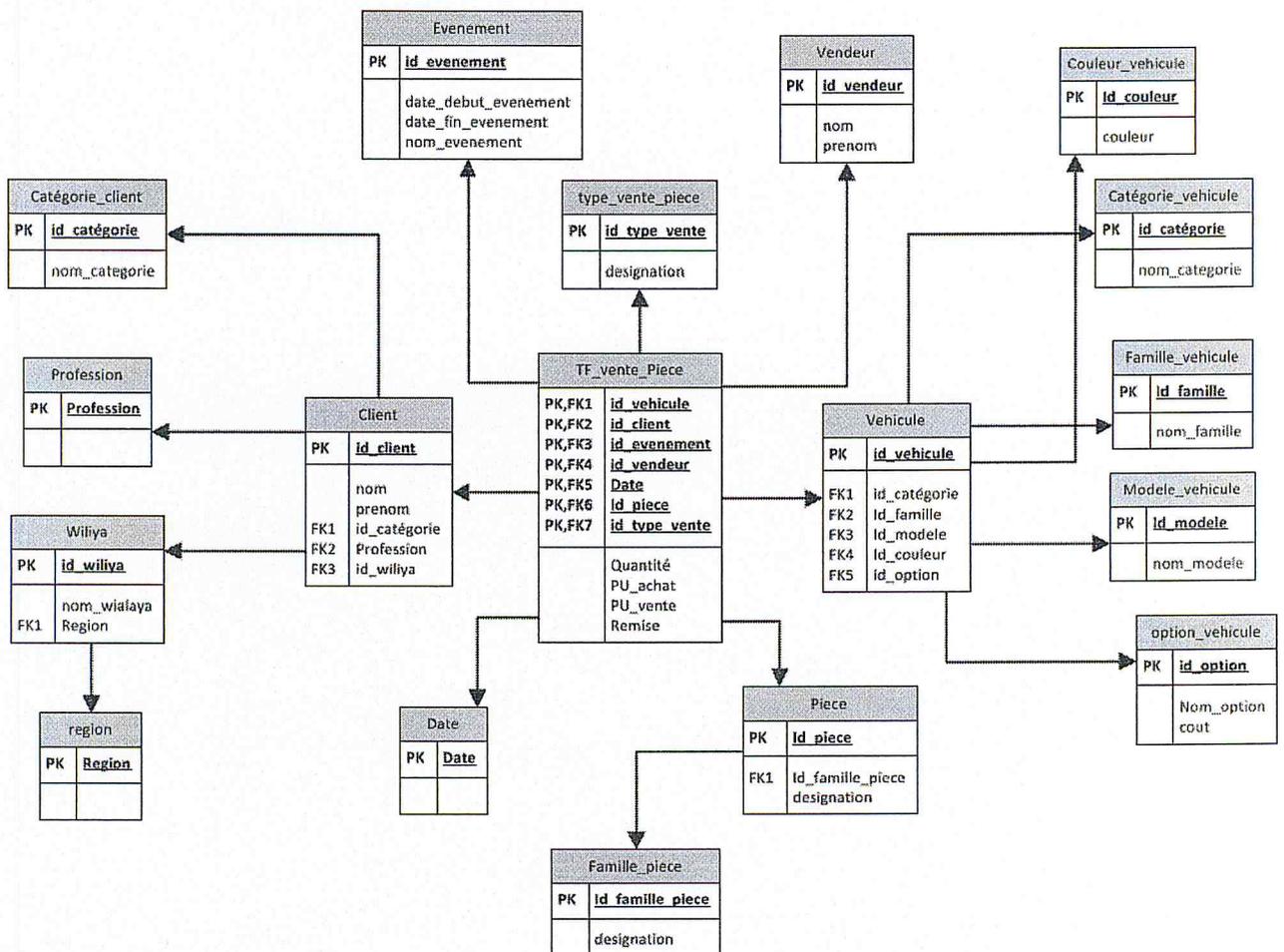


Figure 15 : Modèle multidimensionnel en flocon de neige pour l'activité « vente pièces de rechanges ».

❖ **Activité < ventes pièces ratées > :**

Dans ce processus il est important de connaître le **nombre** de ventes ratées, les **pièces** concernées par les ventes ratées, ainsi que la période.

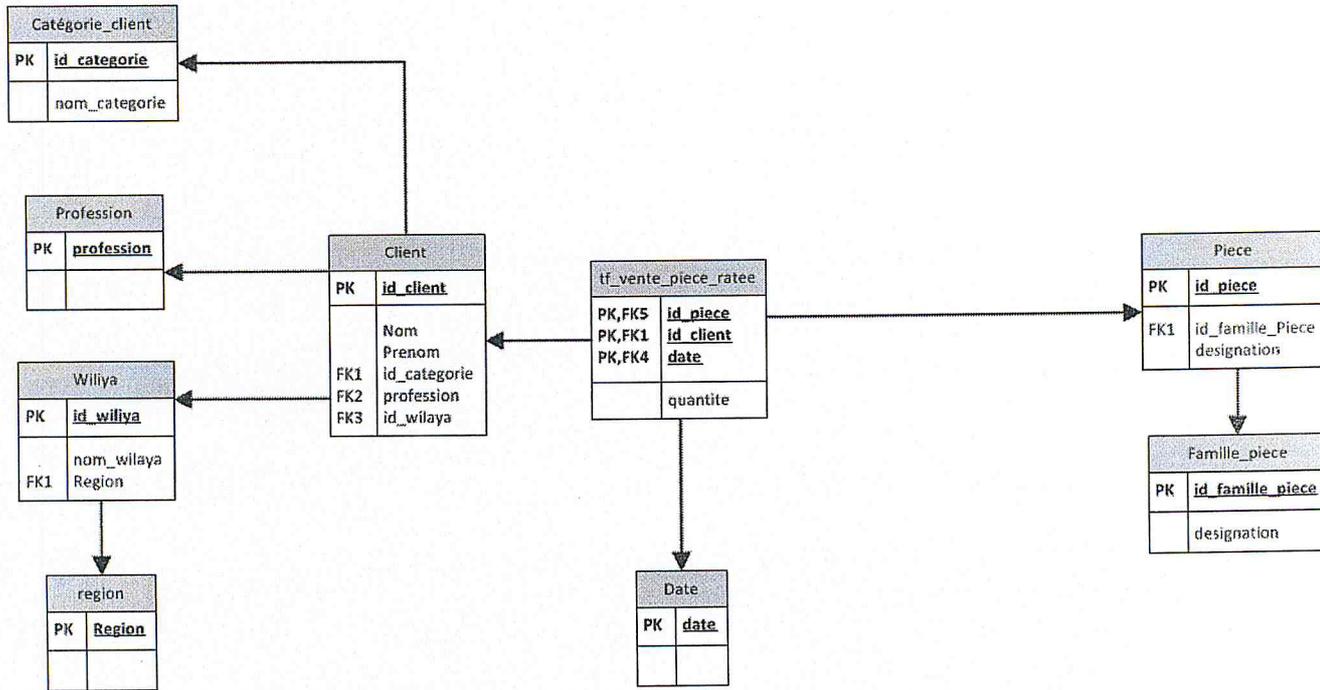


Figure 16 : Modèle multidimensionnel en flocon de neige pour l'activité «vente pièces ratées».

❖ **Activité < service après vente > :**

Cette activité englobe tous les détails du service après vente ,que ca soit du coté des rendements d'argent ou réparation, Dans ce processus il est important de connaitre périodiquement le **nombre de réparations** effectuées, le nombre par événement, les taches les plus effectuées, type de réparations les plus fréquents ,les états de contrôle les plus constatés par tache, le **temps moyen de réparation** par tache, par dossier, **meilleur mécanicien**, Les pièces les plus remplacées, le **chiffre d'affaire** par tache et par événement, le **bénéfice** par tache.

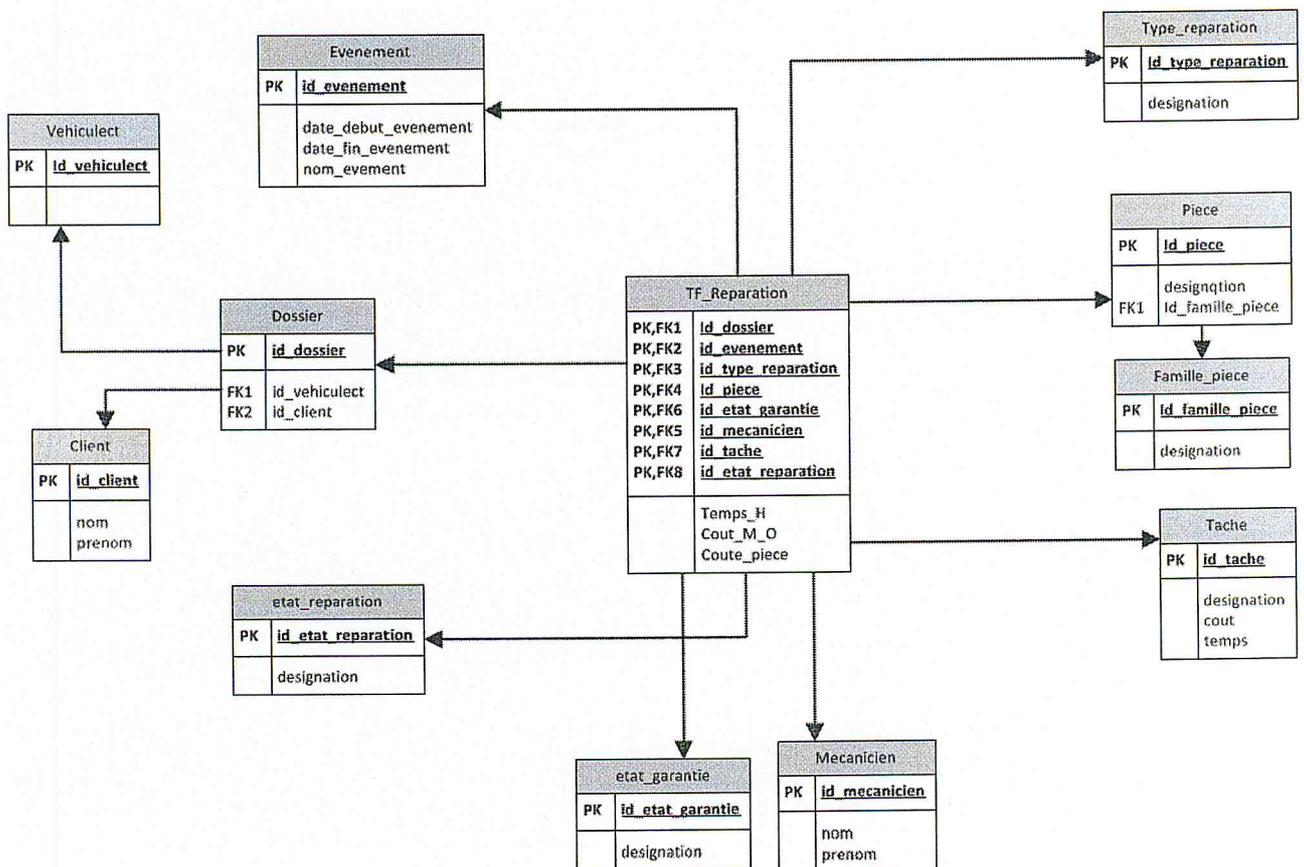


Figure 17 : Modèle multidimensionnel en flocon de neige pour l'activité « service après vente ».

❖ **Activité < pilotage des commandes > :**

Cette activité englobe tous les détails du pilotage des commandes, dans ce processus, il est important de connaître périodiquement le **nombre de commandes** par vendeur, par véhicule, par client, événement, classer les véhicules les plus commandés par période et par rapport à un événement, le **temps moyen de livraison** d'un véhicule par rapport à sa catégorie, famille et son modèle.

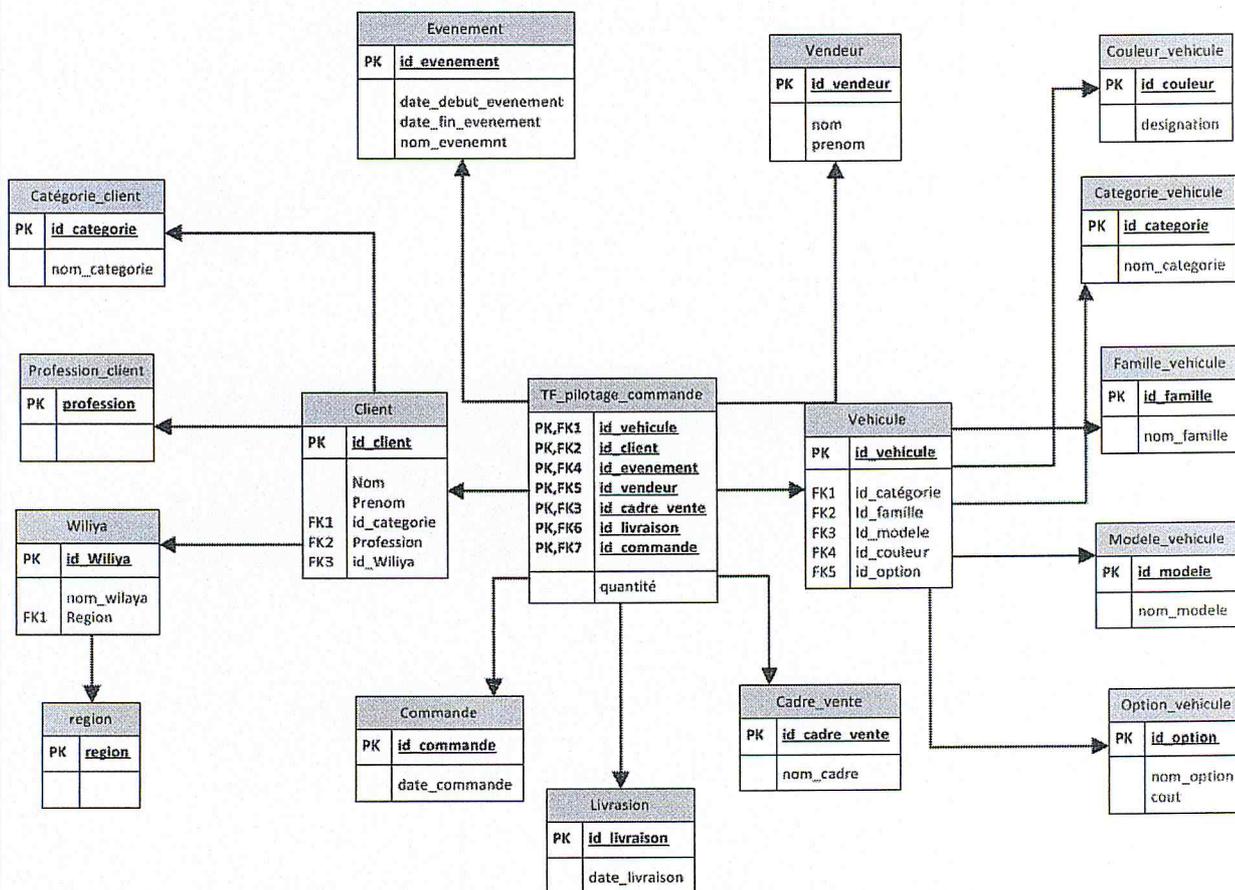


Figure 18 : Modèle multidimensionnel en flocon de neige pour l'activité «pilotage des commandes».

1.5) Schéma de l'entrepôt de données :

La transformation de la conception logique en une base de données physique est représentée comme suit.

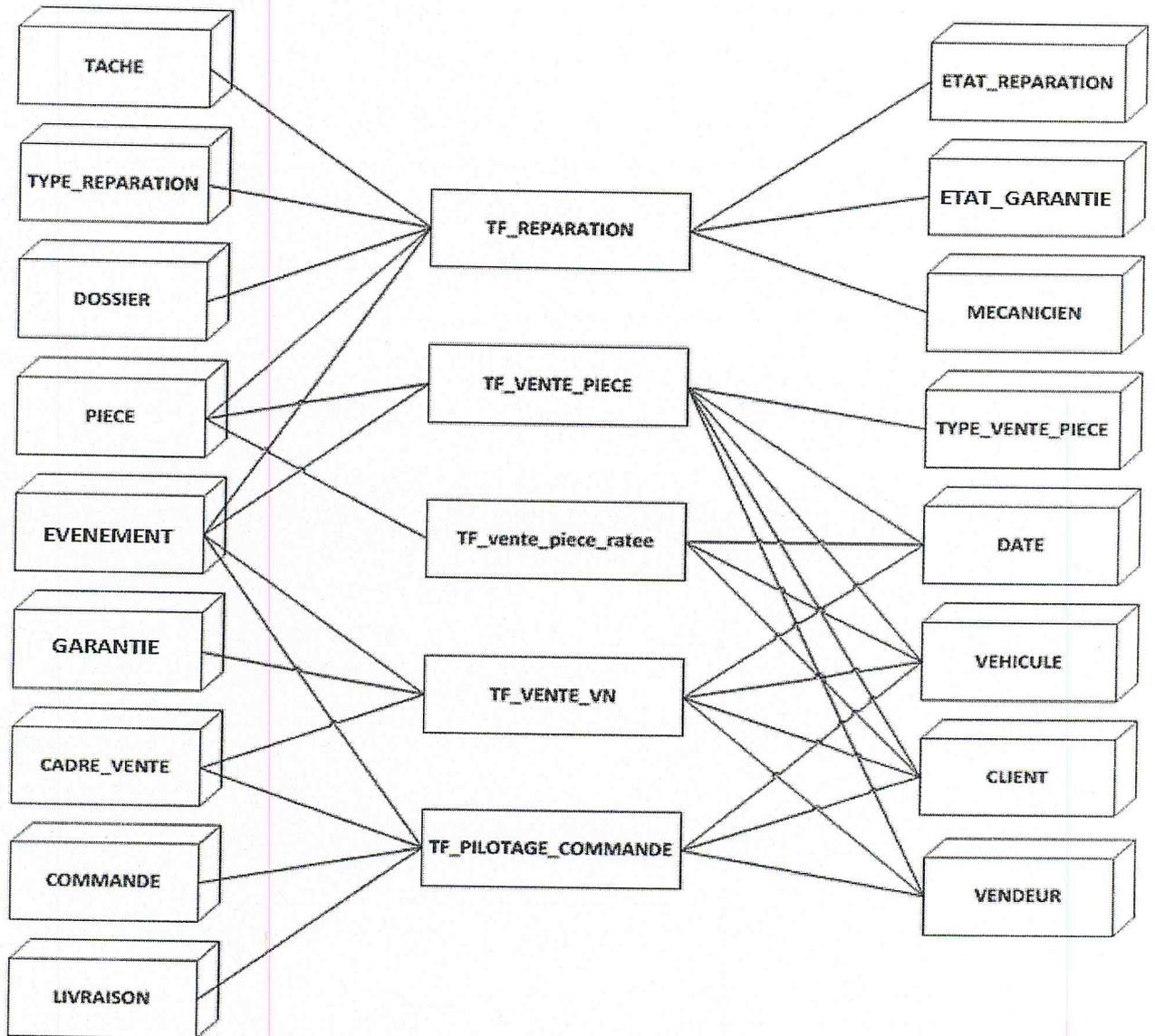


Figure 19 : Schéma relationnel de la base décisionnelle.

2) Architecture détaillée du système :

La figure suivante illustre l'architecture détaillée de notre Système. Cette architecture comprend les différents composants du système décisionnel à savoir l'entrepôt de données, l'ETL et le portail de restitution.

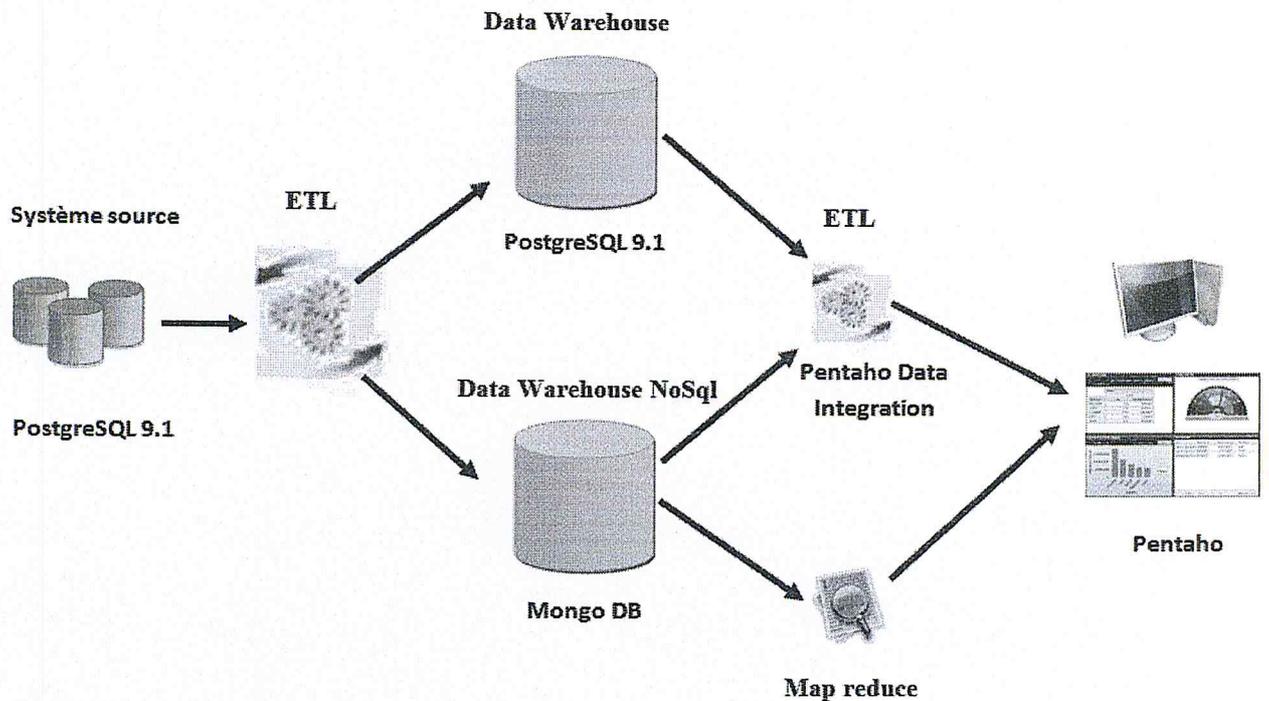


Figure 20 : Architecture du système.

3) Construction du Data Warehouse NoSQL :

Dans ce qui suit, nous allons décrire l'étape requise pour transformer la conception logique de notre entrepôt de données en une base de données physique. Notre objectif est d'apporter aux responsables de l'entrepôt de données suffisamment de connaissance pour qu'ils puissent prendre des décisions en termes d'administration et de suivi.

Le point de départ de la conception du modèle physique de données est le modèle logique de données. Le premier doit refléter le second au plus près, bien que certains changements soient nécessaires dans la structure du modèle pour s'adapter aux particularités du SGBD et des outils d'accès.

Nous avons choisit de créer un Data Warehouse NoSQL pour avoir un schéma de données dynamique et pour la rapidité, et parmi les quatres solutions du NoSQL citées précédemment nous avons opté pour la solution **orienté document** qui d'après nos recherches est réputée par sa performance et sa simplicité, et les outils qui l'implémentent sont plus avancés par rapport aux outils qui implémentent les autres solutions du NoSQL.

Notre modèle multidimensionnel est alors traduit de la manière suivante :

- Une table correspond à une collection.
- Une ligne correspond à un document.
- Une Jointure correspond à un lien entre les collections, on a choisit ce type de jointure pour minimiser les redondances par rapport a la solution de documents imbriqués.

3.1) Liste des collections :

Selon notre modélisation logique, chaque entité est transformée en une collection de base de données NoSQL.

Le tableau suivant contient la liste des collections de notre Data Warehouse NoSQL, et les attributs normalement contenu dans chaque document des collections, sachant que les attributs peuvent changer d'un document à un autre, car le schéma est dynamique.

Collections	Attributs	JSON
TF_Vente_VN	date Id_vehicule Id_client Id_evenement Id_garentie Id_cadre_vente Id_vendeur Quantite Remise Prix_vente Pourcentage_vente	<pre>dw.TF_Vente_VN.insert ({ "date" : " ", "id_vehicule" : " ", " id_client" : " ", "id_evenement" : " " , "id_garentie" : " " , " id_cadre_vente " : " " , " id_vendeur" : " "});</pre>
TF_Vente_Piece	date Id_vehicule Id_client Id_piece Id_vendeur Id_type_vente Id_evenement Quantite Prix_achat Prix_vente Remise	<pre>dw.TF_Vente_Piece.insert ({"date" : " ", "id_vehicule" : " " , " id_client" : " " , "id_piece" : " " , " Id_vendeur" : " " , " Id_type_vente" : " " " , " id_evenement" : " " });</pre>
TF_Vente_Piece_Ratee	Id_piece Id_client Id_vendeur Date Quantite	<pre>dw.TF_Vente_Piece_Ratee.insert ({ "id_client" : " " , "id_piece" : " " , "date" : " " });</pre>
	Id_piece	dw.TF_Pilotage_Commande.insert

CHAPITRE 5 : CONCEPTION DES ELEMENTS DU SYSTEME

TF_Pilotage_Commande	Id_vehicule Id_livraison Id_commande Id_evenement Id_vendeur Id_cadre_vente Quantite Remise	({ "id_piece" : " " , "id_vehicule" : " " , "id_livraison" : " " , "commande" : " " , "id_evenement" : " " , "Id_vendeur" : " " " , "Id_cadre_vente" : " " });
TF_Reparation	Id_dossier Id_piece Id_client Id_tache Id_mecanicien Id_type_reparation Id_etat_controle Id_garentie Temp_h Cout_piece Cout_MO	dw. TF_Reparation.insert ({ "Id_dossier" : " " , "id_piece" : " " , "id_client" : " " , "id_tache" : " " , "id_mecanicien" : " " , "id_type_reparation" : " " , "Id_etat_controle" : " " " , "Id_garentie" : " " });
Client	Id_client Nom Prenom Id_wilaya Id_categorie_client Profession	dw. Client.insert ({ "Id_client" : " " , "nom" : " " , "prenom" : " " , "id_wilaya" : " " , "id_categorie_client" : " " , "profession" : " " });
Categorie_Client	Id_categorie Nom_categorie	dw. Categorie_Client.insert ({ "Id_categorie" : " " " , "nom_categorie" : " " });
Profession_Client	Profession_client	dw.Profession_Client.insert ({ "profession_client" : " " });
Wilaya	Id_wilaya Wilaya	dw. Wilaya.insert ({ "Id_wilaya" : " " , "wilaya" : " "

	Region	, "region" : " " });
Vehicule	Id_vehicule Id_modele Id_famille Id_categorie Id_couleur Id_option	dw.vehicule.insert ({ "Id_vehicule" : " " , "id_modele" : " " , "id_famille" : " " , "id_categorie" : " " , "id_couleur" : "" , "id_option" : " " });
Famille_vehicule	Id_famille Nom_famille	dw.Famille_vehicule.insert ({ "Id_famille" : " " , "nom_famille" : " " });
Categorie_Vehicule	Id_categorie Nom_categorie	dw.Categorie_Vehicule.insert ({ "Id_categorie" : " " , "nom_ categorie" : " " });
Couleur_Vehicule	Id_couleur Designation	dw.Couleur_Vehicule.insert ({ "Id_couleur" : " " , "designation" : " " });
Modele_Vehicule	Id_modele Nom_modele	dw.Modele_Vehicule.insert ({ "Id_modele" : " " , "Nom_modele" : " " });
Option_Vehicule	Id_option Nom_option	dw.Option_Vehicule.insert ({ "Id_option " : " " , "Nom_option" : " " });
Piece	Id_piece Designation	dw.Piece.insert ({ "Id_piece " : " " , "designation" : " " });
Famille_piece	Id_famille_piece Designation	dw.Famille_piece.insert ({ "Id_famille_piece" : " " , "designation" : " " });
Evénement	Id_evenement Date_debut_evenement Date_fin_evenement Nom_evenement	dw.Evénement.insert ({ "Id_evenement" : " " , "Date_debut_evenement" : " " , "Date_fin_evenement" : " " ,

CHAPITRE 5 : CONCEPTION DES ELEMENTS DU SYSTEME

		“Nom_evenement” : “ “ });
Vendeur	Id_vendeur Nom Prenom	dw.Vendeur.insert ({ "Id_vendeur" : " " , "nom" : " " , "prenom" : " " });
Cadre_Vente	Id_cadre_vente Nom_cadre	dw.Cadre_Vente.insert ({ "Id_cadre_vente" : " " , "Nom_cadre" : " " });
Garantie	Id_garantie Type_garantie	dw.Garantie.insert ({ "Id_garantie" : " " , "Type_garantie" : " " });
Date	Date	dw.Date.insert ({ "date" : " " });
Type_Vente_Piece	Id_type_vente Designation	dw.Type_Vente_Piece.insert ({ "Id_type_vente" : " " , "designation" : " " });
Dossier	Id_dossier Id_vehiculect Id_client Date	dw.Dossier.insert ({ "Id_dossier " : " " , "Id_vehiculect" : " " , "Id_client" : " " , "date" : " " });
Type_Reparation	Id_type_reparation Designation	dw.Type_Reparation.insert ({ "Id_type_reparation" : " " , "designation" : " " });
Etat_Reparation	Id_etat_reparation Designation	dw.Etat_Reparation.insert ({ "Id_etat_reparation " : " " , "designation" : " " });
Tache	Id_tache Designation Cout Temps	dw.Tache.insert ({ "Id_tache " : " " , "Designation" : " " , "Cout" : " " , "temps" : " " });
Commande	Id_commande Date_commande	dw.Commande.insert ({ "Id_commande" : " " , "date_commande" : " " });

Livraison	Id_livraison Date_livraison	dw.Livraison.insert ({ "Id_livraison" : " " ,"date_livraison" : " " });
etat_garantie	Id_etat_garantie Garentie	dw. etat_garentie.insert ({ "Id_etat_garantie " : " " , "garentie" : " " });
Mecancier	Id_mecancier Nom Prenom	dw.Mecancier.insert ({ "Id_mecancier " : " " , "nom" : " " , "prenom" : " " });
VehiculeCT	Id_vehiculect	dw.VehiculeCT.insert ({ "Id_vehiculect" : " " });

Tableau 6 : Liste des collections du Data Warehouse.

4) Alimentation du Data Warehouse NoSQL:

Les données doivent être chargées depuis leurs sources initiales vers le Data Warehouse.

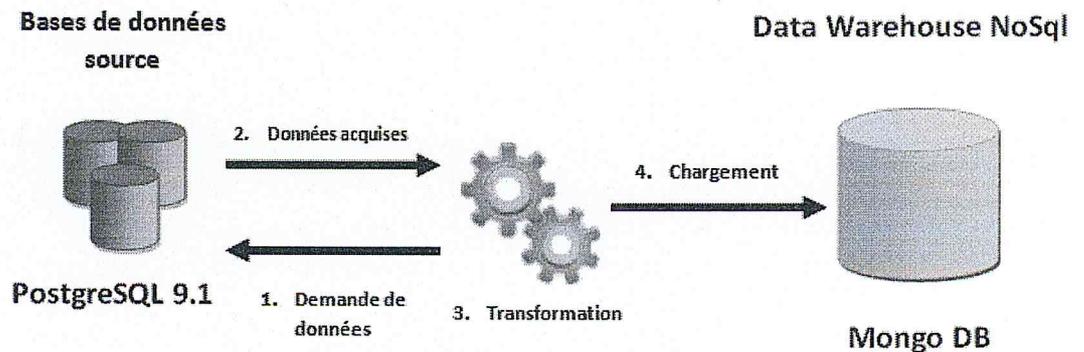


Figure 21 : Processus d'alimentation du Data Warehouse NoSQL.

Pour l'alimentation du Data Warehouse nous avons utilisé un programme en Java qu'on a nommé 2F Loader qui suit la logique suivante :

4.1) Demande et Acquisition des données :

À partir des bases de données sources qui sont implémentées avec PostgreSQL, nous avons choisi les tables, les colonnes dont on aura besoin, et qui nous intéressent dans notre étude. La récupération des données se fait via notre programme.

4.2) Transformation :

Dans cette étape plusieurs transformations ont été faites, à titre d'exemple nous avons recensé tous les véhicules possible et pour chaque cas nous avons attribué un identifiant unique, codifier les taches, concernant les pièces de rechanges nous avons fait des jointures à partir de différentes tables pour différencier les ventes comptoir et atelier, Nous avons aussi calculer le cumul par rapport a chaque id_piece et par type de vente(atelier /comptoir), nous vérifions aussi avant le chargement dans MongoDB les attributs qui sont nulles ou vides.

4.3) chargement :

Dans cette étape 2F Loader charge les données dans le Data Warehouse NoSQL, en créant lors du chargement les collections et les attributs.

Nous avons utilisé **JSON** (*JavaScript Object Notation*) qui est un format de données textuel que MongoDB reconnait.

5) Construction du Data Warehouse Classique :

Pour pouvoir comparer l'approche classique SQL et l'approche NoSQL nous avons décidé de construire un Data Warehouse classique.

Notre modèle multidimensionnel est alors traduit de la manière suivante :

- chaque fait correspond à une table, appelée table de fait.
- Chaque dimension correspond à une table, appelée table de dimension.

La table de fait est constituée par des attributs qui sont des clés étrangères de chacune des tables de dimension et des mesures d'activités. Les tables de dimension contiennent des attributs textuels et une clé primaire qui permet de réaliser des jointures avec la table de fait.

Le nombre de tables de notre Data Warehouse classique est le même que le nombre de collections du Data Warehouse NoSQL créé précédemment avec MongoDB.

Nous avons utilisé PostgreSQL pour la création du Data Warehouse classique.

Le tableau suivant montre les tables du Data Warehouse classique ainsi que les attributs de chaque table.

Table	Attributs
TF_Vente_VN	date Id_vehicule Id_client Id_evenement Id_garantie Id_cadre_vente Id_vendeur quantite Remise Prix_vente Pourcentage_vente
TF_Vente_Piece	date Id_vehicule Id_client Id_piece Id_vendeur Id_type_vente Id_evenement Quantite Prix_achat Prix_vente remise

TF_Vente_Piece_Ratee	Id_piece Id_client Id_vendeur Date Quantite
TF_Pilotage_Commande	Id_piece Id_vehicule Id_livraison Id_commande Id_evenement Id_vendeur Id_cadre_vente Quantite
TF_Reparation	Id_dossier Id_piece Id_client Id_tache Id_mecanicien Id_type_reparation Id_etat_controle Id_etat_garantie Temp_h Cout_piece Cout_MO
Client	Id_client Nom Prenom Id_wilaya Id_categorie_client profession
Categorie_Client	Id_categorie

	Nom_categorie
Profession	Profession
Wilaya	Id_wilaya Nom_wilaya Region
Vehicule	Id_vehicule Id_modele Id_famille Id_categorie Id_couleur Id_option
Famille_vehicule	Id_famille Nom_famille
Categorie_Vehicule	Id_categorie Nom_categorie
Couleur_Vehicule	Id_couleur designation
Modele_Vehicule	Id_modele Nom_modele
Option_Vehicule	Id_option Nom_option
Piece	Id_piece designation
Famille_piece	Id_famille_piece designation
Événement	Id_evenement Date_debut_evenement Date_fin_evenement Nom_evenement
Vendeur	Id_vendeur Nom prenom

Cadre_Vente	Id_cadre_vente Nom_cadre
Garantie	Id_garantie Type_garantie
Date	date
Type_Vente_Piece	Id_type_vente designation
Dossier	Id_dossier Id_vehiculect Id_client date
Type_Reparation	Id_type_reparation designation
Etat_Reparation	Id_etat designation
Tache	Id_tache Designation Cout temps
Commande	Id_commande Date_commande
Livraison	Id_livraison Date_livraison
etat_garantie	Id_etat_garantie garentie
Mecancien	Id_mecanicien Nom prenom
VehiculeCT	Id_vehiculect

Tableau 7 : Liste des attributs des tables du Data Warehouse classique.

6) Alimentation du Data Warehouse Classique :

Pour cette partie nous avons utilisé Talend, et dans les cas où il y a des traitements complexes on utilise notre ETL 2f Loader, qui récupère les données des systèmes sources de PostgreSQL, fait le même traitement qu'avec MongoDB sauf qu'il ne vérifie pas les données nulles ou vides avant le chargement car le schéma ici n'est pas dynamique, et ensuite il charge les données dans le Data Warehouse créé dans PostgreSQL.

Nous utilisons ici le langage SQL pour la récupération et l'insertion dans les tables.

7) Exemple de comparaison entre Data Warehouse classique et Data Warehouse NoSQL :

Afin d'illustrer la différence entre un Data Warehouse classique et un Data Warehouse NoSQL, nous allons comparer MongoDB et PostgreSQL à travers un exemple simple.

Avec PostgreSQL, nous aurions été obligés dans un premier temps de créer deux tables :

- **Piece** (id_piece, designation, id_famille_piece)
- **famille_piece** (id_famille_piece, designation)

Puis nous aurions inséré les données avec une requête INSERT INTO(SQL), tandis qu'avec MongoDB, la structure est créée automatiquement à la première insertion et peut changer à tout moment.

Exemple :

```
var P = {  
  
    id_piece : "0007452y7",  
  
    designation : « ENS enjoliveurs »,
```

```
id_famille_piece : "690"    }
```

ou

```
var P = {
```

```
  id_piece : "0007452y7",
```

```
  designation : « ENS enjoliveurs »,
```

```
  famille_piece : [ {
```

```
    id_famille_piece : "690",
```

```
    designation : "divers pare chocs"
```

```
  } ]
```

```
}
```

```
db.piece.insert(P);
```

Ce qui nous permet de stocker nos données pièce et famille_pièce dans MongoDB.

Un champ « *_id* » est présent pour chaque donnée afin de conserver l'identification et l'unicité des données, Cette tâche est automatiquement effectuée par MongoDB.

MongoDB n'utilise pas des tables, mais des collections. En faisant `db.piece.insert`, nous avons inséré les données dans la collection « *piece* » si elle existe sinon MongoDB la crée automatiquement.

Maintenant, comment récupérer notre pièce ?

Avec PostgreSQL nous aurions fait un simple SELECT sur la table pièce, avec MongoDB, nous allons procéder de la manière suivante :

```
db.piece.find ({désignation : "ENS enjoliveurs"});
```

Ici, nous cherchons les pièces dont la désignation est « ENS enjoliveurs ».

8) MapReduce :

Nous avons défini dans la partie précédente le modèle de programmation parallèle MapReduce.

L'équipe d'Icon Software nous ont proposé d'utiliser MapReduce pour l'analyse des données, qui est un moyen efficace, il permet de traiter de grands volumes de données et de paralléliser et distribuer les traitements et il est souvent associé au NoSQL.

La logique d'utilisation du MapReduce dans notre projet est comme le montre le schéma suivant :

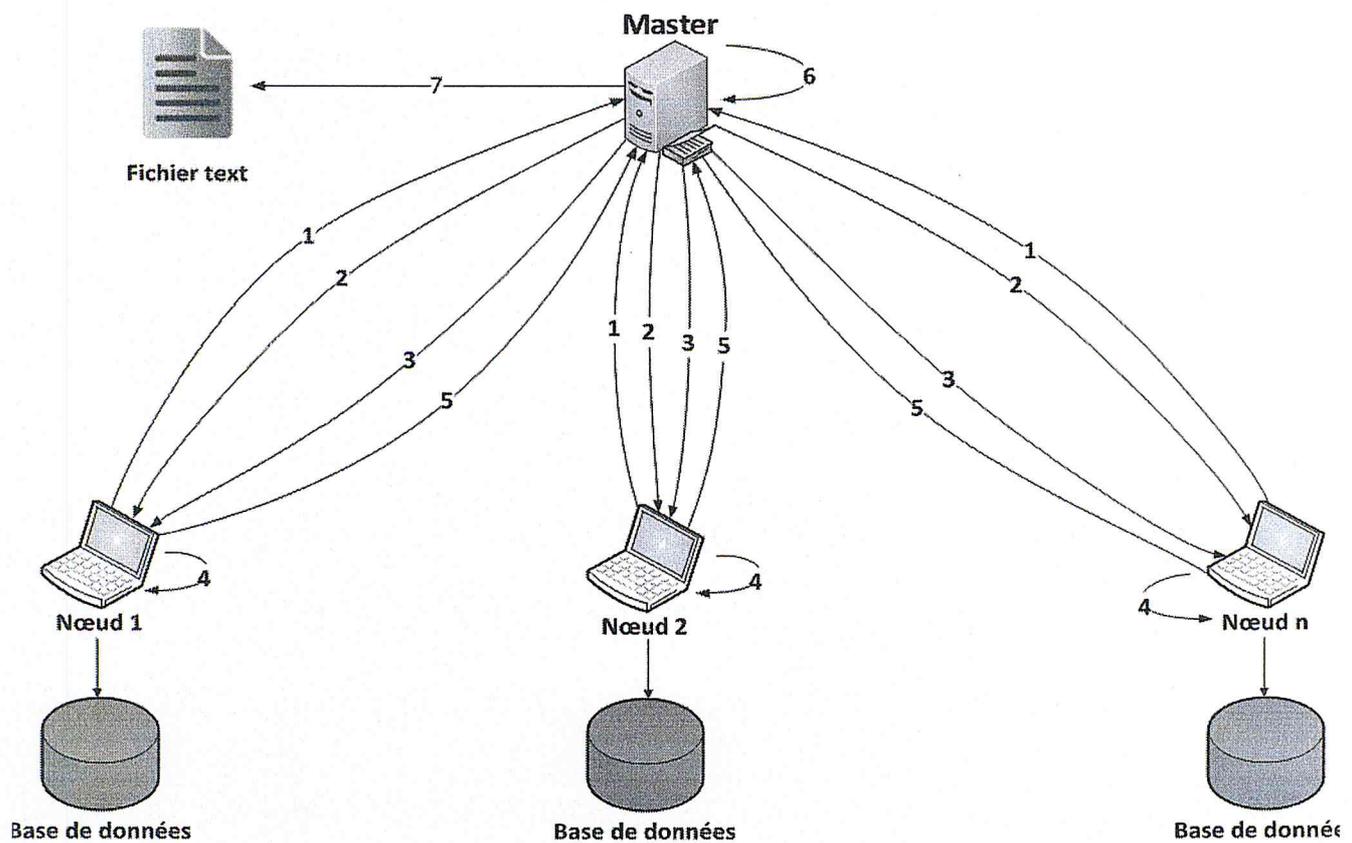


Figure 22 : Schéma technique de l'application du MapReduce.

- 1- **première étape** : dans cette étape le master reste en écoute, chaque nœud se connecte au master lui envoie son adresse Ip.
- 2- **deuxième étape** : dans cette étape le master récupère les adresses Ip des nœuds et se connecte aussi à ces derniers
- 3- **troisième étape** : Master choisit un des besoins, et envoi le numéro de la méthode correspondante aux nœuds pour qu'ils lancent le traitement
- 4- **quatrième étape** : chaque nœud récupère les données à partir de sa base mongo DB et exécute la méthode Map sur ces données.
- 5- **cinquième étape** : des qu'un nœud achève l'étape Map, envoi un fichier au master avec le résultat de cette étape, et il se met en écoute pour une nouvelle demande.

- 6- **sixième étape** : le master attend que tous les nœuds envoient le résultat du Map pour qu'il exécute à son tour la fonction Reduce.
- 7- **septième étape** : le master écrit le résultat dans un fichier texte, et il reste en écoute d'une nouvelle demande.

9) Conception Du Tableau de bord :

Dans cette phase nous allons décrire les différents cas d'utilisation du tableau de bord.

9.1) Diagrammes des cas d'utilisation et de séquence :

9.1.1) Tableau de Bord :

Dans le diagramme des cas d'utilisation du tableau de bord tout les cas d'utilisation sont illustrés, en représentant les trois fonctionnalités (vente pièces de rechange, vente véhicules neufs et l'après vente), dont chacune sera détaillée dans son diagramme.

Le système attribue les droits d'accès selon le profil utilisateurs comme suit :

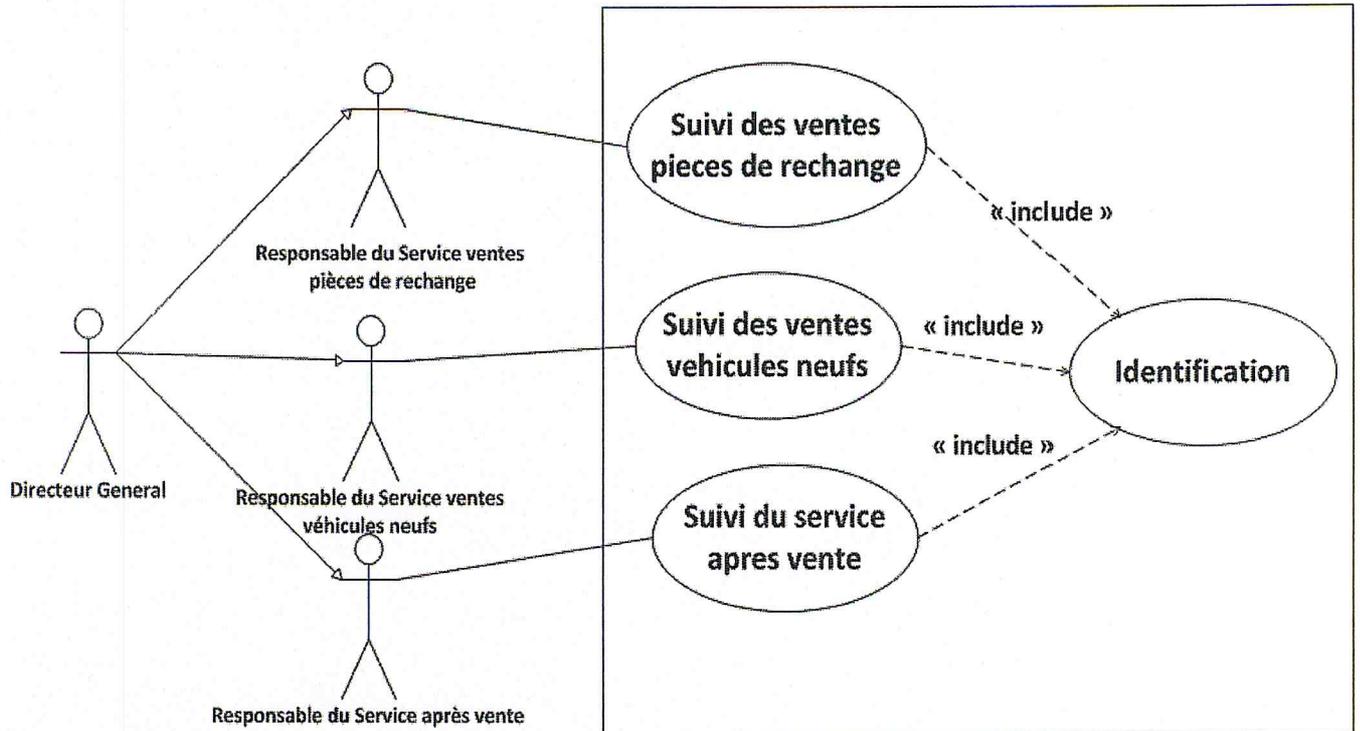


Diagramme 1 : Diagramme de cas d'utilisation du Tableau de Bord.

9.1.2) Identification:

Tous les utilisateurs peuvent se connecter au système, chacun s'identifie par son nom de compte utilisateur et son mot de passe.

S'il n'y a pas d'erreur la connexion au système s'effectue.

9.1.3) Cas normal :

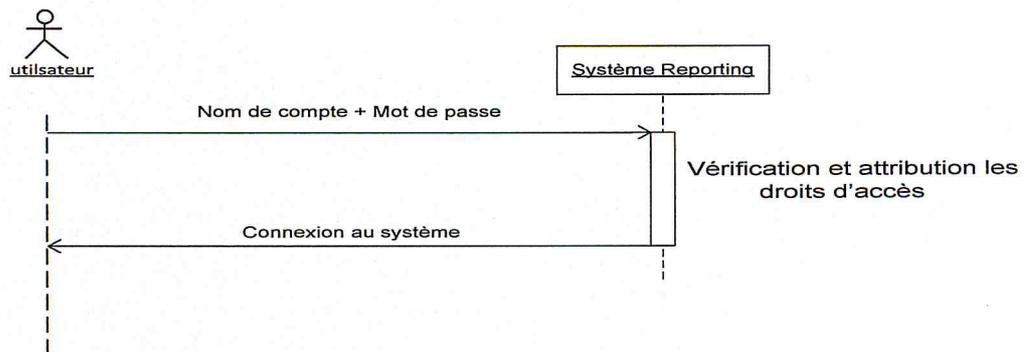


Diagramme 2 : Diagramme de séquence du processus de connexion au système (cas normal).

9.1.4) Cas d'erreur :

Dans ce cas l'utilisateur souhaite accéder par un compte inexistant.

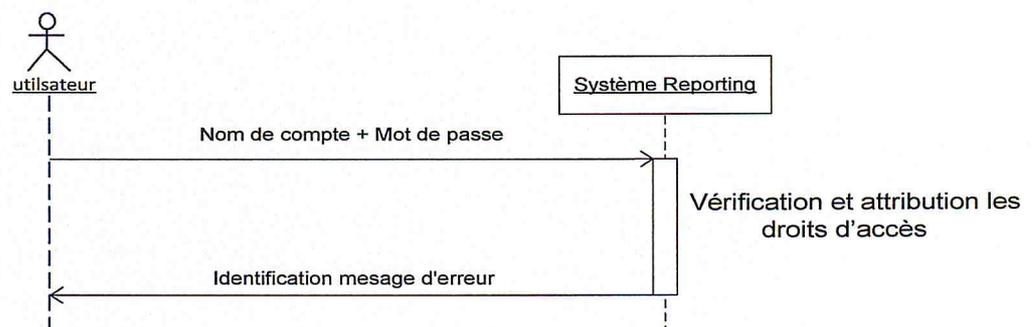


Diagramme 3 : Diagramme de séquence : Connexion au système (cas d'erreur).

9.1.5) Suivi des Ventes Pièces de rechange:

Un utilisateur une fois connecté, concernant son profil le système attribue les droits d'accès aux fonctionnalités offertes.

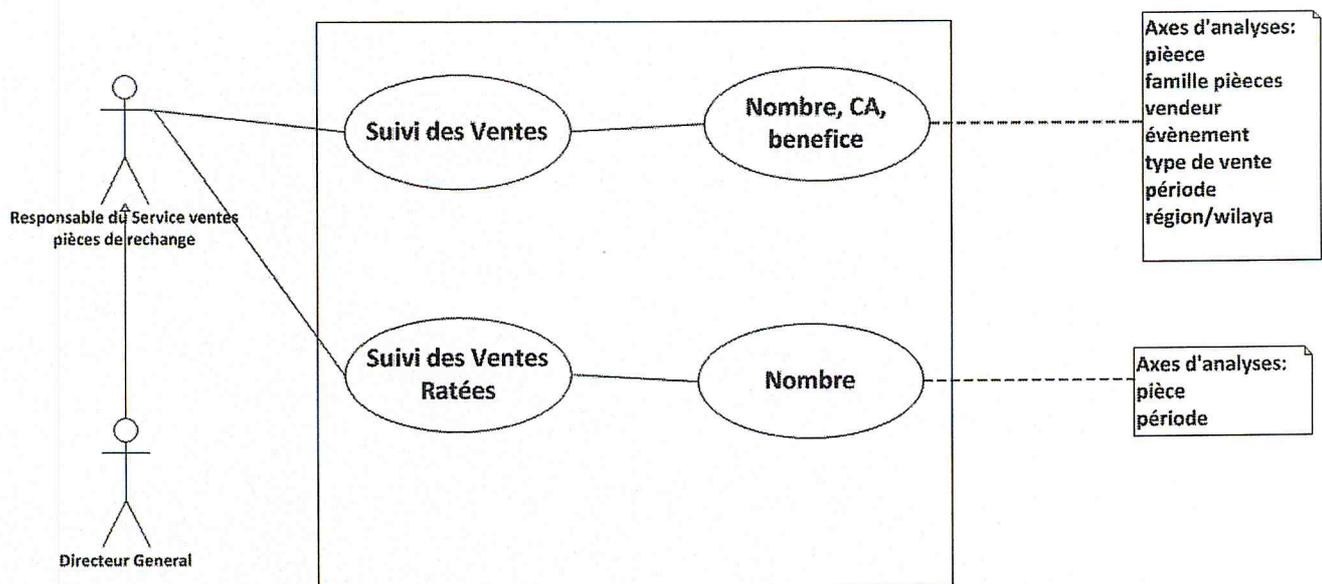


Diagramme 4 : Diagramme de cas d'utilisation du suivi des ventes pièces de rechange.

Description :

Le responsable ainsi que le directeur général ont le droit de consulter les rapports, tableaux de bord des ventes pièces de rechanges (Nombre de vente, ca, bénéfice) ainsi que les rapports concernant les ventes ratées (nombre).

9.1.6) Suivi des Ventes Véhicules neufs:

Le système offre la fonctionnalité du suivi des ventes véhicules neufs au profil 'Responsable des ventes véhicules neufs' et aussi au 'Directeur général', comme suit :

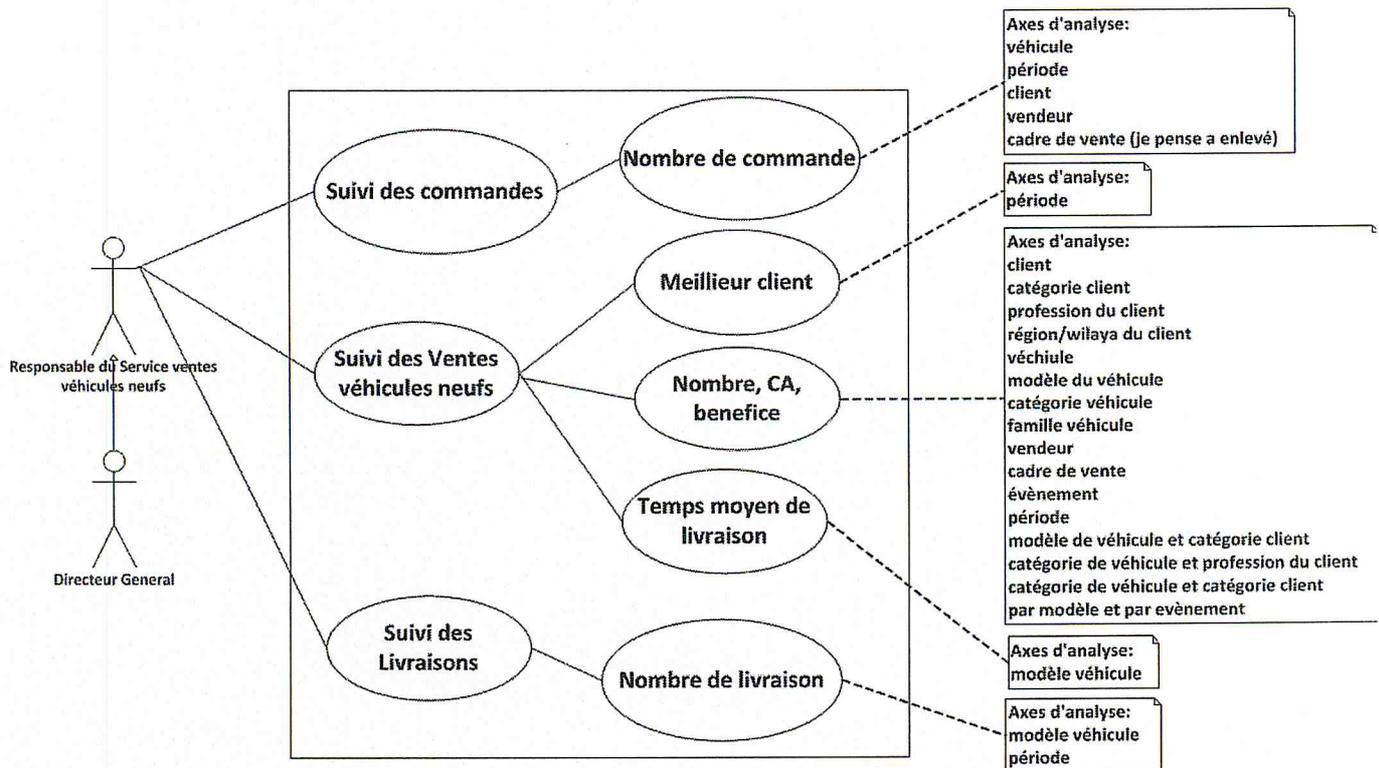


Diagramme 5 : Diagramme de cas d'utilisation du suivi des ventes véhicules neufs.

Description :

Le responsable des ventes véhicules neufs ainsi que le directeur général ont le droit de consulter les rapports, tableaux de bord des ventes véhicules neufs (Nombre de vente, ca, bénéfice) ainsi que les rapports concernant les commandes (nombre de commande), livraison (nombre de livraison, temps moyen de livraison).

9.1.7) Suivi du service après vente:

Le système offre la fonctionnalité du suivi du service après vente au profil 'Responsable du service après vente' et aussi au 'Directeur général', comme suit :

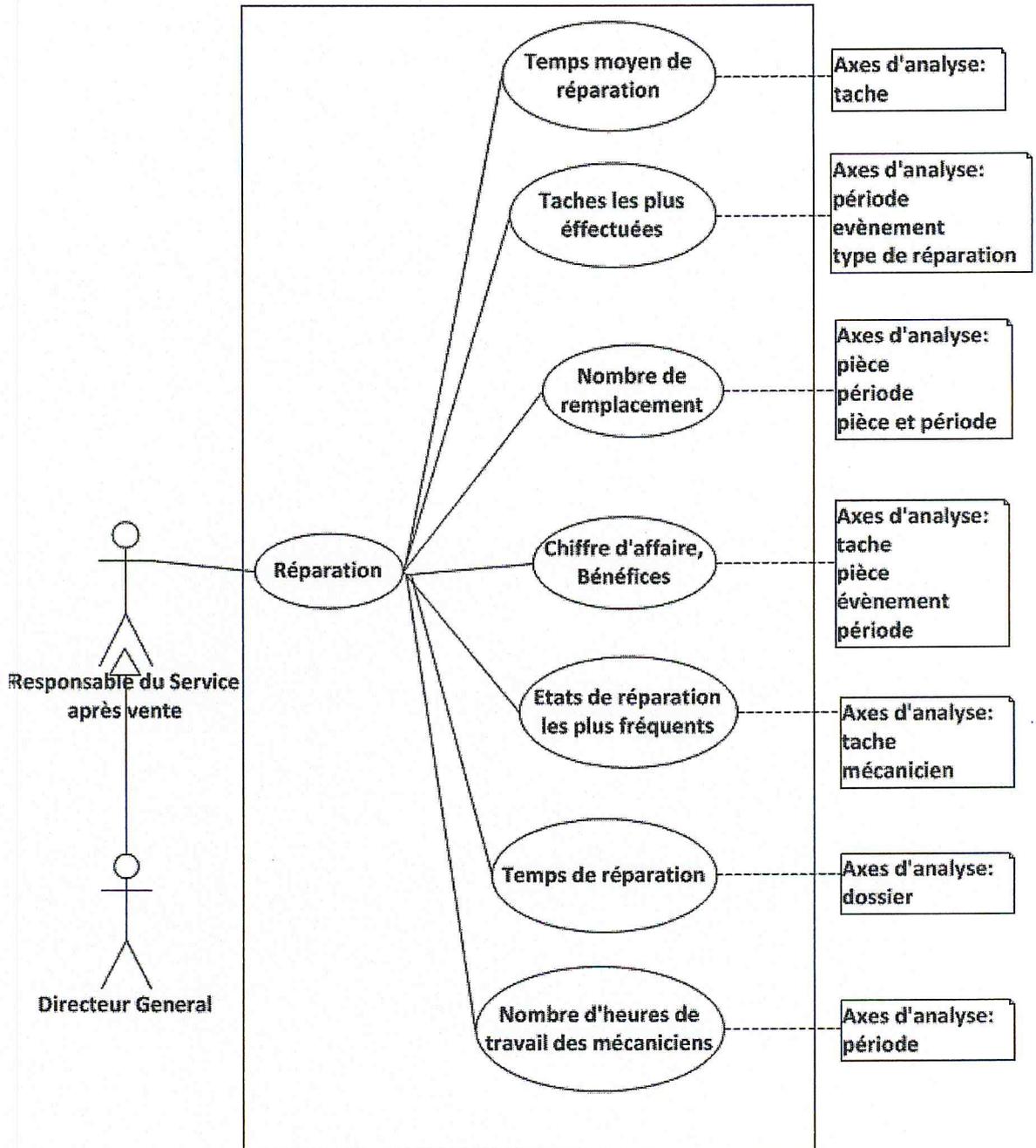


Diagramme 6 : Diagramme de cas d'utilisation du suivi service après vente.

Description :

Le responsable du service après vente ainsi que le directeur général ont le droit de consulter les rapports, tableaux de bord des réparations (Nombre de réparations, chiffre d'affaire, bénéfice, temps moyen de réparation, les tâches les plus effectuées, les pièces les plus rechangées, Etats de réparation les plus fréquents).

9.1.8) Diagrammes de séquence :

Voici le diagramme de séquence pour la création d'un tableau de bord à partir d'un fichier.

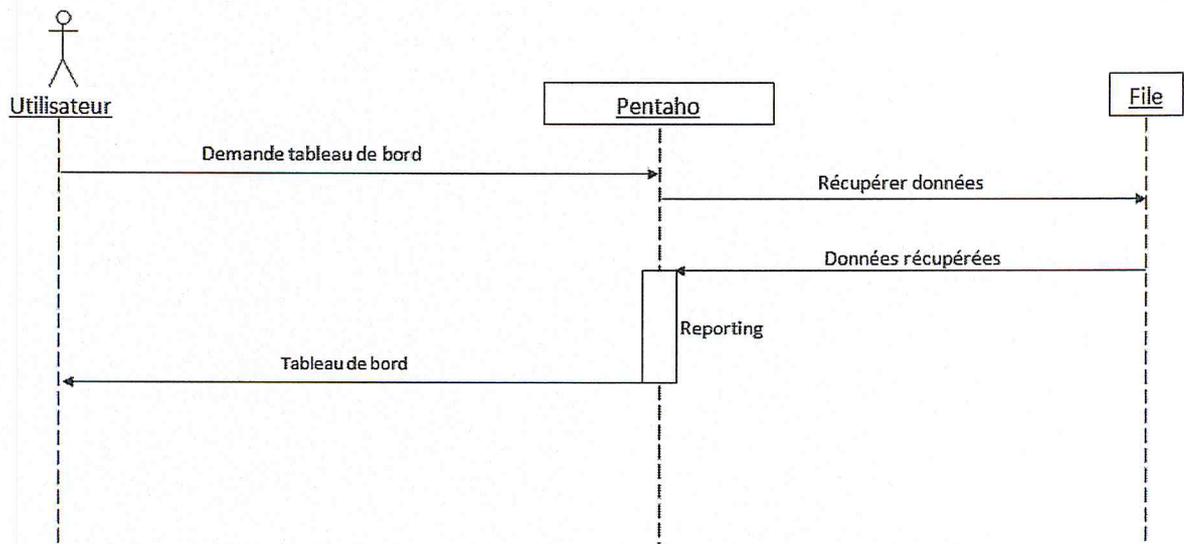


Diagramme 7 : Diagramme de séquence création d'un tableau de bord.

Voici le diagramme de séquence pour le remplissage du fichier avec MapReduce

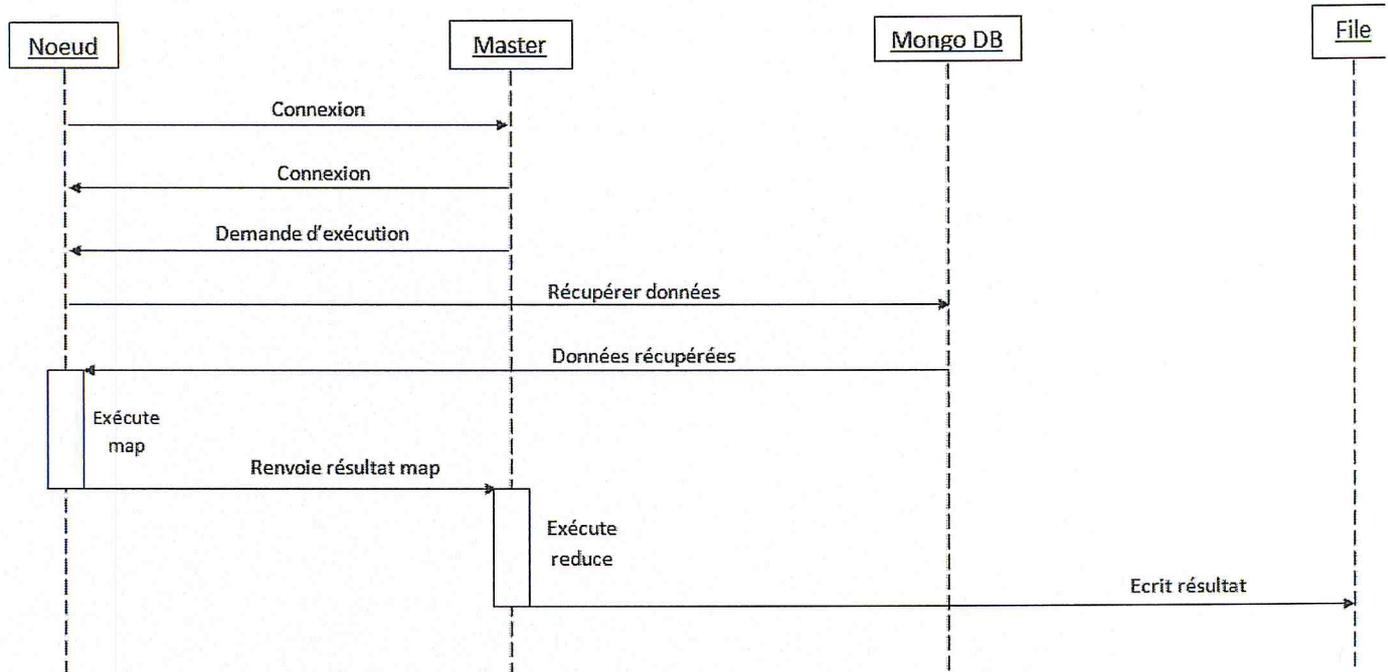


Diagramme 8 : Diagramme de séquence remplissage fichier avec MapReduce.

Voici le diagramme de séquence pour le remplissage du fichier avec Pentaho Data Integration.

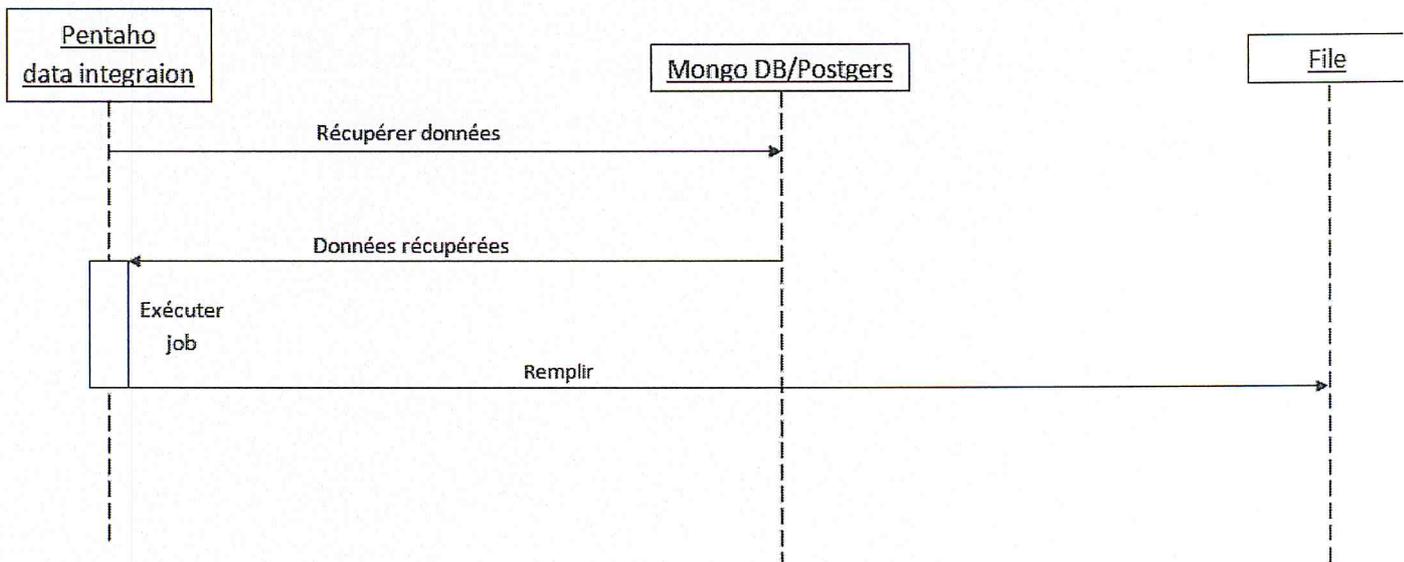


Diagramme 9 : Diagramme de séquence remplissage du fichier avec Pentaho Data Integration.

Conclusion :

Dans ce chapitre, nous avons présenté la modélisation dimensionnelle de notre entrepôt de données, cette modélisation repose sur un schéma multidimensionnel qui intègre les concepts de faits et de dimensions

Nous avons aussi présenté l'architecture de notre système décisionnel ainsi que la conception de chaque composant (ETL, Data Warehouse, Tableau de bord), ainsi que la logique d'utilisation du MapReduce dans notre projet.

Dans le chapitre suivant nous allons présenter le détail de la mise en œuvre de notre système, ainsi que les outils qui l'implémentent.

PARTIE 3 : MISE EN ŒUVRE

**CHAPITRE 6 : CHOIX TECHNOLOGIQUES ET ARCHITECTURE
PHYSIQUE**

**CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU
SYSTEME DECISIONNEL**

PARTIE 3 : MISE EN OEUVRE

Dans cette partie nous allons entamer la mise en œuvre des trois éléments composant l'architecture du futur Système décisionnel de SID AUTO. Avant cela nous commencerons par présenter les choix technologiques dans le chapitre 6, ensuite on entamera la mise en œuvre du système dans le chapitre 7 ou nous clôturons cette partie du document par une démonstration du système.

CHAPITRE 6 : CHOIX TECHNOLOGIQUES ET ARCHITECTURE PHYSIQUE

Introduction :

Dans ce chapitre, nous allons présenter les outils de développement et solutions choisies pour implémenter les éléments du Système décisionnel, ensuite nous allons présenter l'architecture physique adoptée pour déployer le système.

1) OUTILS DE DEVELOPPEMENT :

Pour le choix des outils et solutions de développement, notre choix s'est porté sur les solutions Open source, car celles-ci sont gratuites ou à faible coût, libre de redistribution, et bénéficient d'un support communautaires et Professionnel garanti.

1.1) MongoDB :

Nous avons utilisé MongoDB pour notre Data Warehouse NoSQL, qui Est un système de stockage de données NoSQL, orientée documents, Open source (libre), montant bien en puissance, à performance raisonnable, ne nécessitant pas de schéma prédéfini des données, écrit avec le langage de programmation C++. Il fait partie de la mouvance NoSQL et vise à fournir des fonctionnalités avancées.

Parmi les caractéristiques démarquant MongoDB des bases de données relationnelles, on retiendra :

- Le concept de « schéma » n'existe pas. Chaque document est libre de suivre sa propre structure.

- Jointures : sont les données généralement embarquées dans le même « document ». Ou bien, la possibilité de *linkage* entre documents.
- L'atomicité des transactions n'est garantie que sur un seul document.
- La modification concurrente doit être gérée au niveau de l'application.
- La flexibilité, la performance et la scalabilité se font au détriment de la capacité de gérer des transactions complexes.
- Les notions de clé étrangère et clé primaire n'existent pas, il y a un identifiant pour chaque document qui est attribué par MongoDB automatiquement. [Katia Aresti, 2010]

1.2) JMongoBrowser :

JMongoBrowser est un outil pour administrer et gérer les instances de MongoDB et leurs bases de données.

1.3) PostgreSQL 9.1 :

PostgreSQL est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES. Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley.

PostgreSQL est un descendant OpenSource du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes : requêtes complexes, clés étrangères, déclencheurs (triggers), vues, intégrité des transactions, contrôle des accès concurrents. [Web 2]

Nous avons utilisé PostgreSQL pour implémenter le Data Warehouse classique.

1.4) Talend:

Talend Open Studio est un ETL open source, développé par la société française Talend. Il permet de créer graphiquement des processus de manipulation et de transformation de données puis de générer l'exécutable correspondant sous forme de programme Java. [Smile, 2011].

1.5) Pentaho Data Integration (Kettle) :

KETTLE a été développé il y a 5 ans par Matt CASTERS, un consultant en Business Intelligence (BI) indépendant, dans un premier temps pour ses propres besoins.

Le projet a été rendu open Source l'année suivante et PENTAHO l'a acquis au début de l'année 2006. C'est ainsi que KETTLE est devenu Pentaho Data Integration (PDI), qui est un E.T.T.L, C'est-à-dire qu'il permet :

L'Extraction des données depuis divers source (fichiers, bases de données)

Le Transport des données d'une unité de stockage à une autre

La Transformation des données

Le chargement (Loading en anglais) des données dans un entrepôt

[Samatar HASSAN, 2006]

Nous avons utilisé Pentaho Data Integration car il permet de faire des jointures complexes de plusieurs sources de données, Il nous permet aussi de transformer les données textuel (document mongo) en données tabulaire qui seront des sources pour Pentaho lors de la création du Tableau de Bord.

1.6) Pentaho Buisness Analytics :

Pentaho est une plate-forme décisionnelle open source complète possédant les caractéristiques

Suivantes :

- Une couverture complète des fonctionnalités de la Business Intelligence :
 - ✓ ETL (intégration de données)
 - ✓ Reporting
 - ✓ Tableaux de bords (Dashboards)
 - ✓ Analyse ad-hoc (requêtes à la demande)
 - ✓ Analyse multidimensionnelle (OLAP)
 - ✓ Data Mining
- Le serveur web Pentaho comporte également une plate-forme d'administration (*Pentaho Administration Console*) permettant la gestion

des droits d'accès, la planification d'évènements, la gestion centralisée des sources de données.

Dans notre cas nous l'avons utilisé pour la création de notre Tableau de Bord et la gestion des droits d'accès. [Web 3]

2) Architecture physique du système :

Pour avoir une architecture robuste, modulable, et évolutive, L'architecture adoptée pour la réalisation et le déploiement du système est l'architecture web 3 tiers, Cette architecture permet de séparer les différents types de traitement de l'application (Accès aux données, Métier et Présentation)

- **La couche présentation (ou affichage) :**

Associée au client, elle est dite (légère) dans la mesure où le client n'assume aucune fonction de traitement à la différence du modèle 2-tiers. Celle-ci est représentée par un navigateur web.

- **La couche métier (fonctionnelle) :**

Est liée au serveur, qui dans de nombreux cas est un serveur web muni d'extensions applicatives. Dans notre cas c'est le BI Server qui héberge Pentaho.

- **La couche d'accès aux données :**

Est représenté par la base de données. MongoDB dans notre cas.

Les avantages de l'architecture 3-tiers sont principalement au nombre de trois : [Web 4]

- 1- Les requêtes clients vers le serveur sont d'une plus grande flexibilité, en effet les appels clients ne spécifient que des paramètres et des structures de données pour les valeurs de retour.
- 2- L'utilisateur n'est pas supposé connaître le langage de requête (SQL, JSON), qui ne sera pas implémenté dans la partie client qui ne s'occupe que de fonctions d'affichage. De fait des modifications peuvent être

faites au niveau du Data Warehouse sans que cela est un impacte sur la couche client.

- 3- Plus de flexibilité dans l'allocation des ressources, la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique aux grés des besoins évolutifs au sein de l'entreprise.

La figure suivante illustre l'application de l'architecture 3-tiers :

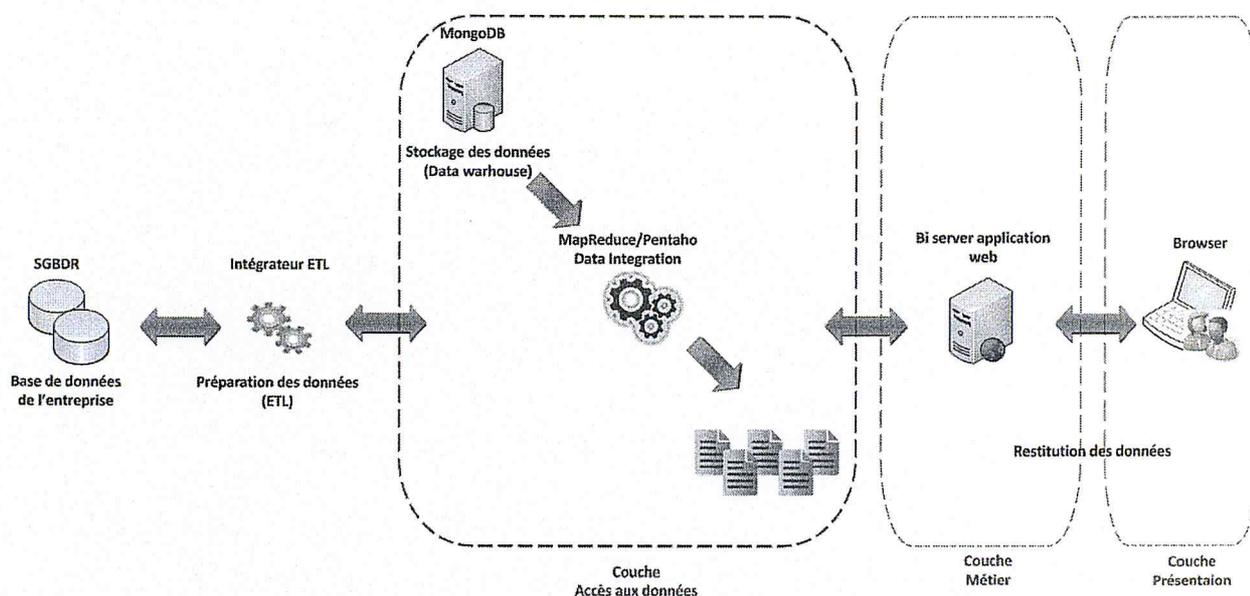


Figure 23 : Architecture physique du system.

Conclusion :

Nous avons vu dans ce chapitre l'ensemble des outils et solutions utilisés pour concevoir et mettre en œuvre le Système décisionnel, avec une architecture 3-tiers.

Maintenant que les choix technologiques sont présentés, on peut entamer la phase de mise en œuvre des éléments de l'architecture du Système décisionnel dans le chapitre suivant.

2) La mise en œuvre de l'alimentation :

L'alimentation de notre Data Warehouse classique ainsi que Data Warehouse NoSQL, se fait en trois étapes qui sont l'extraction de données à partir des bases de données de production, la transformation des données et le chargement des données dans le Data Warehouse.

2.1) Alimentation avec Talend :

Pour l'alimentation de notre Data Warehouse classique Nous avons utilisé Talend.

On crée la connexion avec PostgreSQL, on importe les schémas des bases sources, ensuite on crée un nouveau job, et on fait glisser/déposer pour créer les processus de manipulation de données, en choisit les composants input et output, et le composant tMap qui les relie, ensuite on change les paramètres du tMap et on fait les jointures, les figures suivantes montrent toutes les étapes :

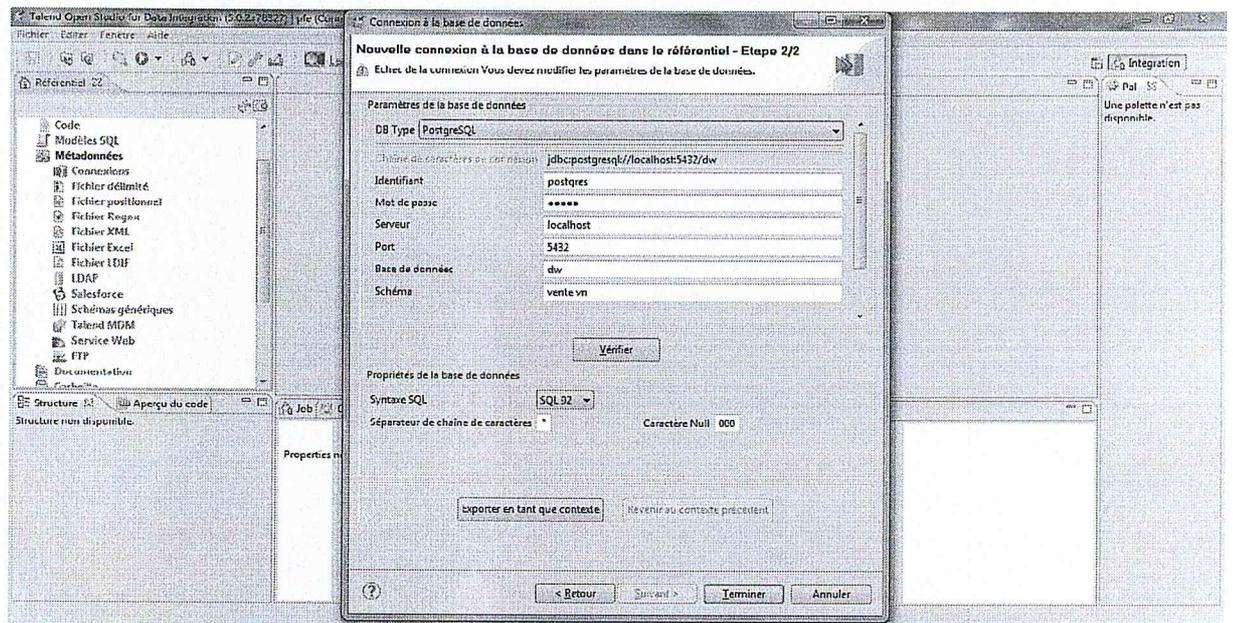


Figure 25 : Création d'une connexion dans Talend.

CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

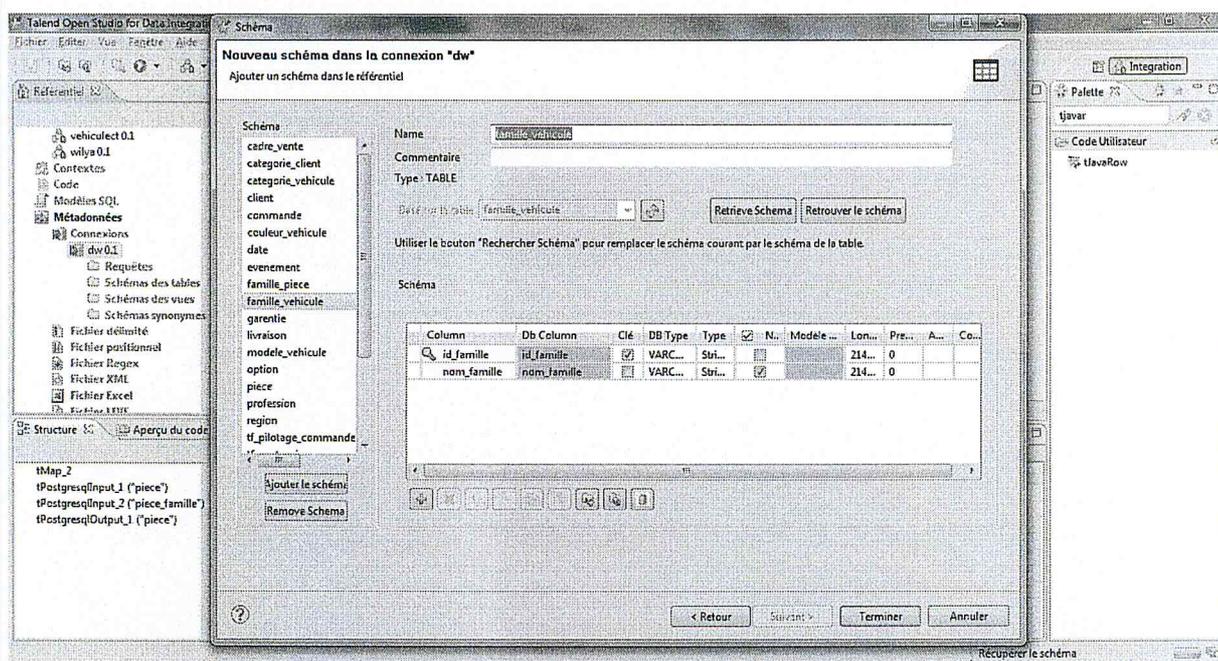


Figure 26 : Récupération d'un schéma de la base source.

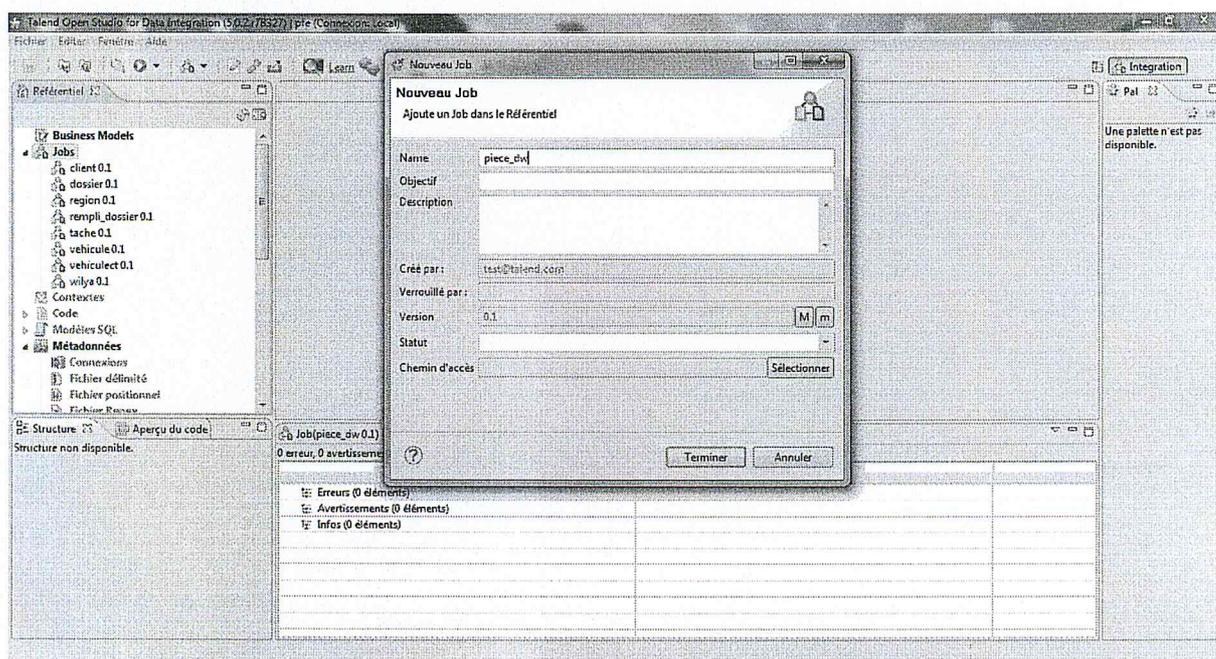


Figure 27 : Création d'un nouveau Job dans Talend.

CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

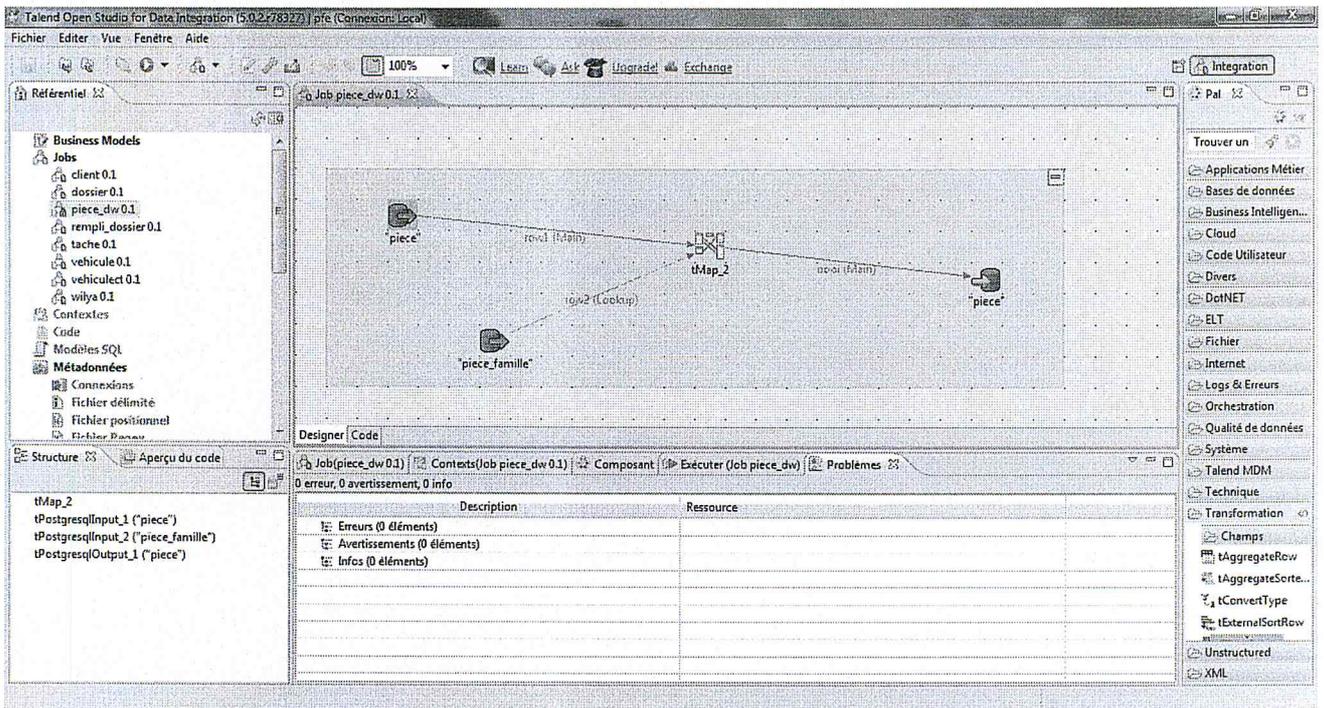


Figure 28 : Job ETL piece.

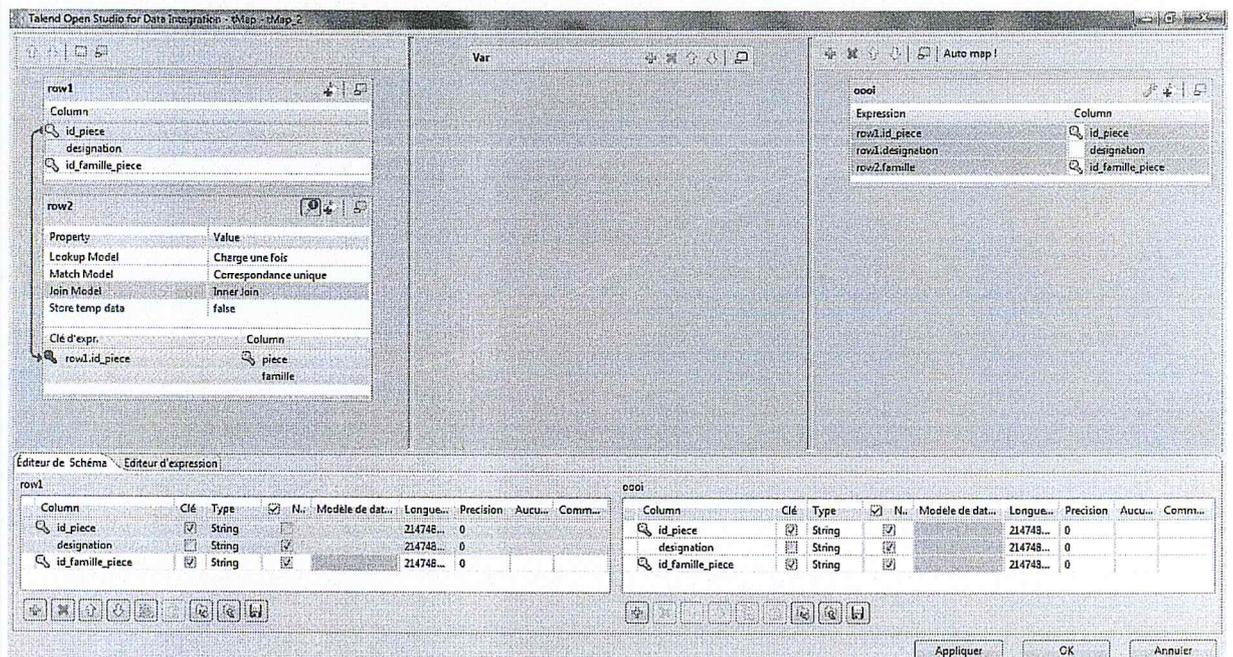


Figure 29 : Configuraton du tMap.

2.2) Alimentation avec 2F Loader :

2F Loader c'est un ETL maison développé en java avec lequel nous avons fait l'extraction à partir des bases sources, la transformation et le chargement des collections du Data Warehouse dans MongoDB.

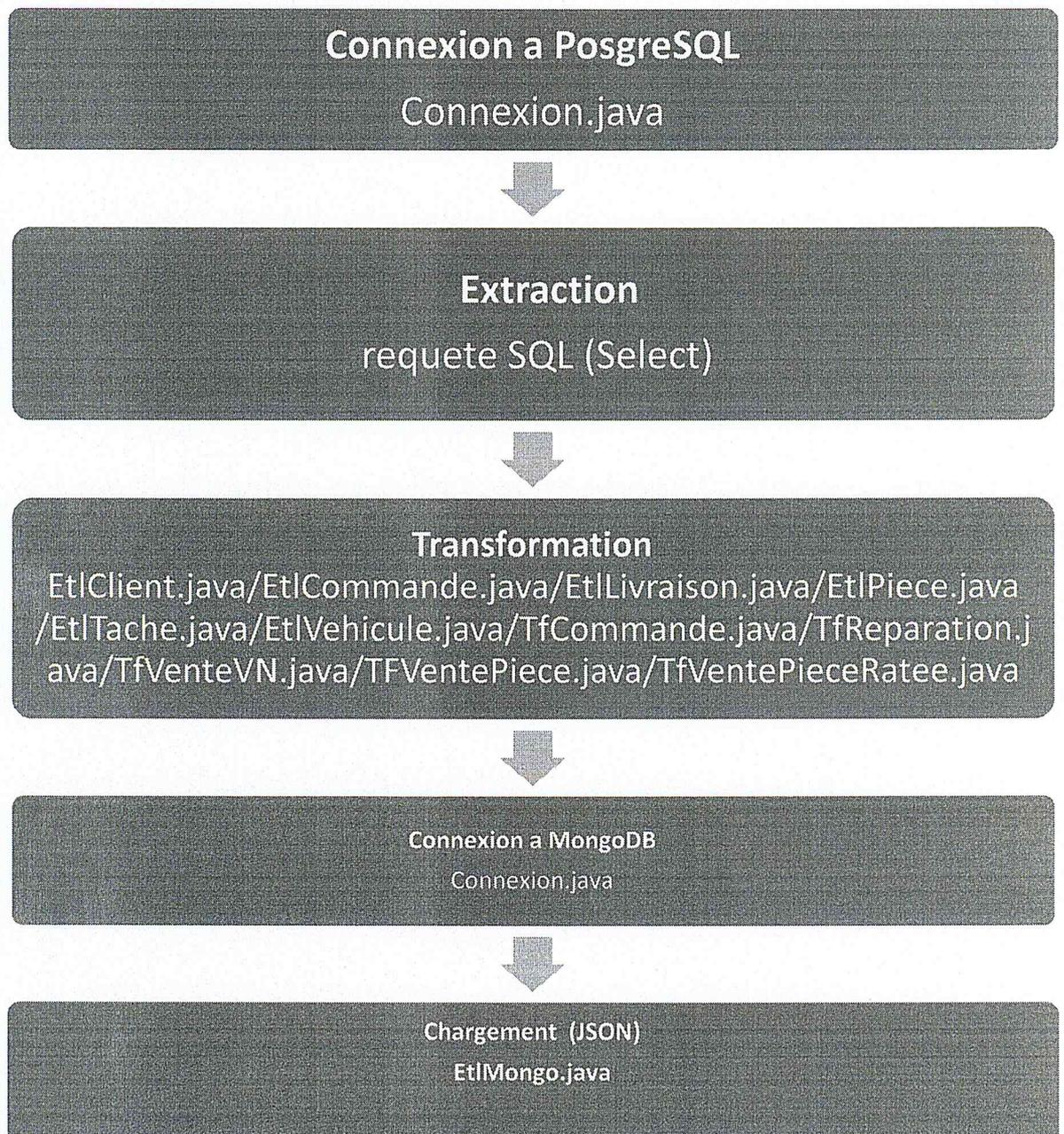


Figure 30 : Processus de 2F_loader.

3) La construction de l'outil de restitution :

La construction du tableau de bord est basée sur la plateforme Pentaho Business Analytics. On a créé nos tableaux de bord à partir de Data Source (sous format CSV) qui sont remplis avec notre programme MapReduce ou avec Pentaho Data Integration.

1ere Phase : construction du fichier texte

Dans cette phase, nous avons préparé les données à analyser avec Pentaho Business Analytics sous format tabulaire, en utilisant MapReduce, et Pentaho Data Integration (Kettle).

- **MapReduce :**

Nous avons utilisé MapReduce pour préparer les mesures calculables de chaque Nœud, ce dernier exécute le programme Map en parallèle et renvoi le résultat au Master, qui récupère les résultats et exécute à son tour le Reduce. Il crée un répertoire pour chaque analyse, et un fichier texte où il écrit le résultat du Reduce. Ce fichier texte sera utilisé par la suite comme Data source pour la création du tableau de bord.

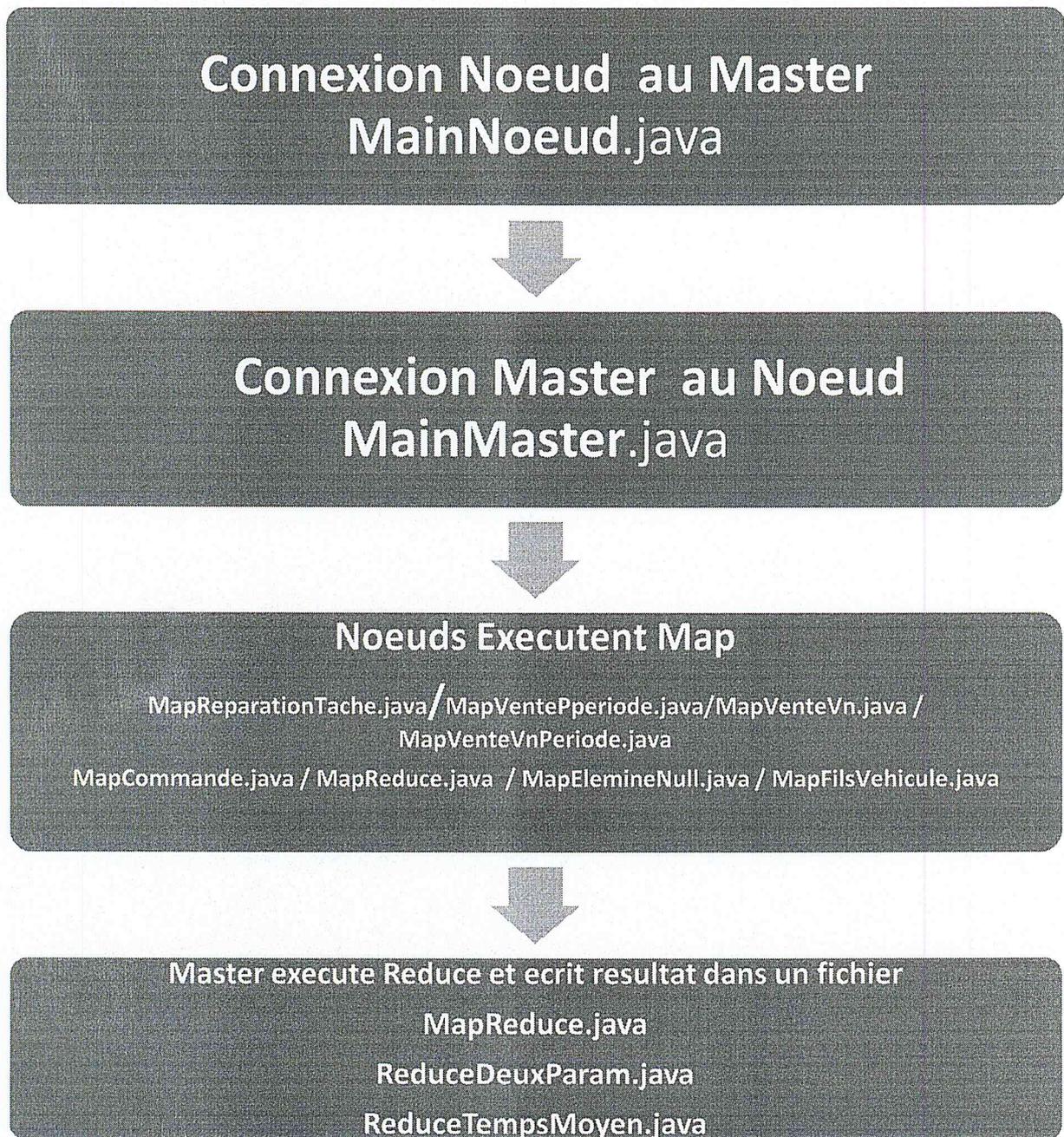


Figure 31 : Processus du programme MapReduce.

- **Pentaho Data Integration :**

Nous avons utilisé Pentaho Data Integration pour la transformation des données textuelles (documents) en données tabulaires (tableau), et pour les jointures complexes.

Pour nouvelle transformation, par exemple, on choisit la fonction **MongoDb Input(source)**, on change les paramètres (serveur, port, la base de donnée, la collection), ensuite on choisit la fonction **Extraction depuis Json** ou on spécifie les champs qu'on veut récupérer ,et enfin on choisit la fonction **alimentation fichier** ou on spécifie (chemin d'enregistrement du fichier, son nom, les champs qu'on veut écrire dans le fichier ainsi que le séparateur), et le fichier pourra être utilisé comme Data source pour la création du tableau de bord, la figure suivante montre un Job dans Pentaho Data Integration.

On peut aussi utiliser plein d'autres fonctions selon le besoin par exemple la fonction **produit cartésien** qui fait la jointure dans le cas ou on a plusieurs input, il ya aussi **extraction depuis chaînes de caractères**, qui fait le SubString on lui donne la position de début et la position de fin elle récupère les caractères entre les deux positions, par exemple si on veut récupérer juste le mois d'une date.

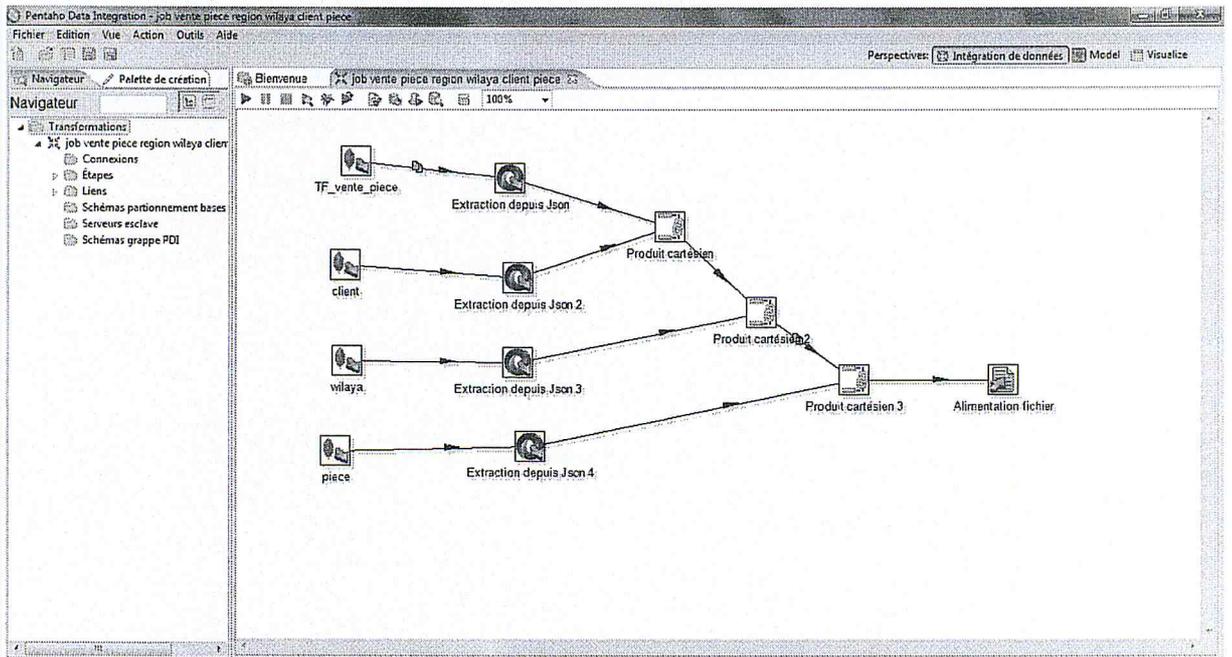


Figure 32 : Job vente pièce, client, wilaya avec Pentaho Data Integration.

2eme Phase : construction du tableau de bord

Pour la création d'un tableau de bord avec Pentaho on fait **new Dashboard**(figure 33), on choisit le Template du tableau de bord(figure 34), on

choisit le type soit Data table, soit chart(**figure35**), ensuite on choisit la Date Source s'il elle existe déjà(**figure 36**), sinon on ajoute une nouvelle source de données de type CSV FILE, en important le fichier texte qui contient les données qui nous intéressent, on donne un nom à la source qui va s'enregistrer et qu'on va choisir par la suite pour notre tableau de bord(**figure 37 ,38**), après le choix de la source de données on choisit les attributs à représenter (**figure 39**), et au final on configure le schéma avec chart designer(**figure 40**), et le tableau de bord est créé on peut nommer chaque zone de Template d'après ce qu'elle contient(**figure 41**).

Pour les rapports, on fait **New Analysis**, on choisit la source si elle existe sinon on ajoute une nouvelle source de la même façon expliquée avec le tableau de bord on enregistre, ensuite on pourra afficher le rapport sur notre tableau de bord.

Les figures suivantes illustrent les étapes de création du tableau de bord.

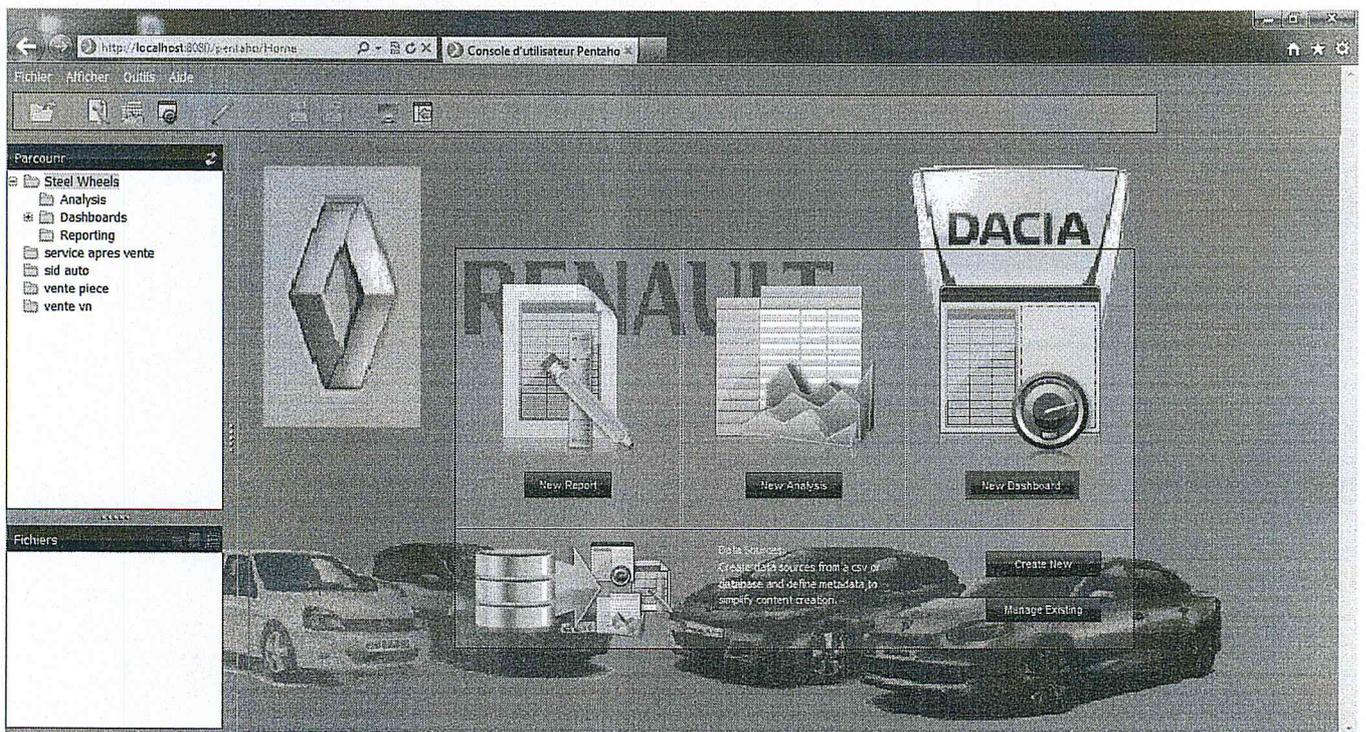


Figure 33 : Création d'un nouveau tableau de bord.

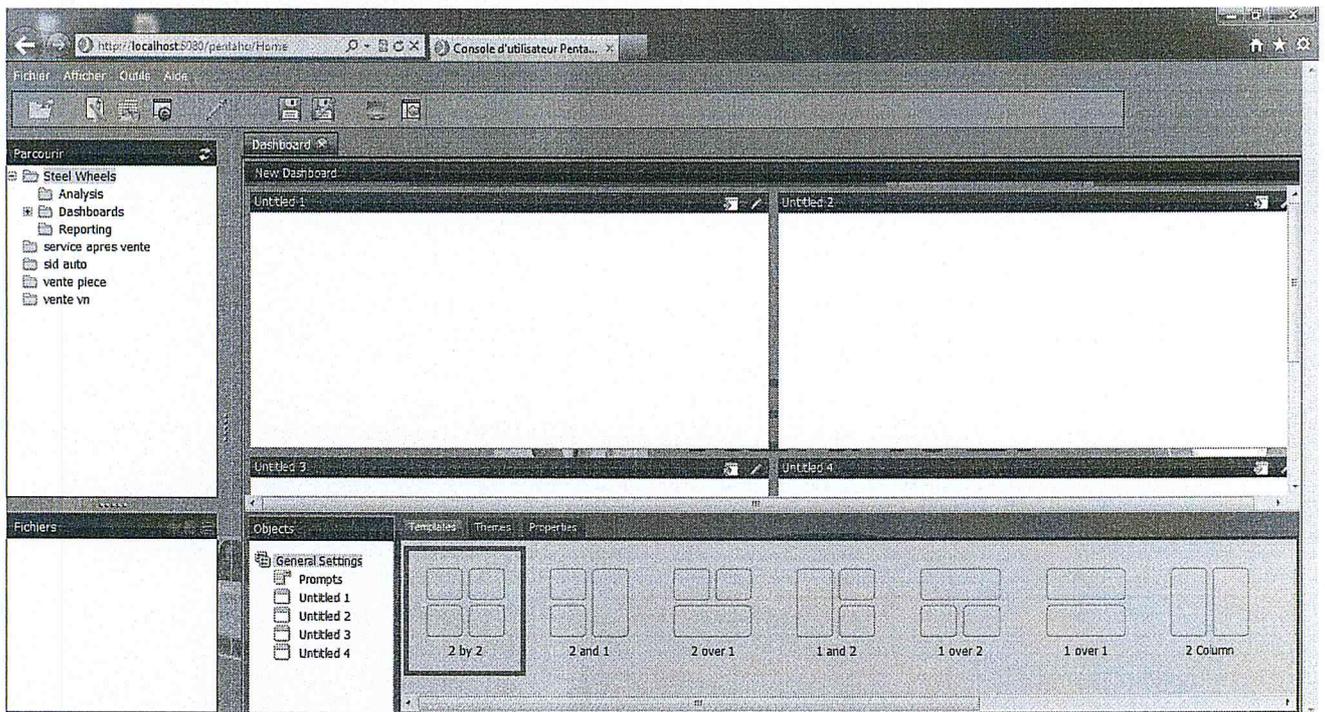


Figure 34 : Choix Template du tableau de bord.

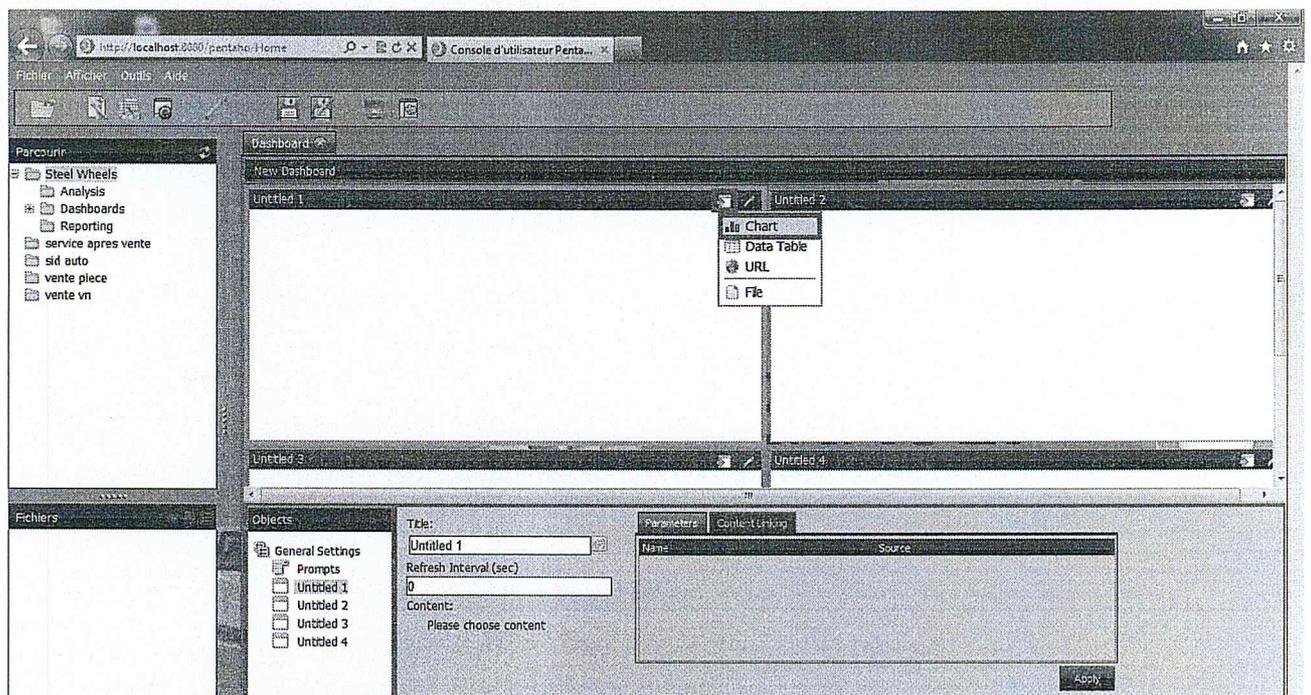


Figure 35 : Choix type tableau de bord.

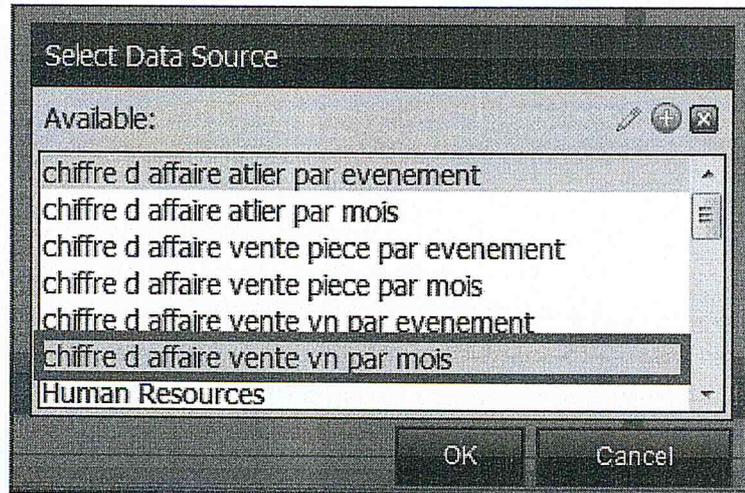


Figure 36 : Sélection de la source de données existante.

Si on veut ajouter une nouvelle source de données, on click sur (+), on choisit le type CSV et on importe le fichier texte, on nomme la nouvelle source

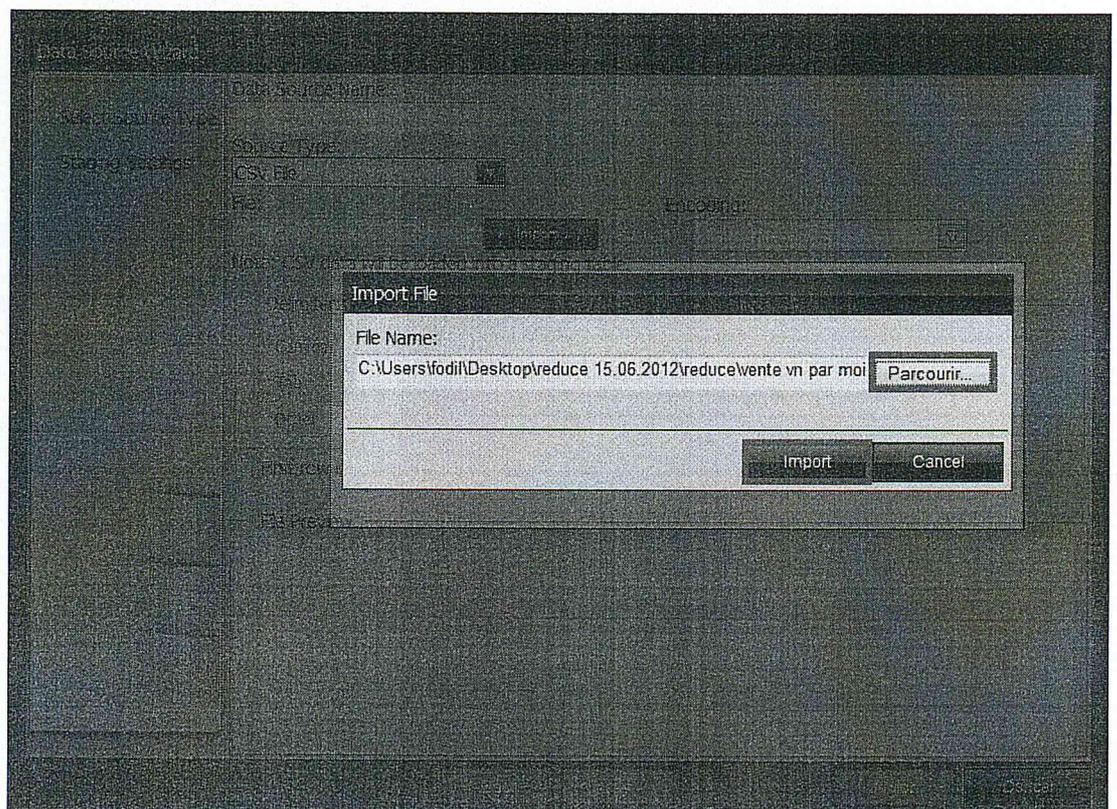


Figure 37 : Importer un fichier texte comme source de données.

CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

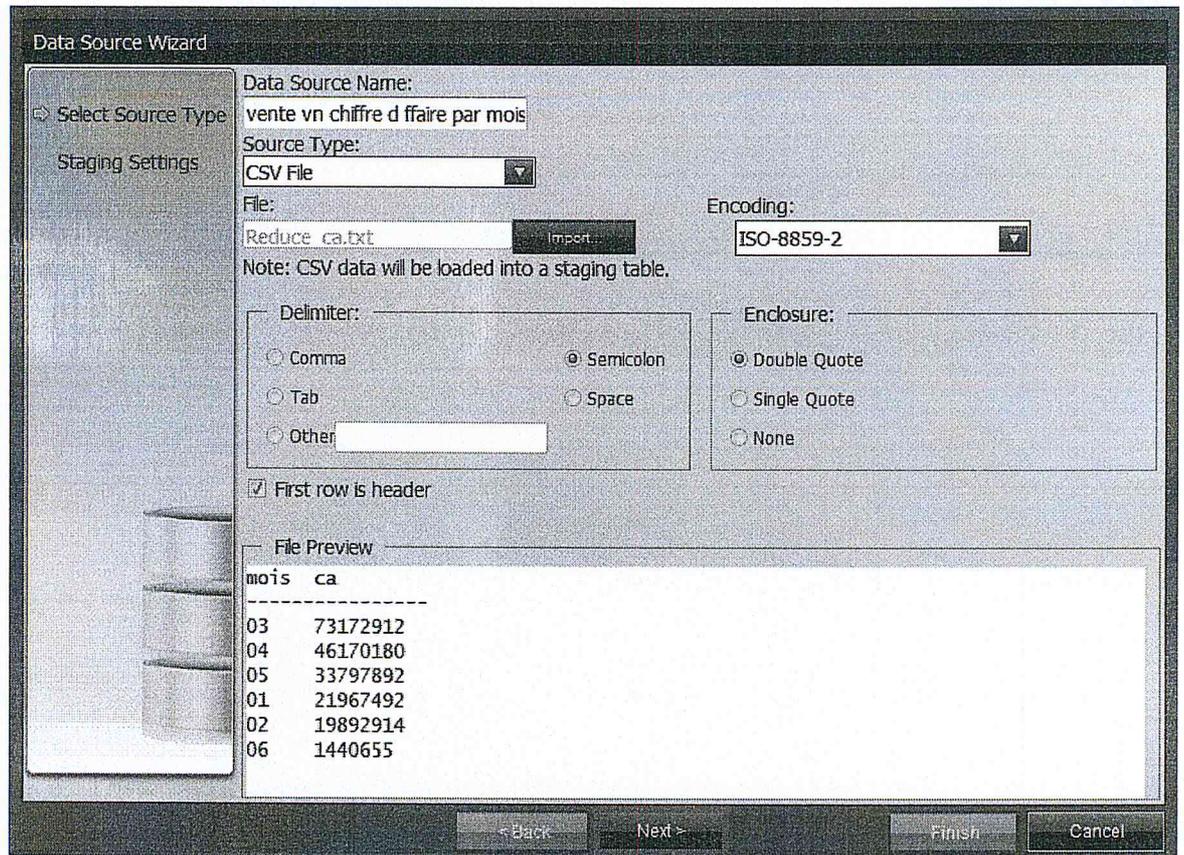


Figure 38 : Création de la source de données.

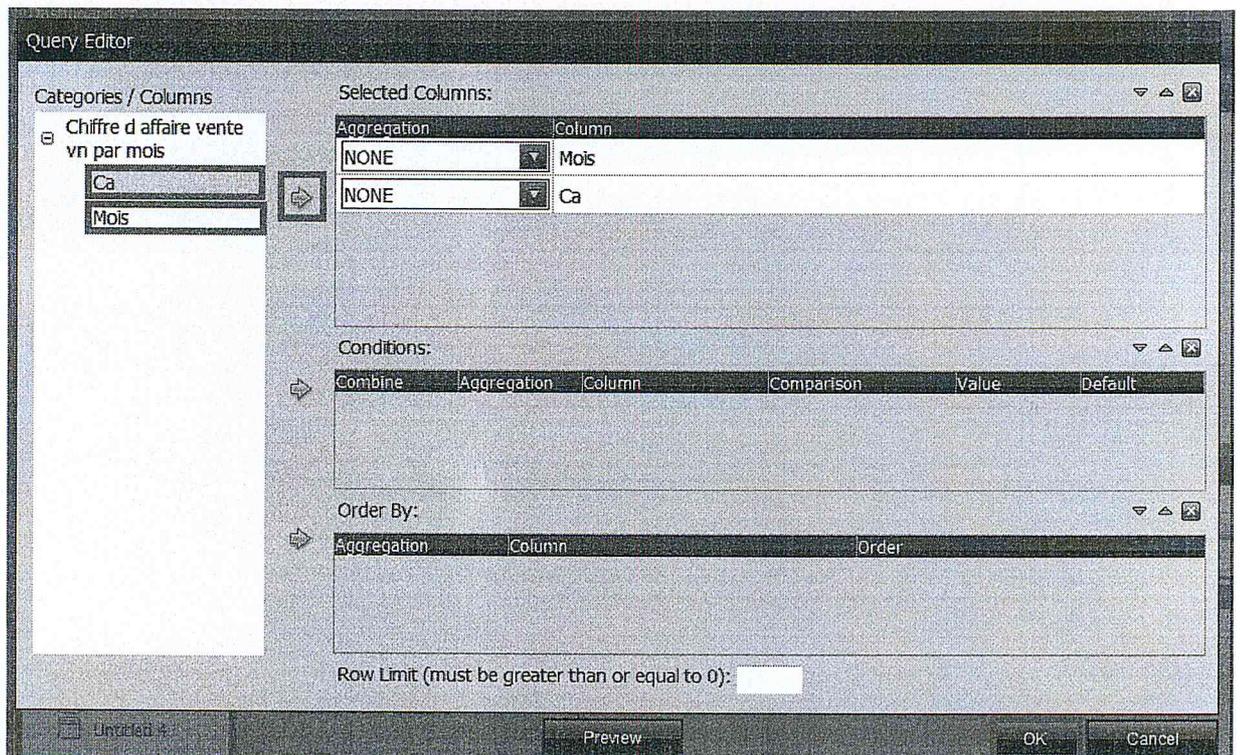


Figure 39 : Choix des attributs.

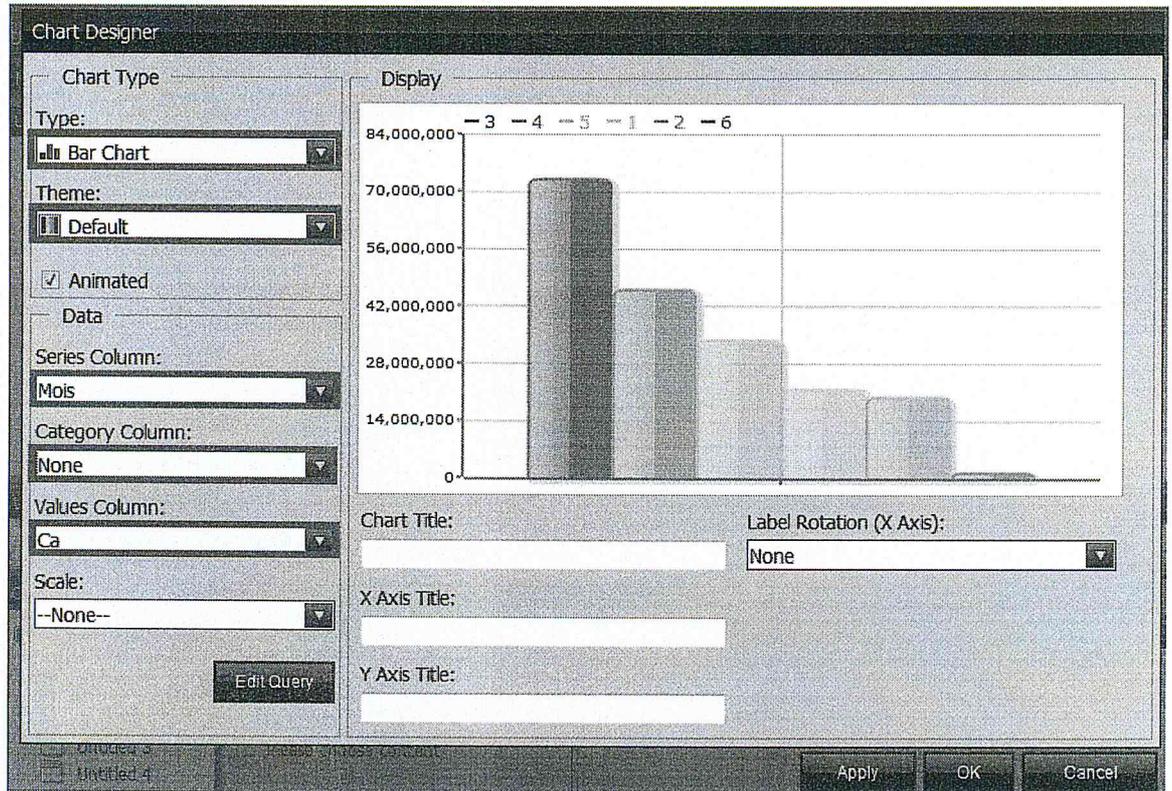


Figure 40 : Configurer chart designer.

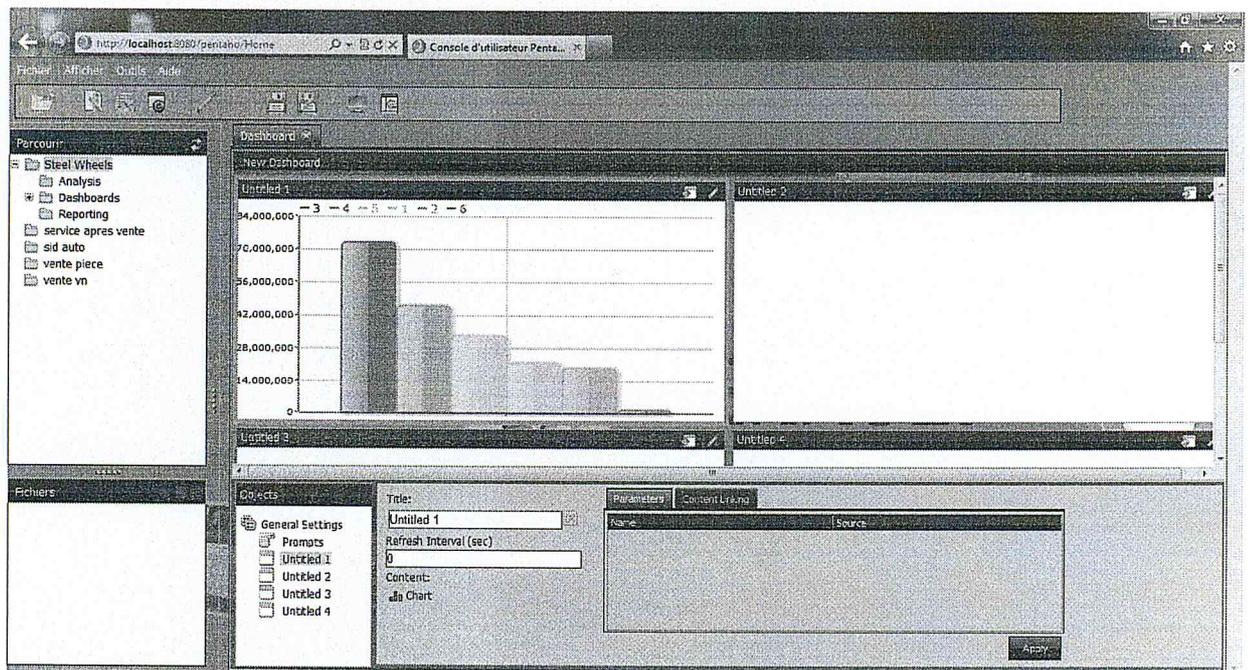


Figure 41 : Tableau de bord créé.

Il y a aussi la possibilité de multiple fonctions, comme ajouter les liens hyper texte qui nous mènent sur un autre tableau de bord ou lors du click affichent d'autres schémas qui détaillent le tableau affiché.

Dans notre cas nous avons créé quatre tableaux de bord :

- Tableau de bord du service vente pièce : Celui-ci contiendra les indicateurs concernant les ventes pièces et les ventes ratées.
- Tableau de bord du service vente Véhicule neuf : Celui-ci contiendra les indicateurs concernant les ventes véhicule neuf, les commandes, et les livraisons.
- Tableau de bord des services après vente : Ce tableau de bord contiendra les indicateurs concernant les réparations, les mécaniciens, et les taches.
- Tableau de bord récapitulatif : Il contiendra un indicateur de chaque service(le plus important) du concessionnaire.

4) Sécurité et habilitations :

Les utilisateurs et les droits d'accès sont gérés par la plateforme Pentaho BI Server, on peut créer des utilisateurs et des rôles, on peut attribuer pour un utilisateur un à plusieurs rôles.

Les utilisateurs principaux des tableaux de bord sont :

- les responsables de chaque service (responsable service vente véhicule neuf, responsable service vente pièce de rechange, responsable du service après vente)
- le directeur général

Nous avons créé avec la console d'administration de Pentaho les différents utilisateurs, en attribuant pour chacun un identifiant et un mot de passe.

CHAPITRE 7: MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

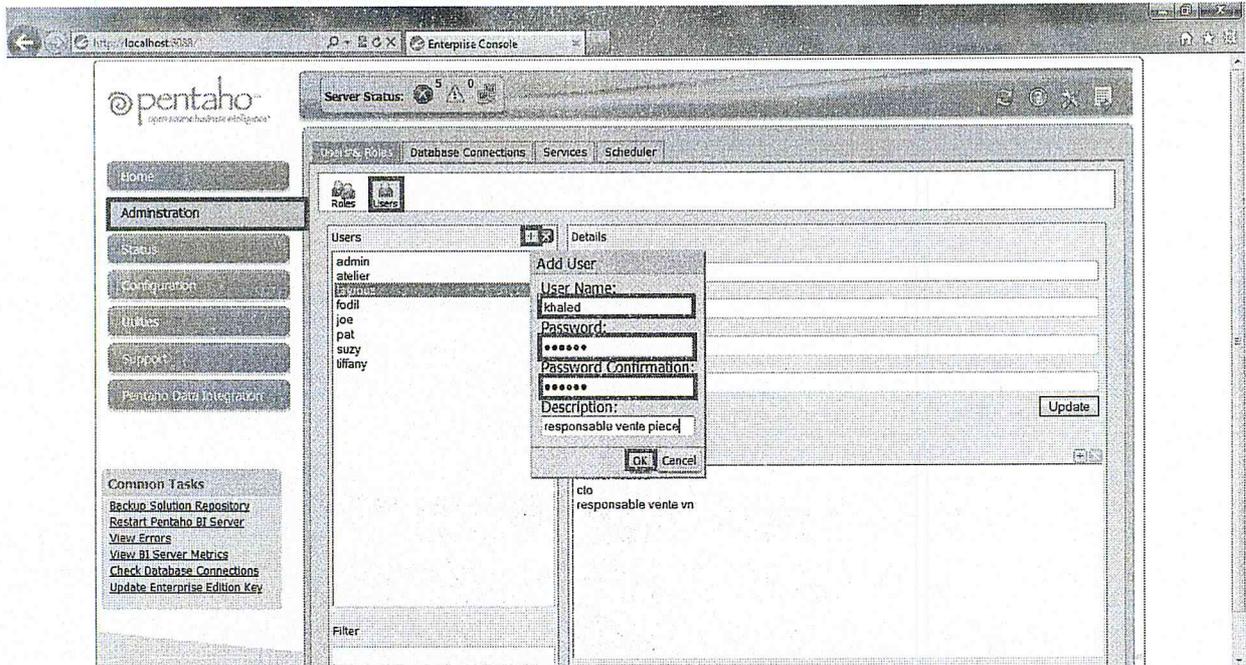


Figure 42 : Création d'un utilisateur Pentaho.

Nous avons aussi créé 3 rôles, un rôle pour chaque tableau de bord, par la suite on a attribué à chaque utilisateur le rôle qui lui est spécifié et on attribue au directeur général les trois rôles.

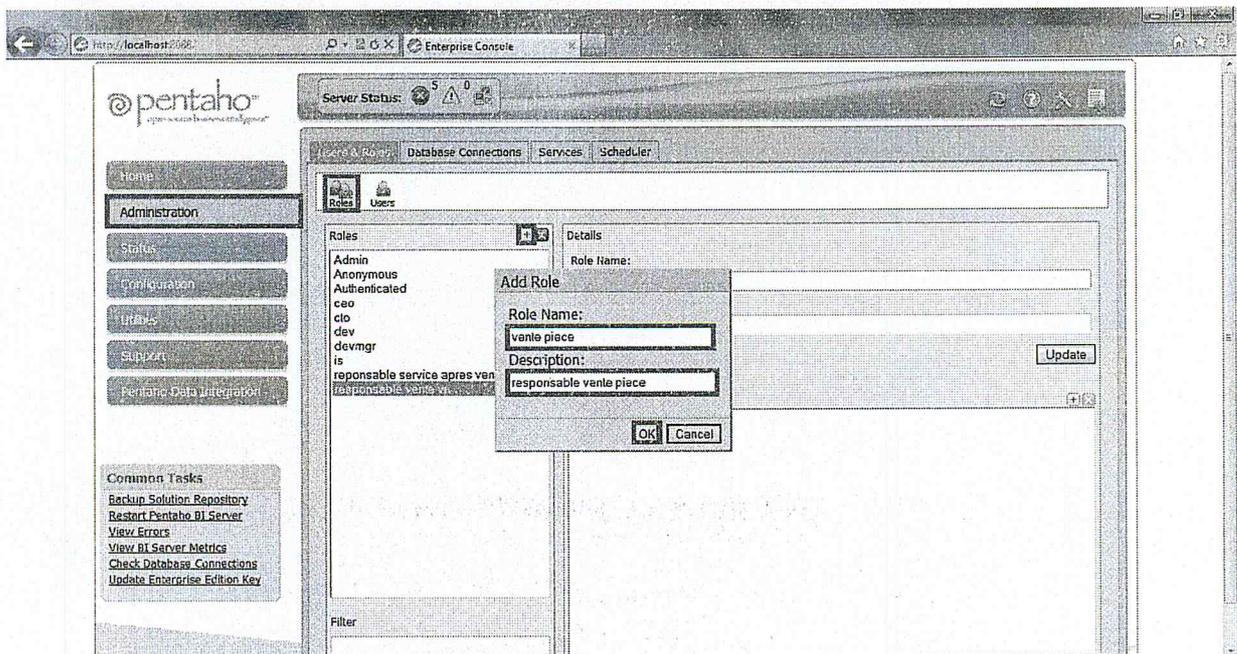


Figure 43 : Création d'un rôle Pentaho.

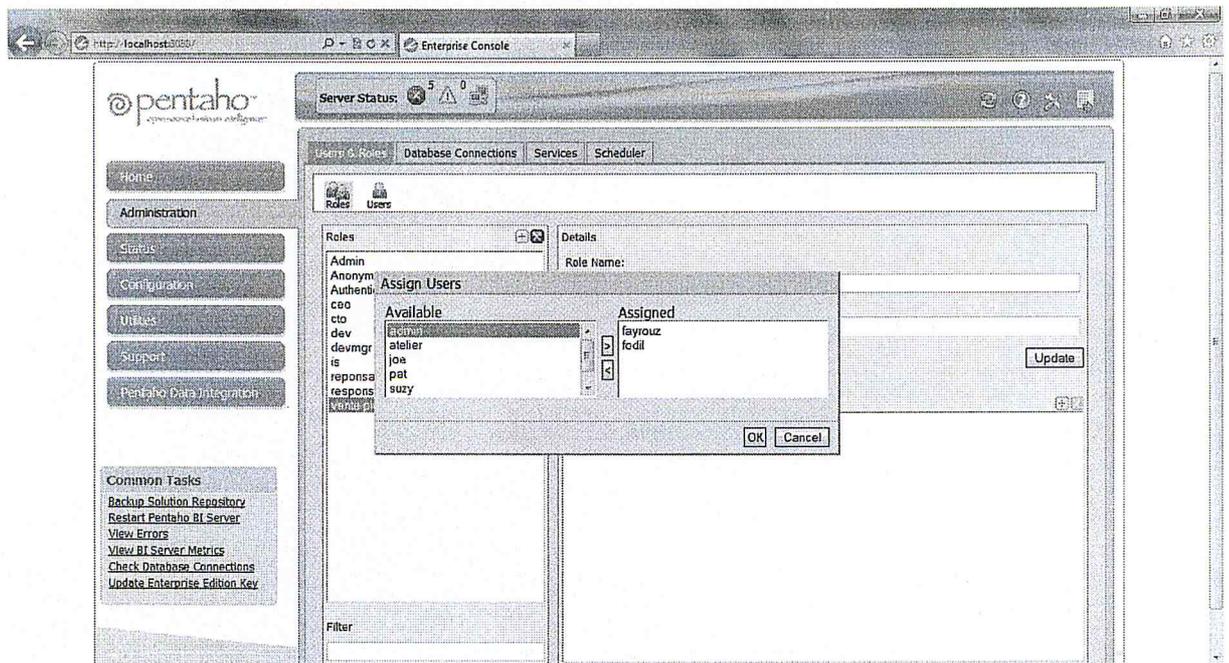


Figure 44 : Attribution des rôles aux utilisateurs.

Chaque utilisateur peut aussi créer un rapport spécifique à lui qu'il peut partager avec d'autres utilisateurs.

5) Démonstration du système :

Nous allons maintenant présenter notre système, ceci en illustrant le déroulement d'un cas d'utilisation classique par un utilisateur.

5.1) Connexion au système :

L'utilisateur qui peut être un décideur entre dans Pentaho par un navigateur internet grâce à un lien prédéfini, il sera invité à s'authentifier avec un nom d'utilisateur et un mot de passe comme le montre la figure suivante :

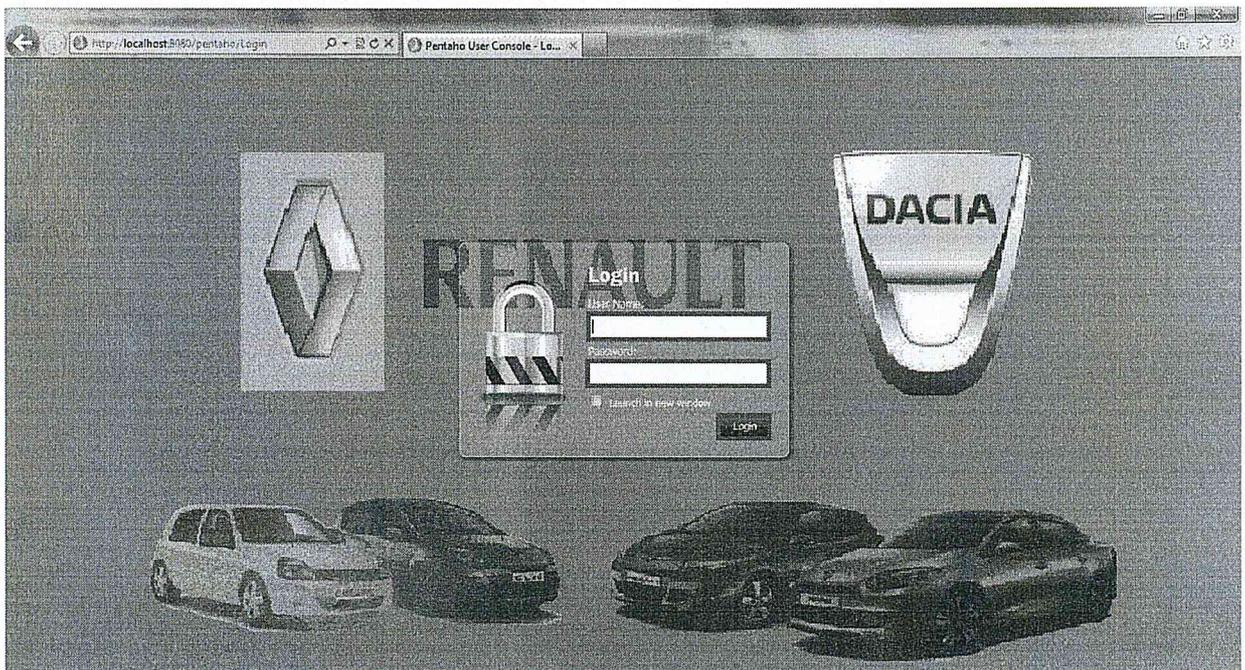


Figure 45 : Interface d'accueil.

Dans le cas où il rentre un nom d'utilisateur ou un mot de passe incorrecte, un message d'erreur lui est affiché.

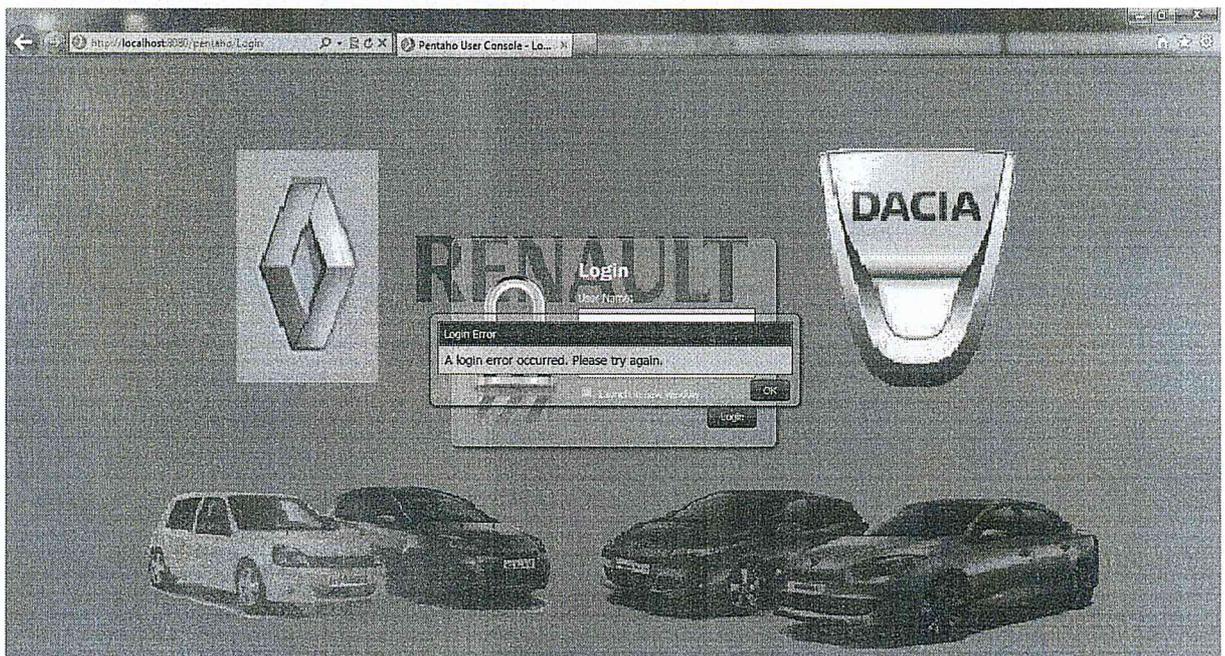


Figure 46 : Message d'erreur.

5.2) Consultation :

Après l'authentification de l'utilisateur, ce dernier peut naviguer selon les rôles qui lui sont attribués, dans le référentiel du portail contenant les tableaux de bord des 3 services, organisés en dossiers. Il choisit un tableau de bord, comme le montre la figure suivante avec l'affichage du tableau de bord récapitulatif des chiffres d'affaire de tous les services.

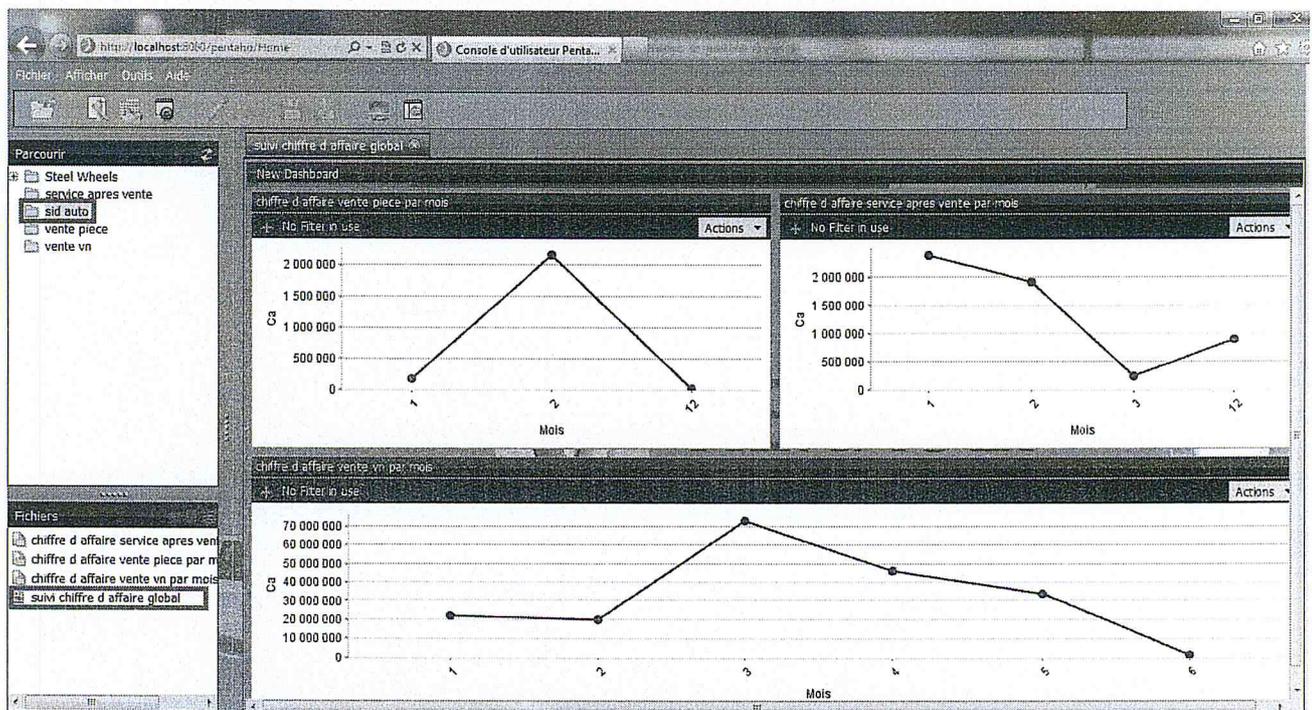


Figure 47 : Tableau de bord suivi des chiffres d'affaire des services.

Ensuite, à partir du tableau récapitulatif il peut accéder au tableau de bord d'un service spécifique en cliquant sur le schéma, comme l'illustre la figure suivante, 4 schémas concernant les ventes Véhicules neufs s'affichent.

CHAPITRE 7 : MISE EN ŒUVRE DE L'ARCHITECTURE DU SYSTEME DECISIONNEL

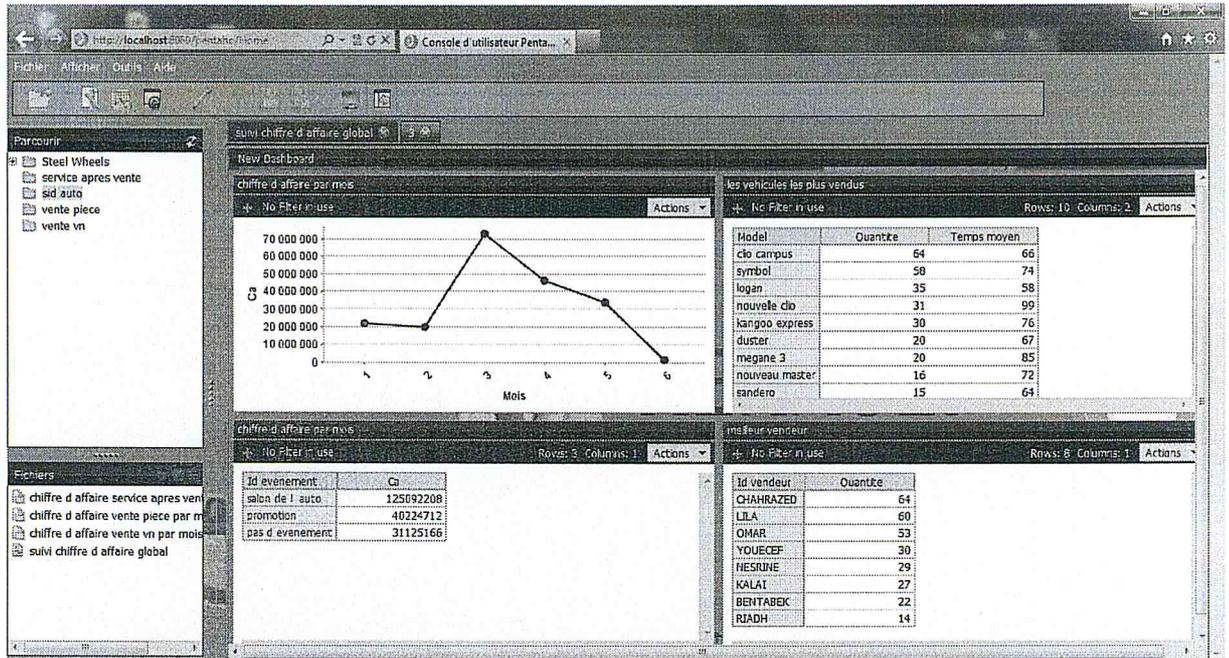


Figure 48 : Tableau de bord suivi vente véhicules neufs 1.

Par la suite s'il veut encore détailler l'un des schémas, il clique et un autre tableau plus détaillé est affiché, comme le montre la figure suivante.

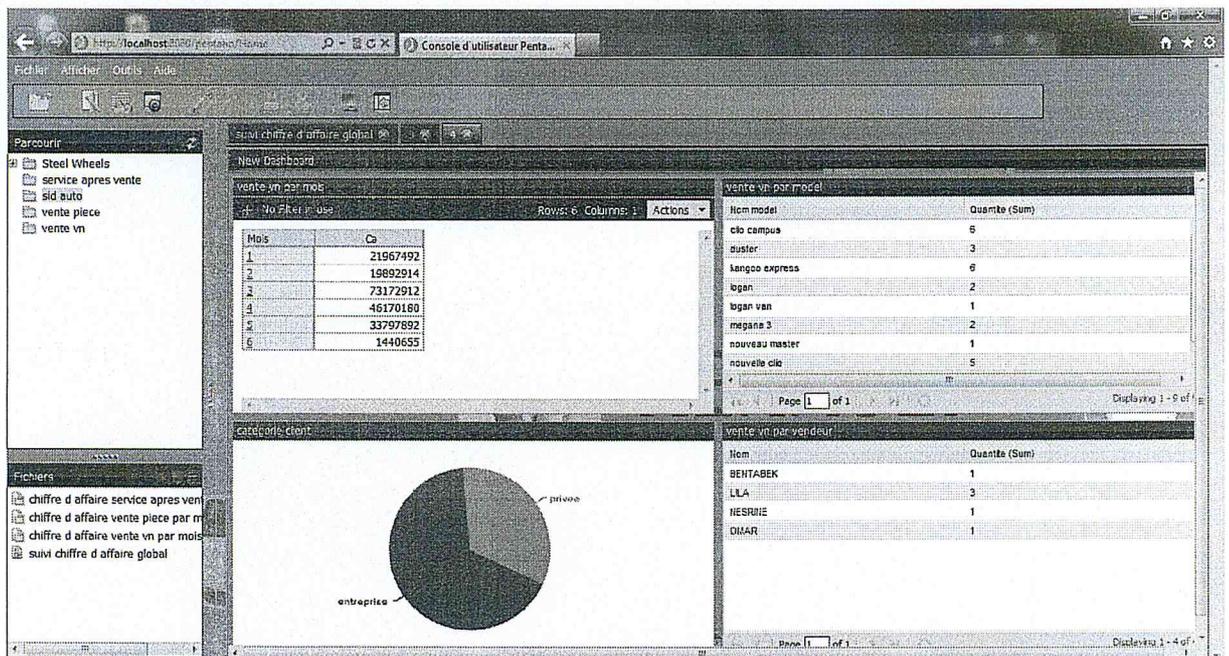


Figure 49 : Tableau de bord suivi vente véhicules neufs 2.

Il peut créer aussi son propre tableau de bord ou rapport et le partager avec d'autres utilisateurs, la figure suivante montre comment partager un tableau de bord ou un rapport avec d'autres utilisateurs .

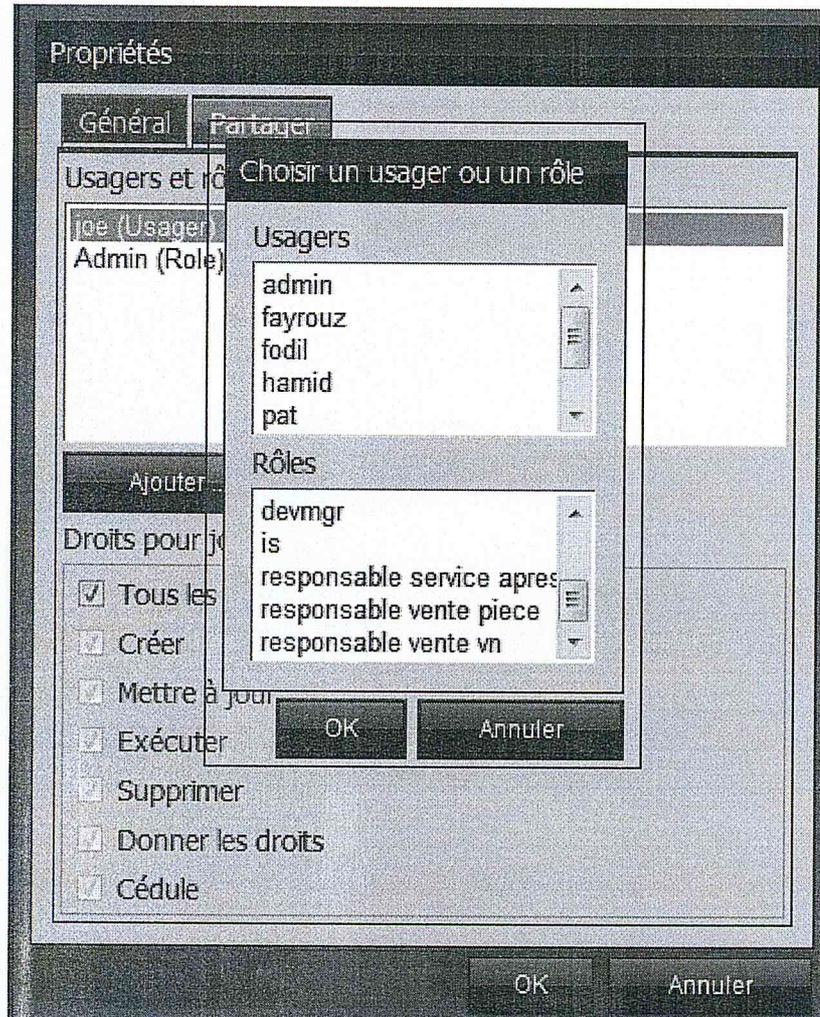


Figure 50 : Partage des rapports.

Conclusion :

Nous avons décrit dans ce chapitre la mise en œuvre de notre système et les outils utilisés, ainsi que la démonstration du système final qui est créé avec la plateforme Pentaho.

Conclusion et perspectives:

Le défi actuel des entreprises modernes est d'exploiter et d'analyser les données de leurs systèmes opérationnels afin de créer de la valeur ajoutée associée aux investissements.

Dans ce cadre, et afin de palier à des problèmes récurrents dans le processus de prise de décision, Icon Software qui fait l'infogérance du service informatique de SID AUTO, a initié le projet de réalisation et de mise en place d'un système décisionnel fiable et efficace basé sur les Data Warehouse NoSQL.

Tout au long de notre travail nous avons adopté la méthode de Ralph Kimball, qui consiste à concevoir le système en suivant une approche qui se base sur les besoins de l'entreprise. Cette approche passe principalement par trois étapes, identification des besoins, la conception du système et la mise en œuvre du système.

Dans la première étape, les différents besoins ont été identifiés, cette étape est basée principalement autour des méthodes de collecte d'information précisément les entretiens et les formulaires.

La deuxième étape a été consacrée à la conception de notre système décisionnel et la construction des différents composants, en commençant par la modélisation multidimensionnelle du Data Warehouse, qui offre une vision claire et compréhensible.

Après la modélisation multidimensionnelle, vient la construction du Data Warehouse NoSQL, ou la principale cause du choix NoSQL est d'avoir un schéma dynamique et flexible. Parmi les solutions du NoSQL nous avons choisi la solution orienté document pour la simplicité et la logique proche du modèle relationnel où les tables sont représentées en collections et les attributs en documents dans ces collections. Pour l'alimentation du Data Warehouse NoSQL nous avons développé notre ETL en Java 2F Loader. Nous avons aussi construit un Data Warehouse classique (SQL) et fait son alimentation avec TALEND et 2F Loader, pour pouvoir comparer entre les deux approches SQL et NoSQL, d'où on a déduit que le NoSQL est plus adapté au domaine du décisionnel, car le fait d'avoir un schéma dynamique

aide à enrichir dès que possible le Data Warehouse. Ensuite, nous avons appliqué l'algorithme MapReduce, souvent associé au NoSQL, qui est un algorithme de calcul parallèle et rapide pour l'analyse de données des sources hétérogènes selon des indicateurs prédéfinis. Les résultats des calculs sont enregistrés dans des fichiers texte normalisés, ces derniers fichiers vont être l'objet des sources de données pour la création de nos tableaux de bord.

La dernière étape a été consacré à la mise en œuvre du système. Le Data Warehouse est donc créé et relié avec le système opérationnel par l'intermédiaire de notre outil d'alimentation 2F Loader. Nous avons aussi implémenté le MapReduce pour l'analyse des données, dont on a programmé en Java les fonctions Map et Reduce, le Map est exécuté par les nœuds en parallèle, le Reduce est exécuté par le Master. Nous avons au final construit nos tableaux de bord à l'aide de la plateforme Pentaho.

L'utilisation de différents outils est un volet essentiel dans ce projet. Notre système décisionnel est donc implémenté sur : MongoDB pour l'entrepôt, intranet comme infrastructure réseau et l'architecture trois-tiers comme technologie de développement web, ainsi que de multiples outils ont été utilisés : Pentaho Business Analytics, ETL Talend, Tomcat AppServer, Pentaho Data Integration (Kettle), PostgreSQL, Java/JavaEE, HTML/CSS.

Durant le déroulement de notre projet, nous avons acquis une bonne expérience précisément dans les Système décisionnels basés sur le Data warehousing. Le Business Intelligence, le Data Warehouse, l'ETL, le NoSQL, le MapReduce, les tableaux de bord sont des nouveaux concepts que nous avons tenus.

Dans le but d'enrichir notre système, certaines améliorations peuvent être ajoutées par la suite, elles sont définies comme suit :

- Utiliser une plateforme certifiée MapReduce,
- Continuer le développement de la zone de restitution,
- Introduire le data Manning,
- Ajouter un accès mobile aux tableaux de bord (Smartphone, tablette),

Bibliographie:

[Aurelien FOUCRET, 2011]

Aurelien FOUCRET, NoSQL une nouvelle approche du stockage et la manipulation des donnees, 2011

[Ben mehidi et Mbilinyi, 2009]

Ben mehidi Mbilinyi, memoire de fin d'etude « conception et realisation d'un tableau de bord d'activite pour suivre les echanges interbancaires au Centre de pre-compensation Interbancaire (C.P.I) »Ecole national superieure d'informatique, 2009

[BENDAVID, 2010]

Olivier BENDAVID, mémoire de stage de master 2 « BI in the cloud » université de montpellier II ,2010.

[BESSON et POSSIN, 1996]

« Du renseignement à l'intelligence économique » ,1996

[BOU, 2003]

Henri bouquin, « Le contrôle de gestion » ,2003

[BOUSSEBAT, 2003]

Khaled BOUSSEBAT, memoire de fin d'etude «Conception et déploiement d'un entrepôt de données métier, Cas : la fonction commerciale de Naftal Spa » Institut national de formation en informatique (I.N.I), 2003

[Chaudhuri, 1997]

S. Chaudhuri, « *Data warehouse and OLAP for decision support* », Microsoft research, Redmond: SIGMOD Record, 1997, USA.

[Inmon, 2002]

W. H. Inmon; « Building the Data Warehouse Third Edition » ; Wiley Computer Publishing, 2002.

[Ghomari, 2000]

A. R. Ghomari, « *Les apports des nouvelles technologies de l'information pour les systèmes décisionnels* », Séminaire Algéro-canadien, El-Aurassi, Mars 2000.

[Katia Aresti, 2010]

Katia Aresti, MongoDB en pratique, 2010

[Khaled TANNIR, 2011]

Khaled TANNIR, MapReduce Algorithme de parallélisations des traitements, Doctorant CIFRE LARIS/ESTI, 2011

[kim, 1997]

Kimball Ralph, « *Entrepôt des données* », Edition Internationnel Thomson Publishing France 1997.

[Kimball, 2001]

Kimball Ralph, « *Entrepôt des données, Guide Pratique de Construction Data Warehouse* » Edition WILEY 2001.

[kimball, 2005]

Kimball Ralph J. Caserta ; « *The Data warehouse ETL toolkit* » ; wiley computer Publishing 2002

[MARTRE, 1994]

Martre H., Levet J.-L., et al. *Rapport dit « Martre » : Intelligence économique et stratégie des entreprises*, Rapport du Commissariat Général au Plan, Paris, La Documentation Française, 1994.

[Malo Jaffre et Pablo Rauzy, 2010]

Malo Jaffre, Pablo Rauzy, MapReduce, 2010

[Michaël Figuière, 2010] :

Michaël Figuière, Tour d'horizon des bases de données NoSql, 2010

[Mic, 2009]

Figuiere Michael. « *Nosql europe Tour d'horizon des bases de données nosql* », 2009.

[QUESTER, 2004]

Quester C. *Solutions : les français maîtrisent le terrain*. DSI, 2004, n°10.

[Ralph, 2000]

Ralph Kimball -Laura Reeves -Margy Ross-Warren hornthwaite, « Concevoir et déployer un Data Warehouse-Guide de conduite de projet », Edition Eyrolles 2000.

[Robert REIX ,2004] :

Robert REIX, « SYSTEMES D'INFORMATION ET MANAGEMENT DES ORGANISATIONS » de Robert REIX ,2004

[smile, 2011] :

group smile, Livre blanc Décisionnel Solutions open source, 2011

[Samatar HASSAN, 2006] :

Samatar HASSAN, PRESENTATION DE PENTAHO DATA INTEGRATION(PDI),2006

Webographie :

[Web 1]

<http://bc.tech.coop/blog/070520.html>

[Web 2]:

<http://docs.postgresql.fr/8.4>

[Web 3]:

<http://business-intelligence.developpez.com/tutoriels/presentation-pentaho/>

[Web 4]:

fr.wikipedia.org/wiki/Architecture_trois_tiers