République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed khider de Biskra

Faculté des Sciences Exactes et Sciences de la Nature et de la Vie

Département d'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme Master 2 en Informatique

Option: Système d'Information, Optimisation et Décision

Thème

Approche de construction d'un entrepôt de données à base ontologique

Réalisé par: MERIZIG Ihcene.

Encadré par: Dr. REZEG Khaled.

Soutenu-le : …. /…. /2020 **Devant le jury composé de :**

Année universitaire: 2019/2020

Remerciements

Avant tout, Nous tenons à remercier le bon DIEU le tout puissant et clément de nous avoir illuminé le chemin du savoir et de nous avoir donné le courage, la puissance et la volonté pour accomplir ce modeste projet.

Je tiens particulièrement à exprimer notre profonde gratitude à notre encadreur Dr REZEG Khaled pour ses conseils et encouragements durant toute la période d'encadrement.

Je tiens à remercier tous les enseignants du département d'informatique, le chef de département et le doyen.

Je tiens aussi à remercier vivement les examinateurs pour avoir accepté d'examiner ce travail et leurs participations au jury.

Un énorme merci à nos familles et amies pour leurs éternel soutient et la confiance qu'ils ont en nos capacité.

***A** mes parents,*

***A** mes sœurs,*

***A** toute la famille,*

***A** mes amies et collègues,*

***Et** tous ceux qui m'ont aidé.*

Résumé

Le domaine commercial est avant tout au service du client. C'est un élément indispensable dans toute entreprise qui produit des biens et des services à ses clients. Ainsi, il dispose d'un fichier de clients avec lesquels il a déjà signé un ou plusieurs contrats. Son but est de renouveler ces accords commerciaux et d'élargir sa clientèle. Il joue un rôle essentiel pour assurer la réussite des entreprises commerciales, leur continuation et tenue à suivre un développement rapide du monde de la technologie.

De nos jours, le domaine commercial a grandement progressé grâce à utiliser une plate-forme qui offre les informations influençant dans le domaine commercial d'une façon interprétable par les utilisateurs et la machine. A ce titre, l'ontologie est l'un des meilleures solutions informatique de la modélisation et stockage des connaissances.

Dans ce contexte, nous proposons une approche pour la construction d'un système d'intégration via un médiateur caractérisé par les données fraiches, et pour la correspondance entre le schéma global et local, nous avons opté l'approche Local As Views (LAV) qui a permet de rajouter une nouvelle source sans la modification complète de système. Ce système d'intégration baser essentiellement sur l'implication d'une seule ontologies partagée (global) qui a facile d'implémenter et traiter le problème de l'existence au conflit sémantique de système d'intégration afin d'assurer l'automatisation du processus d'intégration sémantique de données. Enfin, nous montrons l'implémentation dans le domaine commercial.

Mot clé: Domaine commercial, Système d'intégration, Médiation, LAV, Ontologie.

Abstract

The commercial domain is primarily at the service of the customer. It is an indispensable element in any business that produces goods and services to its customers. Thus, it has a file of customers with whom it has already signed one or more contracts. Its aim is to renew these trade agreements and expand its customer base. It plays an essential role in ensuring the success of commercial enterprises, their continuation and being kept paced with a rapid development of the technology world.

Nowadays, the commercial domain has widely progressed thanks to the use a platform that offers to use the information influencing the commercial field in a way that users and the machine can interpret. In this context, the ontologies are one of the best solutions for modeling and stocking knowledge.

In this context, we propose an approach for building an integration system through a mediator that characterizes by the data always fresh, and for the correspondence between the global and local scheme, we opted the Local As Views (LAV) approach that adds a new source without the full system change. This integration system based essentially on the involvement of a single shared ontology (global) that has easy to implement and treated the problem of existence the semantic integration system conflict in order to automate the semantic data integration process. Finally, we implement our proposed mediation architecture in the commercial domain.

Keywords: Commercial domain, Integration system, Mediation, LAV, Ontology.

ملخص

يركز المجال التجاري في المقام الأول على خدمة العميل الذي يعد مركز اهتمام جميع الشركات التي تقدم السلع و الخدمات لإرضاء عملائها، حيث أنها تمتلك قاعدة من الزبائن و تسجل جميع التعاملات معهم (سواء بعقد واحد او اكثر)، كما أنها تهدف إلى تجديد الاتفاقيات التجارية و توسيعها لكسب عدد اكبر من العملاء و تؤدي دورا حيويا يساعد على ضمان نجاح و استمرارية المؤسسات التجارية و المضي قُدُمًا نحو عالم التكنولوجيا.

و على هذا الأساس، نستطيع القول ان الانطولوجيا هي واحدة من احسن الحلول الرقمية المساعدة على نمذجة و تخزين المعرفة إذ سهل التفاعل بين الانسان و الآلة من خلال استخدام المنصة للحصول على المعلومات المؤثرة في المجال التجاري و تفسيرها.

و في هذا السياق، نقترح نهجا لبناء نظام تكاملي عن طريق وسيط يصف البيانات الحديثة، أما بالنسبة للمراسلات بين المخطط العالمي و المحلي، اخترنا نهجا "آراء محلية LAV " الذي يسمح لنا بإضافة مصدر جديد دون القيام بتعديل النظام التكاملي. فهذا الاخير يعتمد اساسا على تطبيقات الأنطولوجيا، و التي تتميز بتسهيل تنفيذ و معالجة مشاكل وجود الصراعات في النظام و ذلك من خلال ضمان أتمتة عملية تكامل البيانات الدلالية. و في الأخير نصل لعرض التنفيذ على المجال التجاري.

الكلمات المفتاحية: المجال التجاري، نظام التكامل، الوسيط، LAV، الأنطولوجيا.

Table des matières

Liste des figures

Chapitre 01: Concept de base sur les entrepôts de données

Chapitre 02: Construction d'un entrepôt de données

Chapitre 03: Conception

Chapitre 04: Implémentation

Liste des tableaux

Introduction générale

Introduction générale

a. Contexte

Les entreprises commerciales utilisent les données massives pour gagner plus de revenus et investir dans des applications, il faut savoir que ces dernières ne cessent d'évoluer pour faire bon fonctionnement et bon développement commercial.

En raison du volume important des données qui doivent à stocker, manipuler et à analyser, le Système d'Information Opérationnel (SIO) ne suffit plus pour pérenniser l'activité de l'entreprise, car celui-ci convient bien aux applications gérant l'activité quotidienne de l'entreprise, mais s'avère inadapté au décisionnel. Ainsi que la variété des sources de données disparates, il est difficile de recueillir et d'intégrer des données à partir d'emplacements distribués, pour cela le processus décisionnel est un projet qui se construit, il est né d'un besoin exprimé par les entreprises et pour assurer une certaine performance en termes de récupération rapide des données qui provenant de différentes sources et de différents formats.

L'hétérogénéité des sources des données et leurs variation posent un problème dans l'intégration des données afin de permettre un accès unifie à leur sources et d'offrir à l'utilisateur une vue uniforme et transparente des informations issues de sources hétérogènes et distribuées sans avoir à connaitre leur source ou la façon dont elles sont interrogées.

Dans l'approche d'intégration des données, il existe deux approches: approche d'entrepôt de données (approche matérialisée) et l'approche médiation de données (approche virtuelle). Dans l'approche matérialisée, les sources sont migrées vers un entrepôt, cet entrepôt contient donc toutes les données des sources à intégrer, par contre, dans l'approche virtuelle les données restent au niveau des sources, et le médiateur se charge de diviser la requête de l'utilisateur en sous requêtes et les envoyer

à des adaptateurs qui se chargent de traduire la requête et renvoyer les résultats au médiateur.

Dans l'approche médiateur, les correspondances (mappings) entre le schéma global et les schémas locaux se font de manière manuelle, cependant des travaux ont été réalisés afin d'automatiser ces correspondances. Ils existent deux types de méthodes pour définir ces correspondances, Global-As-Views (GAV) et Local-As-Views (LAV). Pour GAV, consiste à définir le schéma global en fonction des schémas des sources de données à intégrer, mais LAV, les entités des schémas locaux sont définies comme des vues sur le schéma global (contraire à l'approche GAV).

b. Problématique

Le problème majeur des systèmes de médiation est les conflits qui existent entre les sources hétérogènes. Actuellement, la plupart des solutions reposent sur l'utilisation des ontologies afin d'automatiser le processus d'intégration cette ontologie joue le rôle de schéma global dans un système d'intégration.

c. Objectifs

L'objectif principal de notre travail consiste à interroger la diversité et l'hétérogénéité les différentes sources de données au sein de domaine commercial en langage naturel de manière transparente, en utilisant les restrictions sémantiques imposées par l'ontologie, pour cela nous proposons une architecture de médiation où chaque source à intégrer est liée à une seule et même ontologie globale (schéma global). Ceci permet de remédier au problème d'hétérogénéité sémantique des données. Les objectifs de notre travail sont :

- Implémentation d'une ontologie de domaine destinée à la modélisation des connaissances du domaine commercial(***OntoDC***) pour l'automatisation le processus d'intégration et résoudre le problème de conflit sémantique existant dans les différentes bases de données distribuent.

- Développement d'une interface qui facilite l'interaction entre l'utilisateur (saisit la requête en langage naturel), et le système (extraction les données d'après les différentes sources de données puis l'affiche le résultat).

d. Organisation de la mémoire

Pour réaliser notre travail, nous avons structuré notre mémoire en quatre chapitres comme suit :

Chapitre 01 « *Concept de base sur les entrepôts de données* » : dans ce chapitre nous détaillons les notions et les concepts de base liée aux entrepôts de données, nous décrivons l'architecture d'entrepôt de données, les différentes classifications des hétérogénéités qui peuvent apparaitre lors de l'intégration, les langages d'implémentations des entrepôts, et nous faisons un tour sur quelques exemples de projet d'intégration par médiateur et domaines d'application d'entrepôt de données.

Chapitre 02 « *Construction d'un entrepôt de données* » : présente les différentes approches de construction entrepôt de données, nous détaillons les notions de base liées aux ontologies, nous décrivons les différents composants qui constituent une ontologie, les classifications des ontologies, les langages de description des ontologies, nous détaillons ainsi les étapes de construction l'ontologie.

Chapitre 03 « *Conception*» : dans ce chapitre nous présentons notre architecture et les différentes composants du système, nous détaillons les étapes de construction l'ontologie global de domaine commercial (***OntoDC***).

Chapitre 04 « *Implémentation* » : consacré à l'implémentation de notre architecture et la description du fonctionnement.

On termine par une conclusion générale du notre travail.

Chapitre 01: Concept de base sur les entrepôts de données

Introduction

La diversité, l'hétérogénéité des sources et la quantité très important de données électroniques dans l'entreprise. Ces données sont stockées dans les systèmes opérationnels de l'entreprise au sein de bases de données, de fichiers... L'exploitation de ces données dans un but d'analyse et support à la prise de décision s'avère difficile; elle est réalisée le plus souvent de manière imparfaite par les décideurs grâce à des moyens classiques (requêtes SQL[1], outils graphiques d'interrogation…).

Ces systèmes permet de manipuler les activités quotidienne de l'entreprise mais s'avère inadapté au décisionnel. A cause de ce problématique, les entreprises ont recours à des systèmes d'aide à la décision spécifiques qui basés sur approche d'entreposage des données qui ont fait leurs apparitions à la fin des années 1980.

L'entreposage des données est la technologie qui adoptée pour rassembler toutes les informations d'une entreprise en une base de données unique destinée aux analystes et d'aide à la décision spécifiques. Cette technologie, que l'on retrouve aujourd'hui dans plusieurs domaines tels que les entreprises commerciales.

Dans ce chapitre nous commençons d'abord par les différentes définitions d'entrepôt de données et leur origine. Après, nous présentons les concepts et notions de base des entrepôts de données et enfin nous donnons quelques domaines d'application d'entrepôt.

1. Définitions d'un entrepôt de données et leurs historiques

Dans cette partie, nous présentons les différentes définitions d'un entrepôt de données et l'origine de ce dernier.

1.1. Définitions d'un entrepôt de données

Plusieurs définitions ont été données au concept ED.

[1] Structured Query Language.

Définition 01: Selon Bill Inmon en 1996 définit l'entrepôt de données (ED) dans son ouvrage "Building the Data Warehouse" de la façon suivante: "*L'entrepôt de données est une collection de données orientées sujet, intégrées, non volatiles et historiées, organisées pour support d'un processus d'aide à la décision*"[1].

Définition 02: Selon SEN-SINHA " *Un entrepôt de données est: orienté-sujet, intégré, variant dans le temps et non volatile*" [2].

Définition 03: Selon Ralph Kimball en 1996 définit l'entrepôt de données dans son ouvrage " The Data Warehouse Toolkit " comme suite: "*L'entrepôt de données est une copie des données transactionnelles d'une entreprise structurée de manière spécifique pour l'interrogation et l'analyse* "[3].

Autre définitions: "*Un entrepôt de données est une archive numérique qui collecte et diffuse des jeux de données et leurs métadonnées. Un grand nombre d'entrepôts de données acceptent également des publications et permet de lier les publications afférentes* "[4].

A partir de ces définitions précédentes, l'entrepôt de données est l'espace de stockage centralisé d'un extrait des sources de données pertinentes pour les décideurs. Elle caractérisé par les termes suivants: orienté-sujet, intégré, historié et non volatile conçus pour l'aide à la décision.

1.2. Historique d'entrepôt de données

L'histoire de l'entrepôt de données est née par General Millset l'Université Dartmouth en 1960. Il commence par la création de deux termes principaux de l'entrepôt de données: fait et dimension.

En 1983: Teradata introduit dans son SGBD[2] un système purement décisionnel.

[2] Système de Gestion des Bases de Données.

En 1988: Barry Devlin et Paul Murphy publient l'article "An architecture for business and information systems" dans le journal système d'IBM qui utilise le terme Data Warehouse pour la première fois.

En 1990: Red Brick Systems construit le système "Red Brick Warehouse" dédié à la construction d'entrepôt de données.

En 1991: Bill Inmon publie le livre "Building the Data warehouse" (Construire l'entrepôt de données).

En 1995: La création de l'organisation "Data Warehousing Institute" pour soutenir et promouvoir la recherche dans le domaine des ED.

En 1996: Ralph Kimball publie le livre "The Data Warehouse Toolkit" (La boîte à outils de l'entrepôt de données).

En 1997: Réalisation de "Oracle 8", avec la prise en charge des requêtes des schémas en étoiles [5].

2. Caractéristique de l'entrepôt de données

Selon William H. Inmon, l'inventeur englobe les termes suivants (figure 01):

Figure 1.1: Les caractéristiques d'un entrepôt de données [6].

Orienté sujet: Les données de l'entrepôt sont organisées par thème (autour des sujets majeurs et des métiers de l'entreprise). L'intérêt dans cette organisation est de disposer d'un ensemble d'informations utiles sur un sujet transversal aux structures fonctionnelles et organisationnelles de l'entreprise [7]. (Figure 02).

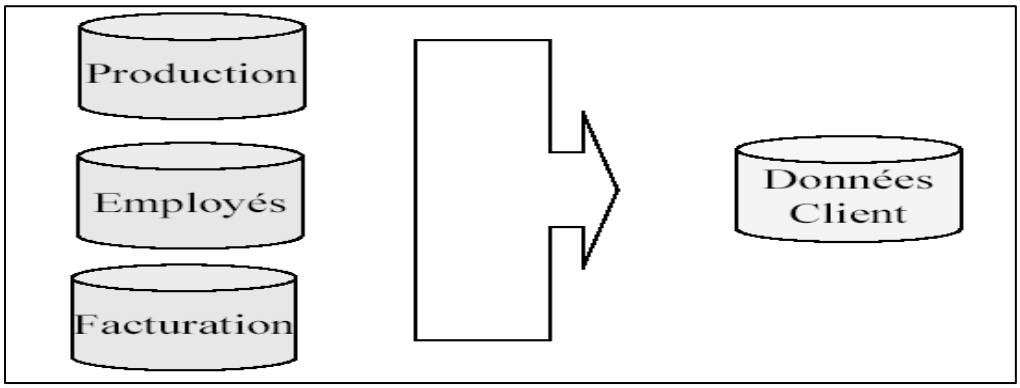

Figure 1.2: Données orienté sujet dans un ED [8].

Intégrée: Les données dans ED proviennent de sources hétérogènes utilisant chacune un type de format. L'intégration permet d'avoir une cohérence de l'information [6]. (Figure 03).

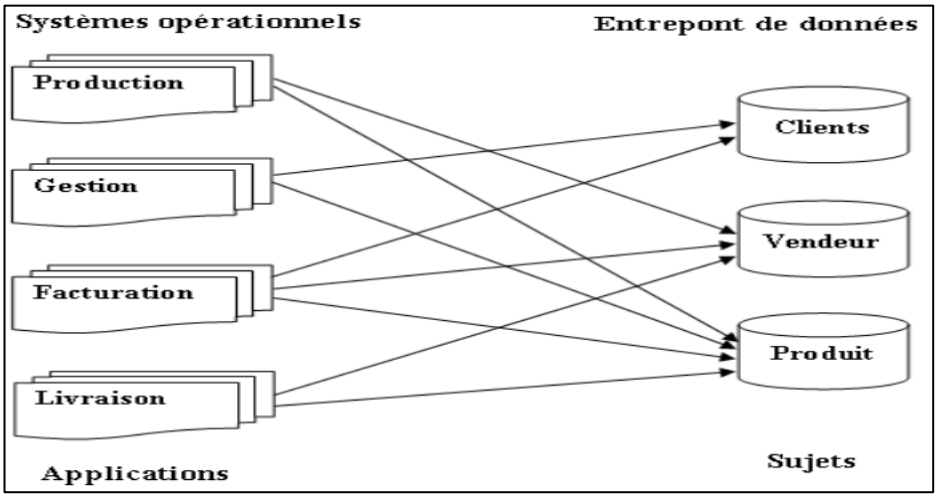

Figure 1.3: Données orienté dans un ED [8].

Non volatile: Les données d'un entrepôt sont généralement utilisées en mode consultation et peuvent être interrogées en lecture seule, elles ne sont ni modifiées ni supprimées (sauf dans les cas de rafraichissement de l'entrepôt) afin de conserver la traçabilité des informations sur une longue période et des décisions prises [9]. (Figure 04).

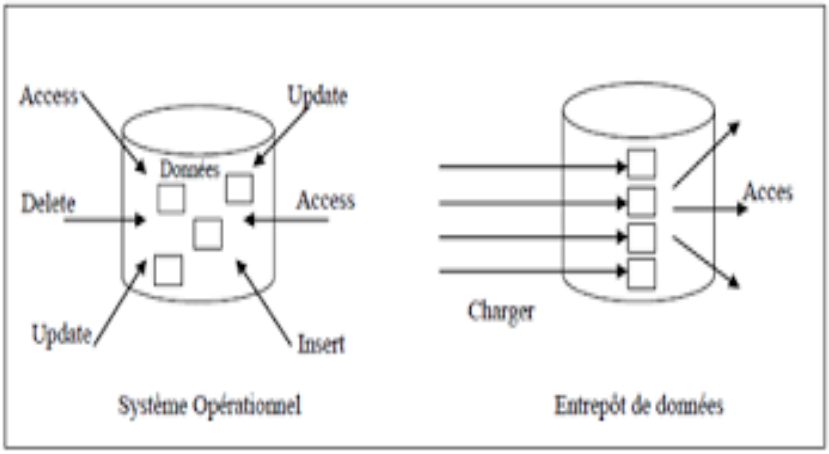

Figure 1.4: Données non volatile dans un ED [10].

Historiée: Pour suivre dans le temps l'évolution des différentes valeurs des indicateurs à analyser, l'historisation est nécessaire. Un référentiel de temps est associé aux nouveaux données a insérée, afin de permettre l'identification dans la durée de valeurs précise [11].

3. Classes des données

L'entrepôt de données se structure en quatre classes de données [11]:

3.1. Données agrégées

Les données agrégées correspondent à des éléments d'analyse représentant les besoins des utilisateurs. Elles constituent déjà un résultat d'analyse et une synthèse de l'information contenue dans le système décisionnel, et doivent être facilement accessibles et compréhensibles.

3.2. Données détaillées

Les données détaillées reflètent les événements les plus récents. Les intégrations régulières des données issues des systèmes de production vont habituellement être réalisées à ce niveau.

3.3. Métadonnées

Les métadonnées constituent l'ensemble des données qui décrivent des règles ou processus attachés à d'autres données. Ces dernières constituent la finalité du système d'information.

3.4. Données historiées

Chaque nouvelle insertion de données provenant du système de production ne détruit pas les anciennes valeurs, mais créée une nouvelle occurrence de la donnée.

4. Architecture d'un entrepôt de données

Le processus de construction d'un ED représenté dans la (figure 05) qui se constitue de plusieurs niveaux: Source de données, Stockage, OLAP[3] tier et les outils Front-end tier. Ces composantes seront détaillées ci-dessous.

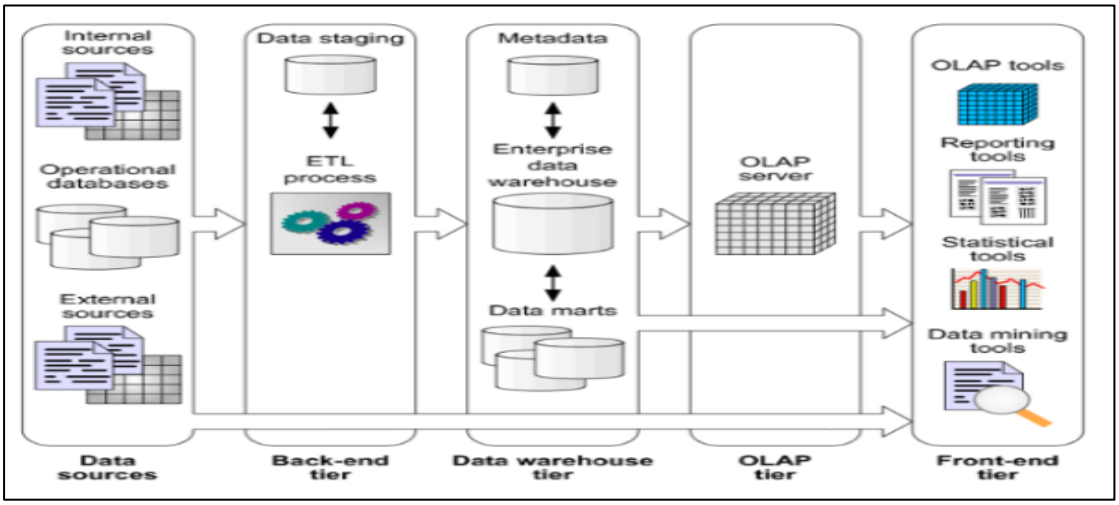

Figure 1.5: Architecture d'un entrepôt de données [12].

[3] On-Line Analytical Processing

4.1. Sources de données (Data sources)

L'entrepôt de données stocke des données qui proviennent de différentes sources d'informations hétérogènes et distribuées. Ces sources peuvent être des bases de données, des fichiers de données, des sources externes à l'entreprise,… etc [13].

4.2. Niveau d'arrière-plan (Back-end tier)

Représente la zone de construction qui contient l'ensemble d'outils et techniques utilisés lors du processus de préparation des données, avant leurs chargements au niveau de l'entrepôt, les données des sources doivent d'abord être nettoyées comme suivantes:

4.2.1. ETL process (Extraction, Transformation and Loading)

Le processus de nettoyage consiste à sélectionner et à Affiner les données pour éliminer toute erreur et réconcilier les différences sémantiques entre ces données. Une fois nettoyées, ces données seront intégrées dans l'entrepôt. Le processus de rafraîchissement consiste à propager vers l'entrepôt, le changement effectué sur les données des sources par le processus ETL [13].

Cette étape se fait en trois comme suivant:

4.2.1.1. Extraction

L'extraction consiste à récupérer les données hétérogènes à partir de multiples sources, base de données opérationnelles, ou des fichiers de différents formats. Cette étape nécessite de gérer la synchronisation des processus d'extraction afin d'assurer l'intégrité des données chargées en utilisant l'interface du programme d'application, telles que ODBC[4], OLEDB[5] et JDBC[6] [12].

[4] Open DataBase Connection.

[5] Open Linking and Embedding for DataBase.

[6] Java DataBase Connectivity

4.2.1.2. Transformation

Transformation Consiste en une série de règles (comporte plusieurs aspects: le nettoyage, l'intégration et l'agrégation [14]) permettant de rendre les données cohérentes avec la structure d'un entrepôt de données c'est-à-dire les données des systèmes de production doivent être agrégées ou calculées avant leur chargement [15].

4.2.1.3 Chargement

C'est la dernière phase d'alimentation d'un DW[7]. Il s'agit d'insérer les données au sein du DW. C'est une phase délicate car les quantités de données sont souvent très importantes [16].

Ce phase inclut également le rafraîchissement de l'entrepôt de données à savoir, la propagation des mises à jour dans l'entrepôt de données à partir des sources de données à une fréquence spécifiée afin de fournir des données à jour pour le processus de prise de décision [7].

4.3. Niveau entrepôt de données (Data Warehouse tier)

Se compose d'un entrepôt de données d'entreprise (Entreprise Data Warehouse) et/ou de plusieurs magasins de données (data marts), et des métadonnées (Metadata) stockant des informations sur l'entrepôt de données et son contenu.

4.3.1. Entrepôt de données d'entreprise (Entreprise Data Warehouse)

L'entrepôt de données d'entreprise est un entrepôt de données centralisé qui englobe tous les domaines fonctionnels ou départementaux dans une organisation [12].

4.3.2. Métadonnées (Metadata)

Les données concernant la création, la gestion, et l'usage de l'entrepôt sont stockées dans un répertoire indépendant de l'entrepôt. Ces données sont appelées "métadonnées". Ces dernières peuvent contenir des informations sur les sources et leurs contenus, le schéma de l'entrepôt, les règles de rafraîchissement… [10].

[7] Data Warehouse.

4.3.3. Magasin de données (Data marts)

Un magasin de données est un entrepôt de données ciblées sur un sujet, alimenté depuis l'entrepôt de données de l'entreprise ou directement à partir de source de données [17]. Les données extraites sont adaptées à une classe de décideurs ou à un usage particulier (recherche de corrélation, logiciel de statistiques,...). L'organisation des données suit un modèle spécifique qui facilite les traitements décisionnels de type OLAP [18].

4.4. On-Line Analytical Processing tier (OLAP)

Se compose d'un serveur OLAP, qui permet d'accéder à l'entrepôt, il convertit les requêtes des clients en requêtes d'accès à l'ED et fournit des vues multidimensionnelles des données à des outils d'aide à la décision [19]. Il y a plusieurs types de serveurs OALP [7]:

- *Relationnelle OLAP (ROLAP)*: utilisent un SGBD relationnel classique avec des adaptations spécifiques à l'OLAP, et la base relationnelle de l'entrepôt est organisée pour réagir comme une base OLAP.
- *Multidimensionnelle OLAP (MOLAP)*: utilisent un SGBD multidimensionnel (MOLAP), ils sont l'application physique du concept OLAP (réellement d'une structure multidimensionnelle).
- *Hybride OLAP (HOLAP)*: C'est un compromis entre une base MOLAP pour les données souvent consultées (la minorité) et une base ROLAP pour les autres (la majorité).
- *Desktop OLAP (DOLAP)*: C'est une base hébergée sur le poste client.
- *Object OLAP (OOLAP)*: utilise un SGBD Orienté Object.

4.5. Front-end tier

Ces outils formatent les données, conformément aux besoins des utilisateurs (l'informatique décisionnelle (BI[8])) [9]. Les différents outils sont les suivants [7]:

[8] Business Intelligence.

4.5.1. Outils OLAP

Les outils OLAP permettent l'exploration et la manipulation des données de l'entrepôt afin de trouver des modèles ou des tendances importantes pour l'organisation. Ils facilitent la formulation de requêtes complexes qui peuvent impliquer de grandes quantités de données. Ces requêtes sont appelées des requêtes ad-hoc.

4.5.2. Outils de reporting

Les outils de reporting permettent la production, la livraison et la gestion des rapports. Les rapports utilisent des requêtes prédéfinies, comme les requêtes demandant des informations dans un format spécifique qui sont effectuées sur une base régulière. Ces rapports peuvent être exécutés automatiquement ou manuellement.

4.5.3. Outils statistiques

Les outils statistiques sont utilisés pour analyser les données du cube en utilisant des méthodes statistiques.

4.5.4. Outils de fouille de données (DM)

Les outils DM permettent d'extraire des modèles d'une base de données historiée afin de décrire le comportement actuel et/ou de prédire le comportement futur d'un procédé. Les modèles peuvent être des modèles de calculs (des équations par exemple) ou des modèles logiques (des règles par exemple).

Certaines architectures d'entrepôts ne peuvent pas comporter toutes les composants illustrés dans la (Figure 05).

L'architecture de l'entrepôt peut contenir seulement un entrepôt d'entreprise sans le data mart ou alternativement un data mart sans l'entrepôt de données d'entreprise.

Dans d'autre situations, il n'existe pas de serveur OLAP et les outils clients accèdent directement à l'entrepôt de données (Figure 05: flèche reliant le Data warehouse tier au Front-end tier).

Une autre situation, un entrepôt de données virtuel, où il y a ni un entrepôt de données, ni un serveur OLAP. Il définit un ensemble de vue sur les bases de données opérationnelles qui sont matérialisées pour un accès efficace (Figure 05: flèche reliant les sources de données au frond-end tier) [7].

5. Intégration et extraction de données

La diversité des sources et leur hétérogénéité représente l'un des obstacles rencontrés par les utilisateurs du Web aujourd'hui. Pour contourner ce problème nous faisons appel à l'intégration des données qui donne l'impression à l'utilisateur qu'il utilise un système homogène, en éliminant les conflits entre ces données, et les présenter de manière cohérente.

Dans la suite nous représentons les différentes types d'hétérogénéité de données, ensuite nous étudions les approches entrepôt et médiation de données.

5.1. Hétérogénéité de données

Face au problème des systèmes d'intégration est l'hétérogénéité des sources réparties de données car chaque système est conçu de manière différente, par des personnes différentes et utilisant des vocabulaires différents. Avant de définir les approches d'intégration de données, il est important d'étudier les différentes hétérogénéités qui peuvent exister lors de l'intégration, deux types majeurs d'hétérogénéité existent:

5.1.1. Hétérogénéité structurelle

Hétérogénéité structurelle lorsqu'il existe différentes représentations des mêmes concepts c'est-à-dire elle provient quand les sources adoptent différents modèles de données, structures de données ou schémas, par exemple utiliser différents unités pour exprimer la même mesure. On l'appelle aussi hétérogénéité des schémas [20].

5.1.2. Hétérogénéité sémantique

Hétérogénéité sémantique se trouve lorsque l'on exprime le même concept mais avec des significations différentes (conflit). Il existe plusieurs types de conflits.

- Selon Assia Soukane définit deux types de conflits sémantiques:

✓ *Conflits sémantiques liés au schéma* où on décrit le même concept utilisant des terminologies différentes.

✓ *Conflits sémantiques liés aux données* où les données proviennent de différentes origines, saisies à des moments différents par des personnes différentes qui n'ont pas la même perception, et utilisent des conventions différentes.

- Selon Dung Xuan Nguyen définit quatre types de conflits:

✓ *Conflits de représentation*, qui apparaissent lorsqu'on utilise des propriétés différentes ou des schémas différents pour décrire un même concept.

✓ *Conflits de noms (termes)*, dans le cas où on utilise des noms différents pour décrire le même concept, ou bien utiliser des noms identiques pour décrire des concepts différents.

✓ *Conflits de contextes*, ils se produisent dans le cas où on définit un concept mais dans des contextes différents.

✓ *Conflits de mesure de valeur* on les trouve dans le cas où on utilise des unités différentes pour mesurer la valeur d'une propriété.

- Selon Bakhtouchi distingue trois types de conflits:

✓ *Conflits technologiques* qui sont liés à la technologie utilisée pour représenter les données, par exemple utiliser des SGBD différents.

✓ *Conflits de schéma* qui peuvent être soit des conflits sémantiques, de description, d'hétérogénéité ou des conflits structurels.

✓ *Conflits d'instances* sont causés par les erreurs de qualité, telles que l'exactitude, la complétude, la fraîcheur, et les erreurs de cohérence; ces conflits sont divisés en deux classes:

Conflits de référence qui surgissent lorsque des instances provenant de relations différentes réfèrent au même objet mais contiennent des références différentes.

Conflits de valeurs d'attributs qui apparaissent lorsque des instances, qui correspondent aux mêmes objets du monde réel et partagent une même référence, diffèrent dans d'autres attributs [21].

5.2. Approches d'intégration de données

L'intégration de données consiste à éliminer d'abord les conflits entre les données et ensuite les représenter dans un seul schéma cohérent. Le schéma suivant illustrer l'architecture générale d'intégration (Figure 06).

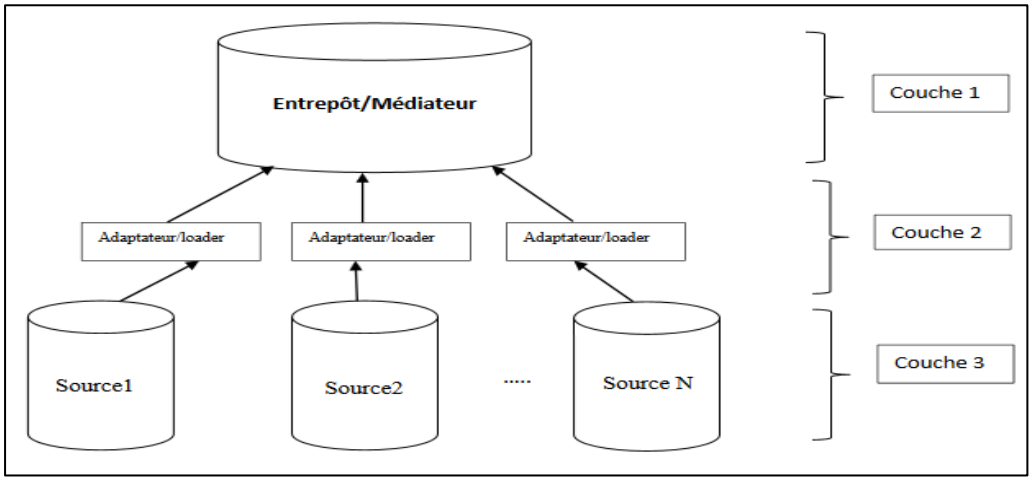

Figure 1.6: Architecture générale d'intégration de données. [21]

Un système d'intégration est composé en général en trois couches principales:

- *Couche de données (couche 3)*: contenant l'ensemble des sources de données à intégrer.
- *Couche des adaptateurs ou des chargeurs (couche 2)*: qui permettent d'extraire les données et de les représenter dans le schéma global, c'est le moyen avec lequel on peut accéder à une source de données, et avec lequel la source peut interagir avec les autres composants de l'architecture.
- *Médiateur/entrepôt (couche 1)*: qui contient le composant qui permet aux utilisateurs d'interroger les différentes sources à travers un schéma global, ce composant peut être un entrepôt de données où toutes les données des sources sont dupliquées (approche matérialisée), ou bien un médiateur (approche virtuelle) [21].

Donc nous distinguons deux approches majeures pour l'intégration de données dans système d'information décisionnel, approche entrepôt de données (matérialisé) et approche médiation (virtuelle).

5.2.1. Approche d'entrepôt de données

Les sources sont migrées vers un entrepôt, cet entrepôt contient donc toutes les données des sources à intégrer pour être utilisée dans les systèmes d'aide à la décision.

La construction d'un entrepôt de données, passe par les étapes de processus ETL (Extract,Transform and Load) suivantes:

- ✓ Extraction des données à partir des sources de données.
- ✓ Transformation des données permettant de formater et nettoyer les données afin de les rendre homogènes pour pouvoir les extraire au niveau de l'entrepôt.
- ✓ Intégration des données, et le stockage des données intégrées au niveau de l'entrepôt (Load). Ces données peuvent être utilisées par des outils décisionnels tels qu'OLAP.

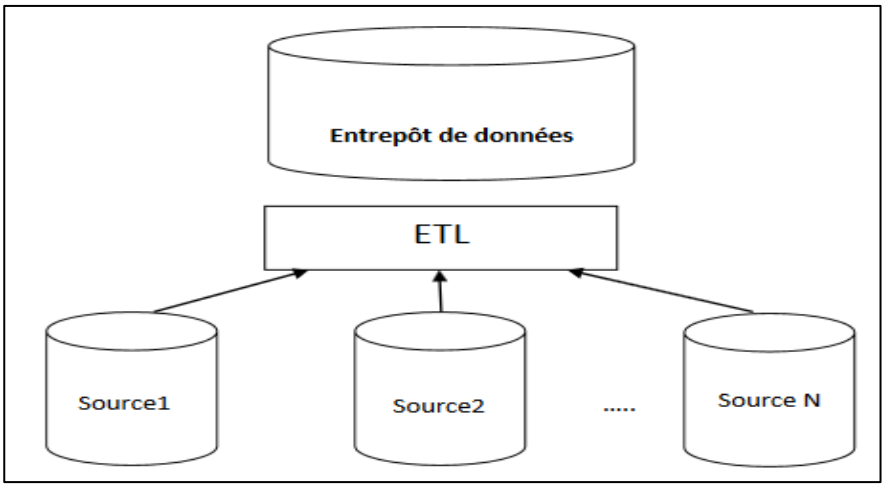

Figure 1.7: Architecture d'entrepôt de données. [21]

L'avantage de cette approche plus de performance (moins de délai), personnalisation des données (nettoyage, filtrage) et les données sont accessibles à partir d'un seul endroit, par contre les données pas toujours fraîches (cohérence),

gestion des mises-à-jour et la gestion de gros volumes de données en termes de stockage (très couteuse).

5.2.2. Approche de médiation de données

Les données de l'approche virtuelle restent au niveau des sources, et le médiateur se charge de diviser la requête de l'utilisateur en sous requêtes et les envoyer à des adaptateurs qui se chargent de traduire la requête et renvoyer les résultats au médiateur.

Généralement, une architecture de médiation est composée de trois couches principales:

- Couche contenant les sources de données à intégrer.
- Couche d'adaptateurs (wrappers) permet de connecter le médiateur à une source de données, il doit être capable d'accepter une variété de questions posées par le médiateur et les traduire en termes de la source.
- Couche du médiateur: c'est l'élément principal dans une architecture de médiation, il permet de formuler la requête de l'utilisateur en sous requêtes destinées au wrappers pour interroger une source de données, à travers un schéma global qui contient des vues sur les sources de données, puis il recompose les résultats obtenus à partir des sources de données en une seule réponse à l'utilisateur [21].

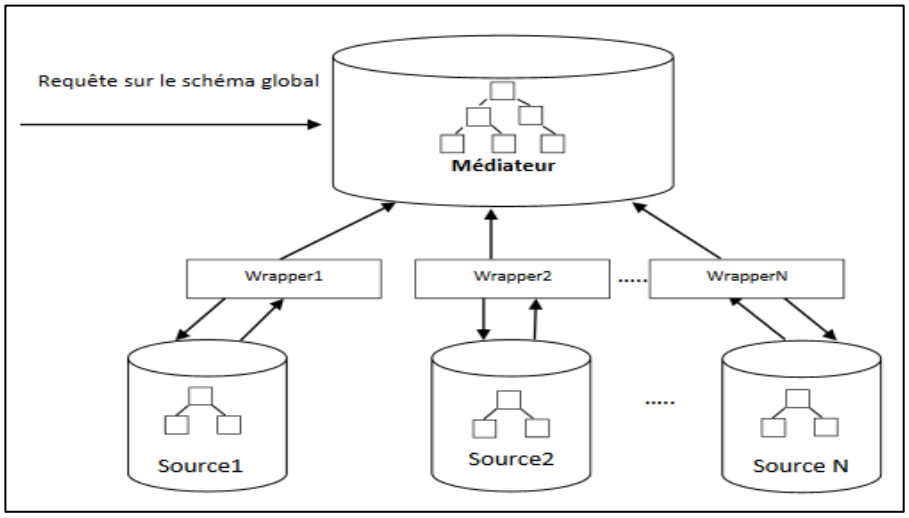

Figure 1.8: Architecture de médiation de données. [21]

L'avantage majeur de cette approche est les données à jours. Par contre, elle est moins performance, traduction des requêtes et capacité et nature différentes des sources.

5.2.2.1. Types de mappings

Mapping fait la description d'un lien entre le schéma global et les schémas des sources (passage entre schémas), il existe deux façons de définition du Mapping. Dans ce qui suit nous détaillons les différents types de Mappings.

5.2.2.1.1. Global-As-View (GAV)

L'approche GAV nécessite que le schéma global soit défini en termes de sources de données. Plus exactement, chaque relation du schéma global est exprimée comme une vue sur les sources, de sorte que sa signification est spécifiée en termes de données résidant aux sources [22]. L'avantage de cette approche est qu'une approche naturelle ainsi la traduction de requête se fait facilement. Mais, qu'on ajoute une nouvelle source -> modification complet du modèle global (il faut considérer l'interaction de la nouvelle source avec les autres).

5.2.2.1.2. Local-As-View (LAV)

Contrairement à l'approche GAV, dans l'approche LAV, les entités des schémas locaux sont définies comme des vues sur le schéma global. L'avantage de cette approche est qu'on ajout d'une nouvelle source au système nécessite uniquement de fournir la définition de la source, et n'implique pas nécessairement des changements dans le schéma global. Une requête est réécrite sous forme de requête conjonctivite selon les vues sur les schémas des sources de données, la réponse à la requête est l'union des résultats obtenues pour les sous requêtes [21].

5.2.2.2. Exemple de projets d'intégration de données par médiation

Dans la suite nous présentons les différents exemples de projets d'intégration de données par médiation [21].

- **Le projet The Stanford-IBM Manager of Multiple Information Sources (TSIMMIS)**

C'est l'un des premiers projets qui utilisent l'approche médiateur-wrapper pour l'intégration de données, il est basé sur l'approche GAV offrant un moyen pour intégrer des sources d'information multiples et hétérogènes [31]. Dans ce projet, un traducteur (wrapper) est associé à chaque source d'information, qui est chargé de réécrire une requête en sous requêtes. Il utilise un modèle de données orienté objet appelé Modèle d'Echange d'Objet (Object Exchange Model : OEM) et un langage de règles (MSL, Mediator Specification Language) pour l'interrogation. Plusieurs médiateurs peuvent exister dans cette architecture, ces médiateurs reçoivent des informations des wrappers afin de les traiter. Le médiateur central permet de générer un plan à la requête.

- **Le système Mediator Environment for Multiple Information Sources (MOMIS**)

Ce système est basé sur l'utilisation de la logique de description de type (ODL-I3) pour décrire les schémas des sources de données à intégrer. Il repose sur un thesaurus dérivé de la base de données lexicale WordNet, il vise à intégrer les données structurées et semi-structurée de manière semi-automatique, dans ce système un adaptateur est associé à chaque source de données, cet adaptateur vise à traduire les descriptions des métadonnées vers une représentation commune basée sur le modèle (ODL-I3).

- **Information manifold**

Ce système décrit de manière déclarative le contenu des sources d'information. Le contenu des sources est décrit par des requêtes sur un ensemble de relations et de classes. Des relations et des classes virtuelles jouent le rôle de schéma global sur lequel l'utilisateur pose ses requêtes.

Ce système suit l'approche LAV où les relations sont décrites au niveau des sources sous forme de requêtes à travers les relations du schéma global.

Le plan d'exécution d'une requête est généré en utilisant un algorithme composé en deux phases : dans la première un plan sémantique est généré sous forme de requête conjonctivite qui utilise les relations des sources et qui est contenue dans la requête de l'utilisateur.

- **Infomaster**

Permet l'accès à diverses sources hétérogènes. Il utilise le format d'échange de connaissances (KIF : knowledge interchange format) comme langage pour décrire le contenu. Il considère trois types de relations : 1-relations d'interface, utilisées pour formuler les requêtes de l'utilisateur. 2-relations des sources qui décrivent les données stockées au niveau des sources. 3-relations globales, qui représentent un schéma de référence.

6. Modélisation d'un entrepôt

Lorsqu'on fait un schéma de BDD pour un système d'information classique, on parle en termes de tables et de relations, une table étant une représentation d'une entité et une relation une technique pour lier ces entités, c'est une modélisation par sujet. Et bien en BI, on parle de la modélisation multidimensionnelle, elle alternative au modèle relationnel et mieux pour bien préparer les données pour l'analyse (décisions).

6.1. Vocabulaire associé à la modélisation multidimensionnelle

Dans modélisation multidimensionnelle, chaque modèle se compose d'une table disposant une clé multiple appelée *table des faits* et d'un ensemble de tables appelé *table des dimensions*.

6.1.1. Table des faits

Dans un modèle multidimensionnel, la table de faits stocke *les mesures* de performances résultant des événements de processus de gestion d'une organisation. Un fait est constitué de plusieurs mesures relatives au sujet traité. Chacune de ces mesures

sont numériques et généralement valorisées de façon continue et prise à l'intersection de toutes *les dimensions*. [1]

6.1.2. Table des dimensions

Le sujet analysé, c'est-à-dire le fait, est analysé selon différentes perspectives. Ces perspectives correspondent à une catégorie utilisée pour caractériser les mesures d'activité analysées, nous parlons de dimensions. Les attributs des tables de dimensions jouent un rôle crucial, ils sont textuels et discrets. Les dimensions sont organisées en hiérarchies pour permettre l'analyse des mesures à différents niveaux de détail. [10]

6.2. Modélisation logique des données

A partir de table des faits et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Cette dernière repose sur le concept de CUBE (ou hypercube) pour représenter les données et obtenir des informations déjà agrégées selon les besoins de l'utilisateur: simplicité et rapidité d'accès. Ce cube organise les données en une ou plusieurs dimensions qui déterminent une mesure d'intérêt.

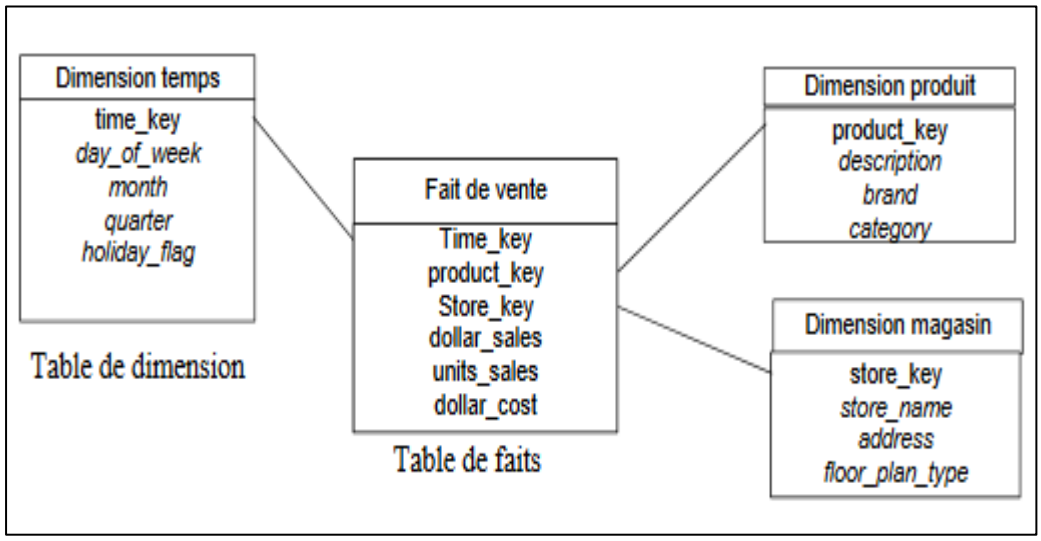

Figure 1.9: Un exemple de modèle multidimensionnel d'une vente. [23]

Nous distinguons deux groupes possibles pour la modélisation d'un entrepôt: le schéma en étoile et le schéma en flocon.

6.2.1. Schéma en étoile

Dans un schéma en étoile, une table centrale de faits contenant les faits à analyser, référence les tables de dimensions par des clefs étrangères. Chaque dimension est décrite par une seule table (feuille de l'arbre de tables) [10]

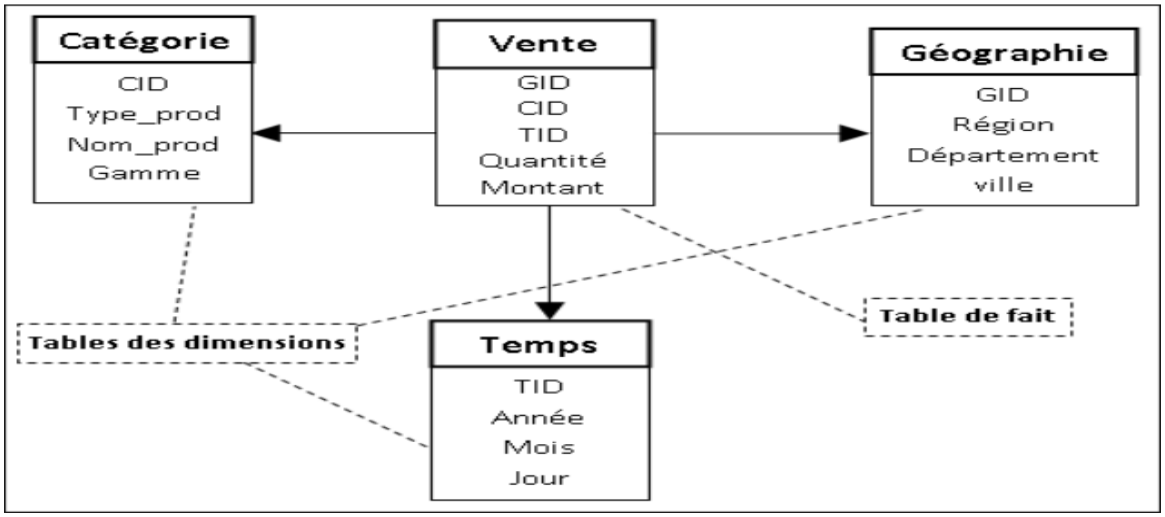

Figure 1.10: Le schéma en étoile. [23]

6.2.1.1. Avantages

- Facilité de navigation.
- Performances: nombre de jointures limité.
- Fiabilité des résultats.

6.2.1.2. Inconvénients

- Toutes les dimensions ne concernent pas les mesures.
- Redondances dans les dimensions.
- Alimentation complexe.

6.2.2. Schéma en flocon

Le schéma en flocon est une extension du schéma en étoile. Les informations dans cette dernière associées à une hiérarchie de dimension, sont représentées dans une seule table, même si les différents niveaux de la hiérarchie ont des propriétés différentes. Ce schéma évite les redondances d'information mais nécessite des jointures ce qui augmente son temps de réponse lors des agrégats de ces dimensions. [10]

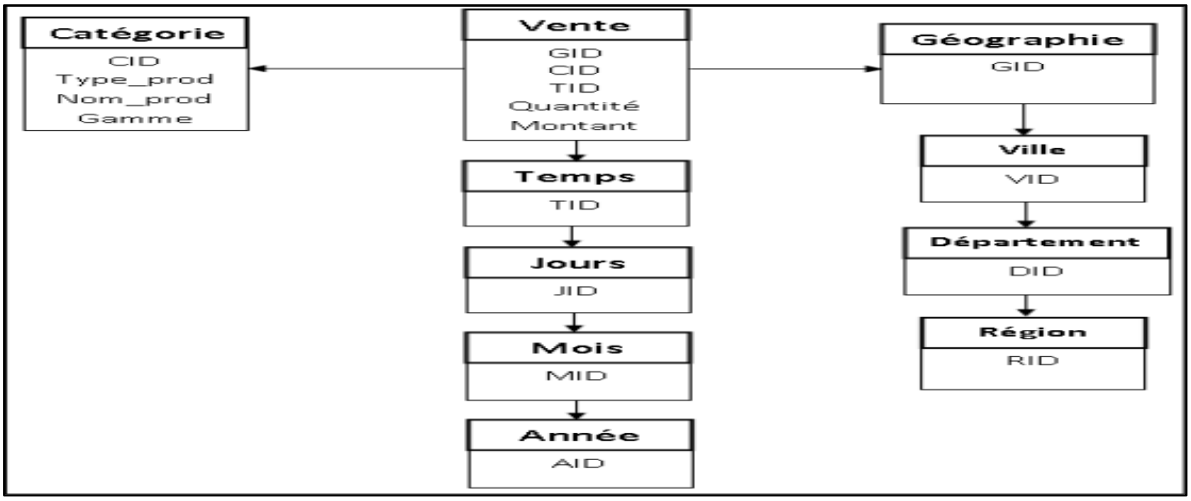

Figure 1.11: Le schéma en flocon. [10]

6.2.2.1. Avantages

- Evite les redondances d'information mais nécessite des jointures lors des agrégats de ces dimensions.
- Normalisation.

6.2.2.2. Inconvénients

- Moins de performance (grand nombre de jointure).
- Complexité de requête haute.

7. Différences entre entrepôt de données et base de données

L'objectif fondamental d'un ED est de stocker les données pertinentes aux besoins de prise de décision. Contrairement aux bases de données opérationnelles qui sont conçues pour supporter des opérations journalières. Le tableau suivant (Tableau 01) récapitule les principales différences entre les BDDs opérationnelles et les EDs.

Caractéristiques	Base de données	Entrepôt de données
Données	Actualisation de données stockées.	Historisation de données stockées.
Usage	Les opérations quotidiennes (la gestion courante).	Les besoins d'information à long terme et aide à la décision.
Principe de conception	Conception troisième forme normale.	Conception multidimensionnelle.
Requêtes	Simples, prédéterminées, données détaillées.	Complexes, spécifiques, agrégations et group by.
Utilisateur	Employés.	Analystes, décideurs.
Opérations	Consultation, mais surtout mise à jour et ajout de données.	Consultation de données uniquement.
Taille	Plusieurs giga-octets.	Plusieurs téraoctets.

Tableau 1.1: Différences entre BDD et ED. [24]

8. Objectifs d'un entrepôt de données

Les objectifs fondamentaux de l'entrepôt de données comme suit: [7]

- Rendre accessibles et cohérentes les informations de l'entreprise.
- Sécurise les informations de l'entreprise.
- Sert de stockage des informations.
- Retrouver et analyser l'information selon plusieurs critères.
- Utilisée lors de la prise de décision.
- Comporter un ensemble d'outils de requêtes, d'analyse et de présentation de l'information.
- Intégrer les informations récoltées et les stocker pour donner à l'utilisateur une vue orientée métier.

- Offre une information compréhensible, utile, rapide et à jour.

9. Fonctions d'un entrepôt de données

Les entrepôts de données mettent en œuvre cinq fonctions fondamentales qu'un entrepôt doit assurer [25]:

- **Collection**

La collecte des données brutes dans leurs environnements d'origine, ce qui implique des activités plus ou moins élaborées de détection, de filtrage et de transfert de données, car un excédent de données, un défaut de fiabilité ou un trop mauvais rapport signal/bruit sont pires que l'absence de données.

- **Intégration**

L'intégration des données, c'est-à-dire leur regroupement en un ensemble technique, logique et sémantique homogène approprié aux besoins de l'organisation.

- **Diffusion**

La diffusion, ou la distribution d'informations élaborées à partir des données collectées, dans des contextes appropriés, et ceci est aux profits des besoins des individus ou des groupes de travail de l'entreprise.

- **Présentation**

La présentation, c'est-à-dire les conditions de la mise à disposition de l'information -format- de façon claire, compréhensible et légale (IHM, contrôle d'accès,…).

- **Administration**

L'administration gère le dictionnaire de données et le processus d'alimentation de bout en bout, afin d'assurer un pilotage et un contrôle des données du système d'information décisionnel.

10. Outils d'implémentation l'entrepôt de données

Les entreprises indispensables de faire l'acquisition d'un logiciel de l'entreposage des données. Il est difficile de choisir un outil d'implémentation l'ED pour prendre une bonne décision car il existe plusieurs outils d'implémentation les EDs.

Dans cette partie, nous allons présenter les différents outils d'implémentation ED le plus utilisé.

10.1. SQL Server

SQL Server est un système de gestion des bases de données (SGBD) et aussi outils d'entreposage de données en langage SQL, ce qui lui confère une très grande capacité à gérer les données tout en conservant leur intégrité et leur cohérence. SQL Server est chargé de :

- Stocker les données.
- Vérifier les contraintes d'intégrité définies.
- Garantir la cohérence des données qu'il stocke, même en cas de panne (arrêt brutal) du système.
- Assurer les relations entre les données définies par les utilisateurs.

SQL Server prend en charge les environnements 32bits et 64bits et parfois appelé MSSQL et Microsoft SQL Server [26].

10.2. Oracle

Oracle est un système de gestion de base de données et aussi un outil d'entreposage de données, développé en langage C. Depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Depuis la version 8.0.5 est disponible sous Linux.

Oracle permet d'offrir les fonctionnalités suivantes:

- La sauvegarde et la restauration des données.
- La gestion des accès concurrents.
- La confidentialité des données.
- L'intégrité des données [10].

10.3. Pentaho

Pentaho est une plate-forme décisionnelle open source, développé en Java, utilisé pour la migration de données d'une base à une autre et l'alimentation d'un data warehouse et de data marts d'une part et destiné au domaine de l'informatique décisionnel d'autre part. Donc porte tous les fonctionnalités de ce dernier:

- ETL (intégration de données).
- Reporting.
- Tableaux de bord.
- Analyse ad-hoc (requêtes à la demande).
- Analyse multidimensionnelle (OLAP) [27].

Pentaho a l'avantage de fonctionner sur Windows, Linux ou Mac-OS. Ses outils peuvent être des applications et des plug-ins. Nous pouvons différencier deux types d'applications dans Pentaho :

- Les applications serveurs, utilisables via une console.
- Les applications client, dans ce cas, l'application ne sera que utilisable via un bureau [28].

10.4. Talend Open Studio (TOS)

Talend Open Studio est un outil d'entrepôt de données Open Source spécialisé pour l'intégration (ETL), développé en java et apparu en 2005. Ainsi, c'est un générateur de code (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 ou Perl).

Talend offre deux produits d'intégration de données:

- Talend Open Studio for Data Integration

- Talend Enterprise Data Integration qui intègre des fonctionnalités avancées de déploiement et de gestion distribué sous licence commerciale. Ce type de produit complète les fonctionnalités de Talend Open Studio avec des fonctionnalités d'entreprise comme:

✓ Référentiel partagé pour le travail collaboratif.

✓ Outils de gestion et de monitoring pour déployer et superviser les traitements [29].

11. Domaines d'application des entrepôts

Plusieurs domaines qui utilisent les entrepôts de données. Dans la suite, nous donnons quelques domaines d'application des entrepôts [30]:

- **Domaine bancaire**

Pour une banque, il est important de pouvoir regrouper les informations relatives à un client afin de répondre à ses demandes de crédit par exemple.

Des mailings ciblés doivent aussi être rapidement élaborés à partir de toutes les informations disponibles sur un client lors de la commercialisation d'un nouveau produit.

L'utilisation des cartes crédits nécessite des contrôles à posteriori, par exemple pour la recherche de fraudes: la mémorisation des mouvements peut rendre de grands services

Les échanges des actions et des conseils de courtages sont facilités par une mémorisation d'histoire et une exploitation par des outils décisionnels avancés par exemple pour déterminer des tendances de marchés.

- **Domaine de la grande distribution**

Intéressant de regrouper les informations de ventes pour déterminer les produits à succès, mieux suivre les modes, détecter les habitudes d'achats, les préférences des clients par secteur géographique.

La fouille de données (Data Mining) a permis de développer des techniques

Sophistiquées d'exploitation de données qui aident à mettre en évidence les règles de consommation.

Explorer le panier de la ménagère est devenu un exercice d'école: il s'agit de trouver à partir de l'enregistrement des transactions quelles sont les habitudes d'achats, plus précisément quels sont les produits achetés en même temps.

Prédiction de ventes en fonctions de données conjoncturelles, gestion des stocks, des approvisionnements.

Contrôle qualité et analyse de défaut des chaînes de production en fonction des centres de production, des organisations, des fournisseurs... [31].

- **Domaine des télécommunications**

Grande masse de données concernant les abonnés et les appels est enregistrée.

Plusieurs mois de description détaillée des appels comprenant, pour chaque appel appelant, appelé, heure et durée sont disponibles chez les opérateurs.

En respectant les lois de sécurité et liberté, que peut-on faire de telles données?

Couplées ou non avec des informations comptables, l'exploitation de ces données regroupées en ED par des techniques d'analyse et d'exploration permet:

- ✓ D'analyser le trafic.
- ✓ De mieux cerner les besoins des clients.

- ✓ De classer les clients par catégories.
- ✓ De comprendre pourquoi certains changent d'opérateurs et mieux répondre à leur besoins.

- **Domaines de l'assurance et de la pharmacie**

L'exercice de base de l'assureur est de déterminer le facteur de risque d'un assuré.

Celui d'un producteur pharmaceutique est de détecter l'impact d'un médicament.

Plus généralement, le suivi des informations relatives à la liaison produit-client sur un ED est souvent synonyme de gains importants: meilleure connaissance des produits, détection des défauts, meilleure connaissance des clients, détection de rejets, ciblage du marketing, etc…

Le couplage aux technologies du Web ouvre aussi des horizons nouveaux pour le suivi des produits, des clients, des concurrents: notion émergente de "Data Webhouse".

Conclusion

Ce chapitre présent les concepts de base d'entrepôt de données (Data Warehouse), les différents types d'hétérogénéité de données et les approches d'intégrations de données, et on fait une petite comparaison entre la base de données et l'entrepôt de données. En fin, on présente les outils pour implémenter les entrepôts de données et les domaines d'application ED.

Dans le système d'intégration, l'hétérogénéité des sources de données reparties rencontrent un problème d'accès à multiples sources de données. La plupart des solutions se basent sur l'utilisation des ontologies. Dans le chapitre suivant nous allons détailler cette approche.

Chapitre 02: Construction d'un entrepôt de données

Introduction

L'information se trouve répartie dans de nombreuses bases de données, souvent hétérogènes. Un des enjeux majeurs dans l'exploitation de ces données consiste à pouvoir interroger différentes sources de données sans nécessairement en connaître la structure exacte.

Dans le domaine des bases de données, la notion d'interrogation répartie des sources hétérogènes est actuellement un domaine de recherche très actif. Ils ont conduit à une architecture fondée sur la notion de médiateur, d'entrepôt de données ou bien hybride.

Par ailleurs, avec l'apparition les domaines variée tels que le web sémantique, le traitement de la langue naturelle, la recherche d'information, l'intégration des données...etc. La solution pour réduire les hétérogénéités sémantiques des sources d'information et rendre leur contenu explicite qui peuvent apparaitre lors de l'intégration c'est l'utilisation les approches ontologiques. Ces dernières alors permettent l'identification et l'association de concepts sémantiquement correspondants.

Plusieurs formes et plusieurs langages existent pour représenter les ontologies. Ces dernières s'intéressent à la notion d'existence et offrent une description structurelle et sémantique des données ainsi elles offrent un moyen pour décrire de manière formelle les connaissances d'un domaine particulier, en définissant des concepts et des relations entre ces concepts.

Dans ce chapitre, nous présentons les différentes approches de construction entrepôt de données ainsi nous définissons la notion d'ontologie, les différents composants d'une ontologie et une méthodologie pour construire les ontologies.

1. Approche de construction un entrepôt de données

La communauté bases de données a beaucoup investi dans le développement des outils pour l'intégration des sources de données hétérogènes distribuées ont été proposées, en l'occurrence des approches de type entrepôts de données (intégration matérialisé) et de type médiateur/wrapper (intégration virtuelle). Le tableau (tableau 02) suivant résume les différentes approches de construction un entrepôt de données et nous définissons les avantages et les inconvénients de chaque approches.

	Avantages	**inconvénients**
Entrepôt	Plus de performance (moins de délai).	Données pas toujours fraîches.
	Personnalisation des données (nettoyage, filtrage).	Gestion de gros volumes de données en termes de stockage (très couteuse).
	Interrogation s'effectue comme sur une BD classique.	Gestion des mises-à-jour.
Médiation	Pas duplication des données (Les données restent au niveau source).	Traduction des requêtes.
	Les données à jours	Moins performances (délai).
	Passage à l'échelle.	Capacités et nature différente des sources.

Tableau 2.1: Avantages et inconvénients des approches d'intégration de données [32].

- **Discussion**

L'intégration de l'information de plusieurs sources de données est effectuée sur la base des besoins des utilisateurs pour une fin donnée. Ces exigences sont représentés par une vue appelée vue intégrée. Il existe plusieurs approches pour

construire l'ED, ces différentes approches peuvent être classées en trois domaines principaux: matérialisée et virtual.

En plus de cela, avec l'existence de plusieurs conflits de l'intégration de donnée. Afin de résoudre quelques conflits, l'utilisation d'une ontologie apparait comme une solution assurant l'automatisation du processus d'intégration sémantique de données.

Dans la suite nous définissons l'ontologie et leurs composants ainsi que l'intégration à base de cette approche.

2. Définition d'ontologie

Plusieurs définitions ont été proposées au fur et à mesure du développement et de l'enrichissement des ontologies [21].

Le terme « ontologie » est apparu pour la première fois selon le dictionnaire anglais de Bailey OED (Oxford English Dictionary), en 1721.

Définition 01: ontologie dans le domaine de philosophie

Ontologie a été définie tout d'abord dans le domaine de la philosophie où on s'intéresse à l'existence, c'est-à-dire les éléments qui peuvent exister, de la nature des objets, des propriétés, des évènements, des processus et des relations dans chaque domaine de réalité. Le mot «ontologie» veut dire «la science de l'être», (onto: l'être ou l'existent et logia: science ou univers).

Définition 02: ontologie dans le domaine d'informatique

Des chercheurs en informatique, plus spécifiquement en intelligence artificielle (IA) se sont inspirés de cette notion d'ontologie pour la représentation et la formalisation des connaissances par l'utilisation de vocabulaire, plusieurs définitions ont été données, la plus célèbre est celle de Gruber qui énonce "*une ontologie est une spécification explicite d'une conceptualisation d'un domaine de connaissance* " où le terme "*conceptualisation*" réfère à un modèle abstrait d'un certain phénomène de la réalité et qui permet d'identifier de manière structurée les concepts pertinents de ce

phénomène. L'expression "*spécification explicite*" signifie que les concepts utilisés et les contraintes sur leur usage sont définis d'une manière explicite.

Nous définissons une ontologie comme suit:

Une ontologie fournit un vocabulaire commun d'un domaine et définit le sens des concepts et les relations entre ces concepts.

Même si les ontologies peuvent avoir certaines divergences, elles sont toujours les mêmes: une ontologie est constituée de concepts et de relations ainsi que des propriétés et des axiomes.

3. Composant d'ontologie

La notion d'ontologie repose sur les termes suivants: les concepts, les relations, les fonctions, les axiomes et les instances. Voici quelques précisions sur ces éléments [21]

✓ **Concepts**

Appelés aussi termes ou classes, ils constituent les éléments de base au sein d'une ontologie et correspondent à l'ensemble des individus à représenter. Il représente des objets réels du monde réel, une idée…etc.

✓ **Relations**

Les relations représentent les interactions qui peuvent exister entre les concepts présents dans l'ontologie considérée et définies comme un sous ensemble d'un produit cartésien de n ensembles. Ces relations incluent les relations de spécialisation-généralisation (sous classes), relations d'agrégation ou de composition (partie de), relations d'association et d'instanciation.

✓ **Fonctions**

Les fonctions sont des cas particuliers des relations dans lesquelles le nième élément de la relation est défini en fonction des n-1 éléments précédents c-à-dire F : C1 x C2... xCn-1 → Cn.

✓ **Axiomes**

Les axiomes sont des expressions qui sont toujours vrais, permettant de combiner des concepts, des relations et des fonctions afin de définir des règles d'inférences.

✓ **Instances**

Les instances sont des extensions des concepts de l'ontologie. Ils représentent les éléments qui véhiculent les connaissances du domaine considéré. Appelé aussi les individus.

4. Classification des ontologies

Il existe plusieurs classifications des ontologies. On peut citer une de ces classifications, proposée par GUARINO et PIERRA où les ontologies sont classées en fonction de leur niveau de conceptualisation [33]:

4.1.Classification de Guarino

Guarino proposé plusieurs classification comme suit:

4.1.1. Ontologie supérieure

Ontologie supérieure sont des ontologies génériques applicables dans des domaines variés. Leur but est l'étude des catégories des choses qui existent dans le monde comme les concepts de plus haut niveau d'abstraction et de généralité, par exemple: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés et qui sont décrits indépendamment d'un domaine particulier.

Appelées aussi ontologies génériques, Upper ou encore Top-level ontologies.

4.1.2. Ontologie de domaines

Ontologie de domaines s'attachent à décrire un domaine précis (géographie, médecine, énergie,…etc.). Le niveau d'abstraction est moins élevé que dans les ontologies globales, elles vont apporter une spécialisation des concepts généraux des ontologies globales.

4.1.3. Ontologie de tâches

Ces ontologies sont utilisées pour gérer des tâches spécifiques à un domaine afin de résoudre les problèmes du système, le vocabulaire qu'elles décrivent est relié à une tâche générique telle que les tâches de classification, et cela à travers la spécialisation des termes dans les ontologies de haut niveau. Les ontologies de tâche permettent de fournir un vocabulaire systématique des termes utilisés pour résoudre les problèmes liés aux tâches.

4.1.4. Ontologie d'applications

Ontologie d'applications sont des ontologies spécifiques à un champ d'application dans un domaine donné, ce dernier pouvant être décrit par une ontologie globale ou de domaine.

4.2.Classification de Pierra

Une autre classification des ontologies est celle de Pierra qui classifie les ontologies en ontologies linguistiques et ontologies conceptuelles.

4.2.1. Ontologie linguistiques

Les ontologies linguistiques ont pour but la représentation de la signification des termes d'un domaine particulier dans un langage précis, permettant de fournir une représentation en langage naturel des concepts d'un domaine.

4.2.2. Ontologie conceptuelles

Ontologie conceptuelles permettent de définir de manière formelle les concepts d'un domaine et les relations entre les concepts, elles-mêmes sont catégorisées en deux types:

- *Ontologies conceptuelles canoniques(OCC)*: les concepts décrits et utilisés sont des concepts primitifs. Ils possèdent une représentation unique et indépendante des autres concepts. Ils sont utiles pour la conception de base de données afin d'éviter les redondances et de permettre la création de formats d'échanges.
- *Ontologies conceptuelles non-canoniques(OCNC)*: les concepts utilisés sont des concepts primitifs et des concepts définis. Les concepts peuvent donc être reliés par des relations d'appartenance, d'équivalence, etc., ce qui permet de réaliser des déductions. Ces ontologies sont à ce titre utilisées dans les domaines de l'intelligence artificielle.

5. Langage de représentation des ontologies

Plusieurs langages de représentation d'ontologies ont été développés. Dans cette section, nous faisons une rapide revue de ceux qui nous paraissent très représentatifs parmi les standards et recommandations du World Wide Web Consortium (W3C): RDF/RDFS, OWL, SPARQL [34].

5.1.Resource Description Framework (RDF)

RDF est un langage qui permet de définir une ontologie de manière très simple et qui représente un modèle conceptuel, abstrait et formel pour la représentation des ressources et les relations entre elles. C'est une recommandation du W3C pour décrire des ressources et leurs métadonnées afin de faciliter leur traitement automatique. Basé sur la syntaxe XML[1]. Le langage RDF est fondé sur un modèle de graphe, permettant de décrire les éléments de manière simple et sans ambiguïté selon un mécanisme basé sur des déclarations RDF. Ces dernières sont des triplets (ressource, attribut, valeur)

[1]eXtended Markup Language.

qui peuvent être traitées par la machine pour permettre à celle-ci de le faire tout en comprenant la signification de ces triplets. Par exemple, l'assertion l'enseignant E1 enseigne le cours C3 peut être représentée par le triplet <E1, enseigner, C3> où E1est le sujet, enseigner une propriété et C3 la valeur de la propriété pour ce sujet. Chaque élément du triplet est identifié par un URI[2]. Cette identification se fait de manière unique à l'aide d'un nom sans avoir à localiser la ressource. Les triplets sont interprétables comme sujet-prédicat-objet où le prédicat exprime la propriété.

L'objectif initial de RDF est une bonne représentation et une meilleure exploitation des métadonnées. Mais son inconvénient est qu'il ne supporte pas la vérification de la cohérence des données par exemple: vérification que le champ «date de naissance» est vraiment une date.

5.2.Resource Description Framework Schema (RDFS)

RDFS est un schéma de base incluant les entités sémantiques généralement utilisées (classes, sous-classes, propriétés, sous-propriétés, etc.) pour la structuration des connaissances d'un domaine. RDFS est une extension de RDF permettant de définir des hiérarchies de classes et des propriétés et permet l'implémentation du modèle RDF pour la définition des ontologies avec une approche orientée objet.

Trois notions principales permettant la définition des primitives: la ressource (rdfs:Resource), la classe (rdfs:Class) et la propriété.

Le langage RDFS offre en plus du contrôle la terminologie et la structure des descriptions RDF, la possibilité de raisonner sur liens de types «est-un» (is a) qui existent entre les concepts et les propriétés.

L'avantage de RDFS est pouvoir créer une hiérarchie de classes et de propriétés, grâce aux notions de subClassOf et subPropertyOf. Et donc on peut par la suite instancier des classes en utilisant le rdf:type. Cependant, ils souffrent de quelques limites puisqu'ils ne permettent pas d'exprimer:

[2] Uniform Resource identifier.

- des propriétés algébriques comme *la transitivité, la symétrie*. Par exemple, aLeMêmeGradeQueest symétrique, aDeMeilleursRésultatsQueest transitive.
- La combinaison booléenne de classes avec des opérateurs d'*union*, d'*intersection* et de complémentarité. Par exemple, EnseignantChercheur est l'union disjonctive d'ATER, MaîtreDeConférenceset Professeur.
- la disjonction entre classes: dire que deux classes sont disjointes. Par exemple, PersonnelEnseignant et PersonnelAdministratif sont disjoints.
- la restriction de cardinalités: c'est-à-dire le nombre de valeurs qu'une propriété donnée peut avoir. Par exemple, un Enseignant enseigne au moins un Cours.

Ainsi, ces deux langages ne permettent pas de représenter ce type d'axiomes et de les utiliser pour effectuer des déductions. En conséquence, un autre langage, OWL (Web Ontology Language), a été proposé pour combler ce manque.

5.3.Web Ontology Language (OWL)

OWL est un langage qui a été créé par le W3C en 2004, il hérite du langage DAML+OIL. C'est le langage le plus expressif des autres langages. Il est dédié à définir les classes et les types de propriétés. Il permet ainsi d'exprimer des propriétés telles que l'équivalence, la disjonction entre classes, la cardinalité, la symétrie, la transitivité d'une relation, etc. C'est un langage inspiré des logiques de description qui offre aussi de nouvelles primitives permettant de définir des classes à l'aide de mécanismes ensemblistes (intersection de classes, union de classes, complément d'une classe). Il permet également d'effectuer des déductions.

C'est un langage qui permet de fournir des mécanismes pour créer tous les composants de l'ontologie c'est-à-dire les classes, les instances, les propriétés et les axiomes. Comme dans le langage RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés.

OWL est doté de trois sous-langages de plus en plus expressifs: OWL Lite, OWL DL et OWL Full. OWL DL est défini comme une extension d'OWL Lite, et OWL Full comme une extension d'OWL DL.

- *Langage OWL Lite* peut être vu comme une extension du langage RDFS, mais avec moins de fonctionnalités. Il contient un ensemble réduit de constructeurs en fournissant l'essentiel pour la construction d'une hiérarchie de classes. C'est le sous langage d'OWL le plus simple. Son principe est de permettre une modélisation d'ontologies moins compliquées, afin de faciliter l'implémentation des raisonneurs corrects et complets.
- *Langage OWL DL* contient des constructeurs supplémentaires, mais avec des contraintes bien particulières, il ne peut être utilisé qu'avec certaines restrictions. Il est plus complexe que OWL Lite et est fondé sur la logique de description d'où son nom, «OWL Description Logics». Malgré cette complexité par rapport à OWL Lite, il garantit la complétude des raisonnements et leur décidabilité.
- *Langage OWL Full* est le plus complexe des sous-langages d'OWL, il dispose de tous les constructeurs du langage OWL permettant ainsi une interprétation plus large. Il permet de traiter les classes comme des individus. OWL Full est utile typiquement pour les gens qui veulent combiner l'expressivité du langage OWL avec la flexibilité et méta-modélisation des caractéristiques de RDF afin d'avoir un haut niveau de capacité de description, néanmoins, son utilisation ne garantit pas toujours la complétude du raisonnement.

5.4. Protocol and RDF Query Language (SPARQL)

Comme le langage SQL, SPARQL est un langage de requêtes destiné à interroger les bases de triplets RDF, appelées aussi triple stores. Il est devenu une recommandation du W3C depuis janvier 2008. Il utilise des patterns de graphe pour déterminer les triplets qui satisfont les conditions des requêtes. En principe, avec ce langage, on peut accéder à toute donnée du Web représentée au format RDF. SPARQL

utilise une syntaxe inspirée de SQL et est à ce titre très similaire à ce langage. Le schéma général d'une requête SPARQL est de la forme suivante:

```
# déclaration de préfixes
PREFIX foo: <http://example.com/resources/> ...
# définition des jeux de données
FROM ...
# clause résultat
SELECT ...
# motif de la requête
WHERE { ... }
# modificateur de requête
GROUP BY
```

Figure 2.1: Schéma générale de requête SPARQL [34].

Une requête SPARQL permet d'extraire des triplets d'un graphe RDF vérifiant certaines conditions définies dans sa clause **where**. SPARQL possède également d'autres clauses telles que les opérateurs booléens (union, intersection), de filtrage sur les valeurs pour exprimer des requêtes plus complexes mais plus spécifiques. SPARQL est souvent utilisé conjointement avec des langages de programmation comme Java pour manipuler des données RDF dans des applications.

6. Intégration de données à base ontologie

L'intégration sémantique de données permet d'éliminer les conflits existent entre les sources hétérogène. La plupart des solutions actuelles reposent sur l'utilisation des ontologies afin d'automatiser le processus d'intégration. Donc l'ontologie est une méthode de représentation un modèle pour interroger les sources de données. Plusieurs approches ont été étudiées. Elles sont fondées sur l'utilisation soit d'une seule ontologie, soit de plusieurs, ou également d'une approche hybride.

6.1.Approches des ontologies dans l'intégration de données

Ontologies permettent l'identification et l'association de concepts sémantiquement correspondants. Plusieurs approches ont été étudiées. Elles sont fondées sur l'utilisation soit d'une seule ontologie, soit de plusieurs, ou également d'une approche hybride.

6.1.1. Approche avec une seule ontologie

Cette approche utilise une seule ontologie partagée, chaque source à intégrer est liée à une seule et même ontologie globale [21].

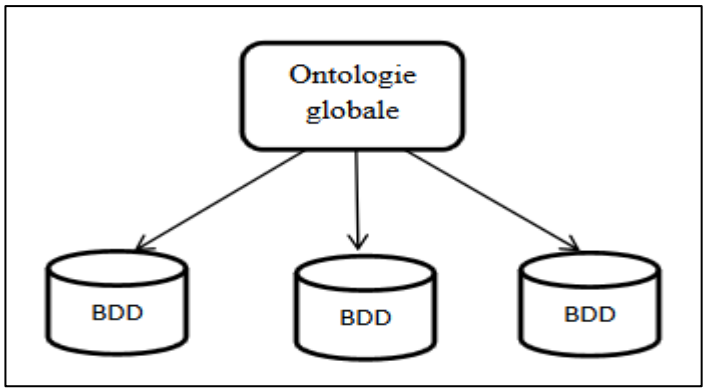

Figure 2.2: Approche avec une seule ontologie [21].

6.1.2. Approche avec plusieurs ontologies

Dans cette approche, chaque source d'information est décrite par sa propre ontologie, indépendamment des autres sources. Le lien entre différentes ontologies des sources permettant d'identifier les correspondances sémantiques ente les termes de ces ontologies [21].

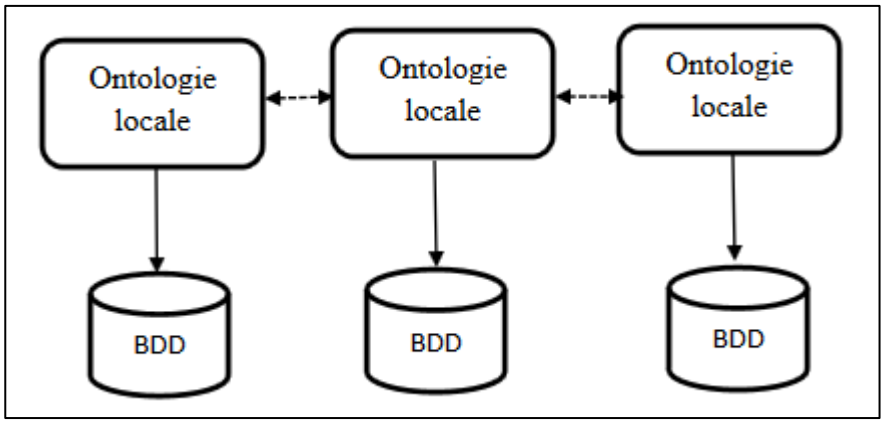

Figure 2.3: Approche avec plusieurs ontologies [21].

6.1.3. Approche hybride

Cette approche est combiné les deux autre approche c-à-dire chaque source possède sa propre ontologie de description mais, afin de permettre de rendre chacune comparable avec une autre, elles se réfèrent toutes à un vocabulaire partagé. Ce vocabulaire comprend des termes fondamentaux d'un domaine qui peuvent être combinés avec ceux présents dans les ontologies locales pour décrire des sémantiques plus complexes. Parfois le vocabulaire partagé est également une ontologie [35].

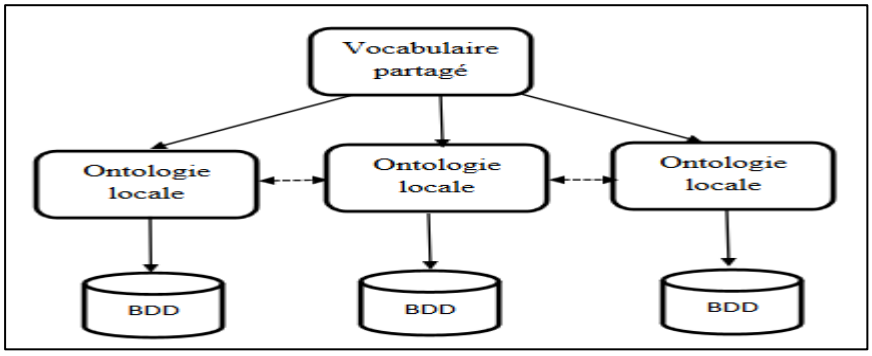

Figure 2.4: Approche hybride [21].

6.2. Avantages d'intégration à base ontologique

Les avantages des méthodes des ontologies illustres dans les points suivants [36]:

- Réutilisation des ontologies déjà existent.
- Fournir une représentation des connaissances dans un domaine.
- Réutilisée et partagée par divers applications et groupes.
- Ontologie définie dans un langage traitable par machine.
- Possibilité de référencer de manière unique les concepts ontologiques sans importation des définitions de ces derniers.
- Compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Assuré l'interopérabilité entre systèmes (machine et système).

- Supporté les requêtes communes: questions sur la véracité des déclarations, requête attendant un objet a envoyé.
- Permettre l'échange de connaissances entre systèmes.

6.3. Inconvénients de méthodes ontologiques

Les inconvénients de méthodes ontologiques [36]:

- Les ontologies peuvent changer pour des raisons diverses: lorsque les besoins des utilisateurs changent, nécessitant une conceptualisation différente.
- Changement l'évolution des ontologies.
- Problème des changements dans une ontologie pour les données instanciées qui sont dépendants de l'ontologie, ainsi que pour toutes les ontologies qui importent cette ontologie modifiée.
- Défaut de conception peut avoir été remarqué dans la conceptualisation d'origine, en appliquant des corrections de modélisation, ou bien encore il peut s'avérer nécessaire d'élargir la représentation du domaine.

7. Etapes de construction l'ontologie

Les ontologies réalisées sont très différentes les unes des autres. Il existe plusieurs méthodologies pour construire l'ontologie tels que METHONTOLOGY (Fernández-López, et al., 1997), TOVE et SENSUS (Swartout, et al., 1997). Nous avons construire l'ontologie à partir de méthodologie *METHONTOLOGY* proposé par Fernandez ses collègues (Fernández-López, et al, 1997). Ce dernière est permettant la construction d'ontologie à partir de zéro et nécessite un professionnel informatique avec la validation d'un spécialiste du domaine étudié (commerçant) et s'applique à clarifier les différentes étapes de la construction en respectant des activités de gestion de projets (planification, assurance qualité), de développement (spécification, conceptualisation, formalisation, implémentation, maintenance) et des activités de support (intégration, évaluation, documentation).

Le procédé que nous allons adopter pour construire l'ontologie se fera en 4 grandes phases: spécification, conceptualisation, formalisation, implémentation [37].

✓ **Etape 01 (Spécification)**

Pour commencer le développement de l'ontologie, une première importante étape doit être effectuée. Elle consiste à établir un document informel de spécification de besoins écrit dans un langage naturel. Nous décrivons dans ce document:

- Le domaine de connaissance qui sera représenté par l'ontologie.
- L'objectif de l'ontologie à créer pour le domaine considéré.
- Les utilisateurs futurs de l'ontologie.
- Les sources d'informations des quelles les connaissances seront obtenues. Ils sont de nature différentes et variées, par exemple: les interviews avec les experts du domaine, les documents techniques (publications scientifiques, livres), les observations, les ontologies existantes qui peuvent être réutilisés...etc.
- La portée de l'ontologie: déterminer la liste des termes candidats du domaine à travers l'analyse des sources d'informations relatif au domaine.

✓ **Etape 02 (Conceptualisation)**

L'étape de conceptualisation est la plus importante dans le processus de développement de l'ontologie. Elle détermine le reste de la construction de l'ontologie et spécifie de manière détaillé. Elle consiste à organiser et à structurer, à partir des sources d'informations, les connaissances du domaine sous forme de tableaux et graphes, indépendamment du mécanisme de formalisation utilisé pour représenter l'ontologie. A la fin nous obtenons une ontologie conceptuelle (model conceptuel). Les principales tâches suivantes sont réalisées:

Tâche 01: construction du glossaire de termes. Il recueille et décrit tous les termes (concepts, instances, attributs, relations entre les concepts, etc.) qui sont utiles et potentiellement utilisables dans le domaine que nous allons représenter leurs descriptions détaillées et non ambiguës en langage naturel.

Tâche 02: classification des concepts en hiérarchies de concepts. Elle est réalisée à travers un choix d'une stratégie d'identification de concepts.

- Approche ascendante (bottom-up strategy): les concepts les plus spécifiques sont identifiés, par la suite, ils sont généralisés en concepts plus abstraits.
- Approche descendante (top-down strategy): les concepts les plus abstraits sont identifiés, par la suite, ils sont spécialisés en plus spécifiques.
- Approche centrifuge (middle-out strategy): les concepts les plus importants sont identifiés (centraux), par la suite, ils sont généralisés et spécialisés comme il est nécessaire.

En outre, METHONTOLOGY propose d'utiliser quatre relations taxonomiques qui sont sous-classe-de (Subclass-Of), décomposition-disjointe (Disjoint-Decomposition), décomposition-exhaustive (Exhaustive-Decomposition), et partition (Partition).

✓ Un concept C1 est sous classe d'un autre concept C2 si et seulement si chaque instance de C1 est aussi une instance de C2.

✓ Une décomposition disjointe d'un concept C est un ensemble de sous classes de C qui n'ont pas des instances en commun et ne couvrent pas C c.à.d. il peut exister des instances du concept C qui sont des instances d'aucun concept de la décomposition.

✓ Une décomposition exhaustive d'un concept C est un ensemble de sous classes de C qui couvrent C et peuvent avoir des instances en commun et des sous classes. Autrement dit, il ne peut pas exister des instances du concept C qui ne sont pas des instances d'au moins un concept de la décomposition.

✓ Une partition d'un concept C est l'ensemble de sous classes de C qui ne partagent pas des instances en commun et couvrent C.

Tâche 03: construction du glossaire de relations binaires où les relations entre les concepts de l'ontologie sont identifiés.

Tâche 04: Construction du glossaire d'attributs qui décrire les attributs apparus dans le dictionnaire de concepts en spécifiant leur contraintes.

Le résultat de cette étape de conception est un glossaire des termes, sous termes et relations qui représente le domaine étudié, le but maintenant et de coder cette ontologie en OWL DL pour obtenir une ontologie opérationnelle.

Conclusion

Dans ce chapitre, nous avons présenté les différentes approches d'intégrations de données. Après nous définissons la solution majeure pour résoudre le problème de conflit sémantique entre les sources de données distribues, ainsi on a décrire les notions liées aux ontologies, leurs classifications, langage de présentation. Enfin, nous détaillons les étapes de méthodologie manuelle pour la construction ontologie.

Dans le prochain chapitre, nous proposons notre architecture pour construire l'ED qui se base principalement sur la notion ontologie et détaillons les étapes de construction ontologie de domaine commercial.

Chapitre 03: Conception

Introduction

Avant d'avoir donné une vision générale sur l'architecture de notre application on va exploiter et présenter le domaine commercial, cette dernière joue un rôle essentiel dans la réussite des entreprises commerciales c.-à-d. toute entreprise profite le plus possible afin d'assurer sa continuité et accroître sa compétitivité.

Ce chapitre est consacré à la conception du système d'intégration des données hétérogènes par l'utilisation d'une Ontologie du Domaine Commercial "***OntoDC*** ". Tous d'abord nous allons présenter le domaine commercial et la raison de choisir ce domaine, après nous présentons une aperçu générale de l'architecture de notre système ainsi nous détaillons les étapes de construction d'une ontologie commercial "***OntoDC***".

1. Domaine commercial

Domaine commercial concerne toutes les opérations de gestion de stock et des vents d'une entreprise. Elle s'appuie sur le traitement et la gestion des données clients, fournisseurs et produits de l'entreprise.

C'est donc la comptabilité, l'administration des ventes, le service de gestion commerciale de l'entreprise qui gèrent les achats, la gestion des stocks, la gestion des clients, la gestion des fournisseurs,…etc.

1.1. Choix de domaine

Le choix de ce domaine est justifié par le rôle essentiel de la gestion commerciale dans la réussite des entreprises commerciales; vu que toute entreprise est tenue à suivre un développement rapide du monde de la technologie et en profitant le plus possible afin d'assurer sa réussite.

1.2. Les éléments influençant dans le domaine commercial

Dans la suite nous définissons quelques éléments qui influencent dans le domaine commercial tel que le commerce lui-même, climat et population.

1.2.1. Commerciaux

Toute entreprises quelle que soit son activité, doit veiller à assurer une bonne gestion de produit, gestion des clients, … etc.

1.2.1.1. Gestion des clients

Les bonnes gestions de clients permettent aux entreprises de mieux connaître leur clientèle et de gagner leur fidélité en utilisant les informations dont elles disposent, de manière à mieux cerner leurs besoins et donc de mieux y répondre. Fidéliser un client coûte beaucoup moins cher que trouver de nouveaux clients par la prospection [38].

La gestion des clients doit suivre une stratégie pour:

- Maximiser les occasions de fidéliser la clientèle en prévoyant ses besoins.
- Déterminer vos meilleurs clients.
- Trouver des clients éventuels.
- Déterminer les produits complémentaires à vendre à vos clients.
- Cibler des campagnes et des documents de marketing et des promotions.

1.2.1.2. Gestion des produits

La gestion des produits consiste à équilibrer le portefeuille de produits d'une entreprise. C'est ce qu'on appelle la gamme de produits: ensemble des produits de même nature qu'une entreprise offre sur son marché.

Elle vise particulièrement à gérer les différents produits afin de les adapter aux caractéristiques et aux évolutions du marché.

1.2.2. Population

Les théories classiques du commerce montrent que chaque population de pays à intérêt se spécialiser dans les productions pour lesquelles il est le plus productif. Ainsi, il exporte les biens ou services dans lesquels il s'est spécialisé et importe les produits fabriqués plus efficacement à l'étranger.

Grace à cette spécialisation, chaque pays peut augmenter sa productivité, donc la quantité de biens et services produits et consommés.

Le commerce peut être comparé à la situation des individus qui se spécialisent chacun dans la production d'un type de biens ou services. La hausse de la productivité permet d'accroître la production et la consommation.

1.2.3. Climat

On entend souvent que les aléas météo comme les fortes pluies, les variations de températures perturbent le secteur de la vente au détail. Certes, ces phénomènes ont un impact significatif sur le commerce.

Les différents secteurs du commerce de détail ont chacun leurs périodes propices qui sont essentielles à leur succès annuel. Par exemple, pour le textile, le printemps et l'automne sont les saisons idéales pour présenter les nouvelles collections. La réussite des ventes dépend en majeure partie du temps qu'il fait au moment de l'arrivée des nouvelles collections pour l'hiver ou l'été.

Pour tous ces secteurs, il ne s'agit pas seulement de savoir si le temps est chaud ou froid, ensoleillé ou nuageux, humide ou sec. L'enjeu majeur est de déterminer comment les consommateurs se comportent en fonction de la météo et d'être agile en s'adaptant au temps.

2. Caractéristiques du système

La prise de décision nécessite l'accès et la consultation à des types variés de données, généralement stockés sur des sites distincts. De plus, l'utilisateur (c'est-à-dire le décideur) ne sait pas toujours à l'avance la totalité des données qu'il doit consulter pour sa prise de décision, ni l'emplacement de ces données.

Le système que nous avons développé permet :

✓ L'interrogation des bases de données de l'entreprise par différents types d'utilisateur, sans pour autant que l'utilisateur ait à connaître la structure des données recherchées.

✓ L'accès à des bases de données hétérogènes, à partir d'une unique requête utilisateur, en langage naturel.

✓ La visualisation de la réponse à la requête de façon compréhensible.

3. Architecture proposé

La figure suivante représente notre architecture:

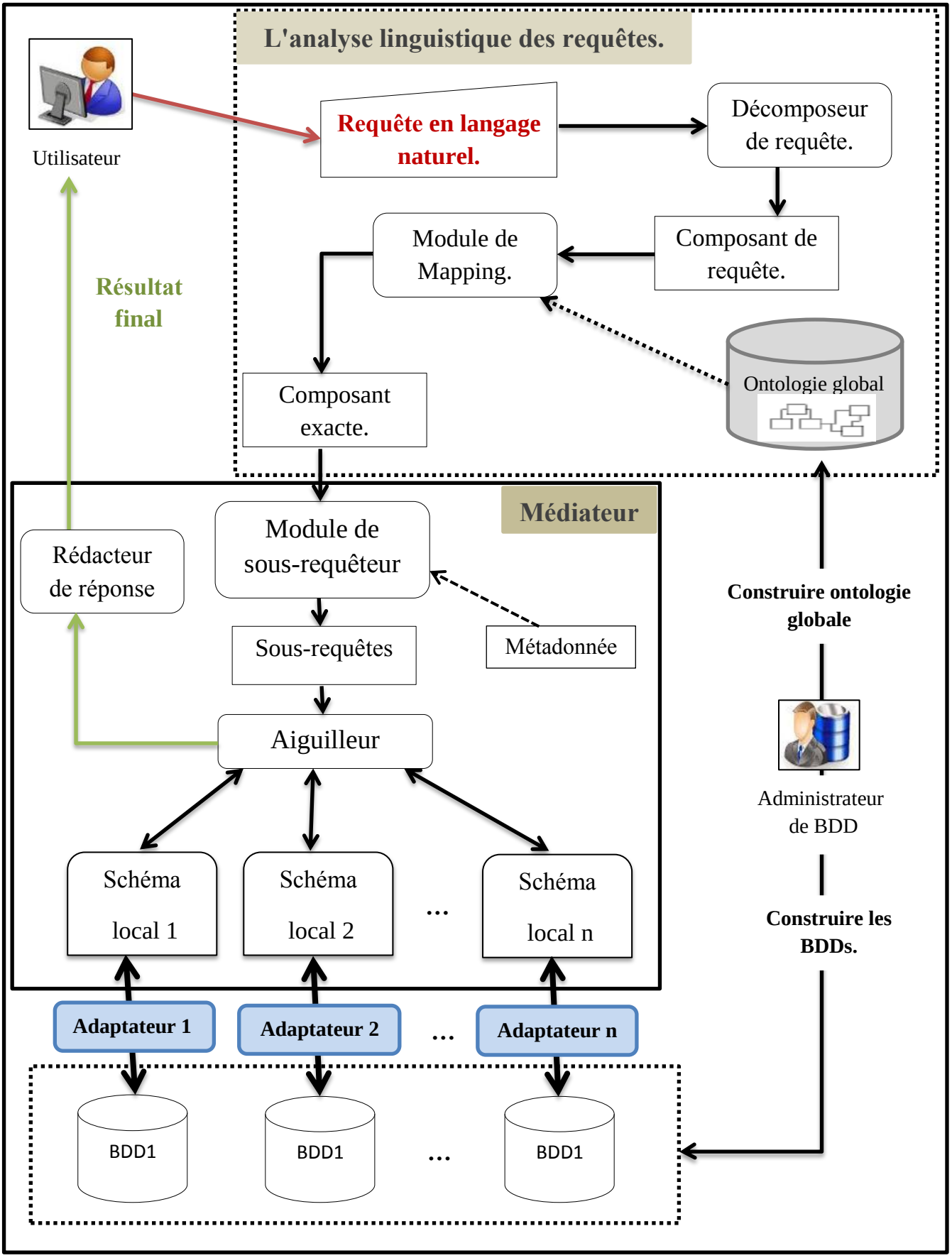

Figure 3.1: Architecture du système.

4. Description de l'architecture proposée

Notre architecture proposée se compose de 2 phases:

La première phase permet de formuler les requêtes d'utilisateur en langage naturel, nous devons par conséquent mettre les fonctionnalités nécessaires pour les traiter linguistiquement c.-à-d. segmenté le composant de la requête en concept puis nous éliminons les mots vides et nous utilisons l'ontologie du domaine (ontologie du domaine commercial) et algorithme de rapprochement Jaro-Winkler pour identifier le concept exact. Après, nous utilisons le résultat de ce traitement pour former une requête globale.

La deuxième phase s'inscrit dans le cadre des approches d'intégration des données via un médiateur à cause de ses avantages, cette approche s'appuient sur la définition de vues, pour simuler l'interrogation des sources de données au travers de ces vues. Nous avons opté l'approche LAV pour la mise en correspondance entre le schéma globale et les schémas locaux, dans cette approche les entités des schémas locaux sont définies comme des vues sur le schéma global qui facilite le passage à des échelles très largement.

La requête globale est décomposée en un ensemble de requêtes locales, à l'aide d'un algorithme de décomposition. Ce dernier, pour chaque concept de la requête global identifie les sources de données approprie en utilisant métadonnées de schéma global (ontologie). Chaque requête locale est dispatchée aux bases de données concernées.

Les résultats des sources sont combinés et supprimés les contradictions et les conflits pour former le résultat final d'après le réducteur de réponse.

5. Illustration sur un scénario

Pour illustrer cette approche de médiation utilisant l'ontologie et la métadonnée de cette ontologie, on considère des sources reparties, relevant une partie des éléments

influençant dans le domaine commercial, le climat, commerce. Ces sources locales sont:

climat: descriptionclimat(date, description_climat).
commercial: client(id_client, code_postal_client, ville_client, etat_client).
population: ville(code_ville, nom_ville).

Figure 3.2: Partie des sources de données.

Les sources sont contenues dans des bases de données relationnelles:

climat: la description de climat dans une date.

client: les information sur les clients; identificateur, code postal, ville et état de client.

population: contient code ville et le nom de cette ville.

On a considéré les données suivantes:

code_ville	nom_ville
1001	L'Abergement-Clemenciat
1002	L'Abergement-de-Varey
1004	Amberieu-en-Bugey
1005	Amberieux-en-Dombes
1006	Ambleon
1007	Ambronay
1008	Ambutrix
1009	Andert-et-Condon
1010	Anglefort
1011	Apremont
1012	Aranc

Figure 3.3: Partie de source de la population.

date	description_climat
02.09.2019	Partly cloudy
01.09.2019	Partly cloudy
31.08.2019	Patchy rain possible
30.08.2019	Partly cloudy
29.08.2019	Partly cloudy
28.08.2019	Sunny
27.08.2019	Partly cloudy
26.08.2019	Sunny
25.08.2019	Partly cloudy
24.08.2019	Partly cloudy
23.08.2019	Partly cloudy

Figure 3.4: Partie de source du climat.

id_client	code_postal_client	ville_client	etat_client
06b8999e2fba1a1fbc88172c00ba8bc7	14409	franca	SP
18955e83d337fd6b2def6b18a428ac77	9790	sao bernardo do campo	SP
4e7b3e00288586ebd08712fdd0374a03	1151	sao paulo	SP
b2b6027bc5c5109e529d4dc6358b12c3	8775	mogi das cruzes	SP
4f2d8ab171c80ec8364f7c12e35b23ad	13056	campinas	SP
879864dab9bc3047522c92c82e1212b8	89254	jaragua do sul	SC
fd826e7cf63160e536e0908c76c3f441	4534	sao paulo	SP
5e274e7a0c3809e14aba7ad5aae0d407	35182	timoteo	MG
5adf08e34b2e993982a47070956c5c65	81560	curitiba	PR
4b7139f34592b3a31687243a302fa75b	30575	belo horizonte	MG
9fb35e4ed6f0a14a4977cd9aea4042bb	39400	montes claros	MG
5aa9e4fdd4dfd20959cad2d772509598	20231	rio de janeiro	RJ

Figure 3.5: Partie de source du commerce.

On a la requête en langage naturel suivante:
Quelle est la description de climat et leur date ?

- Segmentation de la requête:

Quelle / est / la / description / de / climat / et / leurs / date / ? .

- Elimination des mots vides:

description / climat / date.

- Identifier les concepts exacts:

description_climat / date.

Le résultat de cette phase (requête global): **description_climat / date.**

- Module sous-requeteur:

description_climat ϵ table descriptionclimat ϵ source climat.
date ϵ table descriptionclimat ϵ source climat.

- Aiguilleur

Donc cette requête local va envoyer à la source climat et exactement à la table decriptionclimat ni à la source population ni commerce, avec les attributs de sélection description_climat et date.

La syntaxe de la requête est comme suite:

select **description_climat,date** from **descriptionclimat**;

- Rédacteur de réponse

Le résultat de cette requête est:

date	description_climat
02.09.2019	Partly cloudy
01.09.2019	Partly cloudy
31.08.2019	Patchy rain possible
30.08.2019	Partly cloudy
29.08.2019	Partly cloudy
28.08.2019	Sunny
27.08.2019	Partly cloudy
26.08.2019	Sunny
25.08.2019	Partly cloudy
24.08.2019	Partly cloudy
23.08.2019	Partly cloudy

Figure 3.6: Le résultat global de la requête.

6. Composant du système

L'architecture du système proposé est constituée des composants suivants:

6.1. Requête en langage naturel

Les requêtes dans notre système sont formulées en langage naturel qui nécessite une analyse linguistique composé des analyses suivantes [39]:

- Niveau morphologique: l'analyse morphologique consiste à segmenté le texte en unité élémentaire et rendre tous les termes de la requête en minuscule puis identifier le rôle de chaque terme.

- Niveau syntaxique: l'analyse syntaxique signifie la manière dont les mots se combinent pour former des syntagmes (syntagmes nominaux, verbaux, adjectivaux, prépositionnels, adverbaux),
- Niveau sémantique: l'analyse sémantique identifie le sens des mots et de la structure logique d'une phrase.

6.2. Décomposeur de requête

Ce composant permet de segmenter la requête et identifier les différents termes de requête puis en éliminant l'espace vide entre les termes, les caractères spéciaux et les mots vides, le résultat généré sera envoyé au module de Mapping.

6.3. Module de Mapping

Ce composant identifie la correspondance entre les différents termes de la requête et les entités de notre ontologie, utilisons seulement l'ontologie de domaine, Ce module de mapping va déterminer à l'aide de notre ontologie quelles données doit-on y rechercher.

Pour déterminer le sens d'un terme dans une requête, nous vérifions d'abord son appartenance à l'ontologie. Si ce terme appartient déjà à l'ontologie nous n'avons pas besoin de calculer la similarité entre le terme et les concepts dans ontologie. Si le terme n'appartient dans notre ontologie, il est nécessaire d'utiliser une technique de rapprochement pour trouver les simulations entre les caractères des termes.

Dans ce travail, nous avons utilisé la mesure de **Jaro-Winkler** qui principalement utilisée dans la détection de doublons. Cette mesure l'avantage d'être plus efficace dans le cas deux mots qui ont un préfixe commun. Cet algorithme correspond de calculer la distance de simulation entre deux termes, Jaro propose une formule de calcul basé sur le poids de caractères dans la longueur des termes parmi les deux chaînes de caractères, et Winkler se fait l'amélioration de cette algorithme tels qu' il prise en compte le nombre N de caractères communs au début des deux chaînes pour

réduire le taux de comparaisons [40]. Le résultat de cet algorithme entre 0 et 1; '**1**' si les termes sont la même et '**0**' n'existe aucune degré de similarité entre les termes.

$$Jaro\text{-}Winkler(S1, S2) = Jaro(S1, S2) + P * N * (1 - Jaro(S1, S2)).$$

Figure 3.7: Formule de Jaro-Winkler [40].

Sachant que [40]:

*Jaro(S1, S2): $\begin{cases} 0 & ,if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|S1|} + \frac{m}{|S2|} + \frac{m-t}{m}\right) & ,for\ r \end{cases}$ m: le nombre de caractères communs.

t : le nombre de permutation

*P: facteur d'échelle (par défaut 0.1)

*N: le nombre de caractères communs au début des deux chaînes.

6.4. Ontologie domaine

Pour l'automatisation de processus d'intégration et traité l'hétérogénéité sémantique nous adoptons la conception manuelle des ontologies avec la méthodologie METHONTOLOGY [37] qui s'inscrit dans les projets complets (à partir des concepts brute nous faisons notre propre ontologie mais nécessite un professionnel informatique avec la validation par un spécialiste du domaine étudie) c.à.d. Allant de la spécification des besoins jusqu'à la phase réalisation et maintenance.

Dans la suite nous illustrons les étapes de construction l'ontologie pour le domaine commercial (***OntoDC***) d'après des connaissances brutes d'un domaine sont:

- ✓ Spécification.
- ✓ Conceptualisation.
- ✓ Formalisation.

- ✓ Opérationnalisation.
- ✓ Evaluation.

Etape 01 (Spécification)

Une ontologie ne peut être construite qu'après la phase de spécification. Il s'agit d'établir un document informel de spécification de besoins. Au niveau de ce document, nous décrivons l'ontologie à construire à travers les cinq aspects suivants:

Domaine de connaissance

Domaine commercial.

Objectif

Permettre aux différents utilisateurs de système d'avoir un vocabulaire conceptuel commun qui leur permet de partager de façons compréhensible les connaissances commerciales et de communiquer facilement.

Utilisateurs

Client, Employé.

Source d'information

Documents techniques.

Portée de l'ontologie

Client, Employé, Commande,… etc.

Figure 3.8: Document de spécification de besoin.

Etape 02 (Conceptualisation)

Dans cette étape on distingue les principales tâches suivantes:

Tâche 01: Construire glossaire de terme

Le tableau suivant représente une partie des concepts regroupés dans le glossaire ainsi que leurs descriptions.

Terme	Description
Gens	Ensemble d'êtres humains vivant en société, formant une communauté culturelle.
Client	Est une personne morale ou physique (entité) qui prend la décision d'acheter un ou plusieurs marchandises de façon occasionnelle ou habituelle.
Employe	Personne qui occupe un emploi sous les ordres de quelqu'un, dans les sphères non productives de l'économie (commerce, administration, etc.) et dont le travail est d'ordre plutôt intellectuel que manuel (s'oppose à patron, à chef de service, comme à ouvrier).
Categorie	Ensemble ou classe de biens, de services ou d'organisation dont les caractéristiques et/ou les caractères distinctifs sont communs.
Commande	Une intention, soit verbale soit écrite, d'engager une transaction commerciale pour des produits ou services particuliers. Du point de vue de l'acheteur, elle exprime l'intention d'acheter.
Produit	Ensemble des services ou biens fabriqué par les entreprises.
Nature_Personne	Est un sexe de personne soit homme ou bien femme.
Date_Heure	Représente la date et l'heure exacte.
DescriptionClimat	La distribution statistique des conditions de l'atmosphère terrestre dans une région donnée pendant une période donnée.
Humidite	la présence d'eau ou de vapeur d'eau dans l'air ou dans une substance.
Lune	est un objet céleste qui orbite autour de la planète Terre et le seul satellite naturel permanent de la Terre.
Payement	Action de payer, de verser une somme d'argent.
Prix	Il reflète un équilibre entre l'offre et la demande.
Quantité	Tout ce qui peut être mesuré par un nombre.
Soleil	l'étoile du Système solaire. Dans la classification

	astronomique, c'est une étoile de type naine jaune.
Pluie	est un phénomène naturel par lequel des gouttes d'eau tombent des nuages vers le sol.
Pression	une grandeur physique qui traduit les échanges de quantité de mouvement dans un système thermodynamique, et notamment au sein d'un solide ou d'un fluide.
Salaire	une forme de paiement périodique, versé par un employeur à un employé salarié.
Teperature	une grandeur physique mesurée à l'aide d'un thermomètre et étudiée en thermométrie.
Vent	le mouvement au sein d'une atmosphère, masse de gaz située à la surface d'une planète, d'une partie de ce gaz. Les vents sont globalement provoqués par un réchauffement inégalement réparti à la surface de la planète provenant du rayonnement stellaire, et par la rotation de la planète.
Ville	le lieu où une population nombreuse se regroupe sur un espace restraint.
Commune	la commune constitue la plus petite subdivision administrative.
Nom	Mot qui sert à désigner une personne, un organisme ou une chose.
(……)	(…….).

Tableau 3.1: Glossaire de termes.

Tâche 02: Classification des concepts en hiérarchies de concepts

Cette tâche consiste à définir les relations taxonomiques entre les concepts définis dans le glossaire de termes. En souvenir de l'existence de trois stratégies d'identification de concepts, nous avons réellement combiné les trois stratégies pour aboutir aux hiérarchies.

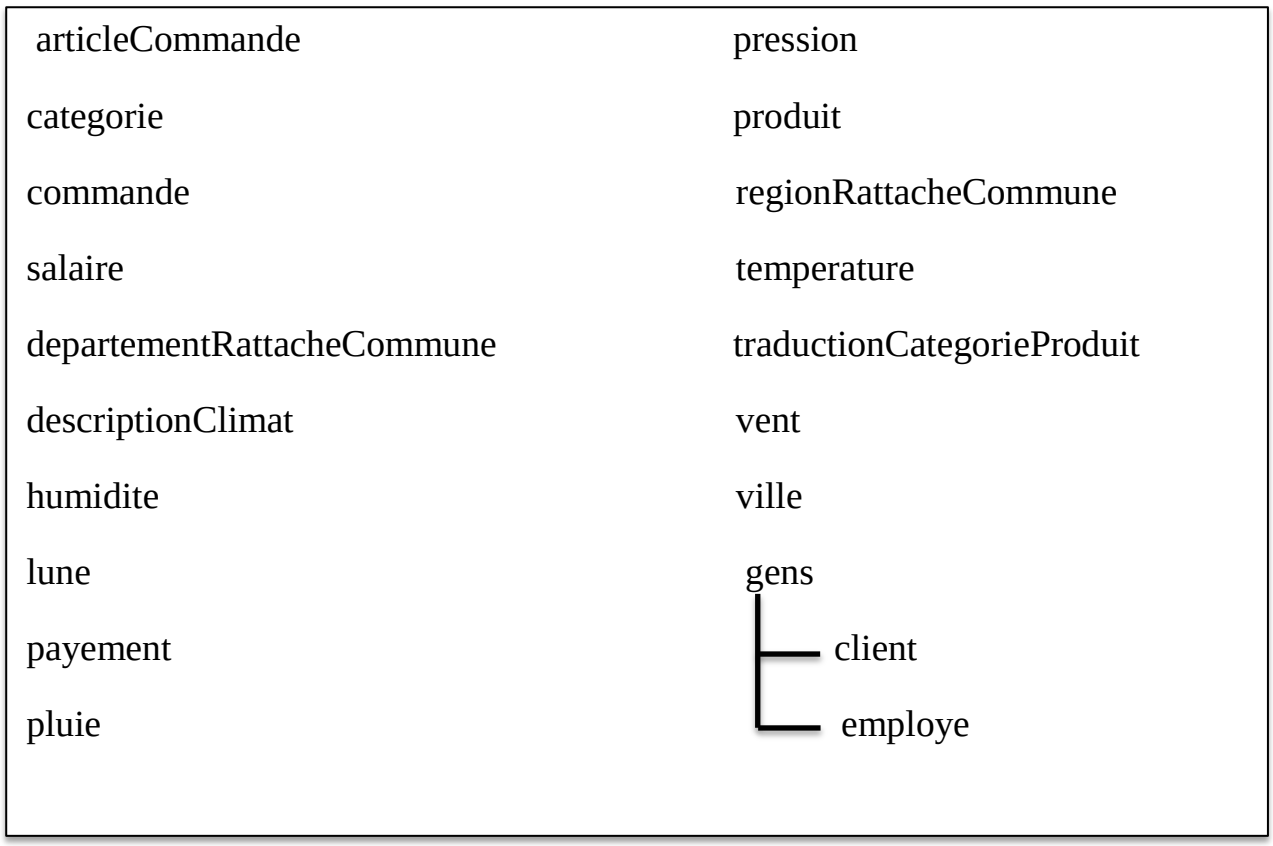

Figure 3.9: hiérarchie de concepts.

Tâche 03: Construction de la table des relations binaires

Le but de cette tâche consiste à construire une table de relations binaires décrites en détail. Pour chaque relation utilisé dans le diagramme des relations binaires, nous définissons le nom de la relation, le nom des concepts sources et cibles, le nom de la relation inverse et les cardinalités source et cible.

Nom de la relation	Concept source	Concept cible	Cardinalité source	Cardinalité cible	Relation inverse
Etablir	Employe	Commande	1…n	1…n	
Effectuer	Client	Commande	1…1	1…n	effectuePar

Possede	Commande	Payement	1…n	1…1	
Rattacher	Departement	Commune	1…n	1…n	
Habite	Gens	Ville	1…1	1…n	
Correspondre	Produit	Categorie	1…1	1…n	
(……)	(……)	(……)	(……)	(…….)	(…….)

Tableau 3.2: Table des relations binaires.

Tâche 04: Construction de la table d'attributs

La table des attributs comporte une description détaillée des attributs inclus dans le dictionnaire de concepts, et l'ensemble de contraintes et de restrictions sur ces valeurs.

Nom attribut	Type de valeur	Cardinalité (max/min)	Valeur par défaut	Domaine de valeur
etat_employe	String	1…1	-	employe
Sexe	String	1…1	-	client
Age	positiveInteger	1…1	-	gens
Code_postal_client	String	1…1	-	Client
nom_ville	String	1…1	-	ville
valeur_payement	Double	1…1	-	payement
Nombre_gens	positiveInteger	1…1	-	gens
situation_region	String	1…1	-	region
coucherLune	dateTime	1…1	-	lune
Taille	positiveInteger	1…1	-	produit
salaireNetMoyen	Double	1…1	-	salaire
(…….)	(……)	(……)	(……)	(……)

Tableau 3.3: Table des attributs.

6.5. Métadonnée

Métadonnée est une description des données (données sur les données), elle permet de décrire les données utilisées dans les analyses et prises de décisions pour identifier les sources de données exactes.

Métadonnées de schéma global fournissent des informations sur la signification et l'utilisabilité des sources, avant d'atteindre les données qu'elles contiennent.

6.6. Module de sous-requêteur

Le sous-requêteur est un des principaux éléments du notre système. Il est aussi l'un des plus complexes. Son rôle est de décomposer la requête initiale afin d'élaborer les sous-requêtes à envoyer à chaque base de données qu'il faut interroger pour récupérer les données utiles à la prise de décision.

Pour cela, il utilise le résultat préliminaire fourni par l'étape d'analyseur linguistique de requête qui se base principalement sur l'utilisation de notre ontologie (***OntoDC***). Ces derniers contiennent toutes les informations nécessaires à la décomposition de la requête en sous-requête.

Le sous-requêteur, à l'aide de ces métadonnées qui fournissent des informations sur la signification et l'utilisabilité des sources, avant d'atteindre les données qu'elles contiennent. On étend ces informations par l'ajout de la clé primaire et les clés étrangères de la relation, pour faciliter la jointure plus tard.

Le résultat de ce composant va envoyer à l'aiguilleur.

6.7. Aiguilleur

Ce composant permet d'envoyer les sous-requêtes aux bases de données concernées et de récupérer les réponses de ces bases.

L'ensemble de réponse des sous-requêtes est envoyé au module suivant, après la réception de la dernière réponse des sous-requêtes.

Les réponses obtenues sont envoyées au rédacteur de réponse. Ce dernier rédige la réponse finale, avec les réponses dont il dispose.

6.8. Adaptateurs

Adaptateur est chargé de traduire la sous-requête qu'il reçoit en un langage compréhensible par les bases de données auquel il est relié. Pour cela, il utilise le module de traduction (traduction de la sous-requête vers langage de base de données de source). De la même façon, il traduit la réponse reçue de la source de données dans le format exigé par le médiateur avant de l'envoyer à l'aiguilleur.

6.9. Rédacteur de réponse

Le dernier élément de notre système est le rédacteur de réponse. Il est chargé d'associer les réponses obtenues afin de former une seule réponse, la réponse attendue par l'utilisateur à sa requête. Pour cela, il dispose des réponses des bases de données interrogées.

Conclusion

Dans ce chapitre nous présentons les différents éléments influençant dans le domaine commercial et la raison de choisir ce domaine. Nous avons présenté une architecture d'un système d'intégration qui base sur une ontologie du domaine commercial et métadonnées de schéma global pour renforcer l'ontologie et interroger des sources hétérogènes, en l'illustrant sur un scénario. Les requêtes dans notre système sont formulées en langage naturel, nous avons mis toutes les fonctionnalités nécessaires pour l'analyser la requête linguistiquement. Nous avons également identifié les différents composants du système. Ainsi, nous détaillons les étapes de construction l'ontologie du domaine commercial qui joue le rôle crucial de système.

Le chapitre suivant est consacré à la présentation des outils utilisés pour son l'implémentions et résultats obtenus.

Chapitre 04: Implémentation

Introduction

L'approche qui nous proposons dans ce mémoire permet de formuler la requête en langage naturel, elle envoyée le requête à décomposeur pour identifie tous les composants existants et éliminée les mots vides, après, le module de Mapping utilise l'ontologie pour déterminer chaque composant par rapport aux entités de l'ontologie, Il supprime également les mots vides. Le résultat de ce dernier sera envoyé aux sous requêteur qui identifie les sources de données approprie en utilisant la métadonnée de schéma global (ontologie). Chaque requête locale est dispatchée aux bases de données concernées par l'aiguilleur et envoyée à adaptateur qui traduit les sous-requêtes en langage compréhensible par les bases de données auquel il est relié. Enfin, le rédacteur de réponse est chargé d'associer les réponses obtenues afin de former une seule réponse compréhensible par l'utilisateur.

Cette approche vise à concevoir une application pour partager de façon compréhensible les connaissances commerciales, faciliter la communication entre l'utilisateur et le système, et garantir la visualisation de la réponse à la requête de façon compréhensible.

Ce chapitre est consacré principalement à l'implémentation de notre application, la présentation des outils et environnements de développements utilisés pour la construction de notre application, et les résultats de cette implémentation.

1. Implémentation de l'ontologie

La phase implémentation est la dernière étape du processus de création d'ontologie, c'est la phase qui consiste à traduire l'ontologie obtenue dans le chapitre précédent en un langage ontologique compréhensible par la machine.

Pour notre implémentation nous avons opté pour le langage OWL (chapitre 02), et l'outil utilisé était Protégé _4,3.

1.1. Protégé

Protégé[1] est un système auteur pour la création d'ontologie. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en java. Il est gratuit et son code source publié sous une licence libre (la Mozilla Public Licence). Protégé peut lire et sauvegarder des ontologies dans la plupart des formats: RDF, OWL, … etc. Il est reconnu pour sa capacité à travailler sur des ontologies de grandes dimensions.

Protégé a été choisi pour implémenter une ontologie interprétable par la machine ce qui doit permet de interroger cette base de connaissance pour les besoins de l'application [41]. La Figure suivante montre la hiérarchie de l'ontologie implémentée via Protégé.

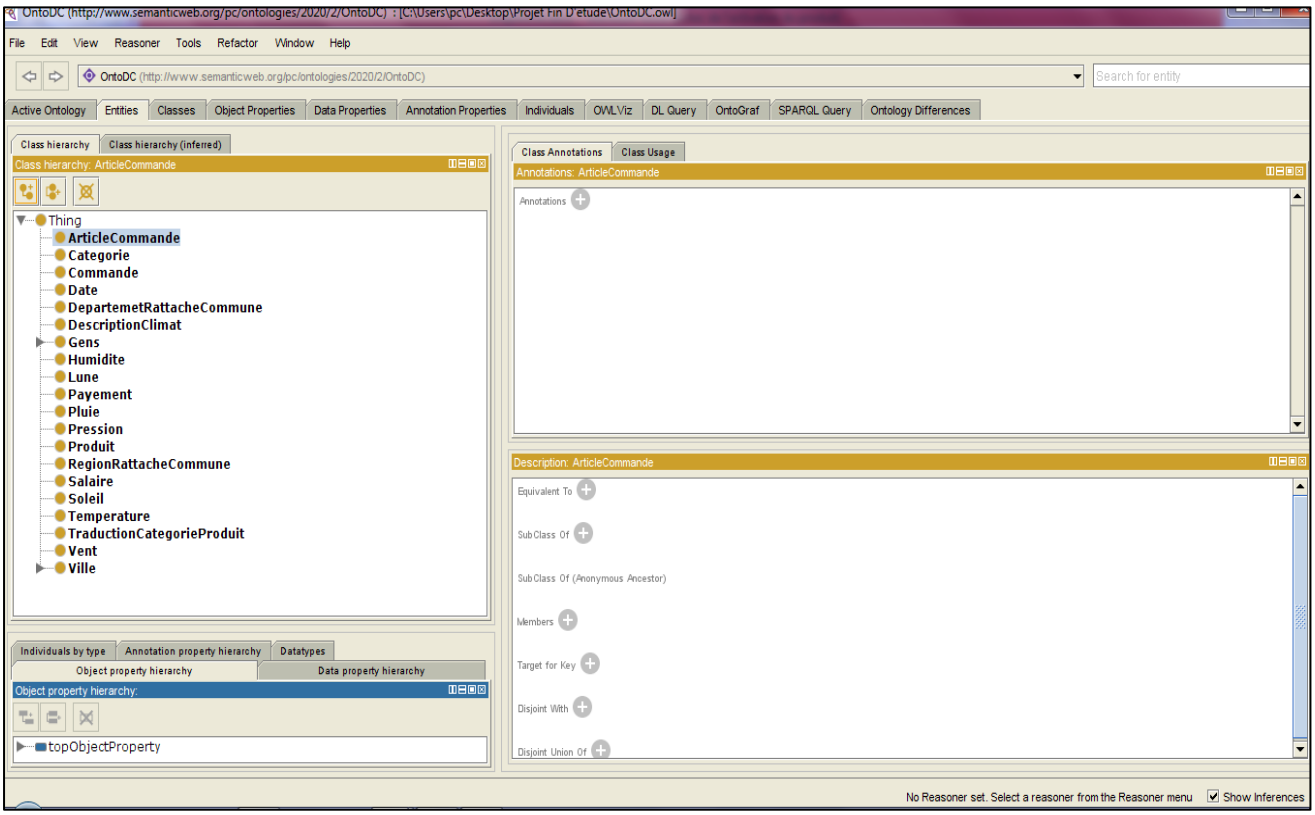

Figure 4.1: Protégé_4,3 environnement de développement de l'ontologie.

[1] https://protege.stanford.edu/.

2. Implémentation de métadonnée

Cette étape consiste à implémenter la métadonnée (format xml) qui fournisse des informations sur la signification et l'utilisabilité des sources, avant d'atteindre les données qu'elles contiennent.

Pour notre implémentation de métadonnée, nous avons opté pour le langage XML[2], et l'outil utilisé était Protégé_3.4.8.

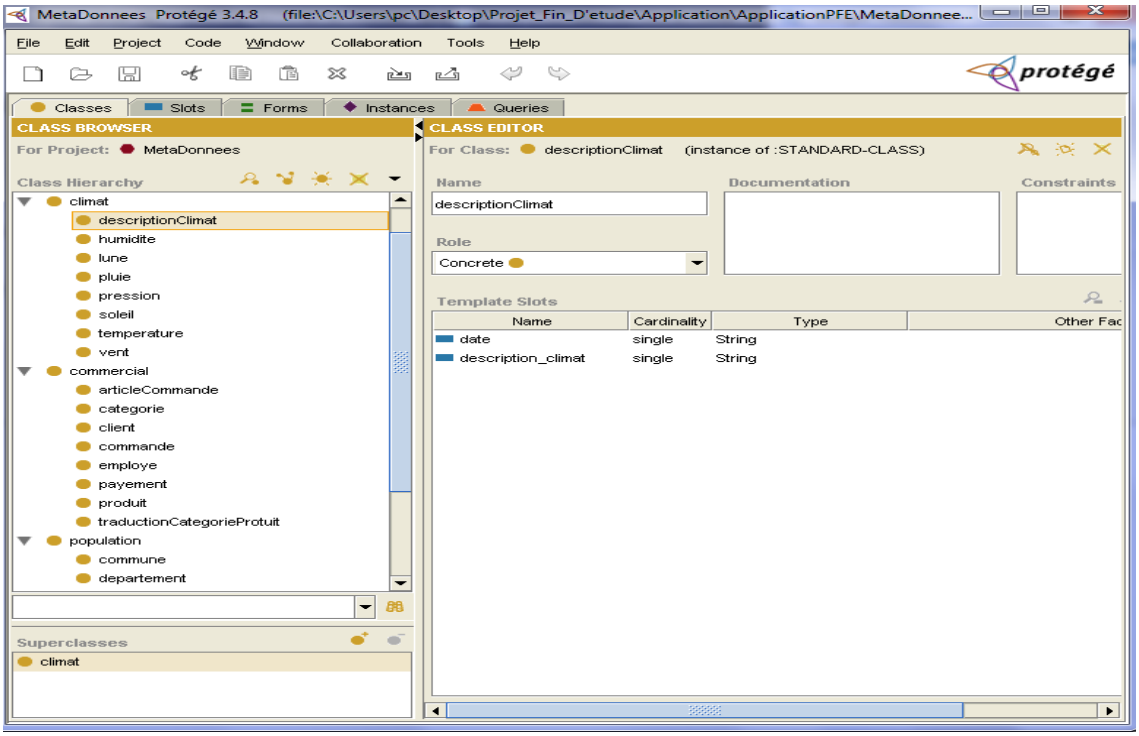

Figure 4.2: Protégé_3.4.8 environnement de développement la métadonnée.

3. Implémentation de l'interface

Cette étape consiste à transformer la maquette de l'interface graphique d'application à une interface-utilisateur exploitable par la machine et manipulable par l'utilisateur, l'outil choisi pour l'implémentation est NetBeans, une Platform de développement orientée Java .Ce choix est motivé par: d'une part les nombreuses

[2] https://fr.wikipedia.org/wiki/Extensible_Markup_Language.

avantages du langage, d'autre part, le fait que la plupart des bibliothèques et des APIs de manipulation d'ontologies sont basés sur le langage Java.

3.1. Netbeans

NetBeans[3] est un environnement de développement intégré (EDI), placé en open source en juin 2000 sous licence CDDL (Common Dévelopment and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme, il s'enrichit à l'aide de plugins [42].

4. Construction de la base de données

Dans cette étape nous allons construire notre base de données permettant de stocker les différentes informations (Population, Commerce et Climat).

4.1. MySql

MySQL (prononcé [maj.ɛs.ky.ɛl]) est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. Son nom vient du prénom de la fille du co-créateur Michael Widenius, MySQL fait référence au Structured Query Language, le langage de requête utilisé. MySQL AB a été acheté le

[3] https://netbeans.org.

16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.Depuis mai 2009, son créateur Michael Widenius a créé MariaDB pour continuer son développement en tant que projet Open Source [43].

5. Connexion ontologie-interface

Cette étape consiste à établir des requêtes orientées Java pour récupérer la connaissance stockée dans l'ontologie pour cela on a utilisé l'API Jena.

5.1. Jena

Jena est un API java open source développé par le laboratoire de Hewlett-Packard permettant la lecture et la manipulation des ontologies décrites en RDFS ou en OWL [44].

La page d'accueil de Jena, illustrée ci-dessous, est http://jena.sourceforge.net/

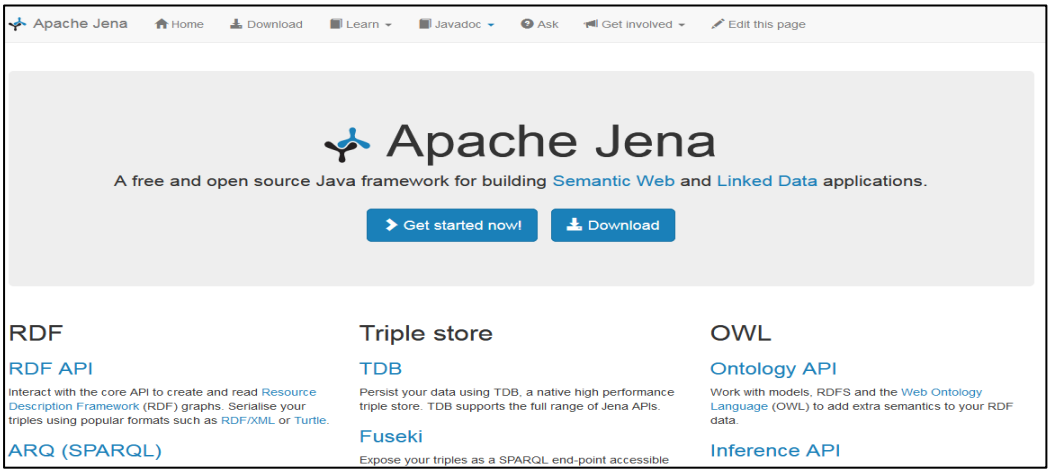

Figure 4.3: Page d'accueil Jena.

6. Plate-forme de gestion base de données

Cette étape consiste à gérer notre base de données facilement.

6.1. WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *tray icon* (icône près de l'horloge de Windows) [45].

7. Architecture logicielle

Dans cette section, nous présentons les différentes parties participant dans notre architecture.

7.1. Ontologie

Selon les étapes représentées dans le deuxième chapitre (Construction d'un entrepôt de données), et en utilisant l'environnement de développement « Protégé », nous avons créé une ontologie version française représentées dans la figure suivante:

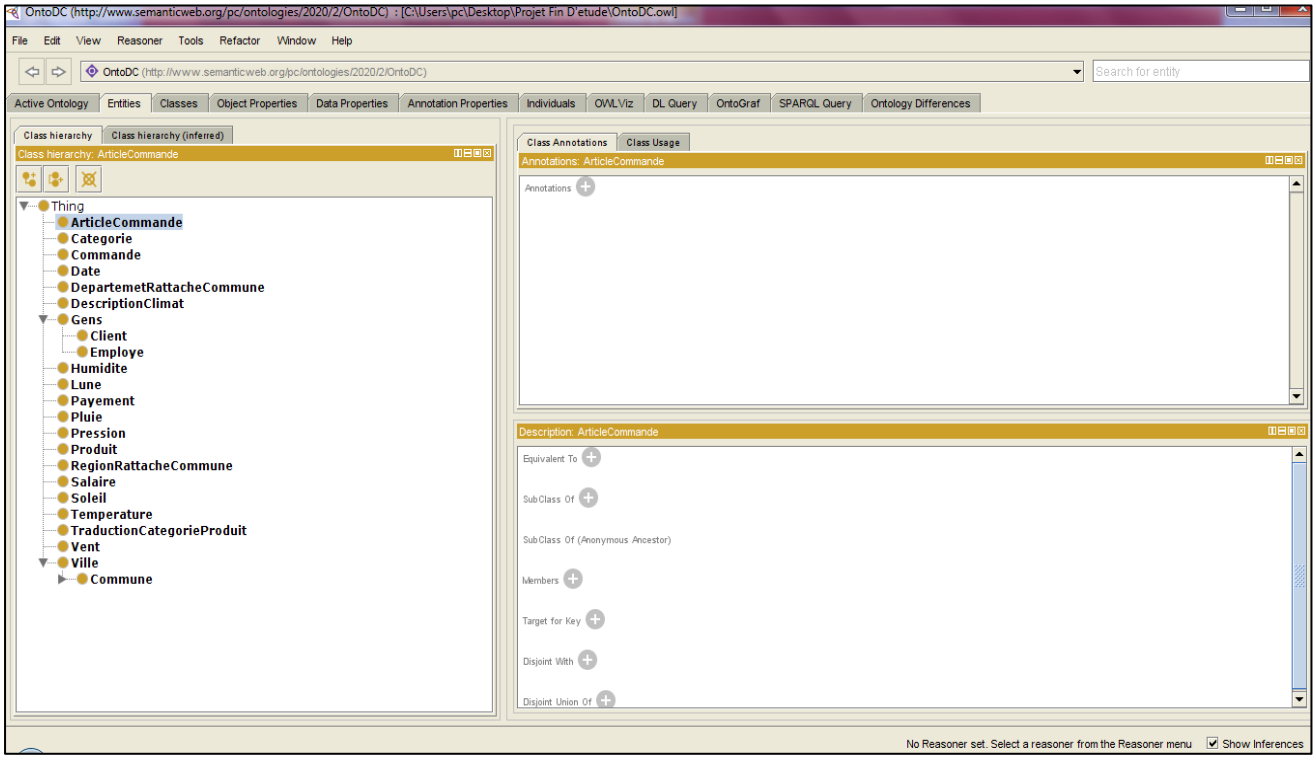

Figure 4.4: Vue globale de notre ontologie (***OntoDC***).

7.2. Base de données

Dans cette partie, nous avons créé les différentes sources de données de domaine commercial pour stocker les informations influençant dans le domaine commercial (base de données Commercial, Climat et Population). Les figures suivantes représentent la forme globale de nos bases.

7.2.1. Base de données climat

La figure suivante représente la base de données de climat.

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	descriptionclimat	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	88.9 KiB	-
☐	humidite	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	77.1 KiB	-
☐	lune	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	120.7 KiB	-
☐	pluie	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	77.1 KiB	-
☐	pression	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	77.1 KiB	-
☐	soleil	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	122.8 KiB	-
☐	temperature	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	92.2 KiB	-
☐	vent	Browse Structure Search Insert Empty Drop	3,896	MyISAM	latin1_swedish_ci	77.1 KiB	-
	8 tables	Sum	31,168	MyISAM	latin1_swedish_ci	732.9 KiB	0 B

Figure 4.5: Vue globale de notre base de données Climat.

7.2.2. Base de données commercial

La figure suivante représente la base de données de commerce.

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	articlecommande	Browse Structure Search Insert Empty Drop	102,425	MyISAM	latin1_swedish_ci	12.5 MiB	-
☐	categorie	Browse Structure Search Insert Empty Drop	73	MyISAM	latin1_swedish_ci	2.8 KiB	-
☐	client	Browse Structure Search Insert Empty Drop	99,441	MyISAM	latin1_swedish_ci	2.7 MiB	-
☐	commande	Browse Structure Search Insert Empty Drop	99,441	MyISAM	latin1_swedish_ci	15.5 MiB	144 B
☐	employe	Browse Structure Search Insert Empty Drop	3,095	MyISAM	latin1_swedish_ci	172.6 KiB	-
☐	payement	Browse Structure Search Insert Empty Drop	103,272	MyISAM	latin1_swedish_ci	6.1 MiB	-
☐	produit	Browse Structure Search Insert Empty Drop	32,951	MyISAM	latin1_swedish_ci	2.2 MiB	-
☐	traductioncategorieprotuit	Browse Structure Search Insert Empty Drop	71	MyISAM	latin1_swedish_ci	3.8 KiB	-
	8 tables	Sum	440,769	MyISAM	latin1_swedish_ci	39.1 MiB	144 B

Figure 4.6: Vue globale de notre base de données Commercial.

7.2.3. Base de données population

La figure suivante représente base de données de population.

Table	Action	Rows	Type	Collation	Size	Overhead
commune	Browse Structure Search Insert Empty Drop	36,722	MyISAM	latin1_swedish_ci	1.2 MiB	20 B
departement	Browse Structure Search Insert Empty Drop	102	MyISAM	latin1_swedish_ci	3.2 KiB	20 B
departementrattachecommune	Browse Structure Search Insert Empty Drop	36,694	MyISAM	latin1_swedish_ci	839 KiB	20 B
employement	Browse Structure Search Insert Empty Drop	36,681	MyISAM	latin1_swedish_ci	1.6 MiB	45 B
gens	Browse Structure Search Insert Empty Drop	127,707	MyISAM	latin1_swedish_ci	3 MiB	20 B
region	Browse Structure Search Insert Empty Drop	28	MyISAM	latin1_swedish_ci	2 KiB	32 B
regionrattachecommune	Browse Structure Search Insert Empty Drop	36,177	MyISAM	latin1_swedish_ci	830.3 KiB	20 B
salaire	Browse Structure Search Insert Empty Drop	5,136	MyISAM	latin1_swedish_ci	507.7 KiB	101 B
ville	Browse Structure Search Insert Empty Drop	36,681	MyISAM	latin1_swedish_ci	837.7 KiB	20 B
9 tables	Sum	315,928	MyISAM	latin1_swedish_ci	8.7 MiB	298 B

Figure 4.7: Vue globale de notre base de données Population.

7.3. Métadonnée

La Figure suivante montre la métadonnée de l'ontologie implémentée via Protégé.

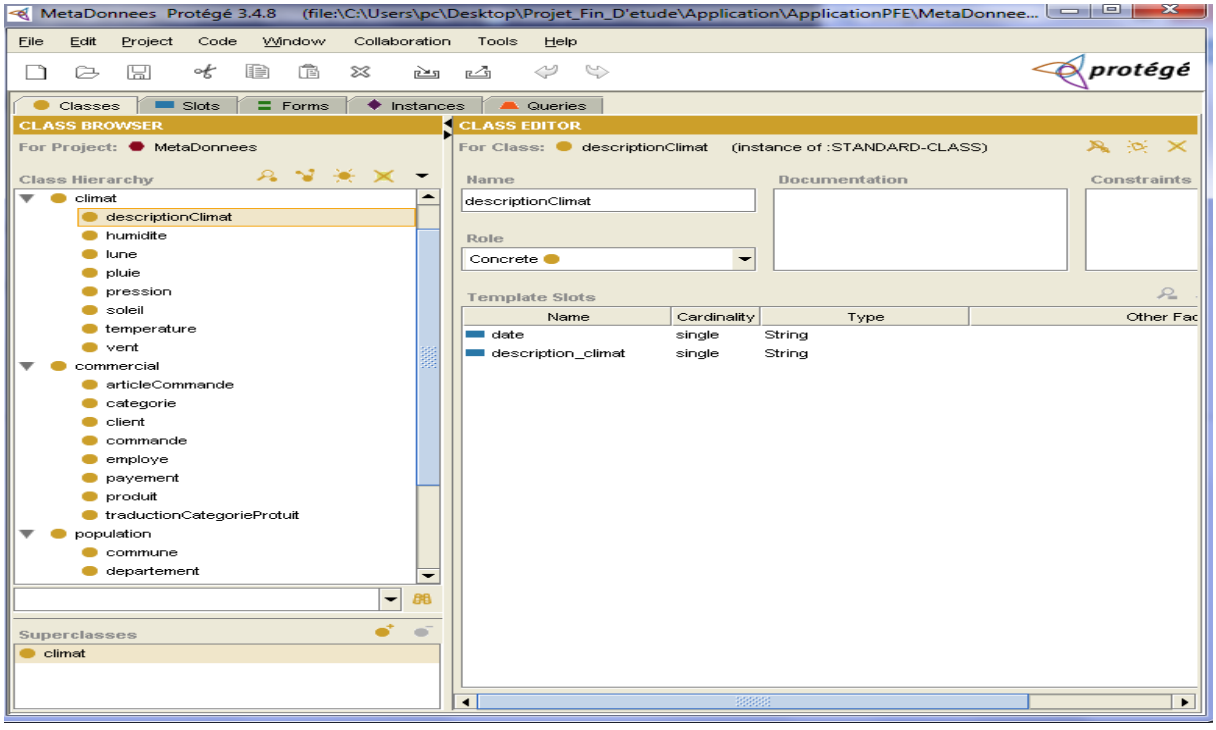

Figure 4.8: Protégé_3.4.8 environnement de développement la métadonnée.

8. Interfaces

Notre logiciel se compose de cinq parties distinctes qui sont illustré dans la suivante.

8.1. Fenêtre d'accueil

C'est la première fenêtre de notre application, la figure suivante représente une capture d'écran de la fenêtre d'accueil de notre application.

Figure 4.9: Fenêtre d'accueil de l'application.

8.2. Fenêtre principale

C'est la fenêtre principale d'application, l'implémentation de cette fenêtre était la partie la plus importante car elle contient des chemins d'accès aux autres fenêtres de l'application (Administrateur, Utilisateur), la figure suivante représente une capture d'écran de la fenêtre principale de notre application.

Figure 4.10: Fenêtre « Principale».

8.3. Fenêtre « Authentification administrateur »

C'est la partie d'application qui permet d'authentefier l'administrateur. Pour entrer aux sessions administrateur il faut passer par l'étape d'authentification admin.

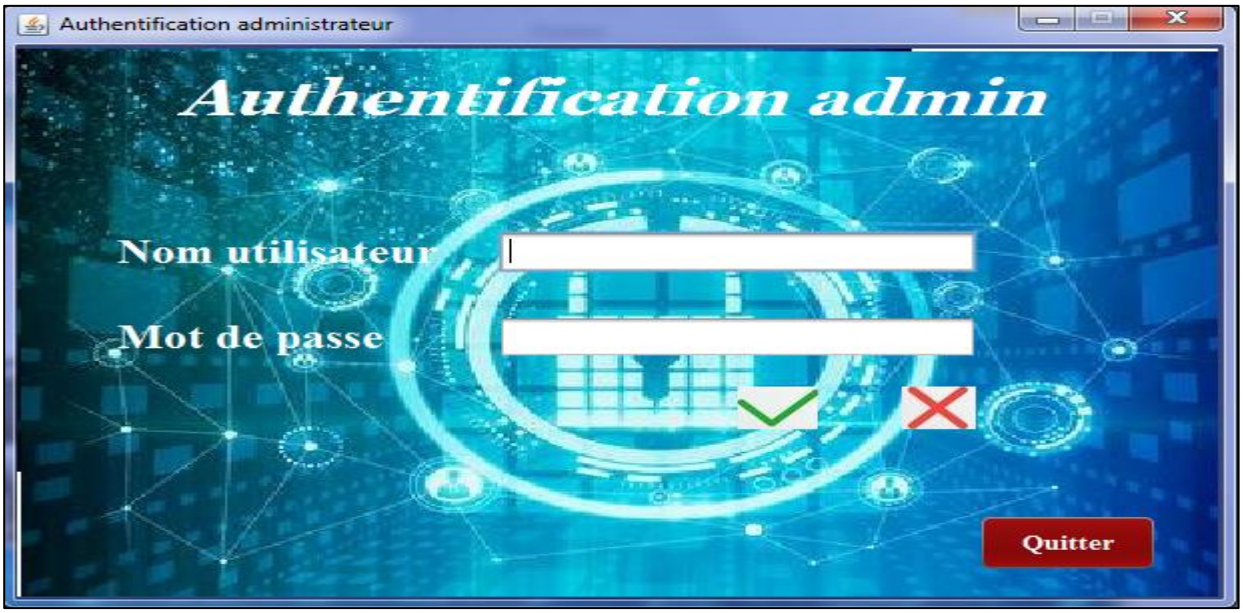

Figure 4.11: Fenêtre « Authentification administrateur».

8.4. Fenêtre « Configuration »

C'est la partie d'application qui regroupe les activités nécessaires à être effectuer par l'administrateur tels que le choix d'autre ontologie et une métadonnée de cette dernière.

Figure 4.12: Fenêtre « Configuration ».

8.5. Fenêtre « Interrogation »

C'est la partie d'application qui permet d'introduire la requête en langage naturel et visualiser la réponse de façon compréhensible.

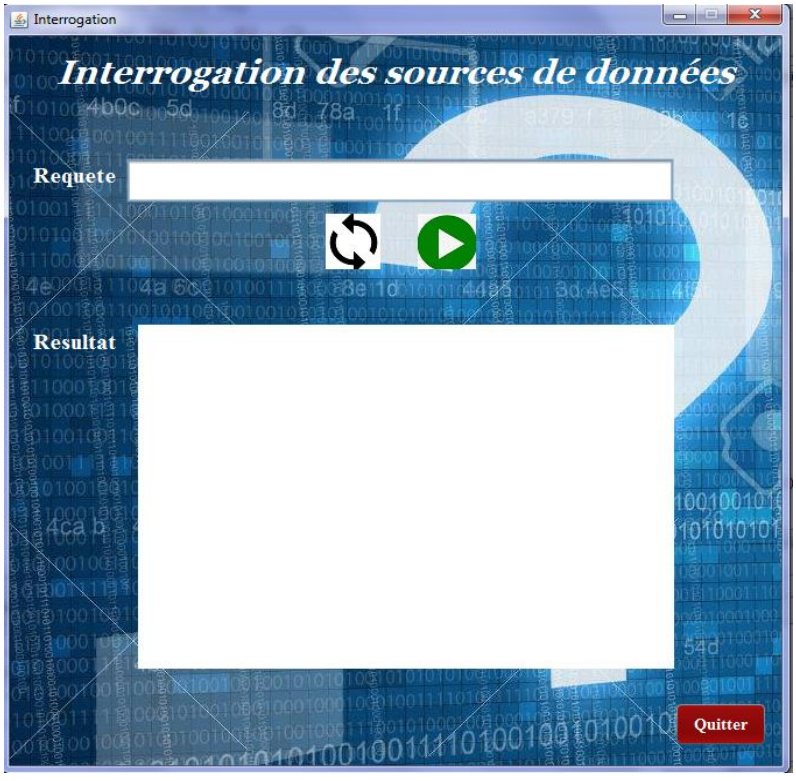

Figure 4.13: Fenêtre « Interrogation ».

9. Fonctionnement

Une fois l'ontologie validée, elle est donc prête pour être interrogée par les requêtes d'utilisateurs. Dans cette section on va expliquer le fonctionnement de l'application.

Nous avons distingué 2 types d'acteurs qui interagissent avec le système: utilisateur du système et administrateur.

Acteur " administrateur "

On va utiliser par défaut notre ontologie « ***OntoDC*** » et dans le cas modification (ajout et suppression concept/rôle,…) le choix d'ontologie ne change pas. Mais, si l'administrateur voudrait de changer le choix d'ontologie global (parcourir et sélectionner seulement le fichier sous forme .owl) et leur métadonnée (format xml), il doit accède à la session d'administrateur.

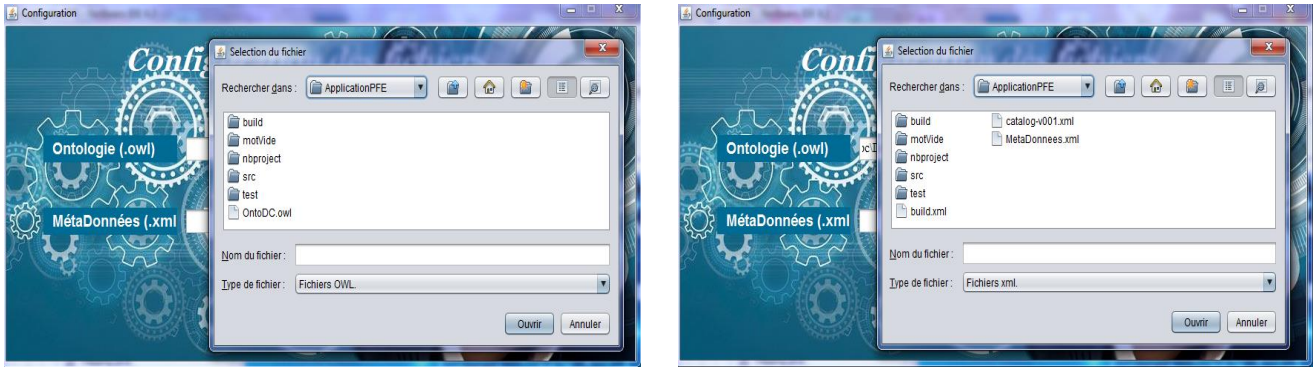

Figure 4.14: Choix ontologie (owl) et métadonnée (xml).

Acteur " utilisateur "

L'utilisateur saisit la requête « Quelle est la description de climat et leur date ?» Le système se fonctionne comme suite:

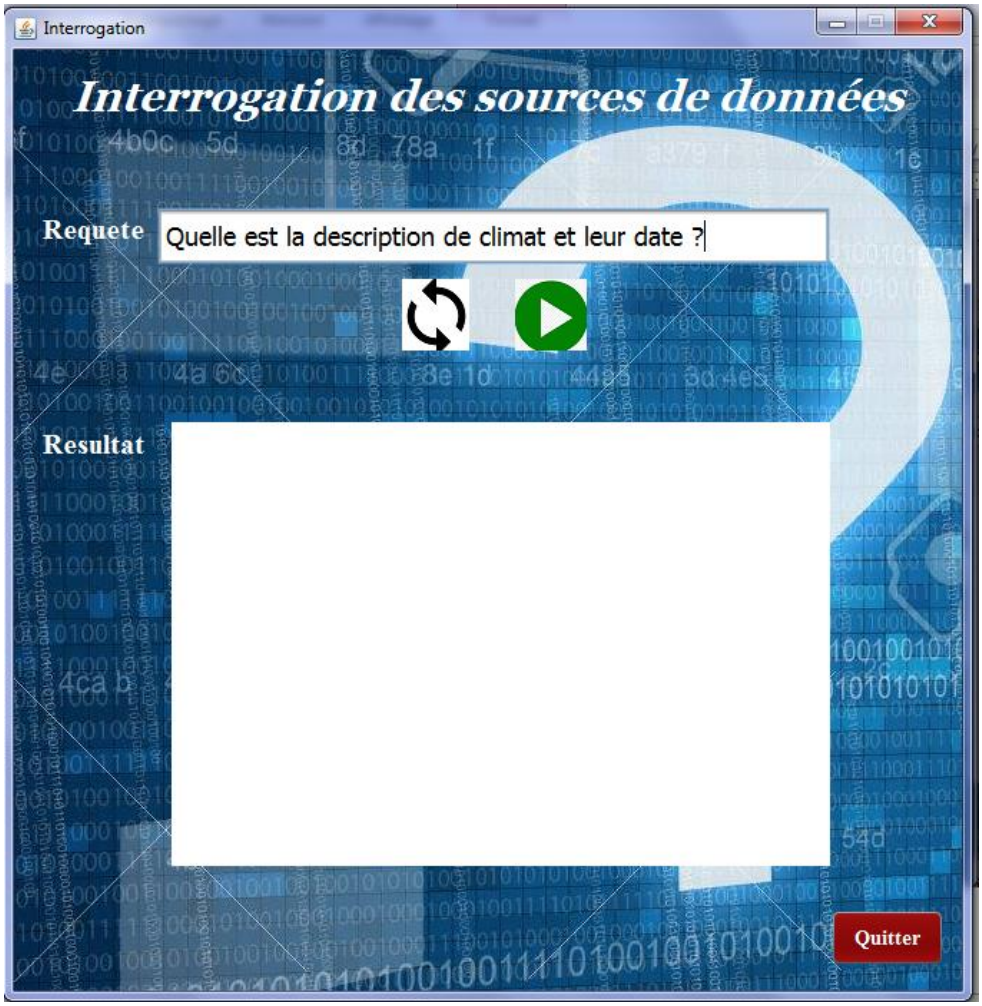

Figure 4.15: Requête initiale.

1. On va d'abord décomposer la requête et identifier les différents rôles de chaque terme utilisant le dictionnaire terme-rôle.

{{quelle, PRO} ; {est, V, N} ; {la, PRO, N} ; {description, N} ; {de, NDET, PREP} ; {climat, N} ; {et, CONJC} ; {leur, PRO, DET} ; {date, V, N} ; {?}}.

Abréviations	Significations
PRO	Propositionnel
V	Verbe
N	Nom
NDET	Nom Déterminant
CONJC	Conjonction de Coordination

DET	Déterminant

Tableau 4.1: Tableau d'abréviation les rôles.

2. Le décomposeur de requête va supprimer les mots vides.

Résultat: description / climat / date.

3. Le module de Mapping permet d'identifier pour chaque terme de la requête ses correspondants dans l'ontologie.

{description} ∉ l'ontologie; on doit donc, utiliser algorithme de rapprochement Jaro-Winkler.

{climat} ∉ l'ontologie; on doit donc, utiliser algorithme de rapprochement Jaro-Winkler.

date ϵ l'ontologie.

Quand le terme n'appartient pas à l'ontologie, nous devons donc calculer le degré de similarité entre le terme et les concepts d'ontologie, et Nous choisissons la valeur maximale.

description -> description_climat.

4. Le module sous requêteur permet d'identifier les sources et les tables de chaque terme.

Description_climat existe dans la table **descriptionClimat** qui appartient dans la source **climat**.

La syntaxe est comme suite: {{Attribut,Source,liste des tables}, …}.

{{description_climat, climat, descriptionClimat} ; {date, climat, descriptionClimat, humidite, lune, pluie, pression, soleil, temperature, vent}}.

5. Aiguilleur qui permet de formaliser la requête et afficher le résultat.

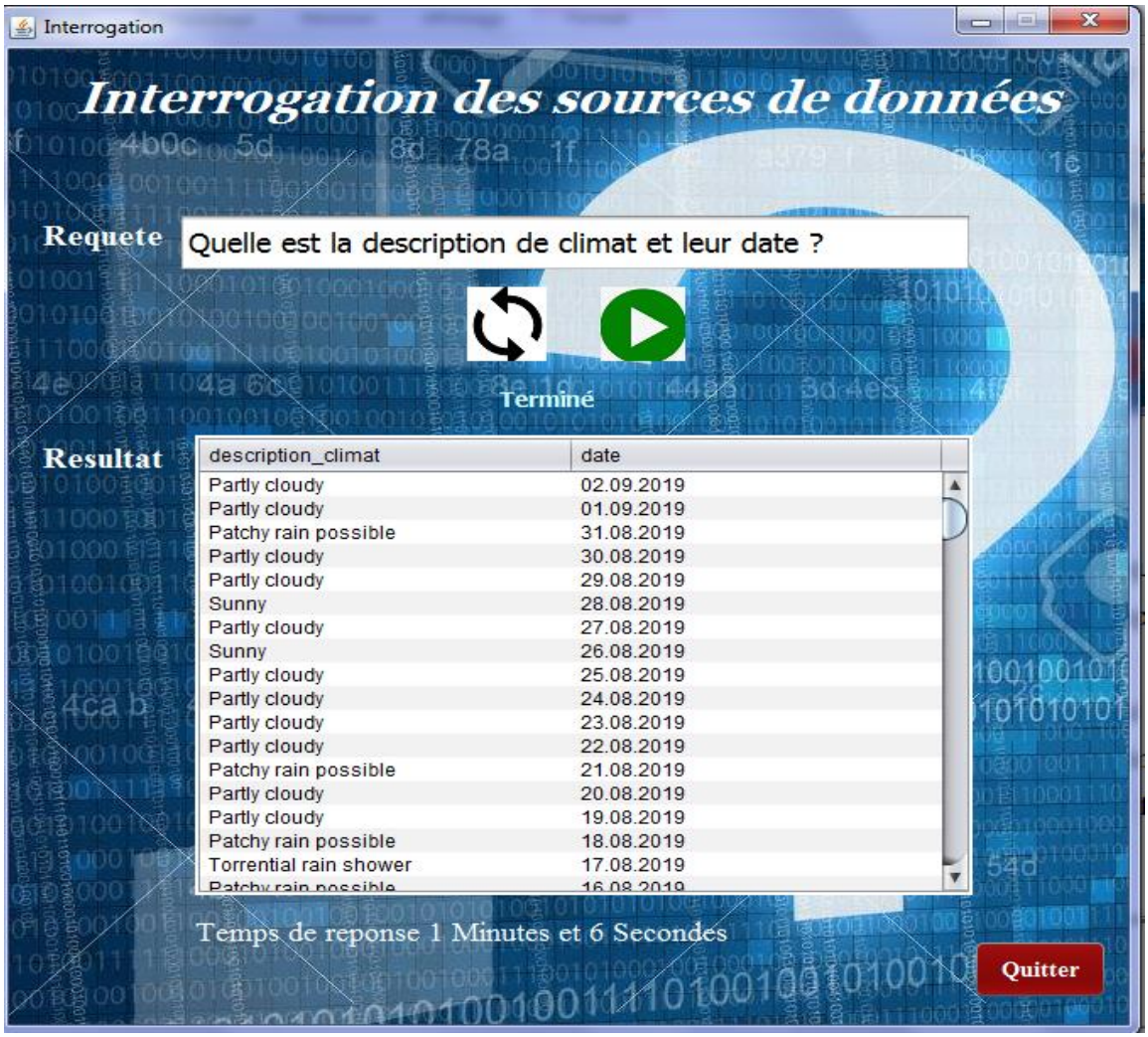

Figure 4.16: Résultat finale de la requête.

10.Analyse et résultats

Dans notre application, nous avons atteint notre objectif principal qui est l'interrogation des sources de données distribuées et l'ontologie qui regroupe les connaissances du notre domaine.

L'interface crée dans cette application permet d'interagir son utilisateur qui n'est pas expert en informatique avec notre application informatique de manière très simple, et la requête est saisit en langage naturel et il obtient la réponse de façon compréhensible.

Conclusion

Dans ce chapitre nous avons présenté l'implémentation de notre application. On définissant tous les outils et langage de programmation utilisé. Par la suite, nous présentons les interfaces de notre application et nous montrons comment les termes initiaux de la requête ont été représentés sous forme des termes spécifiques de l'ontologie avec l'explication de fonctionnement de notre application.

Nous concluons notre travail par une conclusion générales et quelques perspectives.

Conclusion générale et perspective

Conclusion générale et perspective

Le système d'intégration sémantique de données consiste à offrir une description structurelle et sémantique de ces données ainsi elle offre un moyen pour décrire de manière formelle les connaissances d'un domaine commercial afin d'éliminer les conflits entre les données hétérogènes et distribues.

Ces dernières années, l'ontologie utilisée pour donner une conceptualisation structurée des connaissances de domaine commercial. Ainsi, elle joue le rôle de schéma global de médiateur sur lequel l'utilisateur pose ses requêtes.

Au cours de ce travail, on a présenté les différents systèmes d'intégration (entrepôt de données et médiateur), les approches pour le mapping (LAV et GAV) entre schéma global et les sources de données. Ainsi, on a présenté une ontologie dans le domaine commercial qui regroupe de nombre concept ainsi que les liens entre ces concepts (classes et sous classes), des propriétés afin de pouvoir résoudre les hétérogénéités entre les sources de données. Nous avons aussi décrit la description de domaine commercial, le choix du domaine et les éléments influençant dans ce domaine. Puis, une partie a été consacrée à la présentation des méthodologies pour la construction d'ontologie de domaine commercial (***OntoDC***). Enfin, on a abordé la conceptualisation et la réalisation de notre ontologie.

Nous utilisons le médiateur comme un système d'intégration, cette approche permet d'accéder à des données distribuées sans avoir à les migrer vers un schéma global, pour cela choisir cette solution est préférable dans des systèmes où l'information évoluent rapidement, avec l'approche Local As Views (LAV) pour le mapping, cette approche permet d'ajouter une nouvelle source sans modification complète du système. Ce dernier basé principalement sur l'utilisation des ontologies afin de pouvoir résoudre les hétérogénéités entre les sources de données.

Dans ce travail, on a développé un logiciel sous forme une plate-forme qui facilite l'interaction entre utilisateur et système et permet d'accès à des sources à partir d'une requête en langage naturel sans connaitre la structure et l'emplacement des données, ainsi le choix d'une l'ontologie d'après plusieurs ontologie pour le développeur à travers une interface.

Nous avons écrit dans le dernier chapitre le choix du langage d'implémentation qui joue un rôle important dans la construction de l'application. Une liste d'outils de construction d'ontologie et de métadonnées. On a présenté les différentes fonctionnalités de notre application à travers un ensemble des captures d'écran.

Ce travail nous a permis de mettre la lumière sur certaines perspectives de recherche peuvent être envisagées qui sont:

- ✓ Utilisation plusieurs ontologies de domaine.
- ✓ Passe à l'échelle par l'utilisation des ontologies de domaines commercial dans l'étape de conception.
- ✓ Transformé l'application construite à une application Android, pour qu'elle soit plus pratique et convenable.

Références Bibliographiques

[1]: William Inmon, Building the data warehouse, QED Technical publishing groupe, Wel-lesly, Massachusetts, USA, 1992.

[2]: MARIE-CHANTAL DENIS, conception et réalisation d'un entrepôt de données institutionnel dans une perspective de support à la prise de décision, aout 2008.

[3]: Ronan Tournier, thèse: Analyse en ligne (OLAP) de documents, Toulouse, 13 décembre 2007.

[4]: Marilou Pain, Les données de la recherche et leurs entrepôts, de la documentation à la réutilisation: étude de cas pour l'archive HAL, Lyon, Septembre 2016.

[5]: Polraj Ponia, Data Warehousing Fundamentals for IT Professionals, The second edition John Wiley and Sons, 2011,(visité le 4-10-2019).

[6]: Alain Fernandez, Data Warehouse, Entrepôt de données, (visité le 29-09-2019).

[7]: CHIKH Fatima & ZADI Warda, thème: Mise en OEuvre d'un Entrepôt de Données sous Hadoop, Bejaia 2016.

[8]: C.Vangenot, Laboratoire de Bases de Données, (visité le 05-10-2019).

[9]: Louisa Demmou, Exploration de problèmes de performance d'un entrepôt de données, Sherbrooke, Québec, Canada, avril 2010.

[10]: OUHAB Abdallah & RAMTANI Tarik, thème: Etude et Configuration d'un entrepôt de données cas hôpital Khelil Amrane, Bejaia, 2014.

[11]: J.-F. Desnos, Entrepôt de données – Introduction, Spécialité Double Compétence: informatique et Sciences Sociales, Université de Grenoble, 2011.

[12]: Elizabieta Malinowski and Esteban Zimanyi, Advanced Data Warehouse Designe: From Conventional to Spatial and Temporal, Edition Springer, 2008.

[13]: KHOURIS. Thème: Modélisation conceptuelle à base ontologique d'un entrepôt de données, 2008/2009.

[14]: Matteo Glofarelli and Stefano Rizzi, Data Warehouse Design: Modern Principles and Methodologies, juin 2009.

[15]: Omar SENHAJI, Data Warehouse: Entrepôt de données. Sup info international university, 31 octobre 2015.

[16]: Mathieu SANTEL LEBORGNE, Dominique Revuz, Marne La Vallée – Paris -. (visité le 08-10-2019).

[17]: Ralph Kimball, Laura Reeves, Margy Ross et Warren Thornthwaite, Concevoir et déployerun data warehouse : Guide de conduit de projet, édition Eyrolles, octobre 2000.

[18]: Olivier TESTE, Modélisation et manipulation d'entrepôts de données complexes et historiées, 2000.

[19]: KHOURIS, Modélisation conceptuelle à base ontologique d'un entrepôt de données, 2008/2009.

[20]: Telhaoui houria & Boutora asmaa, Utilisation d'une ontologie linguistique (thésaurus) pour l'intégration des sources de données, 2015/2016.

[21]: BENOSMAN Amina, Médiation sémantique de données basée sur les services web: Application dans le domaine médical, 21 novembre 2015.

[22]: Diego Calvanese & Domenico Lemb & Maurizio Lenzerini, Survey on methods for query rewriting and query answering using views, 30 Avril 2001.

[23]: B. Shishedjiev, Entrepôts, France, 20 juillet 2016.

[24]: María Trinidad Serna Encinas, Entrepôts de données pour l'aide à la décision médicale: conception et expérimentation, 2005.

[25]: Abbes RHARRAB & Brahim JIHAD & Mohcine ELJABIRY, Data Warehouse, 25 février 2018.

[26]: Jérôme GABILLAUD, SQL Server 2008 administration d'une base de données avec SQL Server Management Studio.

[27]: ylvain, présentation et installation de pentaho, Develloper.com: club des développeurs et IT pro, tutoriel, 16 avril 2012 (visité le 21-01-2020).

[28]: Gabriel AMOUROUX, Débuter avec Pentaho, 12 juin 2017 (visité le 21-01-2020).

[29]: http://www.open-source-guide.com/Solutions/Developpement-et-couches-intermediaires/Etl/Talend (visité le 22/01/2020).

[30]: Bernard ESPINASSE, Introduction aux entrepôts de données(2), Marseille, Septembre 2013.

[31]: Stéphane Crozat, Introduction au domaine du décisionnel et aux data warehouses, 27 janvier 2016 (visité le 6-11-2019).

[32]: D'après plusieurs littératures; Meriem ARKAM, S. Chafki & C. Desrosiers, Amar bensaber Djamel & Mimoun Malki.

[33]: Kevin Royer, Vers un entrepôt de données et des processus: le cas de la mobilité électrique chez EDF, 12 mai 2015.

[34]: Khadim Drame, Contribution à la construction d'ontologies et à la recherche d'information: application au domaine médical,bordeaux, 22 Jun 2015.

[35]: Frédéric BERTRAND, Utilisation d'ontologies pour l'intégration de données, Rochelle,

[36]: Oucief Affef, Construction d'une Ontologie pour Représenter La Sémantique des Langues Naturelles, 2017/2018.

[37]: Riad LEKHCHINE & Zizette BOUFAÏDA, Construction d'une ontologie pour le domaine de la sécurité: Application aux agents mobiles, 2008/2009.

[38]: GESTION DES RELATIONS AVEC LA CLIENTÈLE.

[39]: Melle HASNA BOUMECHAAL, Conversion des requêtes en langage naturel vers nRQL, 2010.

[40]: andrew1234, https://www.geeksforgeeks.org/jaro-and-jaro-winkler-similarity (Visité le 18/08/2020).

[41]: Rafael Diaz Maurin, Promouvoir les Logiciels Utiles Maîtrisés et Economiques dans l'Enseignement Supérieur et la Recherche, 09/12/2018 (visité le 29/09/2020).

[42]: https://netbeans.org/index_fr.html (visité le 29/09/2020).

[43]: https://fr.wikipedia.org/wiki/MySQL (visité le 30/09/2020).

[44]: Meriem KARAOUZENE & Fatima Zohra BERRADANE, La création d'une ontologie Pharmaceutique, Tlemcen, 2012-2013.

[45]: Romain Bourdon, https://www.wampserver.com/. (Visité le 30/09/2020).