République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8Mai 1945 – Guelma-
Faculté des Mathématiques d'Informatique et des Sciences de la matière
Département d'Informatique

Mémoire de projet de fin d'études Master

Filière : Informatique

Option : Sciences et technologie de l'iNformation et de la communication

---

Une nouvelle approche d'intégration des données des processus métiers basée sur la technologie ETL

Présenté par :
BOUCENA Lilia

Encadreur :
Dr. KHEBIZI Ali

jUIN 2022

REMERCIEMENTS

Nous n'oublions jamais que nous avons rencontré de nombreuses difficultés au cours de notre cursus universitaire, mais tout cela a contribué à la réalisation de ce travail qui a couronné notre parcours.

Tout d'abord, je remercie Dieu qui de m'avoir donné le courage et la patience dont j'avais besoin durant cette langue année.

Je tiens également à remercier mes parents et tous les membres de ma famille qui sont restés à mes côtés pendant mes études, et qui n'ont cessé de m'apporter un soutien moral et matériel, d'autant plus que j'ai terminé ce mémoire.

J'ai remercie mon encadreur Monsieur ***KHEBIZI Ali*** *qui m'a soutenu, guidé, conseillé et m'a apporté tout le soutien dont j'ai besoin tout au long de la réalisation de mon travail.*

Sans oublier de remercier également tout le personnel du département informatique de mon université le 8 mai 1945, en particulier mes professeurs, qui nous ont transmis toutes les connaissances nécessaires à notre formation.

DEDICACES

Cinq années s'achèvent si vite et me voici en train de lever ma plume pour écrire cette dédicace à tous ceux qui sont passés devant mon parcoure et grâce à eux j'ai grandi et j'ai mûrie et je suis devenue ce que je suis maintenant.

Je tiens à remercie du fond du cœur ***mes parents*** *qui m'ont montrée à voir le coté positive de la vie, à être patiente, à vivre le jour au jour. Ma langue s'est retirée en disant au fond de moi : il n'y a pas de mots dans mon dictionnaire qui leurs rendent leurs dû.*

*Je remercie les personnes chères à mon cœur : "****Rania****" et "****Nada****," mes deux sœurs que Dieu me les protège. Mes larmes ne tombaient jamais sans trouver "Rania" et "Nada " qui les essuyait et les remplacerait par un sourire. Je remercie aussi mes deux frères «* ***Ahmed*** *» et «* ***Amir*** *» qui sont si chers pour moi.*

Sans oublier ma tante « ***Samia*** *» ma deuxième Maman, qui m'a aidé financièrement et moralement.*

Enfin, je remercie tous ceux qui ont traversé ma vie, surtout ceux qui s'en sont sortis, je les remercie le plus car ce sont eux avec qui j'ai appris à ne pas faire confiance à tout le monde facilement.

Résumé

Compte tenu de la quantité massive de données manipulées par les processus métiers, qui sont devenus inévitables dans les systèmes d'information des entreprises actuelles (*industrie, administration ; ...*), la technologie de gestion de ces processus métiers (BPM : Business Process Management) est devenu incontournable car elle fournit un ensemble de techniques et de mécanismes pour la gestion de ces processus métiers et de leurs données. Cependant, cette technologie fait souvent face à plusieurs problèmes, à savoir :

(i) la nature hétérogènes des données manipulées.

(ii) Le temps d'analyse et de traitement de immenses quantités de données.

D'autre part, les avancées technologiques dans le domaine des TIC et la démocratisation de l'utilisation de l'internet ont complètement bouleversé les modes de fonctionnement des organisations et les modes de consommation des personnes. Une conséquence immédiate de cette utilisation intensive est l'explosion de la masse des données générées, connue sous le vocable Big-Data. Dans une perspective d'informatique décisionnelle (Business intelligence), l'exploitation rationnelle de cette masse de données exige leur intégration dans des formats et des supports adéquats en vue de leur analyse et afin de faciliter la prise de décision. En effet le processus Extract, Transform and Load (**ETL**) traditionnel vise à répondre à cette préoccupation en offrant des modèles et des outils permettant d'extraire les données de différentes sources et de les intégrer dans des formats homogènes et standards en vue de les exploitation efficacement. Néanmoins, vue la diversité des données, de leur vitesse d'évolution ainsi que de leur volume qui est de plus en plus consistant, les approches classiques ETL ont montré leurs limites et elles sont devenues inadéquates, car ne pouvant plus répondre aux nouvelles exigences.

Dans ce travail nous avons proposé une amélioration de l'architecture ETL afin de prendre en charge les trois propriétés volume, vitesse et variété des données massives. Nous exposons une solution qui devra permettre de récupérer des données hétérogènes de différentes sources, d'analyser leur structure et de formaliser le processus de leur intégration. L'aspect distributivité des données est pris en compte de façon à permettre le stockage et l'exploitation de grands volumes de données stockées dans des bases de données structurées traditionnelles (BDDR), semi-structurées (XML, CSV) et aussi bien que des données en format(EXCEL).

L'approche proposée a été implémentée sous l'environnement PyCharm et elle a été déployée pour expérimenter l'intégration des données d'un domaine de gestion particulier et relatif à gestion des commandes client d'une entreprise commerciale.

Mots clés : *Processus métiers, cycle de vie, modélisation processus métiers, intégration de données, ETL, entrepôt de données.*

Abstract

Given the massive amount of data that business processes bring, which in turn are an inevitable part of today's information systems and enterprises (manufacturing, administration, ...), Business Process Management (BPM) has become a crucial technology that provides a set of techniques and tools for process management. It aims to to deal with several factors/problems, such as :

(i) The nature of the data and their heterogeneity.

(ii) The time needed to analyze and process this immense data.

As well as the technological advances in the field of ICT and the democratization of the use of the Internet have upset the modes of operation of organizations and the modes of consumption of people. An immediate consequence of this intensive use is the explosion of the mass of data generated, known as Big-Data. From a business intelligence perspective, the rational exploitation of this mass of data requires their integration in adequate formats and supports for their analysis and to facilitate decision making.

Indeed, the traditional Extract, Transform and Load (ETL) process aims to respond to this concern by offering models and tools to extract data from different sources and to integrate them into homogeneous formats for their exploitation. Nevertheless, given the diversity of data, their speed of evolution as well as their volume which is more and more consistent, the traditional ETL approaches have shown their limits and have become inadequate, as they can no longer meet the new requirements.

In this work we have proposed an improvement of the ETL architecture in order to take care of the three properties volume, speed and variety of massive data. We expose a solution which will have to allow to recover heterogeneous data from different sources, to analyze their structure and to formalize the process of their integration. The distributive aspect of the data will have to be taken into account in order to allow the storage and exploitation of large volumes of data stored in traditional structured databases (RDB) or semi-structured databases (XML, CSV) as well as data in (EXCEL).

The proposed approach has been implemented under the PyCharm environment and we have modeled the business process of management of a commercial company.

Keywords : *Business process, life cycle, business process modeling, data integration, ETL, Data warehouse.*

Table des matières

Table des figures

Liste des tableaux

abreviations

<**AFD**> <Automate d'états Fini Déterministe>
<**API**> <Application Programming Interface>
<**BDD**> <Base de données>
<**BDD**> <Base de données relationnel>
<**BI**> <Business Intelligence>
<**BP**> <Business Process>
<**BPM**> <Business Process Management>
<**BPEL**> <Business Process Execution Language>
<**BPMN**> <Business Process Modeling and Notation>
<**BPMS**> <Business Process Management System>
<**EAI**> <Entreprise Application Integration>
<**ELT**> <Extract, Load and Transform >
<**EII**> <Entreprise Information Integration>
<**ERP**> <Entreprise Ressource Planning>
<**ESB**> <Entreprise Service Bus>
<**ETL**> <Extract, Transform and Load>
<**GAV**> <Global As View>
<**LAV**> <Local As View>
<**MR**> <Map Reduce>
<**MOLAP**> < Multidimensional On-Line Analytical Processing>
<**OLAP**> <On-Line Analytical Processing>
<**EDD**> <OnLine Extract Selective Transform and Load>
<**ROLAP**> < Relational On-Line Analytical Processing>
<**BP**> <Processus métier>
<**RdP**> <Réseau de Petri>
<**SCD**> <Slowly Changing Dimension>
<**S-ETL**> <Streaming ETL>
<**SGBD**> <Système de Gestion des Bases de Données>
<**SGBDR**> <Système de Gestion des Bases de Données Relationnelles>
<**SI**> <Systèmes d'information>
<**SQL**> <Structured Query Languages>
<**UML**> <Unified Modeling Language>
<**XML**> <eXtensible Markup Language>
<**EDD**> <Entrepôt de Données>

Introduction générale

Les récentes avancées technologiques dans le domaine des TIC, conjuguées avec la démocratisation de l'utilisation d'Internet, ont complètement bouleversé les modes de fonctionnement des entreprises et les modes de consommation des personnes. En effet, de nos jours il est observé une explosion spectaculaire de l'utilisation des machines de traitement automatique de l'information et une large diversité des moyens de communication qui sont dotés de capteurs diversifiés (*téléphones, ordinateurs, smart-télé, smart-home,...*). Une conséquence immédiate de cette exploitation intensive des TIC est l'explosion de la masse des données générées, connue généralement sous le vocable de **données massives** ou **Big-Data**.

Dans une perspective d'informatique décisionnelle (**Business intelligence**), l'exploitation rationnelle de cette masse de données exige leur intégration dans des formats et des supports adéquats en vue de leur analyse qui servira de support d'aide à la prise de décision. D'autre part, il est constaté que le processus **E**xtract, **T**ransform and **L**oad (ETL) traditionnel vise à répondre à cette préoccupation en offrant des modèles et des outils permettant d'extraire les données de différentes sources et de les intégrer dans des formats homogènes en vue de leur exploitation. Néanmoins, vue la diversité des données, de leur vitesse d'évolution ainsi que de leur volume qui est devenu de plus en plus consistant, les technologies ETL classiques ont montré leur limites et elles sont devenues inadéquates, car ne pouvant plus répondre aux nouvelles exigences induites par les données massives.

En effet, avec l'augmentation du débit, les évolutions récentes des TIC et leur démocratisation, les données internes et externes à toute organisation sont devenues de plus en plus variées, instantanées et volumineuses. D'autre part, ces données sont stockées dans plusieurs sources disparates qui ont été conçues indépendamment par des concepteurs différents. Ce phénomène entraine une **hétérogénéité** des données, due aux choix variés opérés pour représenter et stocker des faits du monde réel dans des formats informatiques divers, tels que les bases de données relationnelles, des fichiers semi-structurées (*XML*) ou encore des fichiers plats.

D'autre part, comme ces données se trouvent dans le Cloud, donc issues d'environnement temps réel, alors les systèmes d'informations des entreprises modernes ne peuvent pas attendre des heures ou des jours pour que les applications gèrent les lots de données en question. Au contraire, elles doivent répondre aux nouvelles données en temps réel au fur et à mesure que ces données sont produites par les S.I opérationnels. En effet, les organisations contemporaines génèrent et traitent des données **sous forme de flux continus en temps réel** qui sont de nature éphémère ayant des formats non structurés et des volumes très importants et qui proviennent souvent d'utilisateurs nomades.

De ce qui précède, nous pouvons affirmer que les outils ETL conventionnels demeurent limités pour le traitement des **données hétérogènes** et **en temps-réel** et qu'ils souffrent de certaines limitations fonctionnelles dues à la montée en charge du flux de données. Cela est dû fondamentalement au fait que les volumes de données exponentiellement importants brisent les pipelines ETL au niveau des passerelles. Par ailleurs, plus il faut du temps et des ressources pour transformer ces données, plus la file d'attente des données sources est sauvegardée et les données deviennent obsolètes. De plus, les outils ETL sont incapables de gérer, instantanément, les données hétérogènes et importantes qui pourraient générer

de meilleures informations à valeur ajoutées (*informations commerciales, par exemple*).

Ce projet de fin d'études vise à proposer une amélioration de l'architecture ETL afin de prendre en charge les trois propriétés volume, vitesse et variété des données massives. La solution proposée devra permettre de récupérer des données hétérogènes de différentes sources, d'analyser leur structure et de formaliser le processus de leur intégration. L'aspect distributivité des données devra être pris en compte de façon à permettre le stockage et l'exploitation de grands volumes de données stockées dans des bases de données structurées traditionnelles (BDDR) ou semi-structurées (*XML,EXCEL* et *CSV*). L'approche proposée sera implémenté dans un environnement de développement adéquat et expérimentée via des jeux de données.

En plus de ce chapitre qui présente le contexte de l'étude et la problématique traitée dans ce PFE, le mémoire est structuré en deux parties :

- **La première partie** est un état de l'art du domaine. Elle est composée de trois chapitres.

Le chapitre 1, est dédié à l'introduction et à la présentation des concepts de base du domaine des processus métiers. On y exposera les définitions et notions utiles à la compréhension du mémoire, comme les traces d'exécutions et les instances de processus. Puis, nous exposons le cycle de vie des processus métiers et nous abordons la technologie Business Process Management **BPM**. D'autre part, l'accent sera mis sur les différents modèles de représentation des PM les données qu'ils manipulent.

Le chapitre 2 est consacré à d'intégration des données et à l'analyse des différents problèmes et les technologies associées. Un panorama des différentes approches d'intégration est exposé et une attention particulière sera accordée aux techniques et outils d'intégration ETL.

Le chapitre 3 est un état de l'art du domaine. On y traite des limites de la technologie ETL classique. Nous nous focaliserons sur la problématique abordée dans ce projet de fin d'études et qui est relative à l'intégration des données des processus métiers. Afin de mettre en exergue l'intérêt de notre approche, nous discuterons, dans un premier temps, des avancées technologiques ayant rehausser les outils ETL pour affronter les 3 V du big data, puis une étude comparative des travaux de recherche qui ont traité la question de l'impact des données massives sur les approches classiques ETL est dressée.

- **La deuxième partie** de notre mémoire contient notre contribution. Elle est composée de deux chapitres.

Le chapitre 4 expose le principe général de fonctionnement de la solution proposée, puis l'architecture du système est exposée. Après cela, la description du fonctionnement de la solution est abordée en détails, et enfin nous terminerons le chapitre par l'exposé d'un scénario qui illustre la faisabilité de notre approche par l'examen d'un scénario issue du monde réel.

Le chapitre 5 constitue la mise en eouvre de notre proposition. Il contient l'implémentation de l'approche proposée.

On termine le mémoire par une conclusion générale et des perspectives pour d'éventuels travaux futurs.

Première partie

État de l'art

Chapitre 1

Les Processus métiers

1.1 Introduction

Les entreprises contemporaines sont assujetties à une pression concurrentielle terrible. En effet, avec la mondialisation et la globalisation de l'économie et des échanges, le phénomène de la concurrence est devenu de plus en plus rude. Ainsi, pour survivre aux enjeux concurrentiels, les organisation doivent êtres rentables et performantes. Cette performance passe inévitablement par la maitrise des processus métiers et des informations y afférentes. Dans cette perspective, les entreprises doivent mutualiser rationaliser et travailler de façon transversale, tout en s'appuyant sur les outils technologiques toujours plus performants.

D'autre part, avec la diversité des technologies et des plateformes régissant les systèmes d'information opérationnels des organisations, conjuguée avec le besoin énorme des échanges inter-entreprises, le problème qui se pose est au relatif à la standardisation des normes et règles de travail. Ainsi, il devient impératif de spécifier, sans ambiguïté, les contraintes associées à chaque procédure de travail. En effet, il est souvent fréquent de trouver dans le monde des entreprises des jargons et des "**façon de faire**" spécifiques à chaque employé. Donc, une vision propre du métier. Ce constat complique la procédure qui peut varier d'un individu à un autre. D'où la nécessité de définir une procédure standard qui doit être respectée par tous les employés de la même façon et sans sans déviations, faute de quoi des incohérences lors de l'exécution de ces procédures peuvent se produire. Ce principe de standardisation et d'uniformisation des règles de travail conduit au concept communément désigné par **Processus métier** ou **PM**.

Dans ce chapitre nous allons introduire et illustrer la notion de processus métier et les concepts qui lui sont associés, telles que les traces d'exécutions et instances de processus. Puis nous exposons le cycle de vie des processus métiers et nous abordons la technologie Business Process Management **BPM**. D'autre part, l'accent sera mis sur les différents modèles de représentation des PM et nous terminons le chapitre par l'exposé des données manipulées durant le cycle de vie des PM.

1.2 Notion du processus métier

Le concept de PM est incontournable dans toute organisation et il est fondamental pour l'analyse et la conception des systèmes d'information. Bien que plusieurs autres concepts gravitent autour de cette notion de PM, tels que les notions de "procédure de travail", "protocole de gestion" ou encore "règles de conduite", ce concept trouve un consensus parmi la communauté des intervenants.

Nous exposons, ci-après deux définitions de ce concept fondamental.
Un processus métier ou **"Business Process"** est défini par le Workflow Management Coalition (**WfMC**) comme suit.

Définition 1.1 *Un processus métier est un ensemble de procédures ou d'activités liées les unes aux autres pour atteindre collectivement un objectif métier en définissant les rôles et les interactions fonctionnelles au sein d'une structure organisationnelle". [2]*

Définition 1.2 *Le processus métier consiste en un ensemble d'activités qui sont exécutées en coordination dans un environnement organisationnel et technique. Ces activités réalisent conjointement un objectif de gestion. Chaque activité est mise en œuvre par une seule organisation, mais elle peut interagir avec des processus métier exécutés par d'autres organisations [3].*

Pour illustrer la notion de **PM**, nous exposons ci-après un exemple simple d'un PM relatif au domaine commercial et qui consiste au traitement d'une "commande client" par un fournisseur.

Exemple 1.1 *Exemple d'un Processus métier*

Le processus "commande client" consiste en une séquence d'activités qui commence par l'identification de l'utilisateur et qui se termine, soit par la livraison de la commande ou bien son annulation. Pour représenter ce processus, une panoplie de modèles formels ou graphiques existe dans la littérature. A titre d'exemple, nous le représentons dans la figure 1.1, ci-dessous avec un diagramme de séquence d'UML.

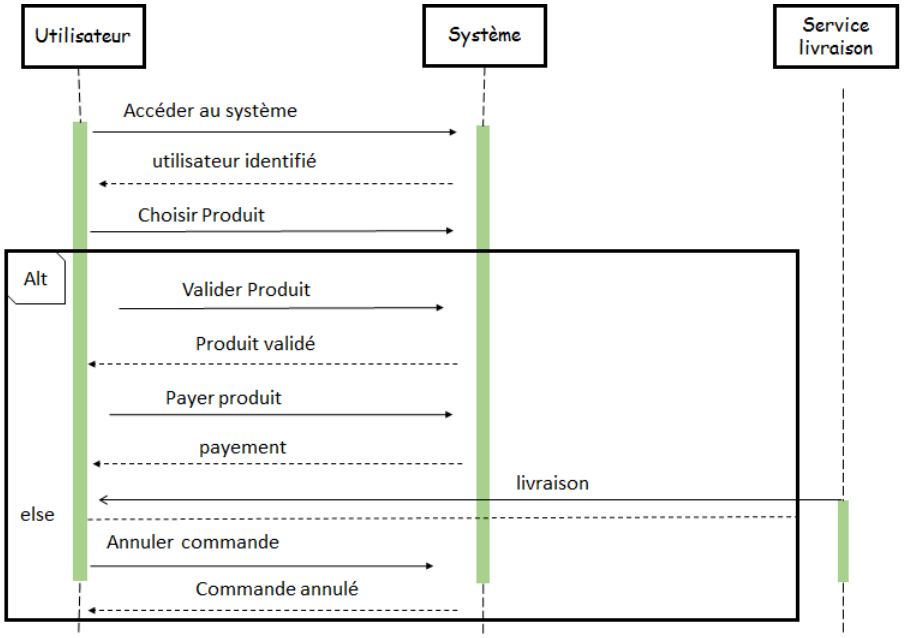

FIGURE 1.1 – Processus métier du traitement d'une commande client

A noter que les différents modèles de représentation seront aborder en détails dans ce chapitre (*voir section 1.6*) Après avoir défini la notion de **PM**, on va aborder dans ce qui suit l'intérêt de leur gestion.

1.3 La gestion du processus métier (BPM)

Vu l'intérêt des PMs et leur rôle en tant que composants centraux sur lesquels s'articule toute la gestion de l'organisation, ils exigent d'être gérés, suivis, supervisés et analysés. Autrement, pris en charge de manière consistante. La gestion des processus métiers ou "**B**usiness **P**rocess **M**anagement" (**BPM**) est la fonction de l'organisation qui assure la prise en charge des différentes phases relatives au cycle de vie des processus métiers . Elle est définie dans [4], comme suit :

Définition 1.3 *La gestion des processus métiers (**BPM**) est l'utilisation de méthodes, techniques et systèmes logiciels pour concevoir, exécuter, contrôler et analyser des processus opérationnels faisant intervenir des hommes, des applications, des documents et d'autres sources d'information.*

Une autre définition du BPM est donnée par [5].

Définition 1.4 *La gestion des PMs consiste en la prise en charge des processus d'entreprise à l'aide de méthodes, de techniques et de logiciels pour concevoir, mettre en œuvre, contrôler et analyser les processus opérationnels impliquant des humains, des organisations, des applications, des documents et d'autres sources d'information.*

Avec cette définition on observe que BPM est une approche globale qui permet de représenter les activités de l'entreprise et de leurs enchaînements dans un contexte donné pour favoriser la conception, l'administration, la configuration, la mise en œuvre et l'analyse des processus métier.

De ce point de vue, la gestion des processus métiers **(BPM)** est considérée comme une approche de management, axée sur l'alignement continue de tous les aspects d'une organisation avec les besoins réels des clients. Elle vise à placer les processus métiers au centre d'une réflexion globale d'intégration, ou sous le concept de "**process-centric**". En effet, le but est de favoriser l'efficacité opérationnelle, tout en abordant la question de l'évolution continue des processus métiers et en avantageant le point de vue "*métier*" sur le point de vue "*technique*". Cette façon de voir l'entreprise rend les processus plus efficaces et plus capables de s'adapter aux éventuels changements de l'environnement.

En définitive, "la gestion des processus métiers est une approche qui favorise la perception de l'organisation en tant que système composé de processus métiers inter-connectés. Cette orientation guide toute l'organisation afin de s'assurer que ses processus métiers sont mis en œuvre efficacement, tout en répondant aux besoins de ses différents interlocuteurs, et avec un niveau de performances optimal aussi bien qu'avec une bonne maîtrise de ses coûts" [5].

Une fois les processus métiers définis et modélisés, ils sont soumis à l'adoption par les différents partenaires qui vont les exécuter. Aussi, ils peuvent être soumis à des actions d'analyse des performances et à des opérations de maintenance et d'actualisation, en vue de les améliorer.

Traditionnellement, les processus métiers sont exécutés manuellement et en conformité avec les règles métiers de l'entreprise. Actuellement, les processus métiers tirent profit des avancées technologiques et sont pris en charge par des logiciels spécifiques, appelés : les systèmes de gestion de processus métiers, ou Business Processes Management Systems (**BPMS**) qui sont exposés ci-dessous.

1.4 Les systèmes de gestion de processus métiers (BPMS)

Un système de gestion de processus métiers, ou Business Process Management System (**BPMS**) est un logiciel générique utilisé pour prendre en charge le cycle de vie des PMs, en assurant la gestion des différents phase, allant de la modélisation jusqu'à la supervision. Ainsi, un logiciel BPMS assure l'exécution des activités de l'organisation et il est souvent guidé par les représentations explicites des PM (*modèles*). Donc, il assure la modélisation, la conception, le développement et l'exécution des tâches et des applications, et aussi certaines tâches de supervision.

Une définition plus formelle des logiciels BPMS est donnée par [6]

Définition 1.5 *Business Process Management Systems*
C'est l'architecture organisationnelle qui intègre toutes les approches, méthodes, techniques et applications technologiques visant à favoriser l'alignement systémique des stratégies et des opérations et à construire une organisation fondée sur les processus et la valeur.

En effet, les logiciels BPMS sont principalement utilisés dans le but de générer du code exécutable qui prend en charge les activités d'un PM, rendre certaines étapes de processus automatique, intégrer les systèmes et les bases de données utilisés par le processus et générer le flux de travail, des documents et autres formulaires manipulés par le processus. Parmi les suites BPMS les plus répandus dans le marché du logiciel, on rencontre Bonitasoft, IBM-websphere, les suite oracle et plusieurs autres outils.

Après avoir présenter les BP et leur gestion, on aborde dans la prochaine section les différentes étapes de leur cycle de vie.

1.5 Cycle de vie des PM

Dans les environnements organisationnels axés sur les processus métiers et en adéquation avec la démarche **BPM**, la gestion du cycle de vie d'un processus métier vise à assurer l'accompagnement du gestionnaire durant toutes les phases, depuis la conception du processus, au pilotage, tout en traitant en permanence les évolutions selon les objectifs métiers et les contraintes qui surgissent dans l'environnement de l'organisation. Afin d'atteindre de tels objectifs, la mise en œuvre de niveaux d'abstraction, aussi bien *théoriques* que *pratiques* pour les processus métiers, à travers plusieurs spécifications et différentes perspectives, est incontournable. Dans la littérature, il n'y a pas de vue uniforme sur le nombre de phases du cycle de vie **BPM**. En effet, le nombre d'étapes varie en fonction de la granularité choisie[2].

Généralement, Il est composé principalement de quatre phases suivantes, comme le montre la figure 1.2.

1. La phase de modélisation "Process Modeling"
2. La phase d'implémentation "Process Implementation"
3. La phase d'exécution "Process Execution"
4. La phase de pilotage et d'optimisation "Process Analysis"

Dans ce qui suit, nous allons expliqué chacune des phases de manière détaillée.

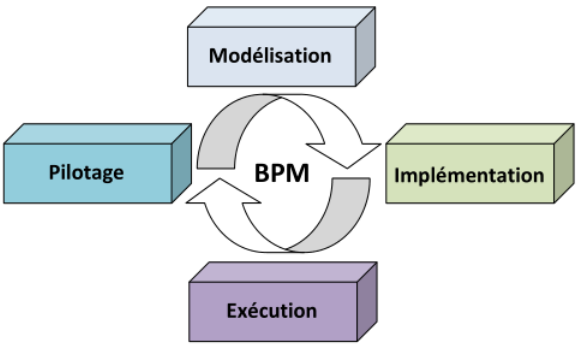

FIGURE 1.2 – Cycle de vie des Processus métiers
d'aprés [2]

1.5.1 Phase de modélisation

La modélisation est la première phase dans le cycle de vie du BPM. Dans cette phase les experts métier définissent, d'une manière abstraite ou détaillée, les processus métiers ou redéfinissent un processus existant dans le but de l'améliorer à l'aide d'un outil de modélisation qui permet de spécifier l'ordre des tâches dans le PM. L'outil de modélisation doit supporter une approche utilisant une notation de modélisation qui soit graphique, basée généralement sur l'adoption du standard "Business Process Modeling Notation" [7].

Les modèles de processus créés dans cette phase sont généralement d'un niveau d'abstraction élevé pour être directement exécutés par un moteur de processus en raison du manque d'informations techniques, telles que les liaisons entre les différents services, les formats de données pour chaque tâche ... etc. Par conséquent, un modèle de processus métier ou "Business Process Diagram" doit être transformé en un modèle de processus exécutable, qui est l'objet de la phase suivante.

1.5.2 Phase d'implémentation

Durant cette phase, les processus créés dans la phase de modélisation sont transformés et enrichis par les ingénieurs informatiques pour qu'ils soient exécutés par le moteur de processus « Process Engine ». A cette fin, le langage standard pour décrire les processus exécutables dans le cadre de l'architecture orientée services (Service Oriented Architecture : SOA) et des services Web est le standard "Business Process Execution Language BPEL" [8].

De manière très simple, la partie exécutable d'un programme BPEL exprime une séquence d'événements d'un PM et dont le déclenchement est conditionné par des structures de contrôle (*conditionnel, boucle, ...*). Ainsi, le modèle de processus exécutable généré peut être déployé et exécuté par le moteur de processus (*Business Process Engine*), qui consiste à mettre en œuvre l'interface avec divers systèmes requis pour le fonctionnement du processus et à mettre en œuvre des règles métier.

1.5.3 Phase d'exécution

La phase d'exécution est la phase opérationnelle de mise en œuvre d'une solution BPM. En effet, à ce stade, le processus exécutable qui précise le déroulement de toutes les activités du processus est interprété par un moteur d'exécution appelé BPE « Business Process Engine ». Le composant BPE du système BPMS est responsable de l'interaction entre les différents participants et ressources du même processus (*documents, informations et tâches*). Il exécute des instances de processus, tout en déléguant les tâches automatiques aux services Web et les tâches manuelles aux acteurs. Si une exception se produit pendant l'exécution du processus, le rôle du BPE est d'initier des actions de compensation pour permettre au processus de s'exécuter efficacement [9].

1.5.4 Phase de pilotage

La phase de pilotage vise à analyser et à optimiser les PMS déployés. Elle consiste à superviser leur exécution opérationnelle et de mesurer les performances, en se basant sur les fichiers logs contenant les différentes traces d'exécution et stockés dans les bases de données du systèmes BPMS. En effet, dans une organisation, le pilotage efficace d'une activité métier représente un point important pour la performance technique et économique de cette dernière.

Le BPM dans son objectif principal de management des processus métiers doit fournir des outils de pilotage favorisant une prise de décision concernant l'efficacité et l'amélioration des processus. Ces outils doivent permettre de mesurer et de présenter la performance de l'activité métier gérée par l'organisation. De manière générale, les solutions de BPM nomment cette fonctionnalité BAM pour "Business Activity Monitoring" ou Supervision de l'activité métier.

Le BAM est une technologie de reporting adaptée à l'ensemble des acteurs métier de l'entreprise ; à savoir : les responsables, les analystes, les dirigeants ou les services informatiques. Cette technologie constitue un élément-clé des solutions de BPM qui permet de surveiller les processus et s'assurer que les performances ne se dégradent pas au fil du temps, d'améliorer l'efficacité des processus, de donner la capacité d'acquérir la maîtrise et la vision d'ensemble du déroulement de l'activité métier ou encore de contrôler le bon déroulement de l'activité à travers des tableaux de bord et en utilisant des KPI "Key Performance Indicators" ou indicateurs clés de performance. Les KPI sont des données collectées lors de l'exécution des PMs et cela dans un but de les améliorer et les optimiser. Les analystes métier ont besoin de ces indicateurs relatives aux différentes instances des processus. Les KPIs permettent de comparer et d'analyser le déroulement des activités basées sur les processus par rapport aux résultats attendus. L'analyse porte sur l'identification des différentes zones du processus qui sont peu ou pas performantes et qui sont susceptibles d'être améliorées.

Après l'exposé des différentes phases du cycle de vie des PMs, nous intéressons, à présent aux techniques de leur modélisation.

1.6 Modélisation des processus métier

La première phase de modélisation des PMs consiste à les représenter de manière plus ou moins abstraite. Elle exige des modèles et des formalismes qui garantissent un niveau de prise en compte des spécifications et règles métiers de manière consistante. La littérature de recherche est très riche en modèles permettant de prendre en charge les différentes contraintes associés à la spécification des PMs. Certains sont formels, tels que les réseaux de pétri, les Automates d'états Finis Déterministes (**AFD**) et les graphes. D'autres modèles sont plutôt graphiques, comme les diagrammes de séquence et les diagrammes d'activités d'UML. Par ailleurs, même certaines catégories de langage offrent à leur tour des mécanismes pour modéliser les PMs.

Dans cette section, nous nous concentrons sur la présentation et l'analyse des différents modèles de représentation des PMs.

1.6.1 Les modèles formels

a) Automates Finis Déterministe (AFD)

Les automates sont des objets mathématiques très utilisés en informatique pour modéliser un grand nombre de systèmes informatiques. D'une façon très simpliste, on considère un automate comme un ensemble d'états reliés entre eux par des transitions qui sont marquées par des symboles. Étant donné un mot fourni en entrée, l'automate lit les symboles du mot un par un et passe d'un état vers un autre conformément aux transitions spécifiés. Le mot lu est soit accepté par l'automate soit rejeté [10].

Les automates ont été intensivement utilisés dans l'algèbre des processus et dans le domaine des processus métiers. Formellement, un AFD est défini comme suit.

Définition 1.6 *Automates finis déterministes [11]*
Un automate fini déterministe est un tuple $\mathcal{P} = (Q, q_0, \mathcal{F}, \mathcal{M}, \mathcal{R})$*, tels que :*

- Q *est un ensemble fini d'états ;*
- $q_0 \in Q$ *est l'état initial du processus ;*
- $\mathcal{F} \subseteq Q$ *est l'ensemble des états finaux du processus (ou acceptant) ;*
- $\mathcal{M}$ *est un ensemble fini d'activités abstraites ;*
- $\mathcal{R} \subseteq Q \times Q \times \mathcal{M}$ *est une relation de transition. Chaque élément* $(q, q', m) \in \mathcal{R}$ *représente une transition d'un état source* q *vers un autre état cible* q'*, suite à l'exécution de l'activité* m*.*

Exemple 1.2 *La figure 1.3, ci-dessous illustre un processus métier modélisé par un AFD. Ce processus métier est relatif à la gestion des commandes client de la figure 1.1.*

Comme il apparait dans la figure, ce processus est composé d'un ensemble d'états (par exemple : utilisateur identifié, Produit choisi, *...) qui commence par un état initial (*début*), suite à la commande d'un produit et qui se termine par des états finaux (*Commande annulée*) dans le cas ou la commande est annulée et (*Fin*) si la commande est livrée.*

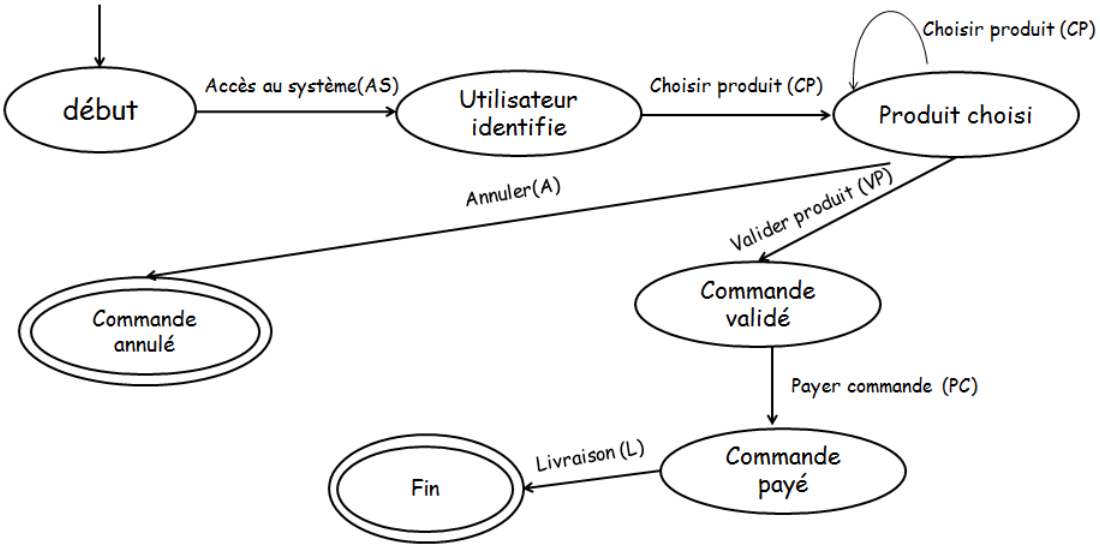

FIGURE 1.3 – Exemple d'un PM modélisé par un AFD

b) Les Réseaux de Pétri (RDP)

Vu leur rôle important en tant qu'outils graphiques de représentation des phénomènes et mécanismes séquentiels, les réseaux de Pétri ont été largement exploités pour décrire les processus métiers [12].

Dans ce qui suit, on s'intéresse à ce mécanisme de modélisation.
Les éléments graphiques de bases utiles à la représentation des PMs par des réseaux de Pétri sont assemblés dans la figure 1.4 suivante.

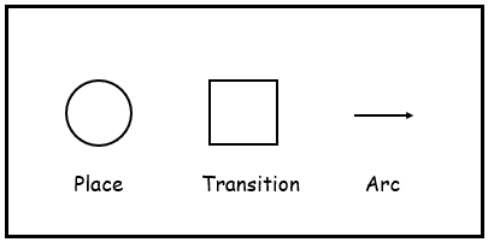

FIGURE 1.4 – Eléments de base pour modéliser un PM par un RDP

Le PM se présente sous la forme d'un graphe biparti dans laquelle les places sont représentées par des cercles ou des ronds, les transitions sont représentées par des nœuds et les arcs permettent de relier les transitions à des places.

D'une façon plus formelle on définit un réseau de pétri comme suit :

Définition 1.7 *Un réseau est un triplet* $N =< P, T; F >$ *où :*

- P *est un ensemble fini de places ;*
- T *est un ensemble fini de transitions ;*
- F *est la relation de flux sur* N*, telle que :* $F \subseteq (P \times T) \cup (T \times P)$ *(ensemble des arcs)*

Exemple 1.3 *La figure 1.5 illustre le même PM de commande client de la figure 1.1) qui est modélisé par un RDP. Ce réseau exprime les mêmes étapes, partant initialement de l'identification jusqu'à la livraison de la commande ou son annulation.*

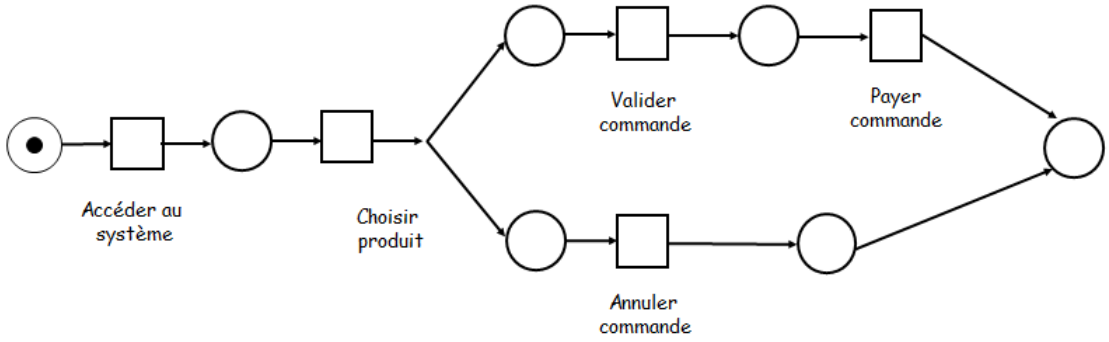

FIGURE 1.5 – Représentation d'un processus de commandes client par un RDP

c) Les graphes

Les graphes sont des outils qui permettent de modéliser et de résoudre de nombreux problèmes et peuvent êtres utilisés aussi pour la représentation des PMs [13].

Ci-après, on va abordé les concepts de graphes et leur utilisation dans le contextes des PMS.

Définition 1.8 *Graphe orienté*
Un graphe orienté $G = (X, U)$ est défini par la donnée de deux ensembles :

- *un ensemble X dont les éléments sont appelés "sommets" ou "nœuds";*
- *un ensemble U dont les éléments sont des couples de sommets appelés "arcs";*

Une autre définition plus détaillée est donnée par [14]

Définition 1.9 *Le graphe* $\boldsymbol{G = (V,E)}$ *est défini par l'ensemble $V = v_1, v_2, ..., v_n$ dont les éléments sont appelés* ***sommets*** *et par l'ensemble des éléments $E = e_1, e_2, \ldots, e_m$ appelés* ***arêtes****. Une arête e_i d'un ensemble E est définie par une paire de sommets, appelés extrémités. Si l'arête e_i relie les sommets a et b, on dit que ces sommets sont adjacents, ou liés à ei, ou bien que l'arête e_i relie les sommets a et b. On appelle ordre d'un graphe le nombre de sommets n de ce graphe.*

Nous exposons ci-dessous quelques concepts associés au modèle de graphes.

- Une arête a est une paire de sommets (x, y) (un élément qui relie deux sommets ensemble).
- Les sommets x et y sont les extrémités de l'arête.
- Le nombre d'arêtes incidentes à un sommet est le nombre d'arêtes quittant ou entrant dans le sommet.

- Deux sommets d'un graphe sont adjacents s'ils ont au moins une arête incidente en commun.
- Deux arêtes d'un graphe sont adjacentes si elles ont au moins un sommet en commun.
- Le degré d'un sommet x de G est le nombre d'arêtes incidentes sur x. Il est noté $d(x)$.
- Un graphe est simple s'il ne contient pas d'anneaux ni d'arêtes multiples.
- Un graphe est connexe si tous les autres sommets sont accessibles depuis n'importe quel sommet.
- Un graphe complet est un graphe dans lequel chaque sommet est connecté à tous les autres sommets.

Exemple 1.4 *La figure 1.6 illustre le même PM de commande client de la figure 1.1) qui est modélisé par un graphe. Ce graphe exprime les mêmes fonctionnalités du PM, mais exprimées par le formalisme des graphes.*

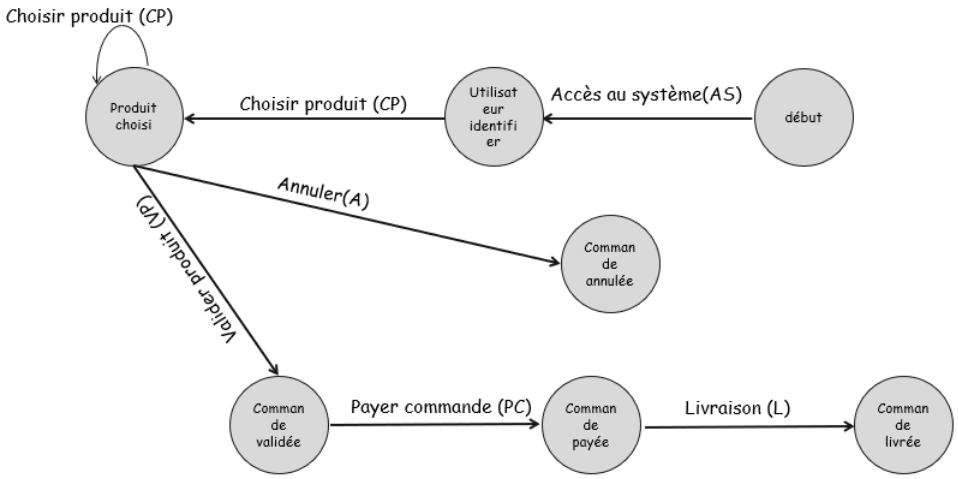

FIGURE 1.6 – Représentation d'un processus de commandes client par un Graphe

1.6.2 Les modèles Graphiques

Parmi les modèles graphiques les plus utilisés pour exprimer l'aspect dynamique des systèmes d'information, on distingue les diagrammes de séquences et les diagrammes d'activités de la méthode UML. Ces diagrammes intégrés sont utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. La suite de cette section est dédiées à la présentation de ces deux diagrammes.

a) Diagramme de séquences

Il est considéré comme l'un des diagrammes UML les plus importants car il aide à comprendre le fonctionnement du logiciel et par conséquent le processus métier pris en charge. D'une part, il montre les interactions entre les objets dans l'ordre chronologique et d'autre part il spécifie les interactions entre objets à l'intérieur du système. Les diagrammes de séquences sont utilisés pour illustrer les cas d'utilisation au début du cycle de développement et constituent également un excellent moyen de communiquer les aspects dynamiques du système [15].

La figure 1.1 de l'exemple 1.1, représente un exemple d'un diagramme de séquences permettant de traiter les commandes client.

b) Diagramme d'activités

Le diagramme d'activités est principalement utilisé pour modéliser des processus métiers ou pour décrire des opérations complexes, y compris des flux de données. son avantage est qu'il présente une vue macro et temporelle du système modélisé sous une forme proche d'un organigramme. Ainsi, il permet de modéliser le processus d'interaction d'un système donné et d'exprimer la dimension temporelle sur une partie du modèle [15].

Dans le contexte des processus métiers, un diagramme d'activités permet de représenter l'ensemble des activités séquentielles d'un PM donné du moment du déclenchement du processus jusqu'à son point d'arrivé.

Exemple 1.5 *La figure 1.7, reprend le même processus de commande client de l'exemple 1.1 et le représente par un diagramme d'activités .*

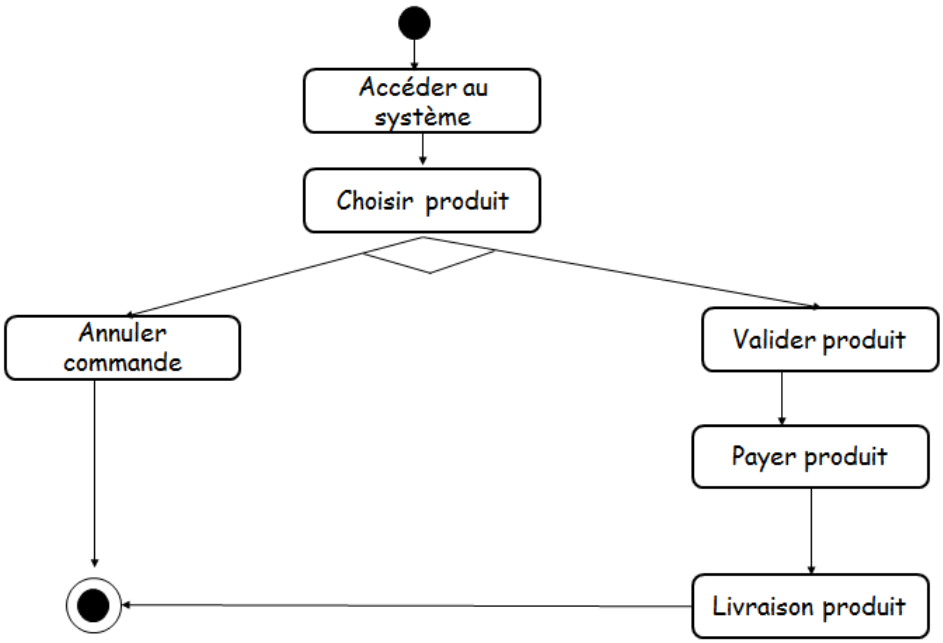

FIGURE 1.7 – Diagramme d'activités pour le processus commande client

1.6.3 Les langages de représentation des PMs

Certains langages offrent des primitives et des commandes permettant de modéliser et de spécifier la façon dont les différentes activités des PMs sont structurées et ordonnées avant leur exécution. Autrement dit, ces langages expriment les scénarios d'exécution des suites ordonnées d'activités. Parmi ces langages on distingue essentiellement BPMN et BPEL.

a) BPMN

La notation de modélisation BPMN (**B**usiness **P**rocess **M**odel **N**otation) [7] est une norme de plus en plus adoptée pour la modélisation des PMs et qui prend de l'essor au sein de la communauté des professionnels de la technologie BPM. Cet interressement est motivé par la richesse du langage qui offre une multitude d'élements de représentation pour exprimer les scénarios d'entreprise [16].

Définition 1.10 *[7]*
BPMN est une norme de notation pour la modélisation de processus métier permet de définir une notation graphique commune à tous les outils de modélisation.

En termes simples, BPMN est une représentation graphique du PM à l'aide d'objets standards. Ces derniers ne sont que des ensembles d'objets graphiques et de règles définissant les connexions disponibles entre les éléments manipulés. Son objectif est de fournir un cadre permettant de décrire un processus d'une manière commune à tous les utilisateurs et ce, indépendamment de l'outil utilisé. L'outil étant bien sûr censé supporter la norme.

Exemple 1.6 *Dans l'exemple de la figure 1.8, notre PM de commande client (exemple 1.1) est modélisé par la notation de BPMN.*

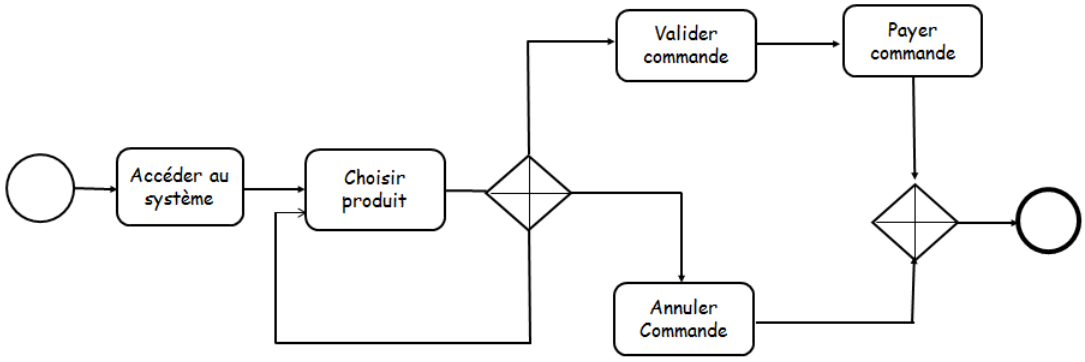

FIGURE 1.8 – BPMN pour commande client

BPMN se compose des blocs de construction de base permettant de représenter un PM, qui sont les éléments nécessaires pour rendre la lecture de diagrammes BPMN compréhensibles [17]. Les éléments manipulés sont les suivants.

- Objets de flux : événements (*cercles*), activités (*rectangles aux coins arrondis*) et passerelles (*losanges*) ;
- Objets de connexion : composés principalement de flèches qui indiquent le flux de séquences (*flèches pleines*), le flux de messages (*flèches en pointillés*) et les associations ;
- Artefacts : objets de données, groupes et annotations ;

b) BPEL

Le langage BPEL (**B**usiness **P**rocess **E**xecution **L**anguage) est devenu une norme pour la mise en œuvre des processus métiers d'entreprise. C'est une extension des langages de programmation impératifs avec des constructions spécifiques aux implémentations utiles aux services web.

Une spécification d'un processus BPEL et décrite dans la **partie Abstract** d'un programme BPEL et elle relie de nombreuses activités par des structures de contrôle. Les activités peuvent être des activités de base ou des activités structurées. Les activités de base correspondent à des opérations atomiques, telles que [18] :

- invoke : déclenche une opération sur un service Web ;
- receive : permet d'attendre un message d'un partenaire ;
- quit, terminer l'intégralité de l'instance de service ;
- cancel : ne rien faire ;

Pour permettre la présentation de structures complexes, les activités structurées suivantes sont définies :

- sequence : définit l'ordre d'exécution ;
- flow, pour le routage parallèle ;
- commutateur : if ... then ...else : pour le routage conditionnel des activités ;
- select : pour les conditions basées sur le temps ou les déclencheurs externes ;
- scope pour regrouper les activités en blocs auxquels des gestionnaires d'événements et d'exceptions peuvent être attachés ;

BPEL utilise le standard XML comme modèle de données. Ces processus sont mis en œuvre en échangeant des messages structurés de types synchrone et asynchrone. Ces messages associés aux activités forment une séquence logique qui forme un processus, qui à son tour peut être synchrone ou asynchrone. De plus, il implémenter son flux sous une forme de couches similaires aux autres langages de programmation (*Java, C++, ...*), en tenant compte des différentes exceptions de gestion possibles.

Les processus BPEL ont des cycles de vie bien définis qui peuvent être contrôlés en appliquant des contraintes spécifiques. La force du langage BPEL réside dans sa structure de processus, qui se déroule sur un seul fichier XML.

Nous observons que les modèles de représentation précédent ce concentre essentiellement, sur les activités réalisées. Or, ces activités manipulent lors de leur exécution des données. De ce point de vue, il est impératif de prendre en compte cette dimension des données et qui est indissociable de la gestion des PM. C'est ce que nous allons exposé dans la prochaine section.

1.7 Les données des processus métiers

Le concept de processus métier présente une certaine particularité relative au fait qu'il existe plusieurs types de données associées à leur spécification et à leur exécution. En ce sens, la modélisation des PM ainsi que leur exécution passe par la manipulation de données spécifiques. Celles ci peuvent porter sur les différents types suivants.

- les données de description des modèles des PMs ;
- les données d'exécution des différentes instances ;
- les données relatives aux ressources ;

Dans cette section, on aborde ces trois types de données.

1.7.1 Données relatives aux modèles

Comme explicité au début du chapitre, les modèles de PM permettent de spécifier les données des activités et les règles métier qui conditionnent leur exécution. Ainsi, le modèle permet de créer une représentation visuelle des données et d'illustrer comment les différents éléments sont liés les uns aux autres. Il répond également aux questions « qui, quoi, où et pourquoi » relatives aux éléments de gestion et à la logique métier de la procédure en question. De ce fait, les données relative aux modèles sont les données nécessaires aux processus ainsi que les règles et concepts métier associés. A titre d'exemple des tables relationnelles ou des fichier XML, des relations entre les structures de données, comme les clés étrangères sont exploités pour stocker les données des PMs. En se sens, des entités et des attributs spécifiques sont définis et les structures de base de données ou de fichiers spécifiques seront utilisées pour assurer le stockage permanent de ces données dans les systèmes BPMS.

Exemple 1.7 *Données du modèle commande client*
Le processus métier commande client de l'exemple 1.1 intègre dans sa description l'ensemble des données suivantes :

- *Les différentes étapes du processus ;*
- *Les transitions possibles entre les étapes du PM ;*
- *Les conditions et les contraintes utiles à exécution des activités.*

Ces données sont stockés dans des structures adéquates, par exemple des tableaux, des listes, des piles, des arbres, ou encore des bases de données relationnelles.

1.7.2 Données sur les ressources

Au sein d'une entreprise, les ressources sont les moyens mis à la disposition des usagers afin de bien faire fonctionner leurs activités. Ce sont des éléments indispensables pour une bonne gestion de toute organisation. Comme, celle dernière déploie des PMs pour réaliser ses objectifs, alors PMs manipulent diverses ressources durant leur exécution. Il peut s'agir d'un déploiement de ressources humaines qui est une information liée à la gestion du personnel. Par exemple, des informations sur les employés qui réalisent les activités, ou les possibilités de validation de certaines tâches, ... D'autres ressources utiles à l'exécution des PMs sont matérielles et englobent tout ce que possède déjà l'entreprise, mais également tout ce qui sera nécessaire à la réalisation du projet, telles que les lieux (*salles, bâtiments, terrains, ...*), matériels et équipements (*ordinateurs, téléphone, ...*), logiciels, outils, machines, matériaux de construction. Le dernier type de ressources sont les ressources financière apportées principalement par des propriétaires (*actionnaires et associés*) de l'entreprise. A signaler enfin que les s ressources associées aux processus métiers peuvent être internes ou externes à l'organisation.

Comme les types de ressources précédents sont utiles et souvent incontournables pour l'exécution des PMs, alors il s'avère impératif de les gérer et de les prendre en charge durant le cycle de vie des PMs. Par conséquent, il est impératif que ces données soient stockées de manière adéquate afin que les informations contenues soient accessibles aux instances qui vont les consommer.

Exemple 1.8 *Ressources du processus métiers gestion de* commande client *(exemple 1.1)*

- ***Ressources humaines :*** *Client, agent commercial, livreur, vendeur, employé de l'entrepôt, ... ;*
- ***Ressources matériels :*** *Matériels informatiques, moyens de transports pour livraison, système de gestion des commandes... ;*
- ***Ressources financières :*** *données sur les soldes clients, systèmes comptables, trésorerie ... ;*

1.7.3 Données d'execution

Une fois le PM est déployé, il devient disponible et donc invocable par les différents utilisateurs potentiels. En effet, chaque exécution du PM par un utilisateur génère une instance du PM en question et la progression de l'exécution se matérialise par la notion de trace d'exécution. D'autre part, pour un PM donné, on peut avoir à un moment donné un nombre important d'instances qui sont en cours d'exécution au même temps et chacune pouvant avoir atteint un niveau d'avancement (état) qui lui est spécifique.

Dans ce qui suit, on va définir les concept d'instance et de trace d'exécution.

a) Notion d'instance d'exécution

Correspond à une invocation particulière d'un PM par un utilisateur.

Instance	ID	Utilisateur	Activité courante	Start-time
1	201140	Lilia	VP(Validé produit)	10 :05
2	125695	Mohammed	PC(Payer commande)	22 :33
3	365916	Nada	A(Annuler)	08 :08
4	201149	Ahmed	L(Livraison)	12 :31
5	601240	Salim	AS(Accès au système)	10 :05

TABLE 1.1 – **Quelques instances du processus commande client**

Exemple 1.9 *En se référant au PM de l'exemple 1.1 qui montre le processus commande client, on va illustrer quelques instances d'exécution de ce processus. Comme il apparait dans la table 1.1, chaque instance est caractérisée par un ensembles d'attributs qui sont :**ID**,**Utilisateur**,**Activité courante**,**Start-time**.*

a) Trace d'exécution

Un trace d'exécution représente l'historique des activités réalisées lors de l'exécution d'une instance, depuis son lancement jusqu'à son état actuel.

Exemple 1.10 *La table 1.2, illustre les traces de l'exécution des instances du processus commande client montrées dans la table 1.1 précédente.*

Instance	Trace
1	début.AS.utilisateur identifier.(CP)*.produit choisi.VP.
2	début.AS.utilisateur identifier.(CP)*.produit choisi.VP.commande validé.PC
3	début.AS.utilisateur identifier.(CP)*.produit choisi.A.commande Annulée.PC
4	début.AS.utilisateur identifier.(CP)*.produit choisi.VP.commande validé.PC .commande payée.L
5	début.AS.utilisateur identifier

TABLE 1.2 – **Exemple de trace d'exécution des instances du PM commande client**

1.8 Conclusion

Dans ce premier chapitre, nous avons introduit la notion de processus métier et nous avons exposé les concepts et les techniques qui leur sont associés. L'accent a été mis, particulièrement, sur leurs modèles de représentation et un panorama des différents modèles a été dressé. Nous avons terminé le chapitre, par un détour sur l'aspect données manipulées par les processus métiers.

Néanmoins, il faut d'ores et déjà signaler que ces données peuvent être issues de diverses sources et peuvent avoir des formats différents, d'où la nécessité de réfléchir à des mécanismes permettant leur intégration et leur uniformisation en vue de leurs exploitation optimale.

Dans cette perspective, le chapitre prochain sera dédié aux techniques d'intégration des données.

Chapitre 2

Les techniques d'intégration des données

2.1 Introduction

L'émergence des systèmes d'informations automatisés et leur omniprésence dans différents secteurs d'activités, conjuguées avec l'apparition de la discipline **Big data**, ont complètement bouleversé notre rapport avec la gestion et le traitement des données. En effet, à l'heure actuelle les données sont générées et collectées à une échelle sans précédent et avec un rythme accru. Par conséquent, le désir d'analyser et d'extraire de la valeur ajoutée de ces données afin de prendre les décisions adéquates exigent des intégrations et des transformations permettant d'aboutir à des format communs et des standards universels. Cette exigence est due au fait que ces données sont issues de diverses sources et sont relatives à différentes applications. Parmi les exemples les plus illustratifs sur le phénomène d'explosion de la masse de données, on peut cités : *les documents Web, les fichiers logs du commerce électronique à grande échelle, les échanges à travers les réseaux sociaux, les informations manipulées par les réseaux de capteurs, les données relatives à l'astronomie, ...*. Ainsi, l'intégration des données devient la clé pour concrétiser les promesses du **Big data**.

Dans ce chapitre, nous allons nous focaliser sur cet aspect d'intégration des données et analyser ses différents aspects. Dans un premier temps, nous allons étudier l'intérêt de cette notion d'intégration pour laquelle nous proposerons plusieurs définitions, puis nous analyserons les techniques d'intégration existantes dans la littérature. Un regard particulier sera accordé aux techniques et outils d'intégration ETL. Nous commençons le chapitre par une discussion sur l'intérêt de l'intégration des données.

2.2 Intérêt de l'intégration des données

Comme les processus métiers doivent spécifier un ensemble de tâches ou d'activités interdépendantes qui permettent d'atteindre des objectifs de gestion de l'organisation, alors la modélisation de tels processus peut faciliter la compréhension et le fonctionnement des procédures et règles de gestion associées au PM. D'autre part, lors de l'accomplissement des différentes étapes du PM, des données sont manipulées et produites au fur et à mesure de sa progression. En effet, en plus des données d'entrée, d'autres nouvelles données d'exécution sont générées. Dans ce contexte, le système de gestion des processus métier (**BPMS**) doit assurer une prise en charge adéquate de la gestion, du stockage et du rafraichissement de ces données. Néanmoins, les BPMS doivent faire face à de nouveaux défis dans le domaine du big data. De plus, les processus métier font partie d'un domaine complexe où le stockage et l'intégration des données sont des étapes importantes pour les futures applications d'analyse et de prise de décision. Cependant, les systèmes de gestion de bases de données relationnelles (SGBDR) ont des difficultés à exécuter des données à

partir d'environnements hautement distribués dans divers systèmes hétérogènes et à très grande vitesse. De ce fait, la quantité de données numériques devient massive, disparate, diversifiées et en continuelle expansion, on parle de Big Data ou données massives.

La technologie du big data est en pleine évolution et elle a été adoptée dans divers domaines, tels que le marketing, l'e-commerce, l'e-santé, l'e-learning, l'e-gouvernement ... [19]. Les Big data sont caractérisées par les 3V (Volume, Variété et Vitesse), est sont définies dans [20] comme *"un actif d'information à grand volume, à grande vitesse et multivariété qui nécessite des formes de traitement de l'information rentables et innovantes pour une meilleure compréhension et une meilleure prise de décision."*. Les concepts 3 V peuvent être brièvement décrits comme suit :

- **Volume :** fait référence à une grande quantité de données de tout type provenant de différentes sources.
- **Diversité :** fait référence à différents types de formats des données, tels que des vidéos, des images, du texte, de l'audio, ... qui sont collectées via des capteurs, des smart-phones ou des réseaux sociaux. De plus, ces données peuvent être sous un format structuré ou non structuré.
- **Vitesse :** fait référence à la vitesse de transmission des données, car le contenu des données est en constante évolution.

Cette définition a subi plusieurs améliorations qui tentent de prendre en compte les évolutions technologiques. Par exemple, les auteurs dans [21] précisent que :*"Les données massives sont une ressource d'information à volume élevé, à grande vitesse et/ou diversifiée qui nécessite de nouvelles formes de traitement pour améliorer la prise de décision, la découverte d'informations et l'optimisation des processus."*. D'autres auteurs, chercheurs et ingénieurs en ajoutent de la valeur et de la précision aux définitions précédentes et étendent les 3V de base à 4V et 5V.

Dans le concept de big data, ce n'est pas la quantité de données qui génère vraiment de nouvelles idées, mais la combinaison des 3V. De plus, le domaine des mégadonnées se développe rapidement, avec un accent particulier sur le stockage et le traitement de grands ensembles de données. Ainsi, de nouvelles méthodes de collecte, de traitement et d'analyse de grandes quantités de données ont été proposées et adoptées. Par conséquent, l'intégration de l'information est l'un des enjeux centraux des systèmes d'information manipulant des données massives. De la discussion précédente se pose, alors, les questions inhérentes à la manipulation des données des processus métiers.

- En quoi consiste l'intégration de données ?
- Quelles sont les techniques existantes permettant d'assurer une intégration correcte des données massives, telles que explicitées ci-dessus ?

Dans la section suivante nous essayerons de répondre à ces préoccupations.

2.3 Présentation de l'intégration des données

L'intégration des données offre de grands avantages pour les entreprises qui l'utilisent de plus en plus dans le cadre de la gestion de leurs données. Néanmoins, il n'existe pas

d'approche unique pour l'intégration des données, ni de définition standard. En effet, plusieurs définitions ont été proposées en vue de spécifier ce concept d'intégration. Dans ce qui suit, nous donnons deux définitions et nous faisons ressortir les aspects fondamentaux des techniques d'intégration.

2.3.1 Définitions de l'intégration de données

Définition 2.1 *L'intégration de données est le processus technique et métier consistant à combiner des données provenant de différentes sources pour exploiter pleinement les données. Plus simplement, l'intégration de données consiste à rassembler des sources de données disparates dans une vue unifiée. Elle permet aux outils analytiques de produire des informations exploitables [22].*

Une autre définition est données dans [23].

Définition 2.2 *L'intégration des données est le processus consistant à combiner des données provenant de sources disparates dans une vue unifiée. Ce processus assure l'importation, le nettoyage en passant par la cartographie et la transformation pour cibler les gisements, rendant finalement les données plus utilisables et utiles pour les utilisateurs.*

De ces deux définitions nous pouvons retenir que l'intégration des données désigne le processus consistant à :

- l'échange de données des sources vers des structures cibles ;
- la copie de données dans des formats standards et unifiés
- le déplacement et la transformation de données

On constate que l'intégration de données est un élément essentiel de nombreux projets de gestion de données critiques, tels que la création d'entrepôts de données d'entreprise, la migration de données d'une ou plusieurs bases de données vers une autre et la synchronisation des données entre différentes applications. Par conséquent, les entreprises utilisent diverses techniques d'intégration de données pour intégrer des données provenant de différentes sources afin de créer une version unique de la réalité de l'entreprise.

2.3.2 Exemples illustratifs d'intégration de données

Dans cette section, nous exposons quelques exemples concrets permettant d'illustrer le principe d'intégration des données dans différents domaines pratiques.

- **Intégration d'attributs de plusieurs tables** : pour aboutir à une base de données unique en utilisant l'opération de **jointure** naturelle (sur la base d'un même Identifiant commun) assure cette intégration. Exemple : Soient les tables **produit** et magasin. La jointure de ces deux tables sur la base du code produit fournira une nouvelle table contenant les lieux de stockage de chaque produit dans chaque magasin.

- **Intégration d'enregistrement de bases de données** : La commande SQL Append assure l'adjonction d'une table d'une base de données avec une autre ayant la même structure. Base de données des étudiants de l'université de Guelma avec celle des étudiants de l'Université de Annaba.

Les deux exemples précédents sont élémentaires et sont pris en charge directement par les SGBD. Dans ce qui suit, on va se focaliser sur des techniques d'intégration des données plus avancées et on va exposer un panorama des techniques d'intégration existante dans le domaine.

2.4 Les techniques d'intégration

Le processus d'intégration consiste à combiner des informations provenant de sources diverses, y compris des bases de données, telles que les BDD relationnelles. Donc, nous commençons par un bref rappel sur les bases de données relationnelles.

2.4.1 Rappel sur les BDD relationnelles

Le modèle relationnel est une méthode très populaire d'organisation des données, plus explicitement :

Définition 2.3 *Une base de données relationnelle est une collection de données organisées dans des tables formellement définies à partir desquelles les données peuvent être consultées et assemblées sans avoir à réorganiser les tables de la base de données [24].*

En effet, une base de données relationnelle est une structure qui stocke et donne accès à des données liées les unes aux autres. Les bases de données relationnelles sont basées sur le modèle relationnel qui est un moyen intuitif et simple de représenter des données dans des tableaux. Dans une base de données relationnelle, chaque ligne d'une table est un enregistrement avec un identifiant unique, appelé clé. Les colonnes du tableau contiennent les attributs des données et chaque enregistrement a généralement une valeur pour chaque attribut, ce qui facilite l'établissement de relations entre les points de données.

Pour gérer une BDD relationnelle, un système logiciel nommé système de gestion des bases de données relationnelles (**SGBDR**) est indispensable. C'est un logiciel standard basé sur les principes suivants [25] :

- la définition des données sous forme de relations ;
- la manipulation des données par un langage déclaratif ;
- l'administration des données.

C'est un système qui permet aux utilisateurs d'interagir avec la base de données. Pour cela, le SGBD doit disposer d'un modèle qui définit l'organisation des données.

L'interface standard pour les bases de données relationnelles est SQL (Structured Query Language). Les commandes SQL sont utilisées pour interroger de manière interactive les informations contenues dans la base de données et collecter des données pour les rapports [24].

Le principale avantage du BDDR est la cohérence élevée des données. Ce modèle permet un stockage adéquat qui contribue à leur cohérence et qui favorise leur intégration [26].

La figure 2.4.1 illustre un exemple d'un schéma d'une base de données relationnelle relatif au traitement d'une "Commande produit". A titre d'illustration, considérons, la table Client, qui peut inclure les champs suivants :

- Numcl (Numéro client) ;
- Nomcl (Nom client) ;
- Prénomcl (Prénom client) ;
- Adrcl (Adresse client) ;
- Telcl (Téléphone client) ;

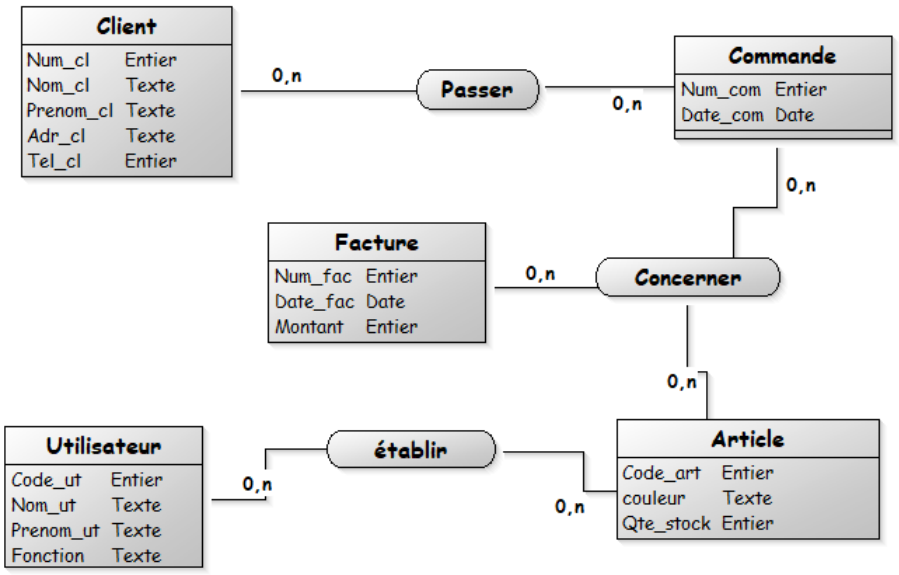

FIGURE 2.1 – Exemple d'une BDDR "Gestion des commandes produit"

Il existe plusieurs techniques principales de l'intégration des données. Il s'agit de Global As View (GAV), Local As View (LAV), Entrepôt de données(Data Warehouse), Adaptateur, Middleware

2.4.2 GAV : Global As View

Cette technique est utilisée pour l'intégration des informations d'entreprise qui crée des bases de données distinctes appartenant à une entreprise et qui fonctionnent ensemble et pour intégrer des catalogues comme la combinaison des informations sur le produit de chaque fournisseur[27]. La méthode GAV implique de définir les éléments du schéma global

comme un ensemble de vues de la source de données. A partir du schéma global, l'utilisateur peut formuler des requêtes. Elles seront converties en sous-requêtes sur diverses sources de données. La reformulation des requêtes sur le schéma global en sous-requêtes se fait en remplaçant les éléments de la requête globale par leurs définitions [28].

GAV est un médiateur (*Wrapper en anglais*) d'intégration de données basé sur la vue. Le schéma global agit comme une vue du schéma source, c'est-à-dire que le schéma intermédiaire est décrit en termes de schéma local. Étant donné une requête au schéma global, la médiation suivra les règles et les schémas existants pour transformer la requête en une requête spécifique à la source. Il envoie une nouvelle requête au wrapper pour exécution. Le Wrapper recherche toutes les expressions possibles et comment les combiner pour répondre à une requête donnée.

Exemple 2.1 *Dans le domaine d'enseignement supérieur, chaque université dispose de sa propre base de données d'étudiants et chaque base de données a son propre format. Ainsi, chaque BDD est considérée comme une vue locale. D'autre part, le ministère dispose d'une base de données intégrée contenant toutes ces bases de données qui est appelé concept global. La figure 2.4.2 illustre un exemple de l'approche GAV.*

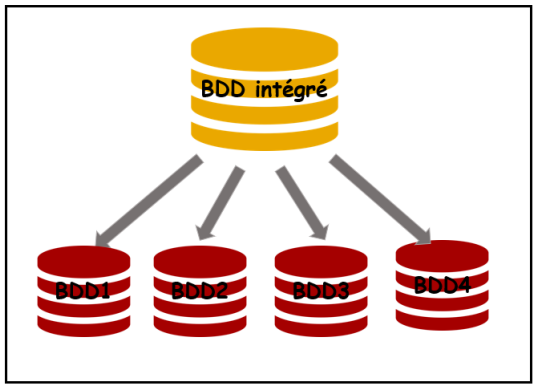

FIGURE 2.2 – Approche GAV d'intégration

2.4.3 Local As View (LAV)

L'approche d'intégration LAV consiste à décrire les sources de données en fonction des éléments du schéma global. Elle fait le cheminement inverse que celui adopté par l'approche GAV [28].

Contrairement à l'approche GAV, l'approche LAV est une technique d'intégration de données basée sur la vue pour effectuer l'intégration de données, c'est-à-dire rechercher et combiner des données provenant de diverses sources. Elle permet de décrire chaque schéma local comme une fonction sur le schéma global. Ici, une source de données est définie comme une vue du schéma fourni. L'architecture est conçue de manière à rester stable même lorsque certaines sources de données rejoignent ou quittent le système intégré. Ainsi, LAV permet d'ajouter d'autres sources ou de supprimer des sources de manière

autonome au système intégré. Elle est utilisée dans les systèmes d'intégration de données pratiques ainsi que pour la vérification et la de récupération des données[29].

L'approche LAV permet d'ajouter très facilement des sources d'information, elle n'a aucun impact sur l'architecture globale. D'autre part, la construction des réponses aux requêtes est complexe, contrairement à la construction des réponses dans les systèmes utilisant les méthodes GAV, qui remplacent simplement les prédicats du schéma de requête global par leurs définitions [30].

La figure 2.4.3 illustre l'approche LAV.

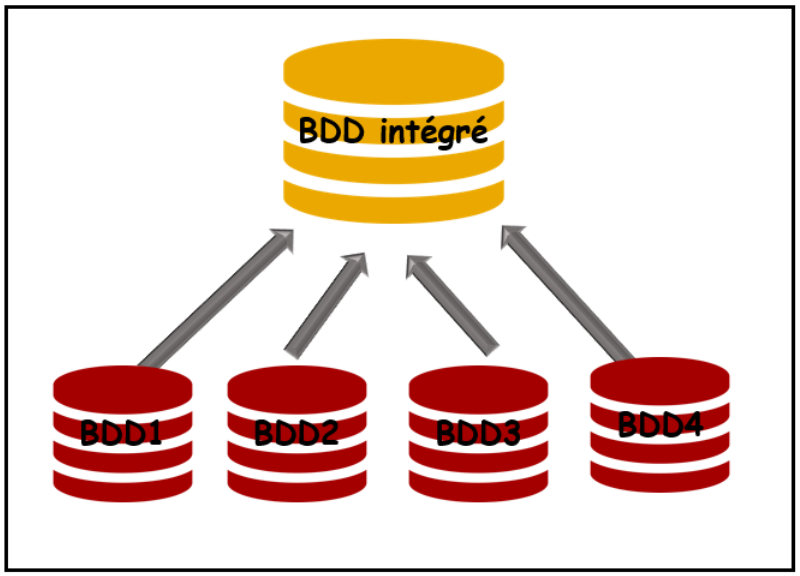

FIGURE 2.3 – Approche LAV pour l'intégration des données

2.4.4 Enterprises Application Intégration(EAI)

L'intégration d'applications d'entreprise EAI est une solution qui a pour but d'assurer la connexion entre différentes applications du système de données manipulées et leur conversion extraites dans un format commun. C'est une approche qui correspond à un ensemble de technologies, d'outils et de framework permettant une intégration en temps réel basée sur des messages entre des applications disparates. L'envoi de ces messages est déclenché par des modifications ou des paramètres dans chaque application. Les données prises en charge sont intégrées dans le cadre de la solution EAI puis transférées vers un point central appelé Middleware ou inter-giciel qui va les exploiter.

Il existe deux modèles d'EAI :

- **Le modèle point à point**
  Ce modèle assure la communication des applications entre elles et avec les éléments de l'environnement informatique. Par conséquent, chaque ressource doit être personnalisée en fonction de toutes les ressources auxquelles elle est connectée. C'est une tâche fastidieuse, donc le modèle est très sujet aux erreurs. Pour ne rien arranger,

la maintenance du modèle se complexifie à chaque mise à jour de l'infrastructure et des applications.

- **Le modèle en étoile**
  Ce nouveau modèle surmonte les insuffisances du modèle précédent en offrant un point de connexion central (**noyau** ou **le cœur**) qui interconnecte toutes les applications et les services. Il permet, ainsi, de faire une maintenance individuelle grâce aux liens qui relient le noyau aux applications et aux services. De cette manière, il est possible d'élaborer des applications plus spécialisées et réserver les tâches d'intégration au noyau et aux liens entre composants.
  Le principal inconvénient de cette approche réside dans la centralisation du noyau, car il devient le point unique de défaillance de tout le système et des communications au sein de l'infrastructure. Dans un modèle en étoile, toutes les intégrations dépendent, par définition, du bon fonctionnement du noyau.

la figure 2.4, illustre les deux modèle de d'EAI

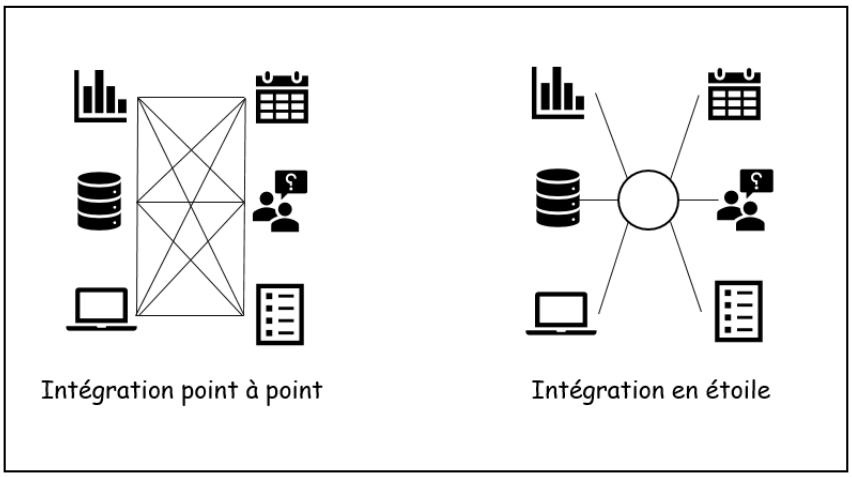

FIGURE 2.4 – Les deux modèle de d'EAI

2.4.5 Entreprise Service Bus (ESB)

La technologie d'intégration des données basée sur l'ESB remplace avantageusement l'intégration en étoile. Elle représente un outil d'abstraction orienté messages qui offre des modules de service entre les applications. C'est une nouvelle génération d'intégration d'application, considérée comme l'héritière de la solution EAI qui permet de surmonter les limitations de l'EAI. Pour assurer la description des messages et les services web pour l'échange de données, un ESB est construit sur des standards ouverts, tels que le protocole XML, afin de connecter les applications et les données entre elles. L'ESB tient également la fonction de point central où tous les modules de service sont partagés, redirigés et organisés. Mais, la solution ESB n'est pas non plus la panacée, surtout dans le cas d'une entreprise qui croît et qui acquiert de nouvelles ressources, car elle requiert alors une plus grande rapidité au niveau des propriétés et des ressources logicielles [31, 32].

La figure 2.5 suivante schématise le fonctionnement d'une solution d'intégration ESB.

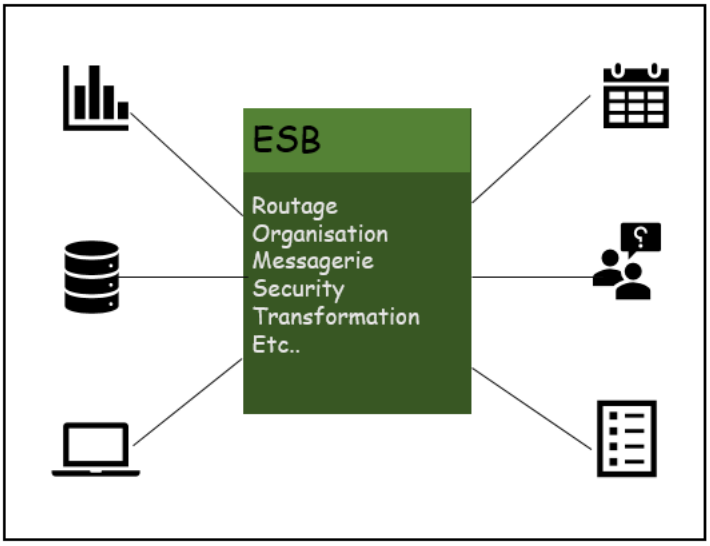

FIGURE 2.5 – L'intégration ESB

2.4.6 Entreprise Information Intégration (EII)

L'intégration d'information d'entreprise (EII) est une approche d'intégration articulée autour d'un système logiciel qui fournit une vue unifiée des données de l'entreprise, où les sources de données sont fédérées à l'aide d'une base de données virtuelle, de manière transparente aux applications utilisant ces données. Ces sources de données dispersées sont consolidées et intégrées dans une structure intermédiaire. Ainsi, toute requête à la base de données virtuelle sera décomposée en sous-requêtes correspondantes est envoyées aux sources respectives. Par la suite, les réponses aux requêtes partielles sont assemblées en un résultat unifié et consolidé. La solution EII permet de consolider uniquement les données à utilisées, uniquement au moment de leur utilisation effective (source data pulling). Cependant, le traitement en-ligne des données peut cependant entraîner des délais importants [33].

2.4.7 Entreprise Ressource Planning (ERP)

Entreprise **R**essource **P**lanning (ERP) est un progiciel intégrée de gestion utilisé par les organisations afin de gérer les fonctions quotidiennes, telles que la fonction commerciale, la comptabilité, l'approvisionnement, la gestion de projet, la gestion des risques et la gestion des ressources humaines. Généralement, une suite ERP complète inclut aussi la gestion des performances de l'entreprise, un logiciel qui aide à planifier, budgétiser, prévoir et rendre compte des résultats financiers d'une organisation.

Dans une perspective de gestion des PM, les systèmes ERP relient un nombre important de processus métier et permettent d'échanger les flux de données entre eux. Ils éliminent, ainsi, la duplication des données et assurent l'intégrité des données avec une source unique reflétant la réalité. Les systèmes ERP sont essentiels pour gérer des milliers d'entreprises de toutes tailles et dans tous les secteurs d'activités, en collectant les données transactionnelles partagées d'une organisation à partir de plusieurs sources.

2.4.8 Les techniques d'adaptateurs

Un adaptateur est un outil logiciel qui permet à un (*ou plusieurs*) médiateurs d'accéder au contenu des sources d'information dans un langage unifié.

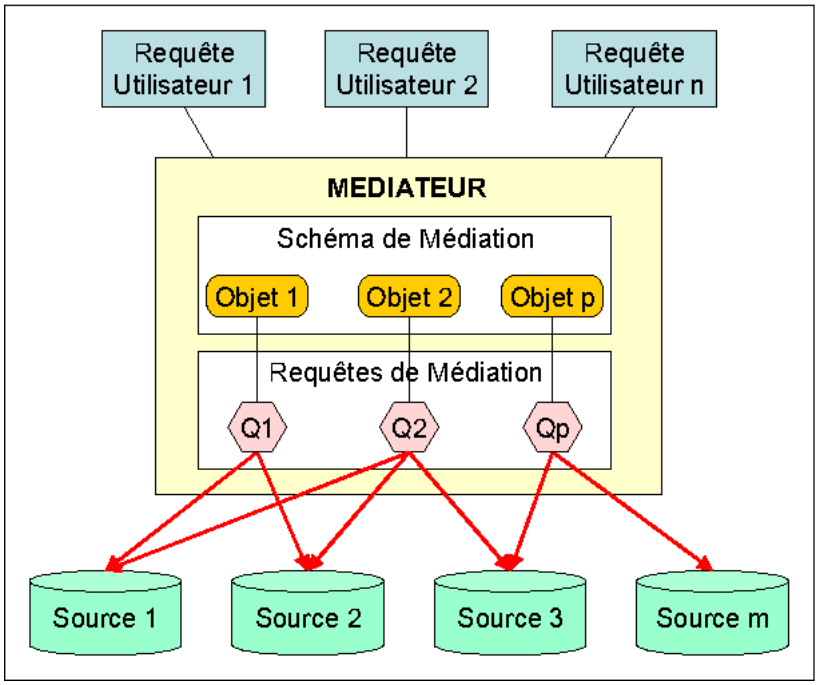

FIGURE 2.6 – L'intégration ESB

Comme illustré dans la figure 2.6, un adaptateur permet d'extraire et d'acheminer les données des différentes sources et de les convertir dans un format cible, appellé "**schéma de médiation**". Pour cela, il accède aux bases de données, aux fichiers, aux systèmes de messagerie, aux applications d'entreprise et aux autres sources et cibles de données, puis il établit une correspondance entre la représentation locale de l'information et sa représentation dans le modèle de médiation [34].

Le principe de fonctionnement du médiateur est basé sur des modules spécifiques qui effectuent des mises en relation des requêtes des différents utilisateurs avec le schéma de médiation, en transformant les requêtes d'origine en des requêtes qui soient conformes aux avec le schéma contenu dans les sources de données. Ces requêtes dont appelées "**requêtes de médiation**". Les requêtes et les réponses d'une source donnée sont dans leur propre format, et l'adaptateur convertit la demande au format de la source et convertit la réponse dans un format adéquat qui correspond à la base de données ou de connaissance cible, donc, au schéma de médiation du système. C'est cette réponse au format obtenu qui est ensuite retransmise au médiateur, où elle est combinée avec d'autres réponses d'autres adaptateurs [35].

Vue l'importance des entrepôts de données, en tant que technique d'intégration des données, nous allons les aborder à part dans la prochaine section.

2.5 Les Entrepôts de données ou Data warehouse

Les entrepôts de données ont prouvé leur importance dans de nombreux de projets d'intégration et en informatique décisionnelle ou business intelligence. En effet, l'entrepôt de données est une architecture pouvant servir de base à des applications décisionnelles.

De manière très simple, on peut voir une entrepôt de données comme une base de données multidimensionnelle. C'est à dire qu'elle prend en compte plusieurs dimensions ou axes de données. Les données qui alimente l'entrepôt de données sont collectées à partir de diverses sources afin de créer un référentiel d'informations communs à l'ensemble des sources. Les données peuvent être extraites des sources de production, ou systèmes d'information opérationnels, comme elles peuvent parvenir de sources externes à l'entreprise. Ces données sont récupérées de manière instantanée ou périodiquement. Pour être utilisables, toutes les données des systèmes distribués doivent être organisées, coordonnées, intégrées et finalement stockées pour fournir aux utilisateurs une vue globale de l'information.

En résumé, on peut percevoir un entrepôt de données comme une base de données relationnelle conçue pour l'interrogation et l'analyse de données, la prise de décision et les activités de type informatique décsionnelle, plutôt que de traiter des transactions ou d'autres utilisations traditionnelles des bases de données [36].

2.5.1 Illustration de l'usage des Entrepôts de données

Dans le domaine de la santé, les entrepôts de données, comme référentiels centraux, sont utilisés pour enregistrer les informations sur les patients des différentes unités de l'organisation médicale. Cela inclurait les informations personnelles des patients, les transactions financières avec l'hôpital et les données d'assurance, les dossiers de laboratoires, de consultation et les opérations effectuées sur les malades et ainsi de suite.

Comme illustré dans la figure 2.7, toutes ces données sont consolidées dans l'entrepôt de données et connecté via le schéma de la base de données.

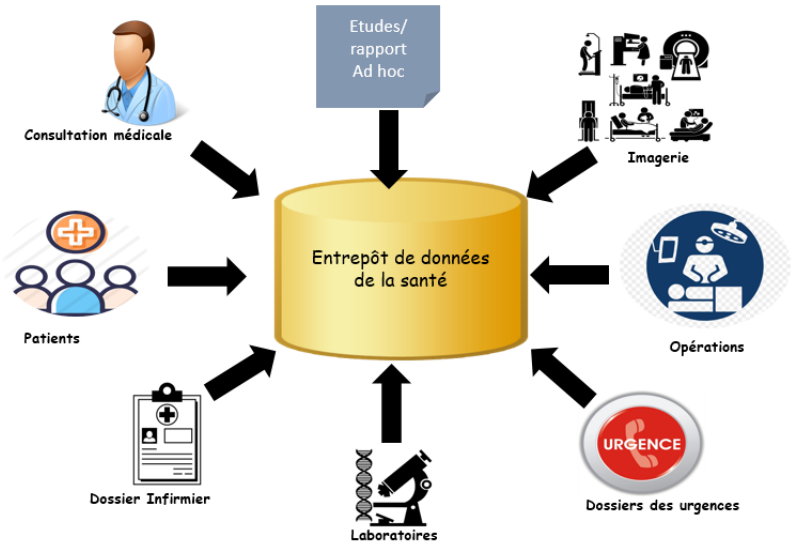

FIGURE 2.7 – Exemple d'entrepôt de données

2.5.2 Architecture des entrepôts de données

Une architecture d'entrepôt de données définit la disposition des données dans les différentes bases de données. Étant donné que les données doivent être organisées et nettoyées pour être utiles, alors il existe trois types de modèles différents à prendre en compte pour construire des couches d'entrepôt de données [37]. Ces trois modèles sont exposés dans ce qui suit.

- **Architecture à un niveau**
  Le but d'un niveau unique est de minimiser la quantité de données stockées. L'objectif est d'éliminer la redondance des données. Cette architecture est rarement utilisée en pratique et elle a prouvé ses limites pour des entreprises ayant des données complexes et avec plusieurs flux de données.

- **Architecture à deux niveaux**
  La structure d'un modèle d'entrepôt de données à deux niveaux consiste à séparer les sources physiquement disponibles et l'entrepôt de données. Cette architecture n'est pas évolutive et ne prend pas en charge un grand nombre d'utilisateurs finaux. De plus, cette configuration engendre des problèmes de connectivité en raison des limitations du réseau. La conception à deux niveaux utilise, à la fois, des serveurs système et de base de données. Les petites organisations qui utilisent des serveurs pour le stockage de données utilisent généralement une architecture à deux niveaux. Bien qu'elle soit plus efficace pour stocker et organiser les données, la structure à deux niveaux n'est pas évolutive. De plus, elle ne prend en charge qu'un nombre nominal d'utilisateurs.

- **Architecture d'entrepôt de données à trois niveaux**
  C'est l'architecture d'entrepôt de données la plus largement utilisée dans les organisations. Elle offre l'avantage de gérer des flux de données bien organisés, en partant des informations brutes contenues dans les sources aux informations cibles de l'entrepôt. Elle se compose de trois couches suivantes : supérieure, intermédiaire et inférieure, que nous allons détaillé par la suite.

  - **Niveau inférieur :** la base de données du serveur (Datawarehouse) agit comme un niveau inférieur. Il s'agit généralement d'un système de base de données relationnelle qui stocke les données sont nettoyées, transformées et chargées à l'aide d'outils back-end.
  - **Niveau intermédiaire :** appelé l'entrepôt de données axé sur OLAP. C'est un serveur OLAP implémenté à l'aide du modèle ROLAP (Relational OLAP) ou MOLAP (Multidimentional OLAP). Pour l'utilisateur, une application à ce niveau présente une vue abstraite de la base de données. Cette couche sert également d'intermédiaire entre l'utilisateur final et la base de données, car elle inclut un serveur OLAP préconstruit dans l'architecture.
  - **Niveau supérieur :** C'est le niveau client qui comprend les outils et l'interface de programmation d'application (API) ou couche client frontale. Ce niveau assure les fonctions qui utilisent l'analyse de données de haut niveau, les enquêtes et les rapports. Il peut contenir des outils de requêtes ou de création de rapports, des outils d'analyse et des outils d'exploration de données.

La figure 2.8, illustre plus en détail les trois niveaux d'un entrepôt de données par le diagramme d'entrepôt de données, qui sont les différents types d'architecture d'entrepôt de données traditionnelle.

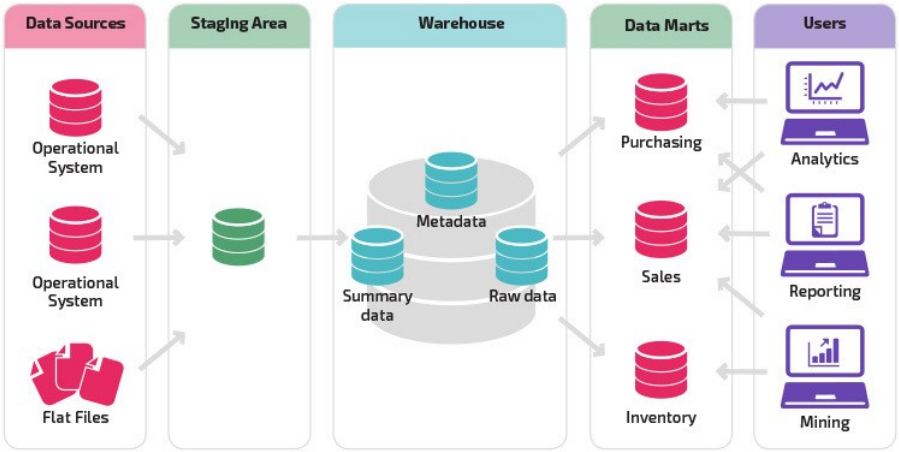

FIGURE 2.8 – Les trois niveaux d'un entrepôt de données d'aprés [1]

2.5.3 Fonctionnement des entrepôts de données

Un entrepôt de données fonctionne comme un référentiel central des données. Les informations proviennent d'une ou plusieurs sources de données, telles que des systèmes transactionnels ou d'autres bases de données relationnelles. Les données peuvent être de différents format, tels que les données structurées, semi-structurées ou non structurées. Une fois ingérées dans l'entrepôt, elles sont traitées et transformées et prêtes pour toutes future utilisation. En effet, les utilisateurs peuvent ensuite y accéder à l'aide d'outils d'informatique décisionnelle (requêtes **OLAP** : Online analytical Processing), de clients SQL ou simplement via de feuilles de calcul. De plus, les entrepôts de données rendent possible l'exploration de données et leur analyse. Ce processus implique de rechercher les tendances des consommateurs et client par le biais d'extraction des modèles à partir de l'analyse des données. Les décidreus s'appuient sur les modèles élaborer à partir de l'entrepôt augmenter les ventes et les revenus de l'entreprise [38].

Après avoir présenter les entrepôts de données, la suite du chapitre sera consacrée aux outils **ETL** (**E**xtrat, **T**ransform and **L**oad) qui sont incontournables vue leur importance, à la fois dans le cadre des outils d'intégration et en même temps en tant que composant fondamental permettant d'alimenter les entrepôts de données.

2.6 La technologie d'intégration basée sur ETL

ETL est un acronyme qui désigne les termes "Extract-Transform-Load". Il s'agit d'un type de logiciel permettant de collecter des données en provenance de sources multiples pour ensuite les convertir dans un format adapté à une Data Warehouse et les y transférer.

2.6.1 Intérêt de la technologie ETL

Les avancées technologiques dans le secteur des système d'information et la démocratisation de l'utilisation d'Internet ont bouleversé le mode de fonctionnement des organisations et les modes de consommation des individus. En effet, on assiste aujourd'hui à une explosion étonnante de l'utilisation des machines de traitement de l'information et de communication équipées de multiples capteurs (*téléphones, ordinateurs, smart TV, smart homes, ...*). Une conséquence directe de cette utilisation intensive est l'explosion des données qui sont générées massivement, on parle de données massives ou (*big data*).

Dans une perspective d'informatique décisionnelle (Business intelligence), l'utilisation rationnelle de grands volumes de données nécessite de les intégrer dans des formats appropriés, et de les rendre disponibles à des fins d'analyse qui permettent de faciliter la prise de décision.

Pour réaliser cet objectif, le processus ETL est une technique courante pour trouver des réponses à ces préoccupations, par la création d'une version unifiée des données et une vision centrale et unique de la réalité de l'entreprise. Assurez la collecte, la transformation et l'utilisation des données en fournissant des modèles et des outils pour extraire des données de sources disparates, tout en les intégrant dans un format unifié d'utilisation et en assurant les liens entre les composants ne peut se réaliser que par le déploiement d'un logiciel dédié qui est l'outil ETL.

D'autre part, les outils ETL permettront de produire et d'exécuter des fonctions spécifiques liées à l'accroissement spectaculaire des données, tels que des outils d'analyse et de reporting (OLAP).

De ce qui précède, le développement des moteurs ETL est devenu un processus omniprésent dans le traitement et la gestion des données qui vise à la préparation des ensembles de données volumineux et disparates pour des objectifs d'informatique décisionnelle basée sur l'exploitation des entrepôts de données. Ainsi, l'intégration des données contenues dans l'entrepôt permettra de concrétiser les scénarios d'analyse de données complexes.

Après avoir exposé l'intérêt des outils ETL, dans ce qui suit on va présenter quelques définitions qui lui sont associés et par la suite on va aborder leur mode de fonctionnement.

2.6.2 Quelques définitions des outils ETL

Les outils ETL sont des logiciels qui sont apparus dans les années 1970 pour intégrer des données éparses et hétérogènes, les préparer et les centraliser dans une structure de données unique.

Plus explicitement, on donne ci-dessous quelques définitions précises.

Définition 2.4 *Un logiciel ETL est un intergiciel (middleware) pour la gestion de gros volumes de données au sein d'un système complexe. Il assure la collecte des données d'une ou plusieurs sources pour les transformer en des des ressources exploitables et enfin, les charger sous une vue centralisée dans un entrepôt de données [39].*

Définition 2.5 *ETL est un procédure permettant d'effectuer des synchronisations massives d'informations entre bases de données qui commence par l'extraction des données des bases de données de production. Puis, leur transformation pour effectuer des calculs, pour les enrichir avec des données externes et enfin, le chargement des données dans les différentes applications décisionnelles [39].*

Une autre définition plus précise est donnée dans [40]

Définition 2.6 *Un ETL (Extract Transform Load) est un middleware permettant d'effectuer des synchronisations de données entre différents systèmes. Il extrait les données, les manipule (conversion, suppression des doublons, ...) et les intègre dans un référentiel commun qui est l'entrepôt de données (datawarehouse).*

D'après ces définitions, on énonce ci-après notre propre définition et que nous allons retenir dans la suite du mémoire.

Définition 2.7 *Un logiciel ETL est un intergiciel (middleware) qui permet de* ***collecter*** *les données en provenance de sources multiples pour ensuite les* ***convertir*** *dans un format adapté à un entrepôt de données, et enfin de les y* ***charger****.*

Cette dernière définition, met en exergue les trois opérations fondamentales utiles à l'intégration des données via un outils ETL. Il s'agit des opérations d'extraction, de transformation et de chargement.

La figure 2.9 suivante, illustre le séquencement des trois opérations assurées par un outil ETL.

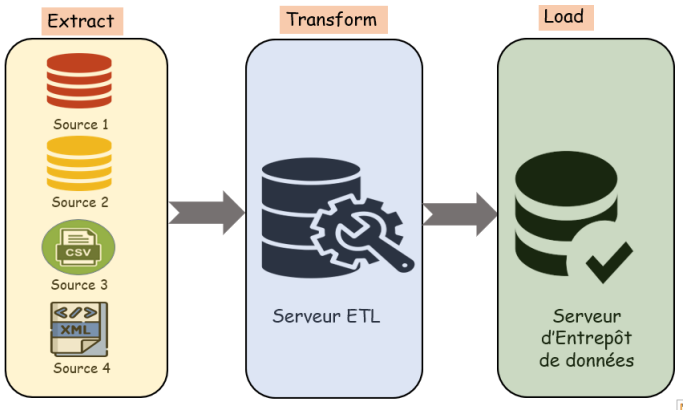

FIGURE 2.9 – Les trois opération d'outil ETL

Ces trois opérations sont examinées en détails dans ce qui suit.

- **Extraction**
  Cette opération permet d'identifier et d'extraire les données des sources utiles à l'alimentation de l'entrepôt de données. Elle peut se faire de manière totale, dans le cas d'un chargement initial, ou bien de manière incrémental si certaines sources ont subi des modifications depuis la dernière exécution du chargement.

- **Transformation**
  Consiste à appliquer certaines règles de transformations aux données pour les nettoyer, les intégrer et les agréger.

- **Chargement**
  Action qui consiste à insérer les données transformées dans l'entrepôt et de gérer les changements des données existantes.

2.6.3 Principe de fonctionnement des outils ETL

Comme il a été déjà expliqué dans la sous-section 2.6.2, les outils ETL assurent l'extraction des données des différentes sources, puis opèrent leurs transformations en des formats plus adéquats et enfin, ils les stockent dans l'entrepôt de données. Partant de ce rôle primordial des ETL, cette section est réservée exclusivement à l'analyse de leur fonctionnement.

La figure 2.10, ci-dessous illustre le principe général de fonctionnement d'un processus ETL. Comme il est observé dans la figure, le mécanisme ETL est un processus incrémental qui passe par plusieurs opérations complémentaires, dont l'explication détaillée est donnée ci-dessous. Les trois premières opérations constituent l'étape d'extraction, les trois suivantes l'étape de transformation et les trois dernières forment l'étape de chargement.

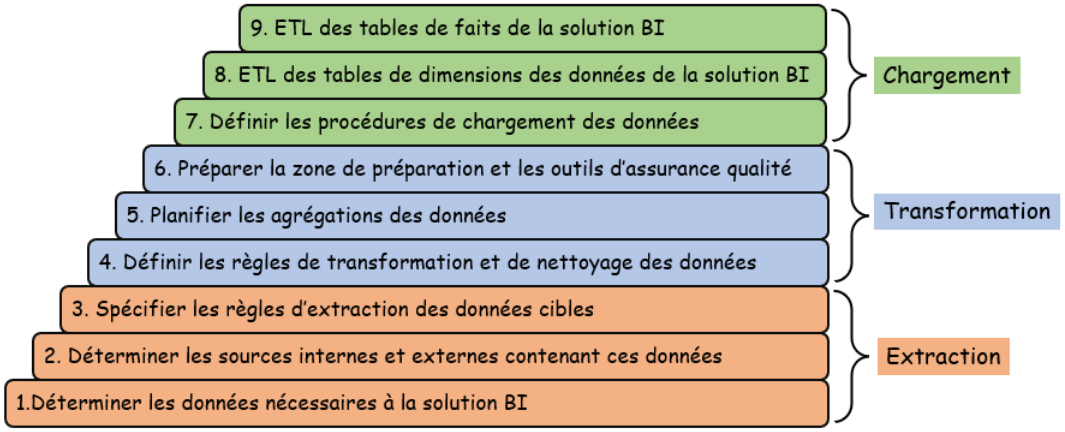

FIGURE 2.10 – Enchainement général des opérations du processus ETL

2.6.4 Analyse des étapes du processus ETL

Chacune des trois étapes est étudiée de manière consistante.

a) Étape d'extraction des données (Extract)

Avant toute action d'extraction, il faut tout d'abord identifier les sources de données. En effet, la grande diversité des sources de données impose une recherche exhaustive des données pertinentes pour la solution BI cible.

a1. Identification des sources

La procédure suivante cerne les actions à suivre pour l'identification des sources et la conduite à tenir pour surmonter les contraintes rencontrées durant cette phase.

- ☐ **Recenser les métriques et attributs de dimension** : consiste à énumérer les attributs cibles nécessaires à l'entrepôt de données ;
- ☐ **Trouver les correspondances source-cible** : Pour chaque attribut cible, il faut trouver la source et l'attribut correspondant de cette source ;
- ☐ **Sélection des sources pertinentes** : Si plusieurs sources sont trouvées lors de l'opération précédente, alors il faut choisir la plus pertinente ;
- ☐ **Consolidation des attributs** : Dans le cas où l'attribut cible exige des données de plusieurs sources, alors il faut formaliser les règles de consolidation ;
- ☐ **Expression des règles de découpage** : Si l'attribut source renferme plusieurs attributs cibles, alors il faut spécifier les règles de découpage. Par exemple, si l'attribut cible Nom-client contient, à la fois le nom et le prénom du client, il faut opérer son découpage en deux attributs distincts (Nom-client, Prénom-client).
- ☐ **Élimination des données manquantes** : Dans le cas où plusieurs attributs cibles contiennent encore des valeurs manquantes, alors il faut inspecter toutes les sources possibles afin de localiser ces valeurs.

a2. Extraction des données

Une fois les sources de données identifiées, l'extraction proprement dite peut être lancée. Cette opération peut être activée de deux manières différentes suivant le contexte de déploiement de la solution BI.

- **Extraction complète** :Ce type d'extraction est employé lors d'un chargement initial des données dans l'entrepôt ou bien lors d'un rafraichissement complet des données (*dans le cas de changement d'une source, par exemple*). L'extraction complète permet de capturer l'ensemble des données à un certain instant (*snapshot de l'état opérationnel*). Néanmoins, le chargement complet peut être très coûteux en temps, du fait que toutes les données seront chargées (d*e plusieurs heures à plusieurs jours en fonction du volume des données manipulées*).
- **Extraction incrémentale** : Cette extraction capture uniquement les données qui ont changé ou ont été ajoutées depuis la dernière extraction. Elle peut être faite en temps-réel, c'est-à- dire au moment où les transactions surviennent dans les systèmes sources (*par des triggers ou par les journaux des transactions*), ou bien en différé, en analysant tous les changements effectués pendant une certaine période grâce à des programmes de comparaison des états des sources pour des périodes différentes (*heure, jours, mois ; . . .*).

b. Étape de transformation des données (Transform)

Avant de charger les données émanant des différentes sources dans l'entrepôt, plusieurs catégories de transformations doivent être opérées sur ces données.

La table 2.1, ci-dessous met en exergue les différents types de transformations qu'un outil ETL standard doit garantir. Pour chaque type de transformation, un exemple illustratif est montré dans la dernière colonne du tableau.

N	Type de transformation	Description	Exemples
1	Redressement de format	Changer le type ou la longueur d'un attribut.	Adresse-Client sur 30 caractères α-numériques au lieu de 40 α-bétiques.
2	Unification du codage de champs.	Consolider les données de sources multiples.	[Homme, Femme], [H,F] , [1,2]
3	Transcription des valeurs en codes..	Faire correspondre des codes à des valeurs.	G :Gros client, M : moyen et F :Faible
4	Pré-calcul des valeurs dérivées...	Appliquer les règles de calcul.	Profit=Prix vente- coûts Prix TTC= Prix HT + TVA
5	Découpage de champs complexes.	Extraire des informations atomiques à partir d'autres articulées.	Prénom, nomFamille à partir d'une chaine de caractère Nomcomplet.
6	Fusion de plusieurs champs..	Regrouper les informations d'une même entité.	Source 1 : Code et libellé produit Source 2 : Type de forfaits et remises Source 3 :Coût de fabrication produit et Conditions de stockage
7	Conversion de jeu de caractères.	Unifier les divers jeux de caractères.	EBCDIC (IBM) vers ACSII UNICODE vers UTF8
8	Conversion des dates.	Harmoniser les formats des dates.	24FEB 2021 vers 24/02/2021 02/24/2021 vers 24/02/2021
9	Conversion des unités de mesure.	Utiliser les mêmes unités de mesures. (système international)	Changer les unités impériales à métrique. Exemple : inch en cm
10	Pré-calcul des agrégats.	Calculer les sommes, produits, moyennes.	Total des ventes par semaine et par mois de chaque produit
11	Déduplication ou redondances des tuples.	Plusieurs enregistrements pour la même entité.	Client au magasin, client en ligne, client potentiel.

TABLE 2.1 – **Les types de transformations assurées par un outil ETL**

Comme il est observé dans le tableau, les transformations peuvent porter aussi bien sur le format des données que sur le contenu lui-même.

c. Étape de chargement des données (Load)

Une fois les données sont extraites et transformées dans des formats adéquats, la dernière étape du processus ETL standard consiste à les charger dans leur nouvel emplacement qui est l'entrepôt de données. En général, les entrepôts de données supportent trois modes pour le chargement des données : le chargement **initial**, le chargement **incrémentiel** et le chargement **complet**.

c1. Chargement initial

Ce type de chargement n'est opéré qu'une seule fois, lors de l'activation de l'entrepôt de données. A cause de la longue durée que peut prendre le processus de chargement initial et afin d'éviter la génération d'incohérences au niveau de l'entrepôt, il est impératif de désactiver temporairement les indexes et les contraintes d'intégrité référentielles relatives aux clés étrangères.

c2. Chargement incrémentiel

Ce type de chargement peut être fait soit en temps réel, soit en batch (*traitement par lots*), mais une fois le chargement initial terminé. Il doit tenir compte de la nature des changements survenus dans les sources de données. A cet effet, une stratégie de gestion des changements doit être adoptée pour chaque situation. On parle de dimension de changement lent (**Slowly Changing Dimension : SCD**) qui peut être de différents types. Les stratégies d'historisation possibles pour les différents CSD sont les suivantes :

- **SCD Type 1** : Consiste à écraser l'ancienne valeur avec la nouvelle valeur. Par exemple, le client a changé son adresse de livraison.
- **SCD Type 2** : Consiste à ajouter une ligne dans la table de dimension pour la nouvelle valeur. Par exemple, si le client a changé son adresse de livraison de A à B, alors préserver les deux valeurs A et B. Donc, on aura deux enregistrements du même client avec deux valeurs distinctes pour l'attribut adresse.
- **SCD Type 3** : Permet d'avoir deux colonnes dans la table de dimension correspondantes à l'ancienne et la nouvelle valeur dans la colonne courante. Pour l'exemple de changement d'adresse, il faut créer une nouvelle colonne dont le libellé sera NOUVELLE-ADRESSE, tout en gardant l'ancienne colonne (ADRESSE).
- **Stratégie Hybride** : On combine les stratégies de gestion des types de changements 2 et 3.

c3. Chargement complet

Ce type de chargement est employé lorsque le nombre de changements rend le chargement incrémental trop complexe. Par exemple, lorsque plus de 20A signaler que pour les

différents types de chargement précédents, certaines considérations supplémentaires sont à prendre en compte, à savoir :

- Opérer le chargement des données en périodes creuses (*entrepôts de données non utilisé*).
- Considérer la bande passante requise pour le chargement.
- Prévoir un plan pour la vérification et l'évaluation de la qualité des données chargées.
- Commencer par le chargement des données des tables de dimension avant celles des faits.

2.7 Conclusion

Vue l'importance du mécanisme d'intégration dans les SI, dans ce chapitre nous nous sommes focalisés sur cet aspect en exposant son intérêt puis les différentes techniques permettant d'assurer d'une manière plus ou moins efficace l'intégration des différentes données issues de sources variées.

Un examen des diverses techniques d'intégration existantes dans la littérature a été présenté. Le chapitre a été clôturé par l'étude approfondie des outils ETL, incontournables dans tout contexte d'intégration. Néanmoins, vue la diversité des données, de leur vitesse d'évolution ainsi que de leur volume qui est de plus en plus consistant, les approches classiques ETL ont montré leur limites et elles sont devenues inadéquates, car ne pouvant plus répondre aux nouvelles exigences de distributivité et au volume croissant des données.

Le prochain chapitre sera dédié à une étude de l'état de l'art des travaux qui ont abordé le problématique de la diversité des données et leur intégration en se basant sur des processus ETL.

Chapitre 3

Problématique et travaux connexes

3.1 Introduction

Les deux premiers chapitres de ce mémoire ont été consacrés à la présentation des processus métiers et à l'exposé des technologies supportant l'intégration des données. Ce chapitre est dédié à la présentation de notre problématique ainsi qu'à l'étude et l'exploration des travaux connexes ayant abordé la question de l'intégration des données, tout en exposant les variantes améliorées des outils ETL. A cet effet, la prise en compte des limites des outils ETL causées par l'accroissement spectaculaire du volume des données manipulées par les différentes applications informatiques sera reconsidérée et revue de manière critique. En effet, cette explosion des données a engendré de nouvelles considérations et contraintes sur des données manipulées, à savoir : le volume, la vitesse et la variété des données massives, connues sous le vocable de (**Big data**).

Après l'exposé des limites de la technologie ETL classique, nous nous focaliserons sur la problématique abordée dans ce projet de fin d'études et qui est relative à l'intégration des données des processus métiers. Afin de mettre en exergue l'intérêt de notre approche, nous discuterons, dans un premier temps, des avancées technologiques ayant rehausser les outils ETL pour affronter les 3 V du big data, puis une étude comparative des travaux de recherche qui ont traité la question de l'impact des données massives sur les approches classiques ETL est dressée.

On commence le chapitre par la présentation des insuffisances des outils ETL classiques.

3.2 Limites des outils ETL

La technologie ETL était une première tentative d'intégration de données et elle répondait de manière suffisante aux besoins du traitement par lots qui était suffisant pour les exigences de gestion des données manipulées par ces outils. Néanmoins, avec les derniers développements technologiques et la démocratisation des technologies de l'information et de la communication, les performances matérielles ont largement évolué. En conséquence, les données internes et externes de toute organisation deviennent de plus en plus diverses, instantanées et volumineuses.

Ce constat a directement impacté le fonctionnement des outils ETL qui étaient destinés à prendre en charge des données locales, généralement relationnelles. En effet, ils n'étaient pas conçus pour gérer le flux de données distantes depuis le cloud. Le problème de la gestion du flux de données est particulièrement aigu dans les environnements en temps réel. A vrai dire, de nombreux environnements d'entreprise modernes ne peuvent pas attendre des heures ou des jours pour que les applications gèrent les ensembles de données. Elles doivent répondre aux nouvelles données en temps réel au fur et à mesure qu'elles sont

générées dans les SI de gestion ou les systèmes de production. En fait, les organisations contemporaines créent et traitent des données dans un flux continu en temps réel. Les caractéristiques des données de tels environnements sont les suivants :

- Elles ont un caractère éphémère (*versatiles et changeantes*).
- Elles proviennent d'utilisateurs mobiles (utilisateurs nomades).
- Elles sont de très grande taille et nécessitent des moyens dédiés pour leur stockage et leur traitement.

Dans ce nouveau contexte, les outils ETL traditionnels restent limités et ne peuvent pas faire face à la monté en charge des données issues d'environnements temps réel. Cela est principalement dû à la grande quantité de données qui interrompe et, parfois, débordent les étapes des processus ETL. Ainsi, après la phase d'extraction et le démarrage de la phase de transformation, la procédure de transformation peut être débordée, engendrant un engorgement à cause de la masse des données extraite dans la zone de transit (*staging area*). Le même phénomène peut se déclencher entre les deux phases de transformation et de chargement. On parle de **débordement du pipeline ETL**. Par ailleurs, il faut du temps et des ressources pour transformer les données extraites des différentes sources et qui sont sauvegardées avant qu'elles ne deviennent obsolètes.

En résumé, il y a deux limites majeurs des outils ETL pour le traitement des flux de données en temps-réel qui se résument aux aspects suivants :

- Pour pouvoir gérer les flux de données en temps réel, toutes les exigences de la phase de transformation ETL, telles que le nettoyage, l'enrichissement et le traitement des données, doivent être effectuées plus fréquemment à mesure que le nombre de sources de données augmente et que la capacité monte en flèche. Les outils ETL traditionnels ne peuvent pas garantir cette tâche ni prendre en compte cette préoccupation.
- Les outils ETL ne peuvent pas gérer instantanément les méga-données, alors que ces dernières peuvent générer de meilleures informations à valeur ajoutée, telles que les informations commerciales qui peuvent être introduites dans des systèmes avancés d'analyse de données ou d'apprentissage automatique. De même ces données massives sont souvent exploitées par les algorithmes d'intelligence artificielle, tels que les systèmes de recommandation, les systèmes de prédiction et dans le cadre de fouilles de processus métiers (*process mining*). Donc, ces données sont très utiles dans un contexte d'informatique décisionnelle.

3.3 Problématique

Dans une perspective d'informatique décisionnelle (*Business intelligence*), l'exploitation rationnelle de données crées lors de l'exécution des processus métiers exige leur intégration dans des formats et des supports adéquats en vue de leur analyse et utile à des fins de prise de décision.

Le processus Extract, Transform and Load (ETL) traditionnel vise à répondre à cette préoccupation en offrant des modèles et des outils permettant d'extraire les données de différentes sources et de les intégrer dans des formats homogènes et uniforme en vue de

leur exploitation. Ce format est communément désigné par le l'entrepôt de données ou (**Dataware house**). Néanmoins, vue la diversité des données, de leur vitesse d'évolution ainsi que de leur volume qui est de plus en plus consistant, les approches classiques ETL ont montré leur limites et elles sont devenues inadéquates, car ne pouvant plus répondre aux nouvelles exigences.

En effet, avec l'augmentation du débit, les évolutions récentes des TIC et leur démocratisation, les données internes et externes à toute organisation sont devenues de plus en plus variées, instantanées et volumineuses. D'autre part, ces données sont stockées dans plusieurs sources disparates qui ont été conçues indépendamment par des concepteurs différents. Ce phénomène entraine une **hétérogénéité** des données, c'est-à-dire que les données relatives à un même sujet sont représentées différemment sur des systèmes d'information distincts. Cette hétérogénéité provient des choix différents qui ont été opérés pour représenter et stocker des faits du monde réel dans des formats informatiques divers, tels que les bases de données relationnelles, des fichiers semi-structurées (*XML*) ou encore des fichiers plats.

Il faut signaler que cette hétérogénéité se situe à deux niveaux distincts. Le premier est sémantique et consiste à définir le même concept mais avec des significations différentes. Et le deuxième niveau est structurel et concerne la représentation des mêmes concepts avec la même signification mais avec des présentations différentes. Par exemple, le concept de client peut être vu comme abonné, touriste ou patient suivant son contexte d'utilisation. Néanmoins, dans une optique de fusion de plusieurs entreprises manipulant ce même concept, il faut trouver une correspondance adéquate pour surmonter ce problème de diversité sémantique. Dans le même sens, chaque entreprise manipulant ce concept peut le traiter de manière différente, en spécifiant un format adéquat, par exemple (*20 caractère alphabétique, ou 30 caractère alphanumérique*).

D'autre part, comme ces données se trouvent dans le Cloud, donc issues d'environnement temps réel, alors les systèmes d'informations des entreprises modernes ne peuvent pas attendre des heures ou des jours pour que les applications gèrent les lots de données en question. Cependant, elles doivent répondre aux nouvelles données en temps réel au fur et à mesure que ces données sont produites. En effet, les organisations contemporaines génèrent et traitent des données **sous forme de flux continus en temps réel** qui sont de nature éphémère ayant des formats non structurés et des volumes très importants et qui proviennent souvent d'utilisateurs nomades.

De ce qui précède, nous pouvons affirmer que les outils ETL conventionnels demeurent limités pour le traitement des **données hétérogènes** et **en temps-réel** et qu'ils souffrent de certaines limitations fonctionnelles dues à la montée en charge du flux de données. Cela est dû fondamentalement au fait que les volumes de données exponentiellement importants brisent les pipelines ETL au niveau des passerelles. Par ailleurs, plus il faut du temps et des ressources pour transformer ces données, plus la file d'attente des données sources est sauvegardée et les données deviennent obsolètes. De plus, les outils ETL sont incapables de gérer, instantanément, les données hétérogènes et importantes qui pourraient générer de meilleures informations à valeur ajoutées (*informations commerciales, par exemple*).

Après l'exposé de la problématique, la suite du chapitre est consacrée à l'étude et à l'analyse des travaux ayant abordé cette problématique de différents points de vue. Nous commençons par présenter quelques variantes améliorées d'outils ETL qui ont tenté de faire face aux limites des outils ETL standards.

3.4 Les variantes améliorés des outils ETL

Les nouvelles variantes d'outils ETL, présentées dans cette section, ont été proposées par les industriels du logiciel pour surmonter les problèmes induits par l'augmentation du volume de données.

3.4.1 Extract, Load and Transform (ELT)

La première version améliorée est l'outil ELT (**E**xtract, **L**oad and **T**ransform. De toute évidence, les outils ELT sont une évolution de l'approche ETL classique. Il existe une différence essentielle entre les outils ETL et ELT. En effet, l'ETL transforme les données avant de les charger dans les systèmes cibles, tandis que l'ELT transforme les données directement dans ces systèmes. Cette distinction sous-tend de nombreux processus en aval et elle est pertinente pour les étapes qui suivent. Un outil ELT consiste, tout simplement, à extraire les données brutes à partir des différentes sources, puis de les charger directement dans une source de données cible (*généralement un entrepôt de données ou un lac de données*) avant même leur transformation. Contrairement à l'approche ELT classique qui implique la transformation des données dans le système cible avant leur chargement, ELT réduit, par son mécanisme d'anticipation, le besoin d'infrastructure physique et de niveaux intermédiaires.

La figure 3.1, illustre la différence entre les outils ETL et ELT.

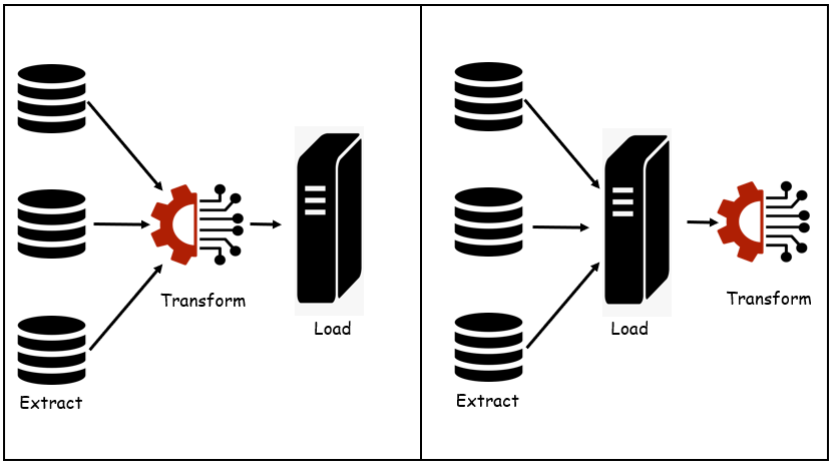

FIGURE 3.1 – Séquencement des étapes ETL vs ELT

Tandis que les outils ETL sont responsables de l'exécution du processus de préparation des données, au cours duquel les données sont nettoyées et prêtes à être transformées, pour ELT la préparation des données se produit après le chargement des données dans un entrepôt de données, un lac de données ou un emplacement de stockage de données dans le cloud. Cette façon de faire vise à augmenter l'efficacité et réduire la latence. Par conséquent, les meilleurs outils ELT mettent moins de pression sur la source de données initiale et éliminent complètement les étapes intermédiaires de l'ETL, puisque la majeure partie du traitement des données se produit dans le système cible.

Dans la pratique, il a été observé que lorsqu'il s'agit de traiter des données à grande échelle, les outils ELT sont nettement plus performants que les outils ETL et puisque ces mécanismes utilisent leurs propres serveurs et moteurs pour transformer les données tous ces pétabits de données provoquent facilement des engorgements avec les outils ETL. De plus, ces engorgements ETL sont susceptibles de prolonger considérablement la latence pour accéder et à analyser des données dans les entrepôts de données. Ce qui signifie que la complexité de cette transformation est encore accrue par le mélange de données semi-structurées et non structurées qui peuplent systématiquement les sources de données massives [41].

3.4.2 Streaming ETL (S-ETL)

Le streaming ETL est le traitement et le déplacement en temps réel des données d'un emplacement physique à un autre de manière continue. Dans ce mécanisme, l'ensemble du processus est effectué pour les données de streaming en temps réel dans une plate-forme de traitement de flux. Ce type d'ETL s'avère très important compte tenu de la vitesse à laquelle les nouvelles technologies génèrent d'énormes quantités de données à un rythme sans précédent. Nous décrivons ci-dessous le fonctionnement d'une telle architecture.

L'architecture de streaming en temps réel et l'architecture ETL traditionnelle sont fondamentalement de même nature, et comme le processus ETL se compose principalement des sources de données, d'un moteur ETL et d'une destination, dans l'architecture de streaming en temps réel les données proviennent des sources de données, elles sont ensuite utilisées comme entrée des outils ETL afin de traiter et transformer les données.

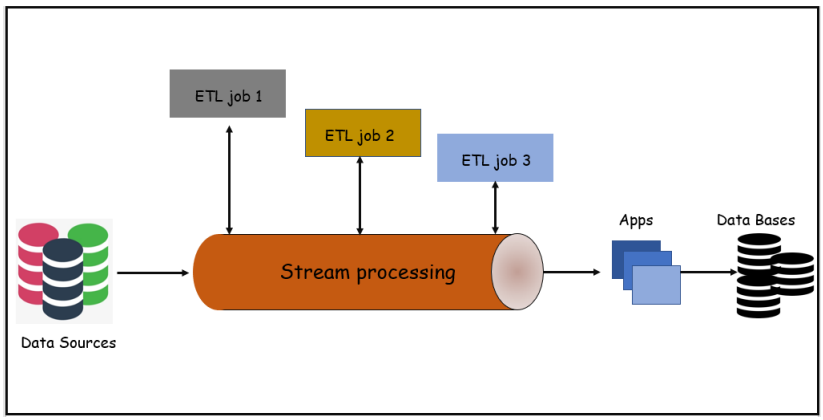

FIGURE 3.2 – Mécanisme de fonctionnement d'un outil Streaming-ETL

Comme il est observé dans la figure 3.2, les données transformées sont transférées de manière progressive vers un entrepôt de données situé au centre de la partie données du système. Par la suite, toutes les données sont transmises de l'entrepôt de données aux applications et aux requêtes. Les sources de données alimentent en données les plateformes de streaming qui agissent comme l'épine dorsale des applications ETL de streaming. Les applications ETL peuvent extraire des flux de données à partir des sources, ou les sources de données peuvent pousser ou publier des données vers des outils ETL en vue de leur

transformation. Ensuite, après avoir traité les données, elles seront transférées vers la destination (*entrepôt de données*). Dans le contexte des données temps-réel, les événements produits et enregistrées dans les sources surgissent un par un en temps réel. Comme, ces données fournissent toujours de nouvelles données dès qu'elles sont disponibles, alors la latence des données exige une exploitation immédiate après enregistrement. Cela aide ensuite à réduire les coûts, puisque l'administrateur des données n'aura pas besoin d'exécuter les opérations sur un petit serveur. Donc en résumé, les outils S-ELT, servent à répondre aux nouvelles données en temps réel au fur et à mesure que ces données sont générées [42].

La figure 3.2, illustre la plate-forme Stream Process qui sert de colonne vertébrale aux applications streaming ETL, où l'application ETL de diffusion en continu peut extraire des données de la source, sachant que la source peut publier des données directement dans l'application ETL. Lorsqu'un processus streaming ETL se termine, il peut transmettre des données vers la droite à une destination (*potentiellement un entrepôt de données*). Ou bien il peut renvoyer un résultat à la source d'origine sur la gauche. En outre, il peut fournir simultanément des données à d'autres applications et référentiels.

3.4.3 Pipe line de données : Data Pipe line

Par analogie au pipeline servant au transport à grande distance et en grande quantité de fluides (*pétrole, gaz naturel, ...*), un pipeline de données est un enchaînement d'étapes qui vise à traiter les données et les transférer d'un système à un autre dans un ordre spécifique. Son mécanisme de fonctionnement est basé sur l'extraction des données de la source en entrée comme une première phase, puis la génération de la sortie de chaque phase qui sert d'entrée pour l'étape suivante. Ce processus se poursuit jusqu'à ce que le pipeline soit complètement exécuté. En outre, certaines étapes indépendantes peuvent également s'exécuter en parallèle dans certains cas.

la figure 3.3 suivante schématise les composants et le fonctionnement d'un pipeline de données.

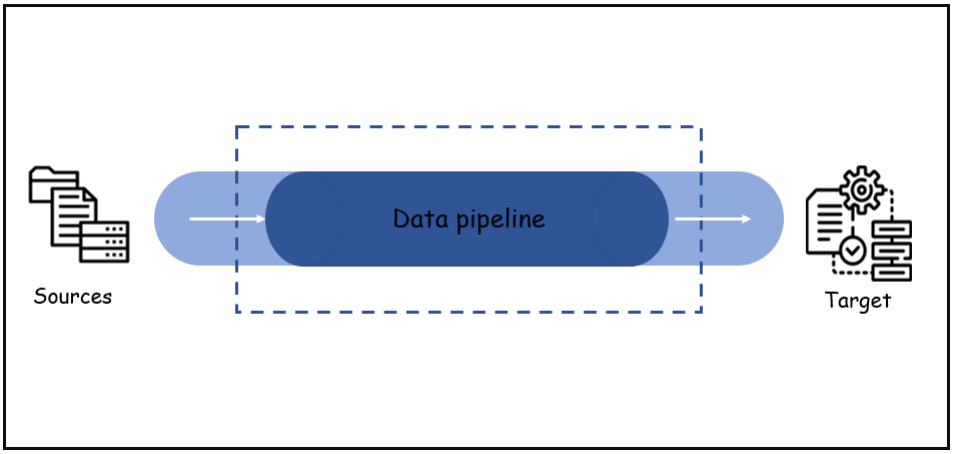

FIGURE 3.3 – Les composants d'une data pipeline

Comme il est observé dans la figure, le fonctionnement du pipe line de données se

compose généralement de trois éléments principaux :

- Une source de données.
- Une ou plusieurs étapes de traitement.
- Une destination finale ou un puits.

L'avantage principale des pipelines et qu'ils permettent aux utilisateurs de transférer des données d'une source vers une destination et d'y apporter les modifications nécéssaires pendant le processus de transfert. Ainsi, ils sont largement utilisés pour intégrer des données qui sont utilisées pour transformer efficacement des données brutes générées en continu et à grands volumes [43].

3.4.4 Différences entre Pipe line de données et pipe line ETL

Un pipeline ETL peut être vu comme une portion (un sous ensemble) du pipeline de données qui permet d'extraire, transformer et charger des données. Cependant, la principale différence entre pipeline de données et pipeline ETL réside dans le fait que ce dernier n'utilise qu'un seul système, afin d'extraire, de transformer et charger les données.

Le temps de chargement des données est plus long avec un ETL qu'un pipeline de données, car ce dernier peut être exécuté en tant réel alors que l'ETL, n'utilisant qu'un seul système, ne peut exécuter les commandes qu'en heures. C'est pour cela que les systèmes ETL fonctionnent souvent par lot au sein d'un pipeline de données, permettant ainsi de réduire le temps d'exécution des commandes.

De plus, un pipeline ETL ne peut charger les données que vers un entrepôt de données spécifique, alors qu'un pipeline de données lui, peut charger les données vers des cibles sélectives et spécifiques, par exemple, un pipeline de données peut charger les données vers le compartiment S3 (Simple Storage Service) d'Amazon, ou connecter les données à un système informatique ne faisant pas partie d'Amazon, ce qui n'est pas le cas avec un pipeline ETL.

Après l'exposé des outils ETL avancés, nous explorons ci-après les travaux de recherche connexes qui ont traité l'intégration des données sur la base des outils ETL.

3.5 Travaux connexes sur l'intégration ETL

La question de l'intégration des données massives a été largement abordée et traitée dans la littérature de recherche. Néanmoins, la façon de percevoir la problème diffère selon les aspects relatifs aux technologie utilisées et conformément aux types de données manipulées. Malgré la diversité des travaux réalisés dans ce domaine, le problème reste toujours d'actualité et les propositions de solutions demeurent insuffisantes par rapport aux avancées considérables, que ce soit sur le plan architectural ou bien d'un point de vue performances.

Dans ce qui suit, nous allons examiné les travaux qui ont traité, de façon plus ou moins approfondie la question de l'intégration des données dans un contexte décisionnel. Pour chaque approche analysée, nous faisons ressortir ses avantages et ses inconvenants. En fin de chapitre, nous dressons un tableau synthétique et comparatif des différents travaux

réalisés.
- Dans [44], les auteurs proposent une approches basée sur les ontologies décrites par le langage OWL-DL pour faciliter la conception des processus ETL. Ils utilisent une représentation sous forme de graphe comme modèle conceptuel pour les entrepôts de données de façon que les données structurées et semi structurés soient prises en charge et traitées de manière uniforme. Cette approche résout le problème de l'hétérogénéité sémantique par l'utilisation des technologies du web sémantique pour annoter de la même façon les sources de données et l'entrepôt de données. L'objectif principale de ce travail est la résolution d'entités et permettre de trouver la correspondance entre les différents attributs. Pour réaliser cet objectif, cette approche propose des opérations conceptuelles pour l'intégration des données dans l'entrepôt de données.

Comme les flux de données ETL (Extract-Transform-Load) alimentent périodiquement les entrepôts de données en informations provenant de différents systèmes sources, cela impose un traitement rapide d'énormes volumes de données. Malgré que MapReduce [????] soit imposé comme la norme de facto pour le traitement intensif des données à grande échelle, il ne prend pas en charge les constructions spécifiques ETL de haut niveau, ce qui entraîne une faible productivité des programmeurs ETL dans les environnements parallèles et distribués. Pour apporter une solution à ce manquement de la fonctionnalité de MapReduce pour les outils ETL, plusieurs travaux de recherche ont tenté de proposer différentes contribution afin de surmonter cette limitation.

Dans ce qui suit, les travaux basés sur l'amélioration des fonctionnalités de MapReduce sont examinés.
- Les auteurs dans [45] présentent un cadre ETL dimensionnel évolutif, dénommé **ETMLR (Extract-Transform-Map-Load-Reduce)**, pour les environnements distribués. Ce système est basé sur le support natif MapReduce auquel ils ont intégré les opérations sur les constructions spécifiques à l'ETL, telles que les schémas en étoile, les schémas en flocon de neige ainsi que les dimensions à évolution lente (**Slowly Changing Dimension** : SCD). Cela permet aux développeurs ETL de construire des flux ETL évolutifs basés sur MapReduce avec très peu de lignes de code. cette approche parallèle/distribuée permet d'améliorer les performances de la phase de transformation (**T**) et de chargement (**L**) des outils ETL, et ce en adoptant pour chacune des deux phases des stratégies de distribution des données qui sont appropriées à la technologie MapReduce. L'outil développé, **ETLMR** comprend deux phases de traitement des données qui sont :

- Le traitement des tables de dimensions.
- Le traitement des tables des faits.

- Contrairement aux travaux précédents, les auteurs dans [19], suggèrent de se focaliser sur le couple Extract-Transform au lieu du couple Transform-Load. Le champ d'application de ce travail est limité à la partie extraction et transformation (**ET**) du processus ETL. Les auteurs présentent une comparaison des coûts et des performances entre les solutions commerciales ETL et les solutions open source basées sur MapReduce (**M/R**). Une double approche d'expérimentation et d'évaluation des performances a été conduite et les résultats obtenus ont été argumentés et discutés. Cette approche permet d'évaluer l'applicabilité des options de solutions commerciales en fonction de la vitesse de traitement, le cout et le déploiement des ressources.

- Dans le but d'accélérer le processus d'élaboration d'une solution BI basée sur les données, d'un côté, et préparer également des solutions de veille stratégique pour l'utilisation des mégadonnées, les auteurs dans l'article [46] présentent une nouvelle approche pour la conception de solutions d'**informatique décisionnelle** (BI). Ce travail suggère l'extension de l'approche existante (ELT) à une nouvelle approche dénommée **ELTA** (**Extract, Load, Transform and Analyse**) dont le fonctionnement est décrit comme suit :
Un processus appelé **Extract** permet d'extraire des données des différentes sources hétérogènes dans des formats hétérogènes, ensuite le processus **Load** assure le chargement des données dans un système de stockage (*zone de transit*). Enfin, le processus **Transform** permet de transformer les données brutes à la demande et en fonction des besoins du processus décisionnel.

La nouveauté dans cette approche consiste à offrir aux gestionnaires un processus nommée **Analyse** qui leur permet d'utiliser efficacement les données pré-traitées pour comprendre le comportement de l'entreprise et de pouvoir prendre les décisions adéquates. L'avantage principale de ce travail consiste à combiner les techniques de la BI avec le domaine des données massives (Big Data) en tirant profit des meilleurs acquis des deux domaine et tout en éliminant les inconvénients de la BI d'une façon parallèle.

- Dans le contexte de l'influence de la discipline big data sur les systèmes ETL et les environnements décisionnels, d'une part, et afin de mieux gérer l'intégration de données distribuées d'autre part, dans [47] les auteurs proposent une nouvelle approche du processus ETL pour laquelle ils définissent des fonctionnalités pouvant s'exécuter sur un cluster selon le modèle MapReduce (**MR**). Dans cette perspective, ils proposent un processus **ETL parallèle**, appelé PF-ETL (**Parallel Functionality-ETL**). Ce nouveau processus PF-ETL garanti un ensemble de fonctionnalités selon le paradigme MR, où chacune peut s'exécuter avec plusieurs instances en parallèle. Ainis, plusieurs exécutions simultanées d'une même fonctionnalité sont lancées et permettent de tirer parti des nouveaux environnements informatiques parallèles et distribuées.

- Toujours dans le même contexte du données massives, un autre travail intéressant est suggéré dans [48]. Dans ce travail, les auteurs proposent et décrivent le fonctionnement d'une plateforme nommée **P-ETL** qui est basée sur l'entreposage des données massives selon le modèle MapReduce. L'approche s'articule sur le paramétrage du processus ETL à l'entrée même du système et consiste en un paramétrage avancé de l'environnement parallèle et distribué. Cette approche est articulée autour des 3 étapes suivants :

- Dans un premier temps les données sont partitionnées conformément à différents algorithme (*simple, round robin, round robin par bloc*) ;
- Après ce partitionnement des données, les primitives **Map** sont introduites pour la normalisation des données ;
- Enfin la primitive **Reduce** assure leur fusion.

La contribution majeur de ce travail consiste en une amélioration des performances dans un contexte de passage à l'échelle et pour faire face au flux de données de plus en plus croissant.

- Par ailleurs, dans une perspective d'améliorer les performances du processus ETL et afin de réduire la charge de travail induite par la phase de migration des données qui ne

seront peut-être jamais utilisées, une nouvelle approche ETL, nommée **TEL** (Transform-Extract-Load) est proposée dans [49]. Cette nouvelle approche utilise des tables virtuelles pour réaliser l'étape de transformation avant l'étape d'extraction et l'étape de chargement. Cette façon de faire permet de réduire la charge de transmission des données et améliore considérablement les performances des requêtes à partir des couches d'accès. Dans cette approche, l'étape de transformation constitue une couche virtuelle qui assure le mapping des schémas. Ainsi, des tables virtuelles sont créées par les utilisateurs, comme des structures qui garantissent la cohérence des sources de données, telles que la sélection des champs, le type de champ, la longueur du champ, ...

L'aspect nouveauté de cette approche est que les tables virtuelles assurent l'élimination des données hétérogènes, fournissent l'accès transparent aux données et offrent une vue unifiée créée à travers des bases de données hétérogènes ou homogènes. Sur la base des tables virtuelles, l'étape d'extraction de l'approche **TEL** effectue un travail d'extraction de données à la demande. Elle comprend l'extraction complète, incrémentielle et par requête, en fonction des différents scénarios d'application. L'étape L (load) de TEL suit rapidement l'extraction des données, en les chargeant dans l'interface des requêtes, le cache ou l'entrepôt de données.

- Afin d'assurer une bonne performance des processus ETL et faire faire au phénomène communément appellé "**volume excessif**" de données, le travail exposé dans [50] proposent une approche original appelée **Big-ETL**. Dans cette approche les auteurs définissent des fonctionnalités ETL qui peuvent être exécutées facilement sur un cluster d'ordinateurs avec le paradigme MapReduce. En effet, Big-ETL permet de paralléliser/distribuer l'ETL à deux niveaux :

- Le niveau de processus ETL
- Le niveau de fonctionnalités pour améliorer les performances d'ETL

Pour contrôler la complexité du processus ETL, cette approche **parallèle/distribuée** est articulée autour des fonctionnalités ETL spécifiques suivantes.

- la capture de données changeantes (CDC),
- la validation de la qualité des données (DVQ),
- la gestion de la clé de substitution (Substitution Key : SK),
- gestion des données dont la dimension d'évolution est lente (Slowly Changing Dimension : SCD),
- le pipeline de clés de substitution (SKP)

- Dans la même perspective et pour face aux problèmes liés au big data, dans [51] et [52], les auteurs proposent une approche baptisée **BigDimETL** (Big Dimensional ETL) qui traite du développement ETL et qui se concentre sur l'intégration des données massives issues de différentes sources en tenant compte de la structure multidimensionnelle (Multi-Dimensionnal Structure) à travers le paradigme MapReduce. Cette approche fonctionne avec les trois phases classique du processus ETL, néanmoins elle consiste à adapter la phase d'extraction et de transformation avec le paradigme MapReduce. Ainsi, pour distribuer

les données d'entrée, les auteurs utilisent le partitionnement vertical selon les dimensions de la structure multidimensionnelle décrite dans les méta-données.

-Enfin, pour modéliser le processus ETL à un niveau conceptuel, tout en l'adaptant aux standards du Web par la prise en charge des formalismes associé aussi bien à UML, qu'au langage BPMN et au web sémantique, les travaux de [53, 20] les auteurs proposent une nouvelle approche pour la modélisation conceptuelle du processus ETL en utilisant un nouveau langage standard de modélisation des systèmes, nommé Systems Modeling Language (**SysML**), qui étend les caractéristiques de UML avec une sémantique beaucoup plus claire du point de vue de l'ingénierie des systèmes.

La contribution majeur de ce travail est l'extension des fonctionnalités UML avec une sémantique beaucoup plus claire du point de vue de l'ingénierie système.

3.6 Synthèse des travaux connexes

La table 3.1 ci-dessus récapitule les travaux connexes ayant abordé la problématique de l'intégration des données de différents point de vues. A signaler que la plus part des travaux recensés lors de l'analyse de l'état de l'art concerne en grande partie les travaux associés aux données massives. Cela s'explique en grande partie par le phénomène de l'explosion des données induites par le développement du web et des plateformes distribuées.

Réf.	**Principe**	**Contribution et modèle proposée**	**Evaluation**
[44]	Approche basée sur les ontologies pour facilité la conception d'ETL	OWL-DL	Permet l'identification des ressources pertinentes avec l'amélioration des scénarios ETL dans le monde réel.
[45]	Approche(ETMLR) parallèle/distribué qui s'intéressent aux phases de transformation et de chargement de l'ETL.	TL avec Map Reduce	Évolutivité, Performances.
[19]	Approche basée sur Hadoop qui s'intéressent aux phases d'Extraction et transformation de l'ETL.	ET avec Hadoop	Améliorer considérablement le débit, la réduction des coûts et les effectifs.
[46]	Approche ELTA pour la conception de solution de BI	ETL-Analyse	Ils abordent la combinaison BI et big data en prenant les meilleurs éléments à la fois et en parallèle en éliminant les désavantages de BI.
[47]	Approche du processus ETL avec définition des fonctionnalités qui peuvent être exécutées en cluster selon le modèle(MR)	ETL avec Map Reduce	permet une migration vers un environnement cloud.
[48]	Platforme P-ETL parallèle/distribué destinée à l'entreposage de données massives	ETL avec Map Reduce	montre une meilleure évolutivité de P-ETL
[49]	Approche TEL utilise des tables virtuelles pour réaliser l'étape de transformation avant l'étape d'extraction et l'étape de chargement	ETL avec CCEVP	Cette technique réduit la charge de travail associée à la migration des données
[50]	Approche Big-ETL parallèle/distribue avec la définition de nombreuses fonctionnalités ETL	ETL avec MapReduce	permet de contrôler la complexité du processus ETL.
[51][52]	Approche BigDimETL qui traite du développement ETL	ET avec MapReduce	l'efficacité d'ajouter facilement d'autres opérations ETL
[53][20]	Approche pour la modélisation conceptuelle du processus ETL	ETL avec SysML	le modèle de système peut être conçu d'une façon plus expressive et plus souple.

TABLE 3.1 – **Tableau d'évaluation des travaux existants**

3.7 Conclusion

Après avoir énumérer les limites des outils ETL classiques, dans ce chapitre nous avons exposé notre problématique, puis nous avons explorer les outils logiciels ETL ayant subi des améliorations afin de faire face aux données massives et en temps-réel. Après cela, nous avons mené une analyse approfondie des travaux connexes existant dans la littérature et qui ont traité le problème de l'intégration des données massives.

Pour surmonter les insuffisances liées au fonctionnement des outils ETL, tels explicitées dans ce chapitre, et afin de prendre en compte la dimension temporelle des données issues des sources diverses, il est impératif de proposer une amélioration du principe de fonctionnement des outils ETL.

Dans le prochain chapitre, nous allons apporter notre contribution et concevoir une solution adéquate permettant de surmonter les limites des outils ETL.

Deuxième partie

Contribution et Implémentation de l'approche

Chapitre 4

Conception de l'approche

4.1 Introduction

Dans le chapitre précédent nous avons exposé notre problématique et nous l'avons positionnée par rapport aux travaux connexes.

Pour surmonter les limites des outils ETL classiques, tout en prenant en compte l'aspect flux des données émanant du Web, nous proposons dans ce chapitre notre contribution qui consiste en une nouvelle approche d'enrichissement des outils ETL pour l'intégration du données.

Nous commençons le chapitre par la présentation du principe général de fonctionnement de la solution proposée, puis on discutera l'architecture du système proposé. Après cela, la description du fonctionnement de la solution est abordée en détails, et enfin nous terminerons le chapitre par l'exposé d'un scénario qui illustre la faisabilité de notre approche par l'examen d'un scénario réel.

4.2 Principe de la solution proposée

Notre approche se base fondamentalement sur la prise en compte des flux de données distribuées et leur **changement avec le temps**. En effet, bien que les données se trouvant au niveau des différentes sources soient rapatriées et intégrées dans l'entrepôt de données, il s'avère que leur variation au niveau des sources soit **instantanée**. Cette variation des données est due aux changements opérés lors des différentes actions de mise à jours. La conséquence immédiate est que les données contenues dans l'entrepôt ne reflètent plus la réalité et, par conséquent, deviennent **incompatibles**.

Notre contribution pour remédier à cette limitation se situe au niveau des étapes d'extraction et de transformation des données aux quelles nous apporterons certaines améliorations afin de prendre en compte les données **versatiles** du Web et aussi pour améliorer les performances des systèmes ETL.

Dans cette perspective et pour permettre un **rafraichissement continu** des informations contenues dans l'EDD, nous considérons une nouvelle perception de la phase d'extraction qui constituera le noyau de notre contribution.

Le fondement de notre approche vise à améliorer les performances de la phase d'extraction de ces outils. Cet objectif est réalise sur la base de deux aspects complémentaires. Le premier consiste à adopter des stratégies pour la phase d'extraction visant à travailler avec des données versatiles et changeantes au fil du temps à partir du Web. Le deuxième est relatif à la spécification des règles de transformation à appliquer.

Nous décrivons, dans ce qui suit les principes fondateurs de notre solution baptisée : **O**n**L**ine **E**xtract **S**elective **T**ransform and **L**oad (**OLE-STL**).

1. Une extraction initiale est opérée à partir des différentes sources ;

2. Au lieu d'appliquer l'ensemble des règles de transformations, telles que décrites dans la section **2.6.4 (b)**, dans notre approche nous classons ces règles en quatre catégories distinctes qui sont les suivants :

   - format ;
   - granularité ;
   - codage ;
   - pré-calcul.

   L'utilisateur aura la possibilité d'**appliquer de manière sélective** une ou plusieurs règles lors du processus de transformation au lieu de les déclenches en totalité de manière séquentielle.

3. Une fois, les règles de transformations choisies, les processus Transform et Load sont ensuite lancés afin d'alimenter l'EDD par les données converties.

4. Si la structure de l'EDD a évolué et exige l'intégration de nouvelle sources, une **extraction additive** de la (les) source (s) en question est faite. Évidemment la transformation et le chargement de ces sources seront réalisés en conséquence ;

5. Le point le plus intéressant dans notre approche consiste en **un mécanisme de test d'existence de modifications** au niveau des sources est activé pour identifier les sources qui ont été changées après le dernier chargement. Ce mécanisme repose sur la vérification et la comparaison des sources de données qui s'appuie sur les fichiers log ;

6. Enfin, la phase d'enrichissement de l'EDD par un chargement (Load) de ces nouvelles données transformées est effectué.

4.3 Contribution majeures de la solution

La solution proposée est un système ETL amélioré nommée $OLE - STL$ dont son mécanisme considère principalement la phase d'extraction (**E**) pour laquelle une transformation sélective est suggérée. Ainsi, notre contribution se situe à trois niveaux :

- Une technique de transformation sélective qui considère uniquement la (les) catégorie (s) de transformations choisies par la gestionnaire de données.
- Au niveau de l'étape Extract une option d'extraction additive est proposée en cas d'évolution du schéma de l'EDD.
- Un technique pour le test d'existence de mise à jour au niveau des sources, basée sur la comparaison des dates de mise à jour des fichiers sources avec la dernière date de chargement. Cette technique utilise des structures de données de type table pour suivre la traçabilité des modifications des sources.

Après avoir expliqué le principe général de notre solution, la prochaine section est dédiée à l'exposé de l'architecture de notre système (**OLE-STL**).

4.4 Architecture générale du système proposé

Pour faire face au problème de la diversité des données ,de leur vitesse d'évolution ainsi que leur volume qui est de plus en plus consistant, et comme nous avons expliqué dans notre problématique, les outils ETL restent limités pour le traitement de ces données qui se caractérisent par leur spontanéité c'est-à-dire les données en temps réel.

Notre système (**OLE-STL**) propose une solution axée essentiellement sur les deux premiers phases Extract et transform de l'ETL dans laquelle on a divisé le système en trois passe qui sont :

- **Passe 1** : Elle se compose de trois étapes et commence par l'étape d'extraction initiale à partir de diverses sources suivie d'une transformation sélective qui permet aux utilisateurs de sélectionner une ou plusieurs règles parmi un ensemble de règles de transformation au cours de ce processus, et enfin il se termine par un chargement (Load) pour alimenter l'EDD. Cette passe est illustrée par la séquence **1-2-3** de la figure 4.1.

- **Passe 2** : Elle est organisé en trois phases qui sont entamées avec l'extraction additive appliquée dans le cas ou la structure de l'EDD a évolué et nécessite l'intégration de nouvelles sources, puis les étapes de transformation et de chargement sont déclenchées pour enrichir l'EDD. Cette passe est illustrée par la séquence **1'-2-3** de la figure 4.1.

- **Passe 3** : cette passe est différente des deux premières passes. Ainsi au cours de la phase d'extraction ou opère une extraction incrémentale qui permet d'identifier les sources qui ont été changées aprés le dernier chargement. Il s'agit d'une extraction fondée sur la vérification et la comparaison des sources de données, et par la suite le lancement des processus de transformation sélective et de chargement pour actualiser l'entrepôt. La comparaison est basée sur les tests de dates de mise à jour. Cette passe est illustrée par la séquence **1"-2-3** de la figure 4.1

En effet la figure 4.1, explique bien l'architecture de notre système (**OLE-STL**) et l'enchainement des trois passe ci dessus citées.

4.5 Description du fonctionnement de la solution

Le système (**OLE-STL**) est articulé autour de trois modules principaux (Extract, Transform and Load) dont le fonctionnement est décrit ci-après :

1. **Phase 1** :
   Ce module comporte trois types d'extraction selon les besoin de l'utilisateur qui peut choisir une des options suivants :

   - **Une extraction initiale** : utilisée lorsque les données sont initialement chargées dans l'entrepôt ou lorsque les données sont entièrement actualisées (dans le cas d'un changement de nouvelle source, par exemple). L'extraction complète permet de capturer toutes les données à un certain moment (snapshot d'état opérationnel).

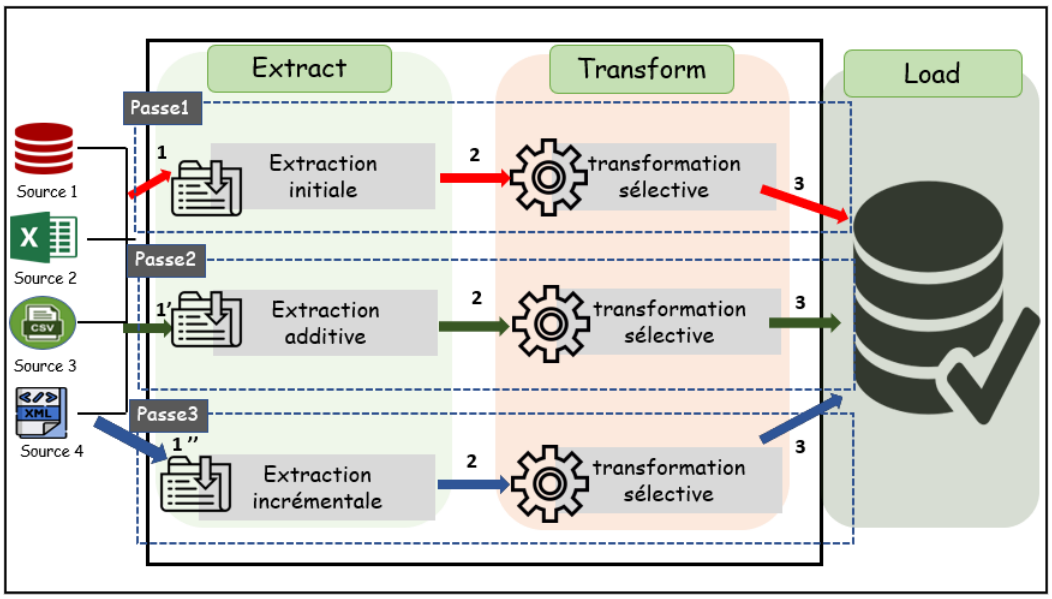

FIGURE 4.1 – Architecture de notre système(**OLE-STL**)

- **Une extraction additive** : utilisée lorsqu'il y a une évolution au niveau de la structure de l'EDD, ceci se manifeste par, soit les créations d'une nouvelle table ou bien l'ajout d'attribut. Par exemple, dans le domaine commerciale l'ajout de la table des types paiement ou l'ajout de l'attribut code à barre de la table produit. Alors il est nécessaire d'incorporer de nouvelles sources de données.
- **Une extraction incrémentale** : Cette extraction ne prend en charge que les données qui ont été modifiées ou ajoutées depuis la dernière extraction en temps réels. C'est-à-dire, quand les transactions se produisent dans les systèmes sources, en analysant tous les changements apportés sur une certaine période au moyen de tests de comparaison des tables sources pour différentes périodes (heures, jours, mois...). Cette extraction repose sur le test des dates entre les sources après le dernier chargement et les sources courantes après la nouvelle extraction. Ainsi après chaque chargement le système (**OLE-STL**) garde la date (date du dernier chargement) et chaque table de l'EDD subir un table date pour stocker l'historique de toutes les dates du chargement.

  Dans le cas ou il y a une modification dans une table, le système (**OLE-STL**) teste la date du chargement par rapport la dernière date du chargement qui est stockée dans la table date qui contient toutes les dates des chargements.

2. **Phase 2** :
   Dans notre système on opte pour une transformation sélective qui permet aux utilisateurs d'avoir la possibilité d'appliquer de manière sélective une ou plusieurs règles de transformation lors de ce processus. En effet le système (**OLE-STL**) offre une transformation organisée en quatre catégories distinctes de transformation. La table 2.1, ci-dessous met en exergue les différents types de transformations offerts par le système (**OLE-STL**).

3. **Phase 3** :
   Une fois que les données sont extraites et transformées en formats appropriés, la

Trans.	**Type de transformation**	**Description**	**Exemple**
Format	Redressement de format	Changer le type ou la longueur d'un attribut	Adresse-Magasin sur 30 caractères α-numériques au lieu de 40 α-bétiques.
	Conversion de jeu de caractère	Unifier les divers jeu de caractère	EBCDIC (IBM) vers ACSII UNICODE vers UTF8
	Conversion des dates	Harmoniser les formats des dates	31 MRS 1997 vers 31/03/1997 31/03/1997 vers 03/31/1997
	Conversion des unités de mesures	Utiliser les mêmes unités de mesures	Client au magasin,client en ligne, client potentiel.
Granularité	Fusion de plusieurs champs	Regrouper les informations d'une même entité	Source 1 : Code et libellé produit. Source 2 : Type de paiement. Source 3 :Quantité livrée.
	Découpage de champs complexes	Extraire des informations atomiques à partir d'autres articulés	Nom, Prénom, Conjoint à partir d'une chaine de caractère Nomcompletépouse.
Codage	Unification du codage de champs	Consolider les données de sources multiples	[Homme,Femme], [H,F], [1,2].
	Transcription des valeurs en codes	Faire corespondre des codes à des valeurs	G :Gros client, M : moyen et F :Faible
Pré-calcul	Pré-calcul des valeurs dérivées	Appliquer le règles de calcul	Profit=Prix vente- coûts. Prix TTC= Prix HT + TVA
	Pré-calcul des agrégat	Calculer les sommes, produit, moyennes.	Total des ventes par semaine et par mois de chaque produit.

TABLE 4.1 – **Les types de transformations assurées par OLE-STL**

dernière étape de notre système consiste à les charger dans leur nouveau lieu qui est l'entrepôt de données.

L'interaction entre les trois modules du système est détaillée ci-dessous.

Scénario ***1⇒2⇒3*** : exprime le scénario d'extraction initial pour alimenter de l'entrepôt de données par des nouvelles données émanant des différentes sources.

Scénario***1'⇒2⇒3*** : c'est le scénario d'extraction additive dans lequel les sources de données a intégrer sont de nouvelles sources ou dans le cas de l'évolution des structures des données par rapport les données de l'EDD.

Scénario***1"⇒2⇒3*** :exprime le scénario d'extraction incrémentale dans lequel les sources de données à intégrer sont des données déjà existantes dans EDD mais dont les contenue ont subi des modifications en terme d'instances (occurrences).

Dans ce qui suit, on va revenir sur chacune des trois phases avec plus d'explications.

La première phase du système (**OLE-STL**) peut avoir une des catégorie suivantes : extraction initiale, extraction additive et extraction incrémentale, en fonction de la nature des données en entrée, elle permet de déterminer si les données disponibles sont de nouvelles données par rapport l'EDD ou des données qui ont changé récemment (ajout , suppression, modification...). Ces catégorie sont schématisées par les symboles **(1)**, **(1')** et **(1")** de la figure 4.1 montrent cette première phase où :

(1) : représente l'état ou la source de données est nouvelle par rapport l'EDD c'est à dire n'existe pas dans l'entrepôt.

(1') : exprime le cas des changements dans la structure de l'EDD dans lesquels il existe un besoin d'incorporer de nouvelles sources de données pour faire évoluer l'entrepot de manière à répondre aux nouvelles exigences des utilisateurs.

(1") : indique que cette source existe déjà dans l'EDD mais qu'elle a été modifiée depuis la dernière extraction.

(2) : Lorsque un type parmi les types d'extraction (*initiale,additive,incrémentale*) est terminée, le système (**OLE-STL**) amorce la phase de transformation sélective proposée.

(3) : exprime que la phase de transformation est réalisée pour lancer l'étape de chargement des données intégrées dans l'entrepôt de données.

4.6 Scénario illustratif de fonctionnement de OLE-STL

Pour montrer la faisabilité et l'utilité de notre approche, considérons l'exemple de la gestion des commandes client d'une entreprise commerciale. Le processus métier de la gestion des commandes comprend toutes les étapes à suivre à partir du moment où un client passe une commande jusqu'à la réception de la commande, c'est-à-dire la livraison du produit ou la réalisation du service.

Le Modèle Conceptuel des données (MCD) est représenté dans la figure 4.2 ci-dessous.

4.6.1 Modélisation des données de l'EDD

4.6.2 Description des règles métier du PM commande client

Le processus commence par la création d'une commande client standard. Selon le client et l'article concernés, différents événements se produisent lors de la saisie, tels que la dé-

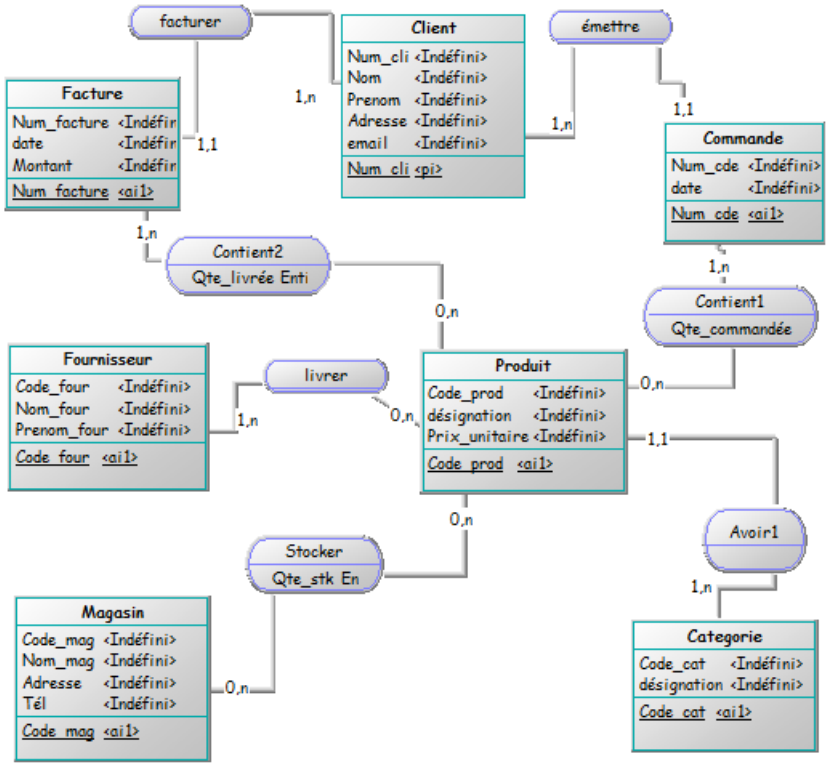

FIGURE 4.2 – Gestion des commandes client d'une entreprise commerciales

termination des prix client/article, l'insertion des remises applicables, la vérification de la disponibilité de l'article et la vérification de l'historique des crédits client. Le processus vérifie s'il y a suffisamment d'articles dans le magasin souhaité. Si ce n'est pas le cas, des mouvements de stocks se produiront. Des listes de sélection sont ensuite générées pour permettre au personnel de l'entrepôt d'expédier les produits aux clients. Après l'enlèvement, la quantité réellement expédiée doit être enregistrée dans le système pour éviter tout écart entre la commande client et le bon de livraison. S'il existe de réelles divergences, vous pouvez les enregistrer pour une publication appropriée. Une fois la cueillette terminée, le commis doit systématiquement réduire le stock. Cette réduction est affectée par la quantité réelle qui est réellement enregistrée pour être expédiée au client. Lorsque les stocks sont réduits, des factures de livraison peuvent être émises. Les prix des produits et de revient sont ensuite enregistrés dans des comptes internes. Cette étape marque la fin de l'opération commerciale dans la composante Gestion des ventes. La dernière section de ce document décrit le processus d'affichage des factures, de préparation et d'impression des relevés des comptes clients, et d'enregistrement et d'équilibrage des reçus dans le but de rapprocher les comptes clients. Envoyez la sortie de facturation avec le fichier de paiement préimprimé au client.

Dans ce qui suit on va appliquer le mécanisme de fonctionnement de notre système (**OLE-STL**) sur ce processus de la gestion des commandes clients, en suivant les trois phases.

4.6.3 Phase d'extraction

Pour alimenter l'EDD de la gestion des commandes clients le processus de notre système commence par la première phase qui est l'extraction initiale à partir les différents sources de données. Ces données peuvent être hétérogènes et de différentes formats, par exemple (XML, CSV ,Excel et BDDR).

a) Exemple d'extraction initiale

Dans cette exemple on illustre une extraction initial des données de PM gestion de commande.

Exemple 4.1

Num	Source	Nom table	Format
1	1	client	Excel
2	2	commande	Excel
3	3	produit	Excel
4	4	contient 1	BDDR
5	5	facture	BDDR
6	6	contient 2	BDDR
7	7	fournisseur	XML
8	8	livrer	XML
9	9	magasin	CSV
10	10	stocker	CSV

Admettons que , par contre, notre EDD est une schéma relationnel, dans ce cas l'extraction doit assurer la migration des différents formats vers le BDDR. Cela passe impérativement, par la zone de transit (staging area) (Disque dur local).

b) Exemple d'extraction additive

Comme on a expliqué au parvant, une extraction additive permet, tout simplement, d'ajouter des nouvelles tables ou des attributs additionnels à l'EDD déjà alimenté.

Nous montrons ce mécanisme à travers l'intégration des données stockées dans le site web marchand tenu par un commerçant qui opère des ventes en lignes.

Exemple 4.2 *L'objectif du commerçant sur internet est bien évidemment d'aboutir, pour chaque visite d'un client sur son site, à une transaction d'achat.*

Ainsi, le site web donne la possibilité au client d'opérer des commandes par la sélection récursive d'article, puis il confirme sa commande et enfin, il procède au règlement de sa commande.

La nouveauté est la proposition par le commerçant de différentes modes de paiement. Ainsi, les données relatifs au mode de paiement sont instanciés par les occurrences de la table type-paiement. Concrètement, il y aura une migration de la clé code-type-paiement vers la table client. Ce fait doit apparaitre dans la table client comme clé étrangère

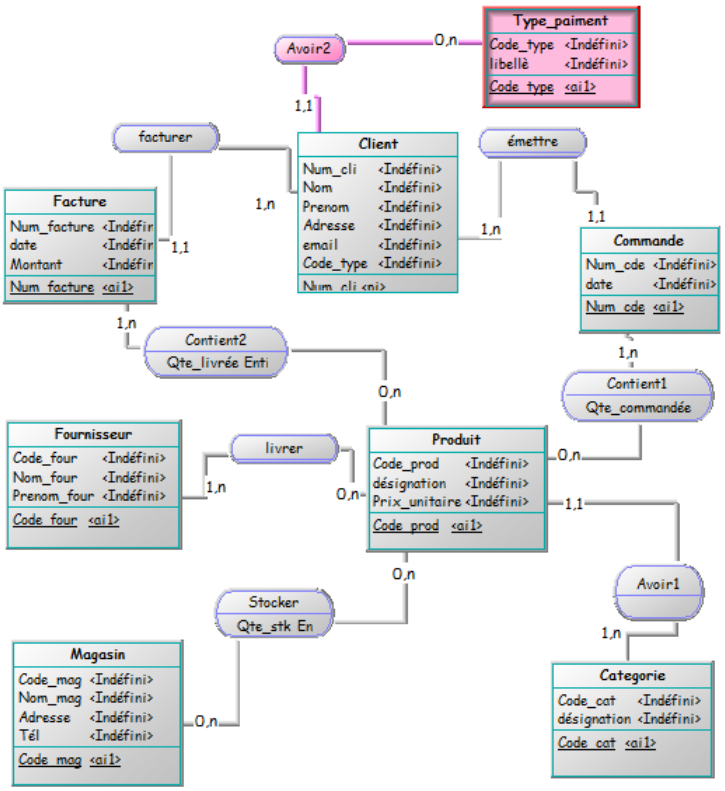

FIGURE 4.3 – Gestion des commandes client d'une entreprise commerciales

La figure 4.3, illustre un exemple d'une extraction additive qui exprime une évolution de l'EDD avec la création d'une nouvelle source qui contient les types de paiements utilisés par chaque client. L'impact de cette évolution consiste à :
i) ajouter la table type-paiement.
ii) ajouter pour chaque client son code-type-paiement

b) Exemple d'extraction incrémentale

Dans le cas où un client passe une commande qui contient certains produits, mais avant la livraison, il décide de changer le contenue de sa commande (modification des lignes de commande). Dans ce cas, la table Contient sera modifiée en conséquence et alors la date système de cette table sera changée.

Pour l'extraction incrémentale, notre système va procéder en 2 phases :
1- Il compare les dates de chaque table contenu dans l'EDD avec celle de la source. Il trouve que pour la table Contient : (Date système > Date EDD). Donc, la table a été modifiée.
2- On agit au niveau enregistrement :
C'est-à-dire, il faut localiser, les enregistrements qui ont subi des modification. Cela est possible par le test de l'attribut date-commande de la table Commande.

4.6.4 Phase de transformation sélective

a) transformation de type Format

a1. Le changement du type ou la longueur des attributs.

Exemple 4.3 *Adresse Magasin sur 30 caractères alphabétique au lieu de 40 alphanumérique.*
Code produit Entier sur 8 position au lieux de 4 position.

a2. Conversion des dates.

Exemple 4.4 *Conversion de la date d'une commande comme suit :*
24AVR2022 vers 24/04/2022
ou
04/24/2022 vers 24/04/2022

b) transformation de type granularité

b1. le regroupement des informations de la même entité qu'elles se trouvent dans une plusieurs source de données à l'aide d'un type de transformation basée sur la fusion de différents champs.

Exemple 4.5 *Source 1 : Code produit et désignation.*
Source 2 : code produit et le prix.

La jointure naturelle de ces deux tables sur la base du code produit permet de regrouper ces données dans la même entité qui est l'entité produit.

b2. L'extraction des informations atomique à partir d'autres articulées sur la base de la délimitation de champs complexes.

Exemple 4.6 *Extraire le prénom et le nom de famille du client d'une chaîne de caractères Nomcomplet afin de remplir les champs Nom-cl et Prénom-cl dans le table du client.*

c) transformation de type Codage

c1. Consolider les données provenant de sources multiples, c'est-à-dire le cas où le système unifie le codage de champs qui se trouve dans de multiples sources et codés de différentes façons.

Exemple 4.7 *Pour obtenir le code d'un type de paiement de chacun des clients, le système constate que ces données sont codées de différentes façons.Dans une source, elle est codifiée par des nombres comme suit :*
Source 1 :
Paiement avec carte bancaire : 1.
Paiement avec chèque : 2.
Paiement avec espèce : 3.
Par contre dans une autre source, elle est codée par des lettres comme suit :
Source 2 :
Paiement avec carte bancaire : CIB.
Paiement avec chèque : CQ.
Paiement avec espèce : ES.

c2. Transcription des valeurs en codes.

Exemple 4.8 *Les clients sont la raison d'exister d'une organisation et comme il est plus économique de garder un client que d'en gagner un nouveau et il est également plus économique de faire affaire avec un client fidèle que de tenter de refaire des affaires avec un ancien client. Dans ce contexte supposons que l'utilisateur a le besoin de connaitre le type de chaque client pour par exemple faire des privilèges pour les clients fidèles et après l'extraction des données il trouve que les types sont des valeurs, alors le système doit faire correspondre des codes à des valeurs afin de les stocker dans l'EDD comme suit :*
P : les clients potentiels
A : les clients actuels ;
F : les clients violons ;
An : les anciens clients.

d) Pré-calcul des agrégats

Permet de calculer au préalable de certains attributs l'EDD. Cette façon de faire permet de gagner en performances du système.

Exemple 4.9 *Montant facture : calcule une seule fois, lors du chargement initial afin d'éviter son recalcule à chaque besoin. Montant=prix-unitaire x Qté-livrée*

Après avoir appliquer toutes les transformations expliquées ci-dessus, le processus se terminera par la dernière étape, qui consiste à charger toutes ces données transformées dans l'EDD.

4.7 Conclusion

Dans ce chapitre nous avons exposé notre contribution qui consiste en une nouvelle approche d'enrichissement des outils ETL pour l'intégration du données nommé **O**n**L**ine **E**xtract **S**elective **T**ransform and **L**oad (**OLE-STL**).

Nous avons commencé le chapitre par la présentation du principe général de fonctionnement de notre solution, puis on a discuté l'architecture du système proposé. Après cela, la description du fonctionnement de la solution est abordée en détails, et enfin nous avons terminé le chapitre par l'exposé d'un scénario réel qui illustre la faisabilité de notre approche.

La conception élaborée ouvre la voie à la mise en œuvre de l'approche proposée.

Le prochain chapitre est dédié à la mise en œuvre du système (OLE-STL).

Chapitre 5

Implémentation et Expérimentation de l'approche

5.1 Introduction

Après avoir terminé l'étape de formalisation de notre approche, dans ce chapitre on va se focaliser sur l'aspect implémentation qui permettra de montrer la faisabilité et l'exploitation concrète de la solution proposée dans un contexte réel.

Le chapitre est structuré comme suit.

Nous commençons par la présentation de l'environnement de travail utilisé pour développer notre application, puis nous exposons les fonctionnalités que l'application permet de réaliser ainsi que l'enchainement général des fonctionnalités offertes. Après cela, nous illustrons le fonctionnement de l'application par son déploiement et son exploitation sur un jeu d'essai relatif aux données du domaine de la gestion commerciale dont le modèle de données a été déjà élaboré et conçu dans le chapitre précédent.

5.2 Présentation de l'environnement de travail

Pour l'implémentation de notre approche et la réalisation de notre prototype dénommé **OLE-STL**, nous avons utilisé un ensemble d'outils qui nous ont permis de réaliser nos objectifs. Ces outils sont les suivants :

- **PyCharm** : c'est un environnement de développement intégré utilisé pour programmer en python. Il permet l'analyse de données et il contient un déboguer graphique. Il offre une productivité élevée et permet de gagner en temps de programmation et aussi de bénéficier d'une assistance intelligente comme la vérification des erreurs à la volée et les correctifs rapides.

- **DB Browser(Sqlite**) : C'est un outil open source visuel de haute qualité pour créer, concevoir et de modifier des fichiers de BDD compatibles avec Sqlite. Il est destiné aux utilisateurs et aux développeurs qui souhaitent créer, rechercher et modifier des BDD [54].

- **Python** : est un langage de programmation interpreté et l'un des langages les plus utilisés actuellement. Il est à la fois simple et puissant, car il permet d'écrire des scripts simples et il a un environnement riche en librairies. Python a été pensé pour créer des codes complexes en peu de lignes [55] .

5.3 Fonctionnalités de l'application

Le système que nous avons développé offre les fonctionnalités suivantes.

- Présentation de l'entrepôt de données : **OLE-STL** assure l'affichage du schéma et de la structure générale du modèle des données pour le domaine choisi à des fins de test.
- Extraction des données des sources : trois possibilités sont offertes à l'utilisateur du système **OLE-STL** pour extraire des données des différentes sources.
- Transformation des données : avant leur chargement dans l'EDD, les données extraites des sources subissent différents types de transformation que l'utilisateur doit spécifier au préalable.

5.4 Enchainement général de l'application

L'exploitation du prototype OLE-STE se fait via un menu principale, tel que illustré dans la figure 5.1.

FIGURE 5.1 – Le menu principales de notre système "**OLE-STL**"

Dans un premier temps l'utilisateur commence par concevoir le schéma de son EDD et sa création dans un environnement dédié (SGBDR).

La figure 5.2, montre le schéma d'EDD du système "**OLE-STL**".

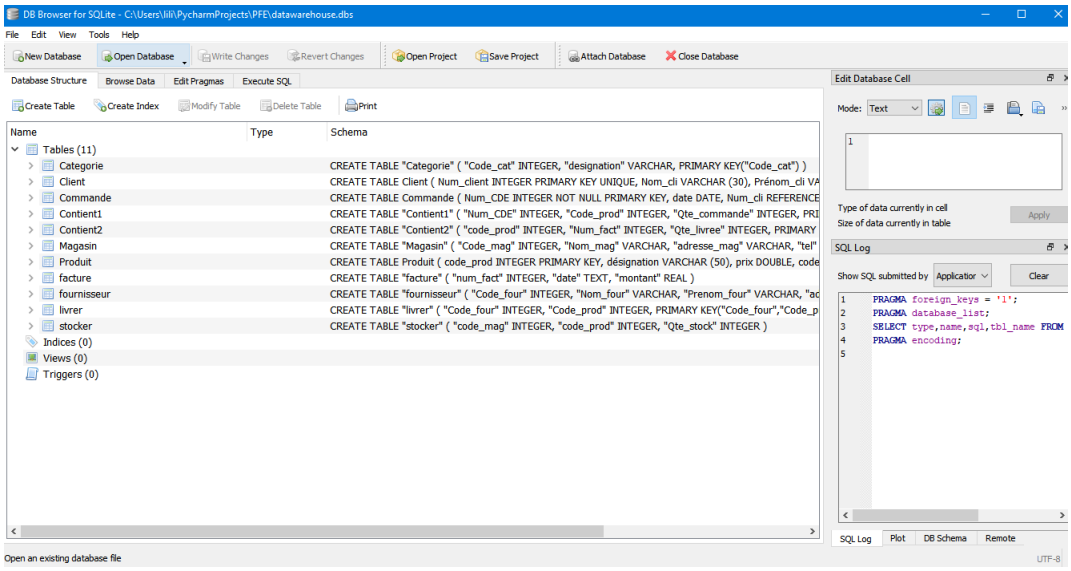

FIGURE 5.2 – Le schéma de l'EDD du système "**OLE-STL**"

- Extraction initial qui est la première étape fournie par notre système permet d'alimenter l'EDD, suivi par des extractions additives pour ajouter de nouvelles tables ou champs et enfin une extraction incrémentale en cas d'existence de mises à jour.

La figure 5.3, montre le menu d'extraction du système "**OLE-STL**".

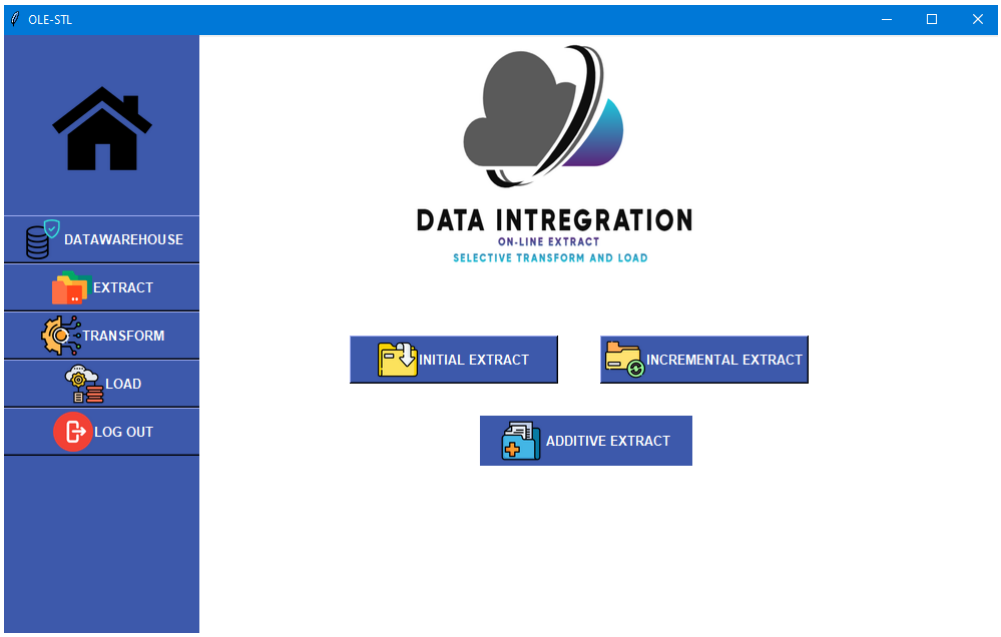

FIGURE 5.3 – Les trois extractions offertes par ""**OLE-STL**"

- Transformation des données qui est une transformation sélective permet aux utilisateurs d'avoir la possibilité d'appliquer une ou plusieurs règles de transformation.La figure 5.4, montre le menu des transformations du système (**OLE-STL**).

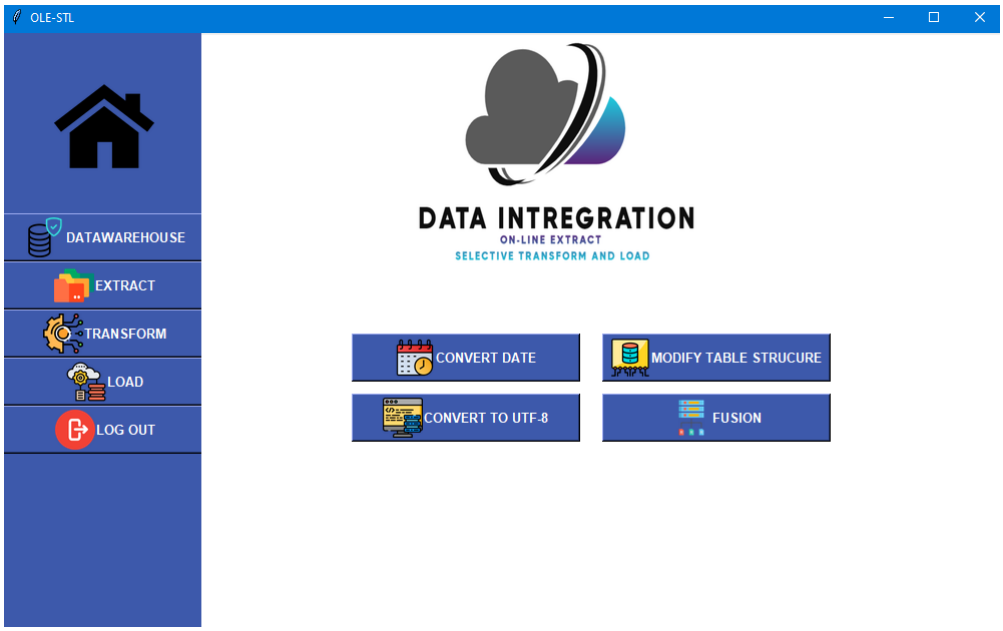

FIGURE 5.4 – Les transformation de notre système "**OLE-STL**"

- Chargement qui est la dernière étapes de notre système. Elle consiste à charger les données extraites et transformés dans l'EDD.

La figure 5.5, montre le menu de chargement du système "**OLE-STL**".

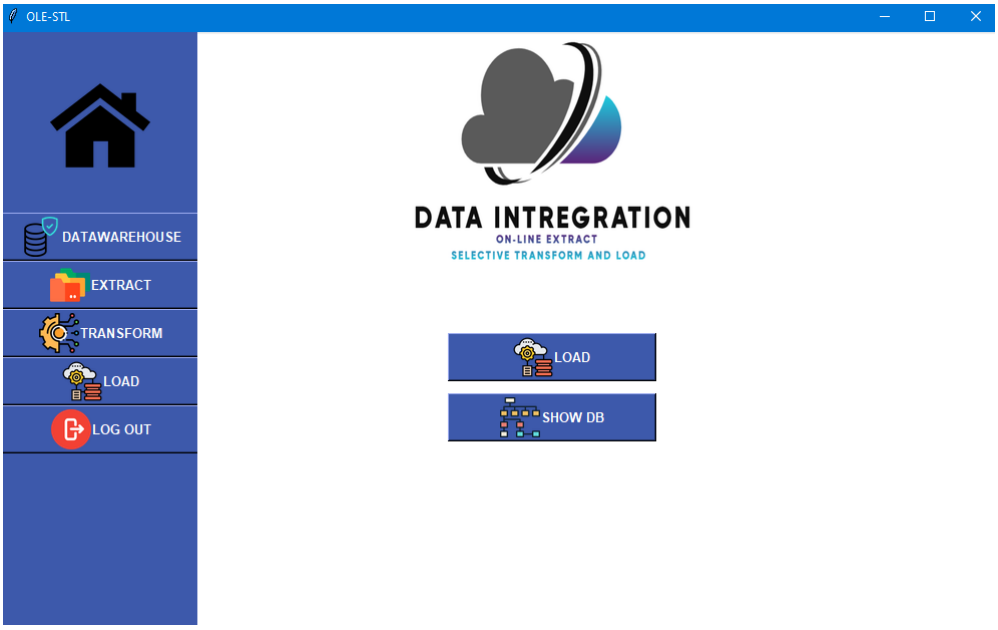

FIGURE 5.5 – Le menu de chargement du "**OLE-STL**"

5.5 Scénario illustratif

Considérons le domaine de la gestion commerciale, dans lequel un client achète des produits auprès d'un fournisseur.

1. Les différentes tables de faits et de dimension de l'EDD sont les suivantes :

- Les tables magasin, stocker et produit qui sont en format CSV.
- Les tables Client, Commande et facture qui sont en format BDDR.
- Les tables catégorie, fournisseur et livrer qui sont en format Excel.
- Les tables Contient1 et Contient2 qui sont en format XML.

2. L'extraction initial permet de lire toutes ces tables et les stockées dans une zone intermédiaire qui est la base de données relationnelles **staging area**.

La figure 5.6 , montre un exemple d'extraction de la table catégorie qui est au format Excel.

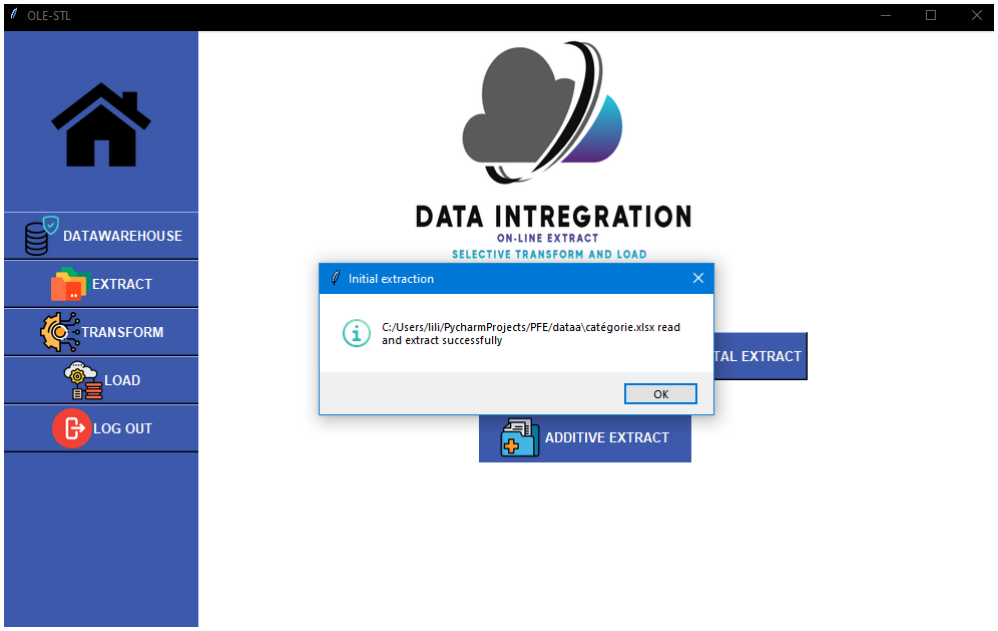

FIGURE 5.6 – Exemple d'extraction de la table catégorie

2-1. L'extraction additive de la table Type-paiement illustré dans la figure5.7.

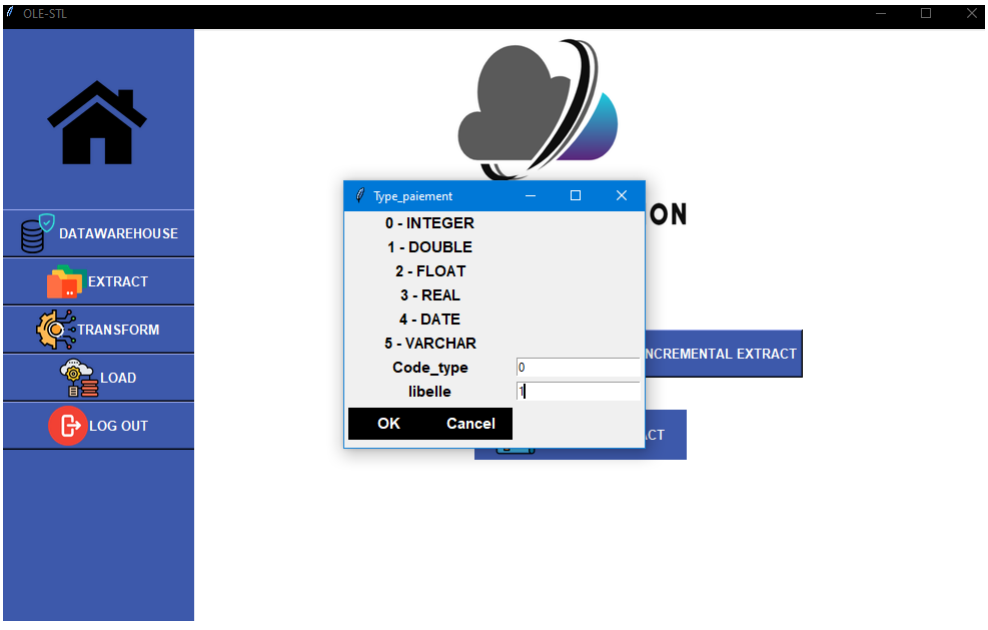

FIGURE 5.7 – Exemple d'extraction additive de la table Type-paiement

3. Appliquer les transformation suivantes :
-la figure 5.8, montre un exemple de transformation de la date.

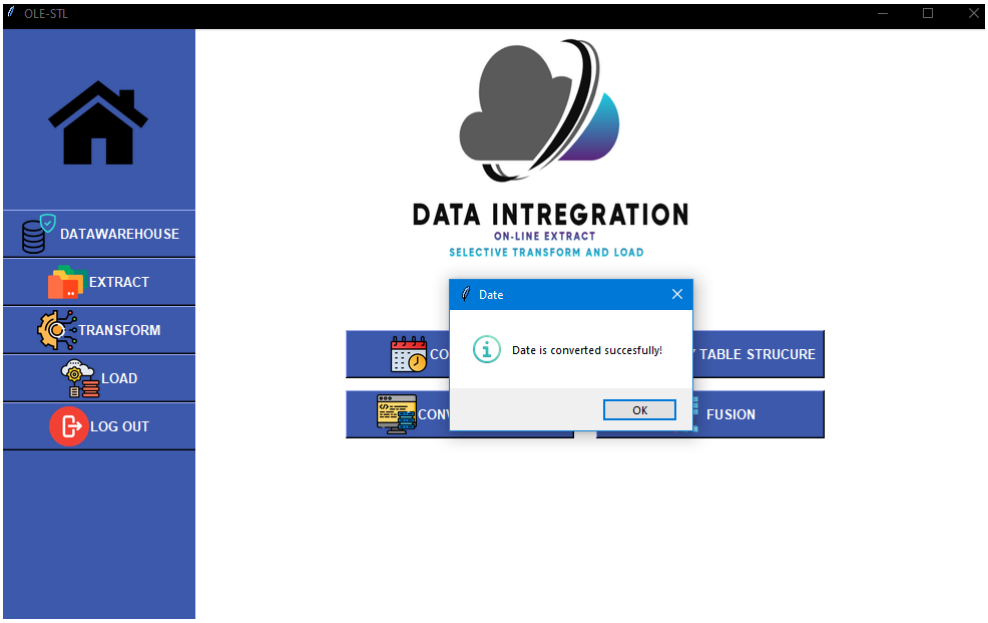

FIGURE 5.8 – Exemple de transformation de la date de la table facture
-la figure 5.9 , montre un exemple de transformation en UTF-8.

FIGURE 5.9 – Exemple de transformation la table magasin en UTF-8
4. Chargement des données dans l'EDD.
-la figure 5.10 , montre un exemple de chargement de la table magasin.

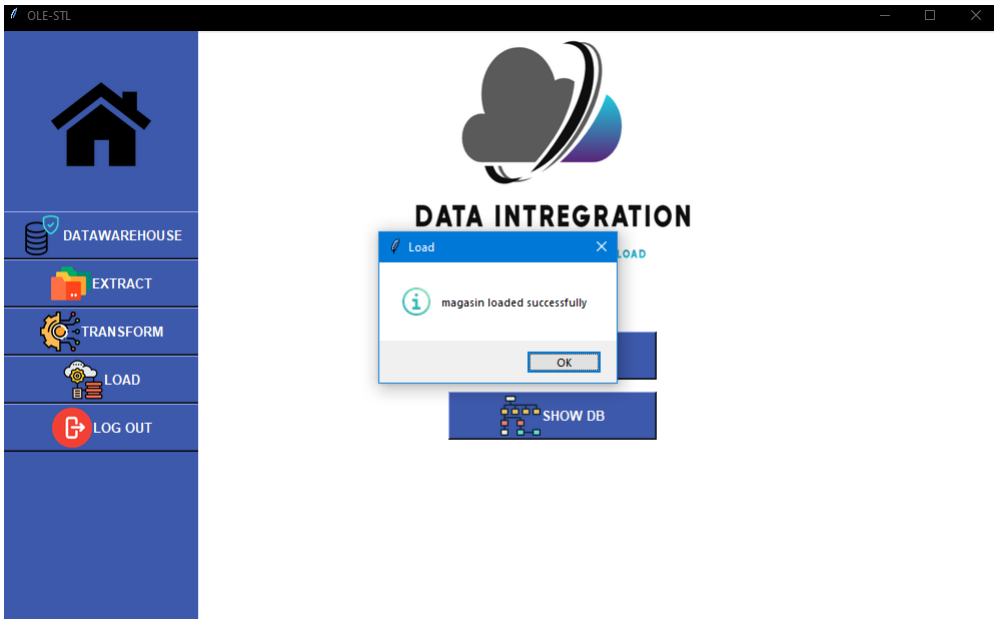

FIGURE 5.10 – Exemple de chargement de la table magasin

5.6 Conclusion

Le dernier chapitre de notre projet a été dédié à l'implémentation de l'application qui mettra en pratique la solution conceptuelle élaborée dans le chapitre précédent. Cette implémentation est en parfaite adéquation avec ce qui a été prévu au début de notre projet et qui n'est autre que la proposition d'une solution pour l'amélioration des techniques d'intégration des données des processus métiers basée sur la technologie ETL.

Ainsi, il apparait que l'application développée prend en compte les aspects essentiels couvrant le thème abordé, à savoir :

- Une architecture générale du système qui offre des fonctionnalités essentielles à l'extraction, la transformation et l'intégration des données.

- différentes illustrations, matérialisées par des captures d'écran mettent en évidence les trois principales phases de notre système.

Cependant, malgré les premiers résultats qui sont satisfaisants dans une large mesure, le prototype **OLE-STL** n'est qu'en sa version initiale qui nécessite des raffinements et des améliorations afin de prendre en charge d'autres formats de données, par exemple des données plats(en format PDF ou texte). Aussi, il demeure clair que l'implémentation actuelle, doit par ailleurs être revue. En effet, elle doit être paramétrable au niveau de la partie entrée afin de pouvoir prendre de manière générique les différents types de données de test qui seront relatives à divers domaines.

Conclusion Générale

Les outils ETL permettent d'améliorer considérablement l'efficacité décisionnelles des organisations par l'alimentation des entrepôts de données. Cependant, le volume croissant des données et les besoins évolutifs des utilisateurs imposent des solutions d'intégration de plus en plus éfféficace et performantes. Dans ce projet de fin d'études, nous avons abordé un problème d'actualité relatif à l'intégration de données hétérogènes au niveau d'un entrepôt de données. Nous avons commencé par comprendre les techniques d'intégration de données et nous avons cerné le problème qui consiste à des limitations dans le fonctionnement des ETL actuels.

Après analyse de l'état de l'art et les travaux existants, nous avons constaté qu'effectivement les outils ETL existants dans le marché présentent des insuffisances dans la prise en charge de données massives (Big data). Pour surmonter le problème nous avons apporté notre contribution qui consiste en une amélioration significative du mécanisme de fonctionnement de ces outils. Ainsi, nous avons proposé une solution améliorée pour résoudre le problème de l'intégration des données et nous avons revu le principe de ces outils. Les contributions majeurs consiste en une amélioration au niveau des deux premières phase ; c'est à dire l'extraction et la transformation. La solution proposée a été conçue et implémentée.

Sur le plan pratique, pour implémenter l'approche proposée nous avons utilisé le langage de programmation python sous l'environnement PyCharm. Le prototype réalisé, a été expérimenté sur un cas réel qui celui du domaine de la gestion commerciale. Les données relatives aux clients, commandes et fournisseurs ont été collectées et testées.

Aux termes de ce travail, nous pouvons affirmé que nous durant ce projet de fin d'études nous avons capitalisé les acquis suivants :

- **Du point de vue théorique :** Compréhension et utilisation des PM dans des domaines multiples et leur enrichissement par les données ainsi que la maitrise des outils permettant leur représentation.

- **Du point de vue pratique :** La capacité de répondre à des problèmes existants au niveau des entreprises et leur apporter les solutions appropriées au moyen de l'outil PyCharm puis les programmer en python. D'autre part, nous avons maitrisé l'environnement de composition de textes scientifique en latex.

- **Sur le plan méthodologique :** durant ce projet, j'ai appris à comprendre et à aborder de manière scientifique une question de recherche de manière générale. Le processus commence par cerner le problème, analyser l'état de l'art et enfin proposer une solution, la concevoir, l'implémenter puis l'expérimenter.

En ce qui concerne les perspectives futures de ce travail, j'espère et je souhaite que ce modeste travail engagera de futures étudiants pour la compréhension de l'intégration des données avec les outils ETL, et que d'éventuels futurs travaux d'exploration théoriques et pratiques viendront enrichir et améliorer ce qui a été déjà réalisé dans ce mémoire.

Bibliographie

[1] https://datawarehouseinfo.com/.

[2] Mohammed Oussama Kherbouche. *Contribution à la gestion de l'évolution des processus métiers*. PhD thesis, Université du Littoral Côté d'Opale, 2013.

[3] Mathias Weske, Marco Montali, Ingo Weber, and Jan vom Brocke. *Business Process Management : 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9–14, 2018, Proceedings*, volume 11080. Springer, 2018.

[4] Wil M. P. Van Der Aalst, Arthur H. M. Ter Hofstede, and Mathias Weske. Business process management : A survey. In *Proceedings of the 2003 International Conference on Business Process Management*, BPM'03, pages 1–12, Berlin, Heidelberg, 2003. Springer-Verlag.

[5] Mathias Weske, Wil MP Van Der Aalst, and HMW Verbeek. Advances in business process management. *Data & Knowledge Engineering*, 50(1) :1–8, 2004.

[6] Alessandro Margherita. Système de gestion des processus métier et activités : deux définitions intégratives pour construire un corps de connaissances opérationnel. *Journal de gestion des processus métier*, 2014.

[7] Object Management Group. Business process modeling notation (bpmn) version 1.0. omg final adopted specification. object management group, 2006.

[8] OASIS. Web services business process execution language version 2.0. http ://docs.oasis-open.org/wsbpel/2.0/, 2007.

[9] Mathias Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition*. Springer, 2012.

[10] luc Maranget Philipe Babstie. Programmation et algorithmique. http://gallium.inria.fr/~maranget/X/421/poly/poly.pdf.

[11] Boualem Benatallah, Fabio Casati, and Farouk Toumani. Web service conversation modeling : A cornerstone for e-business automation. *IEEE Internet Computing*, 8(1) :46–54, 2004.

[12] Hedi Dhouibi. *Utilisation des réseaux de Petri à intervalles pour la régulation d'une qualité : application à une manufacture de tabac*. PhD thesis, Ecole Centrale de Lille ; Université des Sciences et Technologie de Lille-Lille I, 2005.

[13] Les graphes, un outil de modélisation. http://ressources.aunege.fr/nuxeo/site/esupversions/2b1c56b6-109d-488a-94a3-3ea525f8beef/ModAidDec/cours/l2/l2.pdf.

[14] Didier Müller. *Introduction à la théorie des graphes*. Commission romande de mathématique, 2011.

[15] Aloulou and Houssem. Dérivation de diagrammes de séquence uml compactes à partir de traces d'exécution en se basant des heuristiques. 2016.

[16] Jan Recker. Bpmn modeling-who, where, how and why. *BPTrends*, pages 1–8, 2008.

[17] Business process modeling techniques with examples. https://creately.com/blog/diagrams/business-process-modeling-techniques/, 22 April 2021.

[18] Chun Ouyang, Marlon Dumas, Arthur HM Ter Hofstede, and Wil MP Van der Aalst. From bpmn process models to bpel web services. In *2006 IEEE International Conference on Web Services (ICWS'06)*, pages 285–292. IEEE, 2006.

[19] Sumit Misra, Sanjoy Kumar Saha, and Chandan Mazumdar. Performance comparison of hadoop based tools with commercial etl tools–a case study. In *International Conference on Big Data Analytics*, pages 176–184. Springer, 2013.

[20] Neepa Biswas, Samiran Chattapadhyay, Gautam Mahapatra, Santanu Chatterjee, and Kartick Chandra Mondal. A new approach for conceptual extraction-transformation-loading process modeling. *International Journal of Ambient Computing and Intelligence (IJACI)*, 10(1) :30–45, 2019.

[21] Xin Luna Dong and Divesh Srivastava. Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)*, pages 1245–1248. IEEE, 2013.

[22] Qu'est-ce que l'integration des donnees. https://zipreporting.com/fr/data-integration/what-is-data-integration.html, April 05 2021.

[23] Qu'est-ce que l'intégration de données ? https://www.talend.com/fr/resources/what-is-data-integration/, 2022.

[24] La Rédaction TechTarget. Base de données relationnelle. https://www.lemagit.fr/definition/Base-de-donnees-relationnelle, août 2014.

[25] Geneviève PUJOLLE Gilles ZURFLUH Claude CHRISMENT, Jacques LUGUET. Base de données relationnelle. https://www.techniques-ingenieur.fr/base-documentaire/archives-th12/archives-technologies-logicielles-et-architecture-des-systemes-tiahb/archive-1/bases-de-donnees-relationnelles-h2038/qu-appelle-t-on-sgbdr-h2038niv10007.html#:~:text=Un%20syst%C3%A8me%20de%20gestion%20de,les%20principes%20du%20mod%C3%A8le%20relationnel., 10 févr 1997.

[26] David S. Bases de données relationnelles : Tout ce qu'il y a à savoir. https://datascientest.com/bases-de-donnees-relationnelles, 3 juin 2021.

[27] Qu'est-ce que gav (global as view) ? https://www.geeksforgeeks.org/what-is-gav-global-as-view/?ref=lbp, 24 avril 2020.

[28] Bendida sihem Amer fatima. *vers une approche d'intégration des base de données hétérogènes via les méta-shcéma XML*. PhD thesis, Université Dr. Tahar Moulay Saida, 2017.

[29] Local comme vue (lav). https://www.geeksforgeeks.org/local-as-view-lav/, 17 août 2020.

[30] Mohand-Said Hacid and Chantal Reynaud. L'intégration de sources de données. *Revue Information-Interaction-Intelligence*, 3(4), 2004.

[31] https://www.redhat.com/fr/topics/integration/what-is-.

[32] https://www.axysweb.com/integration-applications-eai-esb/.

[33] C. Desrosiers S. Chafki. Mti820 acetates etl 1pp. , 2011.

[34] Lahmar Fatima épouse Boulçane. Une approche hybride d'intégration de sources de données hétérogènes dans les datawarehouses. *Université Mentouri de Constantine Faculté des Sciences de l'Ingénieur*, 2011.

[35] Gabriel Chandesris. Systèmes d'intégration de données en biologie.

[36] Bastien L. Data warehouse (entrepôt de données) définition : qu'est-ce que c'est ?) ? https://www.lebigdata.fr/data-warehouse-entrepot-donnees-definition, 14 février 2018.

[37] https://www.astera.com/fr/type/blog/data-warehouse-architecture/.

[38] Margot. Data warehouse : qu'est-ce que c'est et comment les utiliser ? https://datascientest.com/data-warehouse, 3/2 2021.

[39] La Rédaction JDN. https://www.journaldunet.fr/business/dictionnaire-du-marketing/1198305-etl-outils-definition-traduction/, 03 Février 2019.

[40] https://www.next-decision.fr/wiki/outil-etl-script.

[41] https://www.talend.com/fr/resources/elt-tools/.

[42] https://hevodata.com/learn/streaming-etl/.

[43] https://hevodata.com/learn/data-pipeline/.

[44] Dimitrios Skoutas and Alkis Simitsis. Ontology-based conceptual design of etl processes for both structured and semi-structured data. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(4) :1–24, 2007.

[45] Xiufeng Liu, Christian Thomsen, and Torben Bach Pedersen. Etlmr : a highly scalable dimensional etl framework based on mapreduce. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 96–111. Springer, 2011.

[46] Pablo Michel Marín-Ortega, Viktor Dmitriyev, Marat Abilov, and Jorge Marx Gómez. Elta : new approach in designing business intelligence solutions in era of big data. *Procedia technology*, 16 :667–674, 2014.

[47] Mahfoud Bala, Omar Boussaid, Zaia Alimazighi, and Fadila Bentayeb. Pf-etl : vers l'intégration de données massives dans les fonctionnalités d'etl. In *Inforsid*, pages 61–76, 2014.

[48] Mahfoud Bala, Oussama Mokeddem, Omar Boussaid, and Zaia Alimazighi. Une plateforme etl parallèle et distribuée pour l'intégration de données massives. In *EGC*, pages 455–460, 2015.

[49] Shu-Sheng Guo, Zi-Mu Yuan, Ao-Bing Sun, and Qiang Yue. A new etl approach based on data virtualization. *Journal of Computer Science and Technology*, 30(2) :311–323, 2015.

[50] Mahfoud Bala, Omar Boussaid, and Zaia Alimazighi. Extracting-transforming-loading modeling approach for big data analytics. *International Journal of Decision Support System Technology (IJDSST)*, 8(4) :50–69, 2016.

[51] Hana Mallek, Faiza Ghozzi, Olivier Teste, and Faiez Gargouri. Bigdimetl : Etl for multidimensional big data. In *International Conference on Intelligent Systems Design and Applications*, pages 935–944. Springer, 2016.

[52] Hana Mallek, Faiza Ghozzi, and Faiez Gargouri. Towards extract-transform-load operations in a big data context. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, 12(2) :77–95, 2020.

[53] Neepa Biswas, Samiran Chattopadhyay, Gautam Mahapatra, Santanu Chatterjee, and Kartick Chandra Mondal. Sysml based conceptual etl process modeling. In *International Conference on Computational Intelligence, Communications, and Business Analytics*, pages 242–255. Springer, 2017.

[54] Db browser for sqlite. https://sqlitebrowser.org/.

[55] Db browser for sqlite. https://www.python.org/.

.