

HONORIS UNITED UNIVERSITIES

2024/2025

SPÉCIALITÉ : ERP-BI

Conception d'un tableau de bord de Business Intelligence pour l'analyse de diverses sources de données.

Réalisé par: ELGHARNOUGUI Ismail

Encadré par: DERBEL Mohamed Aymen



Remerciements

En premier lieu, je tiens à remercier toute l'équipe pédagogique de **l'École Supérieure Privée d'Ingénierie et de Technologies** ainsi que les responsables de la formation d'ingénieur pour avoir assuré un cadre théorique solide et stimulant tout au long de ma formation.

Je tiens également à exprimer ma sincère gratitude envers **M. Derbel Mohamed Aymen**, mon encadrant de stage, pour sa disponibilité, ses précieux conseils, et son soutien constant durant tout le déroulement du projet. Son expertise a été déterminante pour la bonne réalisation de ce travail.

Je remercie chaleureusement **M. Chiheb Kitar** et **M. Riadh Chnitir** pour leur aide précieuse, leurs recommandations et leur accompagnement tout au long de cette expérience. Leur soutien a été essentiel pour surmonter les défis rencontrés lors du projet.

Je tiens également à exprimer ma reconnaissance envers toute l'équipe de **P2MS**, pour l'accueil chaleureux et le cadre de travail propice à l'apprentissage. Leur collaboration et leur soutien m'ont permis de tirer pleinement profit de cette expérience professionnelle enrichissante.

Table des matières

Introduction générale	1
1 Cadre général du projet	2
1.1 Introduction	2
1.2 Présentation générale de l'entreprise	2
1.2.1 Présentation de Planet Medical Maintenances Services (P2MS)	2
1.3 Présentation du projet	2
1.3.1 Contexte du projet	2
1.3.2 Critique de l'existant	3
1.3.3 Solution proposée	3
1.3.4 Déploiement et intégration technique	3
1.4 Méthodologie GIMSI	3
1.5 Comparaison des modèles en étoile et en flocon	5
1.6 Conclusion	5
2 La mise en œuvre	6
2.1 Introduction	6
2.2 Environnement matériel	6
2.3 Environnement Logiciel	6
3 Conception	12
3.1 Introduction	12
3.2 Modélisation dimensionnelle des données	12
3.3 Modèles logiques de données	12
3.3.1 Modèle en étoile	12
3.3.2 Modèle en flocon de neige	13
3.3.3 Modèle en constellation	13
3.3.4 Comparaison entre les trois modèles	14
3.3.5 Approche Kimball vs Inmon :	14
3.3.5.1 Approche Kimball (Bottom-Up)	14
3.3.5.2 Approche Inmon (Top-Down)	15
3.3.6 Identification des dimensions et de la table de faits	17
3.3.6.1 La dimension Clients	17
3.3.6.2 La dimension Fournisseurs	17
3.3.6.3 La dimension Régions	17
3.3.6.4 La dimension Articles	18
3.3.6.5 La dimension Dates	18
3.3.6.6 Table des faits – FaitsVentes	19
3.4 Conclusion	19
4 Mise en œuvre et réalisation	21
4.1 Introduction	21

4.2	Implémentation	21
4.2.1	Analyse Exploratoire des Données (EDA)	21
4.2.1.1	Fichier brands-cleaned	21
4.2.2	Phase d'Alimentation ETL (Extract, Transform, Load)	24
4.2.2.1	Définition Phase ETL	24
4.2.2.2	Alimentation de la StagingArea	24
4.2.2.3	Alimentation de la DataWarehouse	27
4.2.3	Phase de Construction du Cube OLAP	39
4.2.3.1	Définition Cube OLAP	39
4.2.3.2	Hiérarchie Cube OLAP	41
4.2.3.3	Mesure :	45
4.2.3.4	Filtres Dynamiques	46
4.2.4	Phase de Développement et Déploiement dans Power BI	48
4.2.5	Phase de Machine Learning	51
4.2.6	Phase de Déploiement d'Application Angular et Spring Boot	52
4.2.6.1	Power BI Service	52
4.2.6.2	Inscription et Connexion Dynamique avec MySQL	53
5	Conclusion	58

Table des figures

1.1	GIMSI methodology	4
2.1	Logo Overleaf	7
2.2	Logo SSIS	7
2.3	Logo SSMS	7
2.4	Logo SSAS	8
2.5	Logo Power BI	8
2.6	Logo Python	8
2.7	Logo Flask	9
2.8	Logo Angular	9
2.9	Logo Spring Boot	9
2.10	Logo MySQL	10
2.11	Logo Postman	10
2.12	Logo VS Code	10
2.13	Logo StarUML	11
2.14	Logo Microsoft Teams	11
3.1	Modélisation en étoile	13
3.2	Modélisation en flocon de neige	13
3.3	Modèle en constellation	14
3.4	Structure de Approche Kimball (Bottom-Up)	15
3.5	Structure de Approche Inmon (Top-Down)	16
3.6	Modele de DataWarehouse	20
4.1	Distribution des familles d'articles	22
4.2	Distribution des prix de vente des articles	22
4.3	Répartition des articles par fournisseur	23
4.4	Evolution des prix de vente moyens par Collection	23
4.5	Staging Area	24
4.6	Structure de la dimension DimClient	25
4.7	Reset Clé technique	26
4.8	Composants Utilisés dans SA	27
4.9	Alimentation de la dimension DimClient	27
4.10	Data Flow de la dimension DimClient	28
4.11	Tri avec clé metier	28
4.12	Merge Join de Type Left Outer Join	29
4.13	Conditionnal Split Insert ou Update	30
4.14	Mapping OLE DB Destination	31
4.15	Requete Update OLE DB Command	32
4.16	Fact Table	33
4.17	Structure de Fact Table	34
4.18	Exemple de Verification de Article-FK	35
4.19	Verification Clé Primaire composé	36

4.20 Lookup No Match Output (insert)	37
4.21 Lookup Match Output (update) Fact Table	38
4.22 Implementation Fact Table	39
4.23 Cube OLAP P2MS	40
4.24 Hiérarchie Date	41
4.25 Attribute Relationships	42
4.26 Execution Hierarchie Date	42
4.27 Hierarchie Region	43
4.28 Attribute Relationships	43
4.29 Execution Hierarchie Region	44
4.30 Mesure Marge Brute	45
4.31 Mesure Totat des Ventes	45
4.32 Filtre dynamique Top 5 Clients	46
4.33 Execution Filtre dynamique Top 5 Clients	46
4.34 Filtre dynamique Top 5 Articles	47
4.35 Execution Filtre dynamique Top 5 Articles	47
4.36 OverView	48
4.37 Dashboard Article	49
4.38 Dashboard Client	50
4.39 Dashboard Fournisseur	51
4.40 Predition Les articles plus demandés dans un periode	52
4.41 Power BI Service	53
4.42 Sign Up Interface	54
4.43 Sign In Interface	54
4.44 Contrôle de saisie	55
4.45 Table User	55
4.46 postman verification sign up	56
4.47 postman verification sign up	57

Liste des tableaux

1.1	Comparaison entre les modèles de datawarehouse en étoile et en flocon	5
2.1	Tableau descriptif de l'environnement matériel	6
3.1	Tableau comparatif entre les trois modèles logiques de data warehouse	14
3.2	Comparaison entre l'approche Kimball et l'approche Inmon	16
3.3	Détails de la dimension Clients	17
3.4	Détails de la dimension Fournisseurs	17
3.5	Détails de la dimension Régions	18
3.6	Détails de la dimension Articles	18
3.7	Détails de la dimension Dates	18
3.8	Détails de la table des faits FaitsVentes	19

Introduction générale

L'essor de l'informatique et des technologies de l'information a considérablement transformé le monde des entreprises et de la gestion des données. Dans un contexte où la prise de décision repose de plus en plus sur des analyses de données fiables et pertinentes, la mise en œuvre de solutions de Business Intelligence (BI) devient un enjeu stratégique pour les entreprises modernes.

Dans ce cadre, l'optimisation de la gestion des données, la structuration des processus décisionnels et l'intégration d'outils analytiques avancés sont des facteurs clés pour améliorer la performance des organisations. L'absence de solutions efficaces de traitement des données peut avoir un impact négatif sur la compétitivité, la réactivité et la qualité des décisions d'une entreprise.

Le présent rapport se propose de retracer les différentes étapes de conception et de mise en œuvre d'une solution BI pour **Planet Medical Maintenances Services (P2MS)**. Il couvre notamment la modélisation des données, la création d'un entrepôt de données, l'intégration des outils de visualisation avec **Power BI**, ainsi que l'exploitation de techniques de **Machine Learning**. Ce projet, réalisé entre le 01/07/2024 et le 31/08/2024, a permis d'améliorer la gestion des ventes, des fournisseurs et des clients au sein de l'entreprise.

Ce stage m'a offert une opportunité précieuse pour découvrir le monde professionnel dans un cadre stimulant. J'ai pu renforcer mes compétences en gestion de données et en développement de solutions décisionnelles, des compétences essentielles pour ma formation d'ingénieur à **ESPRIT**. L'expérience acquise au sein de P2MS a enrichi à la fois mes connaissances techniques et mes aptitudes en gestion de projet.

CHAPITRE 1

Cadre général du projet

1.1 Introduction

Dans ce chapitre introductif, je vais tout d'abord présenter le cadre dans lequel s'est déroulé mon stage, à savoir l'organisme d'accueil P2MS (**Planet Medical Maintenances Services**). Ensuite, je présenterai l'équipe informatique et ses missions.

1.2 Présentation générale de l'entreprise

1.2.1 Présentation de Planet Medical Maintenances Services (P2MS)

Planet Medical Maintenances Services (P2MS) est un fournisseur spécialisé dans la vente d'équipements médicaux, basé à Ben Arous, Tunis. L'entreprise se distingue par son expertise dans la maintenance et la fourniture d'équipements médicaux de haute qualité pour les hôpitaux et les cliniques. P2MS est un acteur clé dans l'industrie médicale en Tunisie, assurant la livraison, l'installation, ainsi que l'entretien régulier des équipements vendus.

Avec une équipe de professionnels expérimentés, P2MS met l'accent sur l'innovation technologique et le service client afin de répondre aux besoins exigeants du secteur médical. L'entreprise s'engage à offrir des solutions complètes, garantissant ainsi la performance et la durabilité des équipements.

1.3 Présentation du projet

1.3.1 Contexte du projet

Ce projet s'inscrit dans le cadre d'un stage de deux mois réalisé au sein de l'école ESPRIT pour l'entreprise Planet Medical Maintenances Services (P2MS). L'objectif principal est de développer un tableau de bord de Business Intelligence (BI) permettant d'améliorer la gestion des ventes, des fournisseurs et des clients à travers des outils analytiques avancés. Ce tableau de bord fournira des indicateurs clés de performance (KPI) sous forme de rapports interactifs, permettant à P2MS d'optimiser ses processus et de mieux comprendre ses performances.

1.3.2 Critique de l'existant

Actuellement, P2MS utilise des outils de gestion basiques qui ne permettent pas une analyse approfondie des données ou des tendances. Les principales lacunes du système actuel incluent :

- Absence de rapports BI pertinents pour une prise de décision rapide et informée,
- Les flux de données sont mal structurés et difficiles à analyser en temps réel.

1.3.3 Solution proposée

La solution proposée repose sur la mise en place d'un système décisionnel structuré en plusieurs phases :

- **Phase 1** : Mise en place du processus ETL (Extraction, Transformation, Chargement). Cette étape vise à intégrer et structurer les données provenant de différentes sources comme les ventes, les fournisseurs et les clients. Ces données sont centralisées dans un entrepôt de données (data warehouse) pour faciliter les analyses.
- **Phase 2** : Création d'un Cube OLAP (Online Analytical Processing). Le cube OLAP est construit à partir des données du data warehouse pour permettre des analyses multidimensionnelles. Ce cube facilite les requêtes complexes, comme l'analyse des ventes par région, par produit, ou par période, et aide à la génération rapide de rapports de synthèse.
- **Phase 3** : Création de tableaux de bord interactifs avec **Power BI**. Ces tableaux de bord visualisent les données clés issues du cube OLAP et permettent à P2MS de suivre les KPI en temps réel. Les utilisateurs peuvent interagir avec ces rapports pour explorer les performances et les tendances des ventes, des fournisseurs et des clients.
- **Phase 4** : Intégration de techniques de machine learning. Cette étape permet d'analyser les données pour anticiper les tendances et fournir des recommandations stratégiques sur la gestion des stocks, la prévision des ventes, et l'optimisation des ressources.
- **Phase 5** : Déploiement de Power BI Service dans un site web d'entreprise utilisant **Angular**, **Spring Boot** et **MySQL**. Le tableau de bord Power BI est intégré dans un site web sécurisé, accessible aux utilisateurs autorisés.

1.3.4 Déploiement et intégration technique

La phase finale du projet consiste à déployer le tableau de bord Power BI via Power BI Service. Celui-ci est intégré dans un site web basé sur Angular et un backend Spring Boot, avec une base de données MySQL ,y compris les ventes, les finances et la logistique. L'intégration avec le backend garantit également une gestion sécurisée et performante des flux de données.

1.4 Méthodologie GIMSI

Dans ce projet, nous avons suivi la méthodologie GIMSI (Gestion Intégrée des Méthodologies et Systèmes d'Information), qui est couramment utilisée pour la mise en œuvre de projets de Business Intelligence. Elle se décompose en 4 phases et 10 étapes clés :

- **Phase 1 : Initialisation du projet**

- Définition des besoins décisionnels

- Analyse des objectifs stratégiques de l’entreprise

- **Phase 2 : Conception du système décisionnel**

- Modélisation des données (modèle en étoile pour le data warehouse)
- Création des tables de faits et des dimensions
- Choix des indicateurs de performance (KPI)

- **Phase 3 : Développement et intégration**

- Processus ETL pour l’extraction, la transformation et le chargement des données
- Mise en place du cube OLAP pour l’analyse multidimensionnelle

- **Phase 4 : Déploiement et suivi**

- Déploiement des tableaux de bord BI
- Formation des utilisateurs
- Suivi et maintenance des systèmes décisionnels

Description des phases Méthode GIMSI

Phase	N°	Étape	Objectifs
Identification	1	Environnement de l’entreprise	Analyse de l’environnement économique et de la stratégie de l’entreprise afin de définir le périmètre et la portée du projet
	2	Identification de l’entreprise	Analyse des structures de l’entreprise pour identifier les processus, activités et acteurs concernés
Conception	3	Définition des objectifs	Sélection des objectifs tactiques de chaque équipe
	4	Construction du tableau de bord	Définition du tableau de bord de chaque équipe
	5	Choix des indicateurs	Choix des indicateurs en fonction des objectifs choisis
	6	Collecte des informations	Identification des informations nécessaires à la construction des indicateurs
	7	Système de tableaux de bord	Construction du système de tableau de bord, contrôle de la cohérence globale
Mise en œuvre	8	Choix des progiciels	Élaboration de la grille de sélection pour le choix des progiciels adéquats
	9	Intégration et déploiement de la solution	Implantation des progiciels, déploiement à l’entreprise
Suivi permanent	10	Audit du système	Suivi permanent du système www.piloter.org (c) Eyrolles

FIGURE 1.1 – GIMSI methodology

1.5 Comparaison des modèles en étoile et en flocon

Critères	Modèle en étoile (Star-Schema)	Modèle en flocon (Snow-flake Schema)
Structure des tables	Les tables de dimensions sont dénormalisées, directement reliées à la table des faits.	Les tables de dimensions sont normalisées, avec des relations entre plusieurs sous-tables.
Simplicité	Plus simple et facile à comprendre. Moins de jointures nécessaires.	Plus complexe, nécessite plus de jointures pour accéder aux données.
Performance des requêtes	Requêtes plus rapides avec moins de jointures.	Requêtes plus lentes avec davantage de jointures à gérer.
Stockage	Prend plus de place en raison de la redondance des données.	Moins d'espace de stockage nécessaire grâce à la normalisation.
Utilisation	Utilisé lorsque la simplicité et la rapidité des requêtes sont importantes.	Utilisé pour des systèmes plus complexes nécessitant une meilleure organisation des données.

TABLE 1.1 – Comparaison entre les modèles de datawarehouse en étoile et en flocon

1.6 Conclusion

Ce premier chapitre présente l'entreprise P2MS, son contexte ainsi que le cadre du projet. Nous avons identifié les problèmes actuels auxquels P2MS est confrontée et proposé une solution BI structurée en plusieurs phases : processus ETL, création d'un cube OLAP, mise en place de tableaux de bord avec Power BI, intégration de machine learning, et déploiement d'une solution complète dans un site web utilisant Angular et Spring Boot. Le prochain chapitre approfondira l'analyse des besoins et détaillera les solutions techniques mises en œuvre.

CHAPITRE 2

La mise en œuvre

2.1 Introduction

Dans ce deuxième chapitre, nous allons présenter les différents outils matériels, logiciels et langages de programmation utilisés pour le développement de notre projet de **Business Intelligence (BI)** et de **Machine Learning** chez **Planet Medical Maintenances Services (P2MS)**.

2.2 Environnement matériel

Caractéristiques	Mon Ordinateur
Modèle	Asus TUF DASH F15 2022
Mémoire vive	16 GB DDR5
Stockage	512 SSD NVME
Carte Graphique	Nvidia RTX 3050
Processeur	I7 12650H
Système d'exploitation	Windows 11 64 bits

TABLE 2.1 – Tableau descriptif de l'environnement matériel

2.3 Environnement Logiciel

Dans cette section, nous allons détailler l'environnement logiciel utilisé pour le développement du projet.

Overleaf :

Overleaf est utilisé pour rédiger la documentation du projet avec LaTeX, permettant de structurer efficacement le rapport tout en collaborant avec l'équipe en temps réel.



FIGURE 2.1 – Logo Overleaf

SQL Server Integration Services (SSIS) :

SSIS a été utilisé pour la mise en place des processus **ETL** (Extraction, Transformation, Chargement), facilitant l'intégration des données provenant de différentes sources vers le datawarehouse.



FIGURE 2.2 – Logo SSIS

SQL Server Management Studio (SSMS) :

SSMS est l'outil utilisé pour administrer et gérer les bases de données, notamment pour créer des requêtes SQL, effectuer des opérations sur les tables et vérifier l'intégrité des données.



FIGURE 2.3 – Logo SSMS

SQL Server Analysis Services (SSAS) :

SSAS permet la création des **cubes OLAP**, une technologie utilisée pour l'analyse multidimensionnelle des données, facilitant la création de rapports et la réalisation d'analyses complexes.



FIGURE 2.4 – Logo SSAS

Power BI :

Power BI est utilisé pour la création de tableaux de bord interactifs permettant une visualisation claire et dynamique des **KPI** (indicateurs clés de performance) et des rapports décisionnels.



FIGURE 2.5 – Logo Power BI

Python :

Python est le langage utilisé pour développer les algorithmes de **Machine Learning**, permettant de réaliser des analyses prédictives et d'identifier des tendances à partir des données. Des bibliothèques comme **Pandas**, **Scikit-learn** et **TensorFlow** ont été employées.



FIGURE 2.6 – Logo Python

Flask :

Flask est un framework léger utilisé pour développer des APIs Python afin d'intégrer les modèles de machine learning dans des applications web ou mobiles.



FIGURE 2.7 – Logo Flask

Angular :

Angular a été utilisé pour développer l'interface **frontend** du site web, permettant d'intégrer les visualisations Power BI de manière interactive et en temps réel pour les utilisateurs.



FIGURE 2.8 – Logo Angular

Spring Boot :

Spring Boot est utilisé pour développer le **backend** de l'application, facilitant la gestion des APIs et la connexion avec la base de données MySQL.



FIGURE 2.9 – Logo Spring Boot

MySQL :

MySQL est la base de données relationnelle utilisée pour stocker les données du projet, notamment les informations relatives aux ventes, clients, fournisseurs et autres dimensions analysées.



FIGURE 2.10 – Logo MySQL

Postman :

Postman est utilisé pour tester les APIs développées avec Flask et Spring Boot, facilitant l'interaction avec les endpoints et la validation des fonctionnalités.



FIGURE 2.11 – Logo Postman

VS Code :

Visual Studio Code est l'éditeur de code utilisé pour écrire et déboguer le code des différents composants du projet, qu'il s'agisse de la partie backend (Spring Boot), frontend (Angular), ou des scripts Python pour le machine learning. Il offre des extensions facilitant la gestion des versions et l'intégration avec Git.

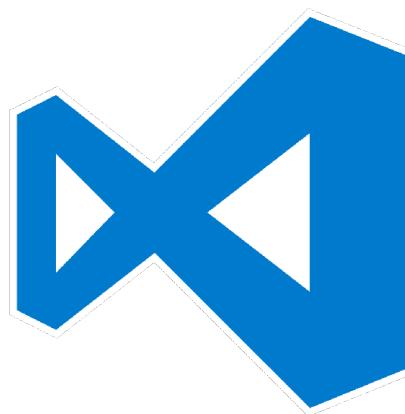


FIGURE 2.12 – Logo VS Code

StarUML :

StarUML a été utilisé pour modéliser le **datawarehouse** en étoile. Ce modèle centralise les données

autour d'une table de faits (liée aux ventes) et des tables de dimensions (produits, clients, etc.), facilitant les analyses OLAP et la génération de rapports dans Power BI.



FIGURE 2.13 – Logo StarUML

Microsoft Teams :

Microsoft Teams est utilisé pour faciliter la communication avec l'équipe projet et l'encadreur, permettant de suivre les différentes étapes du projet et d'assurer la collaboration entre les membres.



FIGURE 2.14 – Logo Microsoft Teams

3.1 Introduction

Dans ce deuxième chapitre, nous allons commencer par présenter la modélisation dimensionnelle utilisée pour notre projet chez **Planet Medical Maintenances Services (P2MS)**. Ensuite, nous aborderons la conception du modèle physique de données, avec les détails des différentes dimensions et tables de faits, ainsi que l'architecture globale du projet.

3.2 Modélisation dimensionnelle des données

La modélisation dimensionnelle dans notre projet consiste à définir les tables de dimensions (Dim) et la table de faits (Fact Table). Ces deux éléments sont cruciaux pour l'analyse des données dans un système de Business Intelligence.

- **Dimensions** : Elles représentent des axes d'analyse qui permettent de catégoriser et d'analyser les données selon plusieurs perspectives (clients, régions, dates, etc.).
- **Table de faits** : Elle centralise les mesures ou indicateurs quantitatifs clés qui sont analysés par rapport aux dimensions.

3.3 Modèles logiques de données

3.3.1 Modèle en étoile

Le modèle en étoile est utilisé dans notre projet pour sa simplicité et sa performance lors des requêtes. Il comprend une table de faits centrale et des dimensions qui sont directement reliées à cette table. Ce modèle est particulièrement adapté pour des analyses simples et rapides des données médicales et commerciales de P2MS.

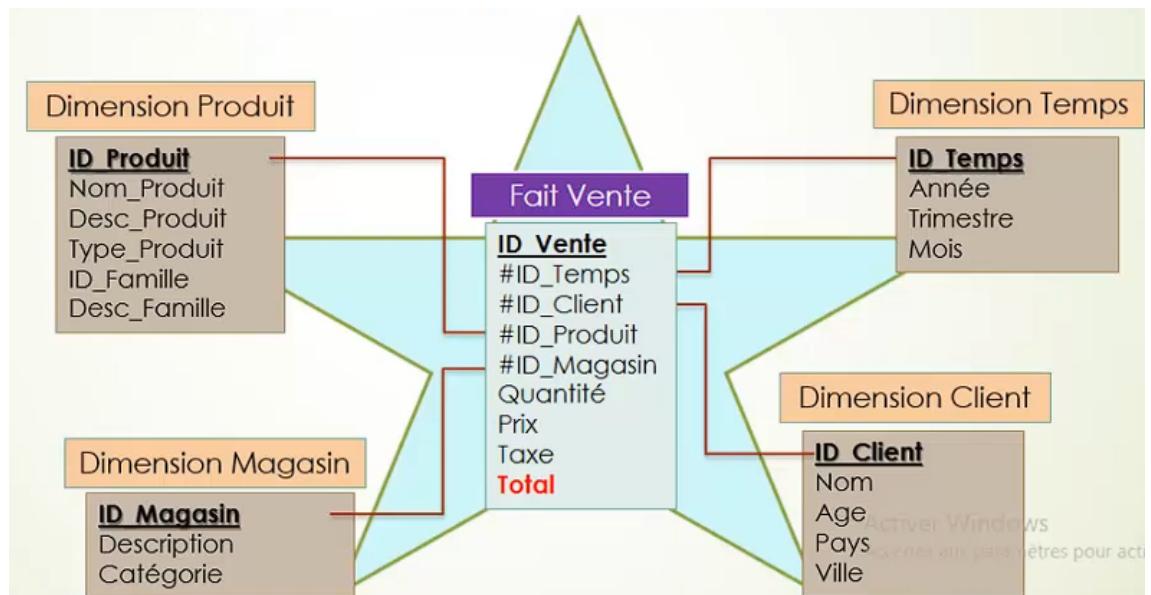


FIGURE 3.1 – Modélisation en étoile

3.3.2 Modèle en flocon de neige

Le modèle en flocon de neige est une extension du modèle en étoile dans lequel les dimensions sont normalisées. Cela permet de réduire la redondance, mais peut ralentir les performances à cause des nombreuses jointures nécessaires pour accéder aux données. Ce modèle est utilisé dans certains cas dans notre projet où l'optimisation du stockage est prioritaire.

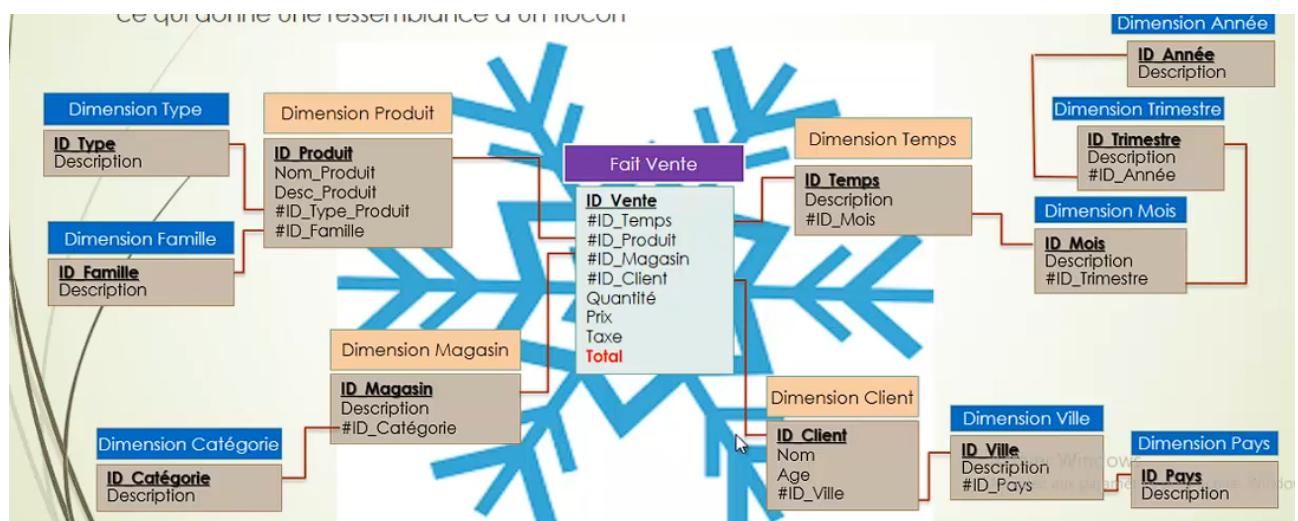


FIGURE 3.2 – Modélisation en flocon de neige

3.3.3 Modèle en constellation

Le modèle en constellation, qui inclut plusieurs tables de faits partageant des dimensions communes, est aussi utilisé dans notre projet pour gérer des analyses complexes. Par exemple, il permet de relier les données des ventes, des équipements médicaux et des clients dans des analyses croisées avancées.

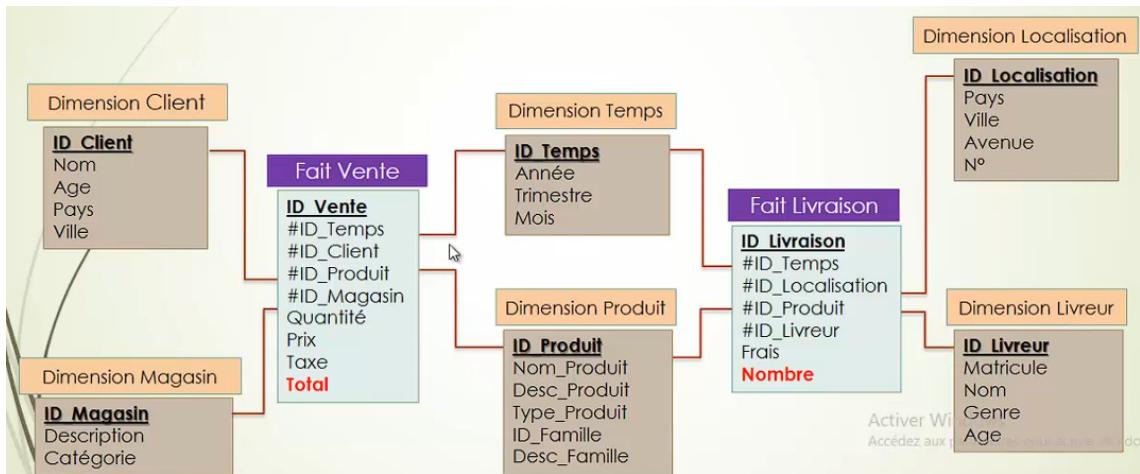


FIGURE 3.3 – Modèle en constellation

3.3.4 Comparaison entre les trois modèles

Le tableau 3.1 présente une comparaison entre les modèles logiques de données utilisés dans notre projet.

Critère	Modèle en étoile	Modèle en flocon	Modèle en constellation
Structure	Table de faits centrale avec dimensions	Dimensions normalisées (décomposées)	Plusieurs tables de faits avec dimensions partagées
Simplicité	Simple et intuitif	Plus complexe avec jointures multiples	Complexé, adapté aux scénarios multi-faits
Redondance	Modérément redondant	Réduction de la redondance	Variable selon la conception
Performance	Haute performance pour requêtes simples	Performance réduite à cause des jointures	Variable, dépend de la complexité des requêtes
Maintenance	Facile à maintenir	Plus difficile à cause de la complexité	Complexé, nécessite une gestion rigoureuse

TABLE 3.1 – Tableau comparatif entre les trois modèles logiques de data warehouse

3.3.5 Approche Kimball vs Inmon :

Les approches Kimball et Inmon sont deux méthodologies bien connues dans la conception des entrepôts de données (data warehouse). Elles se différencient par leur manière d'aborder l'architecture des données.

3.3.5.1 Approche Kimball (Bottom-Up)

L'approche Kimball, aussi appelée approche bottom-up (de bas en haut), se concentre sur la création de data marts spécifiques à des domaines fonctionnels comme les ventes, les finances, etc., qui sont ensuite intégrés pour former un entrepôt de données global.

- Philosophie : Développer d'abord des data marts centrés sur des processus métiers individuels, puis les intégrer dans un entrepôt global.
- Modèle de données : L'approche Kimball favorise l'utilisation des modèles en étoile (star schema) ou en flocon (snowflake schema) pour organiser les données dans des dimensions et des faits.
- Objectif : Fournir rapidement des solutions analytiques pour des besoins métier spécifiques, souvent avec une mise en œuvre plus rapide et plus centrée sur les besoins immédiats.
- Avantages : Mise en œuvre rapide pour des analyses spécifiques. Simplicité et facilité de compréhension pour les utilisateurs métier. Flexibilité dans l'ajout de nouveaux data marts.

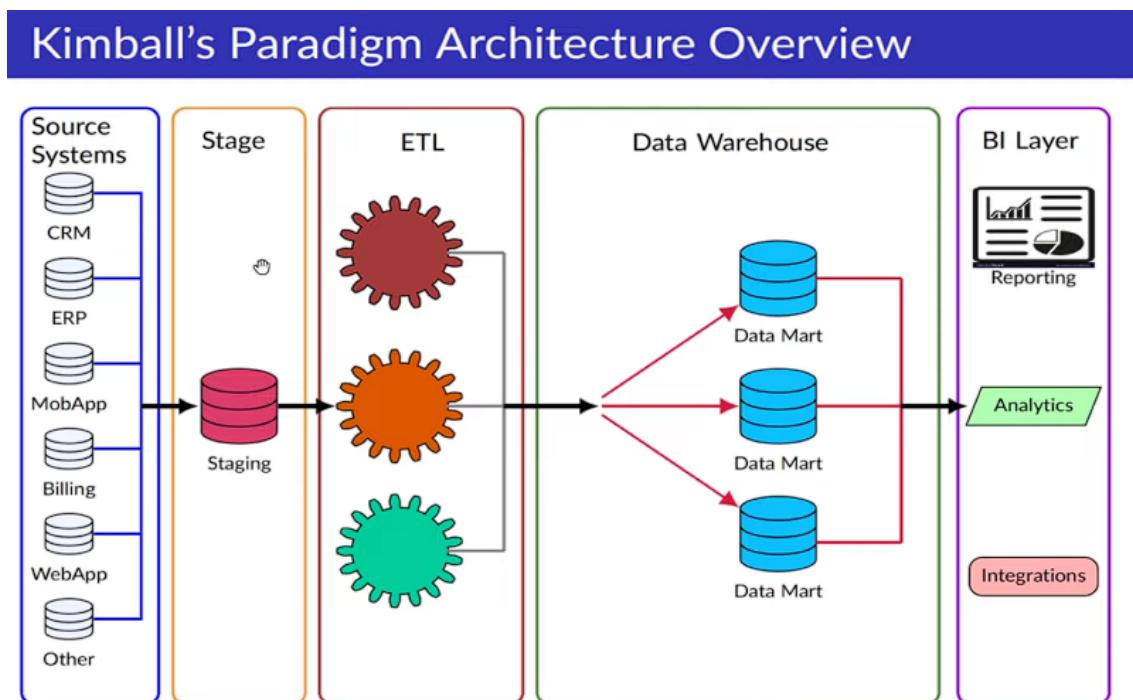


FIGURE 3.4 – Structure de Approche Kimball (Bottom-Up)

3.3.5.2 Approche Inmon (Top-Down)

L'approche Inmon, aussi appelée approche top-down (de haut en bas), se concentre sur la création d'un entrepôt de données centralisé dès le départ, qui contient des données intégrées et normalisées provenant de différentes sources opérationnelles. Les data marts sont créés à partir de cet entrepôt de données centralisé pour répondre aux besoins spécifiques.

- Philosophie : Créer un entrepôt de données unifié et intégré, suivi par la création de data marts à partir de cet entrepôt.
- Modèle de données : Utilise un modèle relationnel normalisé (troisième forme normale - 3NF) pour organiser les données dans l'entrepôt. Cela permet une vue plus intégrée et cohérente des données.
- Objectif : Construire une fondation de données robuste et à long terme, qui peut être utilisée pour une grande variété de besoins analytiques.

- Avantages : Données bien intégrées et cohérentes à travers l'organisation. Structure flexible pour répondre à des besoins d'analyse à long terme. Meilleure gestion des données complexes et intégration.

Inmon's Paradigm Architecture Overview

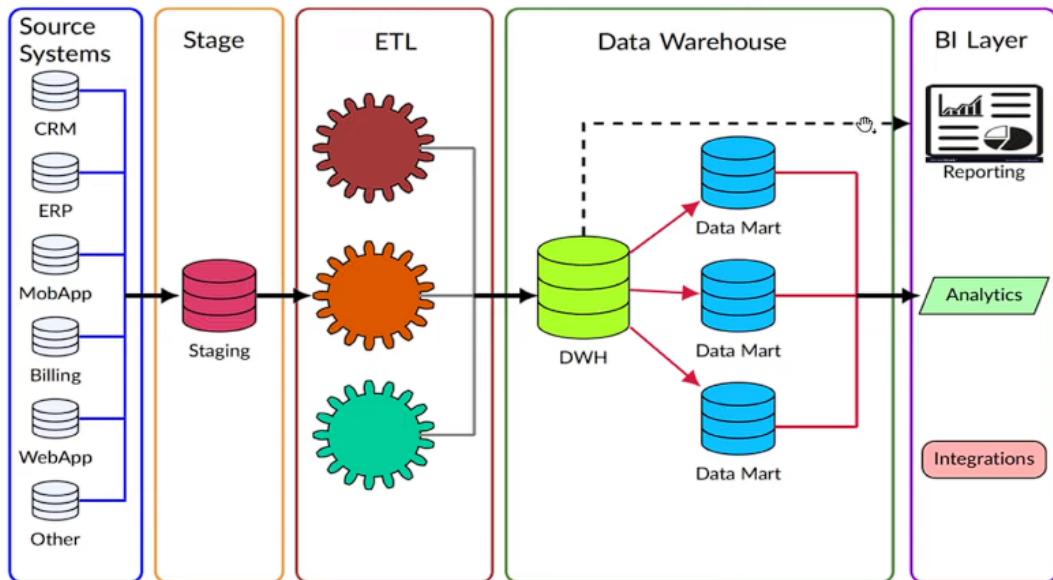


FIGURE 3.5 – Structure de Approche Inmon (Top-Down)

Le tableau suivant présente une comparaison entre l'approche Kimball et l'approche Inmon.

Caractéristique	Approche Kimball	Approche Inmon
Méthodologie	Bottom-up	Top-down
Point de départ	Data marts	Entrepôt de données centralisé
Modèle de données	Modèles étoile ou flocon	Modèle relationnel normalisé
Déploiement	Rapide et itératif	Plus long à mettre en place
Focus sur l'analyse	Processus métier spécifique	Vue d'ensemble des données
Intégration des données	Intégration limitée entre data marts	Intégration complète dans un entrepôt
Complexité initiale	Moins complexe, simple à comprendre	Plus complexe, nécessite plus de planification

TABLE 3.2 – Comparaison entre l'approche Kimball et l'approche Inmon

3.3.6 Identification des dimensions et de la table de faits

3.3.6.1 La dimension Clients

Le tableau 3.2 présente les détails de la dimension Clients.

Champs	Type	Description
Client-PK	Integer	Identifiant unique du client
CodeClient	Varchar(50)	Code client unique
Societe	Varchar(50)	Nom complet du client
Adresse	Varchar(100)	Adresse du client
MatriculeFiscale	Varchar(50)	Matricule fiscale du client
Email	Varchar(50)	Email du client
Telephone	Varchar(50)	Téléphone du client

TABLE 3.3 – Détails de la dimension Clients

3.3.6.2 La dimension Fournisseurs

Le tableau 3.3 présente les détails de la dimension Fournisseurs.

Champs	Type	Description
Fournisseur-PK	Integer	Identifiant unique du fournisseur
CodeFournisseur	Varchar(50)	Code fournisseur unique
RaisonSociale	Varchar(50)	Nom du fournisseur
Adresse	Varchar(100)	Adresse du fournisseur
Telephone	Varchar(50)	Téléphone du fournisseur
Email	Varchar(50)	Email du fournisseur

TABLE 3.4 – Détails de la dimension Fournisseurs

3.3.6.3 La dimension Régions

Le tableau 3.4 présente les détails de la dimension Régions.

Champs	Type	Description
Region-PK	Integer	Identifiant unique de la région
RegionCode	Integer	Code unique de la région
Ville	Varchar(50)	Nom de la Ville
Pays	Varchar(50)	Pays de la région

TABLE 3.5 – Détails de la dimension Régions

3.3.6.4 La dimension Articles

Le tableau 3.5 présente les détails de la dimension Articles.

Champs	Type	Description
Article-PK	Integer	Identifiant unique de l'article
CodeArticle	Varchar(50)	Code unique de l'article
Designation	Varchar(50)	Nom de l'article
Collection	Varchar(50)	Collection de l'article
Famille	Varchar(50)	Famille de l'article
Unite	Varchar(50)	Unité de l'article

TABLE 3.6 – Détails de la dimension Articles

3.3.6.5 La dimension Dates

Le tableau 3.6 présente les détails de la dimension Dates.

Champs	Type	Description
Date-PK	Integer	Identifiant unique de la date
Date	Date	Date complète (yyyy-MM-dd)
Day	Integer	Jour du mois (01 à 31)
Month	Integer	Mois de l'année (01 à 12)
Year	Integer	Année de la date
Quarter	Integer	Trimestre de l'année (1 à 4)
DayOfWeek	Varchar(50)	Jour de la semaine
MonthName	Varchar(50)	Mois de l'année (nom)

TABLE 3.7 – Détails de la dimension Dates

3.3.6.6 Table des faits – FaitsVentes

Le tableau 3.7 présente les détails de la table des faits FaitsVentes, qui regroupe les données liées aux ventes chez P2MS.

Champs	Type	Description
Client-fk	Integer	Clé étrangère liée à la dimension Clients
Fournisseur-fk	Integer	Clé étrangère liée à la dimension Fournisseurs
Article-fk	Integer	Clé étrangère liée à la dimension Articles
Region-fk	Integer	Clé étrangère liée à la dimension Régions
Date-fk	Integer	Clé étrangère liée à la dimension Dates
Quantite	Integer	Quantité d'articles vendus
PrixAchat	Float	Prix d'achat des articles
PrixVente	Float	Prix de vente des articles
TVA	Float	Taxe sur la valeur ajoutée
montant-total	Float	Montant total des ventes
MontantTTC	Float	Montant TTC (toutes taxes comprises) des ventes

TABLE 3.8 – Détails de la table des faits FaitsVentes

3.4 Conclusion

Dans ce chapitre, nous avons détaillé la modélisation dimensionnelle du projet chez P2MS en utilisant un **modèle en étoile** pour l'entrepôt de données. Ce choix simplifie l'exécution des requêtes et optimise les analyses liées aux ventes. Nous avons présenté les dimensions principales (Clients, Fournisseurs, Régions, Articles, et Dates) ainsi que la table de faits FaitsVentes. Dans le prochain chapitre, nous aborderons la mise en œuvre technique du projet.

La figure 3.4 Modele de DataWarehouse :

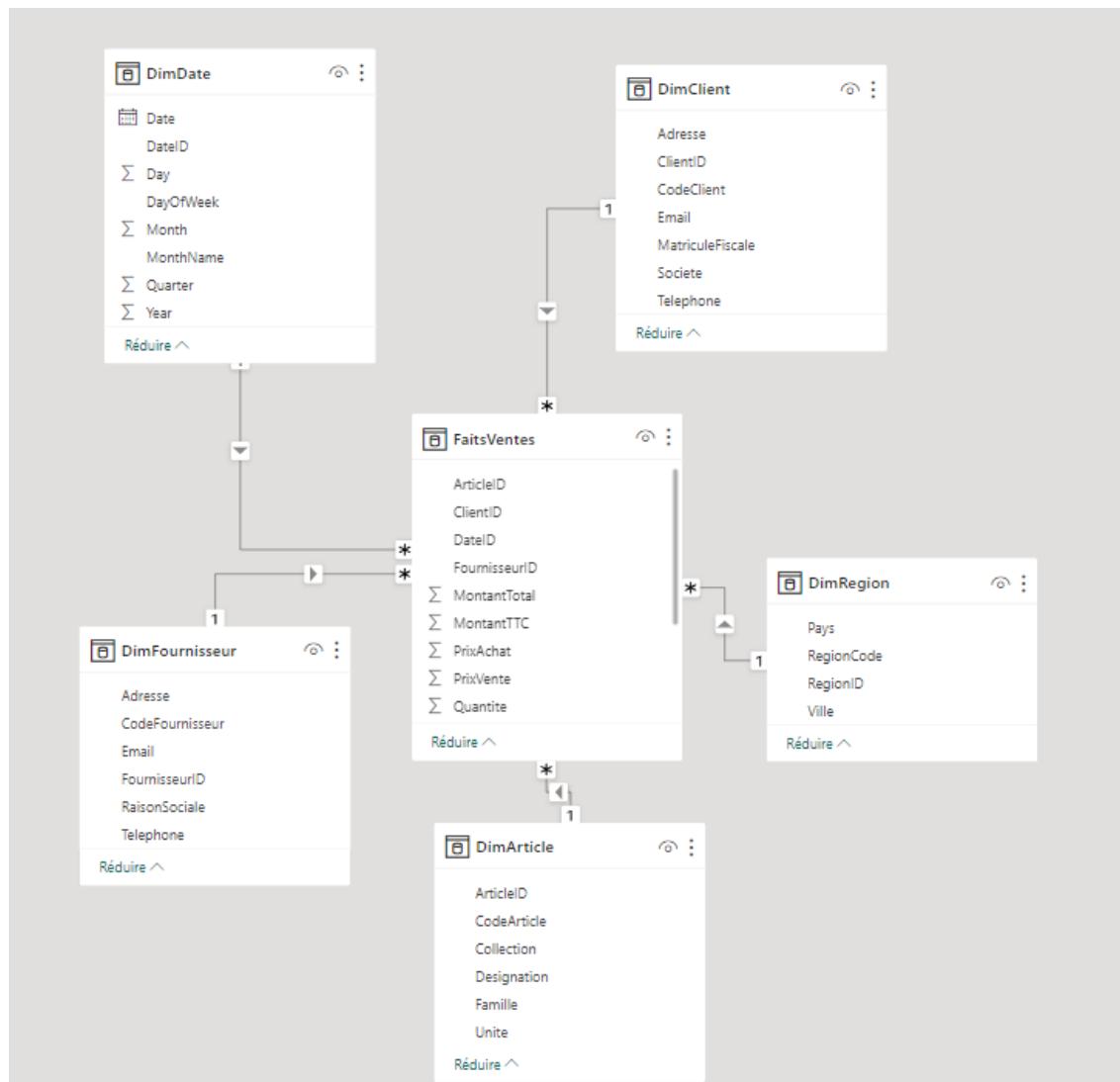


FIGURE 3.6 – Modele de DataWarehouse

CHAPITRE 4

Mise en œuvre et réalisation

4.1 Introduction

Ce dernier chapitre est consacré à la présentation du travail réalisé, en abordant l'implémentation de la solution ainsi que ses différentes phases. Enfin, le déploiement de la solution sera également présenté.

4.2 Implémentation

Après la phase du modèle conceptuel de notre entrepôt de données, on va pouvoir commencer une nouvelle étape intitulée l'implémentation. Cette dernière est constituée de plusieurs phases d'implémentation et de mise en œuvre que nous allons détailler.

4.2.1 Analyse Exploratoire des Données (EDA)

4.2.1.1 Fichier brands-cleaned

Ce graphique en barres montre la répartition des articles dans différentes familles. Il permet d'identifier les familles d'articles les plus courantes dans la base de données, ce qui peut être utile pour comprendre la composition du catalogue de produits.

La figure 4.1 représente la Distribution des familles d'articles :

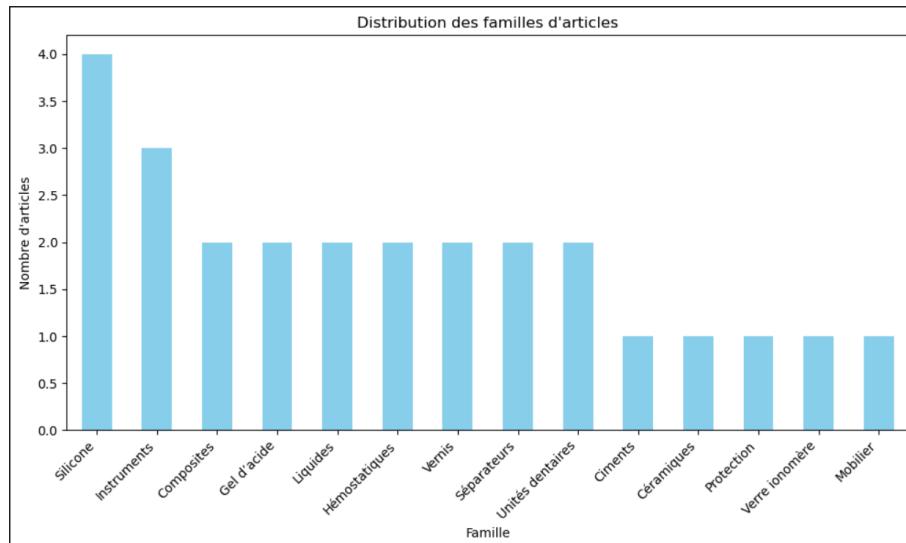


FIGURE 4.1 – Distribution des familles d'articles

Cet histogramme montre la distribution des prix de vente des articles. Il permet de visualiser la répartition des prix dans différentes tranches, mettant en évidence si les articles sont concentrés dans une gamme de prix particulière ou s'ils sont répartis de manière uniforme.

La figure 4.2 Distribution des prix de vente des articles :

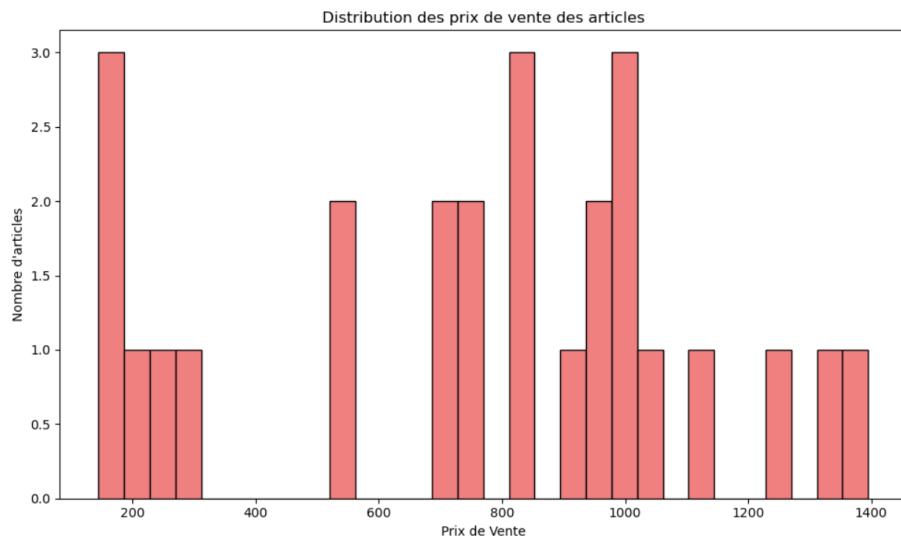


FIGURE 4.2 – Distribution des prix de vente des articles

Ce graphique circulaire (pie chart) illustre la répartition des articles par fournisseur. Chaque segment du cercle représente un fournisseur, et la taille de chaque segment est proportionnelle au nombre d'articles fournis par ce dernier. Les pourcentages affichés sur chaque portion indiquent la part relative des articles provenant de chaque fournisseur par rapport à l'ensemble des articles.

La figure 4.3 Répartition des articles par fournisseur :

Ce graphique en ligne représente l'évolution des prix de vente moyens pour chaque collection d'articles. En regroupant les données par collection, le prix de vente moyen est calculé, puis trié par ordre croissant pour permettre une vue plus claire des variations entre les collections. Chaque point sur la ligne représente la moyenne des prix de vente pour une collection spécifique, tandis que la

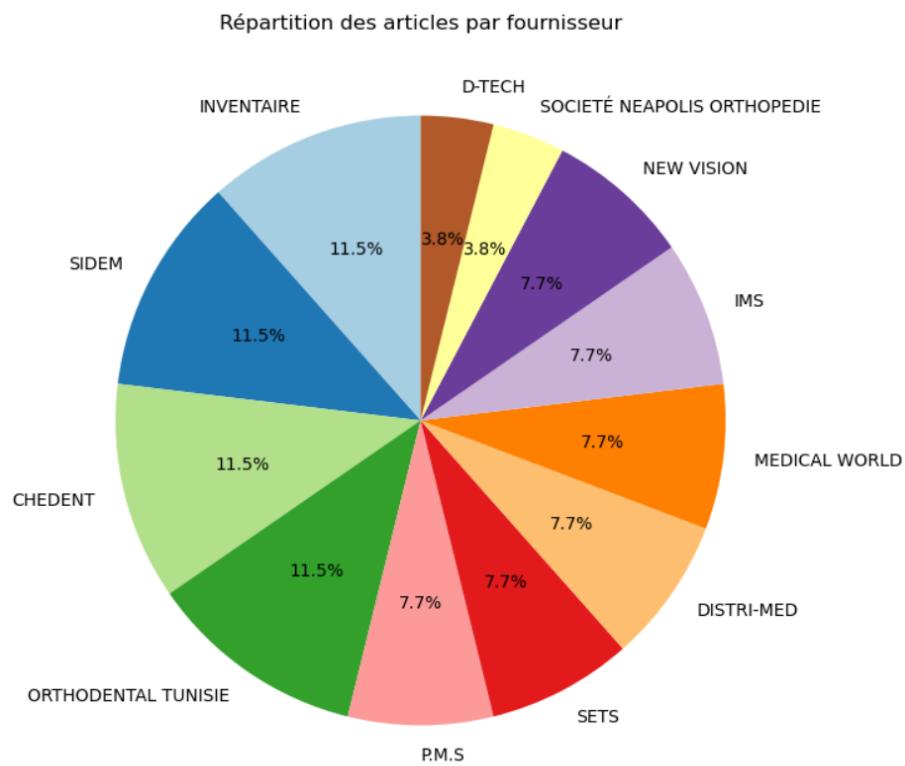


FIGURE 4.3 – Répartition des articles par fournisseur

ligne trace l'évolution des prix à travers les différentes collections. Cela permet d'identifier quelles collections ont des prix de vente plus élevés ou plus bas par rapport aux autres.

La figure 4.4 Evolution des prix de vente moyens par Collection :

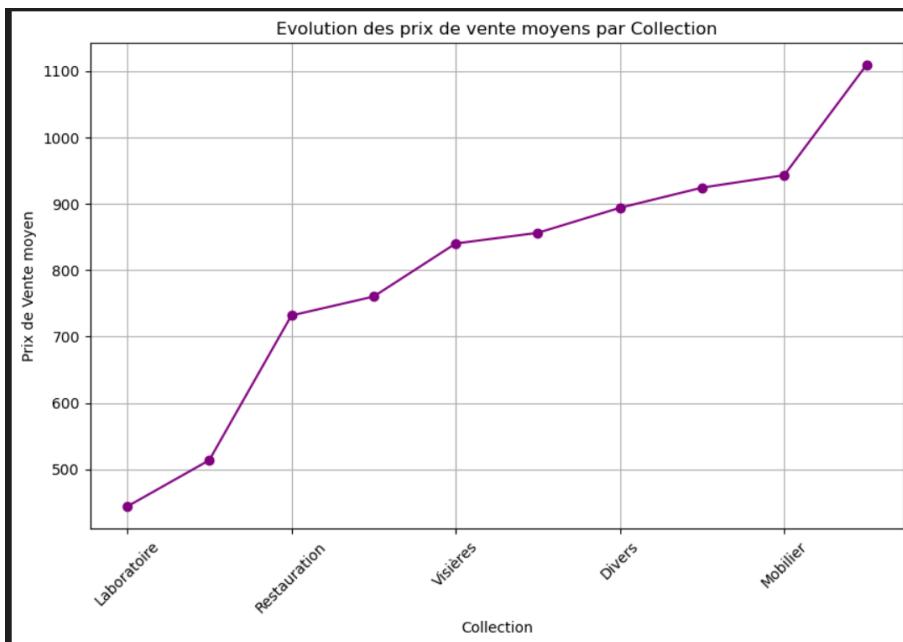


FIGURE 4.4 – Evolution des prix de vente moyens par Collection

4.2.2 Phase d’Alimentation ETL (Extract, Transform, Load)

4.2.2.1 Définition Phase ETL

Cette phase implique l’extraction des données depuis diverses sources, leur transformation selon les besoins analytiques (nettoyage, enrichissement, agrégation) et leur chargement dans une base de données ou un entrepôt de données. L’objectif est de préparer les données pour l’analyse et le reporting.

4.2.2.2 Alimentation de la StagingArea

La staging area représente une étape clé dans la construction de l’entrepôt de données (data warehouse). Elle joue un rôle essentiel dans le processus d’ETL (Extraction, Transformation, Chargement) en fournissant un espace temporaire où les données provenant de diverses sources sont collectées, nettoyées et transformées avant d’être chargées dans le data warehouse final. Dans le cadre de notre projet P2MS, la staging area a permis de centraliser les informations issues des systèmes transactionnels, en particulier des données de vente, d’articles, de clients et de fournisseurs. Cette étape a facilité le nettoyage des données, la gestion des doublons et la standardisation des formats pour garantir la qualité des informations qui alimentent le data warehouse.

La figure 4.5 Staging Area :

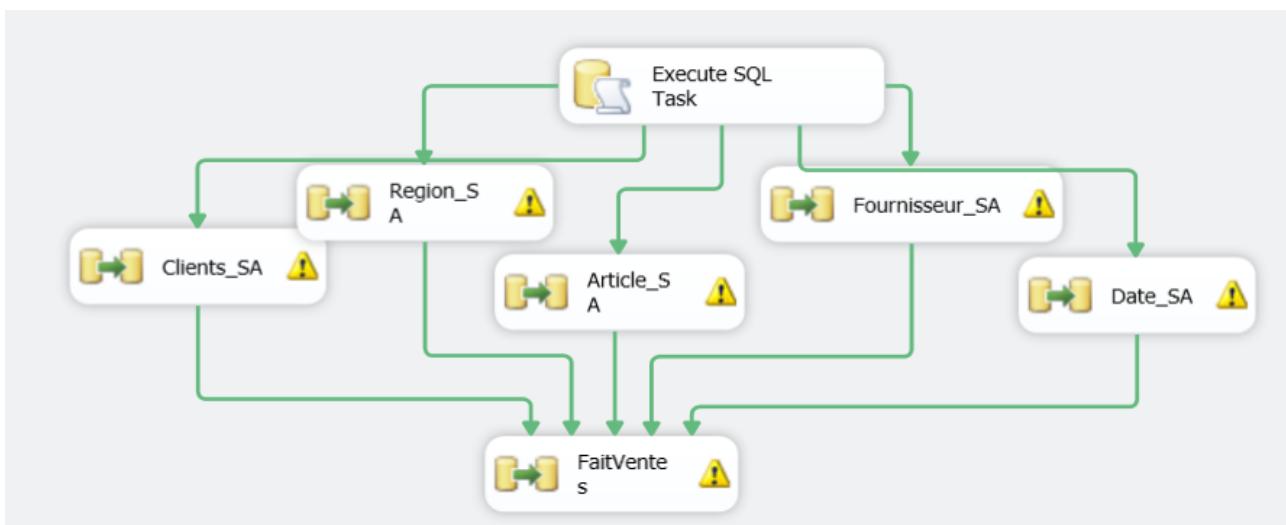


FIGURE 4.5 – Staging Area

Dans le cadre de la modélisation dimensionnelle d’un entrepôt de données, l’Identity Specification, ou clé technique auto-incrémentée, joue un rôle crucial dans la gestion des enregistrements au sein des tables de dimensions et de faits. Il s’agit d’une colonne, généralement un identifiant numérique, dont la valeur est automatiquement générée et incrémentée par le système à chaque insertion de nouvel enregistrement. Cela permet d’assurer l’unicité de chaque enregistrement, facilitant ainsi l’intégration des données dans le data warehouse.

L’utilisation de clés techniques auto-incrémentées présente plusieurs avantages :

- Unicité et intégrité des données : Chaque ligne dans la table a un identifiant unique, ce qui évite les doublons et garantit la consistance des données.
- Performance des requêtes : Les requêtes SQL utilisant des clés numériques (entiers) comme identifiants sont plus rapides que celles utilisant des clés alphanumériques complexes.

- Simplification des relations : Les clés techniques permettent de facilement relier les tables de faits aux tables de dimensions dans un modèle en étoile, assurant une gestion simplifiée des relations entre les tables.
- Dans notre projet P2MS, l'Identity Specification a été appliquée à toutes les tables de dimensions (DimClient, DimArticle, DimFournisseur, DimDate, DimRegion), garantissant une identification fiable des enregistrements tout au long du processus ETL. Chaque clé technique est générée automatiquement lors de l'insertion d'un nouvel enregistrement, facilitant l'intégrité et la traçabilité des données.

La figure 4.6 Structure de la dimension DimClient :

Column Name	Data Type	Allow Nulls
ClientID	int	<input type="checkbox"/>
CodeClient	nvarchar(50)	<input checked="" type="checkbox"/>
Societe	nvarchar(50)	<input checked="" type="checkbox"/>
Adresse	nvarchar(50)	<input checked="" type="checkbox"/>
MatriculeFiscale	nvarchar(50)	<input checked="" type="checkbox"/>
Email	nvarchar(50)	<input checked="" type="checkbox"/>
Telephone	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
Identity Specification	

FIGURE 4.6 – Structure de la dimension DimClient

Réinitialisation des identifiants (clés techniques) : Après la suppression des enregistrements, la commande DBCC CHECKIDENT avec l'option RESEED est utilisée pour réinitialiser l'auto-incrémentation des clés techniques à 0. Cela garantit que les nouveaux enregistrements insérés dans les tables recommenceront à partir de l'ID 1. Par exemple, après l'exécution de cette commande sur [DimArticle], tout nouvel article ajouté à la table aura un identifiant commençant à 1, évitant ainsi des trous dans la séquence des clés.

La figure 4.7 Reset Clé technique :

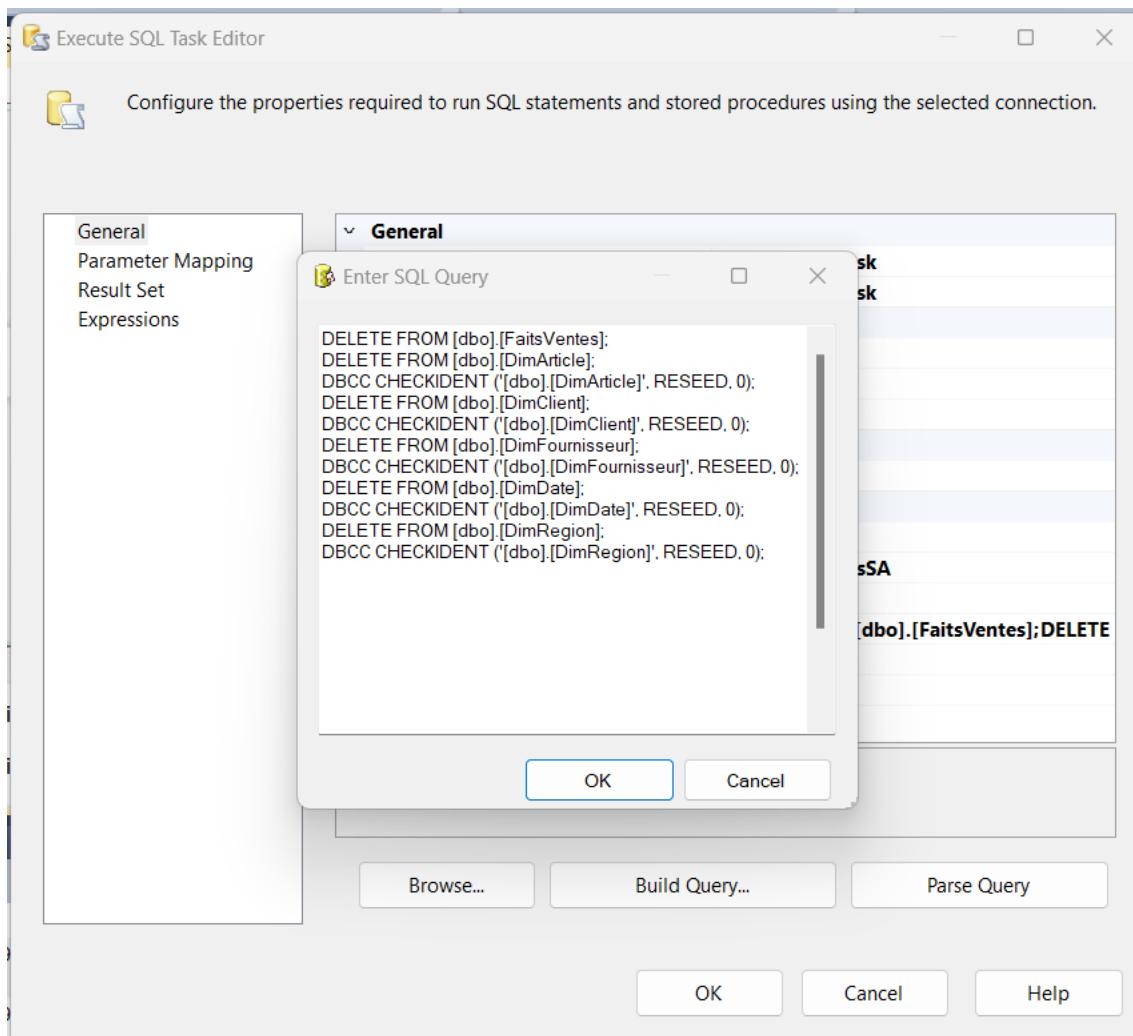


FIGURE 4.7 – Reset Clé technique

Excel Source : Ce composant permet de lire les données provenant de fichiers Excel, qui peuvent être une source importante de données opérationnelles, notamment pour les rapports externes ou les données de fournisseurs. Il extrait les informations depuis les feuilles Excel avant de les intégrer dans le pipeline de traitement ETL.

OLE DB Destination : Ce composant est utilisé pour charger les données dans une base de données relationnelle après leur transformation. Dans notre projet, il permet d'envoyer les données nettoyées vers la table de faits ou de dimensions de l'entrepôt de données, assurant une intégrité et une cohérence des informations.

La figure 4.8 Composants Utilisés dans SA :

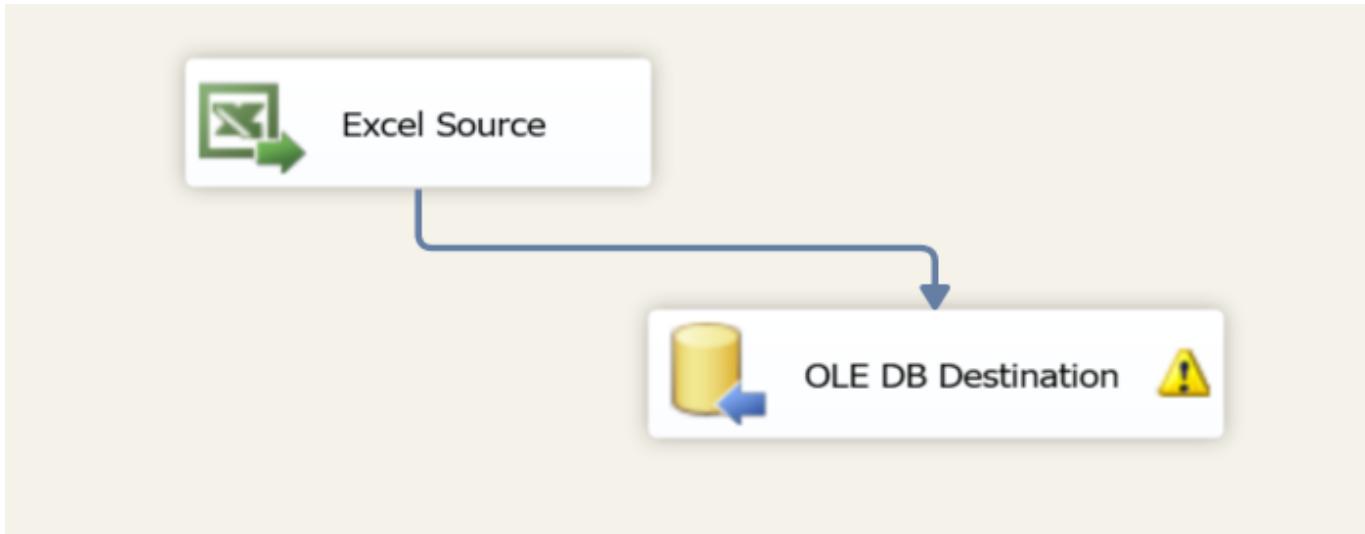


FIGURE 4.8 – Composants Utilisés dans SA

La figure 4.9 Alimentation de la dimension DimClient :

SQLQuery1.sql - ISMAEL.master (sa (53))

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [ClientID]
      ,[CodeClient]
      ,[Societe]
      ,[Adresse]
      ,[MatriculeFiscale]
      ,[Email]
      ,[Telephone]
  FROM [p2msDW].[dbo].[DimClient]
  
```

Results

ClientID	CodeCli...	Societe	Adresse	MatriculeFiscale	Email	Telephone
1	C0025	ZRELLI SLAHEDDIN	NABEUL	1936069 ABC 000	zrellislaheddin@example.com	23795607
2	CO777	ZRELLI AMENI	MENZAH 8	1695325 MPA 000	zrelliameni@example.com	25623021
3	C0815	ZRELI FEDIA	Adresse inconnue	5510468 ABC 000	zrelifedia@example.com	20677119
4	C0348	ZOUARI ABDLTF	MENZEL TMIM	6137392 ABC 000	zouariabdltf@example.com	29157996
5	C0439	ZOUAOUI MOHAMED ALI	Adresse inconnue	9093653 ABC 000	zouauimohamedalii@example.com	21836542
6	C0753	ZOUAGHI WAEL	BEJA AV.FARHAT HACHED	1467020 bpa 000	zouaghiwael@example.com	20003723
7	C1065	ZORIAT FADHLA	Adresse inconnue	5377436 ABC 000	zoriatfadhila@example.com	27774763
8	C1038	ZORGUI SIWAR	ZAGOUAN	1484402/K	zorguisiwar@gmail.com	27918544
9	C0263	ZORGUI SARRA	NABEUL	1108624 CPA 000	zorguisarra@example.com	22036893
10	C0545	ZOERI YOSRA	Adresse inconnue	4338153 ABC 000	zoeriyosra@example.com	25243344
11	C0809	ZNAYDI MOHAMED	KLIBIA	1577247 GAP 000	znaydimohamed@example.com	20608843
12	C0119	ZMERLI FEDIA	Adresse inconnue	3642766 ABC 000	zmerlifedia@example.com	25699217

FIGURE 4.9 – Alimentation de la dimension DimClient

4.2.2.3 Alimentation de la DataWarehouse

La figure 4.10 Data Flow de la dimension DimClient :

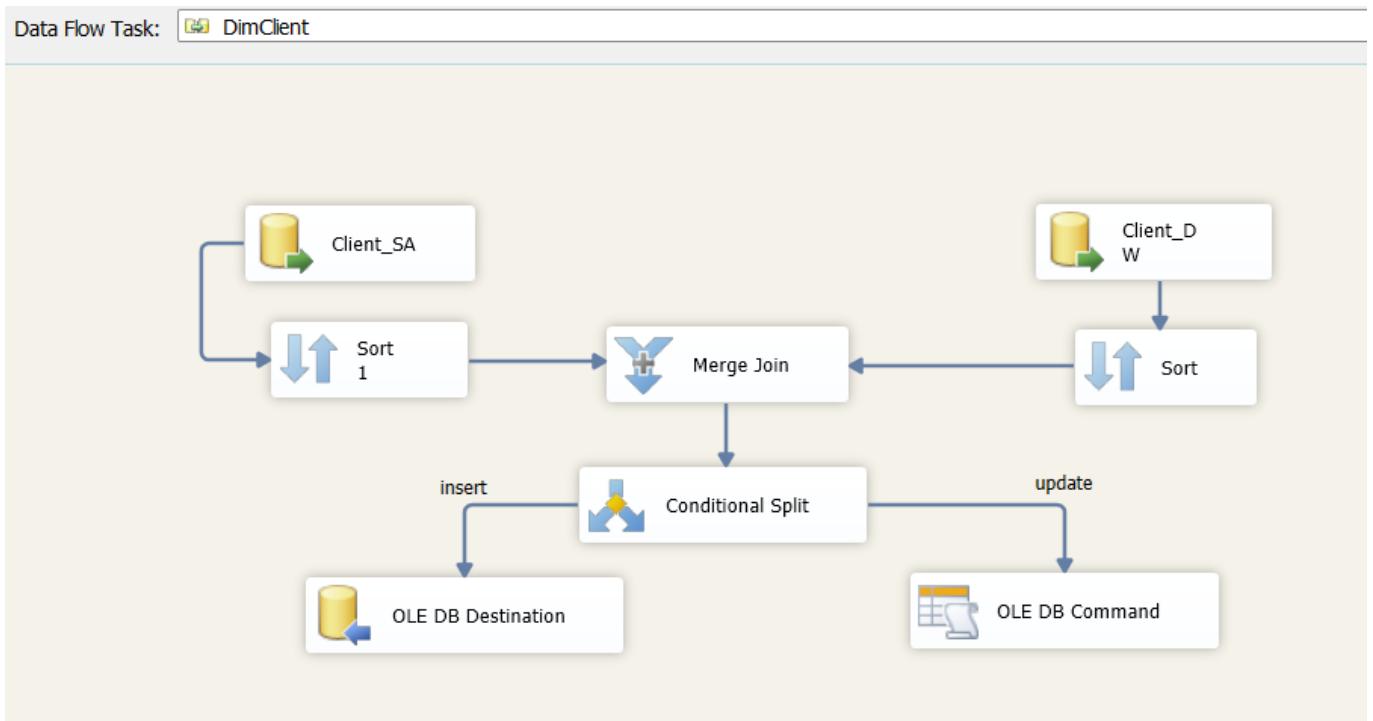


FIGURE 4.10 – Data Flow de la dimension DimClient

La figure 4.11 Tri avec clé métier :

The screenshot shows the "Sort Transformation Editor" window. The top panel displays the message: "Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged." Below this is a table titled "Available Input Columns" containing the following data:

	Name	Pass Through
<input type="checkbox"/>	ClientID	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	CodeClient	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Societe	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Adresse	<input checked="" type="checkbox"/>
<input type="checkbox"/>	MatriculeFiscale	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Email	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Telephone	<input checked="" type="checkbox"/>

Below this table is a detailed configuration table:

Input Column	Output Alias	Sort Type	Sort Order	Compa
CodeClient	CodeClient	ascending	1	

FIGURE 4.11 – Tri avec clé métier

Left Outer Join permet de gérer des scénarios d'insertion et de mise à jour de manière efficace, en vérifiant d'abord si un enregistrement existe déjà (pour mise à jour) ou s'il est absent (pour ajout). C'est une étape essentielle dans les processus d'alimentation de l'entrepôt de données (ETL), en particulier lors de l'intégration de nouvelles données dans des tables de dimensions ou de faits.

La figure 4.12 Merge Join :

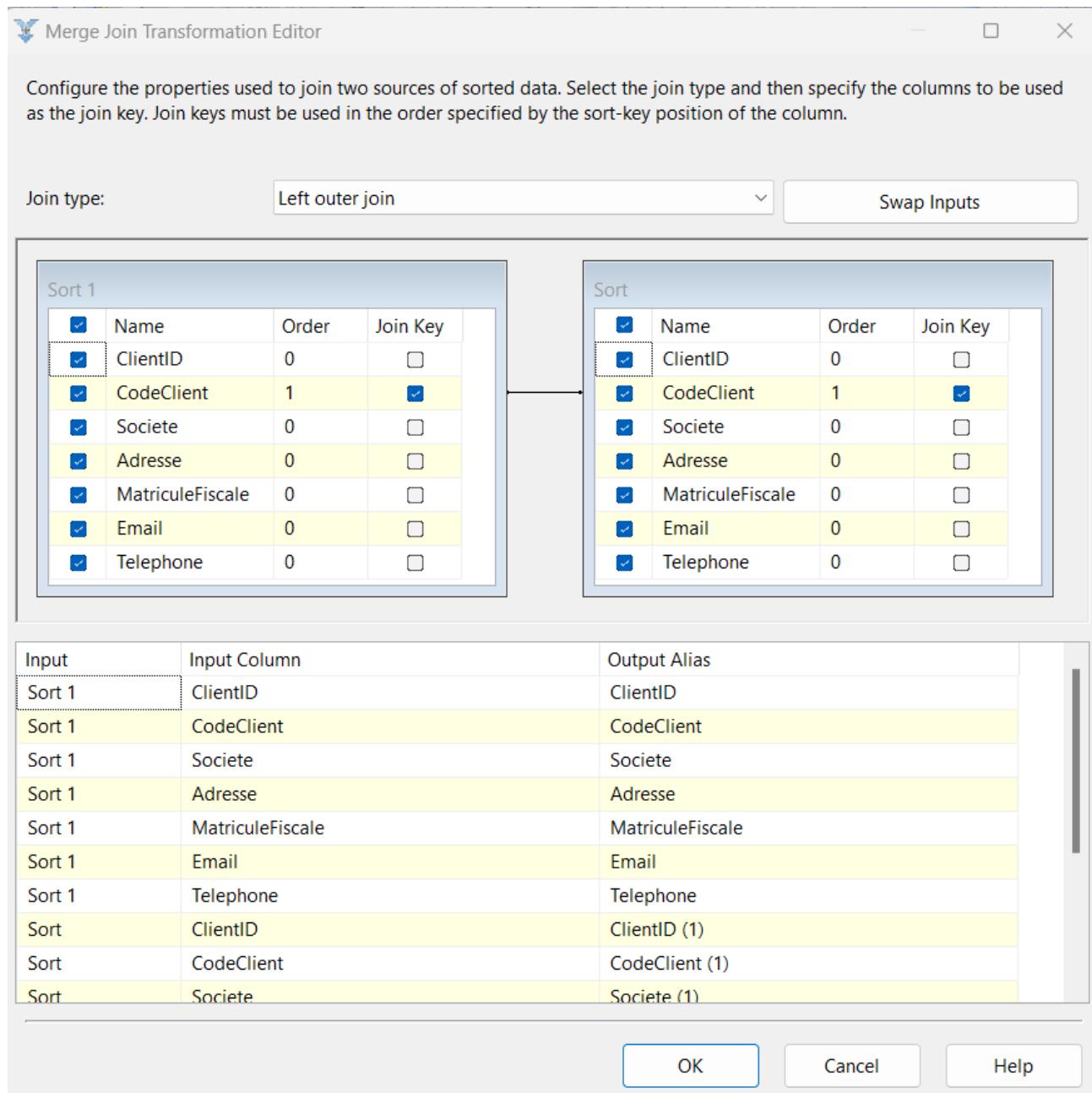


FIGURE 4.12 – Merge Join de Type Left Outer Join

Le composant Conditional Split dans les processus ETL (Extraction, Transformation, Chargement) permet de diviser les flux de données en fonction de conditions spécifiques, facilitant la gestion des décisions lors de l'alimentation des tables d'un entrepôt de données. Dans notre projet, il a été utilisé pour distinguer les cas où une ligne doit être insérée (ajout) ou mise à jour.

La figure 4.13 Conditionnal Split Insert ou Update :

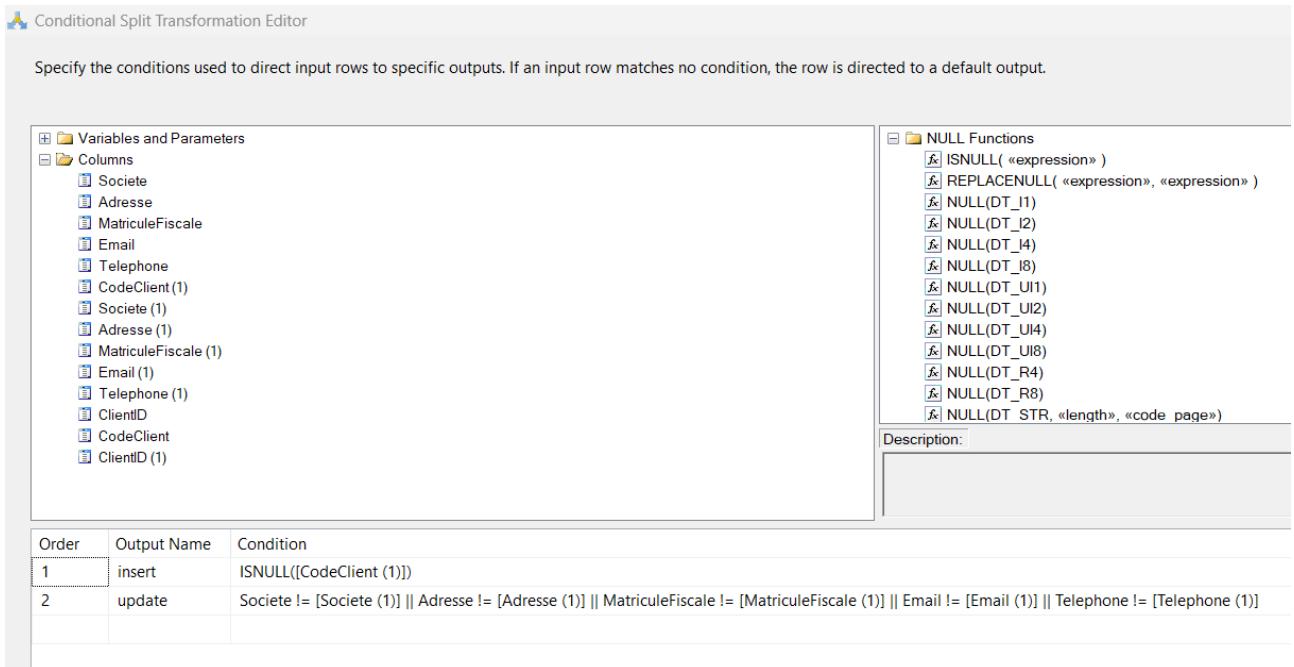


FIGURE 4.13 – Conditionnal Split Insert ou Update

La clé technique est Auto-Incrément. Dans ce cas, elle est ignorée dans le source. L'élément OLE DB Destination est utilisé pour insérer des données dans une table ou une vue dans une base de données via OLE DB. Il prend les données provenant du flux de données et les insère directement dans la destination spécifiée. Vous configurez la connexion à la base de données cible, sélectionnez la table ou la vue, et mappez les colonnes du flux de données aux colonnes de la destination. Cela permet un chargement direct et efficace des données, avec des options pour gérer les insertions, mises à jour ou combinaisons des deux selon les besoins du processus d'intégration des données.

La figure 4.14 Mapping dans OLE DB Destination :

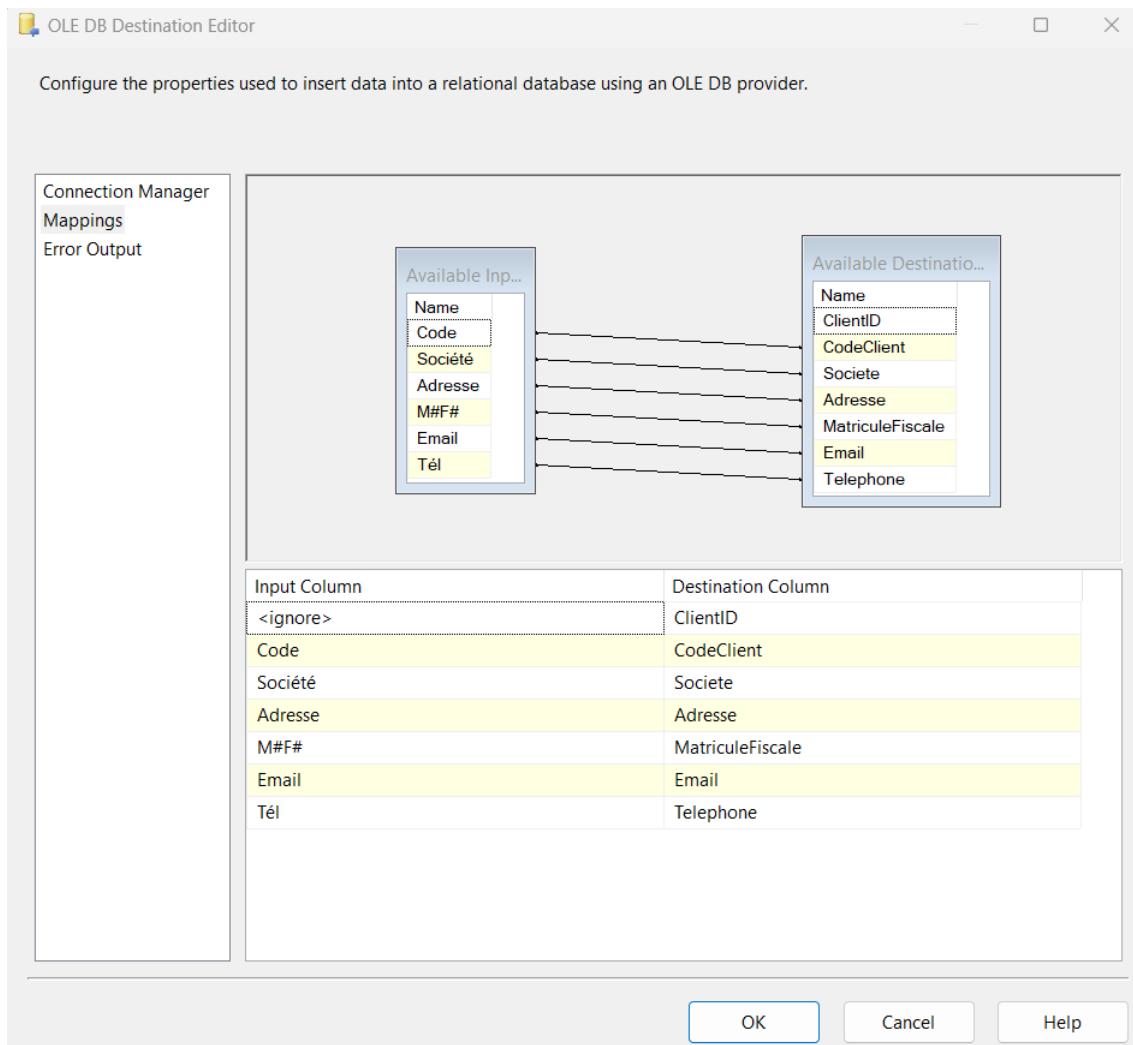


FIGURE 4.14 – Mapping OLE DB Destination

L'élément OLE DB Command dans SSIS est utilisé pour exécuter des commandes SQL personnalisées sur une source de données via OLE DB. Il permet de mettre à jour, insérer ou supprimer des lignes dans une base de données en fonction des valeurs en transit dans un flux de données. Par exemple, pour mettre à jour une ligne, vous pouvez utiliser une commande SQL de type UPDATE avec des paramètres qui correspondent aux colonnes du flux de données. Il est particulièrement utile lorsque vous devez effectuer des opérations de mise à jour conditionnelles ou personnalisées directement dans le flux de données. Requête de mise à jour :

La figure 4.15 Update Dimension :

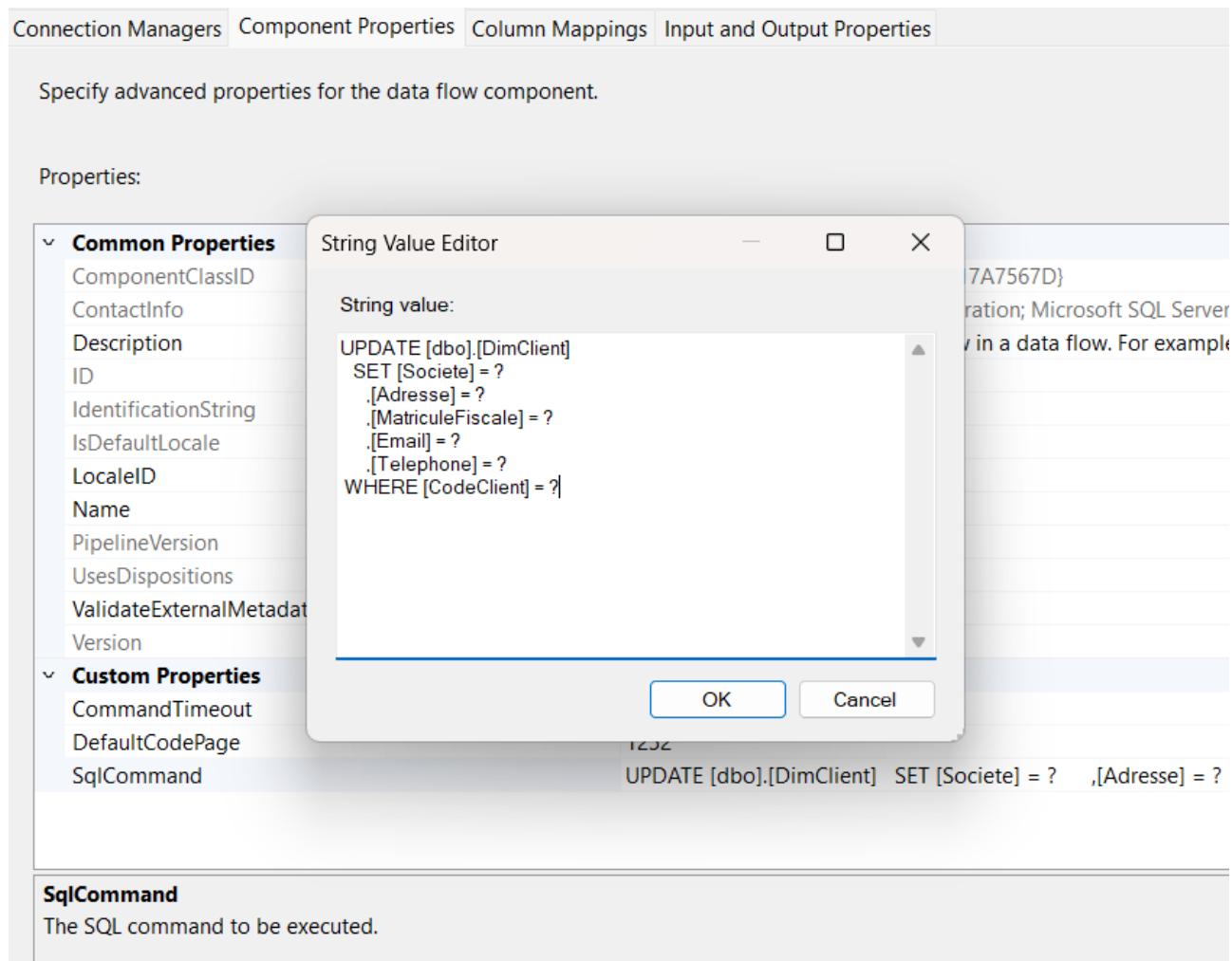


FIGURE 4.15 – Requête Update OLE DB Command

Alimentation de la table de fait :

- Le composant Lookup est utilisé pour effectuer une recherche de correspondance entre les données d'un flux de données et les données d'une table de référence ou d'une autre source de données. Dans le contexte où vous devez vérifier si une clé métier (clé étrangère) existe dans une table de référence (table mère) et retourner la clé technique correspondante, le composant Lookup fonctionne comme suit :
- Retourner la Clé Technique : Si une correspondance est trouvée, le Lookup retourne la clé technique associée (par exemple, l'identifiant unique ou la clé primaire) de la table de référence. Cette clé technique peut être utilisée pour les opérations ultérieures dans le flux de données.
- Configuration : Lors de la configuration du Lookup, vous spécifiez la table de référence comme source et définissez les colonnes de correspondance. Par exemple, vous mappez la clé métier du flux de données à la colonne correspondante dans la table de référence. Vous pouvez également configurer le Lookup pour gérer les valeurs non trouvées, soit en générant une erreur, soit en utilisant une valeur par défaut.

La figure 4.16 Fact Table : Requete de mise à jour :

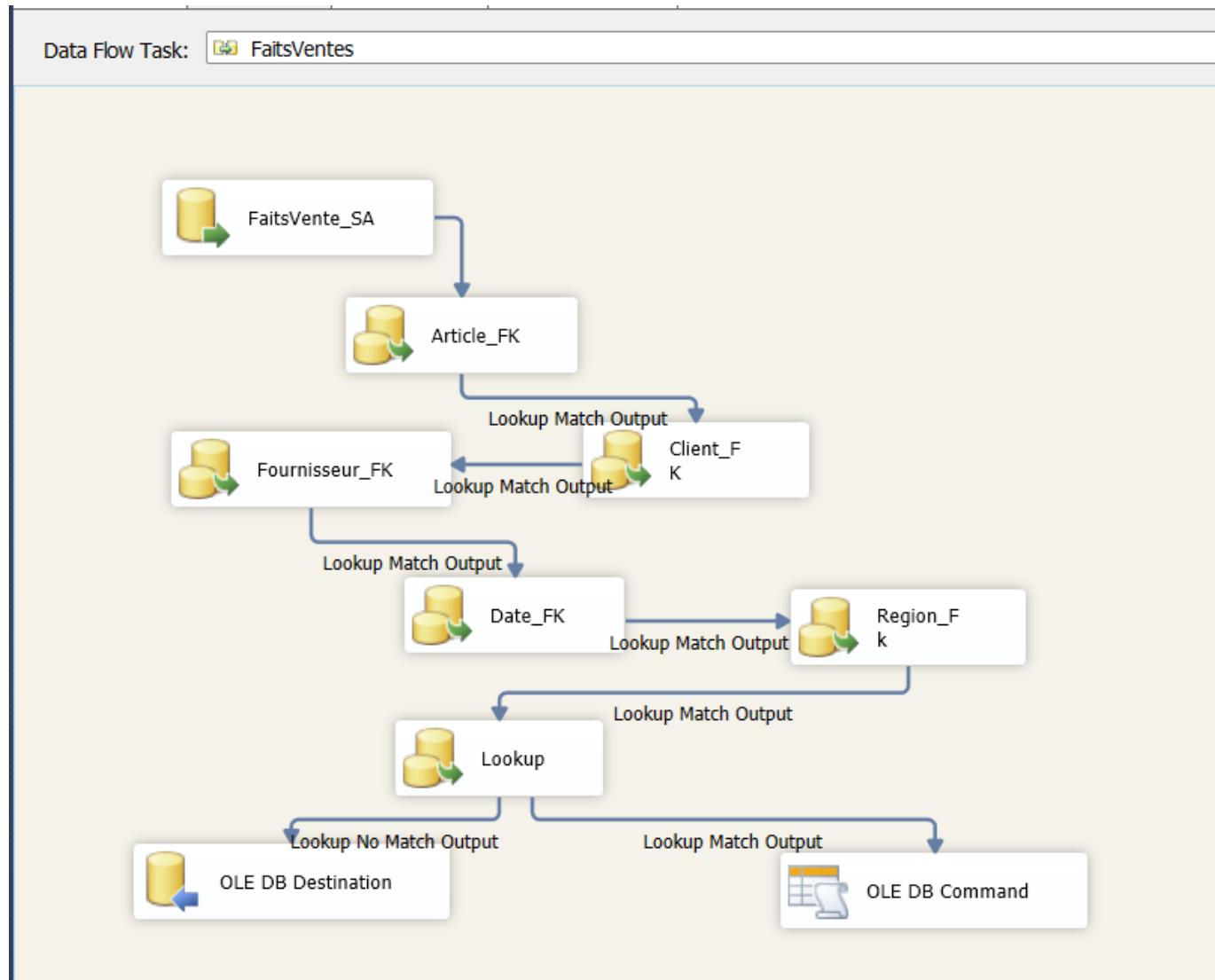


FIGURE 4.16 – Fact Table

La figure 4.17 Structure de Fact Table :

The screenshot shows the 'Object Explorer' pane with the path 'ISMAEL.p2msDW -> dbo.FaitsVentes'. The 'Script' tab is selected. Below it, the 'Table Designer' is open, showing the column properties for 'ArticleID'.

Table Structure:

Column Name	Data Type	Allow Nulls
ArticleID	int	<input type="checkbox"/>
ClientID	int	<input type="checkbox"/>
FournisseurID	int	<input type="checkbox"/>
DateID	int	<input type="checkbox"/>
RegionID	int	<input type="checkbox"/>
Quantite	int	<input checked="" type="checkbox"/>
PrixAchat	float	<input checked="" type="checkbox"/>
PrixVente	float	<input checked="" type="checkbox"/>
TVA	float	<input checked="" type="checkbox"/>
MontantTotal	float	<input checked="" type="checkbox"/>
MontantTTC	float	<input checked="" type="checkbox"/>

Column Properties for ArticleID:

Property	Value
Name	ArticleID
Allow Nulls	No
Data Type	int
Default Value or Binding	<database>.default

FIGURE 4.17 – Structure de Fact Table

La figure 4.18 Exemple de Verification de Article-FK :

Vérification de l'Existence : Le Lookup vérifie si la clé métier (par exemple, une clé étrangère dans le flux de données) existe dans la table de référence (table mère). Cela permet de s'assurer que les données référencées sont valides et présentes dans la table cible. :

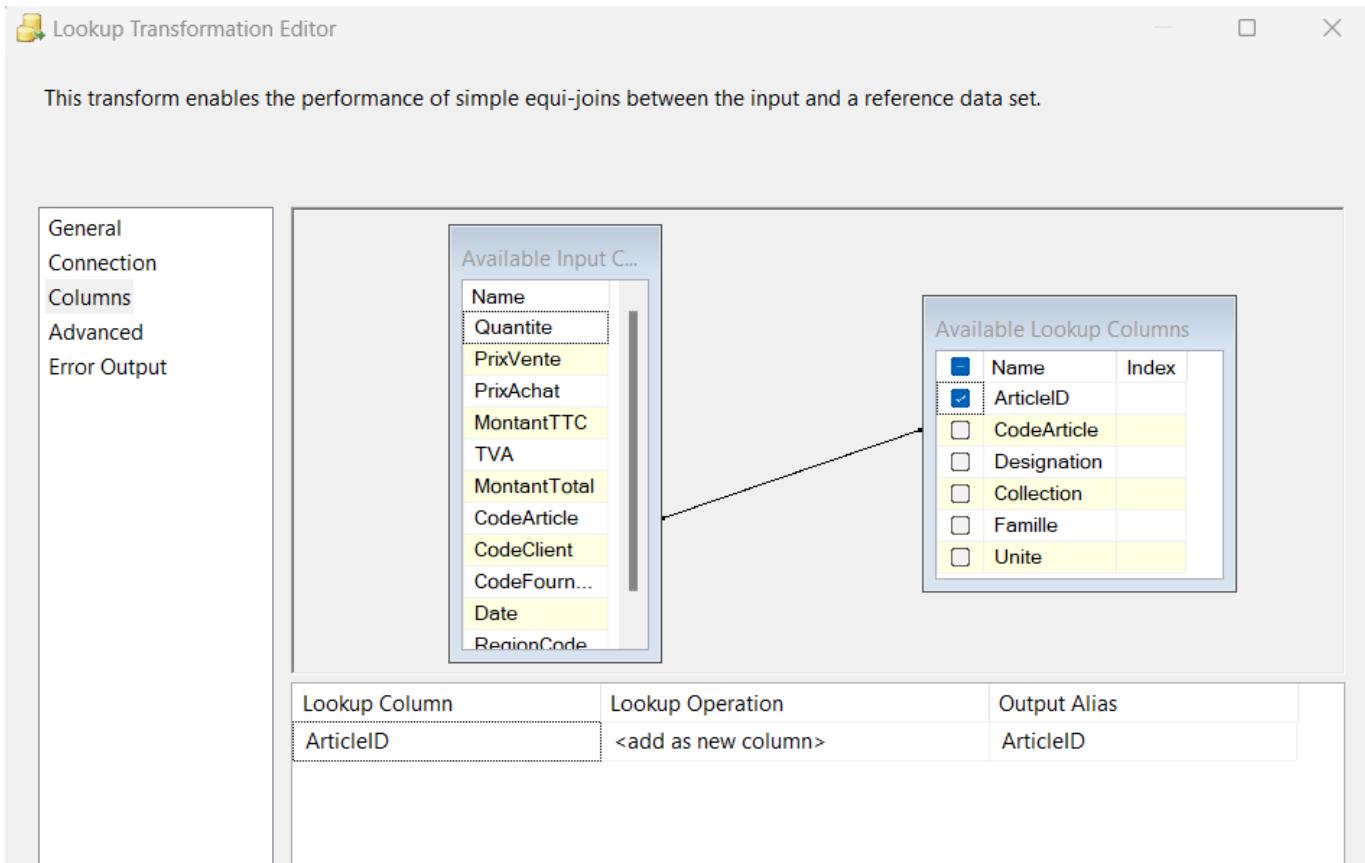


FIGURE 4.18 – Exemple de Verification de Article-FK

Apres verification de tous Foreign Key en doit verifier le clé primaire composé :

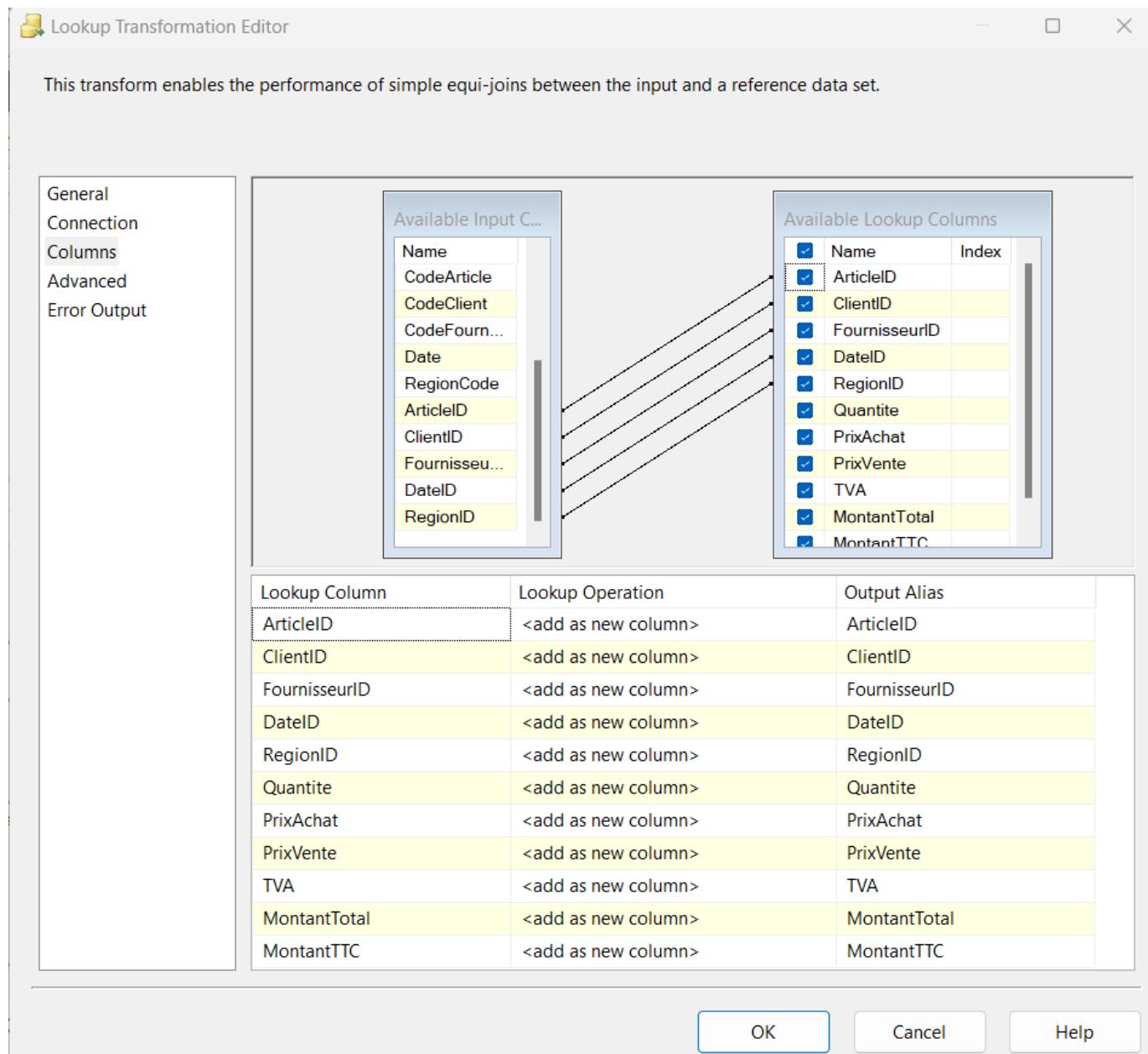


FIGURE 4.19 – Verification Clé Primaire composé

No Match Output Rôle : Cette sortie est utilisée pour renvoyer les lignes du flux de données principal pour lesquelles aucune correspondance n'a été trouvée dans la table de référence. Quand Utiliser : Si une clé ou une valeur dans le flux de données ne correspond à aucune clé ou valeur dans la table de référence, ces lignes sont envoyées à la sortie No Match Output. Utilisation Typique : Vous utilisez cette sortie pour gérer les cas où les données de votre flux principal ne sont pas présentes dans la table de référence. Par exemple, vous pourriez vouloir enregistrer ces lignes dans une table de log, les insérer dans une table d'erreurs, ou les traiter différemment selon la logique métier. Lookup No Match Output (insert) Fact Table :

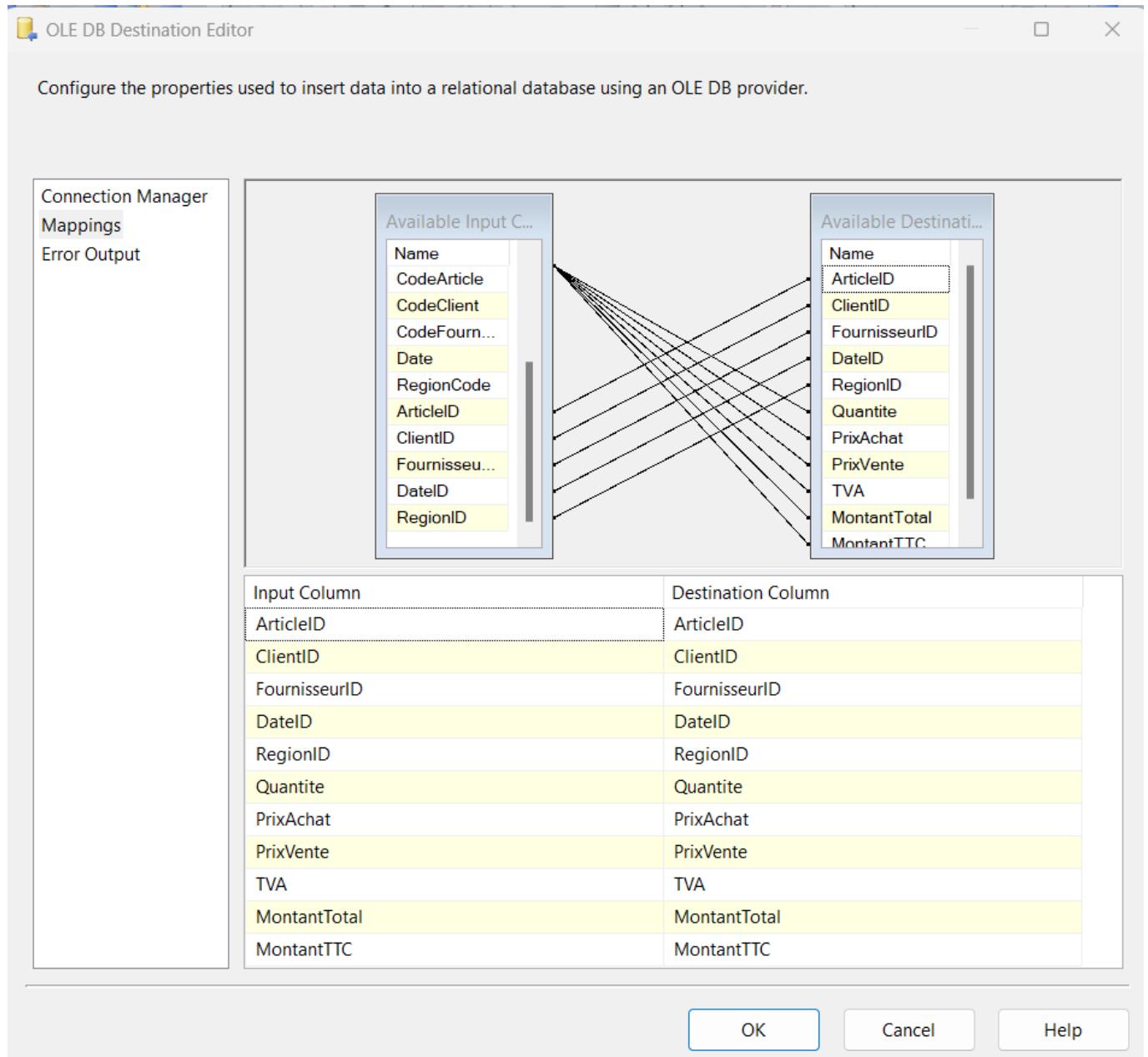


FIGURE 4.20 – Lookup No Match Output (insert)

Match Output Rôle : Cette sortie est utilisée pour renvoyer les lignes du flux de données principal pour lesquelles une correspondance a été trouvée dans la table de référence. Quand Utiliser : Si une clé ou une valeur dans le flux de données correspond à une clé ou une valeur dans la table de référence, les données associées à cette correspondance sont envoyées à la sortie Match Output. Utilisation Typique : Vous utilisez cette sortie lorsque vous avez besoin de traiter ou de transformer les lignes qui ont une correspondance valide dans la table de référence. Par exemple, si vous souhaitez

enrichir vos données avec des informations supplémentaires provenant de la table de référence, vous les récupérez ici. Lookup Match Output (update) Fact Table :

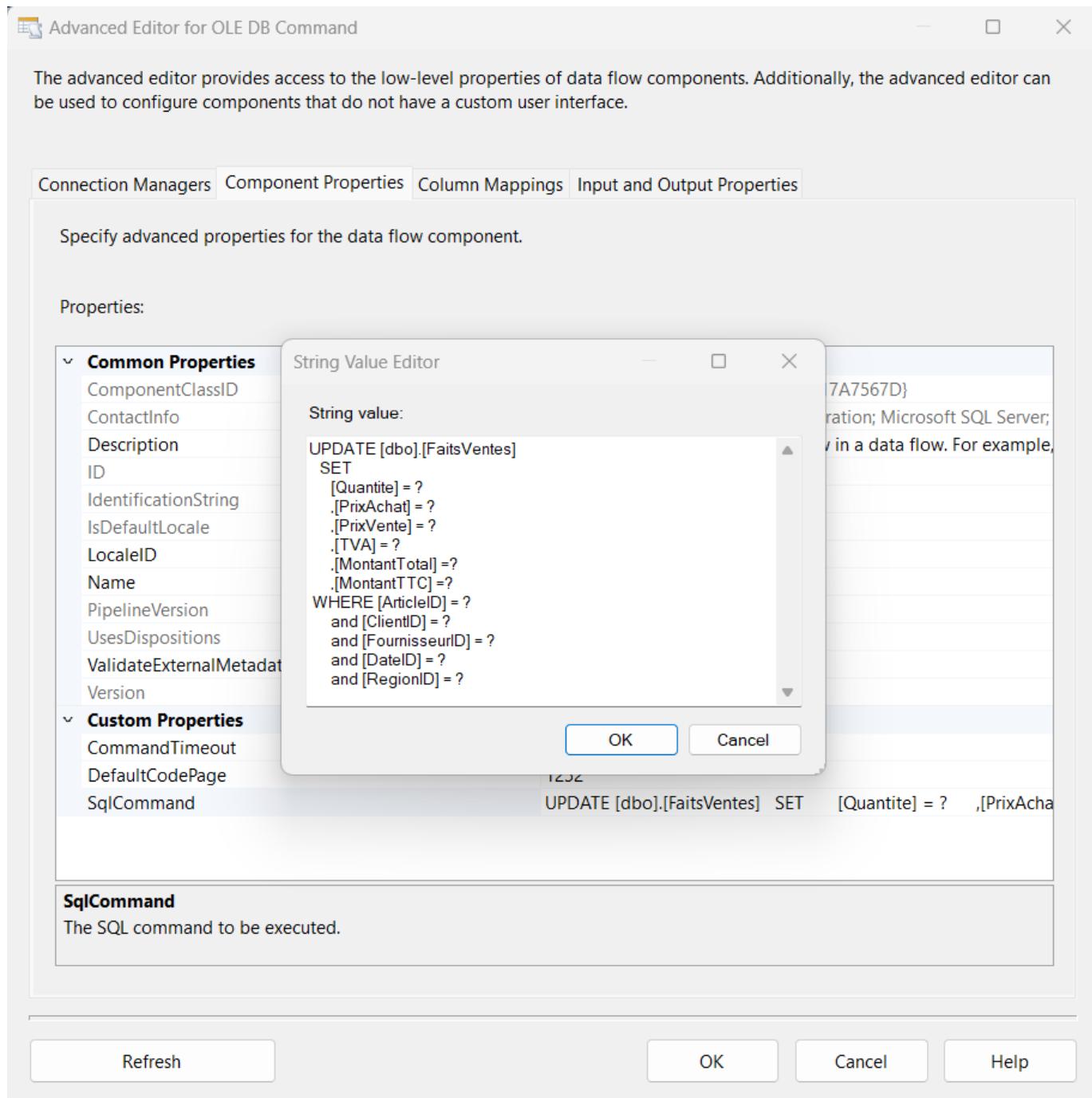


FIGURE 4.21 – Lookup Match Output (update) Fact Table

Implementation Fact Table :

The screenshot shows the SQL Server Management Studio interface. At the top, there are tabs for 'SQLQuery2.sql - ISMAEL.master (sa (59))' (selected), 'ISMAEL.p2msDW - dbo.FaitsVentes', 'ISMAEL.p2msSA - dbo.DimClient', and 'SQLQuery1.sql - ISMAEL.master (sa (53))'. Below the tabs, the query window displays the following T-SQL code:

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [ArticleID]
    ,[ClientID]
    ,[FournisseurID]
    ,[DateID]
    ,[RegionID]
    ,[Quantite]
    ,[PrixAchat]
    ,[PrixVente]
    ,[TVA]
    ,[MontantTotal]
    ,[MontantTTC]
FROM [p2msDW].[dbo].[FaitsVentes]

```

Below the code, the results pane shows a table with 12 columns: ArticleID, ClientID, FournisseurID, DateID, RegionID, Quantite, PrixAchat, PrixVente, TVA, MontantTotal, and MontantTTC. The data consists of 12 rows of sales facts. At the bottom of the results pane, a green checkmark indicates 'Query executed successfully.' and the text 'ISMAEL'.

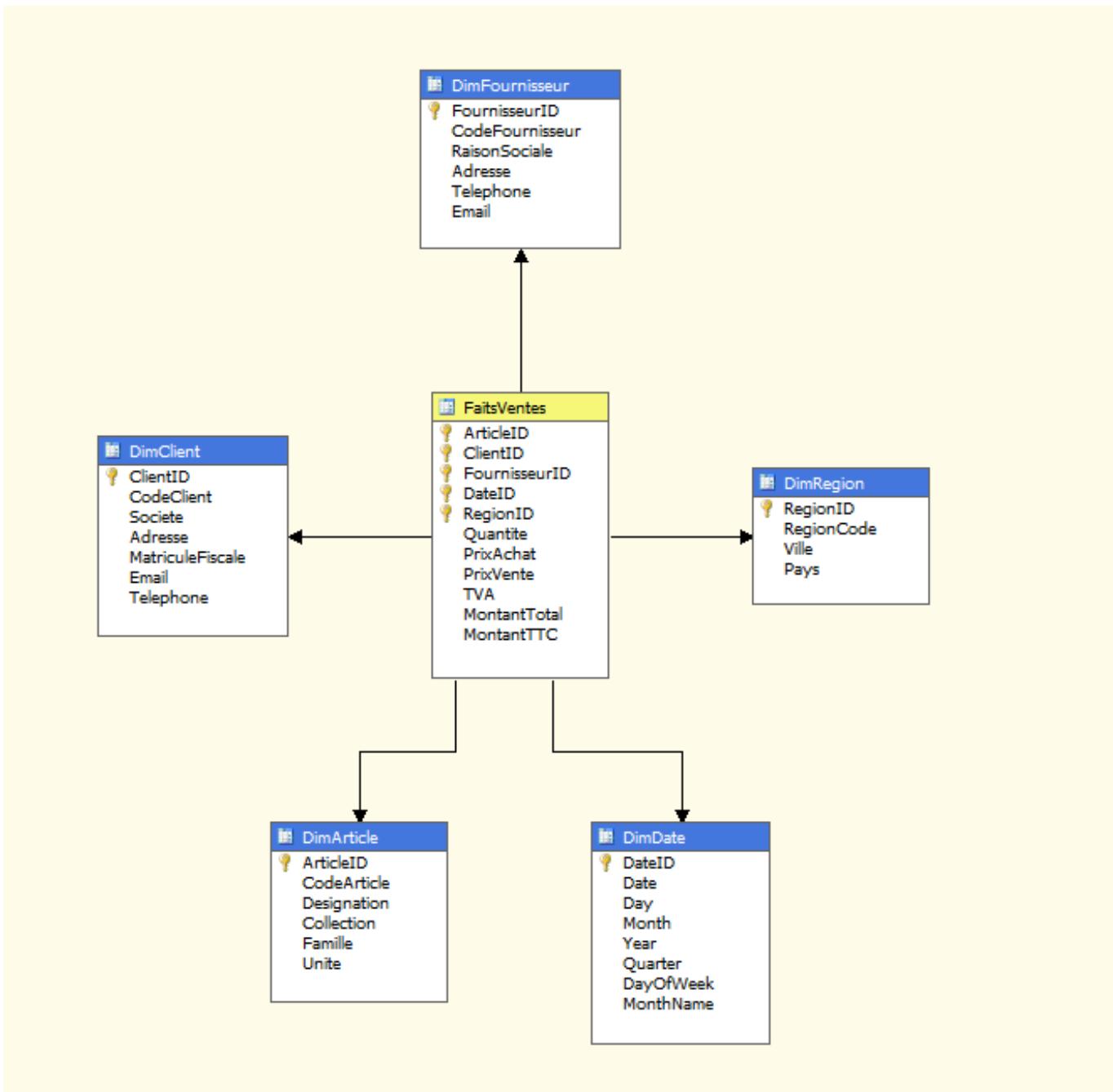
	ArticleID	ClientID	FournisseurID	DateID	RegionID	Quantite	PrixAchat	PrixVente	TVA	MontantTotal	MontantTTC
1	1	1	1	1	10	51,5	82	19	820	975,8	
2	1	1	1	139	1	81910	12336,5	16462	19	1348402420	1604598879,8
3	1	1	1	169	4	27310	4146,5	5542	19	151352020	180108903,8
4	1	1	1	337	7	54610	8241,5	11002	19	600819220	714974871,8
5	1	1	2	131	8	96470	14520,5	19374	19	1869009780	2224121638,2
6	1	1	2	161	2	41870	6330,5	8454	19	353968980	421223086,2
7	1	1	2	329	5	69170	10425,5	13914	19	962431380	1145293342,2
8	1	1	2	359	8	14570	2235,5	2994	19	43622580	51910870,2
9	1	1	3	153	3	56430	8514,5	11366	19	641383380	763246222,2
10	1	1	3	183	3	1830	324,5	446	19	816180	971254,2
11	1	1	3	321	3	83730	12609,5	16826	19	1408840980	1676520766,2
12	1	1	3	351	6	29130	4419,5	5906	19	172041780	204729718,2

FIGURE 4.22 – Implementation Fact Table

4.2.3 Phase de Construction du Cube OLAP

4.2.3.1 Définition Cube OLAP

Cube OLAP : Cette phase consiste à concevoir et déployer des cubes OLAP dans un système de Business Intelligence. Le cube OLAP organise les données multidimensionnelles, permettant des analyses rapides et efficaces via des mesures, des dimensions et des hiérarchies.

Cube OLAP :**FIGURE 4.23 – Cube OLAP P2MS**

4.2.3.2 Hiérarchie Cube OLAP

Hiérarchie Date :

- Dans un cube OLAP, la hiérarchie de date permet de structurer les données temporelles de manière à faciliter l'analyse à différents niveaux de granularité. Par exemple, une hiérarchie date pourrait être organisée comme suit :
- Date : La date exacte de l'événement ou de la transaction.
- MonthName : Le nom du mois, permettant une agrégation mensuelle des données.
- Quarter : Le trimestre de l'année, offrant une vue agrégée par trimestre.
- Year : L'année, la plus grande unité de temps dans cette hiérarchie, permettant une analyse annuelle.
- Cette hiérarchie permet aux utilisateurs d'explorer les données à différents niveaux, en passant de la vue détaillée des dates individuelles à des vues agrégées par mois, trimestre ou année.

The screenshot shows the Microsoft Analysis Services Dimension Designer. The top navigation bar includes 'Dimension Struct...', 'Attribute Relationships', 'Translations', and 'Browser'. Below the navigation is a toolbar with various icons. The main area is divided into three panes:

- Attributes** pane: Shows a tree view of attributes under 'Dim Date'. Nodes include Date, Date ID, Day, Day Of Week, Month Name, Quarter, and Year.
- Hierarchies** pane: Displays a hierarchy named 'Hierarchy_date' with levels: Year, Quarter, Month Name, and Date. A tooltip indicates: "To create a new hierarchy, drag an attribute here." A placeholder '<new level>' is also visible.
- Data Source View** pane: Shows a list of attributes from the 'DimDate' table: DateID, Date, Day, Month, Year, Quarter, DayOfWeek, and MonthName.

FIGURE 4.24 – Hiérarchie Date

Attribute Relationships : Les relations d'attributs dans un cube OLAP définissent comment les différentes dimensions sont liées entre elles. Par exemple, dans une hiérarchie de date, les attributs de niveau inférieur comme MonthName et Quarter sont reliés aux niveaux supérieurs comme Date et Year. Ces relations permettent de naviguer facilement entre les niveaux de granularité et d'effectuer des analyses cohérentes et précises.

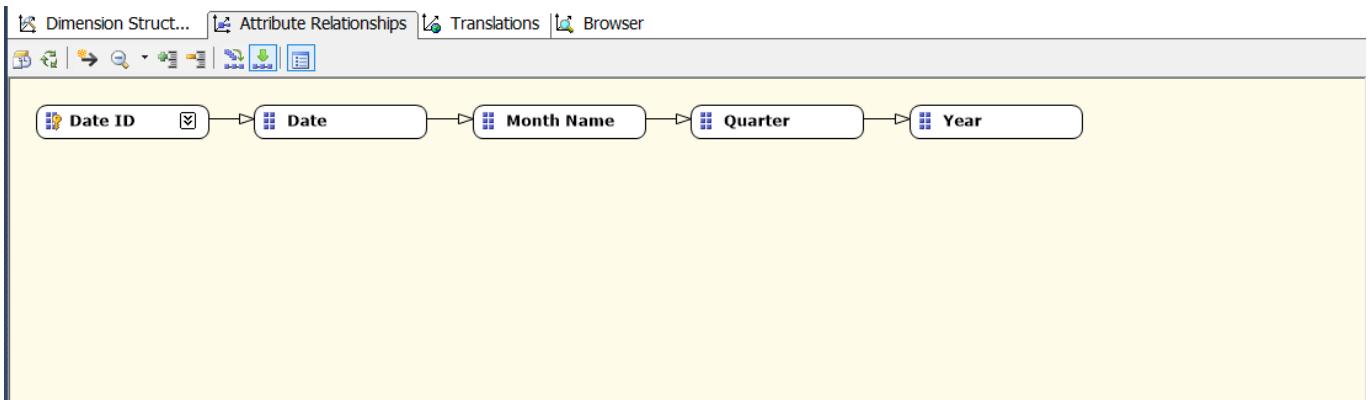


FIGURE 4.25 – Attribute Relationships

Execution Hierarchie Date :

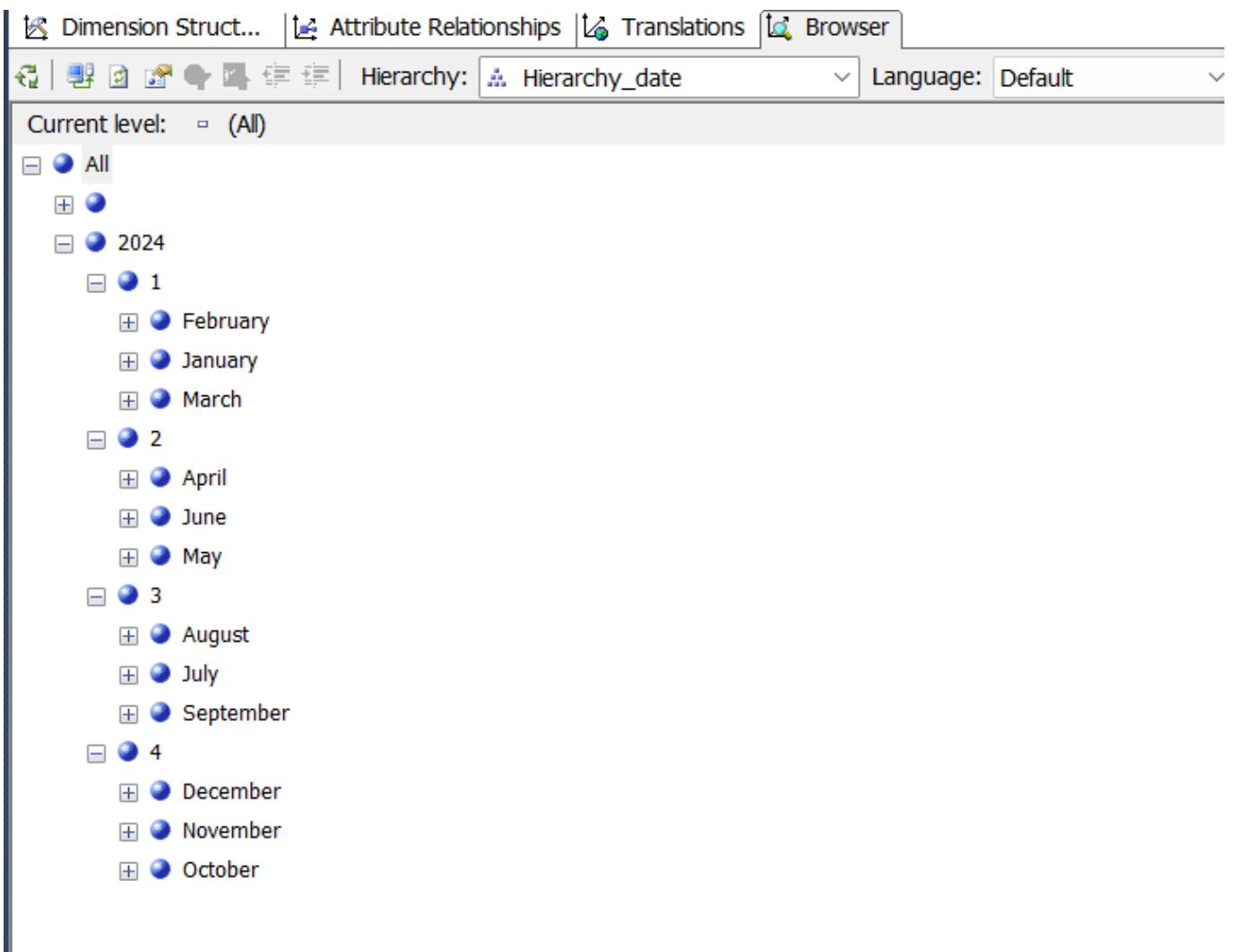


FIGURE 4.26 – Execution Hierarchy Date

Hiérarchie Région :

- La hiérarchie région permet de structurer les données géographiques pour une analyse efficace des ventes et autres métriques par région. Une hiérarchie région pourrait être structurée comme suit :

- RegionCode : Le code de la région, une clé unique identifiant chaque région.
- Ville : Les villes au sein de chaque région, permettant une vue détaillée des données à l'échelle urbaine.
- Pays : Le pays dans lequel les villes et régions sont situées, offrant une vue d'ensemble nationale.

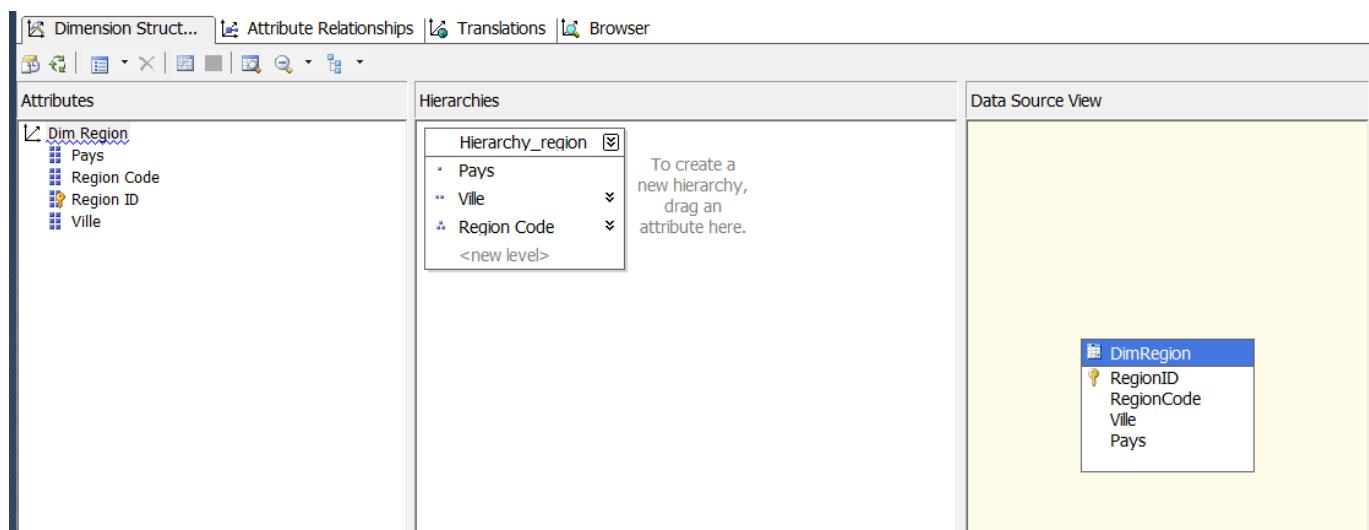


FIGURE 4.27 – Hierarchy Region

Dans un cube OLAP, ces relations d'attributs permettent de naviguer facilement à travers les données en utilisant des agrégations et des décompositions basées sur la hiérarchie. Par exemple, tu pourrais vouloir voir les ventes totales au niveau national (par pays), mais aussi explorer les détails des ventes au niveau régional et par ville.

Attribute Relationships :

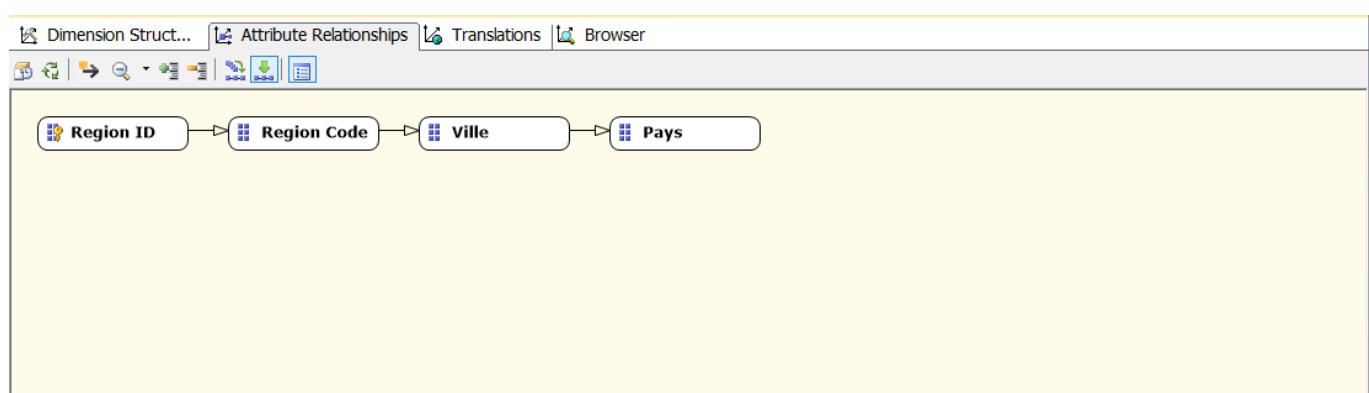


FIGURE 4.28 – Attribute Relationships

Execution Hierarchie Region :

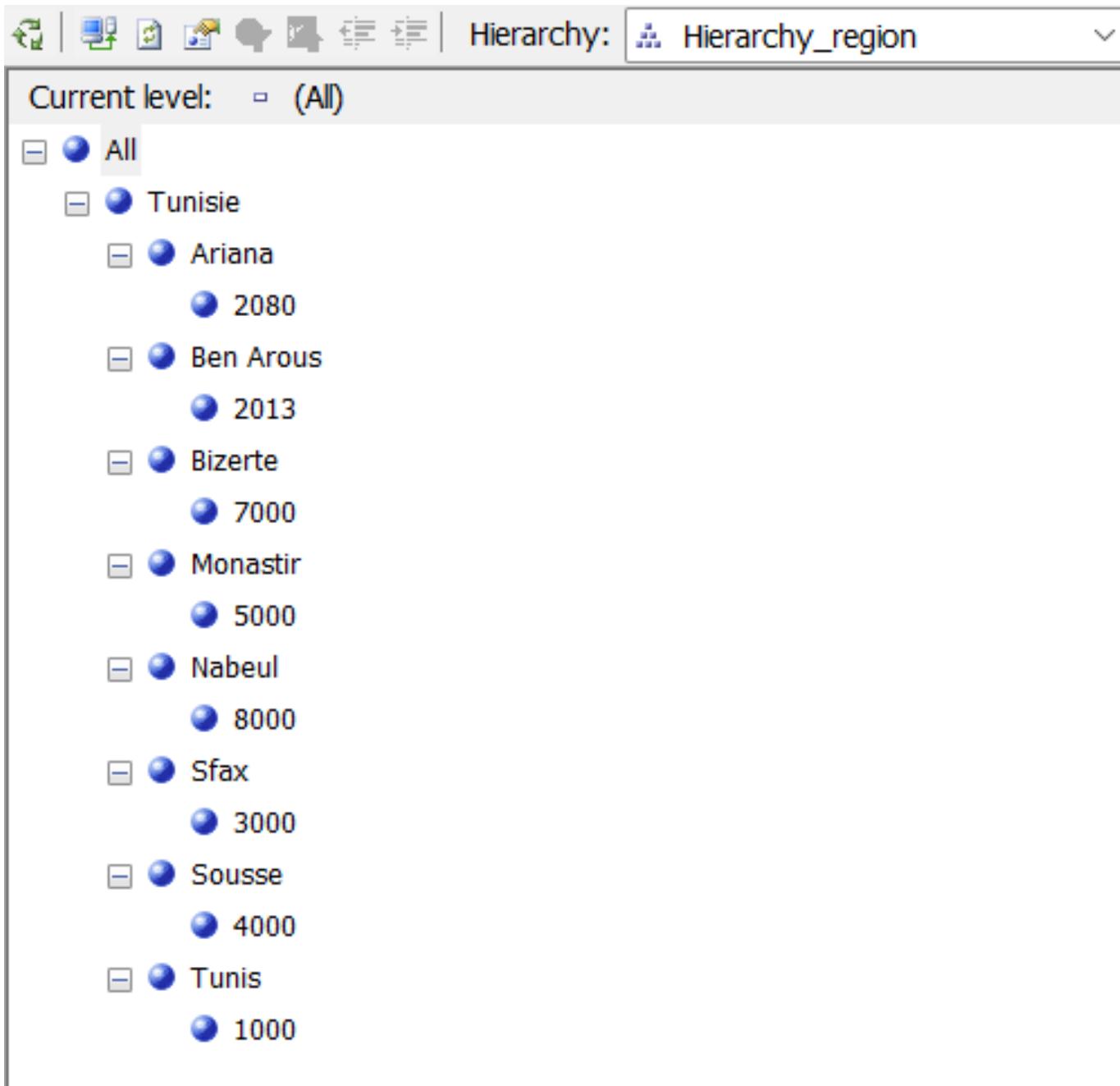


FIGURE 4.29 – Execution Hierarchie Region

4.2.3.3 Mesure :

Mesure Marge Brute :

La mesure MargeBrute est calculée comme la différence entre le montant total des ventes et le prix d'achat des produits. Cette mesure représente le bénéfice brut réalisé sur les ventes après avoir soustrait le coût d'achat des produits.

The screenshot shows the 'Script Organizer' interface in Analysis Services. A script named 'MargeBrute' is selected. The 'Expression' section contains the formula: `([Measures].[MontantTotal] - [Measures].[PrixAchat])`.

FIGURE 4.30 – Mesure Marge Brute

Mesure Totat des Ventes :

La mesure TotalDesVentes est calculée en multipliant la quantité vendue par le prix de vente. Cette mesure indique le montant total des ventes générée par la quantité de produits vendus et leur prix de vente.

The screenshot shows the 'Script Organizer' interface in Analysis Services. A script named 'TotaldesVentes' is selected. The 'Expression' section contains the formula: `Sum([Measures].[Quantite]*[Measures].[Prix Vente])`.

FIGURE 4.31 – Mesure Totat des Ventes

4.2.3.4 Filtres Dynamiques

Filtre dynamique Top 5 Clients :

Le filtre dynamique Top5Clients sélectionne les cinq clients ayant généré le plus de revenus, en se basant sur le montant total des ventes. Ce filtre aide à identifier les cinq clients les plus importants pour l'entreprise en termes de revenus.

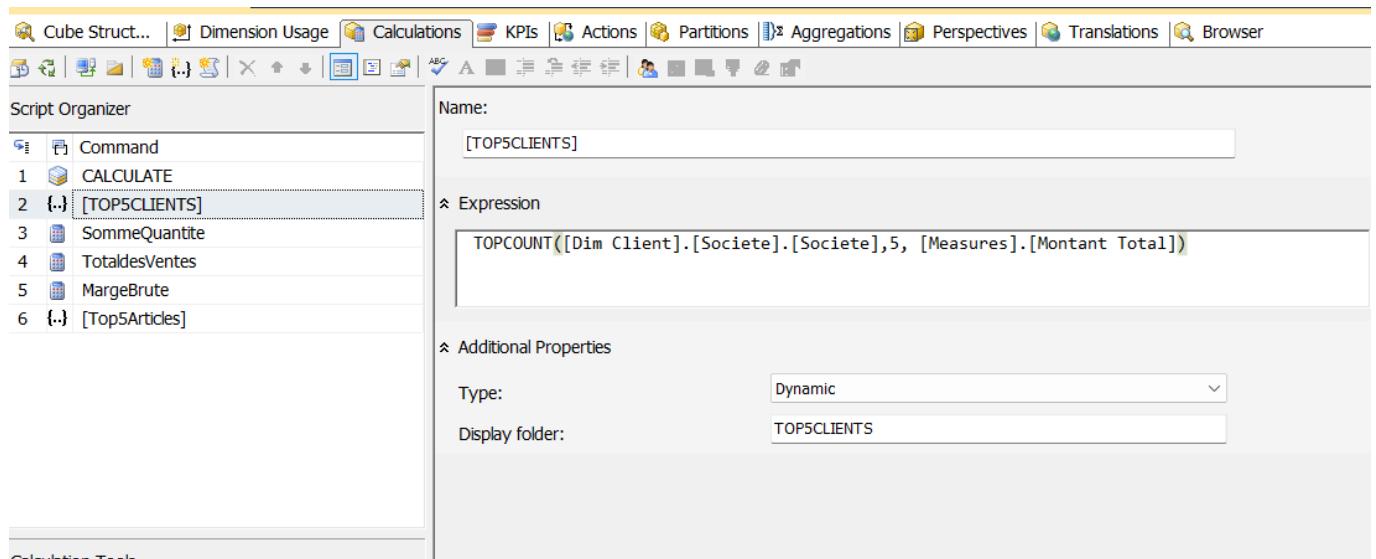


FIGURE 4.32 – Filtre dynamique Top 5 Clients

Execution Filtre dynamique Top 5 Clients :

The screenshot shows the Analysis Services Management Studio interface. On the left, there's a tree view of the database and measure groups. The main pane displays the results of the query:

Societe	Matricule Fiscale	Prix Achat	Prix Vente	Montant TTC	Montant Total
ZORJAT FADHLA	5377436 ABC 000	10784613	14398524	108242518...	9579516993...
ZOUARI ABDLTIF	6137392 ABC 000	10793237	14410036	108372212...	959094721...
ZRELI FEDIA	5510468 ABC 000	1079109...	14407178	108351833...	9588124431...
ZRELLI AMENI	1695325 MPA 000	10788950	14404320	108307352...	9585254714...
ZRELLI SLAHEDDIN	1936069 ABC 000	1078680...	14401462	108286981...	9582385567...

FIGURE 4.33 – Execution Filtre dynamique Top 5 Clients

Filtre dynamique Top 5 Articles :

- Le filtre dynamique Top 5 Articles sélectionne les cinq articles ayant généré le plus de ventes, basé sur le montant total.
- Ce filtre permet d'identifier les cinq articles les plus performants en termes de ventes.

The screenshot shows the Analysis Services Script Organizer interface. In the left pane, under 'Script Organizer', there is a list of objects:

- 1 CALCULATE
- 2 [TOP5CLIENTS]
- 3 SommeQuantite
- 4 TotaldesVentes
- 5 MargeBrute
- 6 [..] [Top5Articles]

In the main pane, the 'Name:' field is set to '[Top5Articles]'. The 'Expression' field contains the formula: `TopCount([Dim Article].[Designation].[Designation],5,[Measures].[Montant Total])`. Under 'Additional Properties', the 'Type:' is set to 'Dynamic' and the 'Display folder:' is set to 'Top5Articles'.

FIGURE 4.34 – Filtre dynamique Top 5 Articles

Execution Filtre dynamique Top 5 Articles :

The screenshot shows the Analysis Services Query Editor. On the left, the 'Measure Group' dropdown is set to '<All>'. The 'Filter' section shows a configuration for the 'Dim Article' dimension with a hierarchy 'Designation' and an operator 'In', using the filter expression 'Top5Articles'. The right side displays a table of product data:

Designation	Famille	Prix Achat	Prix Vente	Montant TTC
VERRE IONOMER PHOTO	Verre ionomère	2911370	3886960	277080338304
VISCOSTATE	Hémostatiques	2910792,5	3886190	308062634803
VISCOSTATE HEMOSTATIQUE	Hémostatiques	2910215	3885420	276914668492
VISIERE	Protection	2909637,5	3884650	307878421851
VITA AKZENT GLAZE 5 G	Liquides	2909060	3883880	276749064592

The left sidebar shows the cube structure with dimensions like 'p2ms DW' and measures like 'Faits Ventes Count', 'Montant Total', etc.

FIGURE 4.35 – Execution Filtre dynamique Top 5 Articles

4.2.4 Phase de Développement et Déploiement dans Power BI

Cette phase implique l'utilisation de Power BI pour créer des rapports et des tableaux de bord interactifs à partir des données préparées et modélisées. Elle inclut la conception des visualisations, la configuration des connexions aux sources de données, et le déploiement des rapports pour les utilisateurs finaux. OverView :

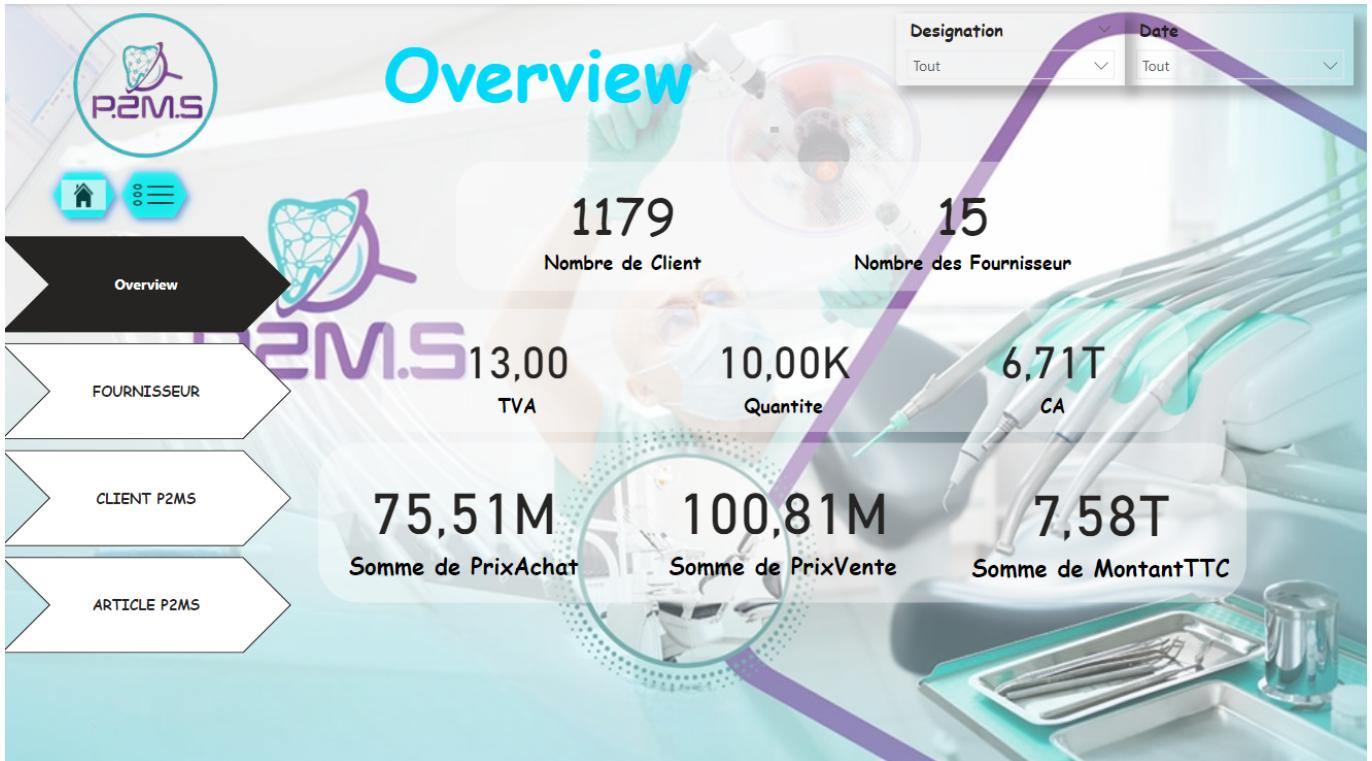


FIGURE 4.36 – OverView

Dashboard Article :

- Graphique en barres et lignes (Nombre de Quantité, Somme de Prix Achat, Somme de Prix Vente par Famille) : Ce graphique combine des barres pour représenter le Nombre de Quantité vendue par différentes familles de produits, et des lignes pour montrer la Somme de Prix Achat et la Somme de Prix Vente. Chaque famille de produit, comme "Silicone", "Instruments", etc., est comparée selon ces trois indicateurs, ce qui permet de comprendre à la fois la performance des ventes et les marges potentielles pour chaque catégorie de produits.
- Diagramme circulaire (Nombre d'Article par Unité) : Le diagramme circulaire présente la répartition des articles en fonction de leur unité de mesure : ml, g, et unité. Il illustre la proportion de chaque unité dans l'ensemble des articles vendus, en mettant en évidence que les unités "g" et "unité" sont presque équitablement partagées avec environ 34
- Graphique en courbes (CA par Mois) Ce graphique en courbes montre l'évolution du Chiffre d'Affaires (CA) mois par mois. Il permet d'observer les variations saisonnières ou les tendances au fil du temps, avec des pics notables en juin (582Md) et en juillet (582Md). Cela aide à identifier les périodes de forte activité et à anticiper la planification des ventes ou des stocks. Carte géographique (CA par Ville et Collection)
- Cette carte de la Tunisie affiche les performances du Chiffre d'Affaires par Ville et par Collection. Les différentes couleurs des segments de diagramme circulaire représentent différentes collections (Adhésifs, Chirurgie, Divers, etc.). Elle permet de localiser les zones géographiques

où certaines collections sont dominantes ou plus performantes, avec des points comme Tunis, Sousse, et Monastir étant des centres clés d'activité.

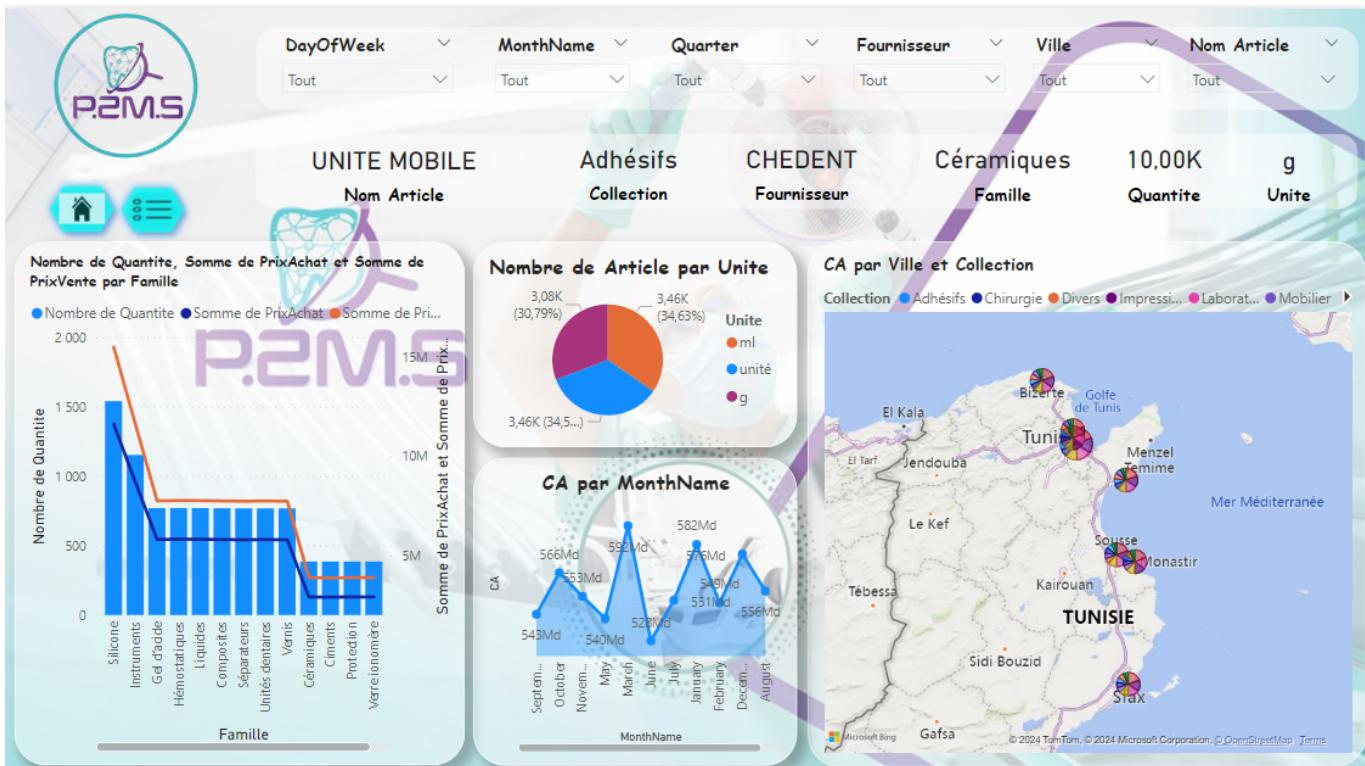


FIGURE 4.37 – Dashboard Article

Dashboard Client :

- **Heatmap (Somme de Prix Vente par Collection et Société)** : Ce graphique représente une heatmap (carte thermique) qui montre la répartition des ventes pour différentes collections de produits (comme Restauration, Impression, Laboratoire, etc.) selon plusieurs sociétés (par exemple ZORATI Fadhma, ZOUAOUI Mohamed, etc.). La taille des segments et les couleurs permettent de visualiser rapidement quelles sociétés réalisent le plus de ventes pour chaque collection de produits, offrant une perspective comparative entre les différentes collections et sociétés.
- **Graphique en barres (Nombre de Société par Adresse)** : Ce graphique illustre la répartition des sociétés par localisation géographique (adresse). Il met en évidence que la majorité des sociétés se trouvent à Tunis, suivie de loin par Nabeul, Béja, Ben Arous, et Ariana. Cela aide à visualiser la concentration géographique des clients et à identifier les zones d'activité majeure.
- **Diagramme circulaire (Top 5 Sociétés par Prix Vente)** : Ce diagramme circulaire met en lumière les cinq plus grandes sociétés en termes de ventes (Somme de Prix Vente). Les segments colorés montrent la contribution de chaque société dans le total des ventes, avec les noms des sociétés comme ZOUARI ABDLTIF, ZRELLI AMENI, etc. Les pourcentages permettent de comprendre la part de marché relative de chaque société parmi les cinq premières.
- **Tableau (Liste des Clients avec Détails)** : Ce tableau fournit une liste détaillée des clients, avec des informations telles que le Code Client, Nom de la société, Matricule Fiscale, Email, Téléphone, et Adresse. Il permet un accès rapide et détaillé aux informations critiques sur chaque client, facilitant la gestion et le suivi.

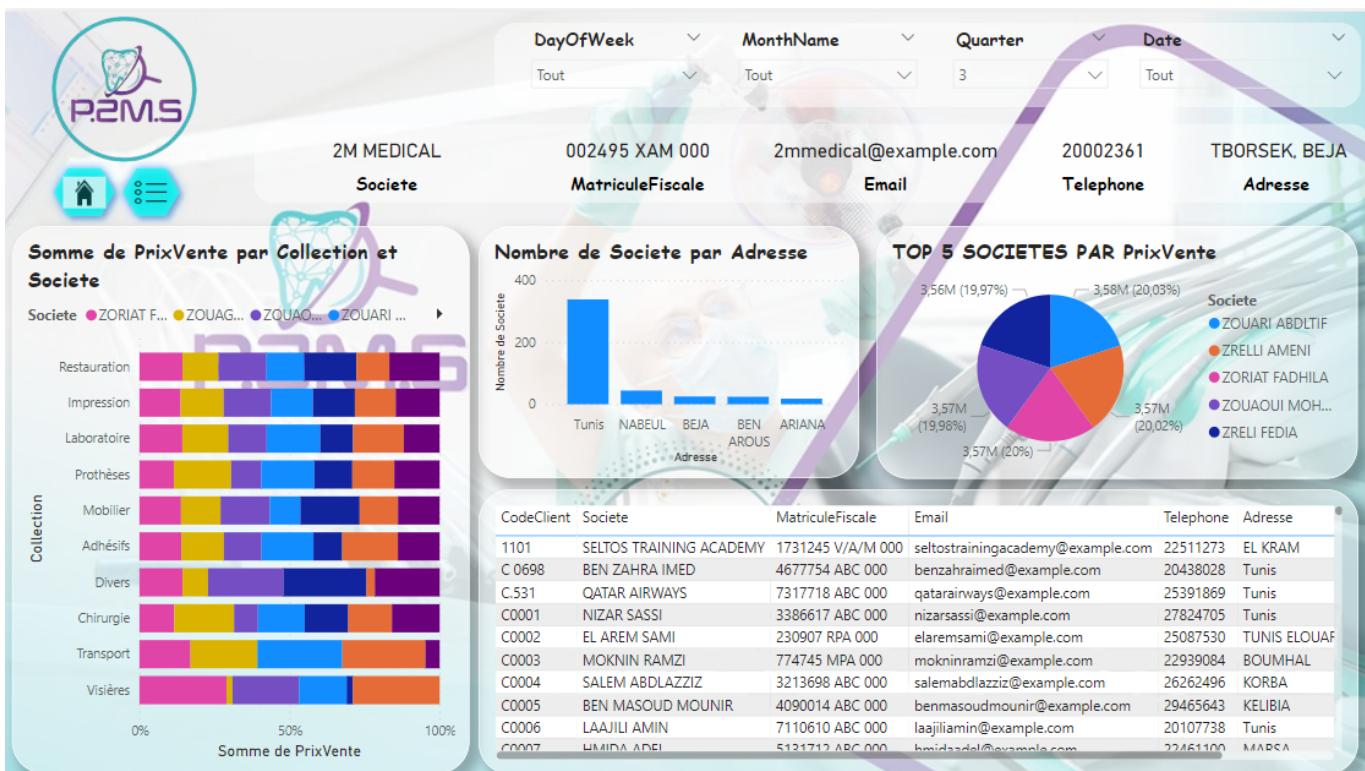


FIGURE 4.38 – Dashboard Client

Dashboard Fournisseur :

- Map des Fournisseurs (Treemap des Fournisseurs par Ville) :

Ce treemap affiche une vue hiérarchique des fournisseurs selon leur localisation géographique. Chaque carré représente un fournisseur, avec la taille indiquant son poids relatif dans l'ensemble. Les villes comme Tunis, Ariana, Nabeul, Sfax, et Sousse sont représentées, offrant une vue rapide sur les principaux acteurs dans chaque zone géographique.

- Graphique en barres (Évolution Prix Achat et Nombre de Fournisseurs par Ville) :

Ce graphique à barres compare le nombre de fournisseurs par ville (axe gauche) avec la somme des prix d'achat correspondants (axe droit). Tunis et Ariana comptent le plus grand nombre de fournisseurs, mais la somme des prix d'achat varie considérablement, ce qui aide à analyser la performance des fournisseurs dans chaque région.

- Diagramme circulaire (Nombre de Fournisseurs par Adresse) :

Ce diagramme circulaire montre la distribution des fournisseurs par localisation géographique. Les différentes couleurs représentent les adresses principales (comme Tunis, Ariana, Nabeul, Sfax, etc.), avec des pourcentages montrant la proportion de fournisseurs dans chaque ville. Cela permet de comprendre la répartition géographique des fournisseurs.

- Diagramme circulaire (Top 5 Fournisseurs par Prix Achat) :

Ce diagramme circulaire présente les cinq principaux fournisseurs en termes de prix d'achat. Les segments colorés (DISTRI-MED, DIVERS, IMS, etc.) illustrent la contribution de chaque fournisseur dans le total des achats, aidant à identifier les principaux partenaires stratégiques de l'entreprise.

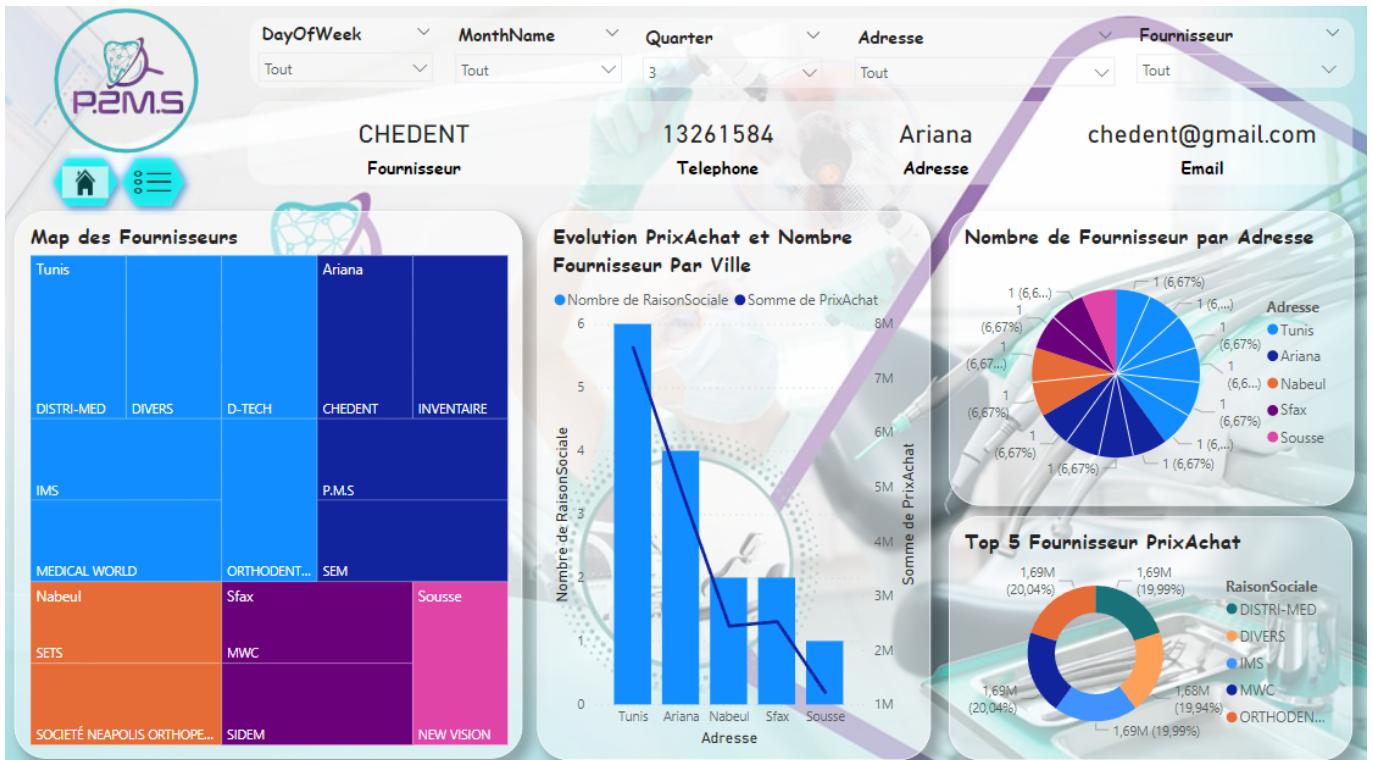


FIGURE 4.39 – Dashboard Fournisseur

4.2.5 Phase de Machine Learning

Dans la phase de machine learning de notre projet, nous avons mis en place un modèle prédictif pour identifier les articles les plus demandés durant une période donnée. Pour ce faire, nous utilisons comme variables d'entrée les dates de début monthstart et de fin monthend du mois en cours. Ces données temporelles permettent de définir la période d'analyse pour la prédiction. En utilisant ces dates, nous extrayons les ventes historiques d'articles et les tendances de demande au cours de la période spécifiée. Le modèle de machine learning, tel qu'un modèle de régression ou une méthode de séries temporelles, est ensuite formé sur ces données historiques pour prévoir les articles qui seront les plus populaires durant la période cible. En analysant les patterns de demande et les variations saisonnières, le modèle fournit des prévisions précises qui aident à optimiser les stocks et à planifier les stratégies de vente. Les résultats de la prédiction sont ensuite intégrés dans les tableaux de bord pour une visualisation et une prise de décision améliorées.

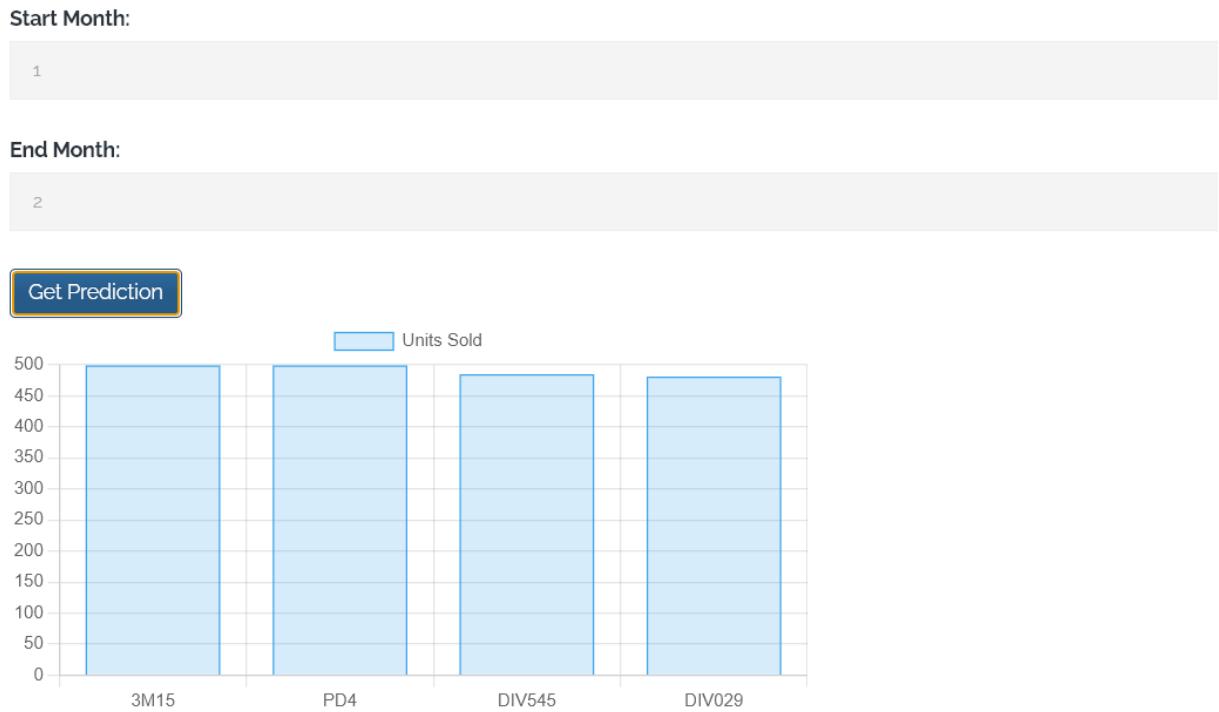


FIGURE 4.40 – Prédiction Les articles plus demandés dans un périod

4.2.6 Phase de Déploiement d’Application Angular et Spring Boot

4.2.6.1 Power BI Service

- L'utilisation de Power BI Service joue un rôle essentiel dans la phase de déploiement, car elle permet de partager, collaborer et publier les rapports et tableaux de bord créés durant le développement.
- Publication des Rapports : Les rapports développés dans Power BI Desktop sont publiés sur Power BI Service, permettant ainsi une distribution facile à tous les utilisateurs autorisés. Cette étape assure que les données analytiques et les visualisations sont accessibles en ligne, facilitant leur consultation depuis n'importe quel endroit.
- Partage et Collaboration : Power BI Service offre des fonctionnalités de partage qui permettent aux utilisateurs de collaborer sur les rapports et les tableaux de bord. Les équipes peuvent commenter, discuter des insights et suivre les modifications en temps réel, améliorant ainsi la prise de décision collective.
- Gestion des Accès et Sécurité : Power BI Service permet de gérer les autorisations d'accès aux rapports et aux tableaux de bord. Les administrateurs peuvent définir qui peut voir, éditer ou partager les contenus, assurant ainsi une sécurité appropriée et un contrôle des accès basé sur les rôles des utilisateurs.

- Mises à Jour Automatiques : Les données publiées peuvent être actualisées automatiquement grâce à des configurations de rafraîchissement des données. Cela garantit que les rapports affichent toujours les informations les plus récentes, ce qui est crucial pour les analyses en temps réel et les décisions basées sur les dernières données disponibles.
- En résumé, la phase de déploiement du projet P2MS implique la finalisation des fonctionnalités d'inscription et de connexion dynamique en utilisant MySQL, ainsi que l'intégration et la publication des rapports via Power BI Service pour une analyse et une collaboration efficaces.

La phase de déploiement du projet P2MS est cruciale pour la mise en production des fonctionnalités développées ainsi que pour l'intégration des outils d'analyse et de reporting. Voici les principaux aspects abordés durant cette phase :

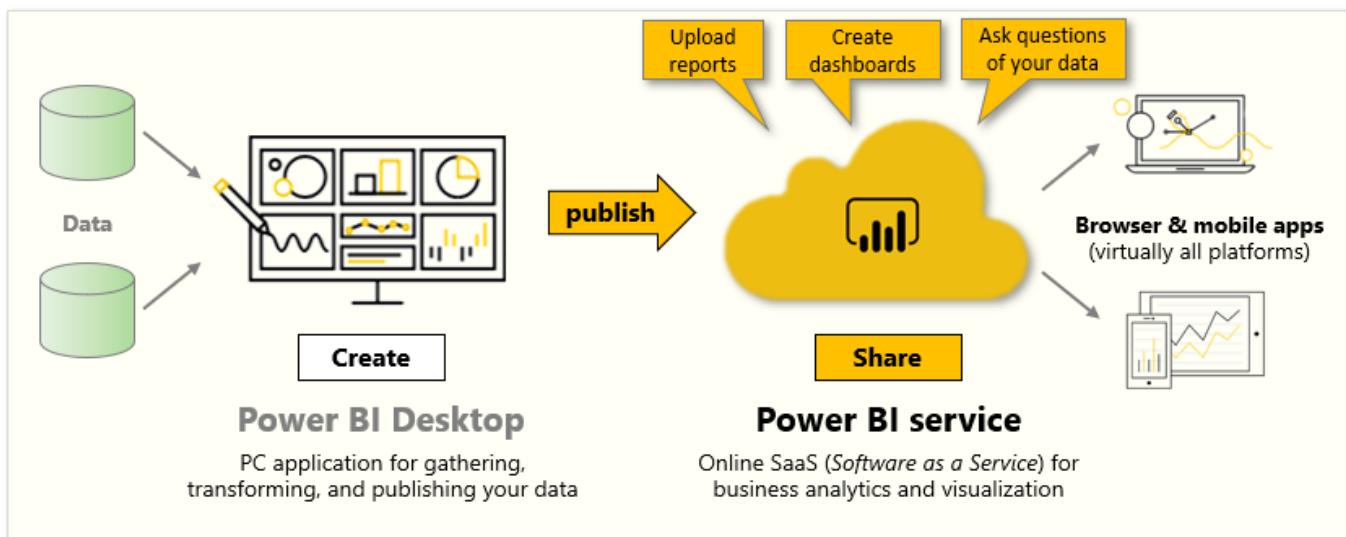


FIGURE 4.41 – Power BI Service

4.2.6.2 Inscription et Connexion Dynamique avec MySQL

Dans cette phase, une attention particulière est accordée à la mise en œuvre des fonctionnalités d'inscription (sign-up) et de connexion (sign-in) des utilisateurs, qui sont essentielles pour la gestion sécurisée des accès au système. Les interfaces d'inscription et de connexion sont conçues pour interagir directement avec une base de données MySQL, permettant une gestion dynamique des utilisateurs.

Inscription (Sign-Up) : Les utilisateurs peuvent créer de nouveaux comptes en fournissant des informations telles que le nom, l'adresse e-mail, un mot de passe et le rôle (Manager, Fournisseur, ou Client). En fonction du rôle sélectionné, les données sont validées et stockées dans la base de données MySQL. Un processus d'inscription robuste inclut la vérification des données pour éviter les doublons et garantir la sécurité des informations personnelles.



FIGURE 4.42 – Sign Up Interface

Connexion (Sign-In) : Les utilisateurs se connectent à leur compte en entrant leur adresse e-mail et leur mot de passe. Le système vérifie les informations fournies en les comparant aux données stockées dans la base de données MySQL. Une fois connecté, le système identifie le rôle de l'utilisateur (Manager, Fournisseur, ou Client) et lui accorde l'accès aux fonctionnalités et aux dashboards correspondant à son rôle. Les Managers ont accès à tous les dashboards, les Fournisseurs ont accès à l'Overview et au Fournisseur Dashboard, et les Clients ont accès à l'Overview et au Client Dashboard.



FIGURE 4.43 – Sign In Interface

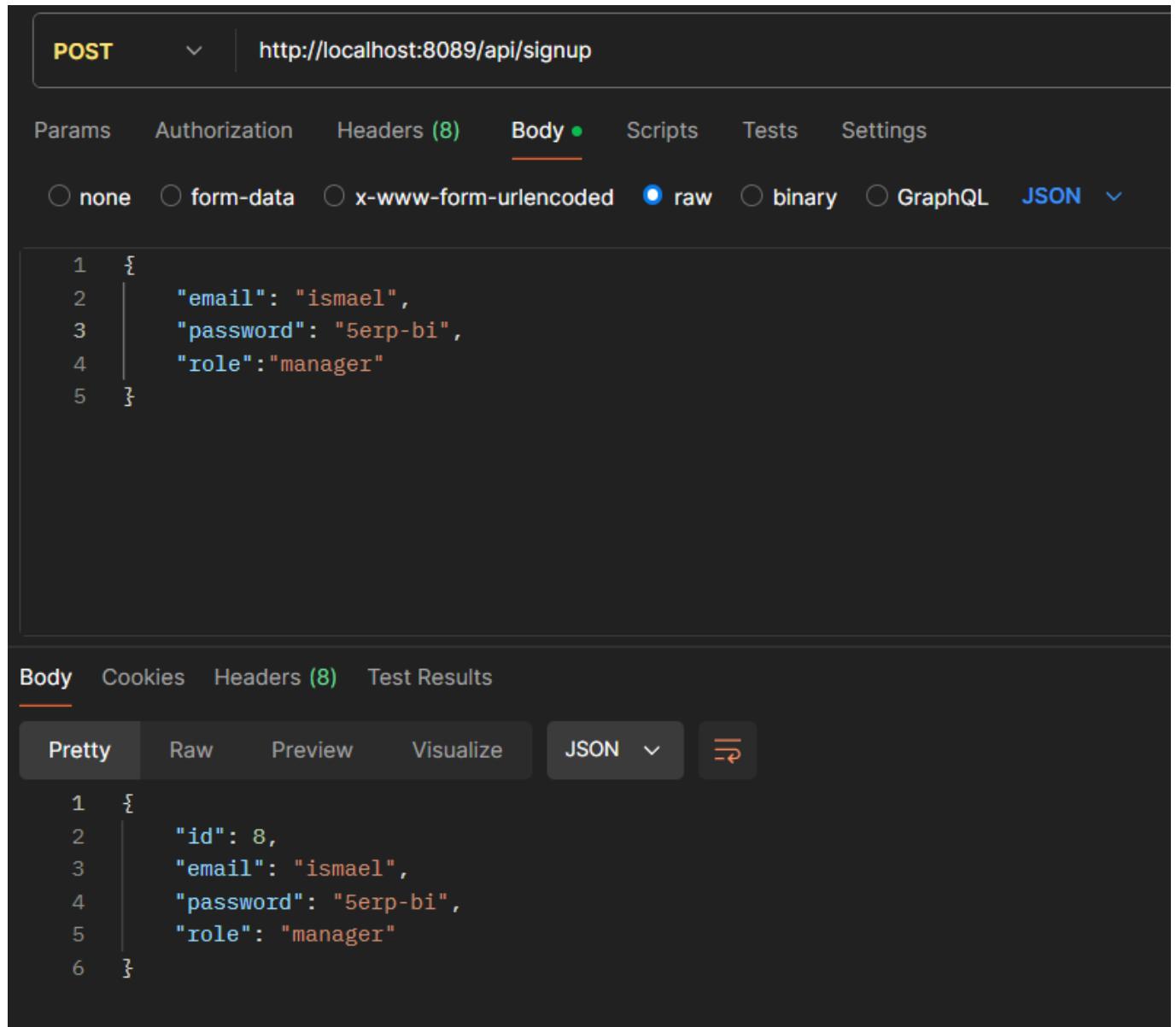
Le contrôle de saisie est un élément crucial pour assurer la sécurité et l'exactitude des informations lors de la connexion des utilisateurs au système. Pour garantir que les utilisateurs accèdent correctement à leurs comptes, un processus rigoureux de validation des informations d'identification est mis en place.


FIGURE 4.44 – Contrôle de saisie

La capture d'écran ci-dessous montre le test de la méthode sign-up (inscription) réalisée avec Postman. Sur cette image, nous observons une requête POST envoyée à l'endpoint /api/signup. Le corps de la requête est en format JSON et inclut les champs nécessaires tels que email, password, et role. Nous avons utilisé un email de test, un mot de passe sécurisé, et spécifié un rôle pour l'utilisateur. Après l'envoi de la requête, la réponse du serveur est affichée, indiquant un code HTTP 201, ce qui confirme que l'inscription a été effectuée avec succès. La réponse JSON contient également un message de confirmation et un identifiant utilisateur, ce qui assure que les données ont été correctement enregistrées dans la base de données. En cas d'erreur, comme un email déjà utilisé ou des informations invalides, nous nous attendons à recevoir des messages d'erreur détaillés, ce qui aide à identifier et à corriger les problèmes potentiels lors de l'inscription des utilisateurs.

	id	email	password	role
<input type="checkbox"/> Éditer Copier Supprimer	1	ismail.elgharnougui@esprit.tn	123	manager
<input type="checkbox"/> Éditer Copier Supprimer	3	ismail	ismail	manager
<input type="checkbox"/> Éditer Copier Supprimer	8	ismael	5erp-bi	manager

FIGURE 4.45 – Table User

**FIGURE 4.46 – postman verification sign up**

La capture d'écran suivante illustre le test de la méthode sign-in (connexion) réalisée avec Postman. Cette image montre une requête POST envoyée à l'endpoint /api/signin. Le corps de la requête, en format JSON, contient les informations de connexion de l'utilisateur, à savoir l'email et le mot de passe. Après avoir envoyé la requête, nous observons la réponse du serveur, qui inclut un code HTTP 200, confirmant que les informations de connexion sont correctes et que l'utilisateur est authentifié avec succès. La réponse JSON renvoie un jeton d'authentification, qui est essentiel pour les sessions utilisateur, ainsi qu'une indication du rôle de l'utilisateur. En cas d'informations incorrectes, telles qu'un email ou un mot de passe erroné, nous nous attendons à ce que le serveur renvoie un message d'erreur approprié, ce qui permet de diagnostiquer et de résoudre les problèmes liés à la connexion.

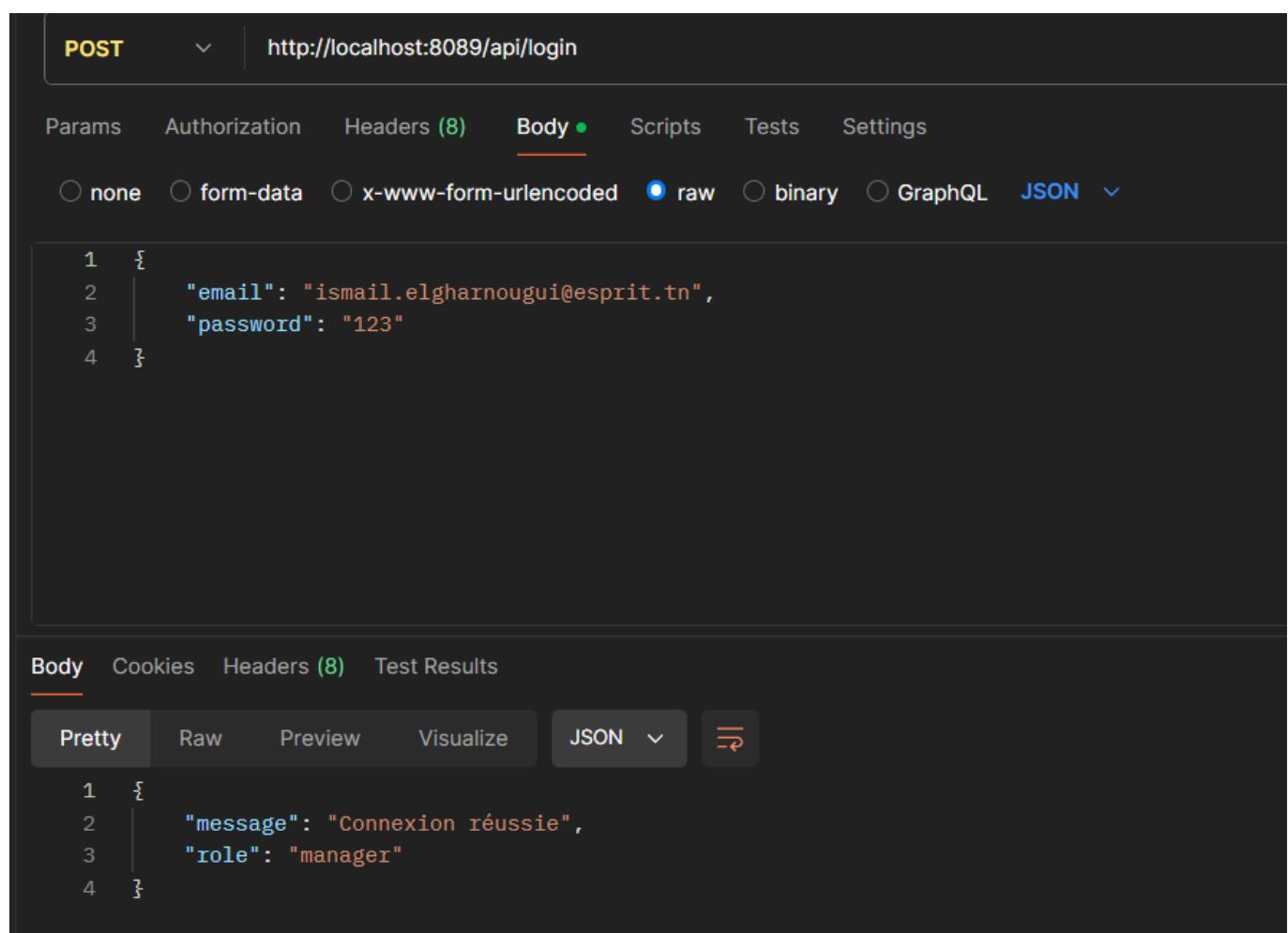


FIGURE 4.47 – postman verification sign up

CHAPITRE 5

Conclusion

Mon stage, effectué au sein de l'école **Esprit** pour l'entreprise **Planète Médicale (P2MS)**, a constitué une expérience particulièrement enrichissante, tant d'un point de vue professionnel que personnel. Cette opportunité m'a permis de mettre en pratique les enseignements théoriques acquis durant ma formation académique et de les appliquer à des projets concrets, dans un environnement de travail dynamique et stimulant.

Durant cette période, j'ai eu l'occasion de m'investir dans plusieurs projets qui m'ont permis de développer des compétences techniques pointues, notamment en **Business Intelligence** et en **Développement Web**. J'ai également pu améliorer mes capacités en gestion de projets, respect des délais et intégration au sein d'une équipe, renforçant ainsi mes compétences en gestion du temps et en communication professionnelle.

Cette expérience m'a également enseigné l'importance de la flexibilité et de la capacité à apprendre rapidement de nouvelles compétences dans un environnement en constante évolution. Les interactions régulières avec mes collègues et mon encadrant ont été d'une grande valeur, me fournissant des conseils avisés et de nouvelles perspectives pour la suite de mon parcours professionnel.

En conclusion, ce stage a non seulement confirmé mon intérêt pour les domaines du **Business Intelligence** et du **Développement Web**, mais il m'a également donné les outils et la confiance nécessaires pour aborder les défis futurs avec sérénité. Je tiens à exprimer ma profonde gratitude à **Planète Médicale (P2MS)** pour cette expérience inestimable, ainsi qu'au personnel de l'**école Esprit** pour leur soutien constant. Fort de cette expérience déterminante, je me sens désormais mieux préparé à poursuivre ma carrière dans ces domaines avec ambition et motivation.



ESPRIT SCHOOL OF ENGINEERING

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : 1-2 rue André Ampère - 2083 - Pôle Technologique - El Ghazala - Tél +216 70 250 000 - Fax +216 70 685454