



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Tunis

Institut Supérieur de Gestion de Tunis

RAPPORT DE STAGE DE FIN D'ÉTUDES

Présenté en vue de l'obtention
Du Diplôme de la Licence en Business Intelligence
Parcours: Business intelligence

Implementation d'un système de Reporting

Réalisé au sein de l'entreprise :ELCO Solutions

Élaboré par:
Bouzidi Dhia

Encadré par:

Enc. Pédagogique:
Mme.Meriam Jemel

Enc. Professionnelles:
Mr.Rami Taghouti

Année Universitaire:
2023/2024

DÉDICACES

Je dédie ce travail à :

Mes très chers parents

dont le soutien moral et matériel, ainsi que l'affection inconditionnelle, ont été constants tout au long de ma carrière scolaire et étudiante, et continuent de me donner le courage, l'inspiration et la force nécessaire pour avancer avec succès. C'est grâce à eux que je suis ce que je suis et que j'ai la possibilité de devenir ce que je souhaite être.

Toute ma famille et Tous mes amis

qui ont partagé avec moi les moments les plus précieux de ma vie et qui m'ont soutenu avec gentillesse et compassion dans les moments difficiles.

Tous ceux qui me sont chers

dont le soutien indéfectible et l'amour inconditionnel ont été des sources d'inspiration et de motivation tout au long de ce parcours. Votre présence et votre soutien ont été des éléments essentiels de ma réussite, et je vous en suis profondément reconnaissant.

REMERCIEMENTS

Je tiens à exprimer ma gratitude à l’Institut Supérieur de Gestion de Tunis pour la qualité exceptionnelle de l’enseignement et la sélection rigoureuse du personnel enseignant. Je tiens à exprimer ma sincère gratitude à Mme Meriam Jemel pour ses critiques constructives, ses précieux conseils et sa disponibilité sans égal durant mon parcours. Je remercie également profondément Rami Taghouti pour son accompagnement attentif et ses conseils éclairants, qui ont grandement influencé l’avancée de mon projet.

De plus, je tiens à remercier tout le personnel d’Elco Solutions pour sa foi inébranlable, son soutien constant et ses précieux encouragements. Leur coopération a été très importante pour la réussite de mon projet et mon évolution professionnelle.

En conclusion, je suis honoré d’avoir eu l’opportunité de bénéficier des connaissances et du soutien de ces institutions et individus remarquables qui ont facilité mon parcours académique et professionnel.

TABLE DES MATIÈRES

Introduction Générale	xi
1 Etude du projet	1
1.1 Introduction	1
1.2 Présentation de l'Organisme d'Accueil	1
1.3 Les secteurs d'activités	2
1.3.1 Automobile :	2
1.3.2 Industrie :	3
1.3.3 Outil de surveillance et de supervision Industriel :	3
1.4 Organigramme de Elco Solutions	4
1.5 Presentation du projet	5
1.5.1 Contexte du Projet	5
1.5.2 Cadre du Projet	5
1.5.3 Critique de l'Existant	6
1.5.4 Solution Proposée	7
1.6 Méthodologie de Travail	8
1.6.1 Méthode GIMSI	8
1.6.2 Méthode Scrum	9
1.7 Conclusion	12

2 Analyse et spécification des besoins	13
2.1 Introduction	13
2.2 Analyse des besoins	13
2.2.1 Spécification des besoins fonctionnels	13
2.2.2 Spécification des besoins non fonctionnels	14
2.3 Le pilotage du projet avec Scrum	15
2.3.1 Identification de l'équipe	15
2.3.2 Diagramme de Gantt	16
2.3.3 Backlog de produit	17
2.4 Conception globale du DataWarehouse	18
2.4.1 Étude du modèle conceptuel	18
2.4.2 Conception Globale du dataWarehouse	19
2.5 Environnement de travail	20
2.5.1 Environnement Matériel	20
2.5.2 Environnement Logiciel	20
2.5.3 Etude comparative entre les outils BI	22
2.6 Conclusion	24
3 Création datamart "Elco MES"	25
3.1 Introduction	25
3.2 Backlog de sprint	25
3.3 Identification des indicateurs de performance du datamart 'Elco MES' .	26
3.4 Étude du modèle dimensionnel	27
3.4.1 Modèle en Étoile	27
3.4.2 Modele en flocon	28
3.4.3 Modelé en Constellation	29
3.4.4 Choix du type de modèle	30
3.5 Conception du datamart 'Elco MES'	31
3.5.1 Choix des dimensions	31
3.5.2 Choix des mesures	32

3.5.3	Table de faits	33
3.6	Alimentation du data mart ‘Elco MES’	34
3.6.1	Nos sources des données	34
3.6.2	Phase ETL par VS CODE :	35
3.6.3	Phase ETL pour les Sources CSV par TALEND :	41
3.6.4	Phase ETL pour les Sources JSON par TALEND :	44
3.6.5	Phase ETL pour les Sources SSMS par TALEND :	51
3.7	Conclusion :	67
4	Sprint 3:	68
4.1	Introduction	68
5	Environnement de travail	69
5.1	Introduction	69
5.2	Environnement Matériel	69
	Conclusion Générale	70

LISTE DES FIGURES

1.1	Logo de Elco Solutions	2
1.2	Le domaine automobile	2
1.3	Carte arduino	3
1.4	dashboard monitoring tools	3
1.5	Organigramme de Elco Solutions	4
1.6	Solution Proposée	7
1.7	Méthodologie SCRUM vs GIMSI	10
1.8	Méthodologie SCRUM vs GIMSI	11
2.1	Équipe Scrum.	15
2.2	Diagramme de Gantt	16
2.3	Schéma Conceptuelle de l'entrepôt de données	19
2.4	processus ETL du Logiciel Talend	21
2.5	Logo SSMS	21
2.6	Logo PowerBI	22
2.7	Architecture du PowerBI	22
2.8	Logo Tableau	23
2.9	Logo Pentaho	23
3.1	Modèle en Etoile	28
3.2	Modèle en Flocon	29

3.3 Modéle en Constellation	30
3.4 Source JSON	35
3.5 Source CSV	36
3.6 Source SSMS	36
3.7 Donnée Json avant transformation	37
3.8 Etape 1	38
3.9 Etape 2	39
3.10 Application du Script	40
3.11 La division des fichiers	40
3.12 Resultat Finale	41
3.13 Etape 1	42
3.14 Etape 2	43
3.15 Resultat	44
3.16 Architecture TALEND	45
3.17 tFileList	46
3.18 tFileinputDelimeter	47
3.19 TMAP	48
3.20 Load	49
3.21 Resultat final	50
3.22 Etape 1	51
3.23 Etape 2	52
3.24 Etape 3	53
3.25 Etape 1	54
3.26 Etape 2	54
3.27 Etape 3	55
3.28 Etape 1	55
3.29 Etape 2	56
3.30 Etape 3	57
3.31 Etape 4	58

3.32 Etape 1	58
3.33 Etape 1	59
3.34 Etape 3	60
3.35 Etape 4	61
3.36 Etape 5	62
3.37 Etape 6	63
3.38 Etape 7	64
3.39 Resultat	65
3.40 Fait Magny	66
3.41 Fait LeonFrot	67

LISTE DES TABLES

2.1	Tableau d'équipe	15
2.2	Backlog produit	17
2.3	Table de Dimensions	19
3.1	Backlog du premier sprint	26
3.2	description des indicateurs du datamart Elco MES	26
3.3	Les Dimensions	31
3.4	Les mesures des table de fait	32
3.5	Table de fait Magny	33
3.6	Table de fait LeonFrot	33
3.7	Table de fait Voltaire	33

INTRODUCTION GÉNÉRALE

Dans un monde où les données sont omniprésentes, la capacité à extraire des informations significatives est essentielle pour la prise de décision éclairée. L'application Elco MES (Elco Manufacturing Execution System) vise à répondre à ce besoin en fourni ssant des informations précieuses à partir des données collectées par les capteurs. Cependant, la création de rapports sur ces résultats s'est avérée complexe, et c'est là que réside notre défi.

La Business Intelligence (BI) est un domaine clé dans le paysage technologique et commercial. Un domaine qui englobe les méthodes, les technologies et les outils utilisés pour collecter, analyser et présenter des informations pertinentes pour la prise de décision. Elle permet de transformer les données brutes en connaissances exploitables.

Elco MES est une application conçue pour optimiser les processus de fabrication (récupère les données des capteurs et les agrège pour fournir des informations aux opérateurs et aux responsables de la production).

Malgré ses avantages, Elco MES a rencontré des difficultés lors de la création de rapports sur les résultats des capteurs. Les problèmes incluent la complexité des données et le manque de clarté dans les rapports existants.

Pour résoudre ces problèmes, nous adopterons la méthode SCRUM. Cette approche itérative et collaborative nous permettra de gérer efficacement le développement

du système de reporting. Nous détaillerons notre démarche tout au long de ce rapport..

Il est réparti en cinq chapitres:

Le premier chapitre qui s'intitule " présentation générale du projet " est dédié à la présentation de l'organisme d'accueil ainsi que la présentation du cadre général du projet.

Le deuxième chapitre qui s'intitule " Phase de préparation " présente l'environnement de travail et récapitule les différents objectifs du projet et les axes d'analyses et donne une idée générale sur le modèle de conception de notre entrepôt de données.

Le troisième chapitre qui s'intitule.

Le quatrième chapitre qui s'intitule

Le cinquième et dernier chapitre qui s'intitule

En guise de conclusion, nous allons terminer notre rapport avec une conclusion générale qui va récapituler la totalité du travail élaboré durant toute la période de ce stage. Nous allons également énoncer les différentes perspectives.

CHAPITRE 1

ETUDE DU PROJET

1.1 Introduction

Dans le premier chapitre, nous nous concentrerons d'abord sur le cadre général et l'étude du projet. Il s'agit en fait d'une présentation de l'organisme d'accueil, suivie d'une présentation du projet , d'une définition de la mission, décrire l'existant et le critiquer . Puis nous proposons une solution.

1.2 Présentation de l'Organisme d'Accueil

Elco-Solutions, fondée en 2016, est une entité entièrement privée spécialisée dans l'élaboration de logiciels embarqués et les solutions de connectivité. Cette société s'est donnée pour mission de renforcer la compétitivité des grandes firmes internationales en exploitant son expertise pointue dans le domaine du développement logiciel, de l'Internet des Objets (IoT) et des technologies de connectivité. Avec des certifications Iso 9001, Iso 27001:2013 et Tisax, Elco-Solutions se distingue par son engagement envers la qualité et la sécurité dans tous ses processus et solutions.



Figure 1.1: Logo de Elco Solutions

1.3 Les secteurs d'activités

1.3.1 Automobile :

Elco propose une gamme étendue de services dans le secteur de l'automobile. Ils se spécialisent dans l'accompagnement des projets clients liés à divers domaines, Voitures connectées - Contrôle moteur - Processeur de véhicule - Recharge sans fil, Travailler avec plusieurs équipementiers automobiles de niveau 1 pour développer des technologies de pointe. Elco couvre un large éventail de disciplines technologiques et accompagne ses clients tout au long du cycle en V. Elco-Solutions Software est déjà intégrée dans les voitures allemandes et françaises produites en série.

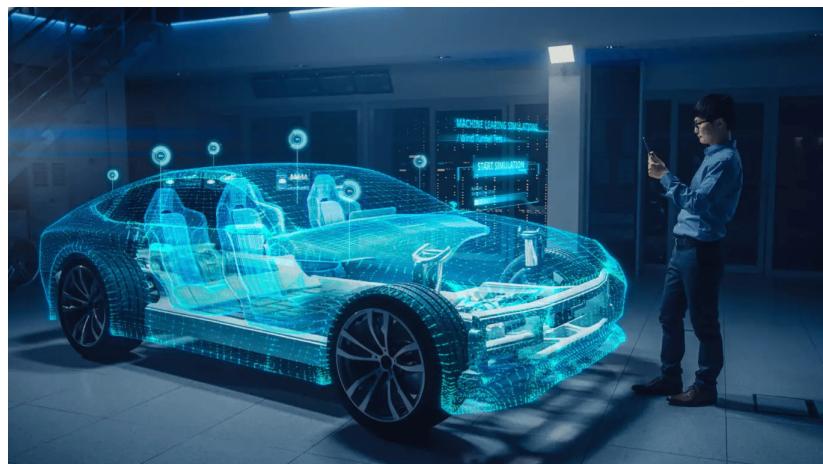


Figure 1.2: Le domaine automobile

1.3.2 Industrie :

L'équipe d'Elco-Solutions met à disposition ses services pour une large gamme de besoins dans le domaine de l'automobile. Leurs domaines d'expertise incluent le développement de Linux embarqué, les réseaux et les bus de terrain, l'internet des objets, ainsi que le développement d'applications industrielles.

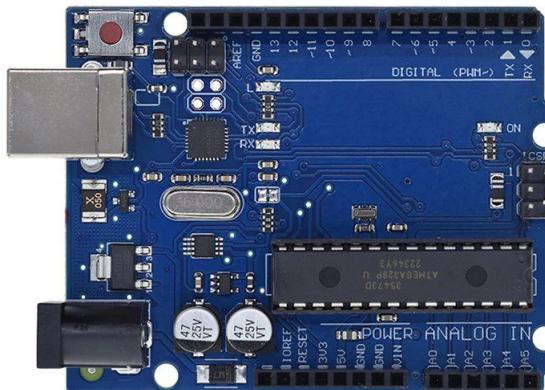


Figure 1.3: Carte arduino

1.3.3 Outil de surveillance et de supervision Industriel :

ElcoMES 4.0 :

- Connectivité
- Décision basée sur les données
- Supervision et contrôle.

Aider les organisations manufacturières à améliorer la productivité et à réduire le temps et les coûts de production en tirant parti des capacités de mesure des indicateurs de performance clés permettant la prise de décision basée sur les données et le contrôle statistique des processus.



Figure 1.4: dashboard monitoring tools

1.4 Organigramme de Elco Solutions

Il existe plusieurs départements au sein de l'entreprise et chaque département est composé de différents départements. Viennent ensuite une équipe dédiée pour assurer un haut niveau de qualité de service aux clients. tout au long du stage, nous avons été intégrés au département Monitoring Tools.

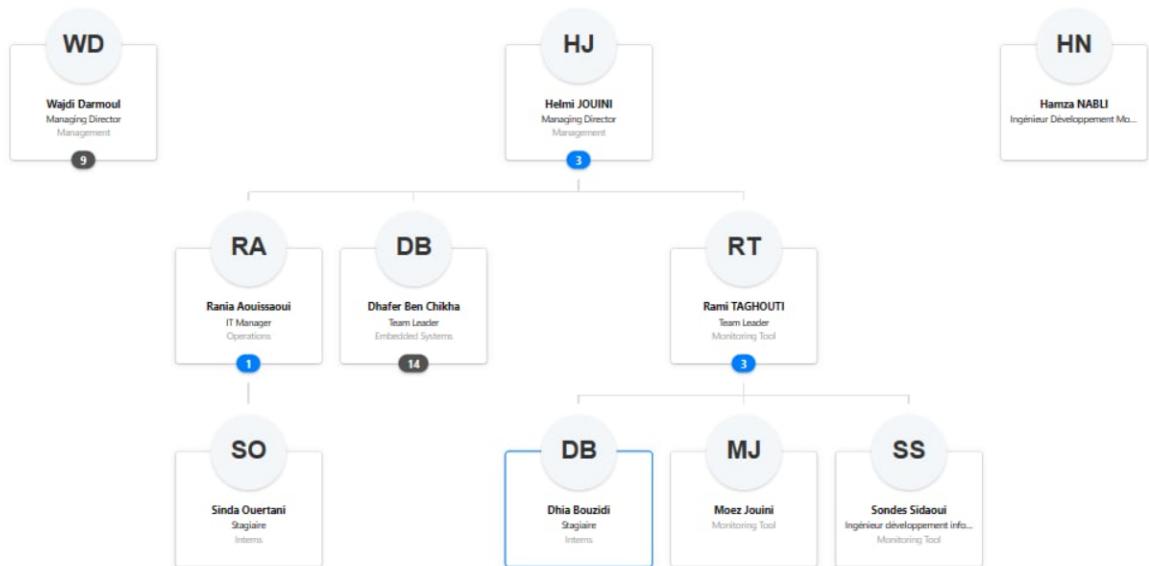


Figure 1.5: Organigramme de Elco Solutions

1.5 Présentation du projet

1.5.1 Contexte du Projet

L'application Elco MES (Elco Manufacturing Execution System) est au cœur de notre projet. Elle collecte des données à partir de capteurs et joue un rôle essentiel dans la surveillance de la production. Cependant, la création de rapports sur ces résultats s'est avérée complexe. Elco Solutions utilise actuellement des fichiers Excel pour gérer ces rapports, mais rencontre des difficultés. Bien qu'Excel soit un outil puissant, il présente des défis spécifiques pour la gestion des rapports dans le contexte d'Elco MES.

1.5.2 Cadre du Projet

Ce projet s'inscrit dans une volonté d'innovation et d'efficacité opérationnelle. L'objectif est de développer un tableau de bord spécifiquement adapté aux besoins de l'application Elco MES . Ce tableau de bord vise à :

- Optimisation des Rapports : Créer des rapports qui synthétisent efficacement les données des capteurs, permettant aux utilisateurs de comprendre rapidement les tendances et les anomalies.
- Clarté et Pertinence : Concevoir des rapports visuellement attrayants et faciles à interpréter, afin que les parties prenantes puissent prendre des décisions éclairées basées sur des informations précises.
- Utilisation de la Méthode SCRUM : Nous adopterons la méthode SCRUM pour gérer le développement du système de reporting de manière itérative et collaborative.

1.5.3 Critique de l'Existant

Elco Solutions utilise actuellement des fichiers Excel pour gérer ces rapports, mais rencontre des difficultés :

- Manque de Visualisations Adaptées : Excel propose des graphiques et des tableaux, mais leur personnalisation et leur adaptabilité peuvent être limitées. Les rapports actuels peuvent manquer de visualisations claires et pertinentes.
- Difficulté d'Automatisation : Les mises à jour des rapports Excel doivent souvent être effectuées manuellement. Cela peut entraîner des erreurs et une perte de temps.
- Limitations de Structuration : Excel n'offre pas toujours la flexibilité nécessaire pour structurer les rapports de manière optimale. Les utilisateurs peuvent se retrouver avec des feuilles de calcul encombrées et peu intuitives.
- Complexité des Données : Les données provenant des capteurs sont souvent volumineuses et complexes. Comment pouvons-nous transformer ces données en rapports clairs et pertinents ?
- Clarté et Structuration : Les rapports Excel actuels manquent de clarté et de structuration. Comment pouvons-nous améliorer la présentation des informations pour faciliter la prise de décisions éclairées ?
- Adaptabilité : Comment pouvons-nous concevoir un système de reporting qui s'adapte aux besoins spécifiques d'Elco MES tout en garantissant une utilisation efficace des données existantes ?

Nous explorerons ces problèmes en détail et proposerons des solutions tout au long de ce rapport.

1.5.4 Solution Proposée

La solution envisagée est le développement d'un tableau de bord Power BI spécialement conçu pour le processus de Elco MES chez Elco Solutions :

- Tableau de Bord Interactif : Un tableau de bord facile à utiliser, offrant une visualisation en temps réel des données, avec des filtres et des interactions pour une analyse profonde
- Indicateurs de Performance Clés (KPIs) : Le tableau de bord mettra en évidence les KPIs essentiels tels que les résultats obtenu des capteurs : les valeurs maximales, minimales et moyennes.
cela facilitera la surveillance continue et l'amélioration des performances.
- Prise de Décision Basée sur les Données : En fournissant des insights instantanés et précis, le tableau de bord permettra une prise de décision rapide et éclairée.

Cette solution surmonte les limitations de l'application Elco MES en termes de visualisation des données et d'analyse, offrant à Elco Solutions un outil puissant pour optimiser son processus MES.

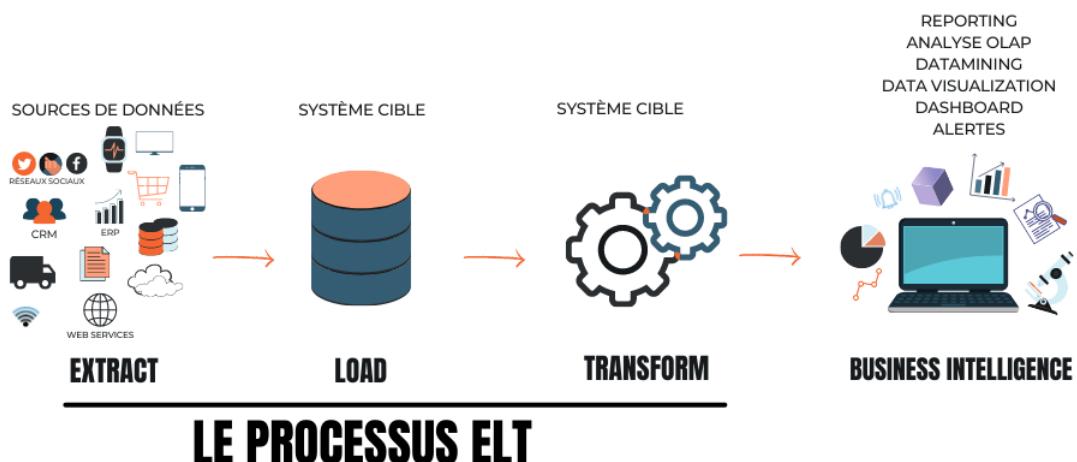


Figure 1.6: Solution Proposée

1.6 Méthodologie de Travail

Pour garantir le succès du projet, une approche adaptable est nécessaire pour permettre une mise en œuvre et une planification efficaces et rentables. Cependant, dans les affaires il n'existe pas d'approche universelle en matière de gestion de projet intelligente. Il est donc nécessaire de considérer plusieurs paramètres pour choisir la méthode la plus adaptée, tels que le type, les caractéristiques, les limites et les objectifs du projet. La gestion de projet joue un rôle essentiel dans la coordination des participants et des tâches pour assurer le succès du projet. Il est donc important d'évaluer soigneusement les différentes options avant de prendre une décision. Par conséquent, avant de commencer à mettre en œuvre notre projet, nous examinerons et évaluons la méthode de gestion de projet les plus adaptées pour optimiser la gestion de projet.

1.6.1 Méthode GIMSI

GIMSI (Guidelines for Information and Management Systems Implementation) est un outil de gestion de projet conçu spécifiquement pour les projets de développement de systèmes d'information. Elle propose une approche de gestion de projet structurée qui met l'accent sur l'importance de l'implication des parties prenantes. Les utilisateurs sont présents tout au long du cycle de vie du projet. L'approche GIMSI comprend quatre phases clés : Initialisation, planification, mise en œuvre et clôture. Chaque étape prend des mesures concrètes pour assurer la réussite du projet.

Avantages :

GIMSI fournit un cadre complet qui couvre l'ensemble du processus de gestion de projet. Ce cadre fournit une approche méthodique et structurée de la planification, de la mise en œuvre et du suivi des projets. En effet, il considère toutes les phases du cycle de vie du projet, de l'analyse des besoins à la production, en passant par la conception, le développement et les tests. Cela encourage la participation des utilisateurs, ce qui permet de garantir que le système répond aux exigences des utilisateurs.

GIMSI accorde une attention particulière à la gestion du changement, ce qui contribue à assurer l'intégration efficace du nouveau système dans l'organisation.

Limite :

La mise en œuvre de GIMSI peut être complexe et longue, ce qui peut ne pas convenir aux petits projets ou aux organisations aux ressources limitées. S'appuyer sur la participation des utilisateurs peut augmenter les délais des projets, car il faut plus de temps pour obtenir les commentaires des utilisateurs et apporter les modifications nécessaires.

1.6.2 Méthode Scrum

Scrum est une méthode de gestion de projet agile conçue pour être flexible et adaptable aux exigences changeantes du projet. Il met l'accent sur le développement itératif, les commentaires réguliers et la collaboration entre les membres de l'équipe. Cette méthode s'appuie sur des cycles de travail courts appelés sprints durant lesquels l'équipe travaille sur des fonctionnalités précises. Les sprints sont suivis d'une revue de sprint, qui évalue les progrès et identifie les améliorations potentielles pour le prochain sprint. La méthode Scrum encourage également la transparence et la communication régulière entre les membres de l'équipe et avec les parties prenantes externes.

Avantages :

Scrum a un haut degré de flexibilité, ce qui le rend idéal pour les projets qui peuvent changer avec le temps. Cette flexibilité est obtenue grâce à l'utilisation d'itérations courtes et d'un processus de développement incrémental qui permet une adaptation aux besoins. L'approche de développement itératif de Scrum peut aider à identifier les problèmes dès le début du cycle de vie du projet, réduisant ainsi le risque de problèmes majeurs. Après, en développant des versions successives du système, l'équipe peut assurer que les problèmes peuvent être rapidement identifiés et corrigés avant qu'ils ne deviennent un obstacle majeur. Des commentaires réguliers et une collaboration entre les membres de l'équipe permettent de garantir que le système



Figure 1.7: Méthodologie SCRUM vs GIMSI

répond aux exigences des utilisateurs. En impliquant les utilisateurs finaux tout au long du processus de développement, Scrum s'assure que le produit final est plus susceptible de répondre à leurs besoins et adoptés avec enthousiasme. **Équipe Scrum et artefacts :**

— Le Product Owner :

occupe le rôle de gestionnaire du carnet de produit. Il détient l'autorité pour accepter ou rejeter le travail présenté .

— L'équipe de développement :

est composée de différents membres qui occupent des rôles variés, tels que les concepteurs, les développeurs, et d'autres contributeurs. Ces membres collaborent ensemble pour créer et livrer le produit.

— Le Backlog de produit :

est un ensemble de demandes qui rassemblent les exigences formulées par les clients et les fonctionnalités à développer. Il représente une liste d'éléments à exécuter pour

Aspects	GIMSI	Scrum
Flexibilité	Moyenne	Élevée
Implication de l'utilisateur	Élevée	Élevée
Structuration du processus	Élevée	Faible
Focus sur la gestion des changements	Élevé	Faible
Documentation	Élevée	Faible
Adéquation pour les projets de petite taille	Non recommandé	Recommandé
Adéquation pour les projets à grande échelle	Recommandé	Recommandé
Gestion du temps et des coûts	Rigoureuse	Flexible
Risque de dérive du scope initial	Faible	Élevé
Identification précoce des problèmes	Faible	Élevée
Communication entre membres de l'équipe	Structurée	Informelle
Gestion de la qualité	Rigoureuse	Flexible

Figure 1.8: Méthodologie SCRUM vs GIMSI

le produit.

— Le Backlog de sprint :

est une compilation des histoires utilisateurs qui sont extraites du Backlog de produit.

Il représente la liste des éléments spécifiques à réaliser pendant un sprint donné.

— User Story :

Une fonctionnalité logicielle, décrite de manière concise et du point de vue de l'utilisateur final.

Limites :

Scrum peut ne pas être adapté pour les projets avec un périmètre fixe ou des exigences bien définies. La focalisation sur la flexibilité peut rendre la gestion des délais et des coûts du projet plus difficile. L'approche de développement itératif peut conduire à un manque de documentation, ce qui peut rendre la maintenance du système difficile à long terme.

L'entreprise Elco Solutions a requis la participation de son équipe pour recueillir leurs opinions et superviser le produit tout au long de la période de développement. Avec la méthode Scrum, le client est impliqué à chaque étape du projet pour valider le travail réalisé, ce qui correspond aux attentes de l'entreprise. La méthode Scrum permet

d'atteindre une meilleure satisfaction du client dans un temps précis en utilisant des "sprints" et en s'adaptant aux changements qui peuvent survenir en cours de projet. Cette méthode assure également une flexibilité face aux changements plutôt que de suivre un plan prédéfini. L'entreprise Elco Solutions est consciente de l'importance de la rétroaction du client et Scrum répond parfaitement à cette exigence en encourageant la participation régulière du client.

1.7 Conclusion

Après une réflexion attentive, nous avons décidé de poursuivre notre projet avec Scrum BI. Ce choix facilitera la communication et la collaboration entre notre équipe et Elco Solutions, permettant un retour régulier et une adaptabilité aux exigences changeantes.

CHAPITRE 2

ANALYSE ET SPÉCIFICATION DES BESOINS

2.1 Introduction

Ce chapitre se concentrera sur l'examen des spécifications fonctionnelles et non fonctionnelles du projet. De cette manière, nous allons observer la mise en place du BACKLOG du projet. En outre, nous allons inclure une partie dans la spécification des outils de décision que nous utiliserons tout au long du projet.

2.2 Analyse des besoins

Nous allons maintenant expliquer les besoins fonctionnels et non fonctionnels de notre projet. Cette étape nous aide à définir clairement ce que notre solution doit réaliser.

2.2.1 Spécification des besoins fonctionnels

Les besoins fonctionnels présentent les actions spécifiques de notre projet :

- **L'importation des données :** Il est important de collecter les données d'exploitation de manière efficace en utilisant le processus ETL, qui consiste à récupérer les

données de diverses sources, à les filtrer et à les adapter pour qu'elles puissent être utilisées dans des prises de décisions.

- **Stockage et modélisation des données** : Les données sont structurées et regroupées, puis rendues disponibles dans l'entrepôt de données.
- **La restitution** : Connue sous le nom de reporting, consiste à présenter les informations enrichies de manière claire et lisible pour faciliter la prise de décision.
- **Un tableau de bord** : Qui permet de visualiser des rapports personnalisés Offrir la possibilité de consultation des rapports sur un portail web tout en respectant les droits d'accès.
- **Deploiement d'un site Web** : Offrir la possibilité de consultation des rapports sur un portail web tout en respectant les droits d'accès.

2.2.2 Spécification des besoins non fonctionnels

Les besoins non fonctionnels définissent comment le système doit réaliser ces capacités :

- **Performance** : Le système doit être réactif , avec des temps de chargement rapides des visualisation et des analyses.
- **Simplicité** : Interface utilisateur intuitive et facile à utiliser, permettant aux utilisateurs de naviguer sans formation préalable.
- **Fiabilité** : Haute disponibilité du tableau de bord , avec des mécanismes pour garantir la continuité de service.
- **Evolutivité** : Capacité à s'adapter à une augmentation du volume de données et à l'intégration de nouvelles sources sans perte de performance.
- **Convivialité** : Une expérience utilisateur agréable qui encourage l'interaction et l'exploration des données.

- **Sécurité** Sécurité : Protection robuste des données , avec des mesures de contrôle d'accès.

2.3 Le pilotage du projet avec Scrum

2.3.1 Identification de l'équipe

Comme c'est le cas pour tous les projets gérés selon la méthodologie Scrum, Notre équipe Scrum se compose de trois membres :

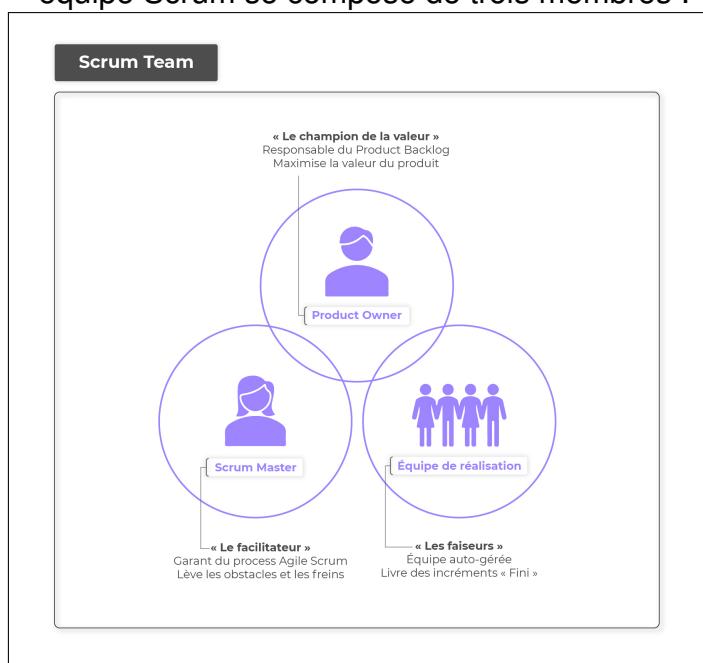


Figure 2.1: Équipe Scrum.

Table 2.1: Tableau d'équipe

Acteur	Rôle
Mr. Rami Taghouti	Product owner
Mme. Meriam Jemel	Scrum Master
Mr. Dhia Bouzidi	Team Member

2.3.2 Diagramme de Gantt

Le diagramme de Gantt est un outil efficace pour gérer et suivre l'avancement des travaux d'un projet. Grâce à sa vision claire et organisée de l'état d'avancement de chaque tâche, il facilite la coordination et la planification du projet dans son ensemble.

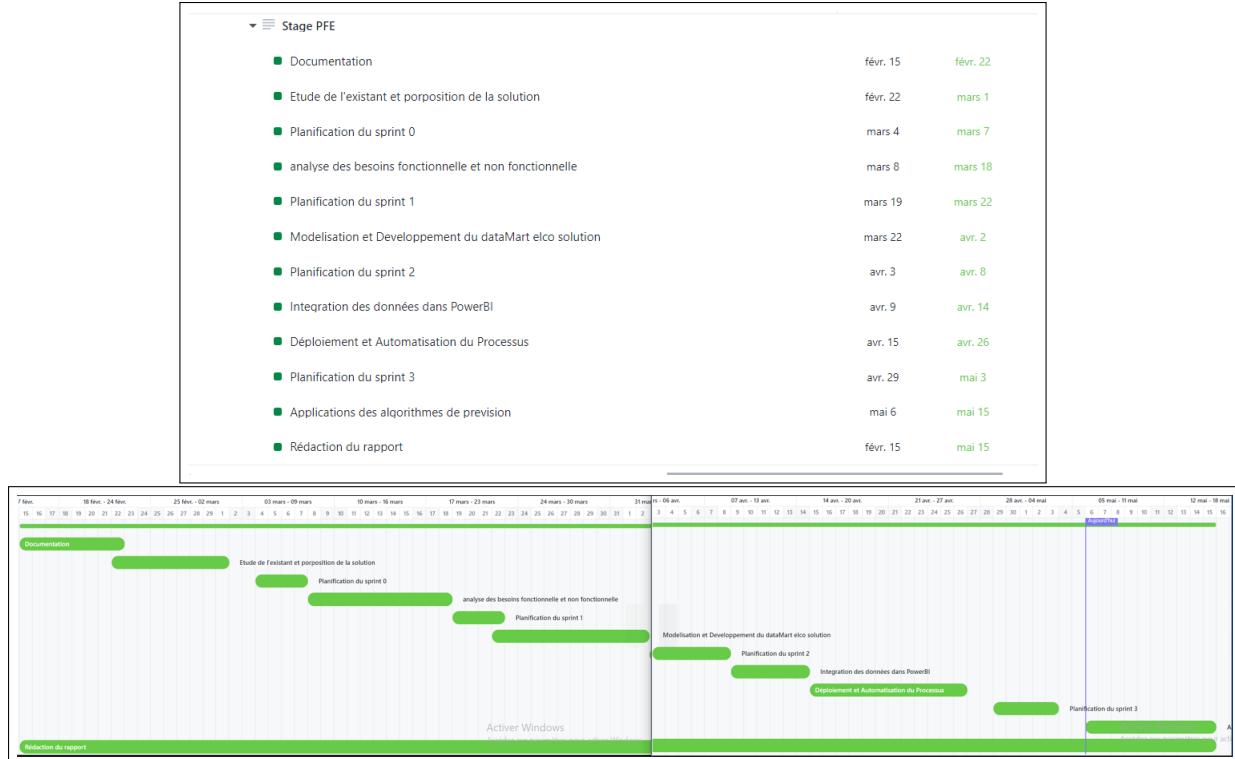


Figure 2.2: Diagramme de Gantt

2.3.3 Backlog de produit

Table 2.2: Backlog produit

ID	en tant que	user story	Priorité
1	Utilisateur	je veux disposer d'un datamart historique et unifié qui contient toutes les données détaillées des capteurs	Élevée
2	Utilisateur	je veux pouvoir extraire les données des capteurs	Élevée
3	Utilisateur	je veux préparer les données extraites pour l'analyse dans Power BI	Élevée
4	utilisateur	je veux accéder à un tableau de bord interactif dans Power BI	Élevée
5	utilisateur	Je veux des tableaux de bord complets qui regroupent les indicateurs de performance clés pour fournir une vue complète	Élevée
6	utilisateur	Je veux des interfaces utilisateur permettant d'accéder aux tableaux de bord et d'afficher les données des capteurs	Élevée
7	Utilisateur	je veux pouvoir filtrer les données dans le tableau de bord Power BI	Moyenne
8	Utilisateur	je veux savoir le capteur qui a le taux de consommation le plus élevé	Faible
9	Utilisateur	je veux savoir le capteur qui a le taux de consommation le moins élevé	Faible
10	Utilisateur	je veux savoir la moyenne de consommation de chaque metric	Faible
11	Utilisateur	je veux intégrer un modèle de machine learning pour prédire les valeurs futures des capteurs	Moyenne

2.4 Conception globale du DataWarehouse

2.4.1 Étude du modèle conceptuel

Avant de passer à la sélection des modèles, nous avons réalisé une étude comparative des différents modèles possibles. Cette étape est très importante car elle nous aide à choisir le modèle le plus adapté.

Modèle en étoile :

le modèle en étoile est une structure dimensionnelle utilisé dans la modélisation des entrepôts des données . Il présente une table de fait centrale entouré par ses différentes axes d'analyse : les dimensions . Techniquement les dimension sont référencés dans la table de fait par des clés étrangers.

Modèle en flocon de neige :

Le Modèle en flocon de neige est similaire au Modèle en étoile qui comprend des dimension en plus . Cependant, la principale différence est une hiérarchisation plus poussée des dimensions, qui sont organisées en fonction de la granularité des informations.

Modèle en constellation :

ce modèle est la combinaison entre plusieurs modèle en étoile , il comporte plusieurs table de fait qui partagent entre eux des dimension .

Choix du modèle :

Suite a l'étude des différents modèle de conception des entrepôts de données , Nous avons choisis le modèle en constellation car dans notre cas on trouve deux tables de faits qui partageant entre eux des dimensions en communs.

2.4.2 Conception Globale du dataWarehouse

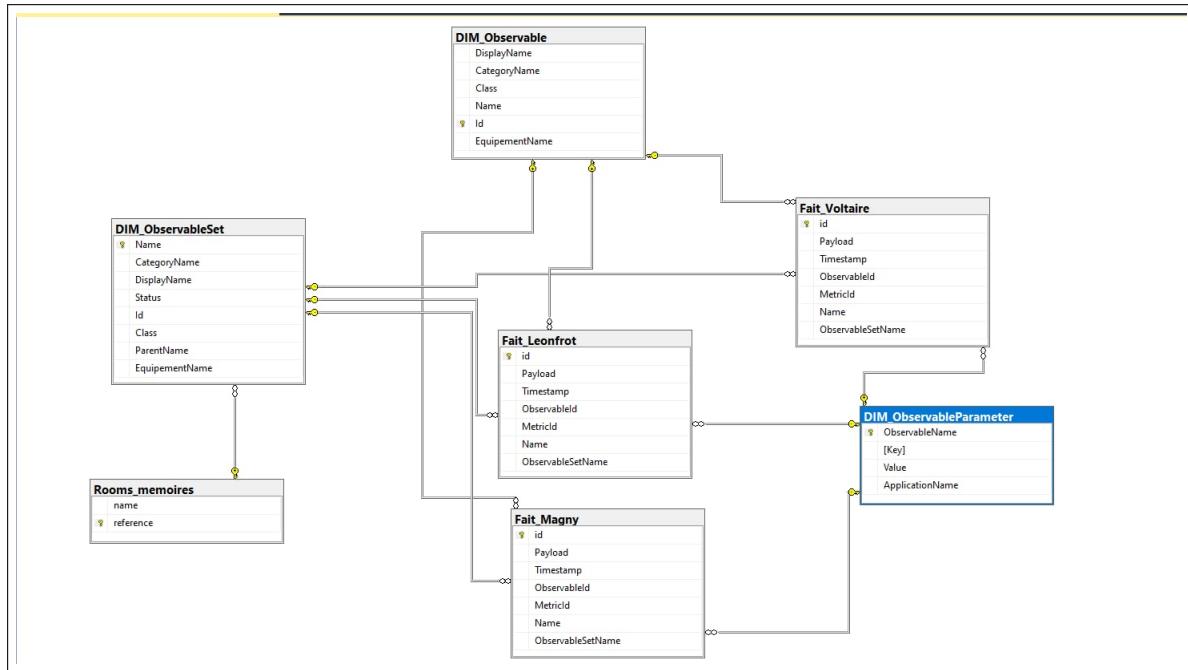


Figure 2.3: Schéma Conceptuelle de l'entrepôt de données

Pour Notre datawarehouse nous avons pu identifié 3 tables de faits :

- Table de fait Voltaire
- Table de fait Magny
- Table de fait LeonFrot

et plusieurs dimensions qui sont décrites dans le tableau suivant :

Table 2.3: Table de Dimensions

Dimension	Description
Dim ObservableSet	Cette dimension présente les informations des mémoires
Dim Observable	Cette dimension présente les informations liées aux capteurs
Dim ObservableParameter	Cette dimension présente les clé des capteurs
Dim RoomsMemoire	Cette dimension présente la relation entre les memoires , les batiments et les rooms

2.5 Environnement de travail

2.5.1 Environnement Matériel

Pour garantir le bon déroulement de toutes les activités du projet, nous avons été équipés d'un ordinateur Portable fourni par l'entreprise Elco Solutions . Les détails complets sur l'équipement matériel utilisé sont présentés dans le tableau ci-dessous :

Processeur	Core i5 23220U 3.00GHZ
RAM	8GB
Disque Dur	250GB SSD
Système d'exploitation	Windows 10 Professionnel 64-bit

2.5.2 Environnement Logiciel

Dès le début, notre organisation hôte, Elco Solutions, a nécessité l'utilisation de Talend et SQL Server Management Studio (SSMS) pour s'adapter à l'environnement technique existant. Cependant, nous étions libres de choisir l'outil de Business Intelligence (BI) pour la création des tableaux de bord.

Talend :

Talend Data Integration est une plateforme complète et puissante conçue pour faciliter l'intégration de données. Avec ses fonctionnalités avancées et sa convivialité, cette solution permet aux utilisateurs de connecter, transformer et charger efficacement des données provenant de diverses sources vers des destinations cibles. Grâce à son approche basée sur des graphes visuels et à sa bibliothèque de composants prédéfinis, Talend simplifie le processus d'intégration en réduisant la nécessité de coder manuellement.

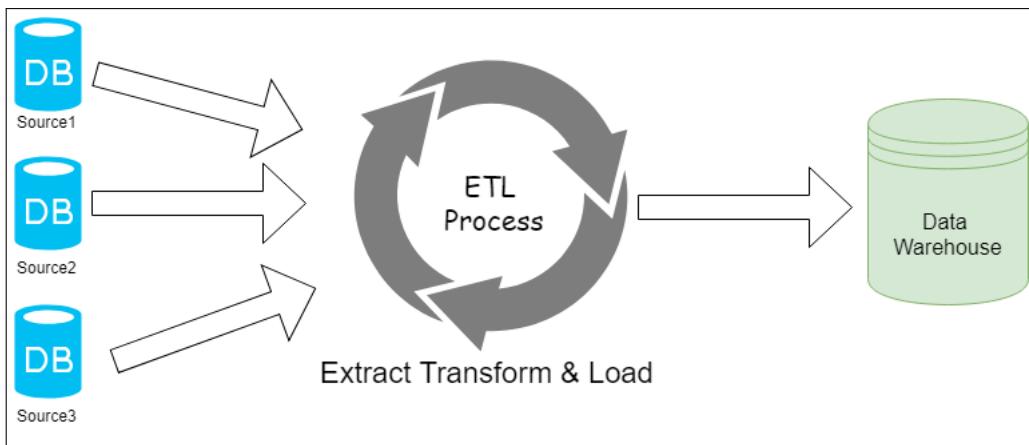


Figure 2.4: processus ETL du Logiciel Talend

SQL Server Management Studio :

SSMS est un environnement de développement intégré (IDE) robuste et convivial conçu spécifiquement pour la gestion et l'administration des bases de données SQL Server. Avec son interface utilisateur intuitive et ses fonctionnalités avancées, SSMS permet aux utilisateurs d'effectuer des tâches essentielles telles que l'écriture et l'exécution de requêtes SQL.



Figure 2.5: Logo SSMS

2.5.3 Etude comparative entre les outils BI

Power BI :



Figure 2.6: Logo PowerBI

Power BI est un outil de business intelligence complet développé par Microsoft qui offre aux entreprises la possibilité de visualiser leurs données provenant de diverses sources. Grâce à des capacités avancées de visualisation et d'analyse, Power BI permet aux utilisateurs d'explorer en profondeur leurs données, de créer des tableaux de bord interactifs et esthétiques et de partager facilement leurs rapports avec leurs collègues. En exploitant le potentiel de Power BI, les entreprises peuvent prendre des décisions plus éclairées, améliorer leur efficacité opérationnelle et acquérir un avantage concurrentiel significatif.

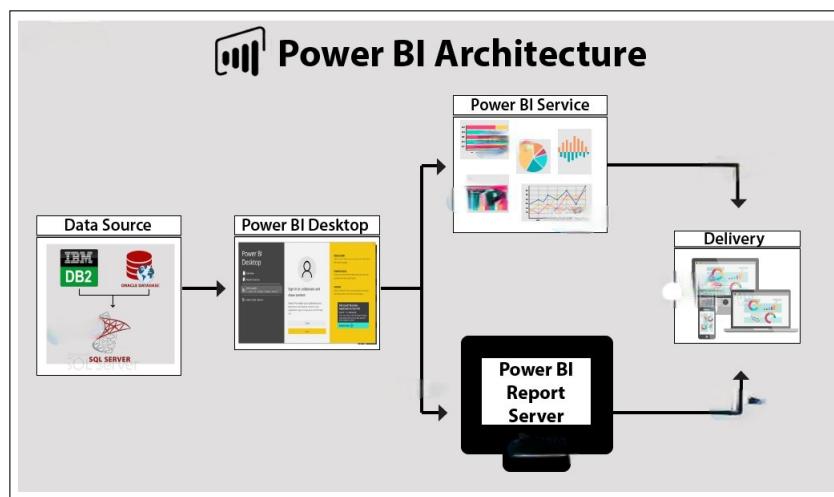


Figure 2.7: Architecture du PowerBI

Tableau :



Figure 2.8: Logo Tableau

Tableau est une puissante plateforme de business intelligence qui aide les entreprises à donner vie à leurs données. Grâce à une interface facile à utiliser, les utilisateurs peuvent créer des visualisations interactives et des tableaux de bord dynamiques pour explorer leurs données sous plusieurs perspectives. Tableau offre également des fonctionnalités avancées d'analyse et de traitement pour aider les utilisateurs à découvrir des informations précieuses et à prendre des décisions éclairées. Avec Tableau, les entreprises peuvent exploiter tout le potentiel de leurs données et bénéficier d'une meilleure compréhension de leurs opérations commerciales pour améliorer leurs performances et leur compétitivité.

Pentaho :



Figure 2.9: Logo Pentaho

Pentaho est une boîte à outils open source dédiée à la business intelligence. Il permet aux entreprises de collecter, gérer, analyser et visualiser efficacement leurs données. Pentaho permet aux utilisateurs d'accéder à des fonctionnalités avancées

qui leur permettent de traiter des données, d'effectuer des analyses personnalisées et de générer des rapports détaillés. L'avantage de Pentaho est sa capacité à s'adapter à différentes sources de données, ce qui offre aux utilisateurs une grande flexibilité. Cette plateforme permet aux entreprises d'exploiter pleinement leurs données pour prendre des décisions éclairées et améliorer leurs performances.

Choix de l'outil de travail :

Après cette recherche, nous avons choisi l'outil Power BI pour plusieurs facteurs. Premièrement, Power BI bénéficie d'une documentation complète et détaillée qui facilite son apprentissage et son utilisation. De plus, son interface conviviale et intuitive en fait un outil fluide et facile à utiliser. Enfin, la mise en œuvre de Power BI dans notre projet s'est avérée simple et rapide grâce à son étape de mise en œuvre simple à utiliser. Compte tenu de tous ces avantages, Power BI s'est avéré être la solution parfaite pour nos besoins de visualisation et d'analyse de données.

2.6 Conclusion

Ce chapitre était consacré à l'identification et à la définition des exigences fonctionnelles et non fonctionnelles de notre projet. Nous avons également défini des indicateurs clés de performance (KPI) pour guider notre analyse. De plus, nous avons choisi des outils appropriés pour nous aider à atteindre nos objectifs. Avec ces premières étapes, nous sommes maintenant prêts à aborder les prochains chapitres, qui se concentrent sur la conception de notre entrepôt de données.

CHAPITRE 3

CRÉATION DATAMART "ELCO MES"

3.1 Introduction

Après avoir effectué une analyse approfondie des exigences de l'application, ce chapitre détaillera les différentes étapes réalisées lors du premier sprint. Nous commencerons par le backlog du sprint. Ensuite, on va détailler la modélisation de ce service et finir par la phase d'implémentation.

3.2 Backlog de sprint

Le Sprint Backlog est un élément essentiel de la méthodologie Scrum qui représente les tâches sélectionnées pour le Sprint en cours. Il est créé lors de la réunion de planification du sprint et contient les tâches prioritaires à réaliser lors du sprint. Avec un Sprint Backlog, l'équipe de développement peut se concentrer sur les tâches à accomplir.

Table 3.1: Backlog du premier sprint

User Story	Sprint	Description
En tant que personne qui opte pour utiliser l'application je veux voir une datamart bien structuré	1	Identification des indicateurs clés de performances (KPI)
	2	Choix du modèle dimensionnel
	3	Conception du datamart 'Elco MES'
	4	Modélisation Multidimensionnelle
	5	Intégration des données

3.3 Identification des indicateurs de performance du datamart 'Elco MES'

La sélection d'indicateurs clés de performance (KPI) joue un rôle central dans le processus de développement d'un projet de business intelligence (BI). Cette étape cruciale permet une évaluation plus efficace du projet et la mise en œuvre des améliorations nécessaires pour atteindre les objectifs fixés.

Le tableau ci-dessous présente de manière organisée la clé que nous avons sélectionnée.

Table 3.2: description des indicateurs du datamart Elco MES

Indicateur	Description
Nombre des capteurs	Nombre totales des capteurs par batiments , etage , Chambre , Armoire
Les résultats des capteurs	Résultat maximale , minimale , moyenne des capteurs par Batiments , Etage , Chambre , Armoire
Nombre des armoires	Nombre des armoires par Batiments , Etage

3.4 Étude du modèle dimensionnel

La planification étant cruciale dans la préparation de l'entrepôt de données, avant d'autres mesures, il est nécessaire de choisir un modèle de mesure adapté à nos données et correspondant aux objectifs que nous nous sommes fixés. En ce sens, trois modèles sont possibles : modéliser une étoile, un flocon de neige ou une constellation.

3.4.1 Modèle en Étoile

Le modèle en étoile se distingue par une structure qui met l'accent sur un tableau de faits représentant un processus métier spécifique. Cette table de données est directement liée aux tables de dimensions qui représentent les différentes tendances d'analyse du processus. Ce modèle se caractérise par la simplicité des relations, ce qui facilite la création de rapports clairs et concis.

Avantage :

- Simplicité : Le modèle étoile est connu pour sa simplicité et sa facilité de lecture.
- performances : offre de bonnes performances grâce à une structure simplifiée et des jointures directes, réduisant le temps de réponse aux requêtes.
- Rapports simples : simplifie le reporting et l'analyse en regroupant les données clés dans un seul tableau de faits.
- Flexibilité : simplifie le reporting et l'analyse en regroupant les données clés dans un seul tableau de faits.

Inconvénient :

- Évolutivité limitée : l'évolutivité peut être limitée à long terme car elle devient difficile à gérer et à maintenir à mesure que le nombre de dimensions augmente.
- Gestion des mises à jour : peut être complexe, nécessitant de la précision dans les processus de téléchargement et de mise à jour des données.
- Redondance des données : car les dimensions sont répétées sur chaque ligne de la table de faits, ce qui peut augmenter le stockage requis.

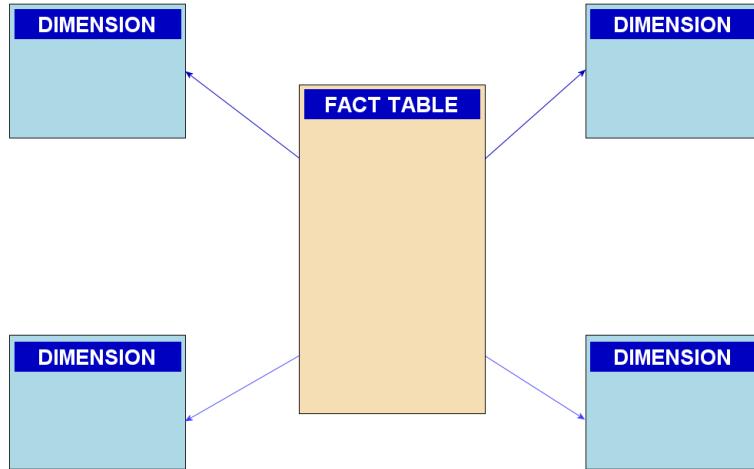


Figure 3.1: Modèle en Etoile

3.4.2 Modele en flocon

Le schéma des flocons de neige est une variante du schéma en étoile avec une différence importante. Cela permet de regrouper un ensemble d'attributs d'une même hiérarchie dans une autre table, créant ainsi des relations avec la table de dimensions. Cette façon de structurer les attributs offre l'avantage de gagner de l'espace de stockage, même si les requêtes peuvent être plus complexes par rapport à un schéma en étoile. Snowflake Schema fournit une méthode de normalisation qui peut être meilleure dans certaines situations, en fonction des besoins spécifiques et des contraintes de stockage du projet.

Avantage :

- Économise de l'espace : améliore la normalisation des attributs, réduit la redondance des données et l'espace de stockage utilisé.
- Autosimilarité : les motifs se répètent sous la forme d'un motif de flocon de neige à différentes échelles, créant des détails visuels passionnants en vue rapprochée et à vol d'oiseau.
- Évolutivité : le modèle de flocon de neige s'adapte à différentes échelles, permettant de créer différents motifs pour différents projets artistiques ou architecturaux.

Inconvénient :

- Complexité de conception. La création d'un modèle de neige nécessite des connaissances mathématiques et algorithmiques, sa conception est donc complexe et demande beaucoup de main-d'œuvre.
- Limites de variation : Les règles prédéfinies du flocon de neige peuvent limiter la variété et l'originalité des motifs créés, limitant ainsi les possibilités créatives.
- Temps et ressources informatiques : La génération de modèles de flocons de neige nécessite des ressources informatiques importantes, limitant l'accès aux utilisateurs disposant de ressources limitées.

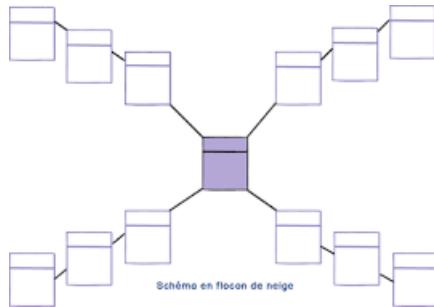


Figure 3.2: Modèle en Flocon

3.4.3 Modelé en Constellation

Ce modèle est constitué de plusieurs tableaux de données liés à des tableaux de dimensions, ce qui en fait une chaîne de plusieurs diagrammes en étoile ou en flocon de neige pour réduire l'espace de stockage et éliminer la redondance. Cela crée davantage d'opportunités pour créer des indicateurs de performance clés (KPI) basés sur des mesures supplémentaires tout en réduisant la redondance.

Avantage :

- Élimination de la redondance : le modèle de constellation réduit l'espace de stockage en éliminant la redondance des données en combinant des tables de faits et de dimensions.
- Flexibilité : le modèle Constellation permet des étoiles ou des flocons de neige concaténés, offrant une grande flexibilité pour répondre à des besoins d'analyse spécifiques.
- Facilité de maintenance : le modèle de constellation simplifie la gestion des données en réduisant la redondance et en consolidant les tables, les processus de chargement et de mise à jour sont plus faciles.

Inconvénient :

- Complexité initiale : la définition d'un modèle de constellation peut être difficile, nécessitant une planification minutieuse et des compétences techniques avancées qui peuvent prendre plus de temps.
- Plus difficile à comprendre : Comprendre et interpréter les relations entre les différentes tables d'un modèle de constellation peut être plus difficile pour les utilisateurs et les analystes. La complexité des graphiques rend la visualisation et l'analyse des données difficiles, nécessitant des connaissances approfondies pour une compréhension précise.

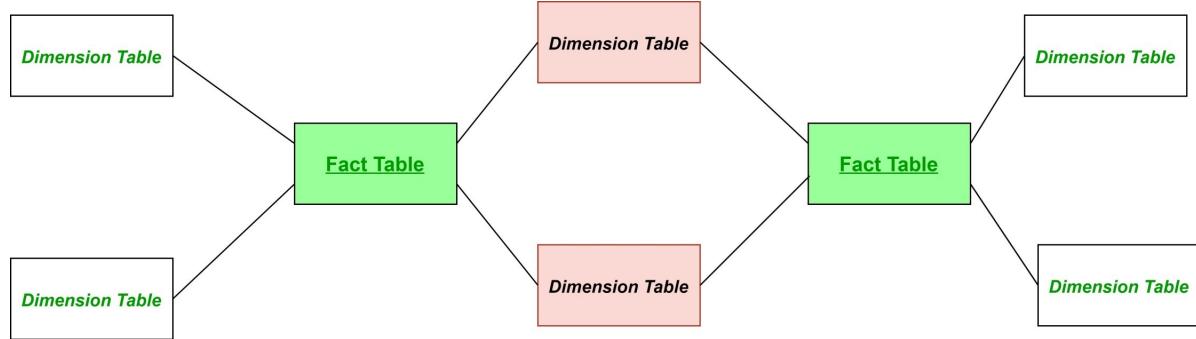


Figure 3.3: Modèle en Constellation

3.4.4 Choix du type de modèle

Dans le cadre de notre projet, nous avons choisi d'adopter un modèle décisionnel basé sur un schéma en Constellation. Cependant, compte tenu de nos axes d'analyse définis, nous avons identifié trois volets importants à étudier : Voltaire , Magny et LeonFrot . Afin de répondre à ces besoins spécifiques, il est devenu nécessaire d'avoir trois tables de faits. Par conséquent, notre structure de données finale pour l'entrepôt adopte une constellation de trois table de faits.

3.5 Conception du datamart ‘Elco MES’

3.5.1 Choix des dimensions

Les tables de dimensions jouent un rôle central dans notre modèle car elles contiennent les attributs clés de notre analyse. Avoir plusieurs dimensions dans notre modèle nous permet d'enrichir et de diversifier nos rapports. Les dimensions incluses dans notre modèle sont présenter dans le tableau ci-dessous :

Table 3.3: Les Dimensions

Dimension	Description
Dim ObservableSet	Cette dimension présente les informations des mémoires
Dim Observable	Cette dimension présente les informations liées aux capteurs
Dim ObservableParameter	Cette dimension présente les clé des capteurs
Dim RoomsMemoire	Cette dimension présente la relation entre les memoires , les batiments et les rooms

3.5.2 Choix des mesures

Pour créer nos table de faits, nous devons mieux connaître nos data afin de pouvoir mesurer les choses. Les métriques vous aident à effectuer des calculs basés sur vos données lorsque vous interagissez avec les rapports. Comme indiqué dans le tableau, nous avons choisi les mesures appropriées :

Table 3.4: Les mésures des table de fait

Mesure	Description
ID	cette mesure présente le clé primaire de la table
Payload	cette mesure présente les résultats des capteur
TimeStamp	cette mesure présente la date et l'heure
ObservableID	cette mesure présente l'ID des observables
MetricID	cette mesure présente l'identifiant des valeurs des capteurs
Name	Cette dimension présente la relation entre les memoires , les batiments et les rooms
ObservableSetName	Cette dimension présente la relation entre les memoires , les batiments et les rooms

3.5.3 Table de faits

La table de fait comprend des clés étrangères qui font référence aux dimensions associées à cette table, ainsi que des mesures sur lesquelles reposent les différentes analyses.

Fait_Magny	
!	id
	Payload
	Timestamp
	ObservableId
	MetricId
	Name
	ObservableSetName

Table 3.5: Table de fait Magny

Fait_Leonfrot	
!	id
	Payload
	Timestamp
	ObservableId
	MetricId
	Name
	ObservableSetName

Table 3.6: Table de fait LeonFrot

Fait_Voltaire	
!	id
	Payload
	Timestamp
	ObservableId
	MetricId
	Name
	ObservableSetName

Table 3.7: Table de fait Voltaire

Les Clés étrangères des tables dimensions :

- ObservableId
- MetricId
- Name
- ObservableSetName

Les Mesures :

- ID
- Payload
- Timestamp

3.6 Alimentation du data mart ‘Elco MES’

La phase ETL est vraiment importante et surtout nécessaire. Il s'agit d'un processus essentiel en business intelligence (BI) qui consiste à prendre des données brutes provenant de diverses sources, à les transformer dans un format cohérent et structuré, puis à les télécharger vers un entrepôt de données ou un système cible. Ce processus agrège, nettoie et intègre les données afin qu'elles soient prêtes à être analysées et utilisées dans des rapports, des tableaux de bord ou d'autres applications d'analyse de données.

Extraction Des données :

Obtention de données à partir de plusieurs sources de données telles que des bases de données, des fichiers plats ou des API

Transformation des données :

implique le traitement et la transformation des données extraites pour les rendre cohérentes, compatibles et adaptées à l'usage final

Chargement des données :

consiste de conversion de données prête et d'ajout à l'entrepôt de données ou au système cible.

Nous avons choisi le logiciel Talend pour définir le flux de traitement ETL.

3.6.1 Nos sources des données

Notre projet est d'une ampleur considérable et implique le traitement de trois sources de données différentes. Tout d'abord, nous avons utilisé des fichiers CSV pour déterminer quels capteurs étaient installés dans les bâtiments et lesquels dans les Chambres. Ces fichiers ont fourni une base importante pour comprendre la configuration de notre système de surveillance. Deuxièmement, nous avions un entrepôt de données dans SSMS appelé Elco Solution. Avec cette plateforme, nous avons pu séparer claire-

ment les tableaux de dimensions, créant ainsi une structure solide pour une analyse ultérieure. Enfin, nous avons travaillé avec des fichiers JSON contenant des données de capteurs. Plus de 40 millions d'enregistrements pour chaque bâtiment, totalisant 120 millions de lignes, étaient massifs et non structurés. Pour les rendre utilisables, nous avons développé un programme permettant de les découper et de les structurer correctement. Les fichiers JSON ont été convertis en plusieurs fichiers texte, chacun contenant 1 500 lignes pour un traitement géré. A noter que notre outil de traitement de données, Talend, était confronté à des fichiers JSON en raison de leur taille excessive, ce qui nécessitait une approche innovante pour surmonter ce défi technique.

```

-----
Directory: C:\NoSqlDataGeneral\LeonFrot
File: C:\NoSqlDataGeneral\LeonFrot\MV_2024_03_05_11_37_17_804002.bin
Size: 232841280 byte(s)
Start: 2024_03_05_11_37_03_277
End: 2024_03_25_23_02_24_934
File: C:\NoSqlDataGeneral\LeonFrot\MV_2024_02_19_16_33_56_561000.bin
Size: 53083464 byte(s)
Start: 2024_02_19_16_34_41_922
End: 2024_02_28_16_06_00_753
-----
[{"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2606, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2609, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2612, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2615, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2618, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2621, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2624, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2630, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2206, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2633, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2209, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2636, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2212, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2639, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2215, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2642, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2218, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2645, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2221, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2648, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2224, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2649, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2230, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2605, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2245, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2248, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2611, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2249, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 50103, "Context": null, "MetricId": 5, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2614, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}, {"Type": "Double", "Timestamp": "2024.02.19 16:34:41.922", "ObservableId": 2617, "Context": null, "MetricId": 1, "MetricIndex": 1, "Payload": "null"}]
```

Figure 3.4: Source JSON

3.6.2 Phase ETL par VS CODE :

Notre première source de données provient des fichiers JSON. Ces données initialement non structurées doivent devenir exploitables. Voici une présentation de ces fichiers et de leur contenu.

A1

	Q	memoire	
1	11.01	11.01A	
2	11.01	11.01B	
3	11.02	11.02A	
4	11.03	11.03A	
5	11.03	11.03B	
6	11.04	11.04A	
7	11.04	11.04B	
8	11.05	11.05A	
9	11.05	11.05B	
10	11.08	11.08A	
11	11.1	11.10A	
12	11.1	11.10B	
13	11.11	11.11A	
14	11.11	11.11B	
15	11.11	11.11C	
16	11.11	11.11D	
17	11.13	11.13A	
18	11.13	11.13B	
19	11.15	11.15A	
20	11.16	11.16A	
21	11.16	11.16B	
22	11.16	11.16C	
23	11.16	11.16D	
24	11.17	11.17A	
25	11.18	11.18A	
26	11.18	11.18B	
27	11.19	11.19A	
28	11.2	11.20A	
29	11.2	11.20B	
30	12.01	12.01A	
31	12.01	12.01B	
32	12.02	12.02A	
33	12.03	12.03A	

Figure 3.5: Source CSV

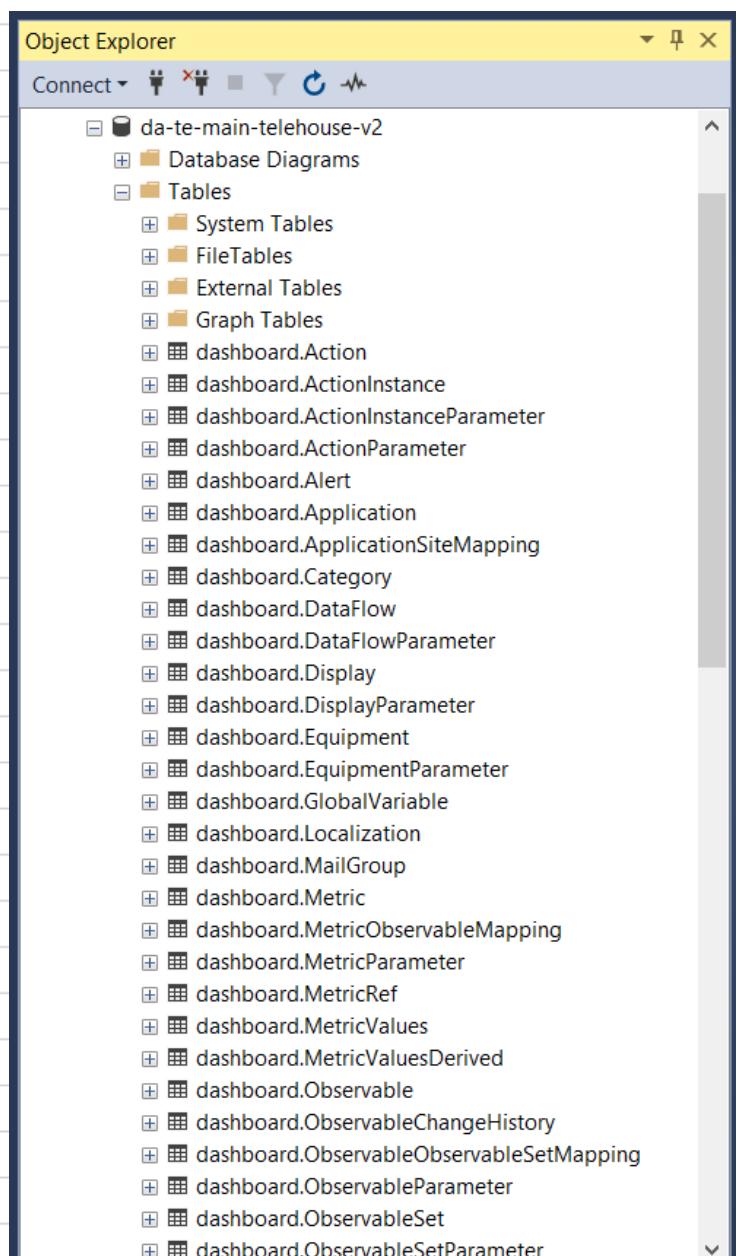


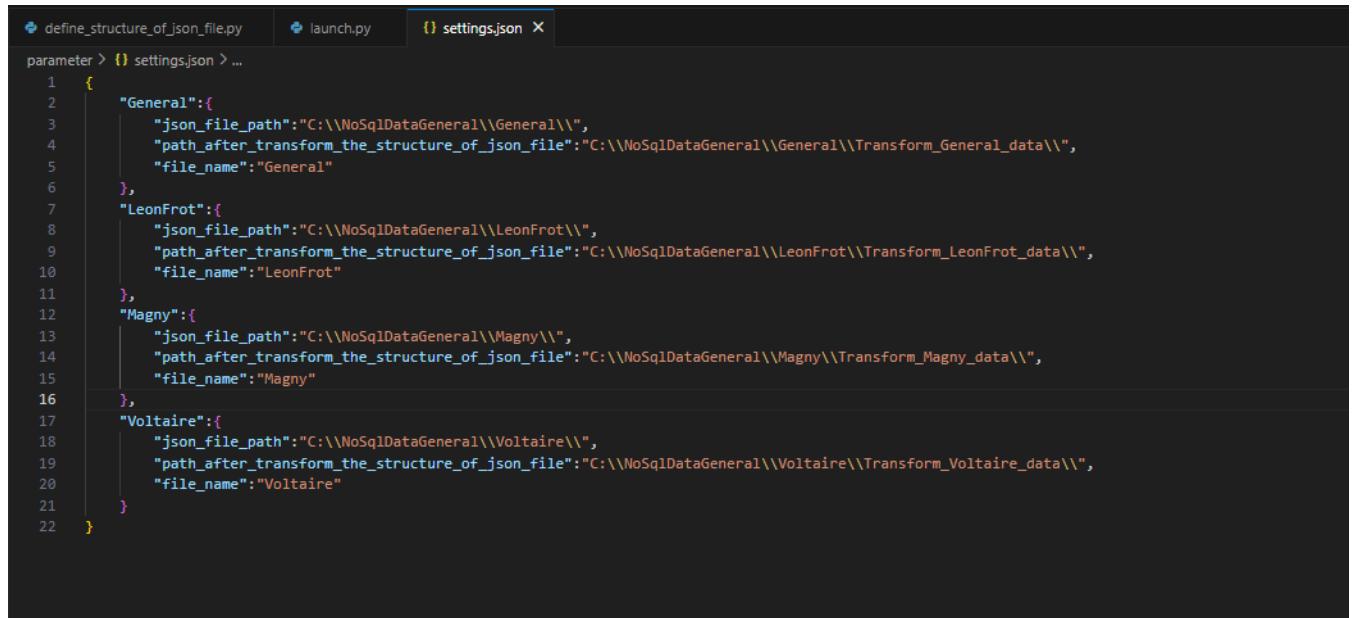
Figure 3.6: Source SSMS

Figure 3.7: Donnée Json avant transformation

3.6.2.1 Definition des paramètres :

Nous définissons d'abord les paramètres de ces fichiers comme indiqué ci-dessous.

Nous avons spécifié le chemin d'entrée, le chemin de sortie et les noms de fichiers



The screenshot shows a code editor with three tabs: 'define_structure_of_json_file.py', 'launch.py', and 'settings.json'. The 'settings.json' tab is active and displays the following JSON configuration:

```
parameter > settings.json > ...
1  {
2    "General": {
3      "json_file_path": "C:\\NoSqlDataGeneral\\General\\",
4      "path_after_transform_the_structure_of_json_file": "C:\\NoSqlDataGeneral\\General\\Transform_General_data\\",
5      "file_name": "General"
6    },
7    "LeonFrot": {
8      "json_file_path": "C:\\NoSqlDataGeneral\\LeonFrot\\",
9      "path_after_transform_the_structure_of_json_file": "C:\\NoSqlDataGeneral\\LeonFrot\\Transform_LeonFrot_data\\",
10     "file_name": "LeonFrot"
11   },
12   "Magny": {
13     "json_file_path": "C:\\NoSqlDataGeneral\\Magny\\",
14     "path_after_transform_the_structure_of_json_file": "C:\\NoSqlDataGeneral\\Magny\\Transform_Magny_data\\",
15     "file_name": "Magny"
16   },
17   "Voltaire": {
18     "json_file_path": "C:\\NoSqlDataGeneral\\Voltaire\\",
19     "path_after_transform_the_structure_of_json_file": "C:\\NoSqlDataGeneral\\Voltaire\\Transform_Voltaire_data\\",
20     "file_name": "Voltaire"
21   }
22 }
```

Figure 3.8: Etape 1

3.6.2.2 Creation d'un Script :

Dans cette étape, nous avons créé un script pour reformuler les données et les rendre bien structurées, en accord avec nos exigences. De plus, comme évoqué précédemment, nous avons découpé les données en fichiers texte contenant 1500 lignes chacun.

The screenshot shows a code editor interface with the following details:

- Menu Bar:** Go, Run, Terminal, Help.
- Toolbar:** Back, Forward, Search icon, Dev.
- Project Explorer:** Shows files: define_structure_of_json_file.py (selected), launch.py, settings.json, and Elc_solution > json_file_to_txt_file.
- Code Editor:**

```

1 import os
2
3 class Elc_solution:
4     def json_file_to_txt_file(input_path,output_path,file_name):
5         """JSON file's contents will be extracted, transformed, and structured, and sent to the TXT file by this method.
6         """
7         index = 1
8         count = 0
9         id = 1
10        Output_file = output_path + file_name + "_" + str(index)
11        #list all files in input_data directory
12        ALL_FILES_FROM_INPUT_DIR = os.listdir(input_path)
13        for file in ALL_FILES_FROM_INPUT_DIR:
14            #
15            if(file.endswith(".json")):
16                Input_file_Json = input_path + file
17                JSON_FILE = open(Input_file_Json, 'r+')
18                for line in JSON_FILE:
19                    if('{\"Type"' and "}" in line):
20                        with open(Output_file, 'a') as f:
21                            #Delete name of fields of json file
22                            line=line.replace("{\"Type\":\"", "")
23                            line=line.replace("","Timestamp":", ", ")
24                            line=line.replace(", \"ObservableId":', ", ")
25                            line=line.replace("Context":', ", ")
26                            line=line.replace(',"MetricId":', ", ")
27                            line=line.replace(',"MetricIndex":', ", ")
28                            line=line.replace(',"Payload":', ", ")
29                            line=line.replace('"}', ", -")
30                            payload = line.split(",")[-2]
31                            metricindex = line.split(",")[-3]
32                            if(payload.strip()!="null" and metricindex!="-1"):
33                                id_ = ", "+str(id)
34                                line = line.replace(",--",id_)
35                                id = id +1
36                                f.write(line)
37                                count += 1
38                                if(count==1500):
39                                    count=0
40                                    index+=1
41                                    Output_file = output_path + file_name + "_" +str(index)
42

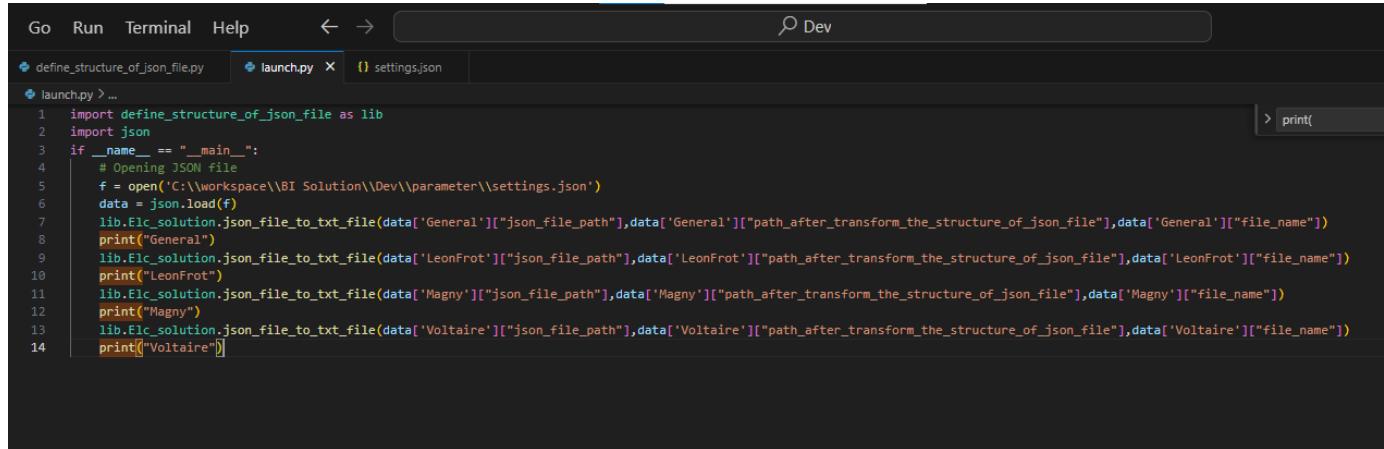
```

Figure 3.9: Etape 2

3.6.2.3 application du script et resultat :

Dans cette étape cruciale, nous avons mis en œuvre le script que nous avons développé pour reformuler nos données. Ce processus a nécessité une exécution méticuleuse pour garantir la transformation correcte des données non structurées en données bien organisées et adaptées à notre analyse. Chaque fichier JSON a été traité individuellement, avec une attention particulière portée à la précision et à l'intégrité des données. Une fois le script appliqué, nous avons vérifié la cohérence des résultats obtenus et nous nous sommes assurés que chaque fichier texte résultant comportait exactement 1500 lignes, conformément à notre stratégie de découpage.

Ce travail de transformation constitue une étape essentielle dans la préparation de nos données pour les étapes ultérieures de notre projet.



```

Go Run Terminal Help ← → ⚙ Dev
define_structure_of_json_file.py launch.py settings.json
launch.py > ...
1 import define_structure_of_json_file as lib
2 import json
3 if __name__ == "__main__":
4     # Opening JSON file
5     f = open('C:\\workspace\\BI Solution\\Dev\\parameter\\settings.json')
6     data = json.load(f)
7     lib.Elc_solution.json_file_to_txt_file(data['General'][ "json_file_path"],data['General'][ "path_after_transform_the_structure_of_json_file"],data['General'][ "file_name"])
8     print("General")
9     lib.Elc_solution.json_file_to_txt_file(data['LeonFrot'][ "json_file_path"],data['LeonFrot'][ "path_after_transform_the_structure_of_json_file"],data['LeonFrot'][ "file_name"])
10    print("LeonFrot")
11   lib.Elc_solution.json_file_to_txt_file(data['Magny'][ "json_file_path"],data['Magny'][ "path_after_transform_the_structure_of_json_file"],data['Magny'][ "file_name"])
12   print("Magny")
13   lib.Elc_solution.json_file_to_txt_file(data['Voltaire'][ "json_file_path"],data['Voltaire'][ "path_after_transform_the_structure_of_json_file"],data['Voltaire'][ "file_name"])
14   print("Voltaire")

```

Figure 3.10: Application du Script

Name	Date modified	Type	Size
□ Voltaire_1	5/2/2024 9:22 PM	File	249 KB
□ Voltaire_2	5/2/2024 9:22 PM	File	248 KB
□ Voltaire_3	5/2/2024 9:22 PM	File	250 KB
□ Voltaire_4	5/2/2024 9:22 PM	File	249 KB
□ Voltaire_5	5/2/2024 9:22 PM	File	250 KB
□ Voltaire_6	5/2/2024 9:22 PM	File	249 KB
□ Voltaire_7	5/2/2024 9:22 PM	File	250 KB
□ Voltaire_8	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_9	5/2/2024 9:22 PM	File	253 KB
□ Voltaire_10	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_11	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_12	5/2/2024 9:22 PM	File	253 KB
□ Voltaire_13	5/2/2024 9:22 PM	File	253 KB
□ Voltaire_14	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_15	5/2/2024 9:22 PM	File	253 KB
□ Voltaire_16	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_17	5/2/2024 9:22 PM	File	252 KB
□ Voltaire_18	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_19	5/2/2024 9:22 PM	File	252 KB
□ Voltaire_20	5/2/2024 9:22 PM	File	253 KB
□ Voltaire_21	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_22	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_23	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_24	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_25	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_26	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_27	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_28	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_29	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_30	5/2/2024 9:22 PM	File	255 KB
□ Voltaire_31	5/2/2024 9:22 PM	File	254 KB
□ Voltaire_32	5/2/2024 9:22 PM	File	254 KB

Figure 3.11: La division des fichiers

```
Double,2024.03.19 10:22:47.933,4419,null,4,1,0,4501
Double,2024.03.19 10:22:47.933,5834,null,1,1,246,4502
Double,2024.03.19 10:22:47.933,7044,null,1,1,251.63,4503
Double,2024.03.19 10:22:47.933,6662,null,1,1,253.33,4504
Double,2024.03.19 10:22:47.933,4420,null,4,1,0,4505
Double,2024.03.19 10:22:47.933,5837,null,1,1,246,4506
Double,2024.03.19 10:22:47.933,7047,null,1,1,251.63,4507
Double,2024.03.19 10:22:47.933,6665,null,1,1,253.33,4508
Double,2024.03.19 10:22:47.933,5840,null,1,1,246,4509
Double,2024.03.19 10:22:47.933,4421,null,4,1,0,4510
Double,2024.03.19 10:22:47.933,6668,null,1,1,253.33,4511
Double,2024.03.19 10:22:47.933,7050,null,1,1,251.63,4512
Double,2024.03.19 10:22:47.933,5843,null,1,1,246,4513
Double,2024.03.19 10:22:47.933,4422,null,4,1,0,4514
Double,2024.03.19 10:22:47.933,6671,null,1,1,253.33,4515
Double,2024.03.19 10:22:47.933,5846,null,1,1,246,4516
Double,2024.03.19 10:22:47.933,4423,null,4,1,0,4517
Double,2024.03.19 10:22:47.933,6686,null,1,1,253.33,4518
Double,2024.03.19 10:22:47.933,4424,null,4,1,0,4519
Double,2024.03.19 10:22:47.933,64124,null,1,3,246.21,4520
Double,2024.03.19 10:22:47.933,6689,null,1,1,253.33,4521
Double,2024.03.19 10:22:47.933,7004,null,1,1,245,4522
Double,2024.03.19 10:22:47.933,5644,null,3,1,3.01,4523
Double,2024.03.19 10:22:47.933,5851,null,1,1,246,4524
Double,2024.03.19 10:22:47.933,6692,null,1,1,253.33,4525
Double,2024.03.19 10:22:47.933,6252,null,1,1,256,4526
Double,2024.03.19 10:22:47.933,7007,null,1,1,245,4527
```

Figure 3.12: Resultat Finale

3.6.3 Phase ETL pour les Sources CSV par TALEND :

Job room memoire :

L'utilisation de la fonction tFileInputDelimited avec les paramètres définis dans Talend permet de lire efficacement le fichier CSV room-memoire. Les spécifications incluent :

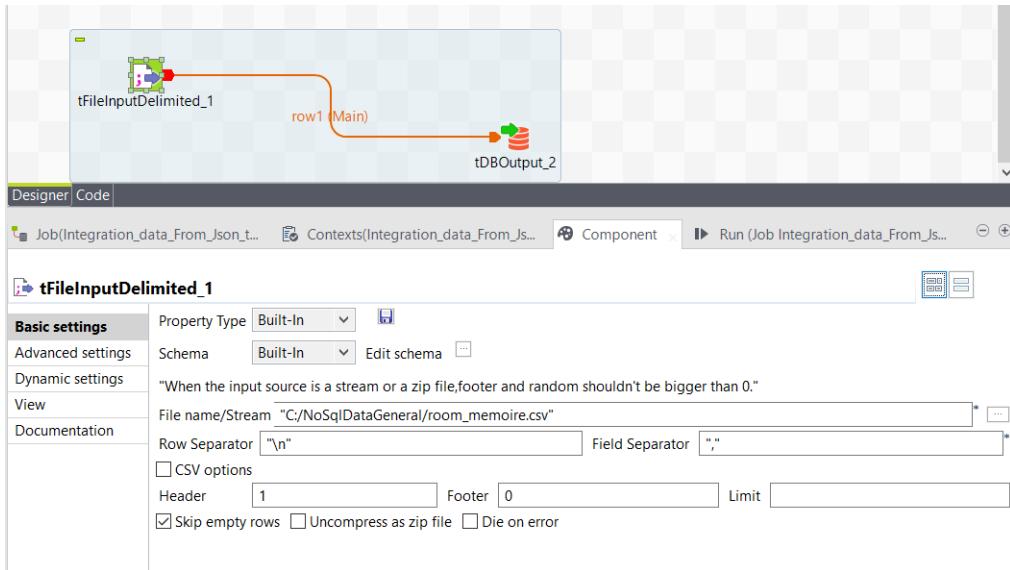


Figure 3.13: Etape 1

"Row Separator" est défini sur "" pour séparer les lignes par des sauts de ligne.

"Field Separator" est défini sur "," les champs sont séparés par des virgules.

"Header" est mis à 1 pour indiquer la présence d'un en-tête.

"Footer" est défini sur 0 pour indiquer l'absence de pied de page.

"Skip empty rows" est activé pour ignorer les lignes vides dans le fichier.

Ces paramètres garantissent une lecture précise des données, grâce à la structure particulière du fichier CSV, qui facilite l'importation des données pour un traitement ultérieur par Talend.

*Après avoir lu les données CSV, j'ai transféré les données vers une base de données SQL Server à l'aide du TDBOutput de Talend. Voici les paramètres utilisés :

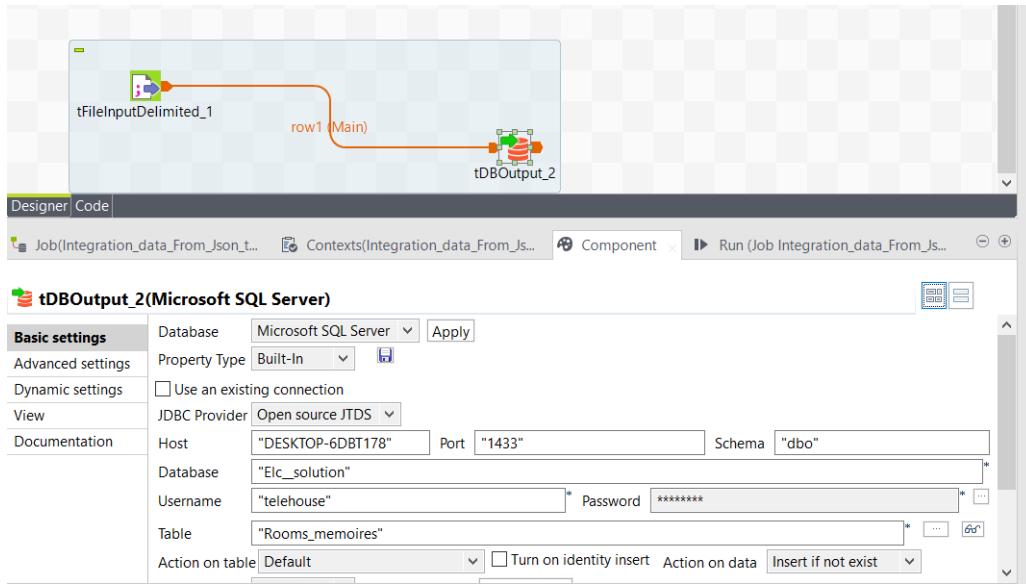


Figure 3.14: Etape 2

"Host" est défini sur 'DESKTOP-6DBT178' : Spécifie l'adresse IP ou le nom d'hôte de SQL Server.

"Port" est défini sur "1433" : Spécifie le port via lequel se connecter à SQL Server (généralement le port par défaut de SQL Server).

"Schema" est défini sur "dbo" : Spécifie le schéma de base de données SQL Server dans lequel la table sera créée.

"Base de données" est défini sur "elc solution" : Spécifie le nom de la base de données à laquelle se connecter.

"username" est défini sur "telehouse" : Spécifie le nom d'utilisateur pour se connecter à la base de données.

"Table" défini sur "room memoire" : Spécifie le nom de la table dans la base de données où les données seront insérées.

**Ces paramètres permettent à Talend de se connecter à la base de données SQL Server spécifiée et d'insérer les données lues dans le fichier CSV dans la table "room memoire". Cela garantit un transfert efficace des données de Talend vers la base de données SQL Server pour un traitement ultérieur.

"Résultat ou Output : "

```
SQLQuery8.sql - DE...on (telehouse (76))  X  SQLQuery7.sql - DE...
SELECT TOP (1000) [name]
,[reference]
FROM [Elc__solution].[dbo].[Rooms_memoires]
```

	name	reference
1	11.01	11.01A
2	11.01	11.01B
3	11.02	11.02A
4	11.03	11.03A
5	11.03	11.03B
6	11.04	11.04A
7	11.04	11.04B
8	11.05	11.05A
9	11.05	11.05B
10	11.08	11.08A
11	11.1	11.10A
12	11.1	11.10B
13	11.11	11.11A
14	11.11	11.11B
15	11.11	11.11C
16	11.11	11.11D
17	11.13	11.13A
18	11.13	11.13B
19	11.15	11.15A
20	11.16	11.16A
21	11.16	11.16B
22	11.16	11.16C
23	11.16	11.16D
24	11.17	11.17A
25	11.18	11.18A
26	11.18	11.18B
27	11.19	11.19A
28	11.2	11.20A
29	11.2	11.20B
30	12.01	12.01A
31	12.01	12.01B

Figure 3.15: Résultat

3.6.4 Phase ETL pour les Sources JSON par TALEND :

Architecture complète de txt file to ssms table VOLTAIRE :

Dans notre architecture Talend, nous avons conçu un flux de données qui traite les fichiers texte au lieu des fichiers CSV, puis charger ces données dans une base de données SQL Server. Nous avons commencé avec le tFileList pour lister les fichiers

disponibles dans un répertoire donné. Ensuite, nous avons utilisé le tFileInputDelimited pour lire chaque fichier texte individuellement, en définissant les paramètres appropriés pour son format de fichier, tels que les délimiteurs de lignes et de champs. Les données extraites étaient dirigées vers tMap, où diverses transformations pouvaient être appliquées si nécessaire, par exemple pour normaliser les données ou les enrichir avec d'autres sources. Enfin, nous avons utilisé le tDBOutput pour insérer les données transformées dans une table spécifique de notre base de données SQL Server. Ce composant définit tous les paramètres de connexion nécessaires tels que l'hôte, le port, le schéma, le nom de la base de données, l'identifiant et le nom de la table pour garantir une connexion appropriée à la base de données cible. Cette architecture garantit un traitement des données fluide et cohérent, du fichier source à la base de données, tout en offrant la flexibilité nécessaire pour répondre à différents formats de fichiers et exigences de conversion spécifiques.

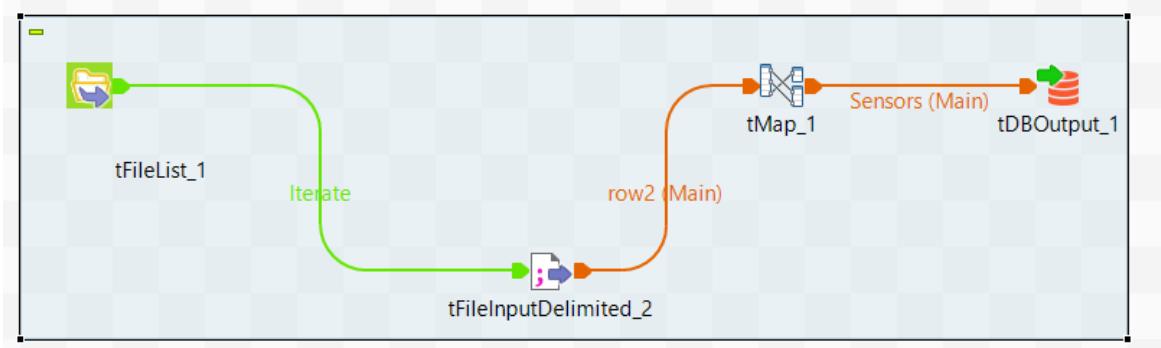


Figure 3.16: Architecture TALEND

tFilelist :

Dans notre architecture Talend, nous avons intégré un composant tFileList pour parcourir le répertoire spécifié "C:NosqlDatageneral/Voltaire/transform-voltaire-data" et lister les fichiers disponibles. Nous avons configuré ce composant avec les paramètres suivants :

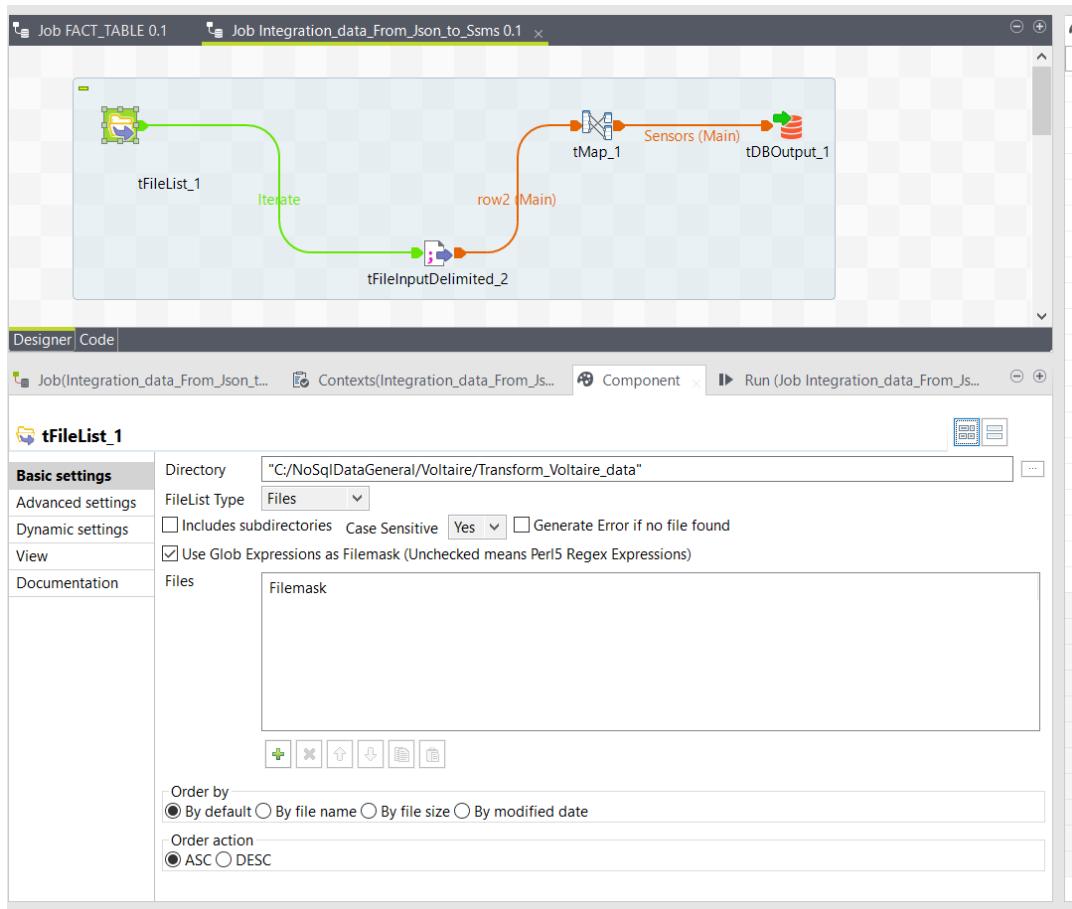


Figure 3.17: tFileList

"Files" : filmask a été configuré pour filtrer les fichiers selon un modèle spécifique, bien que ce détail ne soit pas mentionné.

"Order by" : Par défaut, les fichiers sont triés dans l'ordre spécifié.

"Order action" : Par défaut, la fonction de tri est définie sur ASC (croissant), ce qui signifie que les fichiers sont classés par ordre croissant par leur nom ou leur date de modification.

Cela permet à Talend de récupérer systématiquement tous les fichiers texte dans un dossier spécifique, ce qui facilite le traitement séquentiel de chaque fichier dans le flux de données.

tFileInputDelimited :

pour le composant tFileInputDelimited dans Talend, les paramètres ont été configurés comme suit :

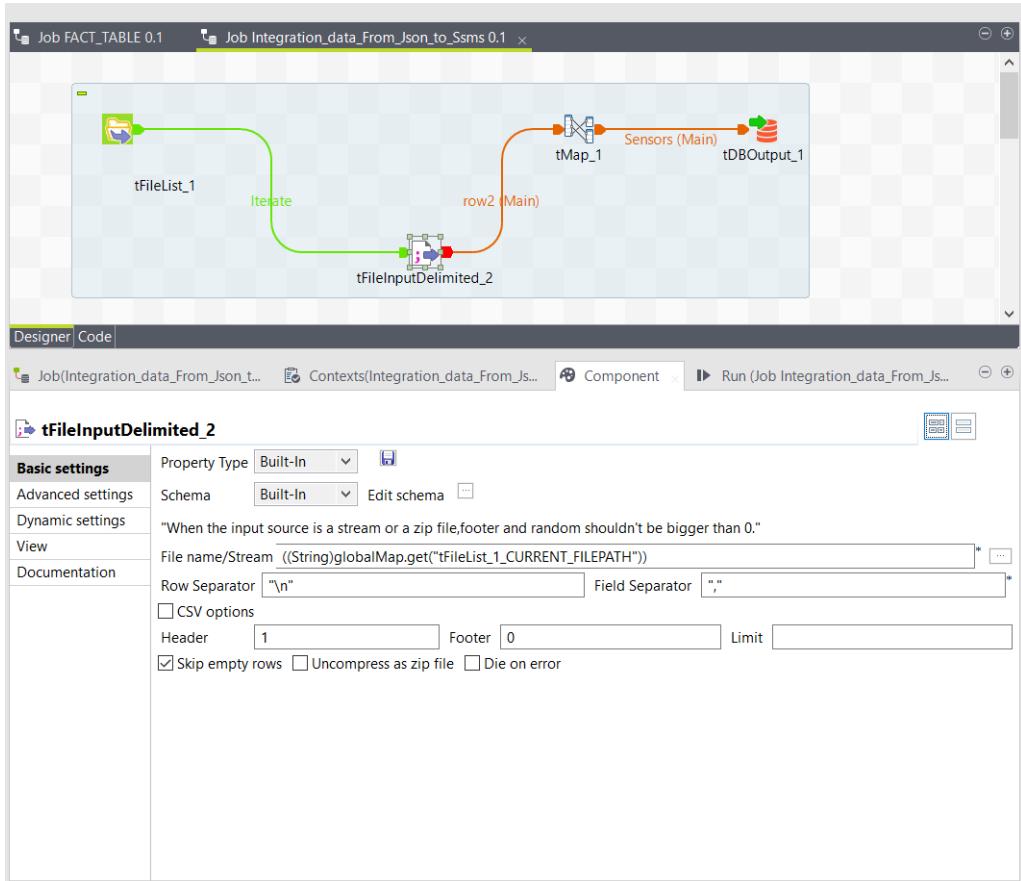


Figure 3.18: tFileinputDelimiter

"Row Separator" : "", indiquant que les lignes du fichier sont séparées par des sauts de ligne.

"Field Separator" : ",," , spécifiant que les champs à l'intérieur de chaque ligne sont séparés par des virgules.

"Header" : Présent, avec le paramètre défini sur 1 pour indiquer la présence d'une ligne d'en-tête.

"Footer" : Absent, avec le paramètre défini sur 0 pour indiquer l'absence de pied de page.

"Skip empty rows" : Activé, pour sauter les lignes vides dans les fichiers.

Tmap :

Le composant TMap du flux de données de talend est utilisé pour transformer les données extraites des fichiers texte. Nous avons configuré TMap pour exécuter diverses fonctions telles que la normalisation des données, le filtrage, l'agrégation de colonnes ou toute autre logique métier requise. Avec TMap, nous avons pu structurer les données selon nos besoins spécifiques et les préparer à être incluses dans une base de données SQL Server. Ce composant joue donc un rôle clé dans le traitement et la modification des données lors du processus d'intégration.

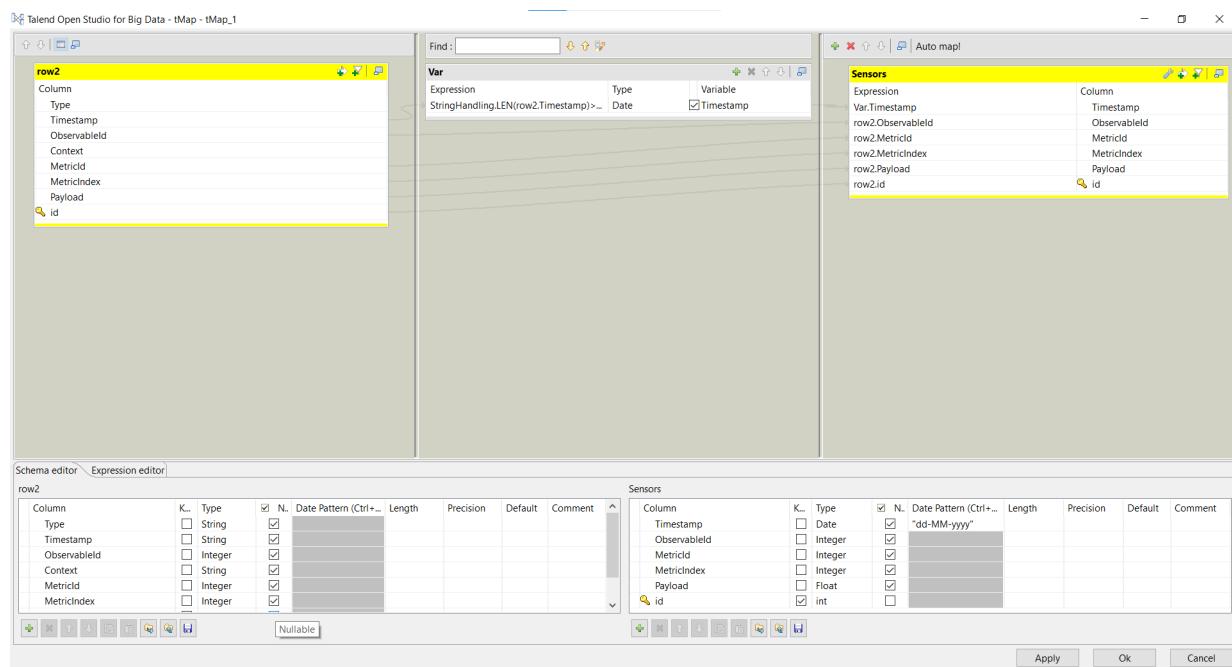


Figure 3.19: TMAP

Tdboutput :

Le composant tDBOutput est utilisé pour insérer les données transformées dans une

table spécifique de la base de données SQL Server. Dans notre cas, nous l'avons configuré pour se connecter à la base de données "elc solution" sur le serveur "DESKTOP-6DBT178" en utilisant l'utilisateur "telehouse". Nous avons défini la table cible "data from sensor voltaire" où les données seront ajoutées. Ce composant crée un lien puissant entre Talend et la base de données, permettant un transfert fiable des données converties pour un stockage permanent et une utilisation ultérieure.

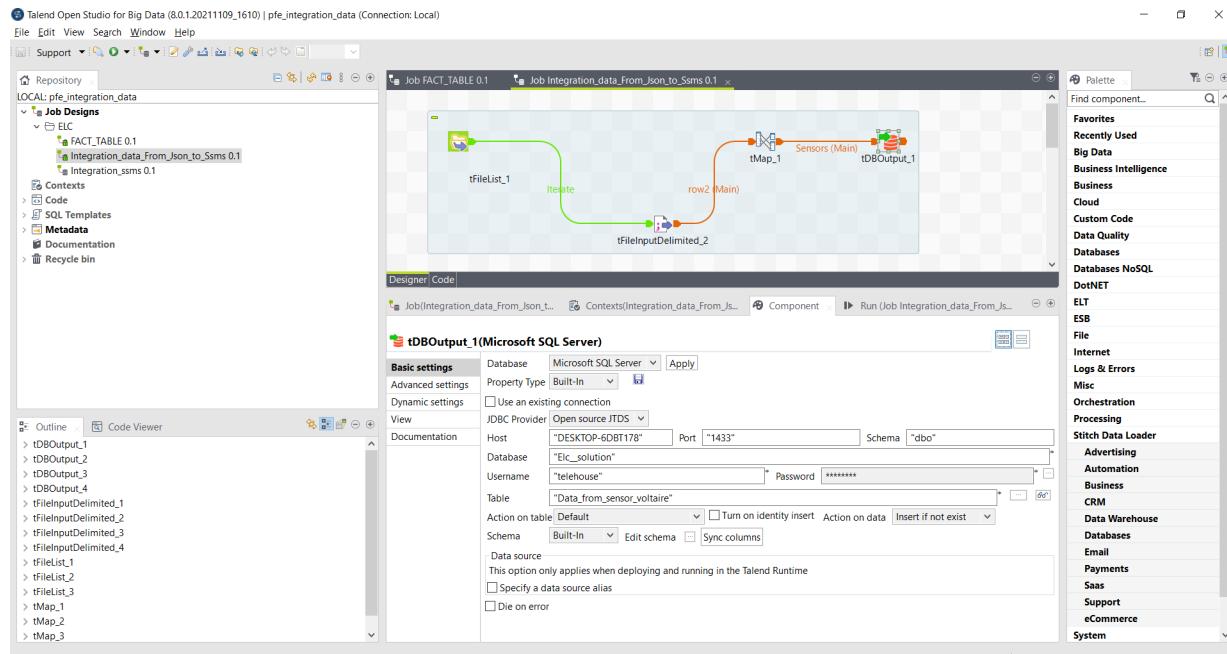


Figure 3.20: Load

Resultat Final :

Notre architecture Talent a finalement permis de convertir efficacement les données à partir de fichiers texte, puis de les insérer avec précision dans une base de données SQL Server. Ce processus garantit que les données sont désormais disponibles pour une analyse plus approfondie dans une structure organisée et facilement accessible. Grâce à cet arrangement, nous avons créé un flux de travail robuste et automatisé qui garantit l'intégrité et la disponibilité des données pour les besoins actuels et futurs de notre projet.

SQLQuery1.sql - DE...on (telehouse (64)) X DESKTOP-6DBT178...ution - Diagram_0*

```

SELECT TOP (1000) [Timestamp]
      ,[ObservableId]
      ,[MetricId]
      ,[MetricIndex]
      ,[Payload]
      ,[id]
  FROM [Elc_solution].[dbo].[Data_from_sensor_voltaire]

```

100 % ▾

Result Messages

	Timestamp	ObservableId	MetricId	MetricIndex	Payload	id
1	2024-03-19	19706	1	1	256.160003662109	2
2	2024-03-19	10309	1	1	242.529998779297	3
3	2024-03-19	10106	1	1	257.510009765625	4
4	2024-03-19	19709	1	1	256.160003662109	5
5	2024-03-19	50006	1	1	241.789993286133	6
6	2024-03-19	19712	1	1	256.160003662109	7
7	2024-03-19	10109	1	1	257.510009765625	8
8	2024-03-19	31706	1	1	246.690002441406	9
9	2024-03-19	10112	1	1	257.510009765625	10
10	2024-03-19	10115	1	1	257.510009765625	11
11	2024-03-19	31709	1	1	246.690002441406	12
12	2024-03-19	10118	1	1	257.510009765625	13
13	2024-03-19	31712	1	1	246.690002441406	14
14	2024-03-19	10121	1	1	257.510009765625	15
15	2024-03-19	31715	1	1	246.690002441406	16
16	2024-03-19	10124	1	1	257.510009765625	17
17	2024-03-19	10127	1	1	257.510009765625	18
18	2024-03-19	31718	1	1	246.690002441406	19
19	2024-03-19	31721	1	1	246.690002441406	20
20	2024-03-19	31724	1	1	246.690002441406	21
21	2024-03-19	31730	1	1	246.690002441406	22
22	2024-03-19	31733	1	1	246.690002441406	23
23	2024-03-19	31736	1	1	246.690002441406	24
24	2024-03-19	31739	1	1	246.690002441406	25
25	2024-03-19	31742	1	1	246.690002441406	26
26	2024-03-19	31745	1	1	246.690002441406	27
27	2024-03-19	31748	1	1	246.690002441406	28

Query executed successfully. | DESKTOP-6DBT178 (15.0 RTM) | telehouse (64) | Elc_solution | 00:00:00 | 1,000 rows

Figure 3.21: Resultat final

REMARQUE:

Je tiens à souligner que le processus ETL (Extract, Transform, Load) détaillé dans mon rapport de stage pour le traitement des fichiers JSON du bâtiment Voltaire est également applicable aux fichiers JSON des bâtiments Magny et LeonFrot. Par conséquent, afin d'éviter la redondance dans ce rapport, je ne répéterai pas en détail le processus déjà décrit , et je ferai référence que les fichiers json contiennent les résultats des capteurs qui seront implémentés dans une table dans ssms.

3.6.5 Phase ETL pour les Sources SSMS par TALEND :

3.6.5.1 Creation de la table DIM Observable :

JOB :

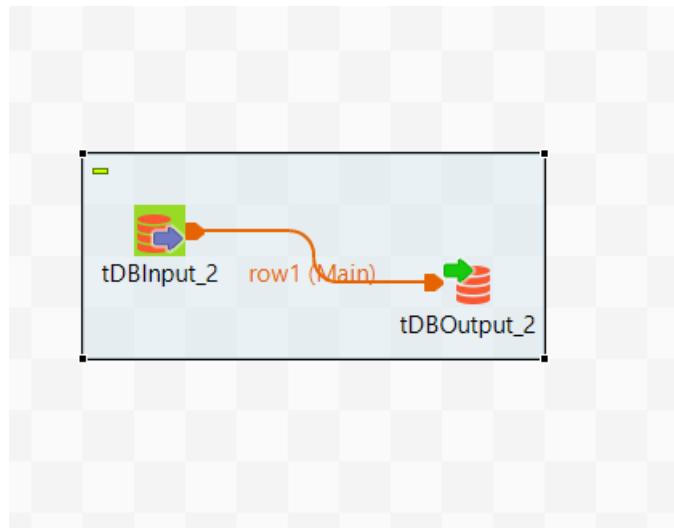


Figure 3.22: Etape 1

TDBInput :

Dans cette étape TDBInput récupère des données de la table "Observable" dans la base de données "da-te-main-telehouse-v2" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

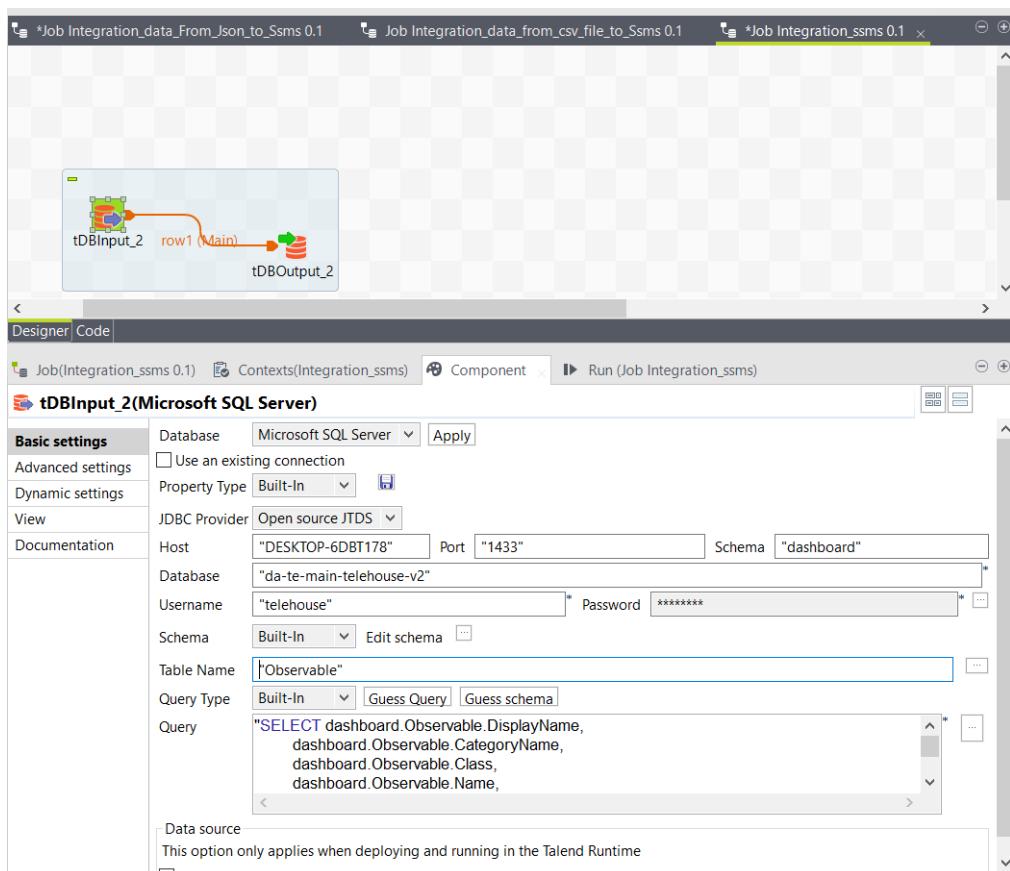


Figure 3.23: Etape 2

TDBOutput :

Dans cette étape TDBOutput écrit les données récupérées de la table "observable" dans la table "DIM Observable"

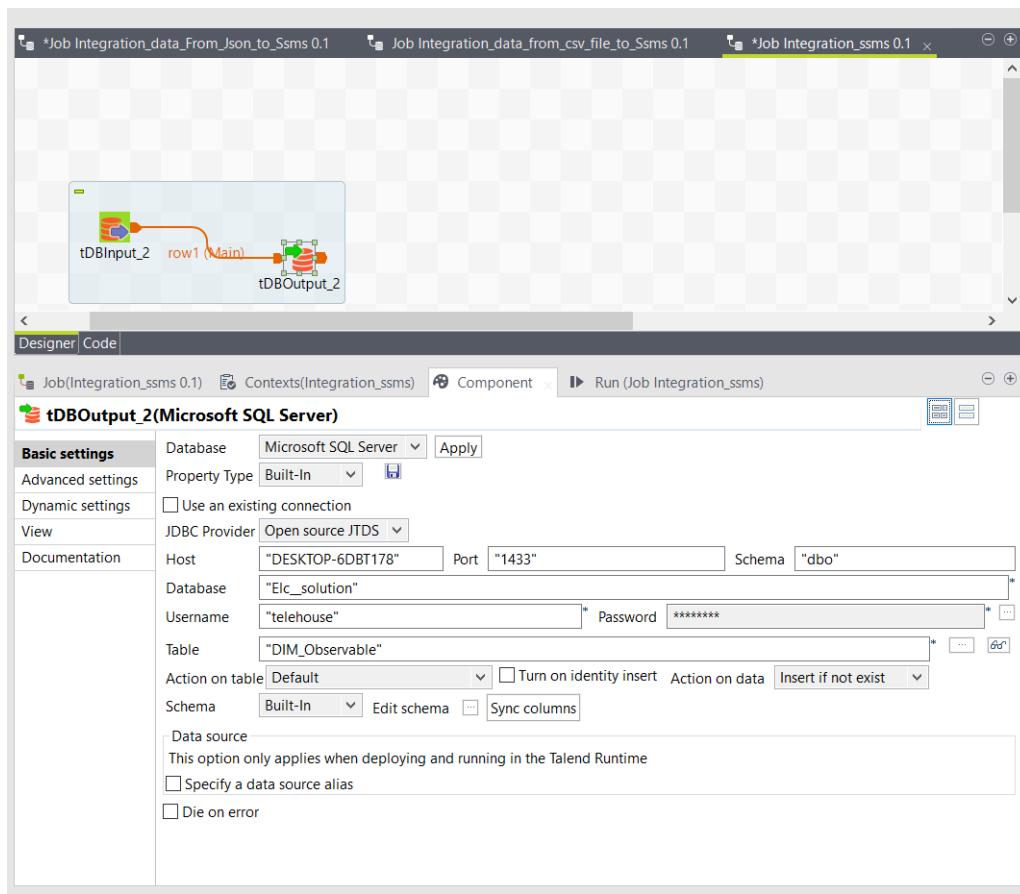


Figure 3.24: Etape 3

3.6.5.2 Creation de la table DIM ObservableSet :

JOB :

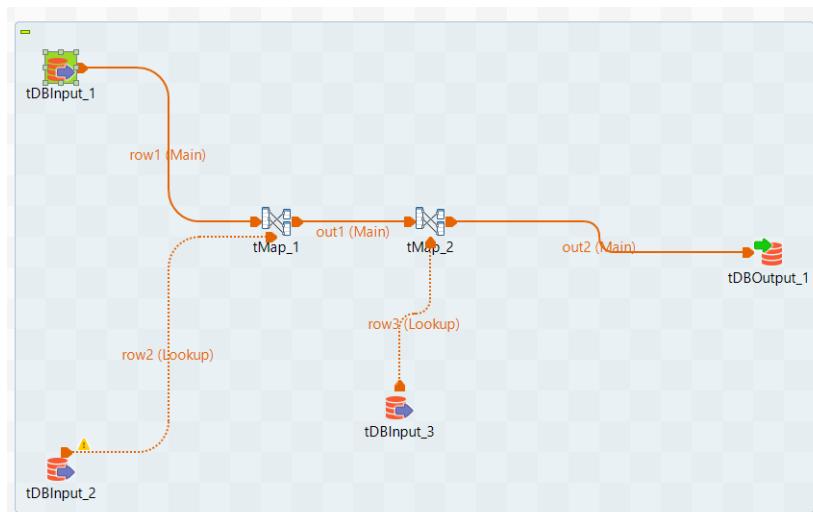


Figure 3.25: Etape 1

TDBInput :

Dans cette étape TDBInput récupère des données de la table "ObservableSet" dans la base de données "da-te-main-telehouse-v2" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

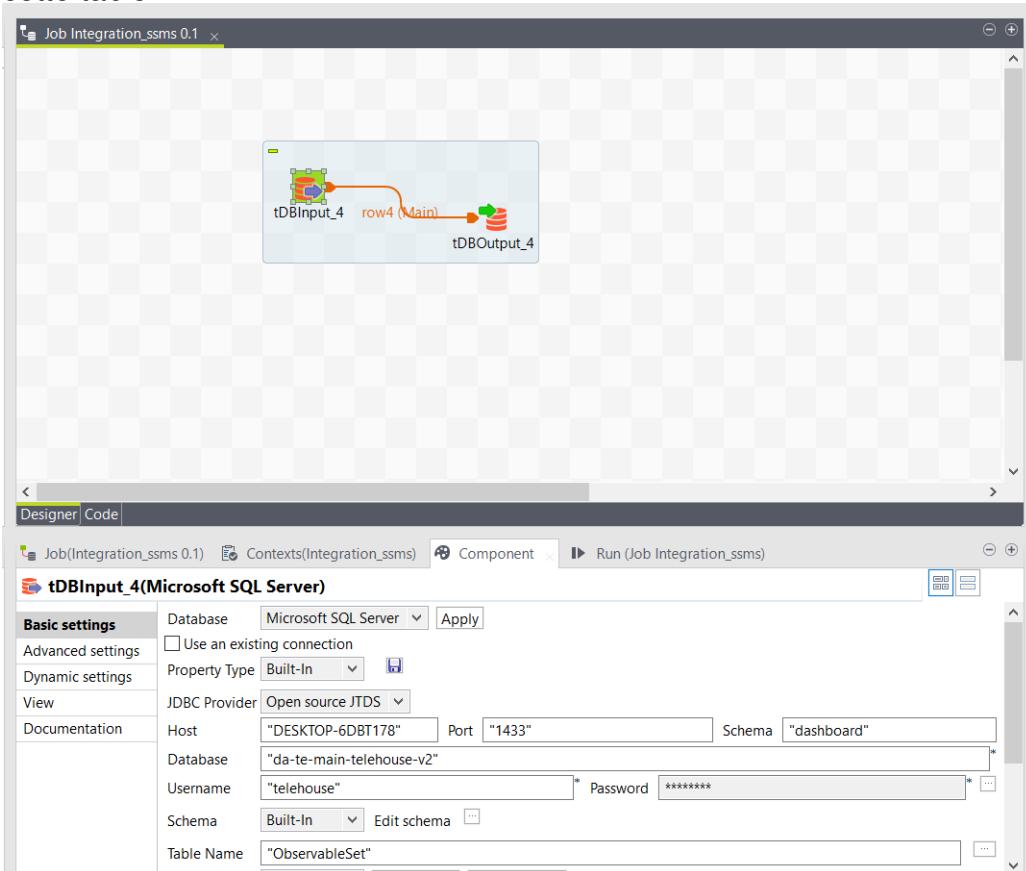


Figure 3.26: Etape 2

TDBOutput :

Dans cette étape TDBOutput écrit les données recuperer de la table "observableSet" dans la table "DIM ObservableSet"

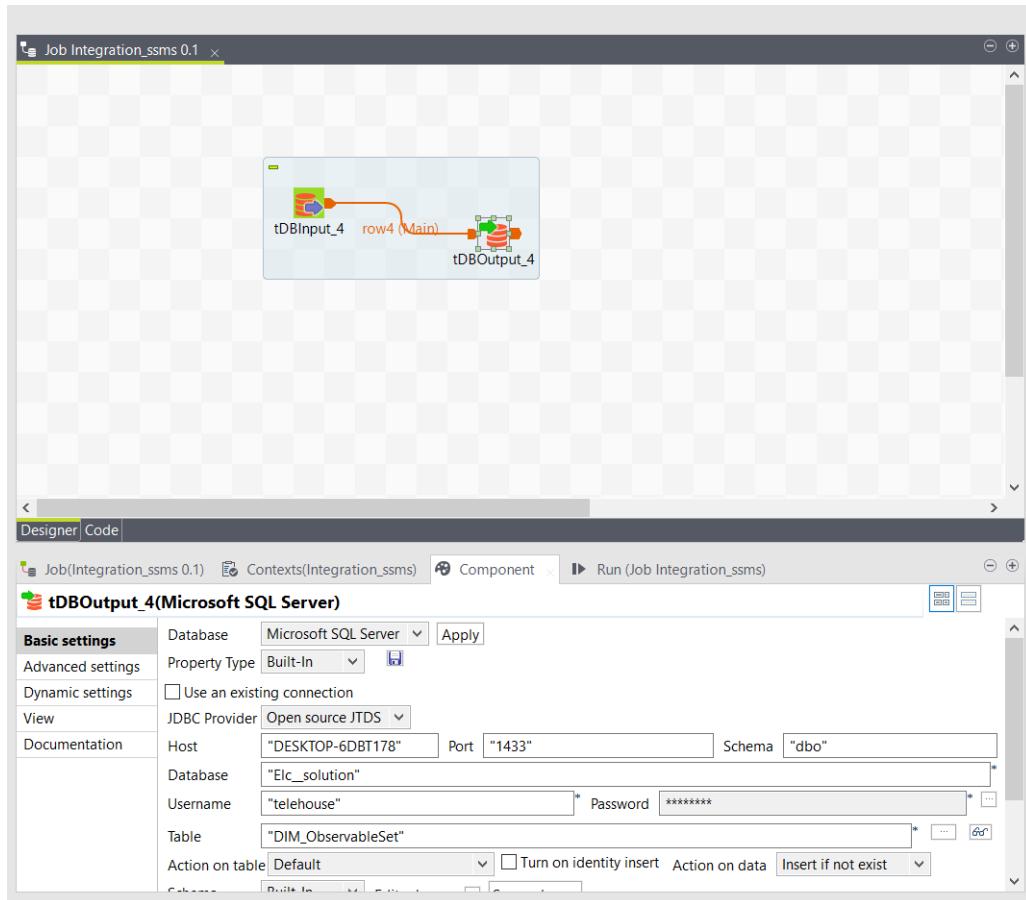


Figure 3.27: Etape 3

3.6.5.3 Creation de la table DIM ObservableParameter :

JOB :

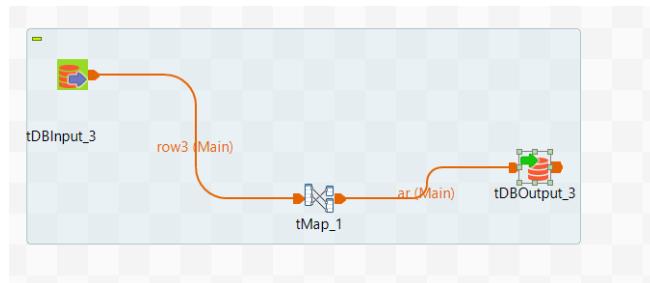


Figure 3.28: Etape 1

TDBInput :

Dans cette étape TDBInput récupère des données de la table "ObservableParameter"

dans la base de données "da-te-main-telehouse-v2" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

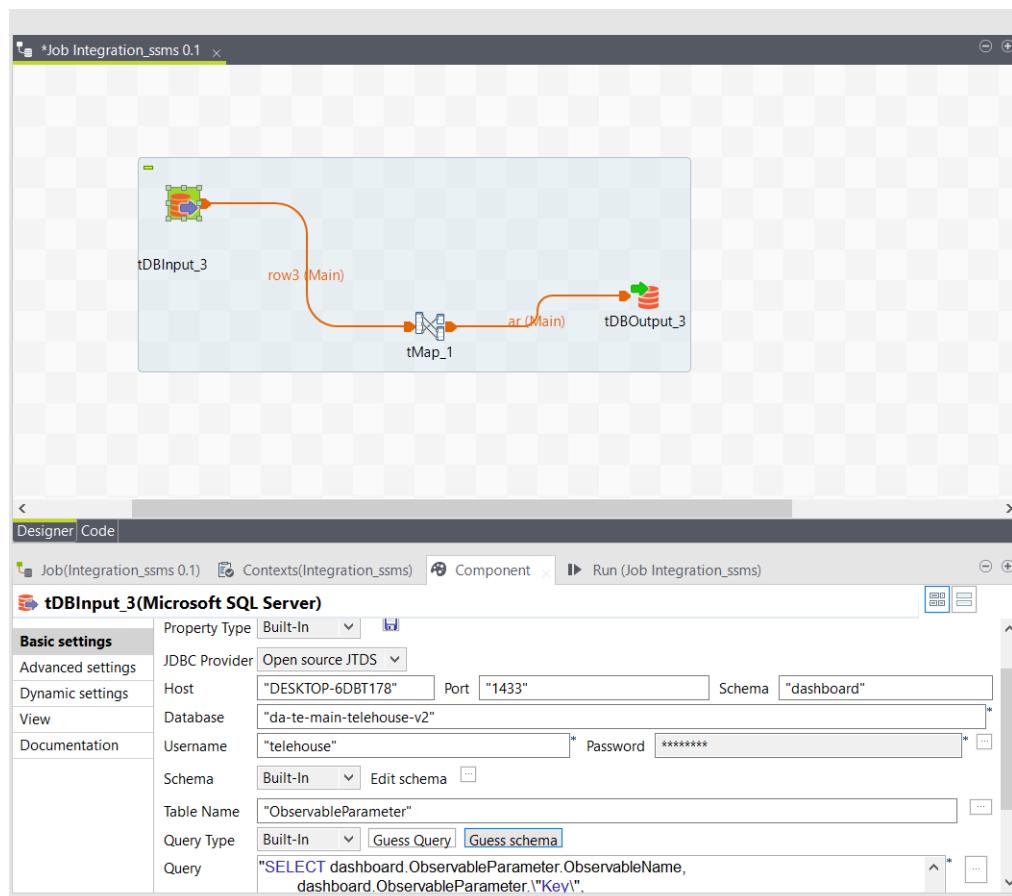


Figure 3.29: Etape 2

TMAP :

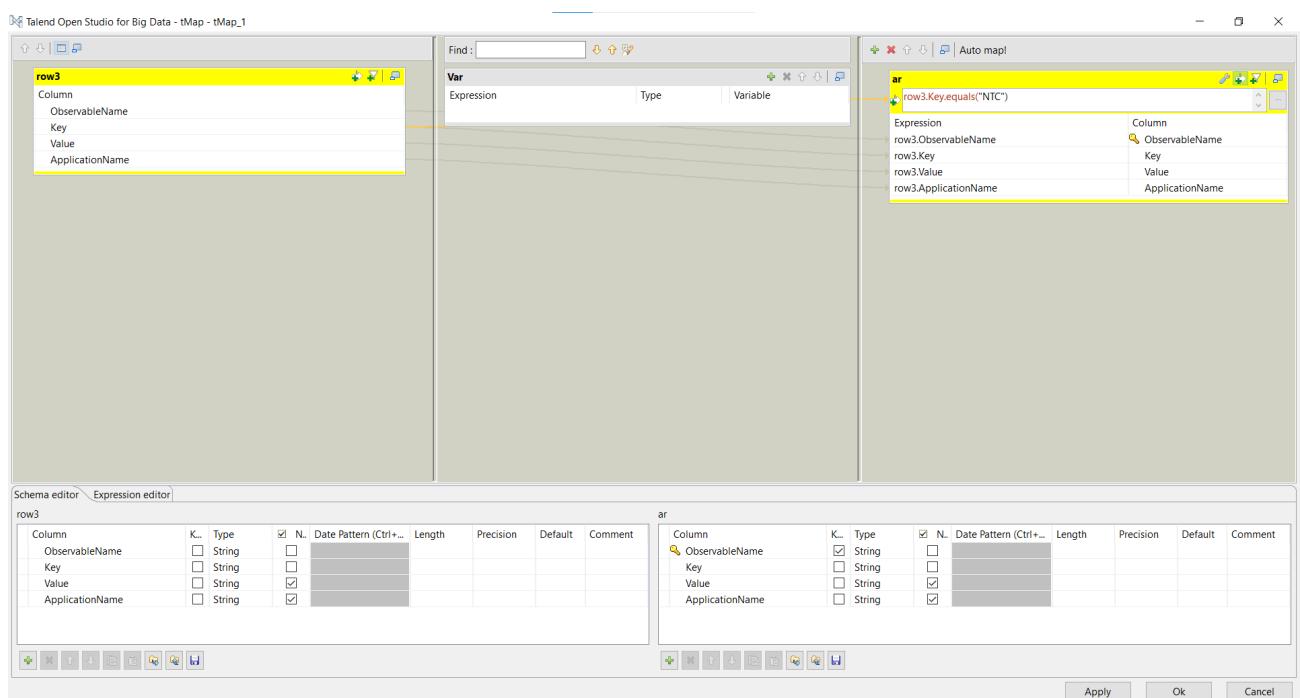


Figure 3.30: Etape 3

TDBOutput :

Dans cette étape TDBOutput écrit les données recuperer de la table "observableParameter" dans la table "DIM ObservableParameter"

[htbp]

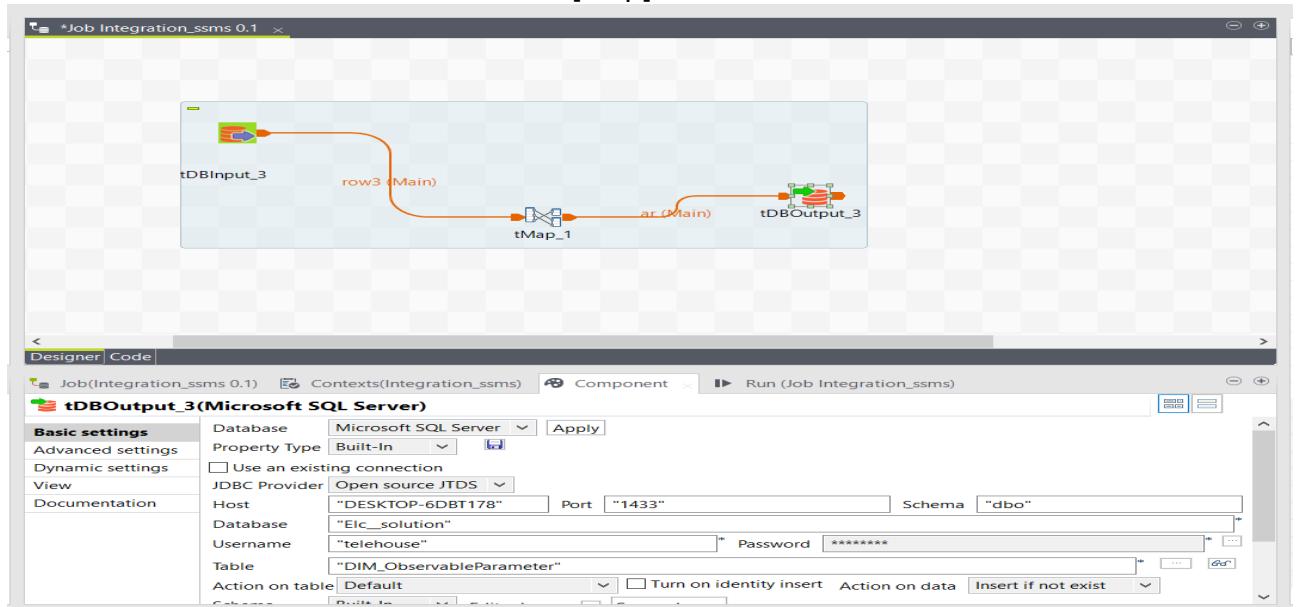


Figure 3.31: Etape 4

3.6.5.4 Creation de la table de fait Voltaire :

JOB :

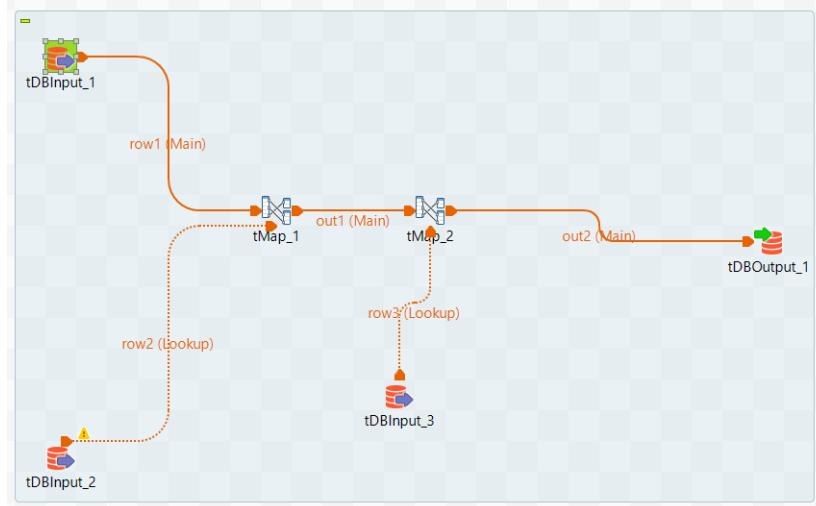


Figure 3.32: Etape 1

Ce job comporte deux composants TDBInput, chacun extrayant des données de sources de données distinctes. Ces données sont ensuite combinées et transformées à l'aide de deux composants TMap, permettant ainsi de créer une structure de données consolidée et enrichie. Ensuite, une troisième source de données est intégrée au

processus à l'aide d'un autre composant TDBInput, puis combinée avec les données précédemment traitées à l'aide d'une autre TMap. Enfin, les résultats finaux sont écrits dans une base de données cible à l'aide du composant TDBOutput. Ce processus permet d'agréger, transformer et charger efficacement les données, facilitant ainsi les analyses ultérieures.

TDBInput1 :

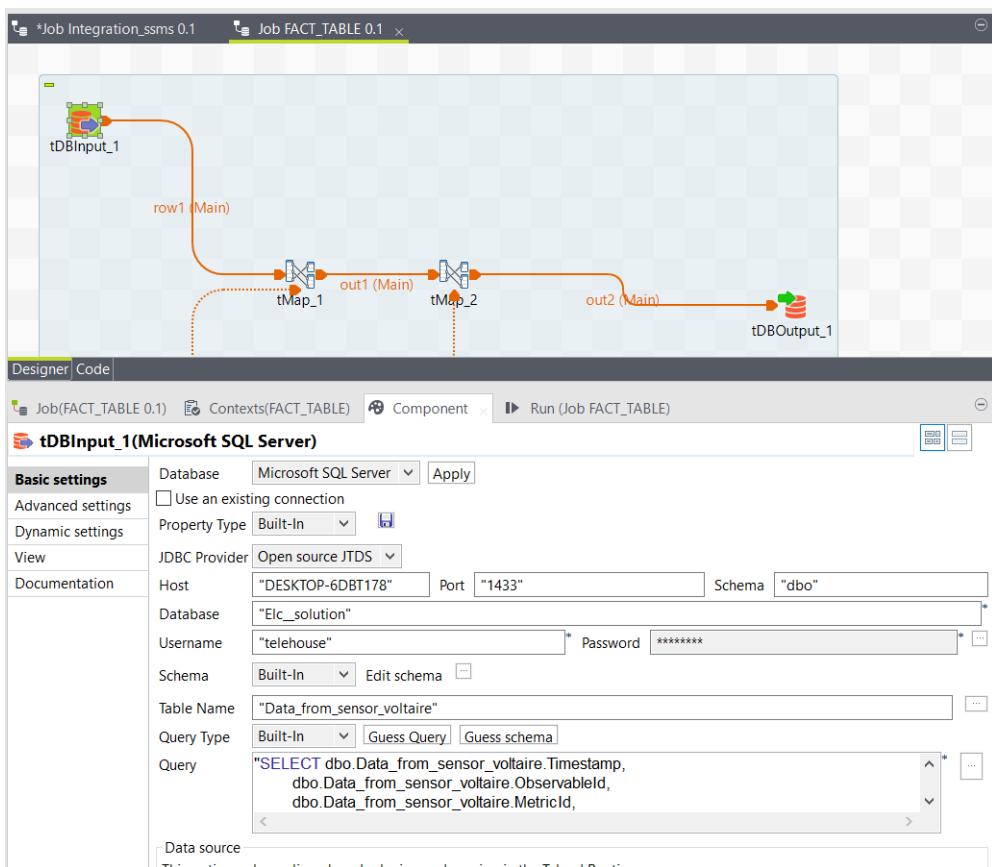


Figure 3.33: Etape 1

Dans cette étape TDBInput récupère des données de la table "Data from sensor Voltaire" dans la base de données "Elc Solution" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

TDBInput2 :

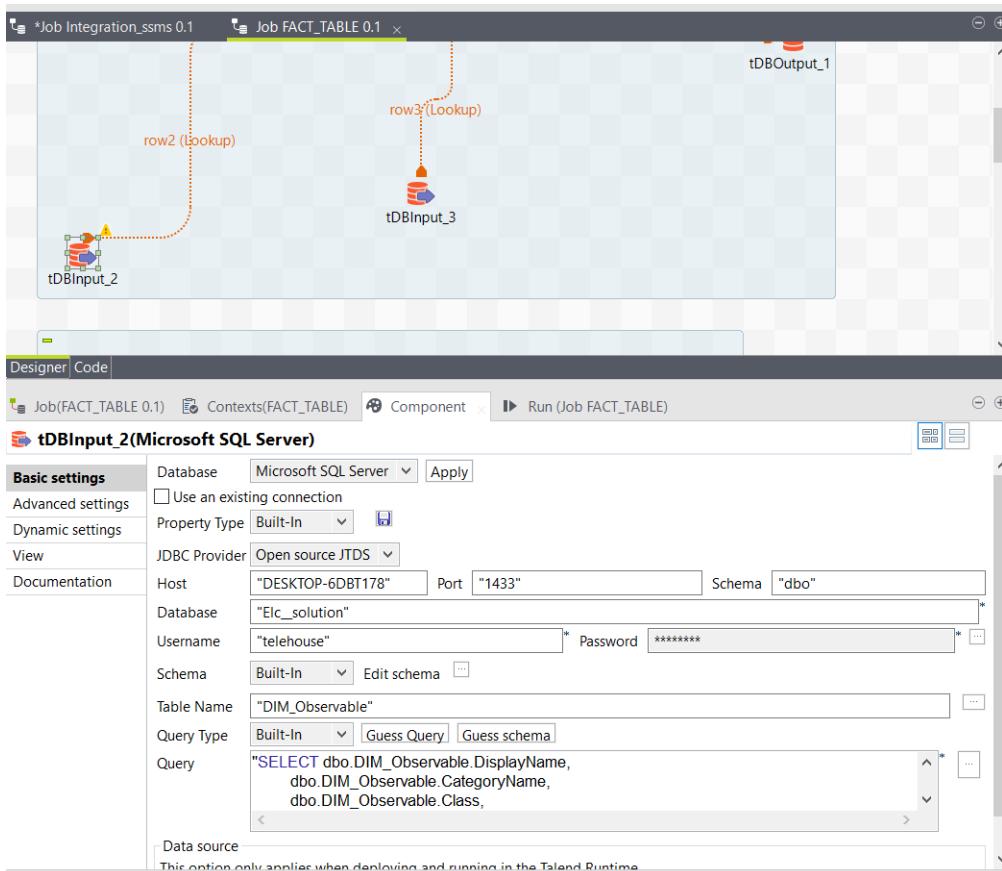


Figure 3.34: Etape 3

Dans cette étape TDBInput récupère des données de la table "DIM Observable" dans la base de données "Elc Solution" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

TDBInput3 :

Dans cette étape TDBInput récupère des données de la table "DIM ObservablesetMapping" dans la base de données "Elc Solution" située sur le serveur "desktop-6DBT178". Elle utilise une requête SQL spécifique pour sélectionner les colonnes pertinentes de cette table.

TMAP1 :

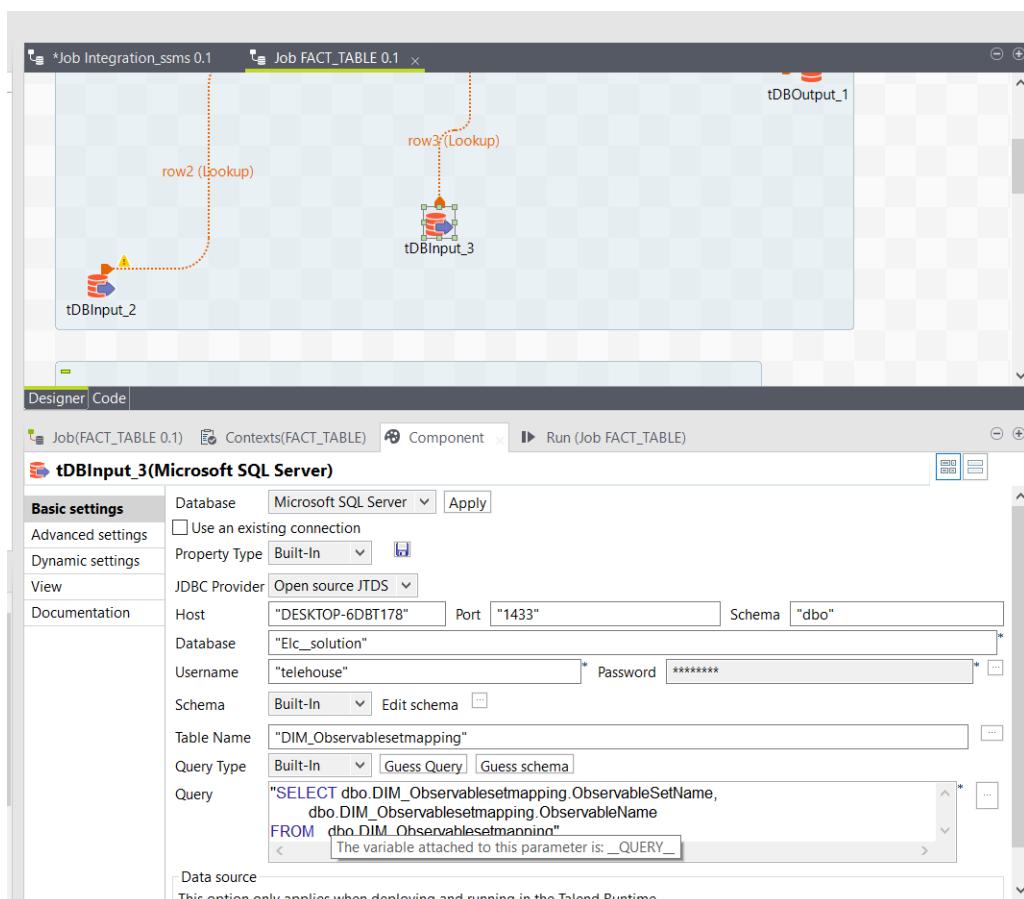


Figure 3.35: Etape 4

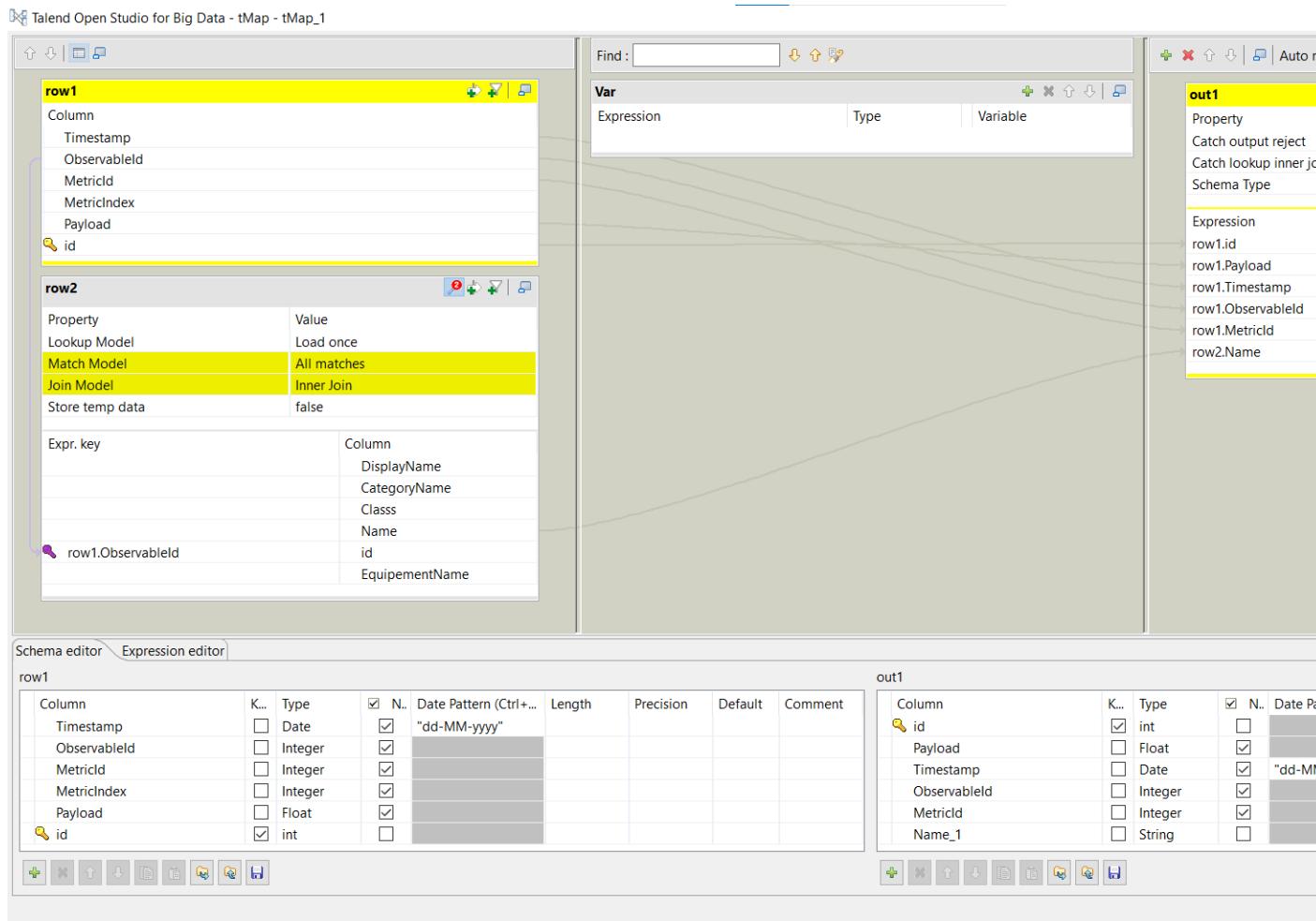


Figure 3.36: Etape 5

Premièrement, dans le premier TMap, nous avons relié les lignes de la table "Data from sensor Voltaire" avec celles de la table "DIM Observable". Cette étape nous a permis de créer une jointure initiale entre ces deux tables, combinant les données pertinentes pour notre analyse.

TMAP2 :

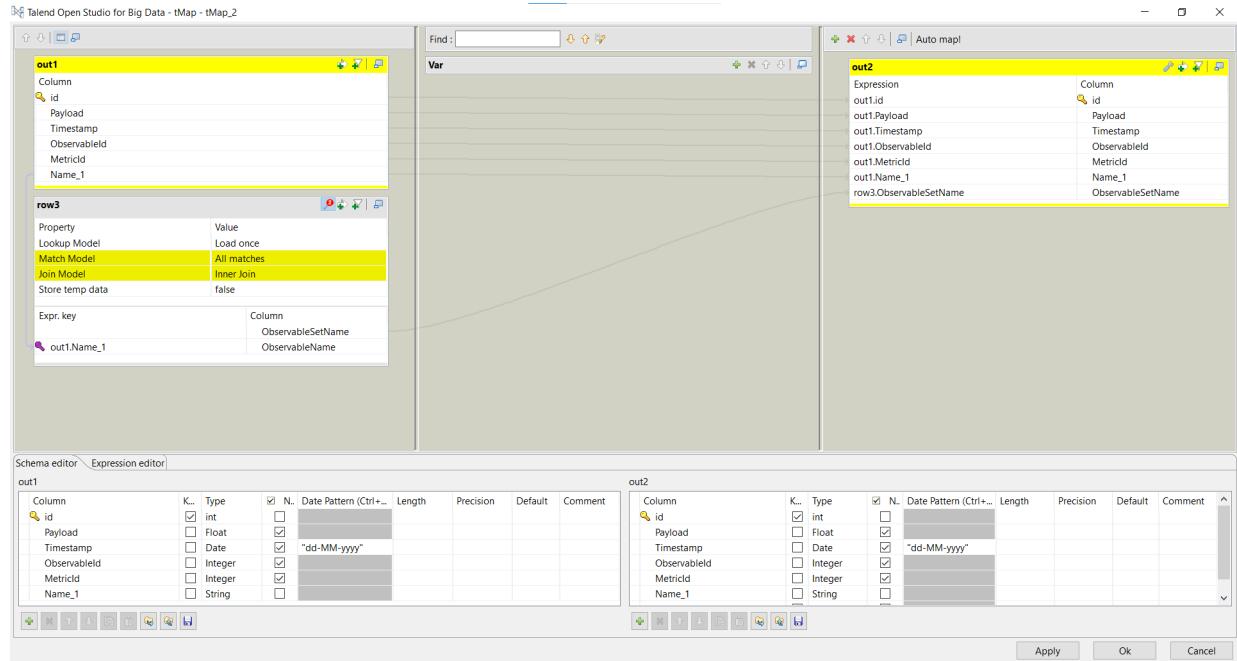


Figure 3.37: Etape 6

Ensuite, dans le deuxième TMap, nous avons étendu cette jointure en intégrant les données de la première jointure avec celles de la table "DIM ObservableSetMapping". Cette étape a enrichi notre jeu de données en incluant des informations supplémentaires provenant de la table "DIM ObservableSetMapping", renforçant ainsi la qualité et la richesse de nos résultats finaux.

→ Ces deux étapes de jointure, réalisées à l'aide des composants TMap de Talend, ont été essentielles pour consolider nos données et préparer le terrain pour les analyses ultérieures

TDBOutput :

Dans cette étape, TDBOutput écrit les données récupérées des tables dans la table "Fait Voltaire".

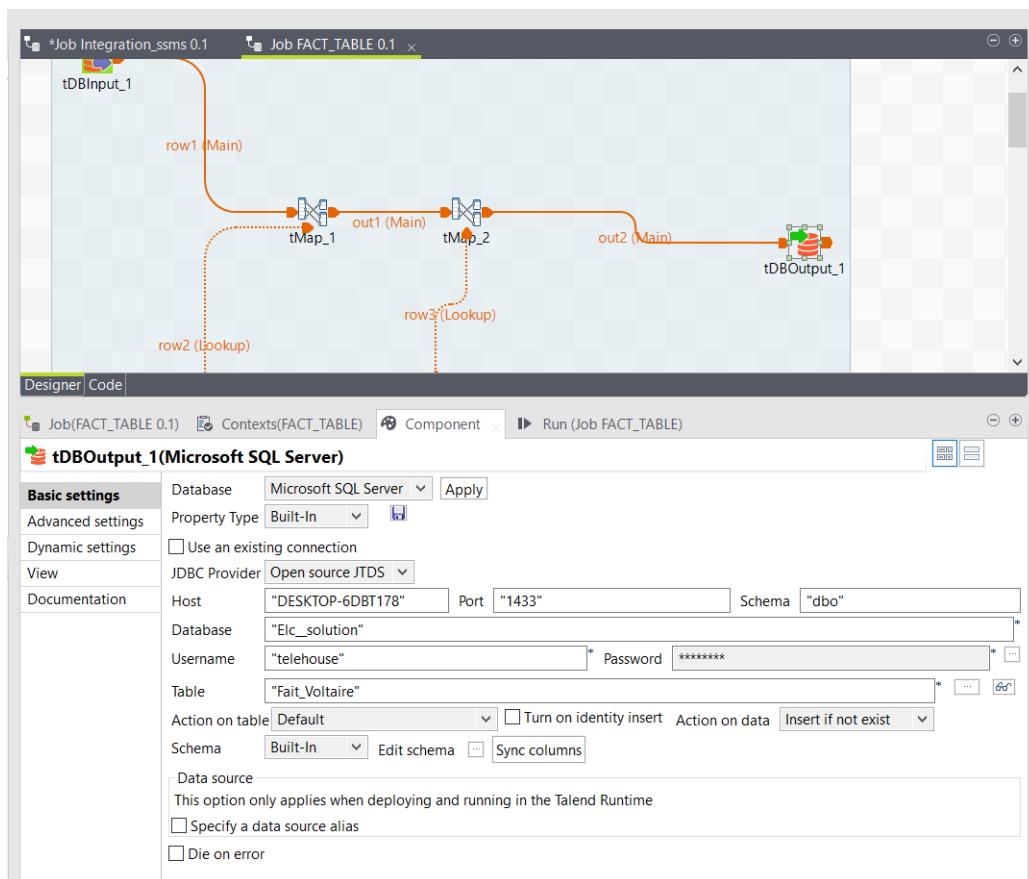


Figure 3.38: Etape 7

Resultat Final :

```

SELECT TOP (1000) [id]
    ,[Payload]
    ,[Timestamp]
    ,[ObservableId]
    ,[MetricId]
    ,[Name]
    ,[ObservableSetName]
FROM [Elc_solution].[dbo].[Fait_Voltaire]

```

id	Payload	Timestamp	ObservableId	MetricId	Name	ObservableSetName
1	256.160003662109	2024-03-19 19706	1	PowerLine-Umg804_Th2_F1_R1_A2-MT1-Alim-03-Sink	Th2_F1_R1_A2-OS	
2	242.510009779297	2024-03-19 10309	1	PowerLine-Umg804_Th2_F1_R2_A2-MT1-Alim-06-Sink	Th2_F1_R2_A2-OS	
3	257.510009765625	2024-03-19 10106	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-03-Sink	Th2_F1_R2_A1-OS	
4	256.160003662109	2024-03-19 19709	1	PowerLine-Umg804_Th2_F1_R1_A2-MT1-Alim-06-Sink	Th2_F1_R1_A2-OS	
5	241.789993286133	2024-03-19 50006	1	PowerLine-Umg804_Th2_F1_R1_A1-MT1-Alim-03-Sink	Th2_F1_R1_A1-OS	
6	256.160003662109	2024-03-19 19712	1	PowerLine-Umg804_Th2_F1_R1_A2-MT1-Alim-09-Sink	Th2_F1_R1_A2-OS	
7	257.510009765625	2024-03-19 10109	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-06-Sink	Th2_F1_R2_A1-OS	
8	246.690002441406	2024-03-19 31706	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-03-Sink	Th2_F2_D22_A	
9	257.510009765625	2024-03-19 10112	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-09-Sink	Th2_F1_R2_A1-OS	
10	257.510009765625	2024-03-19 10115	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-12-Sink	Th2_F1_R2_A1-OS	
11	246.690002441406	2024-03-19 31709	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-06-Sink	Th2_F2_D22_A	
12	257.510009765625	2024-03-19 10118	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-15-Sink	Th2_F1_R2_A1-OS	
13	246.690002441406	2024-03-19 31712	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-09-Sink	Th2_F2_D22_A	
14	257.510009765625	2024-03-19 10121	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-18-Sink	Th2_F1_R2_A1-OS	
15	246.690002441406	2024-03-19 31715	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-12-Sink	Th2_F2_D22_A	
16	257.510009765625	2024-03-19 10124	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-21-Sink	Th2_F1_R2_A1-OS	
17	257.510009765625	2024-03-19 10127	1	PowerLine-Umg804_Th2_F1_R2_A1-MT1-Alim-24-Sink	Th2_F1_R2_A1_Source_OS	
18	246.690002441406	2024-03-19 31718	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-15-Sink	Th2_F2_D22_A	
19	246.690002441406	2024-03-19 31721	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-18-Sink	Th2_F2_D22_A	
20	246.690002441406	2024-03-19 31724	1	PowerLine-Umg804_Th2_F2_D19-MT1-Alim-21-Sink	Th2_F2_D22_A	
21	246.690002441406	2024-03-19 31730	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-03-Sink	Th2_F2_D23_A	
22	246.690002441406	2024-03-19 31733	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-06-Sink	Th2_F2_D23_A	
23	246.690002441406	2024-03-19 31736	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-09-Sink	Th2_F2_D23_A	
24	246.690002441406	2024-03-19 31739	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-12-Sink	Th2_F2_D23_A	
25	246.690002441406	2024-03-19 31742	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-15-Sink	Th2_F2_D23_A	
26	246.690002441406	2024-03-19 31745	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-18-Sink	Th2_F2_D23_A	
27	246.690002441406	2024-03-19 31748	1	PowerLine-Umg804_Th2_F2_D19-MT2-Alim-21-Sink	Th2_F2_D23_A	

Query executed successfully.

Figure 3.39: Resultat

REMARQUE :

Dans le cadre de la conception du processus ETL pour notre entrepôt de données, j'ai élaboré la première table de fait que j'ai nommée "Fait Voltaire". Les étapes détaillées pour sa création ont été soigneusement documentées dans la section correspondante.

Cependant, afin d'éviter de répéter les mêmes étapes pour les deux autres tables de fait, j'ai reproduit les résultats obtenus dans SQL Server Management Studio (SSMS) pour ces tables.

Ces résultats sont présentés ci-dessous :

Fait Magny :

The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery6.sql - DE...on (telehouse (73))' (selected), 'SQLQuery5.sql - DE...on (telehouse (71))', and 'SQLQuery4.sql - DE...on (telehouse (52))'. The main area displays a query result grid for a table named 'Elc_solution.dbo.Fait_Magny'. The columns are: id, Payload, Timestamp, ObservableId, MetricId, Name, and ObservableSetName. The data shows 1,000 rows of measurements from March 19, 2024, with various metric names and observable set names. A status bar at the bottom indicates 'Query executed successfully.' and other session details.

id	Payload	Timestamp	ObservableId	MetricId	Name	ObservableSetName
1	246.5	2024-03-19	21806	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-03-Sink	Th3_B_F1_R2_A1-0S
2	251.729995727539	2024-03-19	21606	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-03-Sink	Th3_B_F1_R1_B2-0S
3	247.080001831055	2024-03-19	30806	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-03-Si...	Th3_B_F1_R1_A2-0S
4	246.5	2024-03-19	21809	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-06-Sink	Th3_B_F1_R2_A1-0S
5	247.080001831055	2024-03-19	30809	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-06-Si...	Th3_B_F1_R1_A2-0S
6	240.339996337891	2024-03-19	21006	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-03-Sink	Th3_B_F1_R1_A1-0S
7	247.080001831055	2024-03-19	30812	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-09-Si...	Th3_B_F1_R1_A2-0S
8	246.5	2024-03-19	21812	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-09-Sink	Th3_B_F1_R2_A1-0S
9	254.130004882813	2024-03-19	30606	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-03-Si...	Th3_B_F1_R1_A1-0S
10	246.5	2024-03-19	21815	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-12-Sink	Th3_B_F1_R2_A1-0S
11	254.130004882813	2024-03-19	30609	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-06-Si...	Th3_B_F1_R1_A1-0S
12	249.910003662109	2024-03-19	31026	1	PowerLine-Umg804_Th3_B_F1_R1_B1_02-MT1-Alim-23-S...	Th3_B_F1_R1_B1-0S
13	254.130004882813	2024-03-19	30612	1	PowerLine-Umg804_Th3_B_F1_A1_02-MT1-Alim-09-Si...	Th3_B_F1_R1_A1-0S
14	246.5	2024-03-19	21818	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-15-Sink	Th3_B_F1_R2_A1-0S
15	254.130004882813	2024-03-19	30615	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-12-Si...	Th3_B_F1_R1_A1-0S
16	257.269989013672	2024-03-19	21209	1	PowerLine-Umg804_Th3_B_F1_R1_A2-MT1-Alim-06-Sink	Th3_B_F1_R1_A2-0S
17	246.5	2024-03-19	21821	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-18-Sink	Th3_B_F1_R2_A1-0S
18	257.269989013672	2024-03-19	21212	1	PowerLine-Umg804_Th3_B_F1_R1_A2-MT1-Alim-09-Sink	Th3_B_F1_R1_A2-0S
19	251.729995727539	2024-03-19	21609	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-06-Sink	Th3_B_F1_R1_B2-0S
20	246.5	2024-03-19	21824	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-21-Sink	Th3_B_F1_R2_A1-0S
21	251.729995727539	2024-03-19	21612	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-09-Sink	Th3_B_F1_R1_B2-0S
22	251.729995727539	2024-03-19	21615	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-12-Sink	Th3_B_F1_R1_B2-0S
23	240.339996337891	2024-03-19	21009	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-06-Sink	Th3_B_F1_R1_A1-0S
24	247.080001831055	2024-03-19	30815	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-12-Si...	Th3_B_F1_R1_A2-0S
25	257.920013427734	2024-03-19	21406	1	PowerLine-Umg804_Th3_B_F1_R1_B1-MT1-Alim-03-Sink	Th3_B_F1_R1_B1-0S
26	247.080001831055	2024-03-19	30818	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-05-Si...	Th3_B_F1_R1_A2-0S
27	240.339996337891	2024-03-19	21012	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-09-Sink	Th3_B_F1_R1_A1-0S
28	247.080001831055	2024-03-19	30821	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-18-Si...	Th3_B_F1_R1_A2-0S
29	240.339996337891	2024-03-19	21015	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-10-Si...	Th3_B_F1_R1_A1-0S

Figure 3.40: Fait Magny

Fait LeonFrot :

The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery5.sql - DE...on (telehouse (71))', 'SQLQuery4.sql - DE...on (telehouse (52))*', and 'SQLQuery3.sql - DE...on (telehouse (63))'. The main area displays a query result grid for the 'Fait_Leonfrot' table. The table has columns: id, Payload, Timestamp, ObservableId, MetricId, Name, and ObservableSetName. The data consists of 28 rows, each representing a different metric ID and name combination. The 'Name' column contains various PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-xx-Sink entries. The 'ObservableSetName' column contains Th3_B_F1_R1_A1-OS, Th3_B_F1_R1_A2-OS, and Th3_B_F1_R1_A3-OS. The 'Payload' column contains various numerical values like 246.5, 251.729995727539, etc. The 'Timestamp' column shows dates from 2024-03-19.

	id	Payload	Timestamp	ObservableId	MetricId	Name	ObservableSetName
1	2	246.5	2024-03-19	21806	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-03-Sink	Th3_B_F1_R2_A1-OS
2	3	251.729995727539	2024-03-19	21606	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-03-Sink	Th3_B_F1_R1_B2-OS
3	4	247.080001831055	2024-03-19	30806	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-03-Sink	Th3_B_F1_R1_A2-OS
4	5	246.5	2024-03-19	21809	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-06-Sink	Th3_B_F1_R2_A1-OS
5	6	247.080001831055	2024-03-19	30809	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-06-Sink	Th3_B_F1_R1_A2-OS
6	7	240.339996337891	2024-03-19	21006	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-03-Sink	Th3_B_F1_R1_A1-OS
7	8	247.080001831055	2024-03-19	30812	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-09-Sink	Th3_B_F1_R1_A2-OS
8	9	246.5	2024-03-19	21812	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-09-Sink	Th3_B_F1_R2_A1-OS
9	10	254.130004882813	2024-03-19	30606	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-03-Sink	Th3_B_F1_R1_A1-OS
10	11	246.5	2024-03-19	21815	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-12-Sink	Th3_B_F1_R2_A1-OS
11	12	254.130004882813	2024-03-19	30609	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-06-Sink	Th3_B_F1_R1_A1-OS
12	13	249.910003662108	2024-03-19	31026	1	PowerLine-Umg804_Th3_B_F1_R1_B1_02-MT1-Alim-23-Sink	Th3_B_F1_R1_B1-OS
13	14	254.130004882813	2024-03-19	30612	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-09-Sink	Th3_B_F1_R1_A1-OS
14	15	246.5	2024-03-19	21818	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-15-Sink	Th3_B_F1_R2_A1-OS
15	16	254.130004882813	2024-03-19	30615	1	PowerLine-Umg804_Th3_B_F1_R1_A1_02-MT1-Alim-12-Sink	Th3_B_F1_R1_A1-OS
16	17	257.269989013672	2024-03-19	21209	1	PowerLine-Umg804_Th3_B_F1_R1_A2-MT1-Alim-06-Sink	Th3_B_F1_R1_A2-OS
17	18	246.5	2024-03-19	21821	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-18-Sink	Th3_B_F1_R2_A1-OS
18	19	257.269989013672	2024-03-19	21212	1	PowerLine-Umg804_Th3_B_F1_R1_A2-MT1-Alim-09-Sink	Th3_B_F1_R1_A2-OS
19	20	251.729995727539	2024-03-19	21609	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-06-Sink	Th3_B_F1_R1_B2-OS
20	21	246.5	2024-03-19	21824	1	PowerLine-Umg804_Th3_B_F1_R2_A1-MT1-Alim-21-Sink	Th3_B_F1_R2_A1-OS
21	22	251.729995727539	2024-03-19	21612	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-09-Sink	Th3_B_F1_R1_B2-OS
22	23	251.729995727539	2024-03-19	21615	1	PowerLine-Umg804_Th3_B_F1_R1_B2-MT1-Alim-12-Sink	Th3_B_F1_R1_B2-OS
23	24	240.339996337891	2024-03-19	21008	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-06-Sink	Th3_B_F1_R1_A1-OS
24	25	247.080001831055	2024-03-19	30815	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-12-Sink	Th3_B_F1_R1_A2-OS
25	26	257.920013427734	2024-03-19	21406	1	PowerLine-Umg804_Th3_B_F1_R1_B1-MT1-Alim-03-Sink	Th3_B_F1_R1_B1-OS
26	27	247.080001831055	2024-03-19	30818	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-15-Sink	Th3_B_F1_R1_A2-OS
27	28	240.339996337891	2024-03-19	21012	1	PowerLine-Umg804_Th3_B_F1_R1_A1-MT1-Alim-09-Sink	Th3_B_F1_R1_A1-OS
28	29	247.080001831055	2024-03-19	30821	1	PowerLine-Umg804_Th3_B_F1_R1_A2_02-MT1-Alim-18-Sink	Th3_B_F1_R1_A2-OS

Query executed successfully. DESKTOP-6DBT178 (15.0 RTM) telehouse (71) Elc_solution 00:00:00 1,000 rows

Figure 3.41: Fait LeonFrot

3.7 Conclusion :

Dans cette section, nous avons minutieusement documenté le processus de création du data mart Elco MES, intégré dans notre solution de Business Intelligence (BI). Chaque étape de conception a été abordée avec rigueur, veillant à ce que le datamart Elco MES soit parfaitement aligné sur les besoins spécifiques de notre entreprise et qu'il fournit des informations pertinentes pour la prise de décision. Dans le prochain chapitre, nous procéderons à la finalisation des rapports générés et à la mise en œuvre de la solution sur le web.

CHAPITRE 4

SPRINT 3:

4.1 Introduction

CHAPITRE 5

ENVIRONNEMENT DE TRAVAIL

5.1 Introduction

Dans ce chapitre, nous présenterons en détail l'environnement matériel, les technologies et langages utilisés, ainsi que l'environnement logiciel essentiel à notre projet d'automatisation des tests. Plus spécifiquement, nous mettrons l'accent sur le plugin Xray Test Management intégré à Jira.

5.2 Environnement Matériel

- **Poste de travail NeoXam (Sur site)**
 - Processeur: Intel I5 8ème Génération
 - RAM : 16 Go (15.8 Go Utilisable)
 - Disque Dur : 222 Go
 - OS : Windows 10 Pro

CONCLUSION GÉNÉRALE

En conclusion, ce projet mené au sein de l'entreprise NeoXam Tunisie a réussi à mettre en place l'automatisation des tests pour le logiciel GP4, en mettant l'accent sur les fonctionnalités de la solution Matching. Grâce à l'adoption d'une approche agile basée sur la méthodologie Scrum, nous avons pu planifier, concevoir et exécuter de manière itérative les tests automatisés, en nous adaptant aux changements et en intégrant les retours d'expérience.

Notre contribution essentielle réside dans la création et l'exécution de scénarios de test spécifiquement conçus pour cibler les actions clés et les boutons de la fonction Matching sous GP4, en veillant à la conformité des résultats obtenus par rapport aux attentes définies. Les scripts automatisés que nous avons développés faciliteront et accéléreront les tests futurs, en réduisant les tâches manuelles et en augmentant la couverture des tests. Cette automatisation des tests jouera un rôle majeur dans l'amélioration globale de la qualité du logiciel GP4 et dans l'efficacité de son processus de développement.

Malgré certains défis rencontrés, tels que la configuration du framework et la couverture de tous les cas de test, nous avons surmonté ces obstacles avec succès grâce à notre détermination, notre collaboration étroite avec l'équipe de développement et notre expertise technique.

Ce projet a été une opportunité précieuse pour appliquer nos connaissances et compétences dans le domaine de l'automatisation des tests. Il nous a permis d'acquérir

une expérience pratique dans la mise en place d'une solution d'automatisation des tests et de renforcer nos compétences en matière de collaboration, de gestion de projet et d'adaptabilité.

En conclusion, ce projet a abouti à la mise en place des tests automatiques pour les scénarios d'utilisation couvrant les allers-retours avec l'utilisateur. Nous sommes convaincus que notre contribution aura un impact significatif sur l'efficacité opérationnelle de l'entreprise NeoXam Tunisie et sur la satisfaction de ses clients. Nous sommes reconnaissants d'avoir eu l'opportunité de réaliser ce projet et confiants dans notre capacité à appliquer les connaissances et compétences acquises dans notre future carrière professionnelle.



BIBLIOGRAPHIE