République Tunisienne

Ministère de l'Enseignement Supérieur

et de la Recherche Scientifique

Université de Tunis El Manar

Institut Supérieur d'Informatique d'El Manar

RAPPORT DE PROJET DE FIN D'ÈTUDES

Présenté en vue de l'obtention du

Diplôme National d'ingénieur

Spécialité : Ingénierie du Développement du Logiciel(IDL)

Par

Ons CHAHED

Conception et développement d'une platforme analytique pour les méta-données de la platforme Snowflake

Encadrant professionnel : **Monsieur Nassim JALLOUD** Team leader Dev/Data

Encadrant académique : **Monsieur Sahbi BAHROUN** Maître Assistant

Réalisé au sein de Avaxia Group

Année Universitaire 2023 - 2024

République Tunisienne

Ministère de l'Enseignement Supérieur

et de la Recherche Scientifique

Université de Tunis El Manar

Institut Supérieur d'Informatique d'El Manar

RAPPORT DE PROJET DE FIN D'ANNÉE

Présenté en vue de la validation de

Diplôme National d'ingénieur

Spécialité : Ingénierie du Développement du Logiciel(IDL)

Par

Ons CHAHED

Conception et développement d'une platforme analytique pour les méta-données de la platforme Snowflake

Encadrant professionnel : **Monsieur Nassim JALLOUD** Team leader Dev/Data

Encadrant académique : **Monsieur Sahbi BAHROUN** Maître Assistant

Réalisé au sein de Avaxia Group

Année Universitaire 2023 - 2024

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue de la validation du stage d'été.

Encadrant professionnel, **Monsieur Nassim JALLOUD**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue de la validation du stage d'été.

Encadrant académique, **Monsieur Sahbi BAHROUN**

Signature

Dédicaces

À mes chers parents, la lumière de ma vie,

Puisque certaines dettes sont difficiles à rembourser, peu importe le temps que cela prend, je dédie ce travail signe de reconnaissance et de dévouement à mes chers parents qui m'ont donné la vie et la tendresse. Leur joie n'est que le succès de leurs enfants. Leur amour, leur soutient et leurs sacrifices ont abordé toutes les limites. J'espère avoir répondu même partiellement aux espoirs que vous avez fondés en moi.

À Mes chères grands parents,,

Ceci est ma profonde gratitude pour votre éternel amour, que ce rapport soit le meilleur cadeau que je puisse vous offrir.

À mon chére frère,

Tu as été à mes côtés pendant toutes les étapes de ce travail, je t'en suis très reconnaissante. Aucune dédicace ne peut exprimer la profondeur des sentiments fraternels, d'amour et d'attachement que j'éprouve à ton égard. Je vous dédie ce travail en témoignage de ma profonde affection.

À mes meilleurs amis,

Parfois, j'ai oublié de remercier les personnes qui font ma vie si merveilleuse à bien des égards.Aujourd'hui c'est le jour où j'aimerais leur dire que je vous remercie pour être là, à mes côtés durant cette période critique. Vous étiez ma source de motivation et je suis et je serais toujours fier de notre amitié ! Je suis impatiente de partager encore beaucoup d'autres moments fantastiques avec vous.

Ons CHAHED

Remerciement

Après avoir rendu grâce à Dieu tout puissant et miséricordieux, je tenu à remercier vivement tous ceux qui, de près ou de loin ont participé à l'achèvement du projet ou à la rédaction de ce document.

Je tiens tout particulièrement à remercier **Monsieur Nassim JaLLOUD**, mon encadrant professionnelle pour son accueil de joindre son personnel durant ce stage et de m'offrir l'opportunité de faire ce travail. Pour sa disponibilité, sa compréhensibilité, sa rigueur scientifique et son sens d'écoute et d'échange.

Je remercie **Monsieur Nassim Jalloud**, mon encadrant académique pour le temps consacré, les astuces et les conseils précieuses prodiguées durant ces années d'étude de cycle ingénieur.

Enfin, je n'oublie pas tous **mes enseignants de l'ISI** qui ont contribué à ma formation, qu'ils trouvent ici toute ma gratitude.

Ons CHAHED

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

— **CQRS** = **C**ommand **Q**uery **R**esponsibility **S**egregation

— **DAG** = **D**irected **A**cyclic **G**raph

— **DAO** = **D**ata **A**ccess **O**bject

— **ETL** = **E**xtract **T**ransform **L**oad

— **MQTT** = **M**essage **Q**ueuing **T**elemetry **T**ransport

— **REST** = **RE**presentational **S**tate **T**ransfer

— **SQL** = **S**tructured **Q**uery **L**anguage

— **SRP** = **S**ingle **R**esponsibility **P**rinciple

Introduction générale

Dans un paysage technologique en constante évolution, la gestion efficace des données est devenue un impératif pour la compétitivité des entreprises. Avec l'avènement des plateformes de data warehousing cloud telles que Snowflake, les entreprises bénéficient désormais d'outils puissants pour gérer, stocker et analyser leurs données à grande échelle, ouvrant ainsi de nouvelles perspectives de croissance et d'innovation.

Cependant, exploiter pleinement les capacités de Snowflake requiert une surveillance constante et une optimisation proactive des opérations effectuées sur cette plateforme. C'est dans ce contexte exigeant que s'inscrit le projet de surveillance des opérations Snowflake.

L'objectif principal de cette initiative est de fournir à Avaxia Group, ainsi qu'à ses clients utilisant Snowflake, une solution exhaustive de surveillance et d'analyse des opérations sur la plateforme. Concrètement, cette solution permettra non seulement de surveiller en temps réel les performances des entrepôts de données, mais également d'analyser de manière approfondie les métriques clés telles que la charge de travail, les temps de réponse et la disponibilité des ressources. En identifiant rapidement les goulets d'étranglement, les tendances émergentes et les variations de charge, elle offrira la possibilité d'anticiper les défis potentiels et de prendre des mesures correctives proactives pour optimiser l'utilisation de Snowflake.

Grâce à cette approche proactive de surveillance et d'optimisation, Avaxia Group vise à renforcer la compétitivité de ses clients en assurant une exploitation optimale de Snowflake et en garantissant une valeur ajoutée maximale à leurs activités.
En outre, cette solution inclura des fonctionnalités avancées telles que des alertes en temps réel, des tableaux de bord personnalisables et des analyses prédictives pour aider les équipes opérationnelles à prendre des décisions éclairées et à maximiser leur efficacité.

Chapitre 1

Cadre du projet

Plan

Introduction

Dans ce premier chapitre, nous plongeons dans le contexte global de notre projet au sein d'Avaxia Group. Cette section établit les bases en présentant le cadre général du projet, en soulignant les objectifs que nous nous efforçons d'atteindre.

1.1 Context général du projet

Ce projet s'inscrit dans le cadre du stage de fin d'études d'ingénieur au sein de la société Avaxia Group, se déroulant de Février 2024 à Mai 2024, dans l'objectif de l'obtension de diplôme national d'ingénieur de l'Institut Supérieur d'Informatique (ISI). "Notre mission centrale consiste à contribuer à la réalisation du projet intitulé "", en apportant notre expertise et notre contribution essentielle

1.2 Organisme d'accueil

1.2.1 Présentation générale de Avaxia Group

Avaxia Group, une entreprise de conseil en technologie créée en 1998 et ayant son siège à Dubaï, étend actuellement sa présence mondiale avec des bureaux au Japon et en Tunisie, et de futurs emplacements prévus au Canada, en France et à Dakar. Avaxia Group se spécialise dans les solutions middleware, la gestion des infrastructures système, ainsi que dans le conseil fonctionnel, couvrant des domaines tels que l'architecture, l'intégration et les opérations [1].

1.2.2 Les services de Avaxia Group [2]

Avaxia Group opère sur le vaste marché international, avec une présence dans de nombreux pays. L'entreprise propose une gamme diversifiée de services, notamment les suivants :

- **Expertise Technique en SAP :**
  - Gestion de l'infrastructure.
  - Optimisation et amélioration des performances.
  - Conception et architecture du paysage SAP.
  - Installation des systèmes, configuration et mises à niveau.
  - Assistance technique 24/7.
- **Conseil Fonctionnel :**

- Gestion fonctionnelle du déploiement et planification des tests.
- Conception de solutions métier pour le déploiement de nouveaux modules SAP.

- **Solutions Logicielles Personnalisées :**
  - Conception et développement de solutions logicielles utilisant des technologies telles que Java J2E, DELL BOOMI Flow, Salesforce et SAP Fiori.
- **Intégration des Systèmes :**
  - Intégration de données et gestion des flux de données.
  - Intégration d'applications métier.
  - Développement et gestion d'API : DELL BOOMI, Atmosphère.
- **Ingénierie des Données :**
  - Modélisation, découverte, traitement, visualisation et exploration des données, ainsi que gestion du Big Data.
- **Nouvelles Technologies :**

  Création et déploiement de solutions logicielles d'entreprise en utilisant les dernières technologies, notamment :
  - Apprentissage automatique (Machine Learning).
  - Réalité virtuelle (VR).
  - Réalité augmentée (AR).
  - Internet des objets (IoT).

Cette gamme diversifiée de services reflète l'engagement d'Avaxia Group à fournir des solutions techniques avancées et à rester à la pointe de l'innovation pour répondre aux besoins variés de ses clients à l'échelle mondiale.

1.3 Étude du projet

Dans cette section nous explorons en détail l'etude de ce projet toutes en commmançant par les raisons fondamentales ayant motivé le développement de notre solution et les spécifications de cette derniére ainsi que son objectif.

1.3.1 Problématique

Avec l'avènement des technologies de cloud computing et des entrepôts de données modernes tels que Snowflake, les entreprises disposent désormais d'une flexibilité et d'une évolutivité inégalées pour la gestion et l'analyse de leurs données. Cependant, cette transition vers le cloud présente des défis spécifiques en termes de performance, de coûts, et de surveillance des opérations.
Dans ce contexte, Avaxia Group se trouve confrontée à des problématiques particulières liées à l'utilisation de Snowflake, son principal entrepôt de données telque :

- **Gestion des performances :**

  - Avaxia Group constate des délais dans l'exécution des requêtes et des temps de traitement prolongés lors de l'utilisation de Snowflake. Ces problèmes impactent directement la productivité des équipes et la satisfaction des utilisateurs finaux.

  - Les requêtes SQL complexes ou mal optimisées peuvent engendrer des goulets d'étranglement et des inefficacités dans l'utilisation des ressources de Snowflake, nécessitant une surveillance et une optimisation constantes.

- **Maîtrise des coûts :**

  - Les coûts associés à l'utilisation de Snowflake peuvent rapidement augmenter en raison d'une gestion inadéquate des ressources. Avaxia Group doit donc trouver des moyens de maîtriser ces coûts tout en garantissant des performances optimales pour ses opérations.

- **Surveillance et analyse des opération :**

  - Pour assurer le bon fonctionnement de ses opérations Snowflake, Avaxia Group a besoin d'une solution de surveillance avancée pour détecter les anomalies, identifier les goulets d'étranglement et optimiser les performances.

  De plus, une analyse approfondie des métriques opérationnelles telles que le temps d'exécution des requêtes, l'utilisation des ressources et les performances des entrepôts est essentielle pour optimiser l'efficacité et l'efficience des opérations.

1.3.2 Analyse et critique de l'existant

Dans cette section, nous passerons en revue les fonctionnalités, les avantages et les limitations de chaque solution, en mettant en lumière les lacunes qui ont conduit au développement de notre propre solution de monitoring des opérations Snowflake chez Avaxia Group.

1.3.2.1 Analyse de l'existant

L'analyse de l'existant est une étape cruciale dans le processus de développement de notre solution de monitoring des opérations sur Snowflake. C'est pour céla, nous allons entammer l'analyse les outils existants utilisés pour surveiller et analyser les opérations sur les divers plateformes de l'analyse des données et le data warehousing cloud

- **Snowflake account usage dashboard :** est un outil intégré qui fournit des informations sur l'utilisation de Snowflake, telles que le nombre de sessions utilisateur, la quantité de données stockées et le temps CPU utilisé. Il offre une vue rétrospective des opérations passées, permettant aux utilisateurs de comprendre l'utilisation historique de la plateforme.

  La figure **1.1** suivante illustre une partie de cette dashboard :

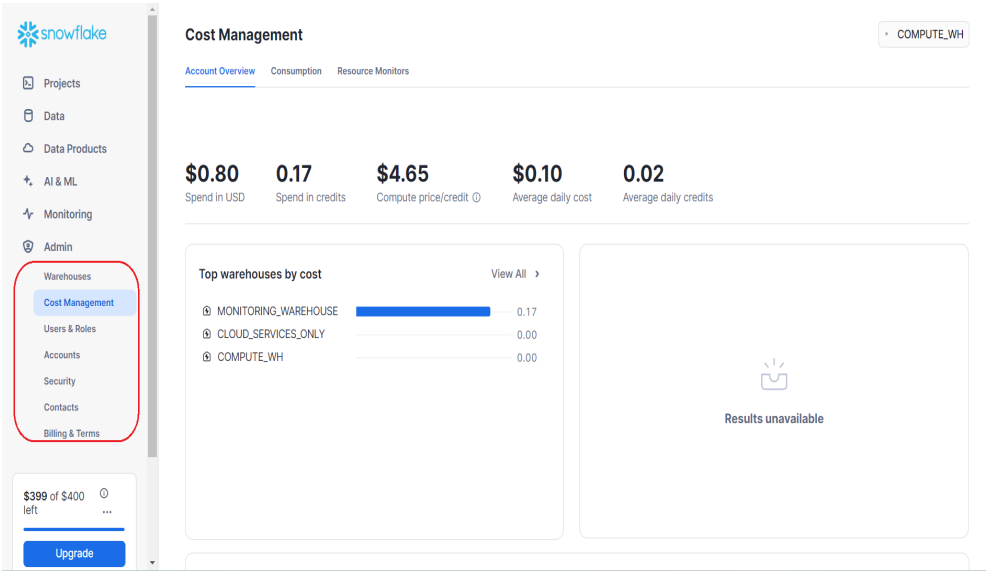


Figure 1.1 : Snowflake account usage dashboard

- **Snowflake Information Schema :** est une collection de vues système qui fournissent des métadonnées sur les objets et les opérations effectuées dans un compte Snowflake. Ces vues offrent une granularité élevée pour examiner les détails des requêtes SQL exécutées, les performances des entrepôts de données et d'autres aspects des opérations Snowflake.

  La figure **1.2** suivante illustre une partie de cette dashboard :

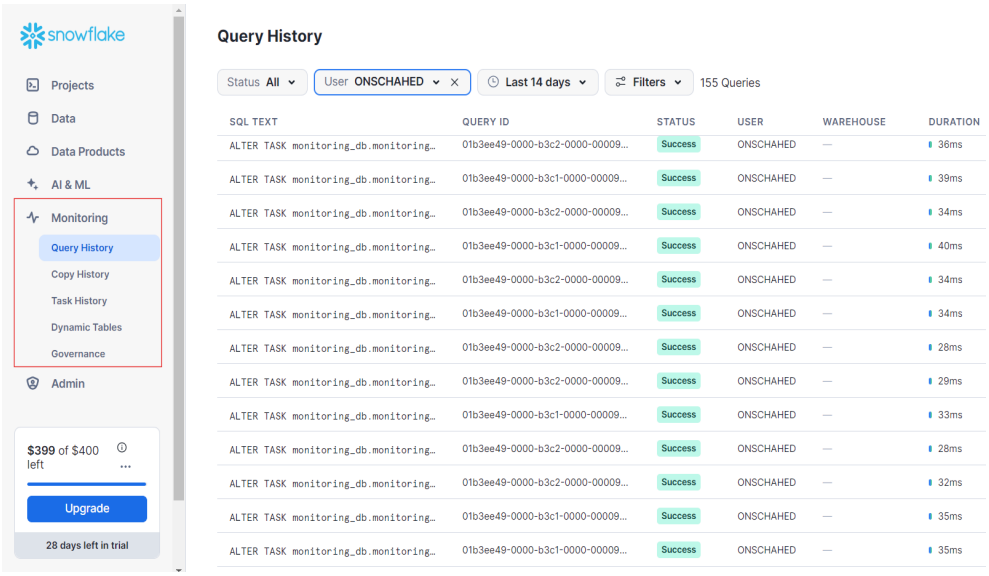


Figure 1.2 : Snowflake account usage dashboard

- **Google BigQuery :** est une autre option populaire pour le stockage et l'analyse des données dans le cloud. Il offre des fonctionnalités avancées telles que le traitement massivement parallèle et la capacité à exécuter des requêtes SQL complexes sur de grands ensembles de données.

  La figure **1.3** suivante illustre une partie de tableau de bord de BigQuery :

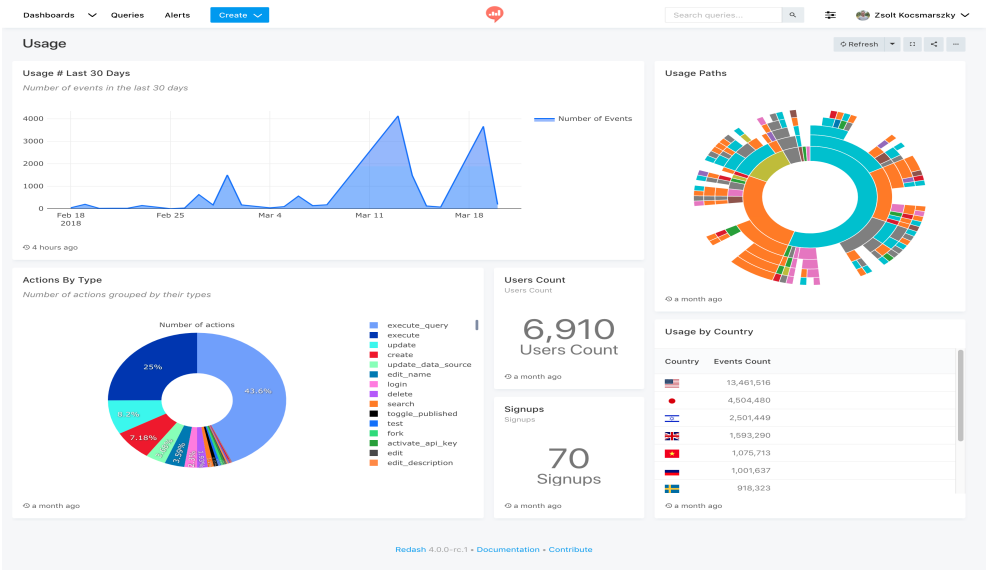


Figure 1.3 : Tableau de bord de "Google BigQuery"

- **Amazon Redshift :** est un entrepôt de données cloud basé sur PostgreSQL, conçu pour gérer de gros volumes de données et exécuter des analyses complexes. Il offre des performances élevées et une extensibilité, mais son modèle de tarification basé sur l'utilisation des ressources peut entraîner des coûts supplémentaires pour les entreprises

La figure **1.4** suivante illustre le tableau de bord de ResShift :

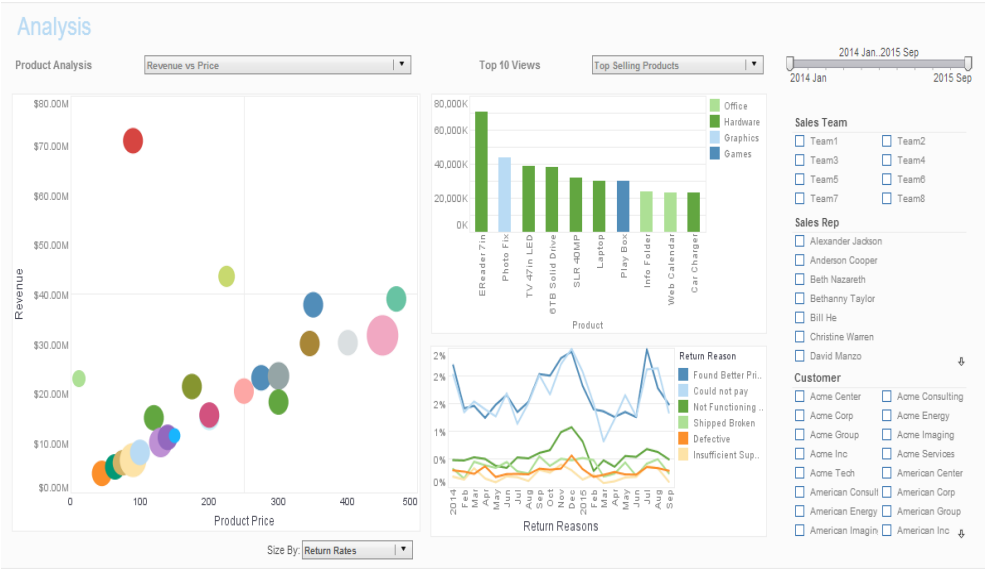


Figure 1.4 : Tableau de bord de "Amazon Redshift"

1.3.2.2 Critique de l'existant

Dans cette section, nous allons examiner de manière critique les outils actuellement utilisés dans le domaine de l'analyse de données, afin de fournir une base solide pour concevoir une solution qui surmonte les limitations et offre une valeur ajoutée significative à Avaxia Group et à ses clients.

- **Manque d'analyse prédictive :** l'un des principaux défauts des outils actuellement disponibles est leur incapacité à fournir une analyse prédictive des performances de Snowflake. Les tableaux de bord existants, tels que le Snowflake Account Usage Dashboard, offrent une vue rétrospective des opérations passées, mais ne fournissent pas d'informations sur les tendances futures ou les possibles goulots d'étranglement.
- **Complexité des outils :** les outils de surveillance existants, comme le Snowflake Information Schema, sont souvent complexes à utiliser et nécessitent des compétences techniques avancées pour interpréter les données fournies. Cette complexité peut rendre difficile la compréhension des métriques et la prise de décisions éclairées par les équipes opérationnelles. (par exemple dans un workflow si une certaine tache est échouée, les indicateurs disponibles sur snowflake ne peuvent pas identifier ou exactement le workflow est suspendu de façon à rendre les choses plus compliqué pour les utilisaturs de Snowflake de détecter les anomalies.)
- **Personnalisation limitée :** les options de personnalisation offertes par les outils existants sont souvent limitées. Par exemple, Google BigQuery propose des fonctionnalités avancées,

mais la personnalisation des tableaux de bord et des rapports est restreinte. Cela peut être un obstacle pour les entreprises ayant des besoins spécifiques en matière de surveillance et d'analyse des performances.

- **Coûts supplémentaires :** enfin, certains outils, comme Amazon Redshift, peuvent entraîner des coûts supplémentaires importants pour les entreprises. La tarification basée sur l'utilisation des ressources peut rapidement augmenter, surtout si les entreprises ne surveillent pas activement leur utilisation. Cela peut constituer une barrière financière pour les petites et moyennes entreprises souhaitant utiliser ces outils de surveillance.

1.3.3 Solution proposée

Notre solution pour résoudre les défis auxquels Avaxia Group est confronté dans l'utilisation de Snowflake repose sur la conception et l'implémentation d'un système de surveillance et d'optimisation des opérations exhaustif. Cette solution se décompose en plusieurs composants clés, chacun ciblant des aspects spécifiques des problématiques identifiées :

- Développement d'algorithmes d'optimisation des requêtes SQL afin de de réduire les temps d'exécution et de minimiser les goulets d'étranglement dans l'utilisation des ressources de Snowflake.
- Mise en place de stratégies de monitoring en temps réel cela permettra d'identifier rapidement les problèmes de performance et de prendre des mesures correctives efficaces.
- Développement d'outils d'analyse avancée des coûts Cette analyse approfondie aidera nos utilisateurs finaux à mieux comprendre ses dépenses et à trouver des opportunités d'optimisation pour réduire les coûts tout en maintenant des performances élevées.
- la mise en place de politiques de gestion des ressources pour optimiser l'utilisation des ressources de Snowflake. Des stratégies telles que le dimensionnement automatique des entrepôts de données et l'optimisation de l'utilisation des crédits de calcul seront explorées.
- Développement des tableaux de bord interactifs et riches en informations pour permettre à Avaxia Group et à ses clients de surveiller en temps réel les performances de leurs opérations Snowflake. Ces tableaux de bord fourniront des indicateurs clés de performance, des graphiques et des visualisations pour faciliter la détection des anomalies et la prise de décisions éclairées.
- Utilisation de techniques d'analyse avancée des données opérationnelles pour permettre aux utilisateurs d'identifier les opportunités d'amélioration et d'optimisation de leurs opérations,

renforçant ainsi leur compétitivité et leur efficacité.

1.4 Choix méthodologique

Le choix de la méthode Kanban pour la gestion de notre projet est le fruit d'une réflexion stratégique minutieuse, loin d'être aléatoire. Kanban, qui tire son origine du terme japonais signifiant "tableau", se démarque par sa flexibilité, sa transparence et sa focalisation sur l'amélioration constante.
Dans le cadre de notre projet, Kanban s'impose naturellement comme le choix optimal. Cette méthode permet une gestion visuelle et en temps réel des tâches et des flux de travail, une caractéristique cruciale pour un projet centré sur l'analyse de données et la visualisation. Chaque étape de notre processus, de la collecte des données à la présentation des résultats, trouve une représentation claire et concise dans le tableau Kanban.
De surcroît, Kanban promeut une approche itérative et progressive, parfaitement en harmonie avec notre objectif d'amélioration continue. Elle autorise une gestion fluide et adaptable des tâches, offrant la souplesse nécessaire pour ajuster notre plan en fonction des découvertes et des besoins changeants du projet.
En optant pour Kanban, nous mettons l'accent sur la transparence et la communication au sein de l'équipe. Chacun dispose d'une vision limpide de l'état d'avancement du projet, favorisant ainsi la collaboration et l'engagement de tous les membres.
En résumé, la méthode Kanban s'impose comme un choix stratégique judicieux pour notre projet, offrant un cadre solide pour une gestion efficace, transparente et itérative, tout en servant l'objectif fondamental d'amélioration continue de la performance de l'équipe Avaxia.

Conclusion

À la clôture de ce chapitre initial, nous avons jeté les fondements pour la compréhension complète de la sphère du projet. Cette phase est cruciale pour cadrer les enjeux et les perspectives qui guident notre travail. Dans le prochain chapitre, nous explorerons en détail l'analyse et la spécification des besoins, une étape essentielle pour la réalisation de notre projet.

Chapitre 2

Analyse et spécification des besoins

Plan

Introduction

Le deuxième chapitre se consacre à l'analyse et à la spécification des besoins. Nous examinons en profondeur les besoins fonctionnels et non fonctionnels, tout en définissant clairement le flux de travail et le backlog du produit. Ces éléments constituent la base essentielle de la phase suivante de planification et de développement.

2.1 Identification des besoins

Dans cette section on va focaliser sur les besoins fonctionnels et non fonctionnels de notre projet.

2.1.1 Besoins fonctionnels

Cette section décrit en détail les besoins fonctionnels du projet de monitoring des opérations Snowflake. Ces besoins ont été identifiés à partir des exigences de l'entreprise et des utilisateurs, et sont essentiels pour le développement d'une solution efficace et adaptée aux besoins de l'entreprise.

- **Collecte automatique des données de performance de Snowflake** : Le système doit être capable de collecter automatiquement les données de performance de Snowflake, y compris les temps de réponse des requêtes, l'utilisation des ressources (CPU, mémoire, stockage) et les statistiques sur les entrepôts de données.
- **Surveillance en temps réel des requêtes SQL** : Le système doit surveiller en temps réel les requêtes SQL exécutées sur Snowflake, enregistrant les temps de réponse, les erreurs éventuelles, et en identifiant les requêtes lentes ou mal optimisées.
- **Surveillance en temps réel des entrepôts de données** : Le système doit surveiller en temps réel les métriques des entrepôts de données, y compris l'utilisation de l'espace disque, la répartition de la charge de travail et la consommation de ressources.
- **Tableaux de bord interactifs pour les métriques clés** : Le système doit fournir des tableaux de bord interactifs permettant de visualiser les métriques clés de performance de Snowflake, y compris les temps de réponse des requêtes, l'utilisation des ressources et les statistiques sur les entrepôts de données.
- **Suivi des tâches et des workflows (DAG monitoring)** : Le système doit permettre le suivi des tâches et des workflows exécutés sur Snowflake, enregistrant les étapes effectuées, les temps d'exécution et les erreurs éventuelles.

- **Surveillance de l'utilisation des crédits** : Le système doit surveiller l'utilisation des crédits de calcul sur Snowflake, enregistrant les crédits utilisés par chaque requête ou chaque tâche, ainsi que les tendances d'utilisation au fil du temps.
- **Surveillance des utilisateurs et des accès** : Le système doit suivre l'activité des utilisateurs sur Snowflake, en enregistrant les requêtes exécutées, les tables accédées et les autorisations utilisées.
- **Alertes et notifications en cas d'anomalies** : Le système doit générer des alertes et des notifications en temps réel en cas d'anomalies ou de situations critiques, telles qu'une augmentation soudaine du temps de réponse des requêtes ou une utilisation anormale des ressources.
- **Personnalisation des tableaux de bord** : Le système doit permettre aux utilisateurs de personnaliser les tableaux de bord en fonction de leurs besoins spécifiques, en sélectionnant les métriques à afficher et en configurant les seuils d'alerte.

2.1.2 Besoins non fonctionnels

Cette section détaille les besoins non fonctionnels du projet de monitoring des opérations Snowflake. Ces besoins concernent principalement les aspects de performance, de sécurité, d'évolutivité et d'expérience utilisateur de la solution. Ils sont essentiels pour garantir que le système répond aux attentes de l'entreprise en termes de qualité et de fiabilité.

- **Besoins en Performance**

  1. **Temps de réponse** : Le système doit fournir des temps de réponse rapides pour l'affichage des tableaux de bord et la génération des rapports, afin de garantir une expérience utilisateur fluide.
  2. **Évolutivité** : Le système doit être capable de gérer une grande quantité de données et de trafic sans compromettre ses performances, en s'adaptant de manière transparente à l'augmentation de la charge.

- **Besoins en Sécurité**

  1. **Confidentialité des données** : Le système doit garantir la confidentialité des données collectées, en assurant leur cryptage lors du stockage et de la transmission.

2. **Authentification et autorisation** : Le système doit mettre en place des mécanismes d'authentification robustes pour vérifier l'identité des utilisateurs et des administrateurs, ainsi que des contrôles d'autorisation pour limiter l'accès aux données sensibles.

- **Besoins en Disponibilité**

1. **Disponibilité** : Le système doit être disponible en permanence, avec un temps de fonctionnement maximal et une reprise rapide en cas de panne.
2. **Tolérance aux pannes** : Le système doit être capable de résister aux pannes matérielles ou logicielles, en assurant la redondance des composants critiques et la sauvegarde des données.

- **Besoins en Expérience Utilisateur**

1. **Facilité d'utilisation** : Le système doit être intuitif et convivial, avec une interface utilisateur bien conçue et une navigation facile.
2. **Personnalisation** : Le système doit permettre aux utilisateurs de personnaliser leur expérience, en configurant les préférences d'affichage et les paramètres de notification.

- **Archivage des données :** Les données historiques doivent être archivées de manière efficace pour garantir l'intégrité des données.

2.2 Flux de travail

Durant la réalisation de notre projet, Ce processus a été notre fil conducteur, nous permettant de passer par plusieurs étapes clés pour atteindre notre objectif. La figure **2.1** suivante illustre notre processus métier :

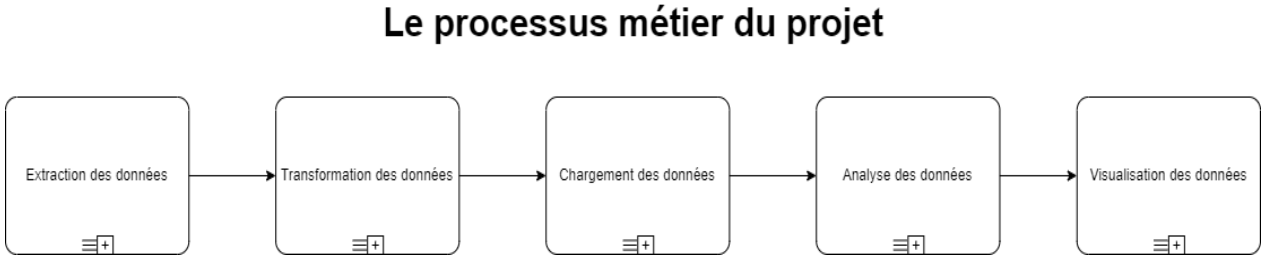


Figure 2.1 : Processus métier du projet

Dans cette partie, nous allons explorer en détail le flux de travail que nous avons suivi, en soulignant l'importance de chaque étape et en expliquant comment elles se sont complétées pour aboutir à une

solution complète. Ensuite, nous examinerons chaque étape de manière approfondie pour comprendre comment elles ont contribué à notre succès global.

Passons maintenant à l'examen détaillé de chacune de ces étapes.

- **Extraction des données :**

  Cette étape consistait à recueillir des informations provenant des différents vues du systéme de Snowflake.

  L'objectif principal de cette étape était de collecter les données nécessaires à notre analyse. Les données brutes sont la matière première de toute analyse, et leur extraction était le point de départ de notre projet.

- **Transformation des données :**

  L'objectif principal de cette étape était de nettoyer et de structurer les données. Les données brutes extraites pouvaient contenir des erreurs, des doublons, des incohérences et des valeurs manquantes. La transformation des données avait pour but de les rendre utilisables, fiables et cohérentes.

- **Chargement des données :**

  Une fois les données nettoyées et préparées, elles sont acheminées vers une base de données ou un entrepôt de données, comme PostgreSQL dans notre projet. Le chargement vise à stocker les données de manière centralisée pour une accessibilité et une analyse ultérieures.

- **Analyse des données :**

  Cette phase consiste à explorer les données pour en extraire des informations significatives. On utilise des méthodes statistiques, des techniques de modélisation, des calculs de corrélation, etc., pour comprendre les tendances et les relations entres les données.

- **Visualisation des données** :

  L'analyse des données est souvent présentée de manière visuelle sous forme de tableaux de bord interactifs, de graphiques, de diagrammes et d'autres représentations graphiques. Ces visualisations permettent aux utilisateurs de comprendre rapidement les résultats de l'analyse.

Ces phases constituent un flux de travail complet pour gérer les données, les transformer en informations exploitables et les présenter de manière efficace pour aider nos utilisateurs finaux à surveiller leurs données d'une maniére concrête.

2.3 Le backlog du produit

Le backlog de produit est destiné à recueillir tous les besoins du client pour l'équipe projet. Il contient la liste des fonctionnalités inclues dans un produit, ainsi que les éléments nécessitant l'intervention de l'équipe projet. Le backlog Scrum classe les éléments par priorité pour indiquer leur ordre de réalisation.[3].

Le backlog en KANBAN n'est pas si différent d'un backlog scrum en réalité. La différence réside surtout sur les règles de sa gestion. Bien que le scrum reste évasif sur le sujet, il rappelle quelques règles importantes. [4]

la table **2.1** suivante présente le backlog de produit de notre projet.

ID	**User Story**	**Priorité**	**Estimation (points)**
US1	En tant qu'utilisateur, je veux un tableau de bord global pour surveiller les performances générales de l'entrepôt de données, y compris les temps de réponse des requêtes, l'utilisation des ressources et les erreurs éventuelles.	Haute	8
US2	En tant qu'administrateur, je veux être en mesure de suivre en temps réel l'exécution des requêtes SQL sur la plateforme Snowflake, y compris les temps d'exécution, le nombre de lignes retournées et les plans d'exécution.	Haute	10
US3	En tant qu'administrateur, je veux surveiller l'utilisation des ressources (CPU, stockage, etc.) de l'entrepôt de données pour identifier les goulots d'étranglement et les problèmes de capacité.	Haute	9
US4	En tant qu'administrateur, je veux recevoir des alertes en temps réel en cas de dégradation des performances de l'entrepôt de données afin de pouvoir réagir rapidement et résoudre les problèmes.	Haute	8
US5	En tant qu'utilisateur, je veux générer des rapports de charge pour analyser la consommation de crédits, le coût par requête et l'utilisation des ressources au fil du temps.	Moyenne	6

US6	En tant qu'administrateur, je veux suivre l'activité des utilisateurs sur la plateforme Snowflake, y compris les requêtes exécutées, les tables accédées et les autorisations utilisées.	Moyenne	7
US7	En tant qu'administrateur, je veux être capable de gérer les sessions utilisateur actives, y compris la déconnexion des sessions inactives et la surveillance des sessions gourmandes en ressources.	Haute	8
US8	En tant qu'administrateur, je veux pouvoir définir des alertes personnalisées basées sur des seuils de performance spécifiques pour les requêtes, les ressources et les sessions.	Moyenne	7
US9	En tant que développeur, je veux accéder à des recommandations d'optimisation des requêtes SQL pour améliorer les performances et réduire les coûts.	Faible	9
US10	En tant qu'administrateur, je veux surveiller l'exécution des tâches planifiées (comme les pipelines de chargement) pour détecter les retards ou les échecs.	Moyenne	6
US11	En tant qu'utilisateur, je veux pouvoir analyser les tendances historiques des performances pour identifier les schémas et les anomalies.	Moyenne	8
US12	En tant qu'utilisateur, je veux s'authentifier d'une façon sécurisée pour accéeder a la platforme de monitoring	Moyenne	5
US13	En tant qu'administrateur, je veux suivre en temps réelle l'execution des taches programmés	haute	13

Tableau 2.1 : Backlog de Produit

Conclusion

Au terme de ce chapitre, nous avons obtenu une vision approfondie des besoins inhérents au projet. Cette analyse détaillée assure que notre solution est en phase avec les attentes et les exigences d'Avaxia Group. Dans le prochain chapitre, nous explorerons les aspects architecturaux et conceptuels de notre projet.

Chapitre 3

Architecture et conception

Plan

Introduction

Le troisième chapitre explore l'architecture du projet et sa étude conceptuelle. Nous allons justifier notre choix architectural et détaillrons la conception qui nous a ménné à la réalisation de notre solution.

3.1 Étude architecturale

L'étude architecturale joue un rôle central dans la conception et le déploiement d'une solution robuste et évolutive.
C'est dans ce contexte,

3.1.1 Architecture logique

Dans cette section, nous examinerons en profondeur l'architecture du système, couvrant à la fois ses dimensions logiques et physiques. Nous discuterons également des défis techniques et logiques rencontrés et des solutions déployées pour assurer une infrastructure solide et efficace.

3.1.1.1 Justification du choix de l'architecture micro-services

Le choix de l'architecture Le choix de l'architecture d'une application revêt une importance capitale dans le processus de conception d'un projet. Il détermine comment les diverses parties de l'application interagiront pour atteindre les objectifs fixés.
Pour la mise en place de ce projet, nous avons délibérément opté pour **une architecture micro-services**, une décision dictée par des critères spécifiques qui s'accordent parfaitement avec les exigences de notre projet ainsi que les differentes parties prenantes. Dans cette section, nous explorerons en détail les raisons pour lesquelles nous avons opté pour cette architecture, en soulignant sa pertinence pour notre projet.

- **Flexibilité et scalabilité :** nous avons choisi l'architecture microservices pour sa flexibilité et sa capacité à évoluer facilement. En adoptant une approche basée sur des services indépendants, nous pouvons développer, déployer et mettre à l'échelle chaque composant de manière autonome, ce qui facilite l'adaptation aux changements de charge et aux besoins évolutifs de notre système,
- **Isolation des fonctionnalités :** grâce aux microservices, le déploiement et la gestion des applications sont simplifiés. Chaque service peut être déployé de manière indépendante, ce qui

réduit les risques et accélère les cycles de déploiement. De plus, cela facilite la gestion des mises à jour et des correctifs, car seuls les services concernés sont impactés,

- **Déploiement et gestion simplifiés :** grâce aux microservices, le déploiement et la gestion des applications sont simplifiés. Chaque service peut être déployé de manière indépendante, ce qui réduit les risques et accélère les cycles de déploiement. De plus, cela facilite la gestion des mises à jour et des correctifs, car seuls les services concernés sont impactés,
- **Évolutivité et résilience :** les microservices favorisent une architecture résiliente et évolutive. En cas de panne d'un service, les autres services peuvent continuer à fonctionner normalement, ce qui réduit les interruptions. De plus, cette approche permet une meilleure répartition de la charge, assurant des performances optimales même lors de pics de trafic.

En conclusion, l'architecture microservices offre une solution flexible, évolutive et facilement maintenable pour notre projet de monitoring des opérations sur Snowflake. Elle nous permet de répondre efficacement aux besoins changeants de nos utilisateurs tout en garantissant des performances et une fiabilité optimales de notre système.

3.1.1.2 Architecture logique adoptée

Comme nous avons déja mentionner, l'architecture logique de notre solution adopte une approche modulaire reposant sur les microservices, afin de constitue le socle sur lequel repose la réalisation de l'objectif du projet. Elle met en lumière les différents composantes fonctionnelles et explique comment ces composantes interagissent pour fournir une expérience utilisateur fluide et des analyses précises [5].

Notre architecture est illustrée par la figure **3.1** suivante :

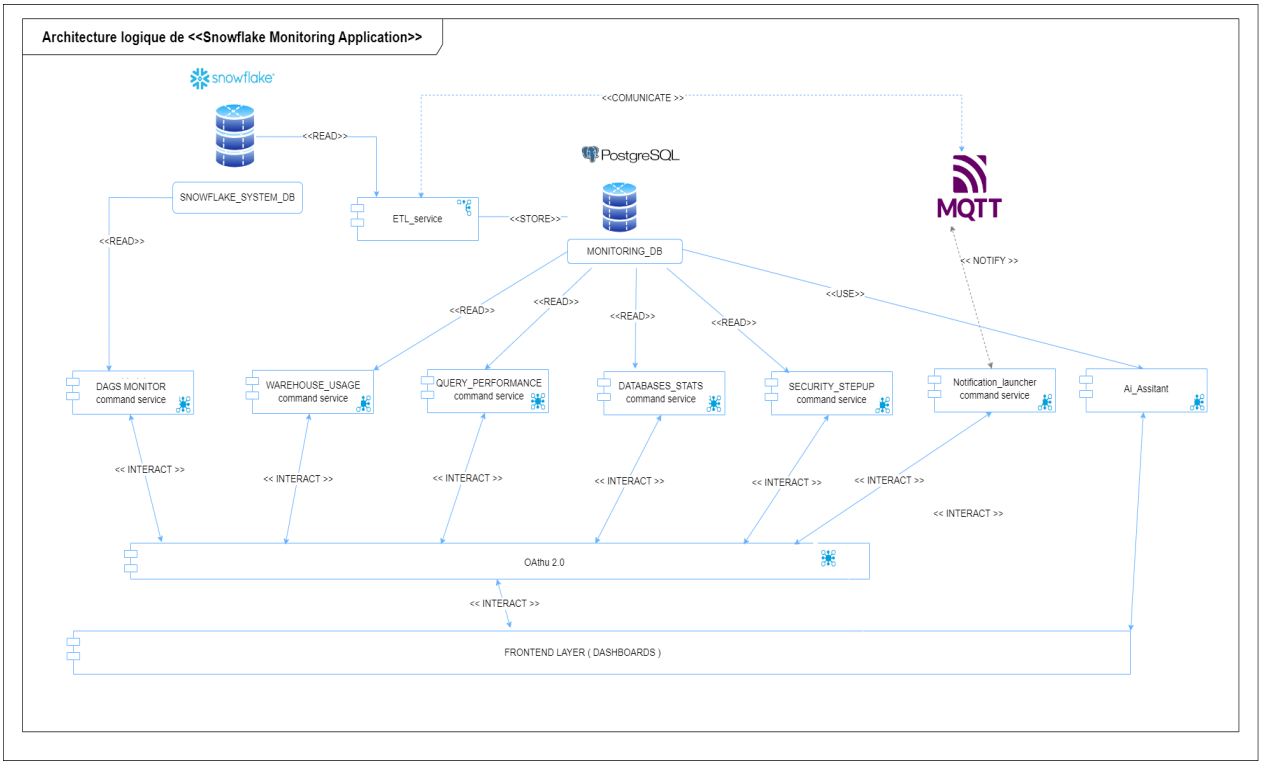


Figure 3.1 : Architecture logique de «Snowflake Monitoring Application»

Notre architecture est repartie comme suit :

1. **Couche des données :**
   - **Snowflake** : ce composant représente les vues sytémes de Snowflake telque «Account_ussage» et «information_schema» qui regroupent tous les données à monitorer.
   - **Monitoring_DB_pg** : c'est une base de données PostgreSQL stockant les données de monitoring collectées.
2. **Couche applicative « micro-services » :**
   - **Etl_Service** : service responsable de l'extraction, de la transformation et du chargement des données depuis Snowflake vers la base de données de monitoring,
   - **Dags_Monitoring_Service** : service responsable du suivi et de la surveillance des workflows «DAG» Snowflake,
   - **Warehouse_Usage_Service** : service chargé de la surveillance de l'utilisation des entrepôts de données Snowflake,
   - **Query_Performance_Service** : service d'analyse des performances des requêtes Snowflake,
   - **Databases_Stats_Service** : service de collecte des statistiques sur les bases de données Snowflake,

- **Access_Stats_Service** : service de collecte des statistiques sur les access au diffrenets comptes et entrepôts de données Snowflake,
- **Authentification_service** : service d'authentification et d'autorisation basé sur OAuth 2.0.

Cette couche assure la gestion de la logique métier de l'application en traitant et examinant les données collectées. Elle interagit avec l'interface utilisateur pour fournir des informations pertinentes, des fonctionnalités d'analyse, et des résultats.

3. **Couche de Communication :**
   - **MQTT** : système de messagerie basé sur le protocole MQTT, utilisé pour la communication asynchrone et découplée entre les différents services.
   - **MQTT Broker** : composant central du système MQTT, chargé de la distribution des événements publiés par les services.
4. **Couche de Présentation :**
   - **UI_Service** : ce servise fournit des tableaux de bord et des interfaces utilisateur pour visualiser les données de monitoring.

     Il offre une expérience utilisateur conviviale et interactive pour la visualisation des données de performance et les résultats d'analyse.

3.1.1.3 Patrons de conception

Les patrons de conception représentent des solutions éprouvées à des problèmes fréquents en conception logicielle. Ce sont comme des modèles ou des structures que l'on peut adapter pour résoudre des problèmes récurrents dans notre code[6].
Lors de l'élaboration de cette architecture, nous avons focaliser sur le fait de respecter les patrons de conception afin d'avoir une architecture solide et structurée.
Nous avons utilisé les patrons de conception suivants :

- **Le patron «Command Query Responsibility Segregation (CQRS)»** : il est aussi un model d'architecture qui sépare les deux parties de traitement (Écriture) et de réponse (Lecture) [7].

  L'architecture applique le modèle CQRS, avec une séparation des responsabilités entre le services de commande (ETL_service) et les services de requête (tout les autres services). Cette séparation optimise les performances des flux de lecture et d'écriture de manière indépendante.

- **Le patron «Façade»** : c'est un modèle de conception structurel qui fournit une interface simplifiée pour accéder à une bibliothèque, un framework ou à tout ensemble complexe de classes[8].

  Le service «ETL_service» fait office de façade, en encapsulant la complexité de l'interaction avec Snowflake, de façon que les autres services n'ont pas besoin de connaître les détails de l'interaction avec Snowflake.

- **Le patron «Observateur»** : c'est un modèle de conception comportemental qui permet d'établir un mécanisme de souscription pour envoyer des notifications à plusieurs objets concernant des événements liés aux objets qu'ils observent[9].

  Le service «MQTT_broker» fait office d'observateur, car implémente un modèle de publication/souscription, permettant aux services de s'abonner aux événements qui les intéressent. Cela découple les services et facilite l'ajout de nouveaux services.

- **Le pricipe «de responsabilité unique (SRP)»** : ce principe dénance que chaque qu'un composant (classe, service, module, etc.) ne doit avoir qu'une seule raison de changer. Cela se traduit par une séparation claire des responsabilités au sein de l'architecture[10].

  Chaque microservice (DAGS MONITOR, WAREHOUSE_USAGE, QUERY_PERFORMANCE, etc.) est responsable d'une fonctionnalité spécifique du monitoring.

  De plus, la séparation entre la base de données Snowflake (données surveiller) et la base de données Monitoring_DB (données de monitoring) respecte le SRP.

- **Le patron «Objet d'accès aux données (DAO)»** : ce moàdéle représente une façon d'organiser le code pour gérer la communication entre votre programme et une base de données. Il permet de conserver un code propre et de séparer la logique d'interaction avec les données du reste de l'application[11].

  L'accès à la base de données de monitoring (Monitoring_DB) est géré via un modèle DAO, séparant la logique d'accès aux données du reste de l'application. Cela améliore la maintenabilité et permet une indépendance entre la couche de données et la logique métier.

3.1.2 Architecture physique

L'architecture physique d'un projet constitue la matérialisation concrète de son architecture logique. Elle se consacre aux aspects matériels et aux ressources nécessaires pour assurer le bon fonctionnement de l'application[12].

Dans ce projet, nous avons adapter l'architecture n-tiers

L'architecture physique du système de monitoring Snowflake suit une approche en n-tiers afin de garantir une séparation claire des responsabilités, une meilleure évolutivité et une plus grande résilience de l'ensemble du système.

Cette conception en couches distinctes permet une gestion efficace des ressources, une répartition optimale des charges de travail et une sécurisation renforcée de l'infrastructure.

la figure **3.2** suivante illustre l'architecture adopté dans ce projet :

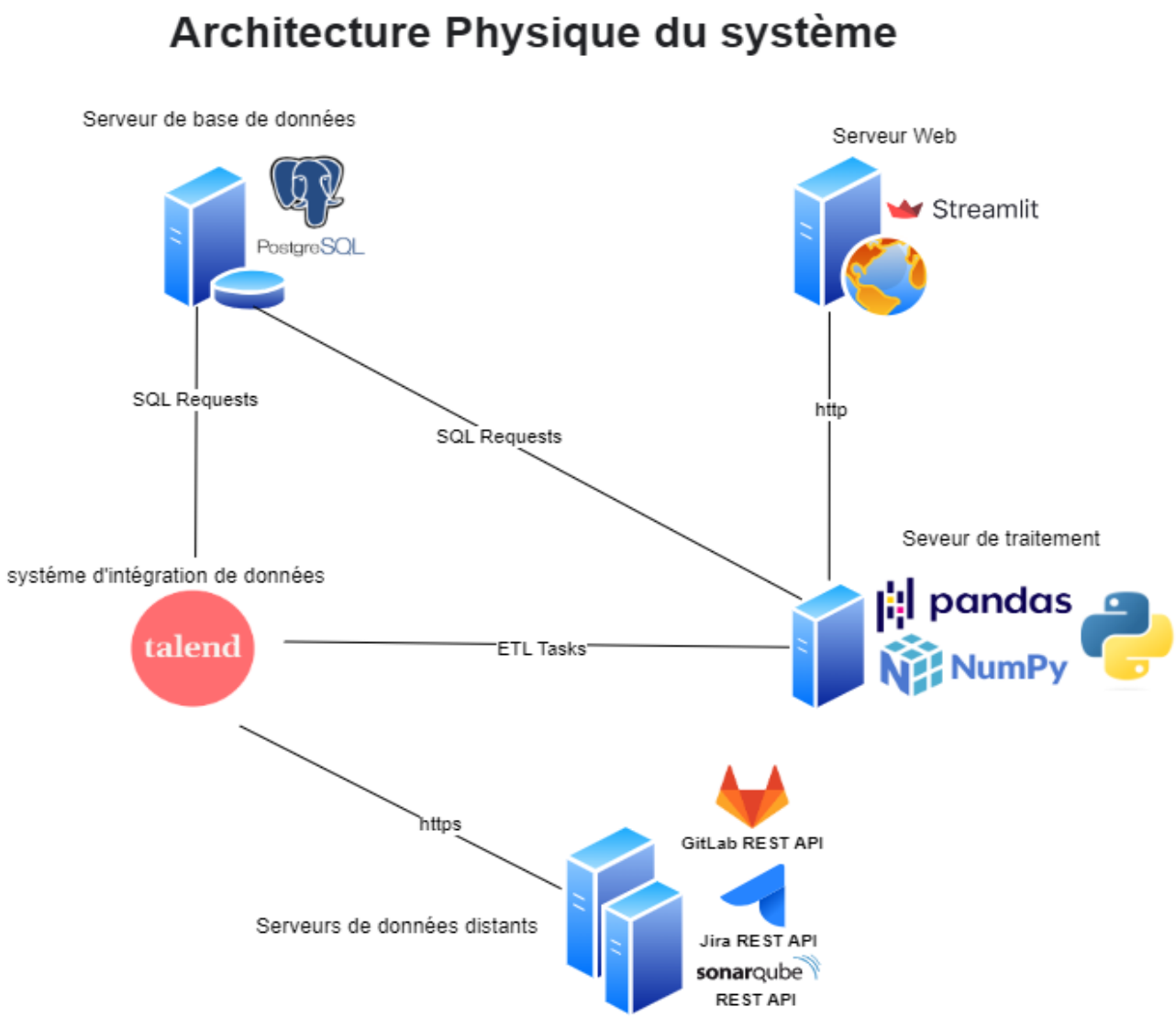


Figure 3.2 : Architecture physique de «Snowflake Monitoring Application»

Notre architecture physique est composée de plusieurs éléments essentiels qui travaillent en tandem pour permettre la collecte, le traitement, le stockage et la présentation des données de manière optimale. Voici un aperçu détaillé de ces composants :

1- **Tier des Services (Tier Application) :**

Ce tier comprend les différents serveurs d'application hébergeant les microservices du système.

Il inclut les serveurs pour les services DAGS_MONITOR, WAREHOUSE_USAGE, QUERY_PERFORMA

ETL_service, AUTH_service, etc.

Ces serveurs sont responsables de la logique métier et du traitement des données.

2- **Tier des Données :**

Ce tier est composé du serveur de base de données hébergeant la base de données MONITORING_DB.

Il est chargé du stockage et de la gestion des données de monitoring collectées. Le serveur Snowflake, hébergeant la base de données SNOWFLAKE_SYSTEM_DB, fait également partie de ce tier des données.

3- **Tier de présentation :**

Ce tier comprend le serveur frontend hébergeant la couche FRONTEND LAYER.

Il est responsable de la présentation des données de monitoring aux utilisateurs via les tableaux de bord et les interfaces.

4- **Tier de Communication :**

Ce tier est représenté par le serveur MQTT, qui héberge le broker MQTT. Il gère la communication asynchrone et découplée entre le systéme de notifications et le service ETL à l'aide du protocole MQTT.

Cette architecture en n-tiers offre plusieurs avantages clés, tels que la séparation des préoccupations, l'évolutivité, la résilience et la sécurité renforcée du système. Chaque tier étant responsable d'une fonction spécifique, il devient plus aisé de maintenir, de mettre à l'échelle et de sécuriser les différents composants de manière indépendante.

3.2 Étude conceptuelle

Dans cette section nous allons s'intéresser à la modélisation de notre projet à l'aide des différents modelés et diagrammes qui vont nous permettre de mieux comprendre le flux de travail ainsi la nature des données et les traitements qui circulent dans notre système.

3.2.1 Conception globale

Dans cette section, nous allons s'intéresser à presenter les differents diagrammes illustant la conception globale de notre solution.

3.2.1.1 Diagramme de paquetages

Les diagrammes de paquetages sont des diagrammes structurels qui illustrent l'organisation et la disposition de différents éléments modélisés sous forme de packages[13].

Il sert à grouper des éléments en un ensemble cohérent, et à fournir un espace de noms pour ces éléments.

La figure **3.3** qui suit représente le diagramme de paquetages de notre système :

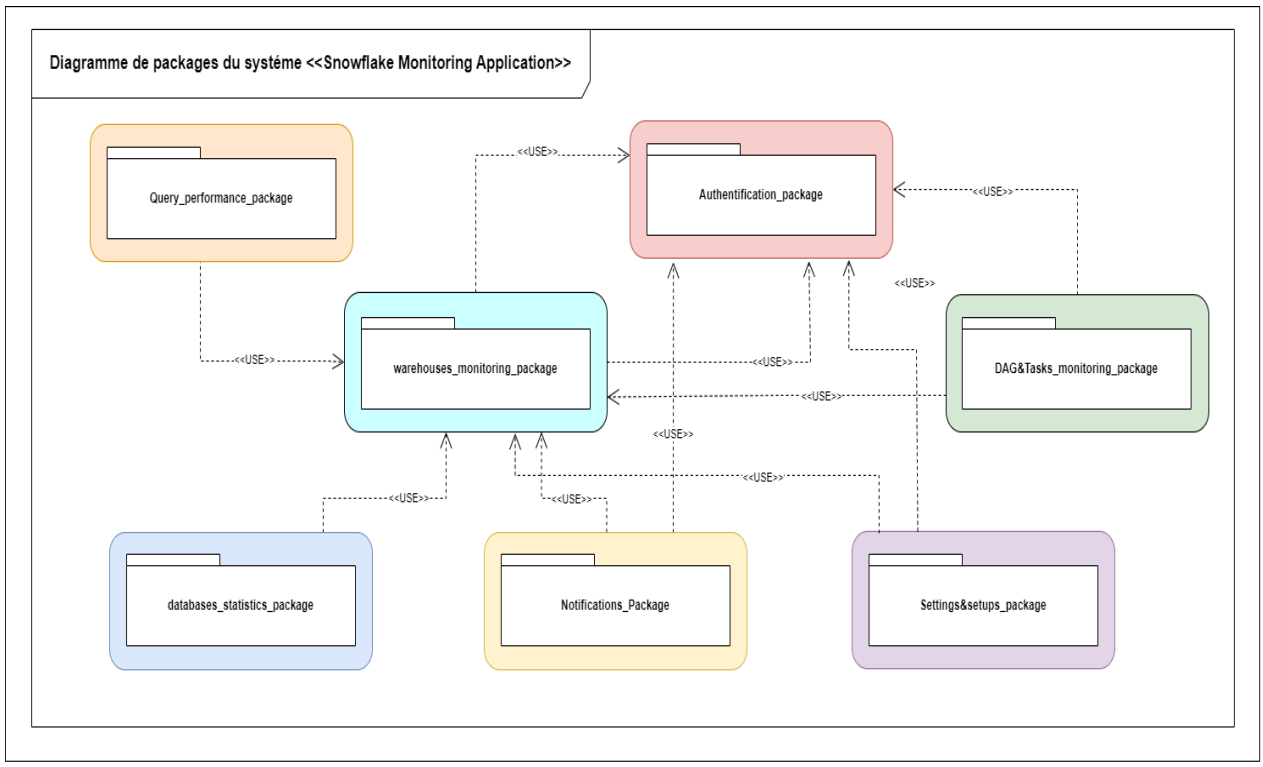


Figure 3.3 : Diagramme de paquetages de «Snowflake Monitoring Application»

3.2.1.2 Diagramme de classe

Les diagrammes de classes fournissent une représentation détaillée de la structure d'un système spécifique. Ils modélisent les classes du système, leurs attributs, leurs opérations ainsi que les relations entre les objets[14].

Ce diagramme permet de visualiser clairement la composition du système, facilitant ainsi la compréhension des interactions et des dépendances entre les différents éléments. En représentant les attributs et les méthodes des classes, ils aident également à définir les responsabilités et les comportements des objets dans le contexte global du système.

La figure **3.4** qui suit représente le diagramme de classe de notre système :

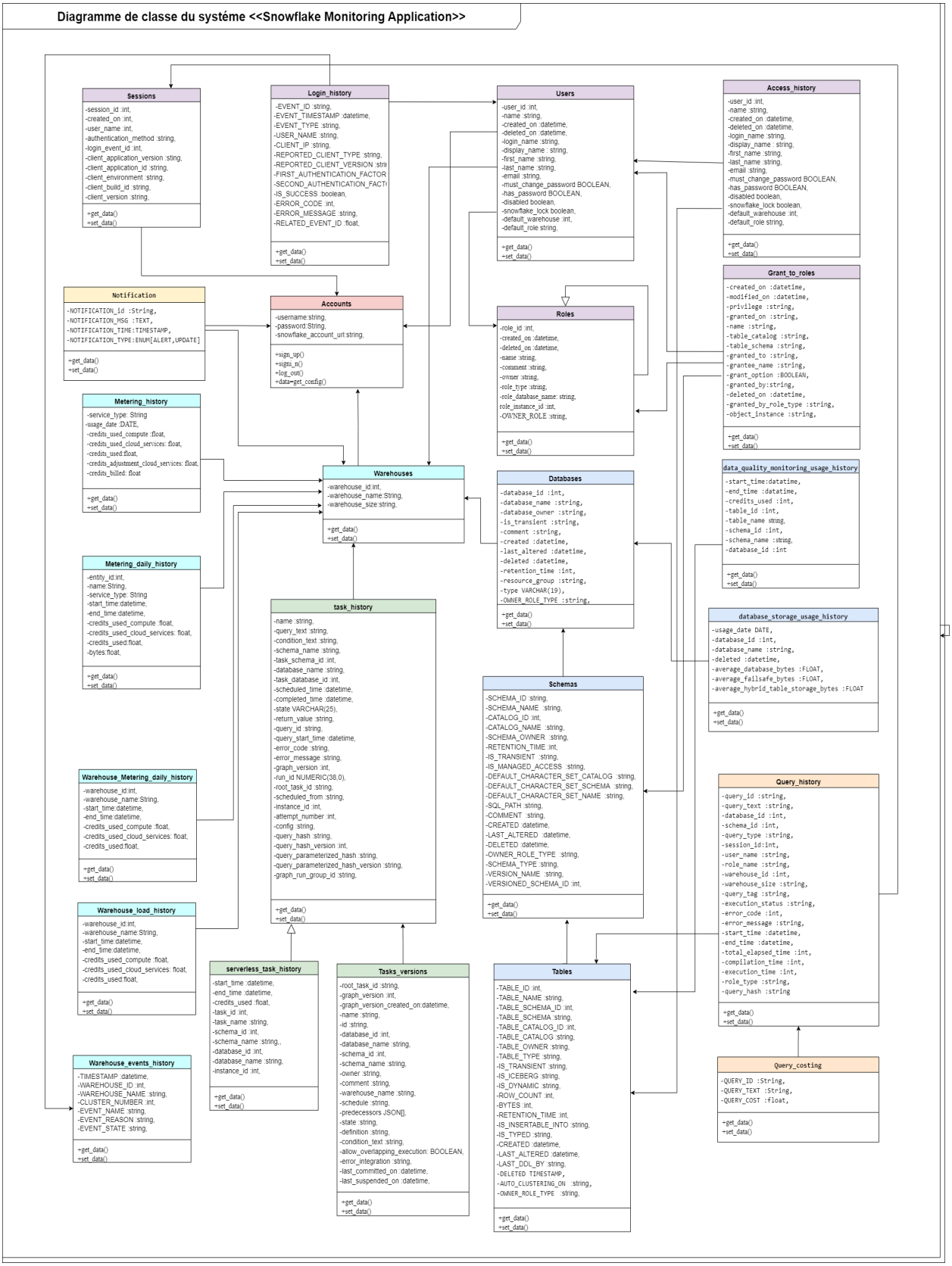


Figure 3.4 : Diagramme de classe de «Snowflake Monitoring Application»

3.2.2 Conception detaillée

Les diagrammes d'activités démontrent la logique d'un algorithme

3.2.2.1 Diagrammme d'activité du micro-service «ETL-service» :

Un diagramme d'activité fournit une vue détaillée du comportement d'un système en décrivant la séquence d'actions au sein d'un processus[15].

Le diagramme d'activité du micro-service ETL-service est représenté dans la figure **3.5** suivante :

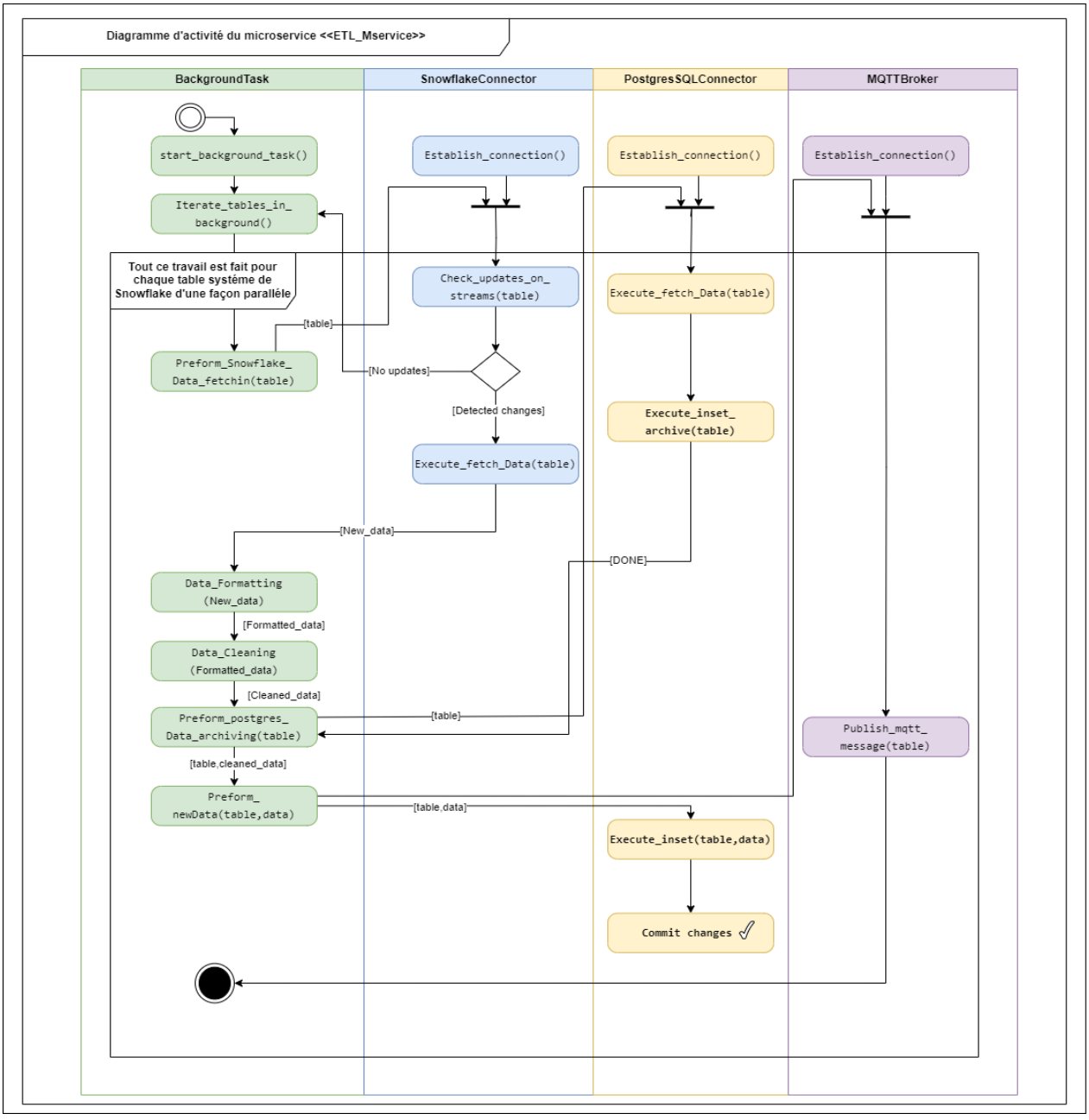


Figure 3.5 : Diagramme d'activité du micro-service «ETL-service»

Le traitement du micro-service ETL commence par l'appele la méthode **«start_background_task()»**, qui est une méthode programmée automatiquement chaque 24 heurs. En lancant cette méthode, elle parcourt les tables du système Snowflake, dont nous voulons projeter, en utilisant la méthode **«iterate_tables_in_background()»**.

À ce stade, le connecteur Snowflake établit une connexion avec le système Snowflake via la méthode **«establish_connection()»**, qui à ca part connecte notre service avec le compte Snowflake dont nous souhaitons monitorer ces données plus tard.

Ensuite, le contrôleur système vérifie s'il y a des mises à jour sur les tables Snowflake en appelant la méthode **«check_updates_on_streams(tables)»**.

Si il y des changements détectés, la méthode **«execute_fetch_data(table)»** est exécutée pour récupérer les nouvelles données de la table Snowflake.

Les nouvelles données récupérées sont ensuite transmises à l'étape de formatage des données **«Data_Formatting()»**, où elles sont mises en forme(typage, longeur, format etc.). Après cela, l'étape de nettoyage des données **«Data_Cleaning()»** intervient pour préparer et nettoyer les données formatées.

postérieurement, l'étape d'archivage des données dans PostgreSQL **«Preform_Data_archiving»**, là où nous allons transformer les anciennes données de cette table vers une base de données d'archivage afin d'avoir toujours une traçabilité des données.

Parallèlement, le connecteur PostgreSQL établit une connexion avec les base de données PostgreSQL, **monitoring_db** et la base d'archive **backup_db**, via la méthode **«establish_connection()»**. tout en éxécutant la méthode **«execute_insert_archive(table, data)»**.

Par la suite,le système déclanche la methode **«Preform_newData(table,data)»** pour insérer les nouveaux données nettoyées dans la base de données de monitoring. Après l'insertion des données, le microservice effectue un commit des changements dans la base de données.

il faut notez bien que, ce flux de travail ce repéte éventuellement pour chaque table, séparament ou/et séquentiellement pour les tables qui on une relation entre eux, dans un **«thread»** à part.

Aprés la mise à jour de tout les tables, il publie un message MQTT via la méthode **«publish_mqtt_message()»** afin de notifier les autres composants de l'ajout de nouvelles données. Cette approche garantit que toutes les étapes sont correctement coordonnées et que chaque composant est informé des mises à jour de manière efficace et en temps réel.

Enfin, le microservice ETL-service termine son traitement en attendant son prochain déclanchement.

Conclusion

Ce chapitre a établi les bases pour la mise en œuvre technique de notre projet. Les choix architecturaux et conceptuels s'avèrent essentiels pour assurer la stabilité et la performance de notre solution. Dans le prochain chapitre, nous plongerons dans la réalisation concrète de notre projet.

Chapitre 4

Réalisation

Plan

Introduction

Le quatrième chapitre se consacre à la mise en œuvre concrète du projet. Nous explorons les choix techniques que nous avons adoptés et détaillons le travail effectué pour donner vie à notre solution. De plus, nous parcourons notre workflow en cinq étapes, de l'extraction à la visualisation des données.

4.1 Choix technologiques

Dans cette partie, nous explorerons les décisions stratégiques que nous avons prises concernant les technologies et les outils que nous avons choisis pour la réalisation de notre solution.

- **Talend :**

  Talend est un outil d'intégration de données open source qui facilite l'extraction, la transformation et le chargement (ETL) des données[16],Son logo est représenté par la figure **4.1**. Il a été utilisé pour gérer le flux de données, les transformations et l'intégration des données dans la base de données.

Figure 4.1 : Logo du Talend[16]

- **PostgreSQL :**

  PostgreSQL est un SGBDR open source **pq**, Son logo est représenté par la figure **4.2**. Il a été choisi pour stocker et gérer les données collectées dans le projet.

Figure 4.2 : Logo du PostgreSQL[17]

- **Streamlit :**

  Streamlit est un framework Python qui facilite la création d'applications web interactives pour la visualisation de données[18]. Il a été choisi pour concevoir des tableaux de bord interactifs permettant de visualiser les résultats de l'analyse de données de manière conviviale. Son logo est représenté par la figure **4.3**.

Figure 4.3 : Logo du Streamlit[18]

- **Jira REST API :**

  Jira REST API était nécessaire pour extraire des données de Jira. Cela a automatisé la collecte de données, assuré la cohérence et amélioré l'efficacité de l'analyse[19]. Son logo est représenté par la figure **4.4**.

Figure 4.4 : Logo du Jira [19]

- **Gitlab REST API :**

  l'API Gitlab REST a été utilisée pour extraire des données du système de gestion de versions Gitlab. Elle a permis d'automatiser la collecte de données liées au code source, aux commits et aux mesures de performance de l'équipe, renforçant ainsi l'efficacité de l'analyse[20]. Son logo est représenté par la figure **4.5**.

Figure 4.5 : Logo du Gitlab [20]

- **SonarQube REST API :**

  SonarQube est un outil d'analyse statique du code qui identifie et corrige les problèmes de qualité du code [21], son logo est représenté par la figure **4.6**. Son API a permis d'intégrer l'analyse de la qualité du code dans le processus, renforçant ainsi la surveillance et l'amélioration continues.

Figure 4.6 : Logo du sonarQube [21]

- **Numpy :**

  NumPy est une bibliothèque essentielle pour le calcul numérique en Python [22], son logo est représenté par la figure **4.7**. Elle offre des structures de données et des fonctions pour effectuer des opérations mathématiques et statistiques sur des tableaux, ce qui est précieux pour la manipulation et l'analyse des données dans le projet.

Figure 4.7 : Logo du Numpy[22]

- **Pandas :**

  Pandas est une bibliothèque de manipulation de données Python qui fournit des structures de données flexibles pour l'analyse des données[23] , son logo est représenté par la figure **4.8**. Elle

facilite l'importation, la manipulation et l'analyse des données provenant de diverses sources, ce qui en fait un choix idéal pour votre projet.

Figure 4.8 : Logo du Pandas[23]

4.2 Exécution du workflow

Gérer efficacement le flux de travail est d'une importance capitale pour notre projet, car il guide chacune des étapes du processus, depuis l'extraction initiale des données jusqu'à leur transformation, leur stockage intérimaire et leur analyse.

Dans cette section, nous plongerons en profondeur dans chacune des étapes du flux de travail, exposant les méthodes, les outils et les solutions pour relever les défis qui se présentent. Notre objectif est de présenter un processus fluide et cohérent, démontrant comment chaque phase contribue à la qualité des données et à l'évaluation de la performance de l'équipe Avaxia.

4.2.1 Exctraction des données

Dans la phase initiale de notre projet, nous avons commencé par extraire des données à partir de plusieurs sources essentielles, dont Jira et GitHub. Cette extraction de données s'est effectuée en utilisant des REST API dédiées à chaque plateforme, ce qui a permis une collecte systématique et automatisée des informations et nous avons procédé comme suit :

- **Gestion de la sécurité des données :** la sécurité des données a constitué une préoccupation majeure durant cette étape initiale. Nous avons mis en place des mesures de sécurité rigoureuses pour préserver l'intégrité des informations sensibles. Cela englobait des processus d'authentification et d'autorisation d'accès aux API, ainsi que des protocoles de stockage des données extraites garantissant leur confidentialité.
- **Défi de la diversité des sources de données :** un défi significatif auquel nous avons fait face était la disparité des formats de données entre les différentes plateformes. Chacune d'entre elles présentait sa propre structure et ses schémas de données distincts. Pour s'assurer de la cohérence et de la comparabilité des données extraites, nous avons dû les normaliser. Cela

impliquait de les convertir dans un format standardisé et compatible avec notre système de gestion de base de données.

À la fin de cette étape, nous avons disposé d'une quantité considérable de données brutes provenant de diverses sources. Ces données représentaient la base sur laquelle nous allions construire nos analyses ultérieures.

Figure 4.9 : Exemple d'extraction des données "Project Extractions"

Les figures qui suit projette des exemples des données extraites :

- Les données représentées dans la figure **4.10** suivante illustres une partie de la liste des projets de Avaxia :

Figure 4.10 : Extrait des données collectés des projects

- Les données représentées dans la figure **4.11** suivante illustres une partie des tableaux Jira(boards) dont les projets sont affectés.

Figure 4.11 : Extrait des données collectées des tableaux jira

- Les données représentées dans la figure **4.12** suivante illustres une partie des la liste des utilisateurs(collaborateurs).

Figure 4.12 : Extrait des données collectées des utilisateurs

- Les données représentées dans la figure **4.13** suivante une partie des la liste des tickets jira(issues).

Figure 4.13 : Extrait des données collectées des tickets

L'extraction des données constitue une phase cruciale, car elle fournit la matière première fondamentale pour notre projet. Ces données brutes forment la base de notre analyse ultérieure. Il est toutefois important de noter que cette étape ne représente que le point de départ de notre workflow. Les données extraites nécessitent encore une préparation, un nettoyage et une analyse approfondie pour en extraire des informations significatives, un aspect que nous explorerons en détail dans les prochaines phases.

4.2.2 Transformation des données

Au cours de cette étape cruciale, nous avons traité les données brutes extraites de diverses sources pour les préparer à l'analyse ultérieure. Cette phase de transformation était essentielle pour s'assurer que les données étaient cohérentes, complètes et prêtes à être exploitées. Voici comment nous avons procédé avec plus de détails :

- **Exploration des données :** nous avons commencé par explorer les données brutes pour comprendre leur structure et leur qualité. Cela incluait l'examen des différents formats, des types de données, des valeurs manquantes, des doublons et d'autres caractéristiques.
- **Nettoyage des données :** pour garantir la qualité des données, nous avons entrepris un processus de nettoyage rigoureux. Cela impliquait la correction des valeurs incorrectes, la

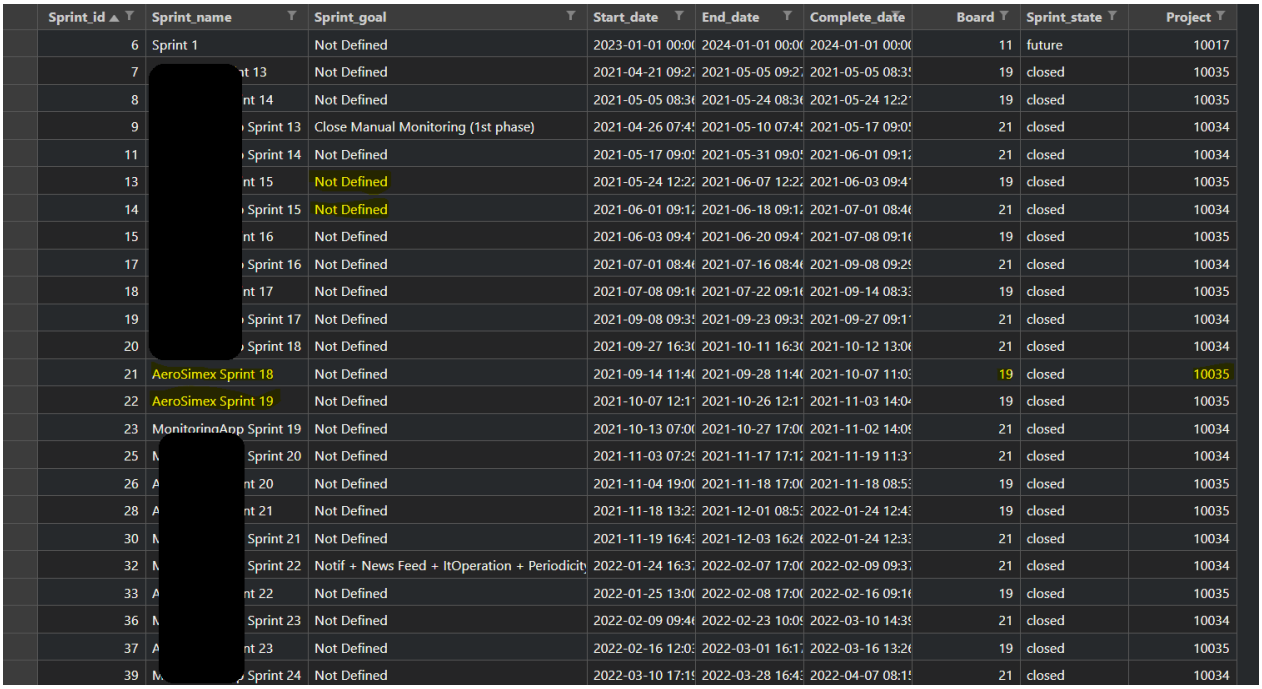
suppression des doublons, le comblement des valeurs manquantes, et la normalisation des données lorsque cela était nécessaire.

- **Transformation et structuration :** Les données brutes étant souvent non structurées, nous les avons transformées en une forme structurée. Cela comprenait la création de schémas de données cohérents et l'organisation des informations en fonction de nos besoins d'analyse.
- **Filtrage des données :** certaines données étaient excessivement volumineuses ou non pertinentes pour notre objectif d'analyse. Nous avons donc appliqué des filtres pour ne conserver que les données pertinentes et significatives.
- **Enrichissement des données :** dans certains cas, nous avons enrichi les données en ajoutant des informations supplémentaires à partir d'autres sources. Cela visait à renforcer la qualité et la pertinence des données.
- **Stockage intermédiaire :** les données transformées ont été stockées de manière sécurisée dans des fichiers CSV, prêtes à être utilisées pour l'analyse ultérieure.

les figures **4.14** et **4.15** suivantes illustre des exemples des données brutes et redondantes avant et après la phase de transformation des données qui sont stocker dans des fichiers CSV.

Board Name	Board Type	Sprint Name
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28
AeroSimex Board	scrum	AeroSimex Sprint 28

FIGURE 4.14 : Exemple des données brutes et erronées avec doublements des valeurs



Sprint_id	Sprint_name	Sprint_goal	Start_date	End_date	Complete_date	Board	Sprint_state	Project
6	Sprint 1	Not Defined	2023-01-01 00:0(	2024-01-01 00:0(	2024-01-01 00:0(	11	future	10017
7	[redacted]nt 13	Not Defined	2021-04-21 09:2	2021-05-05 09:2	2021-05-05 08:3	19	closed	10035
8	[redacted]nt 14	Not Defined	2021-05-05 08:3(	2021-05-24 08:3(	2021-05-24 12:2	19	closed	10035
9	[redacted] Sprint 13	Close Manual Monitoring (1st phase)	2021-04-26 07:4	2021-05-10 07:4	2021-05-17 09:0	21	closed	10034
11	[redacted] Sprint 14	Not Defined	2021-05-17 09:0	2021-05-31 09:0	2021-06-01 09:1	21	closed	10034
13	[redacted]nt 15	Not Defined	2021-05-24 12:2	2021-06-07 12:2	2021-06-03 09:4	19	closed	10035
14	[redacted] Sprint 15	Not Defined	2021-06-01 09:1	2021-06-18 09:1	2021-07-01 08:4(	21	closed	10034
15	[redacted]nt 16	Not Defined	2021-06-03 09:4	2021-06-20 09:4	2021-07-08 09:1(	19	closed	10035
17	[redacted] Sprint 16	Not Defined	2021-07-01 08:4(	2021-07-16 08:4(	2021-09-08 09:2	21	closed	10034
18	[redacted]nt 17	Not Defined	2021-07-08 09:1(	2021-07-22 09:1(	2021-09-14 08:3	19	closed	10035
19	[redacted] Sprint 17	Not Defined	2021-09-08 09:3	2021-09-23 09:3	2021-09-27 09:1	21	closed	10034
20	[redacted] Sprint 18	Not Defined	2021-09-27 16:3(	2021-10-11 16:3(	2021-10-12 13:0(	21	closed	10034
21	AeroSimex Sprint 18	Not Defined	2021-09-14 11:4(	2021-09-28 11:4(	2021-10-07 11:0	19	closed	10035
22	AeroSimex Sprint 19	Not Defined	2021-10-07 12:1	2021-10-26 12:1	2021-11-03 14:0	19	closed	10035
23	MonitoringApp Sprint 19	Not Defined	2021-10-13 07:0(	2021-10-27 17:0(	2021-11-02 14:0	21	closed	10034
25	[redacted] Sprint 20	Not Defined	2021-11-03 07:2	2021-11-17 17:1	2021-11-19 11:3	21	closed	10034
26	[redacted]nt 20	Not Defined	2021-11-04 19:0(	2021-11-18 17:0(	2021-11-18 08:5	19	closed	10035
28	[redacted]nt 21	Not Defined	2021-11-18 13:2	2021-12-01 08:5	2022-01-24 12:4	19	closed	10035
30	[redacted] Sprint 21	Not Defined	2021-11-19 16:4	2021-12-03 16:2(	2022-01-24 12:3	21	closed	10034
32	[redacted] Sprint 22	Notif + News Feed + ItOperation + Periodicity	2022-01-24 16:3	2022-02-07 17:0(	2022-02-09 09:3	21	closed	10034
33	[redacted]nt 22	Not Defined	2022-01-25 13:0(	2022-02-08 17:0(	2022-02-16 09:1(	19	closed	10035
36	[redacted] Sprint 23	Not Defined	2022-02-09 09:4(	2022-02-23 10:0	2022-03-10 14:3	21	closed	10034
37	[redacted]nt 23	Not Defined	2022-02-16 12:0	2022-03-01 16:1	2022-03-16 13:2(	19	closed	10035
39	[redacted] Sprint 24	Not Defined	2022-03-10 17:1	2022-03-28 16:4	2022-04-07 08:1	21	closed	10034

Figure 4.15 : Exemple des données des sprints après la phase de transformation

L'étape de transformation des données était une étape cruciale pour garantir que les informations extraites étaient fiables et prêtes pour l'analyse. Elle exigeait une attention méticuleuse aux détails, un processus systématique de nettoyage et de structuration, et une compréhension approfondie des spécificités des données de votre projet.

4.2.3 Chargement des Données

Après avoir extrait et transformé les données, la prochaine phase de notre workflow était le chargement des données. Au cours de cette étape, nous avons acheminé les données nettoyées et transformées vers notre système de gestion de base de données, en l'occurrence PostgreSQL. Voici un aperçu détaillé de cette étape cruciale :

- **Benchmarking et choix de PostgreSQl** : avant de prendre notre décision, nous avons effectué un benchmarking minutieux en comparant plusieurs systèmes de gestion de bases de données. PostgreSQL a émergé comme le choix optimal en raison de sa flexibilité, de ses performances exceptionnelles et de sa capacité à gérer des charges de travail complexes. Cette étape a jeté les bases pour une intégration réussie.

  la figure **4.16** illustre le benchmarking effectué par le code ci-dessous :

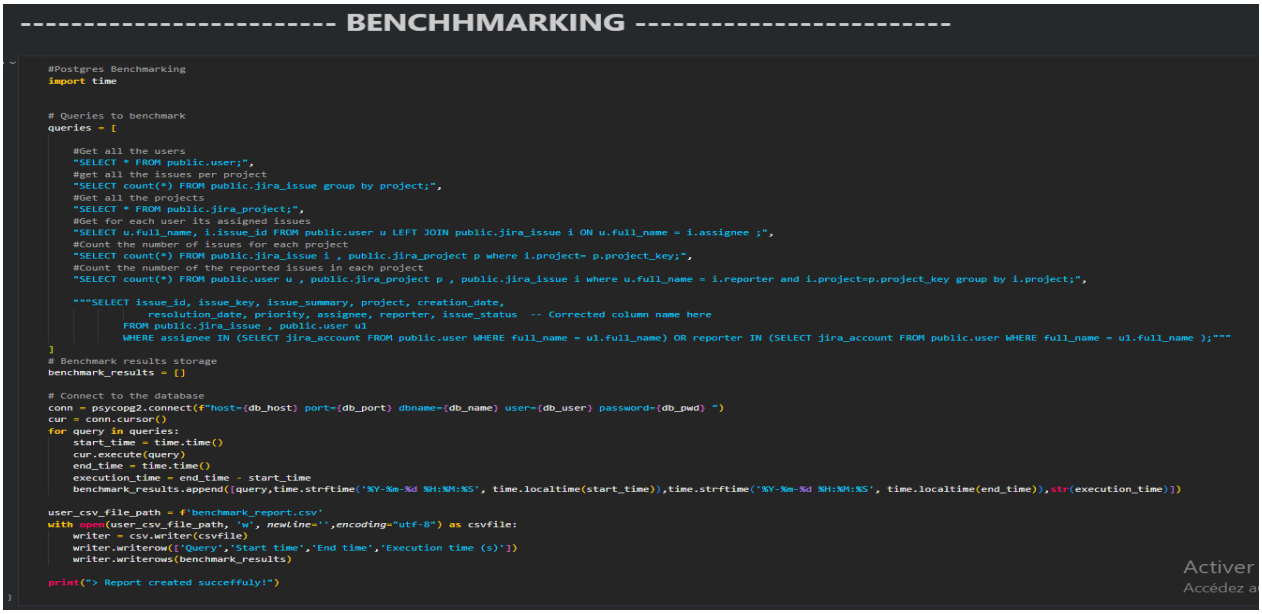


Figure 4.16 : Code de benchmarking d'une base de donné secondaire

la figure **4.17** suivante donne le rapport de benchmarking qui illustre la perfermance de la base de données postgreSQL en terme d'accés aux données :

Query	Start time	End time	Execution time (s)
SELECT * FROM public.user;	2023-07-26 14:32:27	2023-07-26 14:32:27	0.01
SELECT count(*) FROM public.jira_issue group by project;	2023-07-26 14:32:27	2023-07-26 14:32:27	0
SELECT * FROM public.jira_project;	2023-07-26 14:32:27	2023-07-26 14:32:27	0
SELECT u.full_name, i.issue_id FROM public.user u LEFT JOIN public.jira_issue i C	2023-07-26 14:32:27	2023-07-26 14:32:27	0
SELECT count(*) FROM public.jira_issue i , public.jira_project p where i.project=	2023-07-26 14:32:27	2023-07-26 14:32:27	0
SELECT count(*) FROM public.user u , public.jira_project p , public.jira_issue i wh	2023-07-26 14:32:27	2023-07-26 14:32:27	0
SELECT issue_id, issue_key, issue_summary,15.369215726852417, creation_date resolution_date, priority, assignee, reporter, issue_status -- Correctec FROM public.jira_issue , public.user u1 WHERE assignee IN (SELECT jira_account FROM public.user WHERE full_	2023-07-26 14:32:27	2023-07-26 14:32:42	15.37

Figure 4.17 : Rapport de benchmarking PostgreSQL

- **Intégration avec PostgreSQL** : nous avons créé des flux de données pour transférer les données transformées vers notre base de données PostgreSQL. Cette intégration a permis de stocker de manière centralisée toutes les informations nécessaires à nos analyses.

  la figure **4.18** suivante illustre le dshboard de la base de données sur PgAdmin(qui est l'outil de gestion de base de données PostgreSQL) :

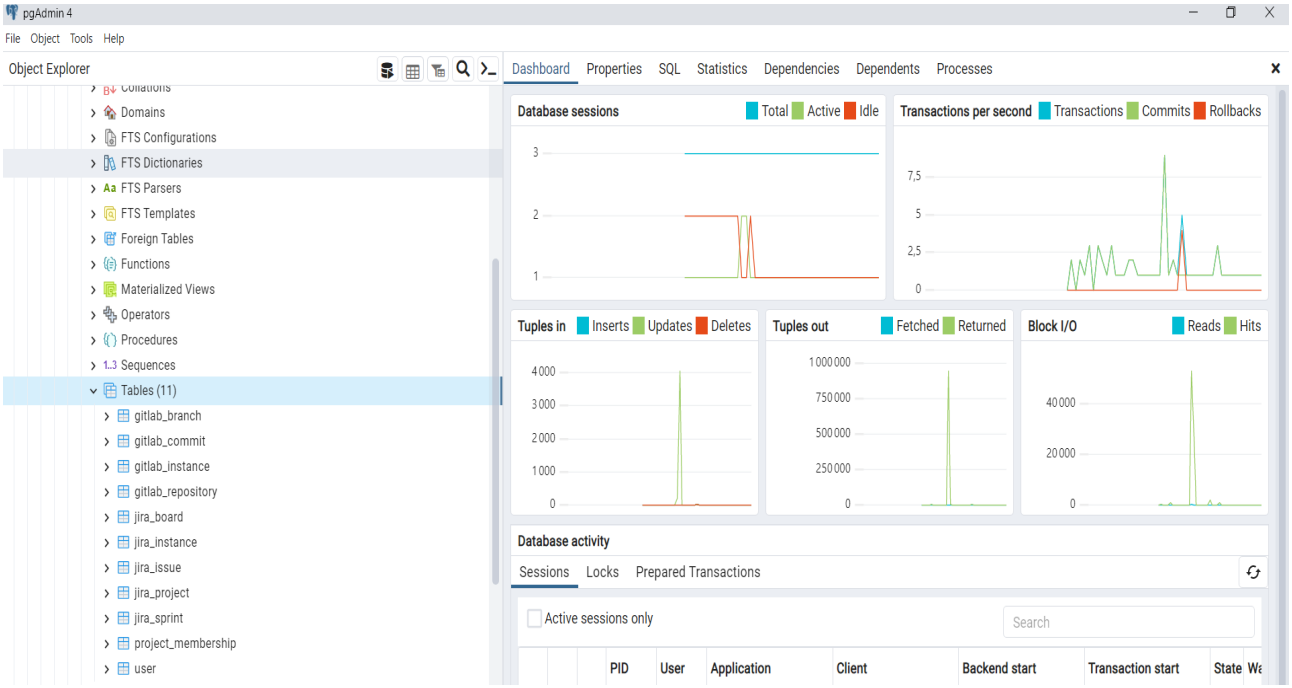
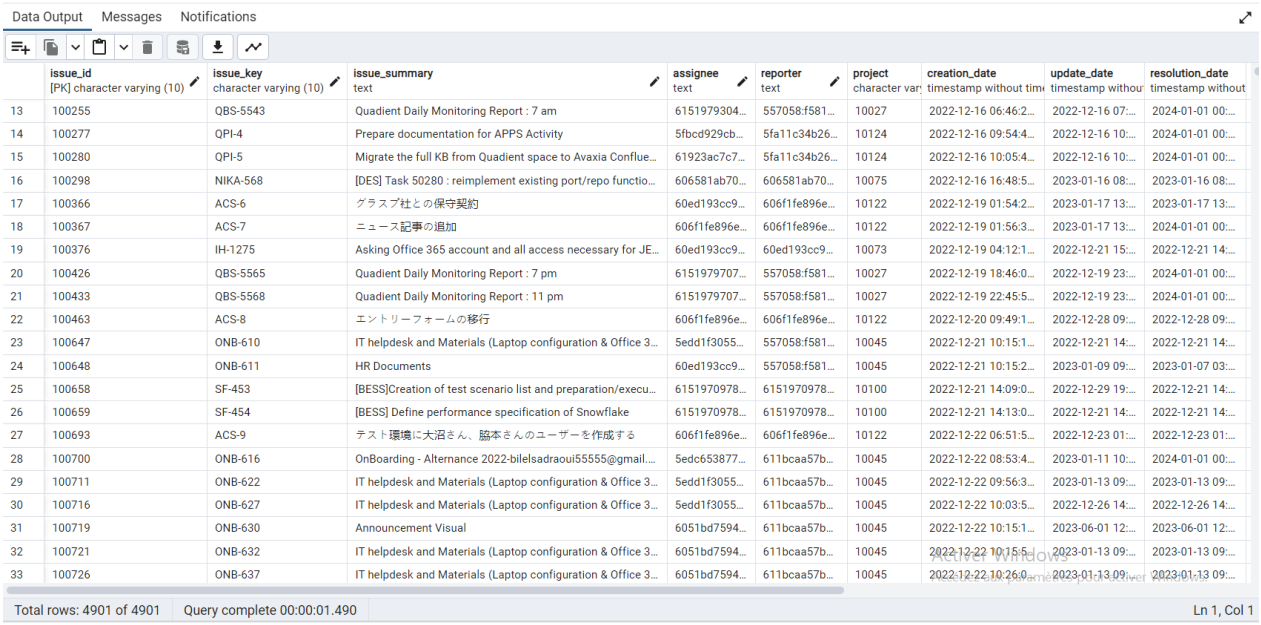


Figure 4.18 : Enter Caption

- **Garantie de la cohérence** : lors du chargement des données, il était essentiel de veiller à la cohérence et à l'intégrité des données. Des mécanismes de validation ont été mis en place pour éviter toute altération ou perte de données lors du transfert.
- **Sécurité des données** : la sécurité des données était une priorité constante. Nous avons mis en place des mesures de sécurité pour protéger les données sensibles tout au long de leur trajet vers la base de données.
- **Sauvegarde et récupération** : des procédures de sauvegarde régulières ont été mises en place pour garantir la disponibilité des données en cas de sinistre ou de défaillance du système.

les figures qui suit illustrent des échantillons des données stockées dans les différents tables de la base :

la figure **4.19** projette une partie des données de la table «JiraIssues»



Data Output | Messages | Notifications

	issue_id [PK] character varying (10)	issue_key character varying (10)	issue_summary text	assignee text	reporter text	project character var	creation_date timestamp without tim	update_date timestamp withou	resolution_date timestamp without
13	100255	QBS-5543	Quadient Daily Monitoring Report : 7 am	6151979304...	557058:f581...	10027	2022-12-16 06:46:2...	2022-12-16 07:...	2024-01-01 00:...
14	100277	QPI-4	Prepare documentation for APPS Activity	5fbcd929cb...	5fa11c34b26...	10124	2022-12-16 09:54:4...	2022-12-16 10:...	2024-01-01 00:...
15	100280	QPI-5	Migrate the full KB from Quadient space to Avaxia Conflue...	61923ac7c7...	5fa11c34b26...	10124	2022-12-16 10:05:4...	2022-12-16 10:...	2024-01-01 00:...
16	100298	NIKA-568	[DES] Task 50280 : reimplement existing port/repo functio...	606581ab70...	606581ab70...	10075	2022-12-16 16:48:5...	2023-01-16 08:...	2023-01-16 08:...
17	100366	ACS-6	グラスブ社との保守契約	60ed193cc9...	606f1fe896e...	10122	2022-12-19 01:54:2...	2023-01-17 13:...	2023-01-17 13:...
18	100367	ACS-7	ニュース記事の追加	606f1fe896e...	606f1fe896e...	10122	2022-12-19 01:56:3...	2023-01-17 13:...	2024-01-01 00:...
19	100376	IH-1275	Asking Office 365 account and all access necessary for JE...	60ed193cc9...	60ed193cc9...	10073	2022-12-19 04:12:1...	2022-12-21 15:...	2022-12-21 14:...
20	100426	QBS-5565	Quadient Daily Monitoring Report : 7 pm	6151979707...	557058:f581...	10027	2022-12-19 18:46:0...	2022-12-19 23:...	2024-01-01 00:...
21	100433	QBS-5568	Quadient Daily Monitoring Report : 11 pm	6151979707...	557058:f581...	10027	2022-12-19 22:45:5...	2022-12-19 23:...	2024-01-01 00:...
22	100463	ACS-8	エントリーフォームの移行	606f1fe896e...	606f1fe896e...	10122	2022-12-20 09:49:1...	2022-12-28 09:...	2022-12-28 09:...
23	100647	ONB-610	IT helpdesk and Materials (Laptop configuration & Office 3...	5edd1f3055...	557058:f581...	10045	2022-12-21 10:15:1...	2022-12-21 14:...	2022-12-21 14:...
24	100648	ONB-611	HR Documents	60ed193cc9...	557058:f581...	10045	2022-12-21 10:15:2...	2023-01-09 09:...	2023-01-07 03:...
25	100658	SF-453	[BESS]Creation of test scenario list and preparation/execu...	6151970978...	6151970978...	10100	2022-12-21 14:09:0...	2022-12-29 19:...	2022-12-21 14:...
26	100659	SF-454	[BESS] Define performance specification of Snowflake	6151970978...	6151970978...	10100	2022-12-21 14:13:0...	2022-12-21 14:...	2022-12-21 14:...
27	100693	ACS-9	テスト環境に大沼さん、脇本さんのユーザーを作成する	606f1fe896e...	606f1fe896e...	10122	2022-12-22 06:51:5...	2022-12-23 01:...	2022-12-23 01:...
28	100700	ONB-616	OnBoarding - Alternance 2022-bilelsadraoui55555@gmail....	5edc653877...	611bcaa57b...	10045	2022-12-22 08:53:4...	2023-01-11 10:...	2024-01-01 00:...
29	100711	ONB-622	IT helpdesk and Materials (Laptop configuration & Office 3...	5edd1f3055...	611bcaa57b...	10045	2022-12-22 09:56:3...	2023-01-13 09:...	2023-01-13 09:...
30	100716	ONB-627	IT helpdesk and Materials (Laptop configuration & Office 3...	5edd1f3055...	611bcaa57b...	10045	2022-12-22 10:03:5...	2022-12-26 14:...	2022-12-26 14:...
31	100719	ONB-630	Announcement Visual	6051bd7594...	611bcaa57b...	10045	2022-12-22 10:15:1...	2023-06-01 12:...	2023-06-01 12:...
32	100721	ONB-632	IT helpdesk and Materials (Laptop configuration & Office 3...	6051bd7594...	611bcaa57b...	10045	2022-12-22 10:15:5...	2023-01-13 09:...	2023-01-13 09:...
33	100726	ONB-637	IT helpdesk and Materials (Laptop configuration & Office 3...	6051bd7594...	611bcaa57b...	10045	2022-12-22 10:26:0...	2023-01-13 09:...	2023-01-13 09:...

Total rows: 4901 of 4901 | Query complete 00:00:01.490 | Ln 1, Col 1

Figure 4.19 : Lignes de la table «JiraIssues»

L'étape de chargement des données était fondamentale pour centraliser nos informations dans une base de données prête à être consultée et analysée. Ces données étaient maintenant accessibles de manière efficace, ce qui a constitué la base de notre workflow d'analyse des performances de l'équipe Avaxia.

4.2.4 Analyse des données

Une fois les données extraites, transformées et chargées dans notre base de données, nous avons progressé vers la phase cruciale de l'analyse des données. L'objectif était de tirer des enseignements concrets à partir des données brutes que nous avions recueillies. Voici comment cette étape s'est déroulée en détail :

- **Exploration des données :** nous avons initié cette phase par une exploration méticuleuse des données. Notre démarche consistait à visualiser les données sous différentes perspectives, à identifier les tendances et les valeurs aberrantes, et à acquérir une compréhension profonde de la structure des informations disponibles. Cette exploration initiale a fourni des indications cruciales sur la nature des données, orientant ainsi nos futures analyses.

  Les figures **4.20** et **4.21** suivantes représentent quelques fonctions utilisées pour l'extraction des corrélations et à identifier les tendances et les valeurs aberrantes :

```python
def get_user_stats(selected_user):
    # Get the number of projects the user is involved in
    project_count_query = f"""
        SELECT COUNT(DISTINCT project) AS num_projects
        FROM public.project_membership
        WHERE user IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}');
    """
    num_projects = pd.read_sql_query(project_count_query, con=engine)['num_projects'][0]

    # Get the number of open tickets for the user
    open_tickets_query = f"""
        SELECT COUNT(*) AS num_open_tickets
        FROM public.jira_issue
        WHERE (assignee IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}') OR
               reporter IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}')) AND
              issue_status = 'open';
    """
    num_open_tickets = pd.read_sql_query(open_tickets_query, con=engine)['num_open_tickets'][0]

    # Get the number of resolved tickets for the user
    resolved_tickets_query = f"""
        SELECT COUNT(*) AS num_resolved_tickets
        FROM public.jira_issue
        WHERE (assignee IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}') OR
               reporter IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}')) AND
              issue_status = 'resolved';
    """
    num_resolved_tickets = pd.read_sql_query(resolved_tickets_query, con=engine)['num_resolved_tickets'][0]

    # Get the average resolution time for the user
    avg_resolution_time_query = f"""
        SELECT AVG(DATE_PART('day', resolution_date - creation_date)) AS avg_resolution_time
        FROM public.jira_issue
        WHERE (assignee IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}') OR
               reporter IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}')) AND
              resolution_date IS NOT NULL;
    """
    avg_resolution_time = pd.read_sql_query(avg_resolution_time_query, con=engine)['avg_resolution_time'][0]

    return num_projects, num_open_tickets, num_resolved_tickets, avg_resolution_time
```

Figure 4.20 : Un bout de code qui traite les status des utilisateurs selon leurs affectations aux tickets «USER»

```python
def update_user_table(selected_user):
    # Call the get_user_stats function to get the data
    num_projects, num_open_tickets, num_resolved_tickets, avg_resolution_time = get_user_stats(selected_user)

    # Generate the HTML table rows for the selected user
    table_rows = [
        html.Tr([
            html.Td(selected_user),
            html.Td(num_projects),
            html.Td(num_open_tickets),
            html.Td(num_resolved_tickets),
            html.Td(avg_resolution_time),
        ])
    ]

    return table_rows
```

Figure 4.21 : Un bout de code qui traite la table de données «USER»

- **Analyse statistique :** une fois que nous avons sondé en profondeur les données, nous avons entrepris des analyses statistiques approfondies. Ces analyses ont fait appel à des techniques de calcul, des méthodes statistiques et des modèles prédictifs pour extraire des informations significatives. Nous avons effectué des calculs de corrélation, des tests statistiques et développé des modèles prédictifs pour évaluer la performance de l'équipe Avaxia.

  Grâce à ces modelés développer, nous avons réussi à extraire des corrélations entre les données de notre base qui ont été représenté ultérieurement par des graphes de corrélation.

La figure **4.22**représente le graphe de corrélation entre le temps des resolution des tickets et le nombres des tickets affectés à chaque collaborateur :

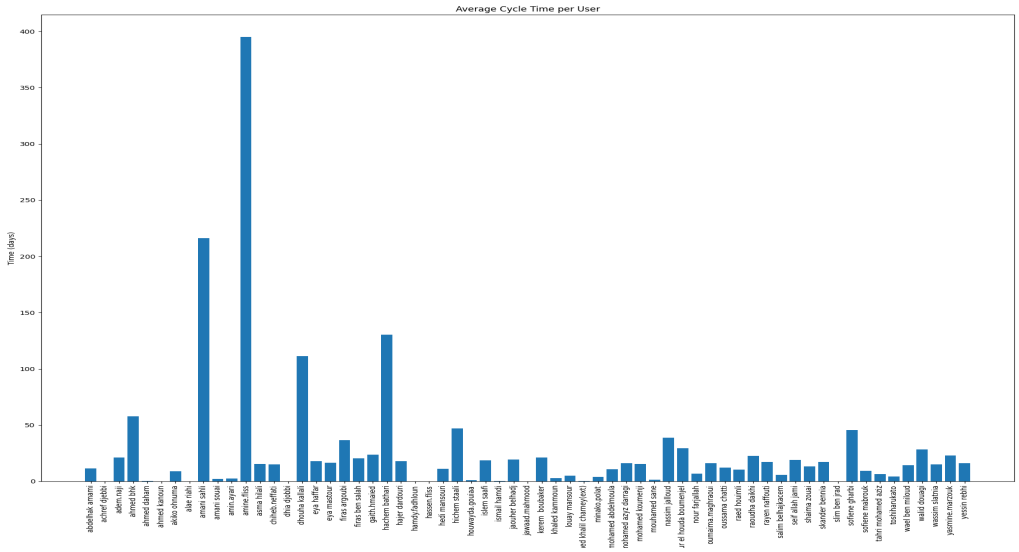


Figure 4.22 : Graphe de temps moyen de résolution des tickets par utilisateur

D'une part, on a constaté que cette corrélation a un impact important sur le temps de résolution des tickets et à la qualité du travail de ce collaborateur, de tel façon que : plus un utilisateur a de tickets assignés, plus son temps de résolution moyen est long, ce qui suggère une charge de travail plus lourde.

D'une autre part, ces modelés aussi ont relevé une autre corrélation entre le nombres des tickets assignées à chaque collaborateur et sa productivité. En effet, les utilisateurs qui ont un grand nombre de tickets assignés sont également les plus productifs, ce qui pourrait indiquer une efficacité dans la gestion de leur charge de travail qui est représenté par la figure **4.23**.

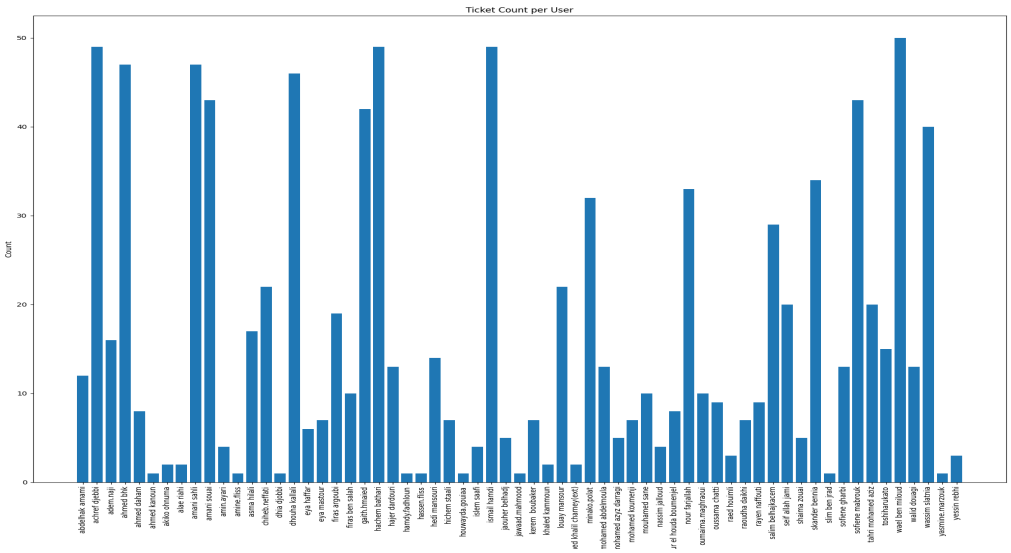


Figure 4.23 : Graphe de productivité par rapport au nombre de tickets assignés à chaque collaborateur

- **Identification des tendances et des axes d'amélioration :** l'analyse des données a mis en lumière des tendances dans les performances de l'équipe. Nous avons identifié les domaines où l'équipe excellait, ainsi que ceux qui nécessitaient des améliorations. Ces constats ont été partagés avec l'équipe pour orienter les décisions et les actions futures.

  la figure **4.24** suivante montre une partie de la fonction qui aide à étudier les KPI de l'équipe dont on va utilisé ultérieurement dans le dashbord.

```python
def extract_kpis(selected_user):
    # Query Jira issue data and user data from the database
    issue_query = f"""
        SELECT issue_id, issue_key, issue_summary, project, creation_date,
            resolution_date, priority, assignee, reporter, issue_status  -- Corrected column name here
        FROM public.jira_issue
        WHERE assignee IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}') OR reporter IN (SELECT jira_account FROM public.user WHERE full_name = '{selected_user}');
    """
    issue_df = pd.read_sql_query(issue_query, con=engine)

    # Merge the 'JiraIssue' table with the 'Project' table to get project keys
    project_query = "SELECT project_id, project_key FROM public.jira_project;"
    project_df = pd.read_sql_query(project_query, con=engine)

    issue_df = pd.merge(issue_df, project_df, left_on='project', right_on='project_id')

    # Calculate 'cycle_time' as the difference between 'resolution_date' and 'creation_date'
    issue_df['cycle_time'] = (pd.to_datetime(issue_df['resolution_date']) - pd.to_datetime(issue_df['creation_date'])).dt.days

    # Calculate KPIs for the selected user
    ticket_count = issue_df.shape[0]
    cycle_time_mean = issue_df['cycle_time'].mean()

    # Calculate priority distribution for the selected user
    user_priority_distribution = issue_df['priority'].value_counts()

    # Calculate issue count per project for the selected user using project_key
    tickets_per_project = issue_df.groupby('project_key')['issue_id'].count()

    # Calculate issue status distribution per project for the selected user
    project_issue_status_distribution = issue_df.groupby(['project_key', 'issue_status'])['issue_id'].count().unstack(fill_value=0)

    # Get unique project keys
    unique_project_keys = issue_df['project_key'].unique()

    # Generate a color palette with enough unique colors for all project keys
    num_project_keys = len(unique_project_keys)
    color_palette = pc.qualitative.Plotly[:num_project_keys]

    # Create a dictionary mapping project keys to colors
    project_key_color_dict = {project_key: color for project_key, color in zip(unique_project_keys, color_palette)}

    return ticket_count, cycle_time_mean, user_priority_distribution, tickets_per_project, project_issue_status_distribution, project_key_color_dict
```

Figure 4.24 : Fonction de l'extraction des KPI des collaborateurs

L'étape d'analyse des données s'est révélée cruciale pour transformer des données brutes en connaissances exploitables. Elle a joué un rôle central dans la prise de décisions éclairées visant à améliorer les performances de l'équipe et à atteindre les objectifs du projet.

4.2.5 Visualisation des données

Suite à l'analyse des données et à l'identification de tendances significatives, il devenait impératif de présenter les résultats de manière claire et concise à l'équipe Avaxia. Cette phase consistait à élaborer des rapports et des visualisations dans le but d'informer et de guider les membres de l'équipe dans leurs actions. Voici comment nous avons abordé cette dernière étape :

- **Élaboration de tableaux de bord :** nous avons utilisé l'outil Streamlit pour créer des tableaux de bord interactifs mettant en évidence les principales conclusions de notre analyse. Ces tableaux de bord offraient une vue synthétique des performances de l'équipe, en mettant en avant les tendances clés, et permettaient une exploration interactive des données.
- **Utilisation de visualisations graphiques :** les visualisations graphiques ont revêtu une importance capitale dans la présentation des résultats. Nous avons employé divers types de graphiques, de diagrammes et de représentations visuelles pour illustrer les données. Cela incluait des graphiques de tendance, des histogrammes, des diagrammes circulaires, ainsi que d'autres formes graphiques pertinentes.
- **Présentation aux membres de l'équipe :** nous avons organisé une réunion de présentation pour communiquer les résultats à l'équipe Avaxia. Lors de cette réunion, nous avons mis en évidence les conclusions majeures, répondu aux questions et entamé des discussions sur les implications pour l'équipe.
- **Feedback et échanges :** après la présentation, nous avons encouragé les membres de l'équipe à partager leurs retours, poser des questions et discuter des prochaines étapes. Cette interaction était cruciale pour garantir une bonne compréhension des résultats et la prise de décisions éclairées.

La figure **4.25**, représente des captures d'écrans extraites du page tableau de bord principale de notre solution dont chaqune d'eux affiches des représenataions graphiques des différents analyses globales d'Avaxia.

Figure 4.25 : Interface d'acceuil du tableau de bord

Les figures **4.26** représente des captures d'écrans extraites du page tableau de bord du section **«Projects Insghits»** dont on a choisi le projet **«AeroSimEx Project»**, son tableau **«jira AeroSimEx Board»** et le **«sprint 13»** pour afficher les différents statistiques de ces tickets.

Figure 4.26 : Interface des statistiques du projet «AeroSimEx Project»

La figure **4.27** représente des captures d'écrans extraites du page tableau de bord du section **«Users Insghits»** dont on a choisi l'utilisateur **«Nassim JALLOUD»** pour afficher ces différents statistiques.

Figure 4.27 : Interface des statistiques de l'utilisateur «Nassim JALLOUD»

L'étape de communication des constatations était essentielle pour s'assurer que les informations et les recommandations découlant de l'analyse des données étaient pertinentes et actionnables. Elle a fourni à l'équipe Avaxia une vue nette de ses performances et des domaines nécessitant des améliorations, contribuant ainsi à la réalisation des objectifs du projet.

Conclusion

Ce chapitre résume le travail accompli pour donner forme à notre projet. Les choix techniques et le workflow nous ont fourni un cadre solide pour concrétiser notre vision. Chaque étape de ce processus s'avère cruciale pour l'atteinte de nos objectifs.
Dans la conclusion générale à venir, nous réunirons l'ensemble de notre travail et soulignerons les perspectives pour l'avenir.

Conclusion générale

Dans un monde de plus en plus orienté vers les données, l'analyse des données joue un rôle crucial, particulièrement dans le domaine de l'informatique et de la technologie de l'information. Dans le cadre de notre projet au sein d'Avaxia Consulting, nous avons pu constater l'importance capitale de l'analyse des données pour obtenir des informations exploitables et prendre des décisions éclairées. Cette démarche ne se limite pas à une simple collecte de données, mais elle englobe une exploration approfondie, une interprétation et une transformation de ces données brutes en connaissances qui guident nos actions.

Au cœur de notre projet, l'analyse des données représente un catalyseur puissant, permettant d'optimiser les performances, de dégager des tendances significatives, et de cerner les domaines nécessitant des améliorations. Elle est le moteur qui propulse l'informatique vers des sommets de plus en plus élevés, contribuant ainsi au succès et à l'efficacité de notre équipe et, par extension, de l'ensemble du secteur des technologies de l'information.

Le premier chapitre de ce rapport a établi les bases de notre projet chez Avaxia Consulting. Nous avons présenté le contexte global du projet, mettant en lumière les objectifs que nous visons. De plus, nous avons donné un aperçu de l'organisme d'accueil, Avaxia Consulting, en expliquant ses services et son rôle dans le domaine de la technologie de l'information. Ce chapitre nous a fourni une solide fondation pour les étapes ultérieures du projet, en clarifiant la portée de notre travail et en soulignant notre ambition d'avoir un impact positif dans ce secteur en constante évolution.

Dans le deuxième chapitre, nous avons effectué une analyse minutieuse des besoins. Cela incluait une classification rigoureuse des besoins fonctionnels et non fonctionnels, ainsi qu'une définition claire du flux de travail et du backlog du produit. Ce chapitre a servi de fondement essentiel pour la phase ultérieure de planification et de développement.

Le troisième chapitre a été dédié à l'architecture et conception. Nous avons justifié notre choix d'architecture en couches et avons fourni des détails sur l'architecture logique et physique de notre solution.

Le quatrième chapitre s'est concentré sur la réalisation pratique du projet. Nous avons abordé en détail les choix techniques et le travail effectué pour donner vie à notre solution. De plus, nous avons décrit notre workflow en cinq étapes, de l'extraction à la visualisation des données. Chacune de ces étapes est cruciale pour la réussite du projet, et nous avons mis en place des mécanismes solides

pour garantir la qualité et la fiabilité de notre travail.

Le potentiel d'amélioration réside également dans les tableaux de bord que nous avons créés. Ils sont actuellement conçus pour fournir des informations essentielles, mais ils pourraient être étendus pour inclure des fonctionnalités plus avancées, telles que des analyses prédictives et des recommandations automatisées. Ces améliorations contribueraient à renforcer la capacité d'Avaxia Consulting à prendre des décisions stratégiques et à anticiper les tendances futures.

En résumé, bien que nous ayons rencontré des défis et des contraintes, ce projet représente une étape essentielle vers la réalisation de perspectives prometteuses pour Avaxia Consulting. Il souligne l'importance de l'analyse des données dans le secteur de la technologie de l'information et montre comment des améliorations continues peuvent renforcer la compétitivité et la réussite de l'entreprise. Notre engagement envers l'excellence et l'innovation nous encourage à explorer davantage ces perspectives pour offrir une valeur accrue à Avaxia Consulting.

Bibliographie

[1] *Presentation Avaxia*, https://www.avaxiagroup.com/fr/, [En ligne ;Accès le 2-sep-2023], 2023.

[2] *Services Avaxia*, https://www.avaxiagroup.com/fr/services, [En ligne ;Accès le 2-sep-2023], 2023.

[3] NUTCACHE, *backlog produit*, https://www.nutcache.com/fr/blog/quest-ce-quun-backlog-scrum/, [En ligne ;Accès le 2-sep-2023], 2019.

[4] J. PAQUET, *difference backlog Kanban /backlog SCRUM*, https://blog.myagilepartner.fr/index.php/2019/10/16/backlog-kanban/, [En ligne ;Accès le 19-sep-2023], 2017.

[5] M. TOUHTOUH, *Architecture Logique des systémes*, https://www.softfluent.fr/blog/architecture-logicielle-pour-application, [En ligne ;Accès le 21-Mars-2024], 2023.

[6] D. ZHART, *Patrons de conception*, https://refactoring.guru/fr/design-patterns/what-is-pattern, [En ligne ;Accès le 21-Mars-2024], 2023.

[7] G. KHERIJI, *CQRS*, https://www.invivoo.com/le-pattern-cqrs/, [En ligne ;Accès le 21-Mars-2024], 2023.

[8] A. SCHEVTS, *Patron façade*, https://refactoring.guru/fr/design-patterns/facade, [En ligne ;Accès le 21-Mars-2024], 2023.

[9] A. SCHEVTS, *Patron observer*, https://refactoring.guru/fr/design-patterns/observer, [En ligne ;Accès le 21-Mars-2024], 2023.

[10] *Single Responsability Principal*, https://www.ionos.fr/digitalguide/sites-internet/developpement-web/quest-ce-que-le-factory-pattern/, [En ligne ;Accès le 21-Mars-2024], 2022.

[11] GEEKSFORGEEKS, *Data Access Object*, https://www.geeksforgeeks.org/data-access-object-pattern/, [En ligne ;Accès le 21-Mars-2024], 2024.

[12] REDHAT, *architecture physique*, https://www.redhat.com/fr/topics/cloud-native-apps/what-is-an-application-architecture, [En ligne ;Accès le 22-Mars-2024], 2023.

[13] LUCIDCHART, *Diagramme de package*, https://www.lucidchart.com/pages/fr/diagramme-package-uml, [En ligne ;Accès le 22-Mars-2024], 2024.

[14] LUCIDCHART, *Diagramme de package*, https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml, [En ligne ;Accès le 22-Mars-2024], 2024.

[15] IBM, *Diagramme d'activité'*, https://www.ibm.com/docs/fr/dmrt/9.5?topic=diagrams-activity, [En ligne ;Accès le 22-Mars-2024], 2021.

[16] *Talend*, https://www.talend.com/, [En ligne ;Accès le 19-sep-2023], 2023.

[17] *postgresql*, https://www.postgresql.org/, [En ligne ;Accès le 19-sep-2023], 2023.

[18] *streamlit*, https://streamlit.io/, [En ligne ;Accès le 19-sep-2023], 2023.

[19] *Jira REST API*, https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro, [En ligne ;Accès le 19-sep-2023], 2023.

[20] *Gitlab REST API*, https://docs.gitlab.com/ee/api/, [En ligne ;Accès le 19-sep-2023], 2023.

[21] *SonarQube*, https://docs.sonarsource.com/sonarqube/latest/web-api/, [En ligne ;Accès le 19-sep-2023], 2023.

[22] *numpy*, https://docs.gitlab.com/ee/api/, [En ligne ;Accès le 19-sep-2023], 2023.

[23] *pandas*, https://pandas.pydata.org/, [En ligne ;Accès le 19-sep-2023], 2023.

[24] O. VOGEL, *"Software Architecture : A Comprehensive Framework and Guide for Practitioners."* Springer, 2011, ISBN : 3642197353.

[25] L. BASS, *"Software Architecture in Practice"*. Addison-Wesley Professional, 2021, ISBN : 0136886094.

[26] E. GAMMA, *"Design Patterns : Elements of Reusable Object-Oriented Software"*. Addison-Wesley Professional, 1994, ISBN : 0201633612.

ملخّص

يوضح هذا التقرير مشروعاًا داخل شركة Avaxia Consulting مركزًا على تحليل أداء الفريق. يستكشف كل مرحلة من مراحل المشروع، بدءًا من استخراج البيانات إلى التحليل، مؤكدًا على أهمية PostgreSQL في العملية. على الرغم من القيود الزمنية، نجح المشروع في تقديم آفاق للتحسين، بما في ذلك لوحات عمل متقدمة وفرص لدمج البيانات الإضافية. يؤكد التقرير على الدور الحيوي لتحليل البيانات في قطاع تكنولوجيا المعلومات.

كلمات مفاتيح : **تحليل البيانات**، Python، PostgreSQL، **لوحات العمل، أداء الفريق**

Résumé

Ce rapport décrit un projet au sein d'Avaxia Consulting, axé sur l'analyse des performances de l'équipe. Il explore chaque phase du projet, de l'extraction des données à leur analyse, mettant en évidence l'importance de PostgreSQL dans le processus. Malgré des contraintes de temps, le projet a réussi à offrir des perspectives d'amélioration, notamment des tableaux de bord plus avancés et des possibilités d'intégration de données supplémentaires. Le rapport souligne le rôle essentiel de l'analyse des données dans le secteur de l'IT.

Mots clés : Analyse de données, Python, PostgreSQL,Snowflake,React Tableaux de bord, Indicateur de Performance d'équipe.

Abstract

This report outlines a project within Avaxia Consulting, focusing on team performance analysis. It delves into each phase of the project, from data extraction to analysis, emphasizing the significance of PostgreSQL in the process. Despite time constraints, the project managed to offer improvement perspectives, including more advanced dashboards and opportunities for additional data integration. The report underscores the critical role of data analysis in the IT sector.

Keywords : Data analysis, Python, PostgreSQL, Dashboards, Key performance Indicators.

منبلازير ـ تونس الهاتف : 013 904 71 الفاكس : البريد الالكتروني : contact@avaxia-group.com
A01, Mont-plaisir, AV Khaireddine PACHA CP 1073 **Tél** : 71 904 013 **Fax** : **Email** : contact@avaxia-group.com

2 نهج أبو الريحان الباروني 2080 أريانة الهاتف : 164 706 71 الفاكس : 698 706 71 البريد الالكتروني : isi@isim.rnu.tn
2, Abou Raihane Bayrouni 2080 l'Ariana **Tél** : 71 706 164 **Fax** : 71 706 698 **Email** : isi@isim.rnu.tn