UEMF
الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

EIDIA
École d'Ingénierie Digitale et d'Intelligence Artificielle

Rapport de Projet

StreamVision : Pipeline de Données pour Plateforme de Streaming

Étudiant :	Yassir Tagemouati
	Bakr El Asmi
	Adil Habib
	Ilyass Bennani
Filière :	Cycle Ingénieur – Big Data
Encadrant :	M. Amamou Ahmed
Année Universitaire :	2025 – 2026
Date :	Janvier 2026

Université Euro-Méditerranéenne de Fès (UEMF)
École d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

Table des matières

Liste des tableaux

Table des figures

Introduction Générale

Contexte

Dans l'ère du streaming numérique, les plateformes de contenu vidéo génèrent des volumes considérables de données utilisateur. L'analyse de ces données est cruciale pour comprendre le comportement des utilisateurs, optimiser les recommandations de contenu, et améliorer l'expérience globale. StreamVision est une plateforme de streaming fictive qui nécessite une infrastructure de données robuste pour supporter ses opérations analytiques.

Problématique

La gestion et l'analyse de données streaming en temps réel présentent plusieurs défis : l'intégration de sources multiples, le traitement de volumes importants, la garantie de la qualité des données, et la fourniture d'analyses en temps opportun aux équipes métier. Ce projet vise à concevoir et implémenter un pipeline de données complet de bout en bout.

Objectifs

Les objectifs principaux de ce projet sont :

- Concevoir une architecture de données scalable pour une plateforme de streaming.
- Implémenter un pipeline ETL/ELT automatisé.
- Créer un entrepôt de données analytique dans Snowflake.
- Développer des tableaux de bord interactifs dans Power BI.
- Automatiser l'orchestration avec Apache Airflow.

Structure du Rapport

Ce rapport est organisé en cinq parties principales :

1. Présentation de l'organisation
2. Architecture et conception technique
3. Implémentation du pipeline de données

4. Résultats et visualisations
5. Webographie

Partie 1 : Présentation de l'Organisation

1.1 Contexte Organisationnel

L'École d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA) de l'Université Euromed de Fès (UEMF) sert de cadre académique pour ce projet. L'EIDIA se consacre à la formation d'ingénieurs en technologies numériques avec une approche pratique et innovante.

1.2 Objectifs Pédagogiques

Ce projet s'inscrit dans le cadre de la formation en Big Data et a pour objectifs pédagogiques :

- Appliquer les concepts théoriques à un cas réel.
- Maîtriser les outils modernes de gestion de données.
- Développer des compétences en architecture de données.
- Acquérir de l'expérience en orchestration de workflows.

1.3 Infrastructure Technique

Le projet a été développé en utilisant l'infrastructure cloud moderne :

- AWS pour le stockage (S3) et l'orchestration
- Snowflake comme entrepôt de données cloud
- Docker pour la containerisation
- Outils open-source : Apache Airflow, dbt, PostgreSQL

Partie 2 : Architecture et Conception Technique

2.1 Introduction

Cette section présente l'architecture globale du système StreamVision et les choix techniques réalisés.

2.2 Architecture Globale

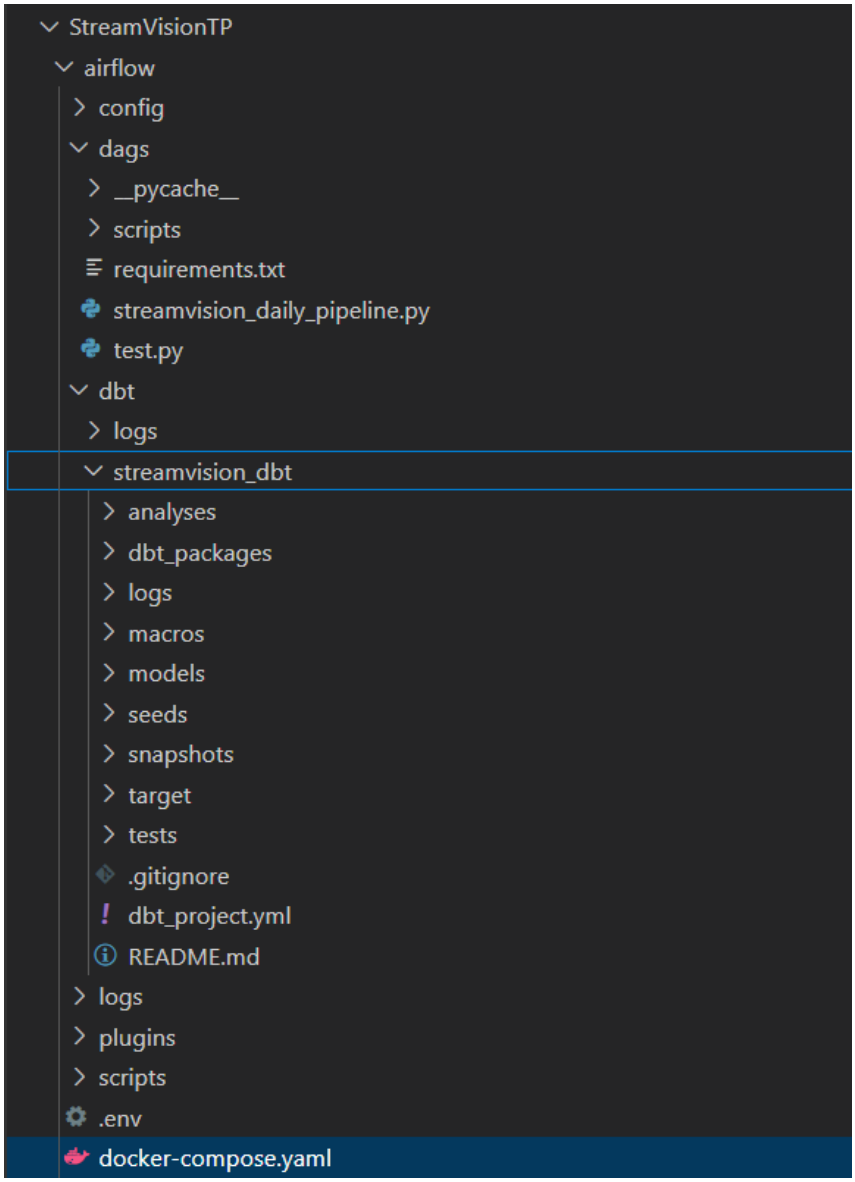


FIGURE 2.1 – Architecture complète du pipeline StreamVision

Le système suit une architecture en couches :

— **Couche Source :** PostgreSQL (données transactionnelles)

— **Couche Ingestion :** Amazon S3 (data lake)

— **Couche Transformation :** Snowflake + dbt
— **Couche Présentation :** Power BI (visualisation)
— **Couche Orchestration :** Apache Airflow

2.3 Stack Technologique

TABLE 2.1 – Stack technologique du projet StreamVision

Composant	**Technologie**
Base de données source	PostgreSQL 15
Stockage objet	Amazon S3
Entrepôt de données	Snowflake
Orchestration	Apache Airflow 2.7
Transformation	dbt (Data Build Tool)
Visualisation	Power BI
Conteneurisation	Docker
Langages	Python, SQL, YAML

2.4 Modèle de Données

Le modèle de données comprend plusieurs tables principales :
— **Utilisateurs :** informations démographiques et abonnements
— **Contenu :** films, séries, documentaires
— **Sessions :** historique de visionnage
— **Évaluations :** notes et commentaires
— **Abonnements :** événements d'abonnement
— **Recherches :** requêtes des utilisateurs

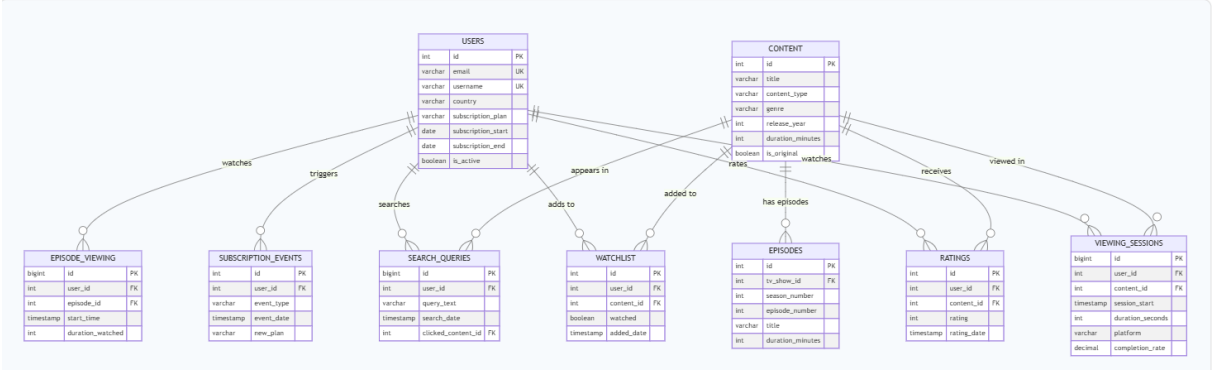


FIGURE 2.2 – Diagramme du modèle de données

2.5 Schéma en Étoile pour l'Analytique

Pour les besoins analytiques, un schéma en étoile a été conçu avec :

— Tables de faits : sessions, évaluations, événements

— Tables de dimensions : utilisateurs, contenu, temps, géographie

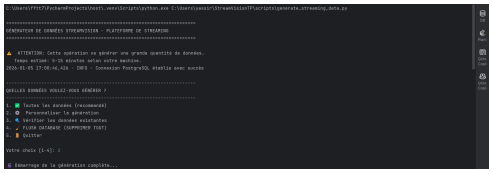
Partie 3 : Implémentation du Pipeline de Données

3.1 Introduction

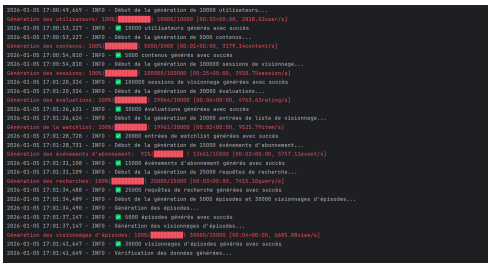
Cette section détaille l'implémentation technique de chaque composant du pipeline.

3.2 Génération des Données

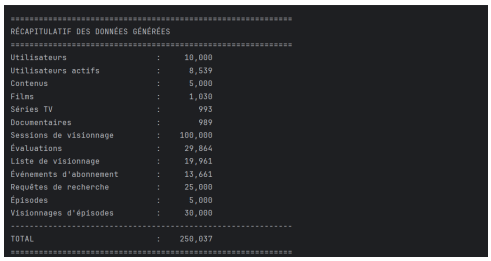
Un script Python a été développé pour générer des données réalistes :



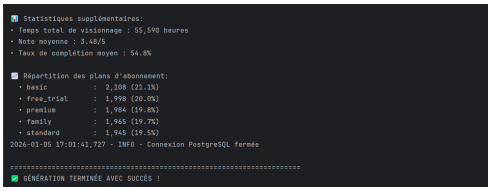
(a) 1



(b) 2



(c) 3



(d) 4

FIGURE 3.1 – Code de génération des données utilisateurs

Listing 3.1 – Extrait du script de génération de données

```
def generate_users(conn, n_users: int = 10000):
    """Génère des utilisateurs réalistes"""
    logger.info(f"Début de la génération de {n_users} utilisateurs...")
    # Code de génération...
```

3.3 Export vers Amazon S3

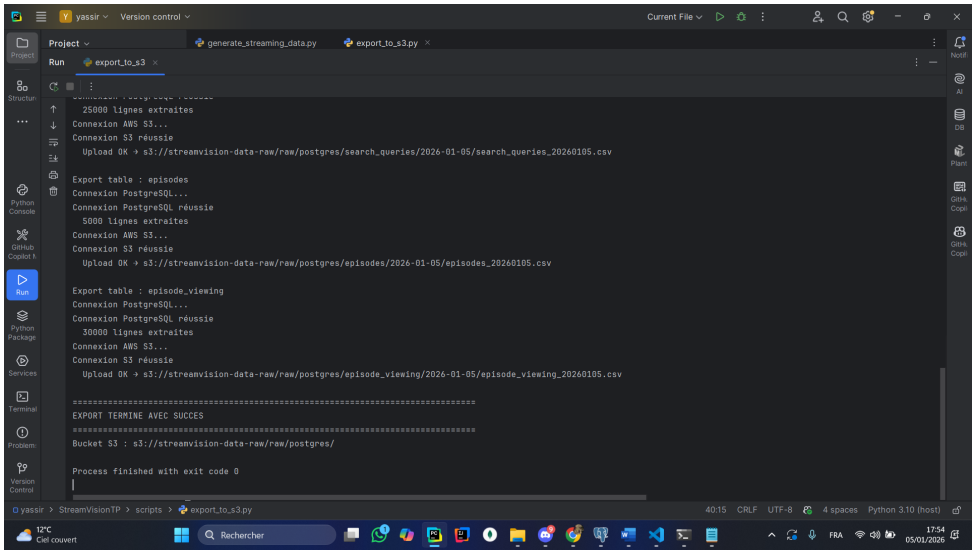


FIGURE 3.2 – Script d'export PostgreSQL vers S3

Le script exporte les données au format CSV avec partitionnement par date :

```
def export_table_to_s3(table_name, date_partition):
    """Exporte une table vers S3"""
    df = pd.read_sql(f"SELECT * FROM {table_name}", conn)
    s3_key = f"raw/postgres/{table_name}/{date_partition}/..."
    s3.put_object(Bucket=bucket, Key=s3_key, Body=csv_buffer.getvalue())
```

3.4 Configuration Snowflake

```
-- ============================================
-- Initialisation Snowflake - StreamVision
-- ============================================

USE DATABASE STREAMVISION_WH;

CREATE SCHEMA IF NOT EXISTS RAW;
CREATE SCHEMA IF NOT EXISTS STAGING;
CREATE SCHEMA IF NOT EXISTS CORE;
CREATE SCHEMA IF NOT EXISTS MARTS;

SHOW SCHEMAS;

SELECT 'Schémas StreamVision créés avec succès' AS message;
```

FIGURE 3.3 – Création des schémas Snowflake

Listing 3.2 – Configuration Snowflake

```
CREATE SCHEMA IF NOT EXISTS RAW;
CREATE SCHEMA IF NOT EXISTS STAGING;
CREATE SCHEMA IF NOT EXISTS CORE;
CREATE SCHEMA IF NOT EXISTS MARTS;
```

3.5 Chargement des Données dans Snowflake

FIGURE 3.4 – Script de chargement S3 vers staging

```
COPY INTO stg_users (
    id, email, username, first_name, last_name, ...
)
FROM @RAW.s3_raw_stage/postgres/users/2026-01-05/
FILE_FORMAT = (TYPE = CSV ...)
ON_ERROR = 'CONTINUE';
```

3.6 Transformations avec dbt

Trois modèles analytiques principaux ont été créés :

Listing 3.3 – Modèle de performance du contenu

```
{{ config(materialized='table', tags=['marts']) }}
WITH content_performance AS (
    SELECT
        c.content_id,
```

```
        c.title,
        COUNT(*) AS total_sessions,
        SUM(f.duration_seconds) AS total_watch_time,
        AVG(f.completion_rate) AS avg_completion_rate
    FROM {{ ref('stg_viewing_sessions') }} AS f
    JOIN {{ ref('stg_content') }} AS c
        ON f.content_id = c.content_id
    GROUP BY c.content_id, c.title
)
```

3.7 Orchestration avec Apache Airflow

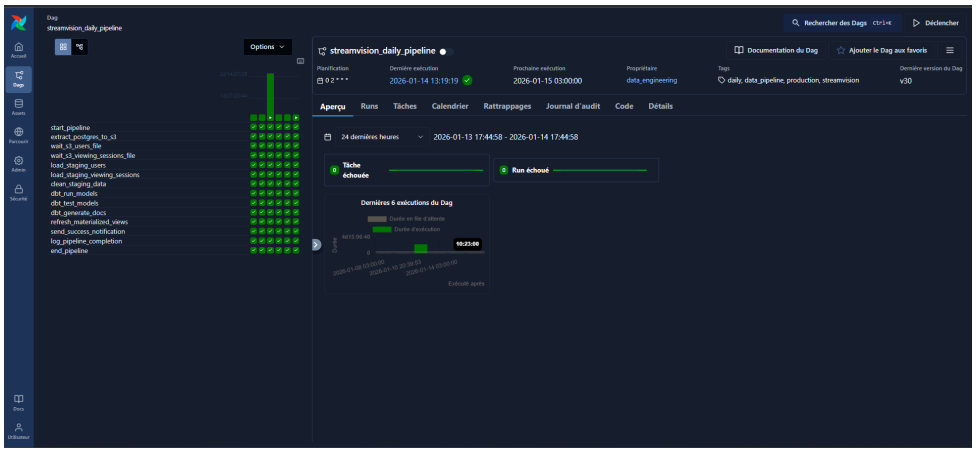


FIGURE 3.5 – DAG Airflow pour le pipeline StreamVision

Le DAG orchestre toutes les étapes :

— Extraction quotidienne PostgreSQL → S3
— Chargement S3 → Snowflake
— Transformations dbt
— Tests de qualité
— Notifications

3.8 Défis Techniques et Solutions

TABLE 3.1 – Défis techniques et solutions implémentées

Défi	**Solution**
Connexion Docker à PostgreSQL	Utilisation de `host.docker.internal`
Accès S3 depuis Airflow	Configuration des credentials AWS
Format de fichiers	Conversion de CSV.gz à CSV simple
Noms de colonnes	Alignement entre sources et destinations
Installation dbt	Installation dans l'environnement Docker
Intégration Snowflake-S3	Configuration des rôles IAM et policies

Partie 4 : Résultats et Visualisations

4.1 Introduction

Cette section présente les résultats concrets de la chaîne de traitement de données à travers deux tableaux de bord interactifs réalisés sous Power BI : la performance du contenu et l'analyse des abonnements.

4.2 Tableaux de Bord Power BI

4.2.1 Analyse de la Performance du Contenu

Le premier tableau de bord (Figure 4.1) se concentre sur la consommation média et la répartition géographique.

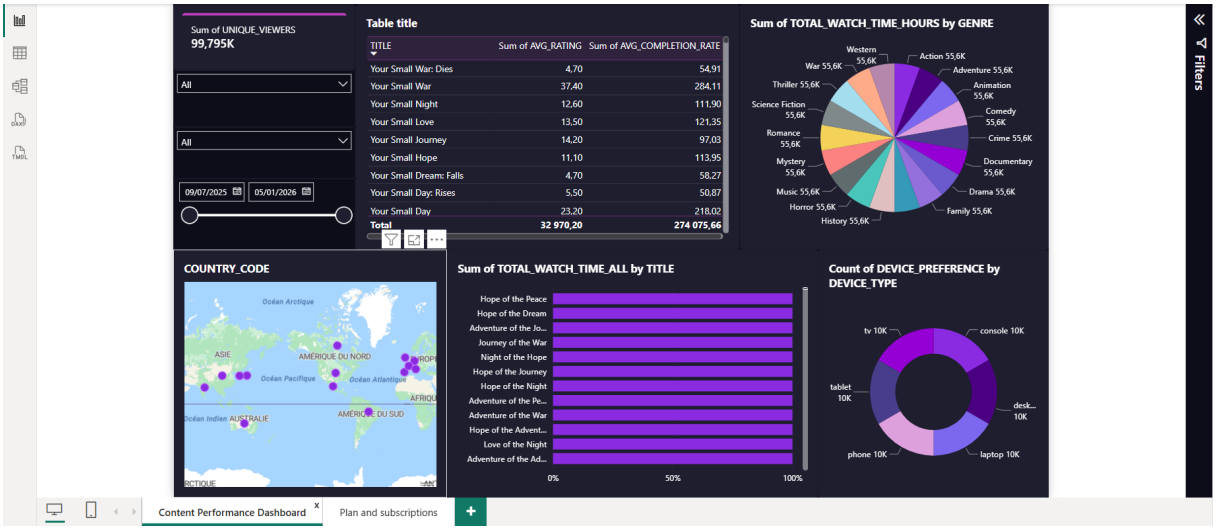


FIGURE 4.1 – Tableau de bord : Performance du contenu et visionnage

Les indicateurs clés visualisés sont :

- **Audience Globale :** Un total de 99 795 unique viewers.
- **Répartition Géographique :** Une carte interactive localisant les sessions de visionnage par pays.
- **Préférences de Lecture :** Analyse du temps de visionnage par genre et par titre, ainsi que la préférence par type de terminal (TV, mobile, desktop, etc.).
- **Qualité de Service :** Suivi des notes moyennes (AVG_RATING) et des taux de complétion par titre.

4.2.2 Analyse des Abonnements et Sessions

Le second volet (Figure 4.2) traite de la santé commerciale et technique de la plate-forme.

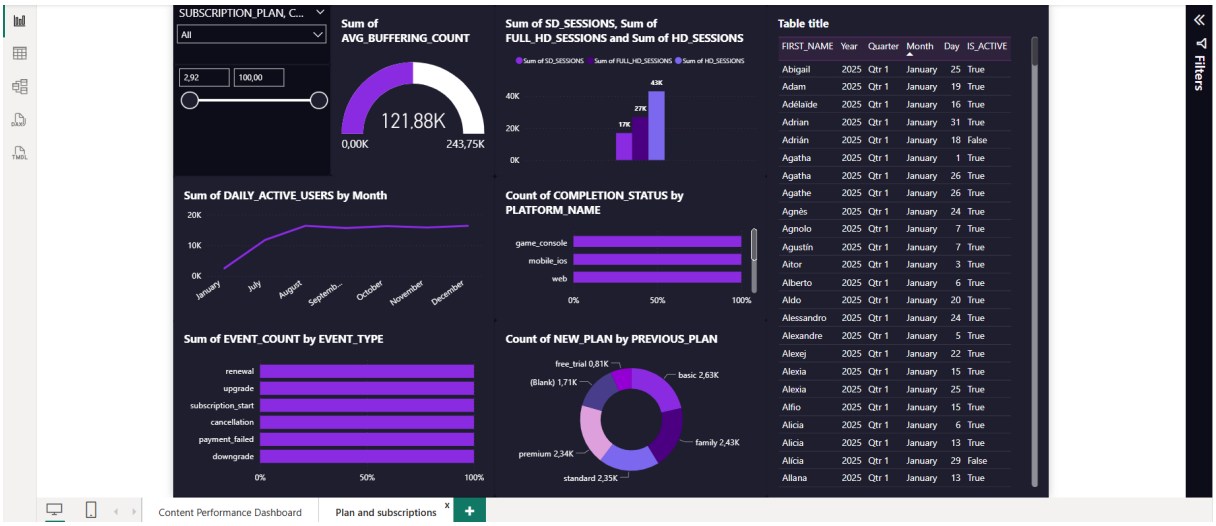


FIGURE 4.2 – Tableau de bord : Plans, abonnements et engagement technique

Cette vue permet d'analyser :

— **Gestion des Flux :** Un indicateur de buffering (121,88K) pour surveiller l'expérience utilisateur.

— **Dynamique des Abonnements :** Analyse des transitions entre plans (Upgrade, Downgrade, Renewal) via un graphique en anneau (Donut chart).

— **Engagement Quotidien :** Évolution des utilisateurs actifs quotidiens (DAU) sur l'année.

— **Qualité de Diffusion :** Répartition des sessions selon la résolution (SD, HD, Full HD).

4.3 Qualité des Données et Fiabilité

Pour garantir la précision de ces visuels, des tests **dbt** automatisés ont été implémentés en amont afin de vérifier :

— L'unicité des clés et l'absence de valeurs nulles sur les colonnes critiques (ID utilisateur, ID session).

— La cohérence des statuts d'activité (IS_ACTIVE) présentés dans les tables de détail.

— L'intégrité référentielle entre les événements de souscription et les types de plans.

Conclusion Générale

Le projet StreamVision a permis de concevoir et implémenter un pipeline de données complet pour une plateforme de streaming moderne. Les objectifs initiaux ont été atteints avec succès :

4.4 Réalisations

- Architecture de données scalable et maintenable
- Pipeline ETL/ELT entièrement automatisé
- Entrepôt de données analytique dans Snowflake
- Tableaux de bord interactifs dans Power BI
- Orchestration robuste avec Apache Airflow
- Tests de qualité automatisés avec dbt

4.5 Apprentissages

Ce projet a permis d'acquérir des compétences précieuses :

- Maîtrise des outils cloud modernes (AWS, Snowflake)
- Expérience en orchestration de workflows complexes
- Conception de modèles de données analytiques
- Gestion de la qualité des données
- Développement de visualisations métier

4.6 Perspectives

Le pipeline peut être étendu avec :

- Intégration de données en temps réel
- Machine Learning pour les recommandations
- Alertes et monitoring avancés
- Gouvernance des données
- Multi-cloud architecture

Ce projet démontre l'importance d'une infrastructure de données robuste pour les plateformes digitales modernes et prépare l'étudiant aux défis techniques du domaine du Big Data en entreprise.