# Rapport de Projet – ATL-Datamart Version Python

EPSI Paris & Arras – I1 EISI, IA Big Data et Cybersécurité

Benjamin AMAND, Augustin DUFETELLE, Guillaume MARECHAL, Toni PIRA

Table des matières

[Rapport de Projet – ATL-Datamart Version Python 1](#_Toc199101398)

[1. Introduction 1](#_Toc199101399)

[2. TP1 – Collecte des données vers un Datalake 1](#_Toc199101400)

[3. TP2 – Du Datalake vers le Data Warehouse 1](#_Toc199101401)

[4. TP3 – Création du Datamart et modélisation multidimensionnelle 2](#_Toc199101402)

[5. TP4 – Visualisation des données 2](#_Toc199101403)

[6. TP5 – Orchestration avec Airflow 3](#_Toc199101404)

[7. Conclusion 3](#_Toc199101405)

## 1. Introduction

Ce rapport présente les étapes réalisées dans le cadre du projet ATL-Datamart, dont l’objectif est de mettre en œuvre une architecture décisionnelle complète, de la collecte de données brutes jusqu’à leur visualisation, en passant par un pipeline de transformation et d’intégration.

## 2. TP1 – Collecte des données vers un Datalake

Objectif :

Télécharger un fichier parquet mensuel des taxis jaunes de New York et l’envoyer vers un Datalake simulé par Minio.

Étapes réalisées :

- Téléchargement du fichier parquet depuis le site officiel de l’État de New York.

- Stockage local dans data/raw/.

- Utilisation du script src/data/grab\_parquet.py pour uploader vers le bucket Minio.

Fichiers modifiés/utilisés :

- grab\_parquet.py  
- Minio bucket

Une image contenant texte, capture d’écran, nombre, logiciel

Le contenu généré par l’IA peut être incorrect.  
Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

## 3. TP2 – Du Datalake vers le Data Warehouse

Objectif :

Intégrer les fichiers parquet depuis Minio vers une base de données PostgreSQL servant de Data Warehouse.

Étapes réalisées :

- Modification du script dump\_to\_sql.py pour se connecter à Minio et charger les fichiers.

- Ingestion des données dans PostgreSQL avec les bons schémas.

Fichiers modifiés/utilisés :

- src/data/dump\_to\_sql.py  
- Configuration PostgreSQL via Docker

Une image contenant texte, logiciel, nombre, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## 4. TP3 – Création du Datamart et modélisation multidimensionnelle

Objectif :

Transformer la structure des données en modèle flocon (ou étoile) et alimenter un Datamart indépendant.

Étapes réalisées :

- Choix du modèle en flocon justifié dans le rapport.

- Création des tables avec creation.sql.

- Insertion des données avec insertion.sql.

Contraintes respectées :

- Deux serveurs SGBD distincts (pas deux bases dans un seul serveur)

Fichiers livrés :

- creation.sql  
- insertion.sql

Code SQL des scriptsUne image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

## 5. TP4 – Visualisation des données

Objectif :

Réaliser une analyse exploratoire ou une visualisation avec un outil connecté au Datamart.

Étapes réalisées :

- Création d’un tableau de bord avec Streamlit/PowerBI/Tableau.

- Connexion au Datamart.

Une image contenant capture d’écran, diagramme, texte

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, diagramme

Le contenu généré par l’IA peut être incorrect.

Une image contenant capture d’écran, Logiciel multimédia, Logiciel de graphisme

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Caractère coloré

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Une image contenant capture d’écran, Caractère coloré, texte, Graphique

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

## 6. TP5 – Orchestration avec Airflow

Objectif :

Automatiser les tâches de collecte et de traitement des données avec Apache Airflow.

Étapes réalisées :

- Complétion du fichier airflow/dags/minio.py pour télécharger et uploader automatiquement.

- Automatisation complète des TP2 et TP3 dans d'autres dags.

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

## 7. Conclusion

Ce projet nous a permis de mettre en œuvre une chaîne complète d’architecture Big Data avec un Datalake, un Data Warehouse, un Datamart et un outil de visualisation. L’orchestration des pipelines avec Airflow constitue un point de départ solide pour des projets plus avancés.