Rapport du Projet Talend : Pipeline de Données Global Flights

Imad Elmanser

January 5, 2025

Contents

1 Contexte du Projet **2**

2 Objectifs du Projet **2**

3 Étapes de Réalisation **2**
3.1 Configuration de Base et Paramétrage de l'Environnement 2
3.1.1 Création du Projet dans Talend Studio 2
3.1.2 Variables de Contexte 2
3.2 Nettoyage et Contrôles Qualité 3
3.2.1 Job 1 : Ingestion des Fichiers 3
3.2.2 Job 2 : Nettoyage des Données 3
3.2.3 Validation des Données 3
3.3 Transformation et Jointure des Données 3
3.3.1 Job 3 : Transformation et Jointure 3
3.4 Variables de Contexte et Paramétrisation 3

4 Chargement Final des Données **3**
4.1 Job 4 : Chargement dans la Base de Données 3

5 Résultats **4**

6 Conclusion **4**

1 Contexte du Projet

Dans le cadre de votre mission au sein d'une startup spécialisée dans l'analytique du secteur du voyage, vous avez été chargé de développer un pipeline de données performant et robuste pour l'analyse des routes aériennes mondiales. Le projet repose sur les jeux de données publics OpenFlights :

- **airports.dat** : contenant des informations sur les aéroports (nom, pays, coordonnées géographiques).
- **routes.dat** : contenant les routes reliant différents aéroports.

L'objectif est d'ingérer, nettoyer, transformer ces données, puis de les charger dans une base de données ou un fichier pour analyse.

2 Objectifs du Projet

Les principaux objectifs du projet sont les suivants :

1. Ingestion des fichiers CSV depuis une source externe.
2. Nettoyage et transformation des données.
3. Jointure et enrichissement des données.
4. Chargement final des données dans une base de données.
5. Mise en place de variables de contexte facilitant la gestion des environnements (DEV, TEST, PROD).

3 Étapes de Réalisation

3.1 Configuration de Base et Paramétrage de l'Environnement

3.1.1 Création du Projet dans Talend Studio

Un nouveau projet intitulé **GlobalFlightsPipeline** a été créé dans Talend Studio.

3.1.2 Variables de Contexte

Les variables de contexte suivantes ont été définies :

- **context.inputDirectory** : Emplacement des fichiers d'entrée.
- **context.outputDirectory** : Emplacement des fichiers de sortie.
- **context.env** : Nom de l'environnement (DEV, TEST, PROD).

3.2 Nettoyage et Contrôles Qualité

3.2.1 Job 1 : Ingestion des Fichiers

Les composants `tFileInputDelimited` et `tLogRow` ont été utilisés pour ingérer et vérifier les fichiers **airports.dat** et **routes.dat**.

3.2.2 Job 2 : Nettoyage des Données

1. Lecture du fichier **airports.dat**.
2. Filtrage des lignes avec des coordonnées manquantes ou invalides via `tFilterRow`.
3. Suppression des espaces inutiles dans les champs texte.
4. Écriture des données nettoyées dans $\mathbf{Cleaned}_{A}irports.csv$.

3.2.3 Validation des Données

Les lignes sans code IATA ont été redirigées vers un fichier de rejet $\mathbf{Rejected}_{R}ows.csv$.

3.3 Transformation et Jointure des Données

3.3.1 Job 3 : Transformation et Jointure

Les étapes suivantes ont été réalisées :

- **Lecture des fichiers** $\mathbf{Cleaned}_{A}irports.csv$**et routes.dat**.$Jointuredes fichiersàl'aideducomposa$
- **Création de nouvelles colonnes combinant les informations des aéroports source et destination.**

- **Écriture du résultat final dans** $\mathbf{Joined}_{F}lights.csv$.

3.4 Variables de Contexte et Paramétrisation

Les variables suivantes ont été ajoutées pour la gestion des bases de données :

- **context.db**$_{h}ost$: $Adressedel'hôte$.**context.db**$_{p}ort$: $Portdelabasededonnées$.**context.db**$_{n}ame$: Nom
  Trois environnements de contexte ont été définis : DEV, TEST et PROD.

4 Chargement Final des Données

4.1 Job 4 : Chargement dans la Base de Données

1. **Connexion à la base de données via `tPostgresqlConnection`.**
2. **Lecture du fichier** $\mathbf{Joined}_{F}lights.csv.Chargementdesdonnéesdanslatable$**flights**$_{e}nrichedvia$`tDBOutp`
   En cas d'erreur, un fichier de log est généré et le job est arrêté via `tDie`.

5 Résultats

Les livrables du projet sont :

- **Fichiers de sortie : Cleaned**$_{A}irports.csv, Rejected_{R}ows.csv, Joined_{F}lights.csv$**.Jobs Talend :** $Fichiers.itemou.zippermettantderecréerlepipeline.$
- **Rapport : Un rapport PDF détaillant le pipeline.**
- **Script SQL : Création et chargement de la table flights**$_{e}nriched.$

6 Conclusion

Ce projet a permis de concevoir un pipeline de données performant et modulaire en utilisant Talend. Chaque étape, de l'ingestion au chargement final, a été réalisée avec succès. La gestion des environnements via les variables de contexte garantit une portabilité et une flexibilité accrues. Ce pipeline peut être étendu pour intégrer de nouvelles sources de données et répondre à des besoins analytiques plus complexes.