# Rapport sur la configuration des variables de contexte et la structure du pipeline

## 1. Configuration des Variables de Contexte

**Les variables de contexte permettent de rendre le pipeline flexible et adaptable à différents environnements (DEV, TEST, PROD). Voici les principales variables configurées :**

- **Chemins de fichiers :**
  - **context.directoryDestination : Chemin pour stocker les fichiers .**
- **Environnement :**
  - **DEV : Environnement de développement.**
  - **TEST : Environnement de test.**
  - **PROD : Environnement de production.**
- **Base de données :**
  - **context.host : Adresse de l'hôte de la base de données.**
  - **context.port : Port de connexion à la base de données.**
  - **context.database : Nom de la base de données cible.**
  - **context.username : Utilisateur de la base de données.**
  - **context.password : Mot de passe de l'utilisateur.**
  - **context.tableName : Nom de la table cible.**

## 2. Structure du Pipeline

**Le pipeline est divisé en plusieurs jobs Talend pour une exécution modulaire et organisée:**

### Job 1 : IngestFiles

- **Rôle : Ingérer les fichiers à partir de GitHub.**
- **Composants:**
  - **tFileFetch : Télécharger les fichiers.**
  - **tFileInputDelimited : Lire les fichiers.**
  - **tLogRow : Afficher les données ingérées.**

### Job 2 : CleanData

- **Rôle : Nettoyer les fichiers aéroports.**
- **Composants:**
  - **tFileInputDelimited : Lire les données.**
  - **tFilterRow : Filtrer les lignes invalides (par exemple, coordonnées).**
  - **tMap : Supprimer les espaces inutiles.**
  - **tFileOutputDelimited : Enregistrer les données nettoyées dans 'Cleaned_Airports.dat' et les lignes rejetées dans `Rejected_Rows.dat'.**

### Job 3 : TransformAndJoin

- **Rôle : Transformer et enrichir les données.**
- **Composants:**
  - **tMap : Associer les aéroports source et destination avec leurs détails (ville, pays).**
  - **tFileOutputDelimited : Enregistrer les données enrichies dans Joined_Flights.csv.**

### Job 4 : LoadIntoDB

- **Rôle : Charger les données dans la base de données.**
- **Composants:**
  - **tDBConnection : Se connecter à la base de données.**
  - **tCreateTable : Créer la table cible flights_enriched si elle n'existe pas.**
  - **tDBOutput : Insérer les données dans la table.**
  - **Gestion des erreurs : Utiliser tLogCatcher pour enregistrer les erreurs dans un fichier log.**

### Job 5 : AutomatePipeline

- **Rôle : Automatiser l'exécution des jobs.**
- **Composants:**
  - **tRunJob: Appeler les jobs dans l'ordre suivant :**
    - **IngestFiles**
    - **CleanData**
    - **TransformAndJoin**
    - **LoadIntoDB.**

# 3. Conclusion

**Ce pipeline est conçu pour être modulaire, adaptatif et robuste. L'utilisation des variables de contexte permet une configuration rapide selon les besoins de l'environnement. Les différentes étapes garantissent un nettoyage, une transformation et un chargement efficace des données aériennes mondiales.**