**README - Projet Football Premier League ⚽️📊**

Ce projet complet a pour but de modéliser, automatiser et analyser un championnat de football (Premier League anglaise) en utilisant Snowflake, Python (Snowpark) et Power BI.

**🎯 Objectifs pédagogiques et techniques**

* Modéliser une base relationnelle complète pour un championnat de football
* Nettoyer, transformer et charger les données réelles et bruitées avec Python
* Créer des procédures automatiques pour la gestion de résultats et classements
* Gérer les rôles et la sécurité des utilisateurs avec Snowflake
* Construire des dashboards interactifs avec Power BI

**📁 Arborescence du projet**

Projet\_Football\_Snowflake/

│

├── sql/

│ ├── create\_tables.sql --> Création des 10+ tables relationnelles

│ ├── insert\_data.sql --> Insertion initiale (équipes, joueurs, stades...)

│ ├── procedures.sql --> Toutes les procédures stockées (résultats, classement...)

│ ├── roles.sql --> Définition et attribution des rôles utilisateurs

│ └── tasks.sql --> Automatisations planifiées avec Snowflake Task

│

├── python/

│ └── main\_elt.py --> Script Snowpark pour nettoyage CSV (joueurs)

│

├── data/

│ ├── joueurs\_dirty.csv --> Fichier source avec doublons, erreurs, champs vides

│ ├── erreurs\_nulls.csv --> Lignes avec valeurs manquantes

│ ├── erreurs\_doublons.csv --> Lignes dupliquées

│ └── erreurs\_equipes.csv --> Joueurs avec `equipe\_id` invalide

│

├── docs/

│ └── documentation.pdf --> Description complète et technique du projet

│

└── README.md

**🧱 Base de données Snowflake (schéma premier\_league)**

**Tables principales créées :**

* federation
* stade
* equipe (+ colonne président)
* joueur (20+ joueurs réels/érronés par club)
* arbitre
* calendrier (matchs programmés)
* match (résultats)
* but (buteurs)
* statistiques\_match (fautes, tirs, possession)
* classement (points, victoires, etc.)

**🔄 Traitement ELT avec Snowpark (Python)**

**Script main\_elt.py :**

* Chargement d’un fichier joueurs\_dirty.csv
* Détection et export des erreurs :
  + Champs null
  + Doublons
  + equipe\_id invalide
* Sauvegarde des erreurs dans des fichiers séparés pour audit

**🔧 Procédures Stockées**

**inserer\_resultat\_auto()**

* Langage : JavaScript
* But : insère automatiquement un résultat de match toutes les 2 minutes
* Liée à une TASK Snowflake planifiée (SCHEDULE = '2 MINUTE')

**mettre\_a\_jour\_classement()**

* Langage : SQL
* Supprime et reconstruit le classement selon les résultats actuels
* Points, victoires, défaites, buts, etc.

**🔐 Sécurité & gestion des rôles Snowflake**

**Rôles créés :**

* role\_arbitre → peut insérer résultats & mettre à jour classement
* role\_federation → gère calendrier, stats, classement
* role\_president → peut insérer, modifier, supprimer des équipes
* role\_entraineur et role\_joueur → lecture seule sur toutes les tables

**Autres actions :**

* Création de **comptes utilisateurs** Snowflake (ex: arbitre1, federation1) avec mot de passe
* Connexion testée avec chaque rôle
* Droits accordés via GRANT USAGE, GRANT INSERT/UPDATE, etc.

**📊 Power BI (à faire)**

Tableaux de bord prévus :

* Classement dynamique des équipes (par saison ou journée)
* Top buteurs
* Fiches clubs/joueurs
* Statistiques de match (tirs, fautes, possession)

**🔮 Fonctionnalités futures**

* 🔁 **Automatisation du classement** avec une tâche planifiée liée à mettre\_a\_jour\_classement()
* 🔍 **Prédiction des résultats** (XGBoost, LogReg...) basée sur les saisons précédentes
* 📈 Dashboards Power BI enrichis avec les prédictions

**✅ Conclusion**

Projet professionnel complet combinant :

* Snowflake (SQL, JS, rôles, automatisation)
* Python Snowpark pour ELT et qualité des données
* Dashboards (Power BI)
* Sécurité avancée (multi-utilisateurs, GRANTs)

À venir : intégration des prédictions + visualisations finales sur Power BI