Projet : Mini-CRM pour la gestion des utilisateurs d'une startup

Contexte :
Une startup souhaite centraliser et nettoyer les données de ses utilisateurs pour mieux gérer ses campagnes marketing et ses newsletters. Les données initiales se trouvent dans un fichier CSV (avec des colonnes comme nom, email, âge, pays) mais elles contiennent des doublons, des valeurs manquantes et parfois des erreurs de saisie (par exemple, des adresses email mal formatées ou des âges incohérents).

Étapes du projet :

1. **Chargement et lecture du CSV**
    - Utiliser **pandas** pour charger le fichier CSV.
    - Mettre en place des options pour gérer des séparateurs personnalisés et définir des valeurs manquantes (ex. "NA", "n/a", etc.).
2. **Nettoyage et validation des données**
    - Supprimer les doublons et les lignes incomplètes avec `drop_duplicates()` et `dropna()`.
    - **Validation des emails :** Utiliser une expression régulière (regex) pour s'assurer que les adresses email sont bien formées.
    - **Vérification de l'âge :** S'assurer que l'âge est un entier réaliste (par exemple, entre 18 et 100 ans) et convertir le champ en type numérique.
    - **Gestion des données non standard :** Par exemple, remplacer les valeurs textuelles non numériques dans la colonne âge par une valeur par défaut ou marquer la ligne comme invalide.
3. **Stockage dans une base SQLite**
    - Créer une base SQLite (par exemple, `utilisateurs.db`) et une table `utilisateurs` avec des colonnes pour nom, email, âge, pays.
    - Définir des contraintes (comme un index unique sur l'email) pour éviter les doublons futurs.
    - Utiliser des transactions pour insérer les données de manière sécurisée et gérer les erreurs.
4. **Notions complémentaires :**
    - **Logging :** Mettre en place un système de log pour enregistrer les opérations de nettoyage et les éventuelles erreurs rencontrées lors du chargement ou de l'insertion des données.
    - **Interface en ligne de commande (CLI) :** Créer un petit menu interactif qui permet de :
        - Consulter la liste complète des utilisateurs.
        - Rechercher par nom ou par pays (ex. récupérer tous les utilisateurs français).
        - Exporter le résultat d'une requête dans un nouveau fichier CSV.
    - **Visualisation de données :** Utiliser **matplotlib** pour générer quelques graphiques simples (par exemple, la répartition des utilisateurs par pays ou la distribution des âges).
5. **Bonus (optionnel) : Création d'une API REST**

- Développer une petite API avec **Flask** qui expose des endpoints pour interroger la base (par exemple, un endpoint `/utilisateurs/francais` qui renvoie les utilisateurs français en JSON).