Projet fin SAS : DATA ANALYST (Jobinetech)

Contexte professionnel

Vous êtes Data Analyst junior dans une entreprise de vélos en libre-service à Londres. L'entreprise vous transmet un fichier brut contenant l'historique des trajets horaires, la météo, la saison et d'autres informations.

Mission :

1. Nettoyer et structurer les données
2. Migrer les données vers PostgreSQL
3. Formuler et exécuter des requêtes SQL significatives
4. Créer un dashboard interactif pour analyser les tendances des trajets

Ce projet simule une mission réelle de Data Analyst en entreprise.

Données fournies

- Fichier CSV brut : london_merged.csv
- Colonnes principales :

Colonne	Contenu
Timestamp :	Date et heure du relevé
Cnt :	Nombre de trajets
t1 / t2 :	Température réelle et ressentie
Hum :	Humidité
wind_speed :	Vitesse du vent
weather_code :	Code météo
is_holiday /is_weekend :	Jour férié ou week-end
Season :	Code de la saison

Objectifs pédagogiques

À la fin du projet, vous serez capable de :

- Lire et comprendre un dataset réel
- Identifier et corriger les problèmes de qualité (doublons, valeurs manquantes, codes numériques)
- Renommer et standardiser les colonnes
- Transformer les codes en valeurs lisibles (saisons, météo)
- Exporter un dataset clean exploitable pour la visualisation
- Migrer un dataset vers PostgreSQL en respectant la sécurité (.env)
- Créer des visualisations interactives et significatives dans Power BI ou Tableau
- Formuler des requêtes SQL réalistes pour analyser le dataset

---

Travail demandé – Étapes détaillées

1. Nettoyage et structuration avec Python / Pandas

1. Charger le CSV brut avec Pandas (london_merged.csv).
2. Observer le dataset :
    - Nombre de lignes et colonnes
    - Types de variables
    - Valeurs manquantes
    - Doublons
3. Renommer les colonnes pour les rendre compréhensibles :
    - cnt → Nombre de trajets
    - t1 → Température réelle (°C)
    - t2 → Température ressentie (°C)
    - hum → Humidité
    - wind_speed → Vitesse du vent (km/h)
    - weather_code → Météo
    - season → Saison
4. Conversion et nettoyage :
    - Humidité → valeur entre 0 et 1

- Codes numériques → valeurs explicites :
  - Saisons : 0 → Printemps, 1 → Été, 2 → Automne, 3 → Hiver
  - Météo : 1 → Clair, 2 → Nuages épars, 3 → Nuages fragmentés, 4 → Couvert, 7 → Pluie, 10 → Pluie avec orage, 26 → Neige

5. Vérification de la cohérence des données.
6. Export du dataset clean (london_bikes_final.xlsx ou .csv).

💡 **Astuce** : Documenter chaque étape dans un notebook Jupyter avec Markdown pour expliquer vos choix.

---

2. Migration vers PostgreSQL

1. **Préparer le fichier clean** : vérifier que toutes les colonnes sont correctes et que les types de données sont compatibles.
2. **Créer un fichier .env** pour sécuriser vos informations de connexion :
3. DB_HOST=localhost
4. DB_PORT=5432
5. DB_NAME=nom_base
6. DB_USER=utilisateur
7. DB_PASSWORD=motdepasse
   - Ne jamais mettre vos identifiants directement dans le script.
8. **Charger les variables d'environnement** dans Python avec dotenv et os.getenv.
9. **Créer la connexion** avec SQLAlchemy :
   - Encoder le mot de passe si nécessaire (quote_plus())
   - Tester la connexion avec SELECT version();
10. **Créer la table et insérer les données** :
    - Choisir un nom clair pour la table (london_bikes_final)
    - Décider si if_exists doit être replace ou append
11. **Vérifier l'insertion** : SELECT COUNT(*) pour confirmer le nombre de lignes

💡 **Conseil pédagogique** : comprendre pourquoi chaque étape est importante : sécurité, types de données, validation.

---

3. Requêtes SQL significatives

Formuler vos propres requêtes pour analyser le dataset :

- Total de trajets par saison, météo, ou jour de la semaine
- Moyenne de trajets par heure ou par jour
- Heures ou jours avec le plus de trajets (Top N)
- Comparaison week-end vs semaine
- Autres questions pertinentes que vous pourriez poser

---

4. Visualisation et dashboard interactif (Power BI ou Tableau)

- Charger le dataset clean (london_bikes_final.xlsx).
- **Liberté totale** pour créer vos visuels :
    - Types de graphiques : ligne, barre, camembert, heatmap, scatter plot...
    - Variables à comparer : météo vs trajets, saison vs utilisation, heure vs trajets...
    - Ajouter des filtres, slicers ou paramètres pour rendre le dashboard interactif
- **Règles pédagogiques** :
    - Chaque graphique doit porter du sens et répondre à une question métier
    - Vérifier la lisibilité : titres explicites, axes clairs, couleurs cohérentes

💡 **Questions à explorer** (non exhaustives) :

- Quels jours ou semaines ont le plus de trajets ?
- Comment la météo influence-t-elle le nombre de trajets ?
- Quels sont les pics horaires d'utilisation ?
- Y a-t-il des tendances saisonnières ou mensuelles ?

Livrables attendus

1. **Dataset clean** : london_bikes_final.xlsx
2. **Table PostgreSQL** : avec vérification du nombre de lignes
3. **Dashboard interactif** : Power BI ou Tableau
4. **Notebook Jupyter documenté** :
    - Étapes de nettoyage
    - Transformation des codes
    - Logique de migration vers PostgreSQL
    - Explications des choix de visualisation
5. **Courte synthèse écrite (5–10 lignes)** :
    - Principaux problèmes rencontrés
    - Décisions de nettoyage
    - Limites possibles du dataset