

Analyser les ventes d'une PME



ÉTAPES DE NOTRE ANALYSE

01

CONTEXTE &
PROBLÉMATIQUE

02

OBJECTIFS DU PROJET

03

MÉTHODOLOGIE

04

VISUALISATION DES
DONNÉES

05

RECOMMANDATIONS &
CONCLUSION

06

MOTIVATION ET
PERSPECTIVES



Contexte & Problématique

Contexte : Projet réalisé dans le cadre de ma formation Data Engineer à Simplon

Brief : Une PME souhaite analyser la dynamique de ses ventes pour mieux piloter sa stratégie

Problématique : Comment structurer, stocker et analyser efficacement les données de ventes issues de plusieurs sources ?

Contraintes techniques : utilisation de Docker, SQLite, analyse en Python

Objectifs du projet

- ❖ Mettre en place une architecture à deux services (script + base de données) avec Docker
- ❖ Collecter, transformer et stocker des données issues de fichiers CSV en ligne
- ❖ Réaliser une première analyse statistique des ventes
- ❖ Visualiser les résultats pour en faciliter l'interprétation
- ❖ Appliquer des compétences en SQL, Python, data visualisation

Méthodologie

❖ **Architecture Docker :**

- service pour exécution de scripts Python
- service SQLite pour stocker les données

❖ **Scripts développés :**

- Téléchargement des données (requêtes HTTP)
- Nettoyage et transformation avec Pandas
- Import en base SQLite

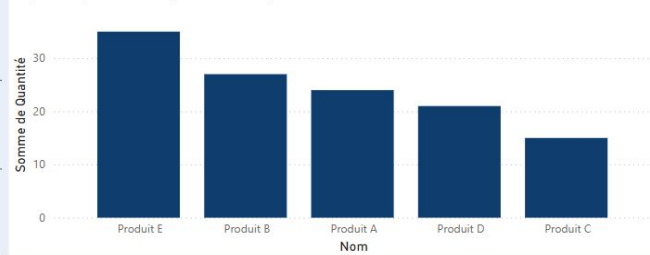
❖ **Analyse des ventes** via SQL + visualisation avec Matplotlib / Seaborn



VISUALISATION DES DONNÉES

Indicateurs clés de performance – Ventes & CA

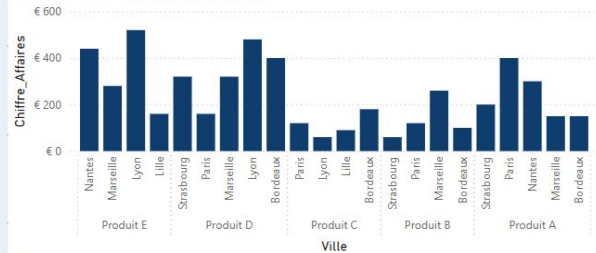
Top 5 des produits les plus vendus (quantité)



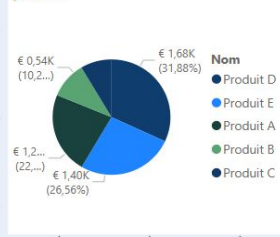
Répartition du chiffre d'affaires par ville



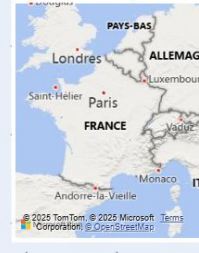
Chiffre d'affaires par produit et par ville



Répartition du chiffre d'affaires par produit



Chiffre_Affaires par Ville



Dashboard KPI- Power BI

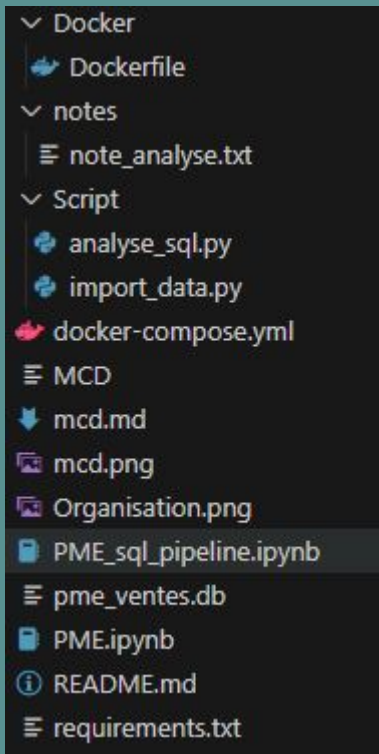
Produit E : le plus vendu, fort potentiel commercial

Lyon & Marseille : villes au chiffre d'affaires le plus élevé

Produit D : plus gros CA global, malgré une quantité vendue modérée

Lille : sous-performance à explorer

SCHÉMA DE L'ARCHITECTURE



📁 Docker/

- Contient le **Dockerfile** utilisé pour construire l'image du service Python

📁 notes/

- `note_analyse.txt` : Synthèse des résultats d'analyse

📁 Script/

- `import_data.py` : Script de **collecte et import des données** depuis les fichiers CSV
- `analyse_sql.py` : Script d'**analyse SQL automatisée**
- `docker-compose.yml` : Fichier pour **orchestration des services** (scripts + base SQLite)

📁 MCD/

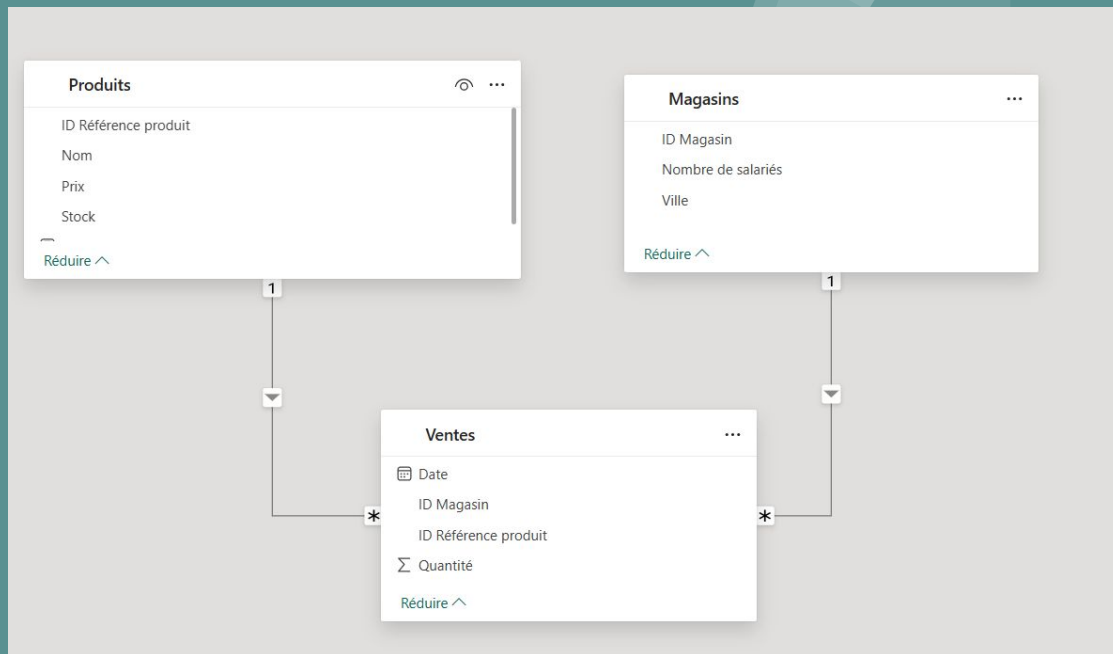
- `mcd.png` et `mcd.md` : **Modèle conceptuel des données** (structure des tables et relations)

📁 Autres fichiers importants :

- `Organisation.png` : Schéma de l'**architecture Docker**
- `PME_sql_pipeline.ipynb` : **Notebook Jupyter** d'analyse complète
- `pme_ventes.db` : **Base de données SQLite**
- `README.md` : Guide de prise en main du projet
- `requirements.txt` : Liste des **dépendances Python**

Cette organisation claire m'a permis de **travailler efficacement**, en séparant les étapes : développement, data analyse, architecture et documentation

SCHÉMA DES DONNÉES



COMPÉTENCES MOBILISÉES

Python : manipulation de données avec Pandas, visualisation

SQL : requêtes d'agrégation et jointures

Docker : création de Dockerfile, orchestration avec Docker Compose

Base de données : création et gestion de schéma sous SQLite

Méthodologie projet : organisation, autonomie, résilience face aux blocages

RECOMMANDATIONS & CONCLUSION

Recommandations stratégiques :

1. **Renforcer les stocks** des produits à fort CA (D & E) dans les villes clés
2. **Concentrer les campagnes marketing** sur les zones à fort potentiel (Lyon, Marseille)
3. **Analyser les faibles performances** à Lille pour ajuster la stratégie locale
4. Adapter l'**approvisionnement des produits peu vendus** pour éviter le surstock

MOTIVATION ET PERSPECTIVES

Ce projet m'a permis :

- d'appliquer mes compétences en **analyse de données**, **Python**, **SQL**, **data viz**
- d'identifier mes **axes de progression** (notamment Docker)
- de confirmer ma motivation à devenir **Data Engineer confirmé**

© Je souhaite vivement intégrer votre programme pour franchir cette étape.