



Analyser les ventes d'une PME

Projet Data Engineer – Simplon
Anissa LAHBIB – Administratrice Système

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

Mission: 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
- ❖ Alimenter une base de données SQLite
- ❖ Réaliser une première analyse statistique des ventes SQL
- ❖ Visualiser les résultats pour en faciliter l'interprétation
- ❖ Stocker les résultats d'analyse

Méthodologie

❖ Architecture Docker :

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

❖ Scripts développés Outils utilisés :

- Dockerfile & docker-compose
- Python (Pandas, SQLite, Matplotlib)
- Power BI pour la data viz

❖

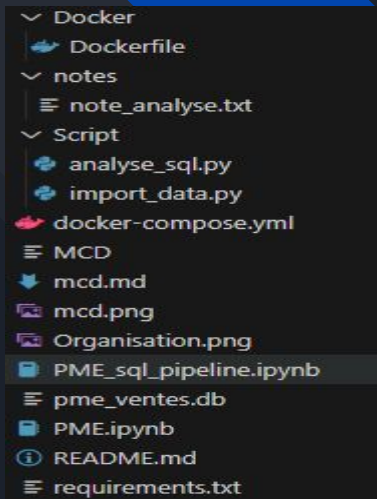
- Téléchargement des données (requêtes HTTP)
- Nettoyage et transformation avec Pandas
- Import en base SQLite

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

⚙ Outils utilisés :


- Dockerfile & docker-compose
- Python (Pandas, SQLite, Matplotlib)
- Power BI pour la data viz

Architecture technique




2 Services Docker

 Service 1 : Python

 Objectif : exécuter les scripts d'import, transformation et analyse des données


 Contenu :

- PME_sql_pipeline.ipynb : Notebook principal pour orchestrer le projet
- import_data.py : script de collecte et insertion des données
- analyse_sql.py : script de requêtes SQL automatisées


 Environnement basé sur une image Docker Python

 Communique avec la base SQLite pour lire/écrire les données


 Service 2 : SQLite

 Objectif : stocker les données structurées dans une base locale

 Fichier principal : pme_ventes.db

 Contient trois tables :

- produits
- magasins
- ventes (liée aux deux précédentes par clés étrangères)

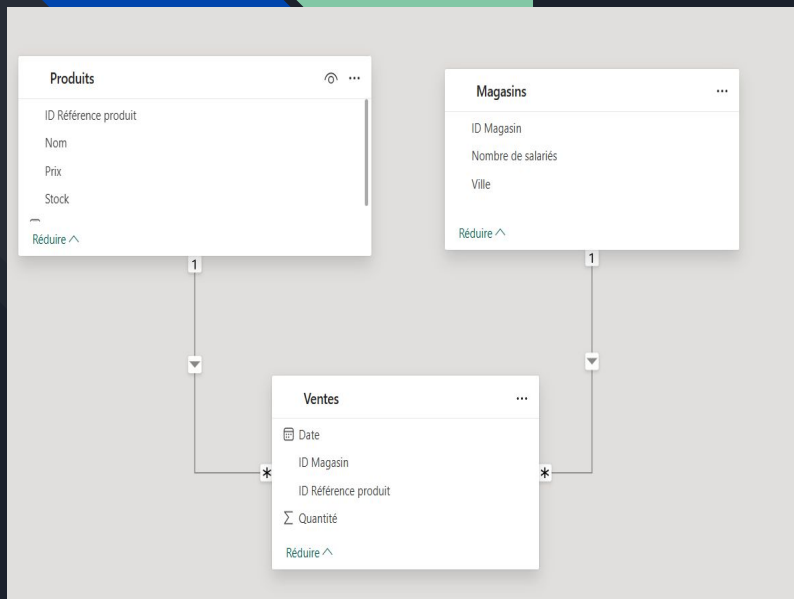
 Monté via Docker Compose, accessible depuis le conteneur Python

 Communication entre les services :

Orchestrée via docker-compose.yml

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

SCHÉMA DES DONNÉES



1. Produits

- **ID Référence produit** : Clé primaire (id_ref_produit)
- **Nom, Prix, Stock**

Cette table contient les informations sur chaque produit vendu par la PME

2. Magasins

- **ID Magasin** : Clé primaire (id_magasin)
- **Ville, Nombre de salariés**

3. Ventes

- **Date**
- **ID Magasin** → Clé étrangère liée à Magasins
- **ID Référence produit** → Clé étrangère liée à Produits
- **Quantité**

Chaque vente correspond à un produit et un magasin → double relation (de type N:1)

La table ventes fait le lien entre les produits et les magasins, et constitue une table de faits au cœur de l'analyse



COMPÉTENCES MOBILISÉES

Python : manipulation de données avec Pandas, visualisation

SQL : requêtes d'agrégation et jointures

Docker : création de Dockerfile, orchestration avec Docker Compose

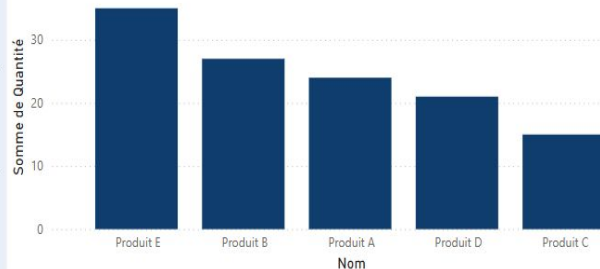
Base de données : création et gestion de schéma sous SQLite

Méthodologie projet : Organisation, curiosité, autonomie, esprit d'analyse

VISUALISATION DES DONNÉES

Indicateurs clés de performance – Ventes & CA

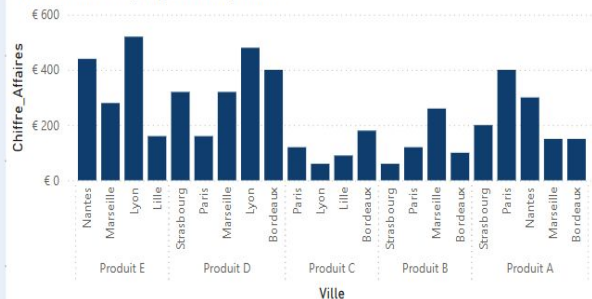
Top 5 des produits les plus vendus (quantité)



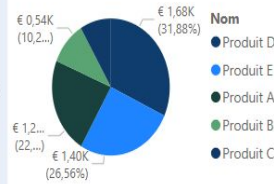
Répartition du chiffre d'affaires par ville



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

RECOMMANDATIONS & CONCLUSION

Recommandations stratégiques :

1. Renforcer les stocks des produits D & E
→ pour soutenir la demande croissante et maximiser les opportunités de vente.
2. Concentrer les efforts marketing sur Lyon et Marseille
→ afin de capitaliser sur les zones les plus rentables et améliorer le retour sur investissement.
3. Mener une analyse ciblée sur Lille
→ pour identifier les causes de sous-performance et ajuster l'offre commerciale locale.
4. Adapter l'approvisionnement des produits à faible rotation
→ pour éviter les surstocks et optimiser la gestion des stocks en entrepôt.

Ces recommandations s'appuient directement sur l'analyse des ventes réalisée via SQL et visualisée dans Power BI.

MOTIVATION ET PERSPECTIVES

POURQUOI JE VEUX DEVENIR DATA ENGINEER CHEZ SIMPLON

Ce projet m'a permis de :

- Mettre en pratique mes compétences en **analyse de données, Python, SQL et visualisation**
- Identifier mes **axes de progression**, notamment sur **Docker** et l'automatisation

🎯 **Objectif : devenir Data Engineer**

Rejoindre Simplon en alternance, c'est pour moi :

- Consolider mes acquis techniques
- Monter en compétence dans un environnement stimulant
- Contribuer à des projets à impact dans la vraie vie

🔗 **Pour aller plus loin :**

Vous pouvez consulter l'ensemble du projet sur mon GitHub :

🔗 <https://github.com/Anissa83/analyse-ventes-pme>