

BRIEF PROJET



Présentation analyse des ventes d'une PME

Sommaire

1)	Context, Objectifs.....	3
2)	Architecture et base de données.....	4
3)	Programmation.....	6
4)	Visualisation & Résultats.....	9
5)	Conclusion.....	11

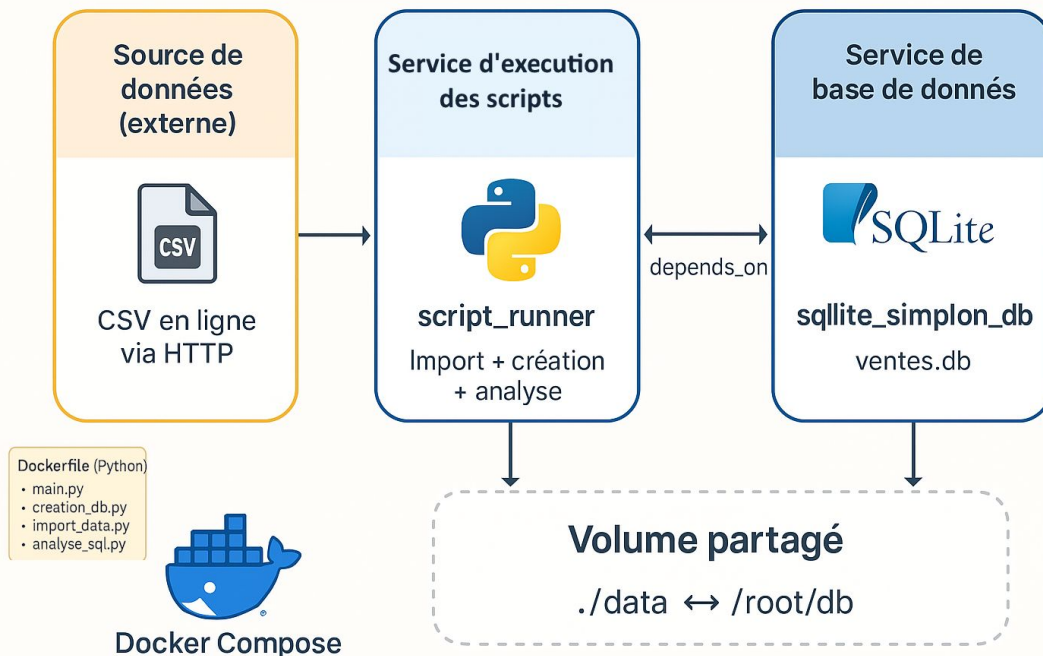
1) Context

Objectifs visés: en tant que admin système, assistez un data engineer pour permettre l'analyse de ventes

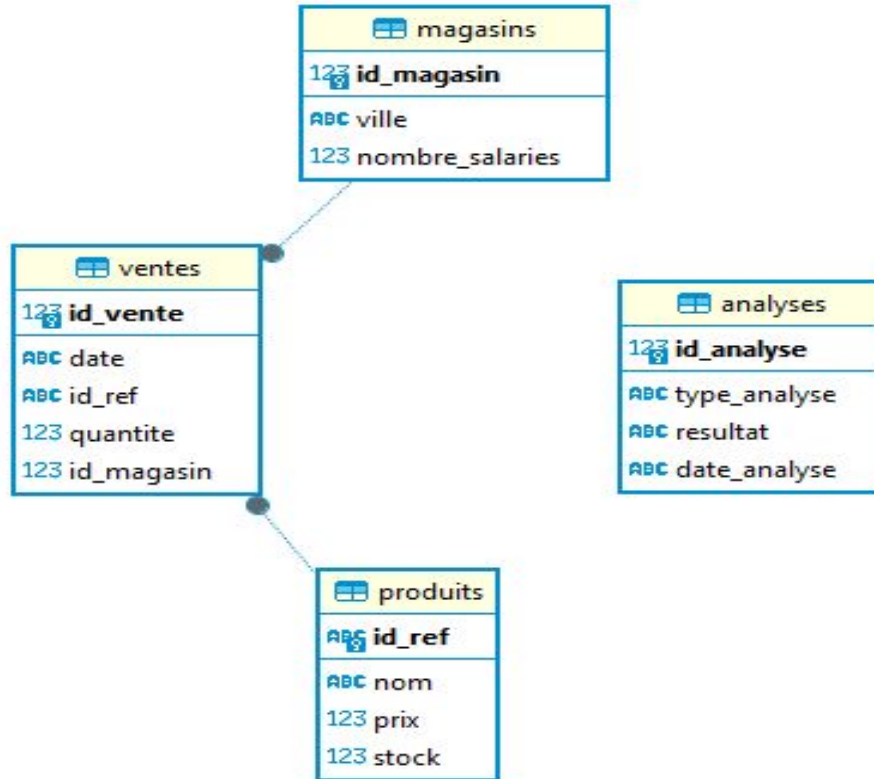
- Créer et mettre en œuvre un environnement à deux services : un pour l'exécution des scripts (import de données) et un autre pour la base de données;
- Analyser un jeu de données et en expliquer ses caractéristiques;
- Créer une base de données adaptée pour le stockage du jeu de données;
- Importer les données;
- Réaliser un premier niveau d'analyses de données avec SQL;
- Stocker les résultats des analyses.

2) Architecture & Base de données

Schéma Architecture de notre brief technique



2) Architecture & Base de données



Réalisé avec:



Power BI



3) Programmation : Docker files

Fichier docker-compose.yaml

```
services:
  db:
    image: nouchka/sqlite
    Volumes:
      - ./data:/root/db
    Tty: true

  Scripts:
    Build:
      context : .
      Dockerfile: Dockerfile
    Depends_on: db
    Volumes:
      - ./data:/root/db
```

Fichier Dockerfile

```
FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .
COPY main.py .

RUN pip install -r requirements.txt

CMD ["python", "main.py"]
```

3) Programmation : python & SQL

Fichier creation_db.py

```
Import pandas
Import sqlite3

conn =
sqlite3.connect("./root/db/notredb.db")
cursor = conn.cursor()

cursor.execute(
    create table ventes (id int pk, ....)
cursor.execute(
    create table magasins (id int pk, ....)
cursor.execute(
    create table produits (id int pk, ....)

conn.commit()
conn.close()
```

Fichier insertdata_from_url.py

```
Import pandas
Import sqlite3
Import requests

response = requests.get('url')

If response.status_code == 200:
    #insertions des données dans la
    base de données et la table
    correspondantes;
Else:
    Print("error")

conn.commit()
conn.close()
```

3) Programmation : python & SQL

Fichier analyse.sql

```

Import pandas
Import sqlite3

conn = sqlite3.connect("/root/db/ventes.db")
cursor = conn.cursor()

# 1.Requete Chiffre d'affaires total
cursor.execute("""
    SELECT SUM(v.quantite * p.prix) AS
chiffre_affaires_total
    FROM ventes v
    JOIN produits p ON v.id_ref = p.id_ref
""")
ca_total = cursor.fetchone()[0]
print(f"=> Chiffre d'affaires total :
{ca_total} €")

```


4) Résultats & Visualisations

Ce document présente une synthèse des résultats obtenus suite à l'analyse des ventes d'une PME.

L'objectif est de comprendre la dynamique des ventes selon les produits(A,B,C,D et E) et les localisations géographiques afin d'éclairer la prise de décision stratégique. Une visualisation a été mis en place avec Google Looker studio(également réalisable avec PowerBI)

Objectifs de l'analyse :

- Identifier le chiffre d'affaires global réalisé
- Repérer les produits les plus vendus
- Analyser les performances par ville (région)

Faits saillants :

✓ Chiffre d'affaires total : **5 268,78 €**

✓ Produits les plus vendus : Produit **E**, Produit **A** et Produit **D**

✓ Villes les plus performantes : **Lyon, Marseille, Bordeaux**

Les données ont été collectées automatiquement via des fichiers CSV en ligne puis stockées dans une base SQLite.
Les résultats ont été exploités à l'aide de requêtes SQL et visualisés avec Google Looker Studio

Résultat d'analyse par Malik HOUNI

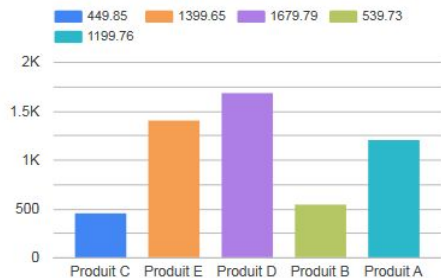
[Interactive visualisation fait avec Google Looker - cliquer ici](#)

Chiffre d'affaire

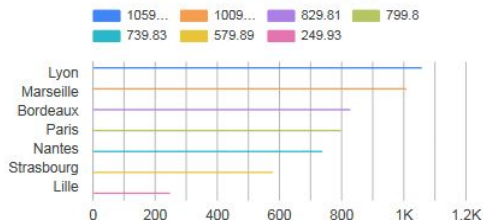
resultat

5,268.78

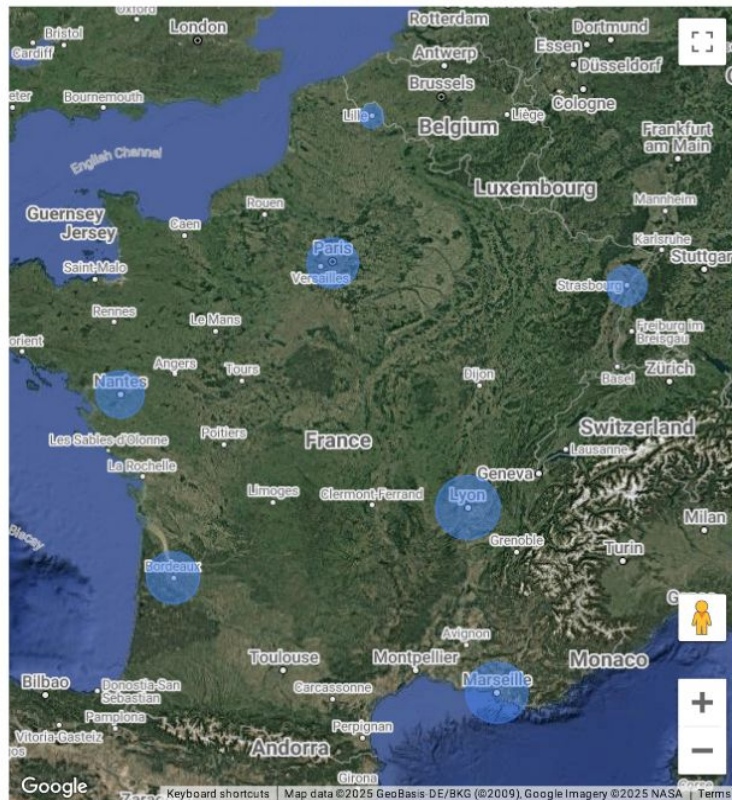
Vente par produits



Ventes par ville et montant



Ventes par villes



montant 249.93 • 1,059.79

Fait avec
 Looker

Conclusion

- **Nous** avons une architecture de données adapté au problème métier
- **Utilisation** de docker (conteneurisation)
- **Résultats** et visualisation(looker, streamlit,...)
- **Possible** améliorations:
 - Schéma en étoiles
 - Autre que Sqlite (Postgre, cloud,...)
 - Sécurité des données
 - Penser grande échelle: RDD pour le futur

Fin de la présentation

Merci! Avez vous des questions?

The logo for 'SIMPLON.CO' is displayed within a black rectangular border. The text 'SIMPLON.CO' is in a bold, black, uppercase sans-serif font. The letter 'O' is replaced by a red circle containing a white exclamation mark, matching the 'simplon' logo.

SIMPLON.CO

Ce projet à été réalisé
par Malik HOUNI dans
le cadre de la candidature
pour la formation
Data Engineer.