

Rapport projet

Nom : cheragui
Prenom : mayar

1. Introduction

Ce projet a pour objectif de concevoir une **solution de Business Intelligence (BI) complète** à partir de la base de données **Northwind**.

L'ensemble du projet est réalisé **exclusivement en Python**, conformément aux consignes, et couvre toutes les étapes classiques d'un projet BI : **extraction des données, transformation, modélisation analytique et visualisation**.

Les données sources sont stockées dans une base **SQL Server**, puis intégrées dans un pipeline ETL développé en Python.

Le projet est réalisé **en monôme (individuel)** et vise à fournir une solution **reproductible, claire et orientée analyse décisionnelle**.

2. Architecture BI

L'architecture du projet repose sur une approche **ETL classique** et une séparation claire des responsabilités.

Architecture générale :

1. **Source de données :**
Base relationnelle SQL Server (Northwind)
2. **ETL (Extract – Transform – Load) :**
Scripts Python utilisant la bibliothèque pandas
3. **Stockage analytique :**
Fichiers CSV organisés selon un modèle en étoile
4. **Visualisation / Analyse :**
Dashboard analytique généré en Python (matplotlib)

Cette architecture permet une **bonne maintenabilité, une reproductibilité complète**, et respecte les principes fondamentaux de la BI.

3. Source de données (**SQL Server – Northwind**)

La base de données **Northwind** est une base d'exemple fournie par Microsoft, représentant l'activité commerciale d'une entreprise (commandes, clients, produits, catégories, etc.).

Tables principales utilisées :

- Customers
- Orders
- Order Details
- Products
- Categories

La base a été déployée sur **SQL Server dans un conteneur Docker**, ce qui permet :

- une installation indépendante du système
- une portabilité du projet
- une exécution stable sans dépendre d'un serveur externe

Les données sont ensuite interrogées depuis Python à l'aide d'un script ETL.

4. ETL en Python

L'ETL est divisé en plusieurs scripts afin de respecter une **bonne organisation du code**.

4.1 Extraction (Extract)

Le script `etl_sqlserver.py` se connecte à SQL Server et extrait les tables nécessaires. Chaque table est exportée sous forme de fichier CSV dans le dossier `data/raw/`.

Exemples de tables extraites :

- Customers
- Orders
- Order Details
- Products
- Categories

Cette étape permet d'isoler les données sources et de faciliter les traitements ultérieurs.

4.2 Transformation (Transform)

Le script `etl_transform.py` réalise les transformations nécessaires :

- conversion des dates
- jointures entre Orders, Order Details et Products
- calcul du **chiffre d'affaires (Revenue)** :

$$\text{Revenue} = \text{UnitPrice} \times \text{Quantity} \times (1 - \text{Discount})$$
$$\text{Revenue} = \text{UnitPrice} \times \text{Quantity}$$
$$\text{Revenue} = \text{UnitPrice} \times \text{Quantity} \times (1 - \text{Discount})$$

- suppression des valeurs incohérentes ou manquantes

- sélection des colonnes utiles à l'analyse

Le résultat est un fichier `sales.csv` stocké dans `data/processed/`.

4.3 Dimensions analytiques

Le script `etl_dimensions.py` permet de créer les **tables de dimensions** :

- `dim_customer`
- `dim_product`
- `dim_category`

Ces dimensions sont séparées de la table de faits afin de faciliter l'analyse et respecter une modélisation BI standard.

5. Modélisation des données (Schéma en étoile)

Les données sont organisées selon un **schéma en étoile**, très utilisé en Business Intelligence.

Table de faits :

- **sales**
 - OrderID
 - OrderDate
 - CustomerID
 - ProductID
 - Quantity
 - UnitPrice
 - Discount
 - Revenue

Tables de dimensions :

- `dim_customer`
- `dim_product`
- `dim_category`

Cette modélisation permet :

- des analyses rapides
 - une lecture claire des indicateurs
 - une séparation logique entre faits et dimensions
-

6. Dashboard analytique en Python

Conformément aux consignes, la visualisation est réalisée **exclusivement en Python**, sans Power BI.

Le script `dashboard.py` génère automatiquement plusieurs indicateurs et graphiques :

Indicateurs clés (KPI) :

- Chiffre d'affaires total
- Nombre total de commandes

Visualisations :

- Chiffre d'affaires par catégorie (bar chart)
- Évolution du chiffre d'affaires dans le temps (line chart)

Les graphiques sont générés à l'aide de **matplotlib** et sauvegardés sous forme d'images dans le dossier `outputs/figures/`.

Ces visualisations constituent le **dashboard analytique final** du projet.

7. Conclusion

Ce projet a permis de mettre en œuvre une **chaîne BI complète en Python**, depuis une base relationnelle SQL Server jusqu'à un tableau de bord analytique.

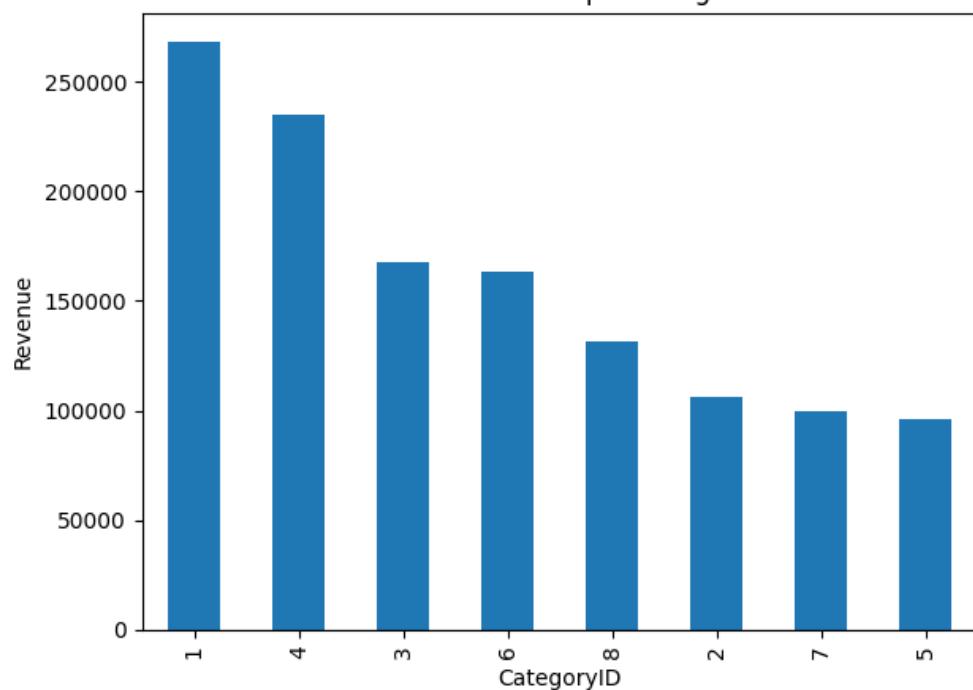
Les objectifs ont été atteints :

- intégration de données multi-tables
- transformation et nettoyage des données
- modélisation analytique
- visualisation des indicateurs clés

La solution proposée est **reproductible, clairement structurée**, et respecte les bonnes pratiques de la Business Intelligence.

Ce projet démontre la capacité de Python à couvrir l'ensemble du cycle BI sans dépendre d'outils externes.

Chiffre d'affaires par catégorie



Évolution du chiffre d'affaires

