ETUDIANT : LAOUAR Mohamed Anis

MATRICULE : 212134016901

RAPPORT DE PROJET (TBI) : Conception et réalisation d'un Data Warehouse Northwind et analyse sous Power BI et Python

1. Introduction

Dans le cadre de ce projet académique, nous avons réalisé une solution complète de **Business Intelligence** basée sur la base de données **Northwind**.
L'objectif principal est de **concevoir un Data Warehouse**, d'y intégrer des données provenant de **plusieurs sources hétérogènes**, puis de réaliser des **analyses décisionnelles** à l'aide de **Power BI** et **Python**.

Ce projet couvre l'ensemble du cycle décisionnel :

- extraction des données,
- transformation,
- modélisation en Data Warehouse,
- visualisation,
- et analyse avancée.

2. Objectifs du projet

Les objectifs principaux du projet sont les suivants :

- Intégrer des données issues de **deux sources différentes** :
    - SQL Server
    - Microsoft Access
- Concevoir un **Data Warehouse** en schéma en étoile
- Réaliser un processus **ETL** (Extract, Transform, Load)
- Analyser les commandes **expédiées** et **non expédiées**

- Créer un **dashboard interactif sous Power BI**
- Exploiter les données avec **Python** pour des analyses complémentaires

3. Sources de données

3.1 Base de données SQL Server (Northwind)

La première source de données est la base **Northwind sous SQL Server**.
Elle contient un volume de données plus important et une structure relationnelle complète.

Les tables principales utilisées sont :

- Orders
- Customers
- Employees

Ces tables contiennent respectivement :

- les informations sur les commandes,
- les clients,
- les employés.

3.2 Base de données Microsoft Access

La deuxième source est une base **Northwind sous Microsoft Access**.

Elle contient :

- moins de données,
- une structure légèrement différente,
- des types de données parfois non homogènes.

Cette base a été utilisée pour simuler un **contexte réel d'intégration de sources hétérogènes**.

4. Conception du Data Warehouse

4.1 Choix de l'architecture

Nous avons choisi une architecture **en schéma en étoile (Star Schema)**, car elle est :

- simple à comprendre,
- optimisée pour l'analyse,

- très utilisée dans les systèmes décisionnels.

4.2 Schéma du Data Warehouse

Le Data Warehouse est composé de :

- **Table de faits**
  - **FactOrders**
- **Tables de dimensions**
  - **DimTime**
  - **DimCustomer**
  - **DimEmployee**

4.3 Question analytique principale

Le Data Warehouse a été conçu pour répondre à la question suivante :

Combien de commandes sont expédiées et non expédiées selon le temps, le client et l'employé ?

5. Processus ETL avec Power BI (Source SQL Server)

Power BI a été utilisé comme outil ETL via le **Power Query Editor**.

5.1 Création de la dimension DimCustomer

- La table **Customers** a été **dupliquée** afin de préserver la table source.
- Suppression des colonnes inutiles.
- Conservation des attributs analytiques :
  - CustomerID
  - CompanyName
  - ContactName
  - City
  - Country
- Suppression des doublons.
- Création d'une **clé technique CustomerKey**.

5.2 Création de la dimension DimEmployee

- Duplication de la table **Employees**.
- Fusion des colonnes **FirstName** et **LastName** pour créer **ContactName**.
- Suppression des colonnes non nécessaires.
- Conservation :
    - EmployeeID
    - ContactName
    - City
    - Country
- Suppression des doublons.
- Création de la clé **EmployeeKey**.

5.3 Création de la dimension DimTime

- Utilisation de la colonne **OrderDate** depuis la table Orders.
- Suppression des doublons.
- Création de nouvelles colonnes :
    - Year
    - Month
- Création de la clé **TimeKey**.

5.4 Création de la table de faits FactOrders

À partir de la table **Orders** :

- **Colonnes de mesure**
    - **ShippedOrders** :
        - valeur 1 si ShippedDate n'est pas NULL
        - valeur 0 sinon
    - **NotShippedOrders** :
        - valeur 1 si ShippedDate est NULL

    - valeur 0 sinon

🔸 **Clés étrangères**

- CustomerKey
- EmployeeKey
- TimeKey

5.5 Création des relations

Après chargement :

- création des relations entre la table de faits et les dimensions,
- validation du schéma en étoile.

6. Intégration de la source Access

- Export des tables Access en **CSV**.
- Import dans Power BI.
- Application des **mêmes transformations ETL** :
    - création des dimensions,
    - création de la table de faits,
    - création des clés.
- **Append** des données Access avec celles de SQL Server.

7. Visualisation avec Power BI

7.1 Page 1 – *Order Fulfillment Overview*

Contenu :

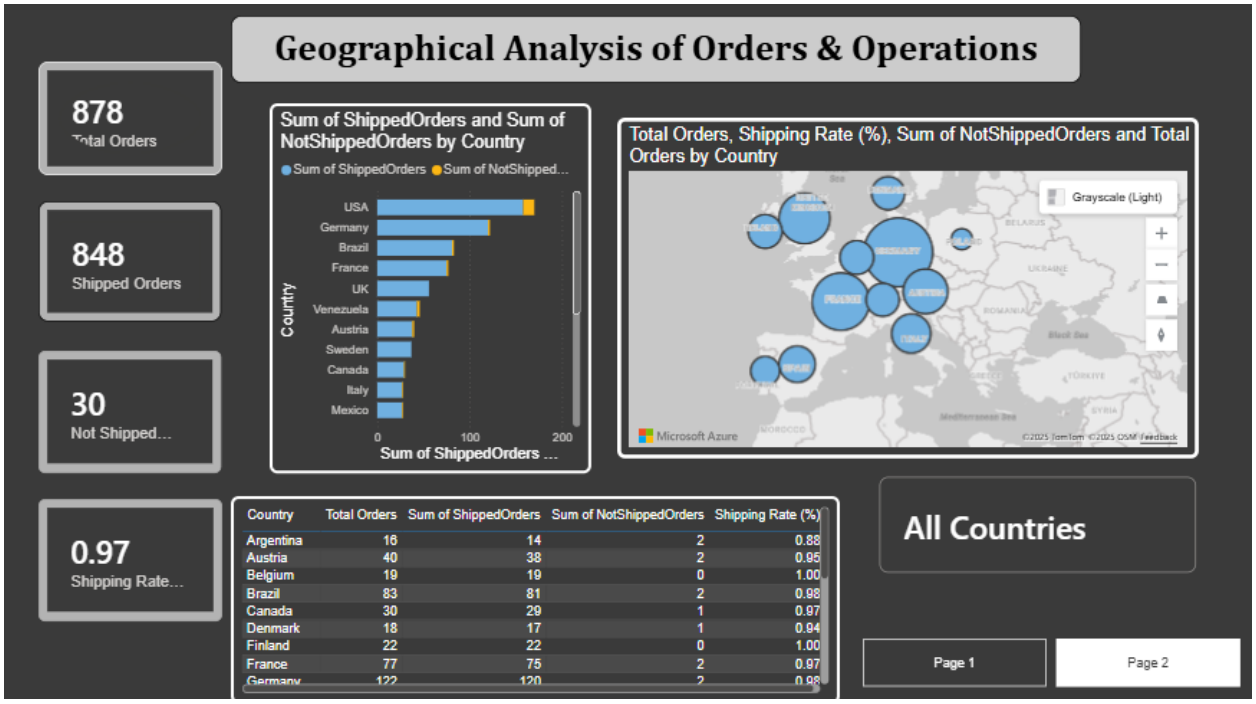
- Cartes KPI :
    - Total Orders
    - Shipped Orders
    - Not Shipped Orders
    - Shipping Rate
- Donut chart (Shipped vs Not Shipped)

- Bar charts :
    - Orders by Employee
    - Orders by Country
- Line chart :
    - Orders by Year
- Slicer sur l'année

7.2 Page 2 – *Geographical Analysis of Orders & Operations*

Contenu :

- Carte géographique des pays
- Mesure dynamique du pays sélectionné
- Tableau récapitulatif
- Bar chart empilé (Shipped / Not Shipped)
- Cartes KPI dynamiques



8. Limites de Power BI pour le chargement

Power BI est efficace pour l'ETL et la visualisation, mais :

- ne permet pas un chargement direct et permanent dans SQL Server,
- n'est pas idéal pour l'automatisation complète.

9. Chargement et analyse avec Python

Pour pallier cette limitation :

- Export des tables en **Excel**
- Connexion à SQL Server via Python
- Insertion des données dans le DWH
- Jointure des tables du schéma en étoile
- Visualisation des données avec Python

10. Conclusion

Ce projet démontre :

- la conception d'un Data Warehouse,

- l’intégration de sources multiples,
- l’utilisation avancée de Power BI pour l’ETL,
- l’analyse et la visualisation avec Python.

Il s’agit d’une solution complète de **Business Intelligence de bout en bout**.