

RAPPORT FINAL

"Projet Business Intelligence"

1. **ÉTUDIANT :**

BALAMANE Mohamed Hani

2. **Matricule :**

232331602002

3. **Groupe :**

02

4. **Filière :**

**3ème année ingénieur en informatique
Spécialité sécurité informatique**

Année universitaire : 2025-2026

Table des matières

1	Introduction	2
2	Objectifs du projet & périmètre	2
3	Architecture générale & structure livrée	2
4	Processus ETL : extraction, transformation, chargement	4
5	Conception du Data Warehouse	5
6	Scripts Python & Notebook	5
7	KPI & Dashboard analytique (toutes les statistiques)	6
8	Exécution et reproduction du projet	10
9	Justification des choix techniques	10
10	Qualité des données & tests	11
11	Conclusion	11

1 Introduction

L'objectif de ce projet est de concevoir une solution BI complète à partir des données Northwind issues de SQL Server et Microsoft Access. Le projet couvre l'ensemble du pipeline décisionnel :

-  Extraction et nettoyage des données
-  Transformation avancée via Python
-  Construction d'un Data Warehouse en étoile
- Calcul d'indicateurs KPI
- Production d'un dashboard analytique (PNG + HTML)
- Documentation complète et vidéo explicative

2 Objectifs du projet & périmètre

⌚ Objectifs principaux

-  Construire un Data Warehouse cohérent basé sur Northwind
-  Réaliser un ETL Python complet
- Générer des indicateurs métier (KPI)
- Produire un dashboard professionnel regroupant toutes les statistiques
-  Livrer un projet entièrement reproductible

📦 Périmètre des livrables

- `build_dw.py`, `compute_kpi_final-B1.py`
- `analysis-final-B1.ipynb`
- Dossiers `/data`, `/scripts`, `/figures`, `/reports`
- Dashboard : PNG + HTML
- Rapport PDF + vidéo explicative
- README pour exécution

3 Architecture générale & structure livrée

Le projet suit une architecture BI standard :

RAW → PROCESSED → DATA WAREHOUSE → KPI → FIGURES → ANALYTICS

📁 Structure finale livrée

```
BL_NORTHWIND_TRADERS_PR...
data
    dw
        DimCustomer.csv
        DimDate.csv
        DimEmployee.csv
        DimProduct.csv
        DimShipper.csv
        FactOrderDetails.csv
        FactOrders.csv
processed
    Customers_clean.csv
    Employees_clean.csv
    kpl_client.csv
    kpl_dataset.csv
    kpl_employee.csv
    OrderDetails_clean.csv
    Orders_clean.csv
    Products_clean.csv
    Shippers_clean.csv
    Suppliers_clean.csv
raw
access
    Customers.csv
    Employees.csv
    Order Details.csv
    Orders.csv
    Products.csv
    Shippers.csv
sqlserver
    Categories.csv
    CustomerCustomerDemo.csv
    CustomerDemographics.csv
    Customers.csv
    Employees.csv
    EmployeeTerritories.csv
    Order Details.csv
    OrderDetails.csv
    Orders.csv
    Products.csv
    Region.csv
    Shippers.csv
    Suppliers.csv
    Territories.csv
```

```
Northwind2012.accdb
figures
    average_basket_final-B1.html
    average_basket_final-B1.png
    dashboard_complet_final-B1.html
    dashboard_complet_final-B1.png
    dashboard_total_final-B1.html
    orders_evolution_final-B1.html
    orders_evolution_final-B1.png
    revenue_per_month_final-B1.html
    revenue_per_month_final-B1.png
    sales_country_pie_final-B1.html
    sales_country_pie_final-B1.png
    sales_per_country_final-B1.html
    sales_per_country_final-B1.png
    sales_per_month_final-B1.html
    sales_per_month_final-B1.png
    top_clients_final-B1.html
    top_clients_final-B1.png
    top_employees_final-B1.html
    top_employees_final-B1.png
    top_products_final-B1.html
    top_products_final-B1.png
notebooks
    analysis-final-B1.ipynb
reports
    data_quality_report.md
    Rapport_final_BL_Northwind.docx
scripts
    build_dw.py
    compute_kpi_final-B1.py
venv
video
env
.gitignore
README.md
requirements.txt
```

4 Processus ETL : extraction, transformation, chargement

Extraction

- Récupération des données Access ([Northwind2012.accdb](#))
- Export des tables en CSV dans [data/raw/access/](#)

- Import de tables SQL Server exportées dans `data/raw/sqlserver/`

Transformation

- Nettoyage des colonnes
- Harmonisation des types
- Conversion des dates
- Génération des clés (`DateKey` au format YYYYMMDD)
- Fusion nécessaire pour préparer les KPI
- Calcul du chiffre d'affaires :

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

Chargement

- Création des dimensions & faits dans `data/dw/`
- Chargement automatique dans SQL Server (schéma dw) via `sqlalchemy` + `pyodbc`

5 Conception du Data Warehouse

Un modèle en étoile (Star Schema) a été conçu :

Dimensions

- **DimCustomer** : CustomerKey, CustomerName, Country
- **DimProduct** : ProductKey, ProductName
- **DimEmployee** : EmployeeKey, FirstName, LastName, Country
- **DimShipper** : ShipperKey, ShipperName
- **DimDate** : DateKey, Date, Year, Month, MonthName, Quarter

Tables de faits

- **FactOrders** : OrderKey, CustomerKey, EmployeeKey, ShipperKey, DateKey
 - **FactOrderDetails** : ProductKey, OrderKey, UnitPrice, Quantity, Discount
- Les fichiers CSV DW sont présents dans `data/dw/`.

6 Scripts Python & Notebook

`build_dw.py`

- Construit les dimensions et faits
- Sauvegarde dans `data/dw/`
- Charge dans SQL Server (schema dw)

compute_kpi_final-B1.py

- Fusionne DW (Dim + Fact)
- Calcule les KPI
- Génère toutes les images PNG
- Génère le Dashboard complet

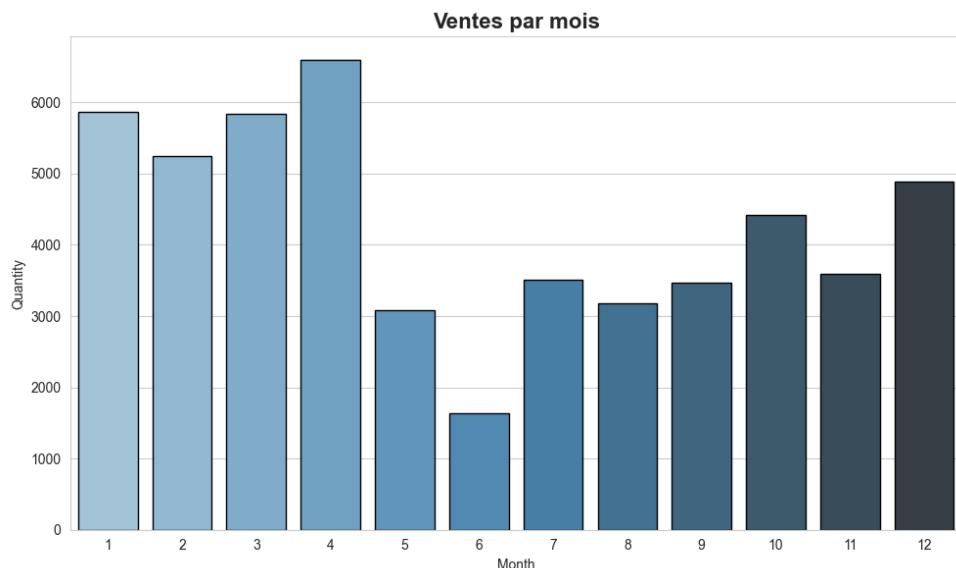
analysis-final-B1.ipynb

- Affiche les images
- Crée un fichier HTML pour chaque image
- Crée le [dashboard_total_final-B1.html](#)

7 KPI & Dashboard analytique (toutes les statistiques)

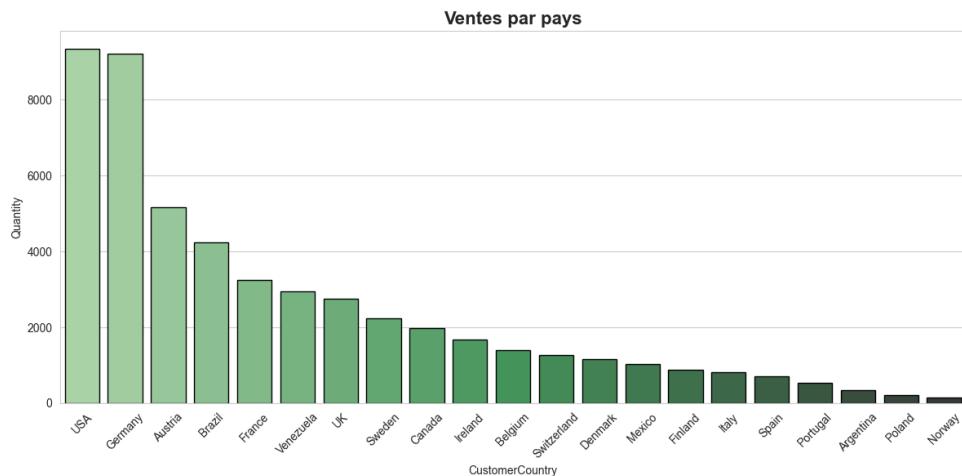
Les visualisations suivantes ont été générées automatiquement :

1. Ventes par mois



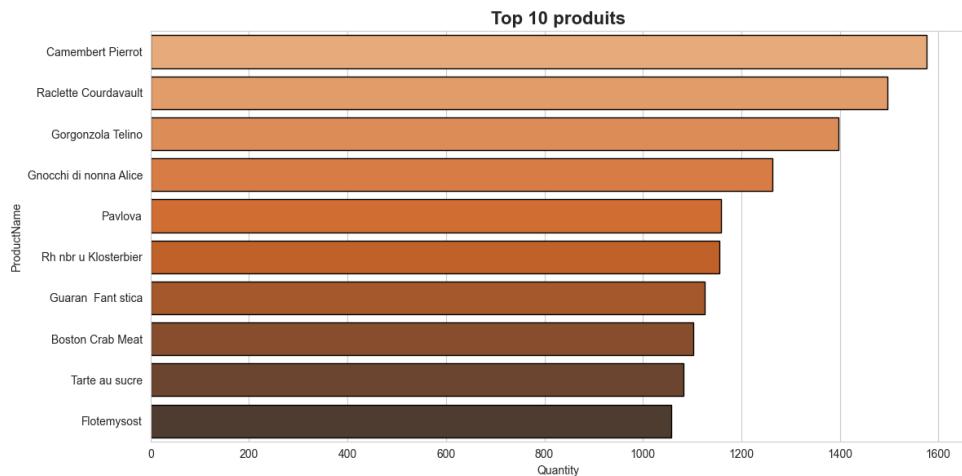
Analyse du volume mensuel des produits vendus.

2. Ventes par pays



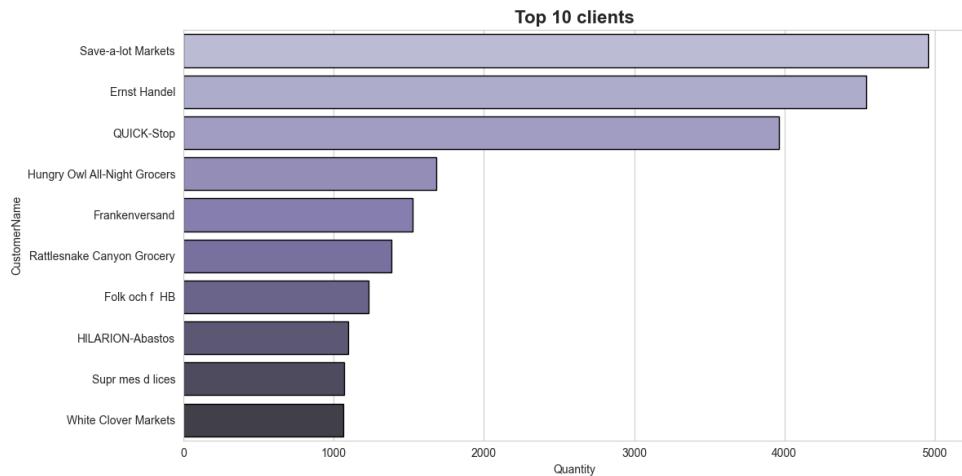
Classement par quantité totale vendue.

3. Top 10 produits



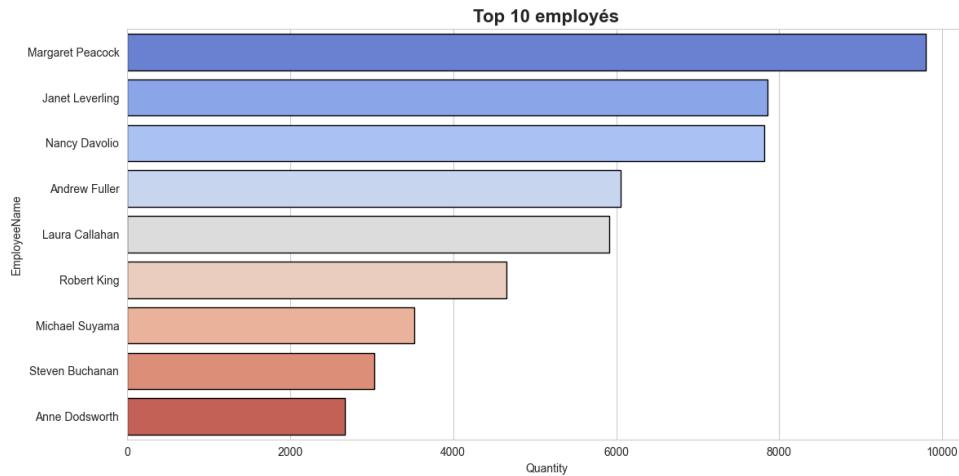
Produits les plus vendus en volume.

4. Top 10 clients



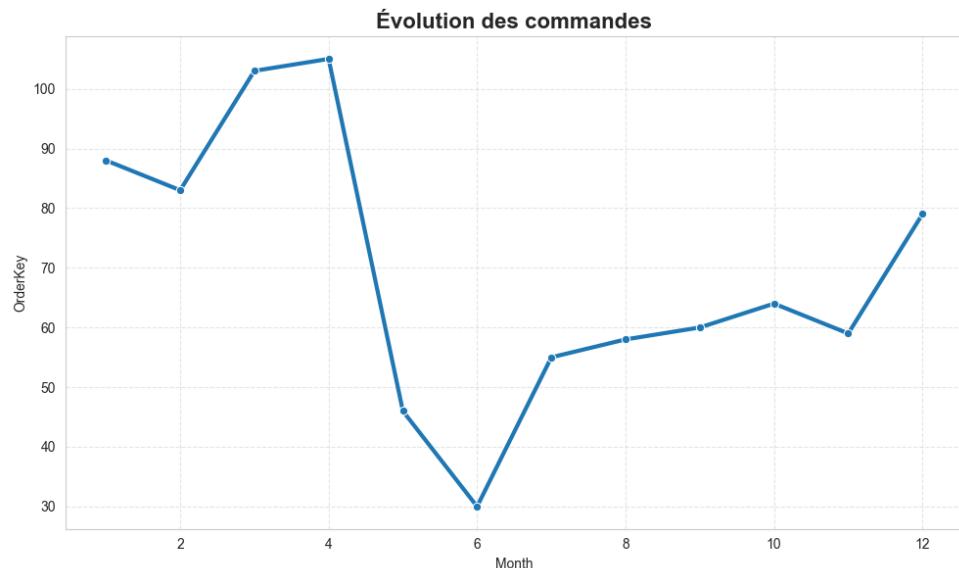
Clients les plus actifs en termes de quantité achetée.

5. Top 10 employés



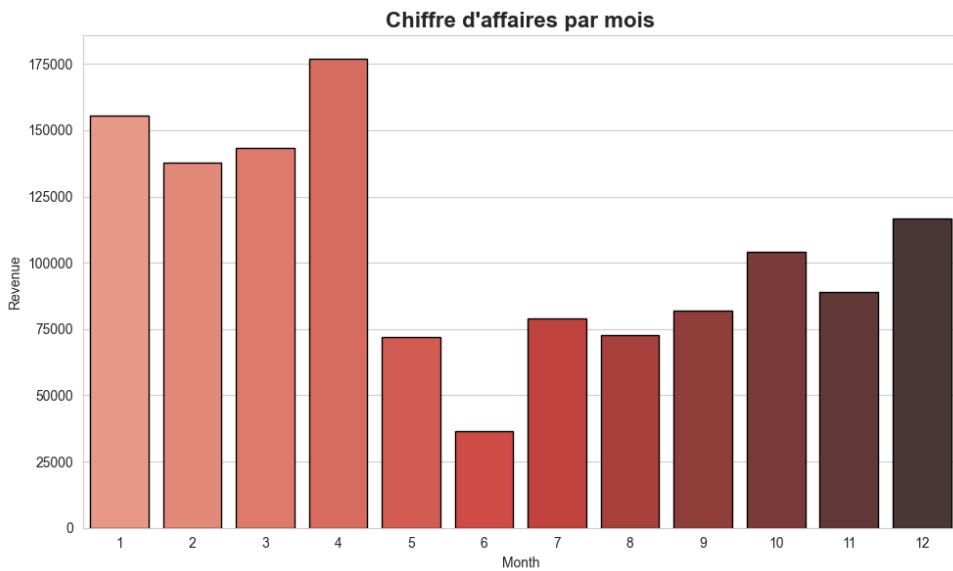
Performances des employés.

6. Évolution des commandes



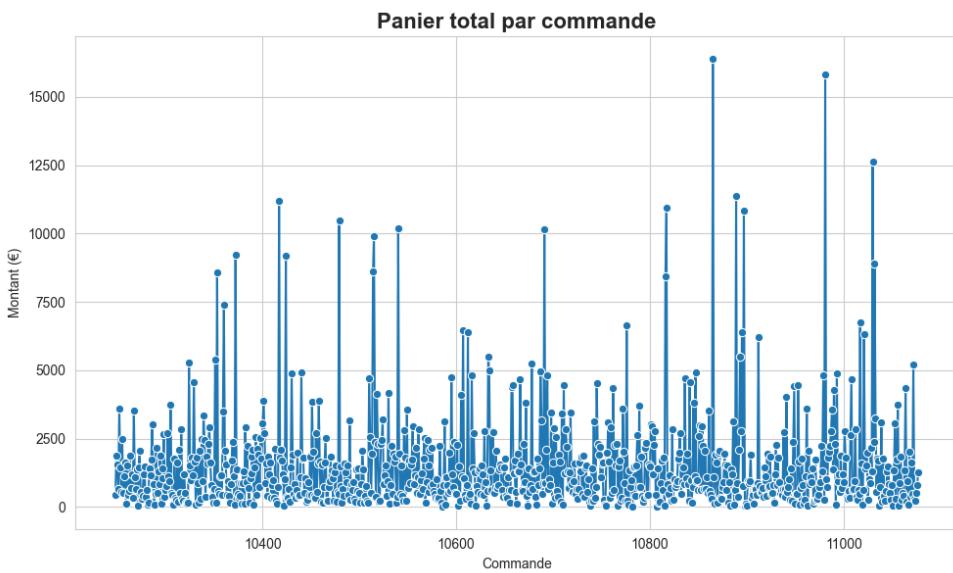
Analyse de l'évolution du nombre de commandes.

7. Chiffre d'affaires par mois



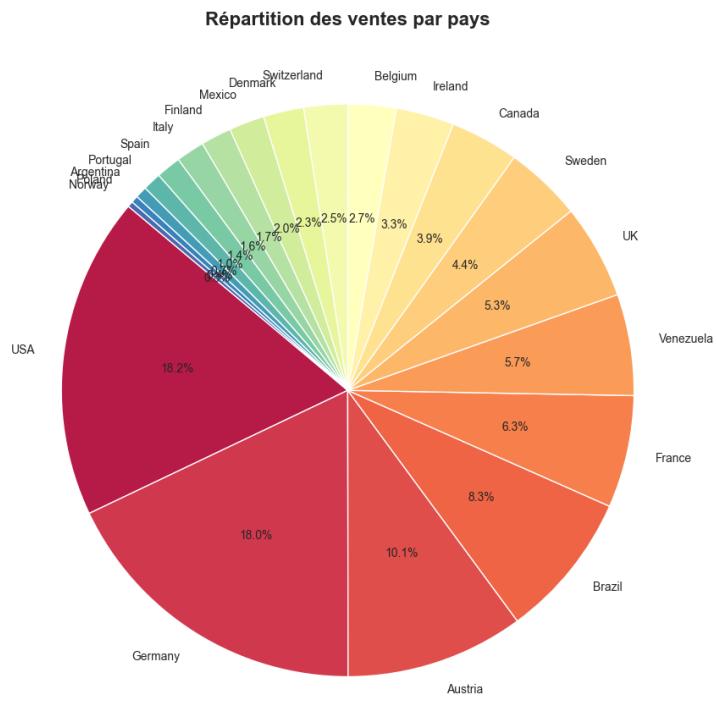
Montant total généré chaque mois.

8. Panier total par commande



Analyse du montant total par commande.

9. Répartition des ventes (Pie Chart)



10. Dashboard HTML final

🌐 [dashboard_total_final-Bl.html](#)

Affiche tous les KPI dans une page web stylisée.

8 Exécution et reproduction du projet

🔧 Installation

```
python -m venv venv
venv\Scripts\activate
pip install -r requirements.txt
```

▶ Étapes d'exécution

```
python scripts/build_dw.py
python scripts/compute_kpi_final-Bl.py
jupyter lab
```

Le dashboard HTML est dans :

[figures/dashboard_total_final-Bl.html](#)

9 Justification des choix techniques

- 🐍 Python : simplicité, richesse des bibliothèques

-  **Pandas** : manipulation performante des données
- **Matplotlib / Seaborn** : graphiques professionnels
-  **SQLAlchemy + pyodbc** : intégration fiable avec SQL Server
-  **Jupyter Notebook** : idéal pour analyses et présentation
-  **Schéma en étoile** : modèle BI classique, simple, efficace

10 Qualité des données & tests

-  Vérification des clés
-  Conversion des dates
-  Comparaison des totaux avant/après transformation
-  Contrôle des colonnes attendues
-  Détection des valeurs nulles
- Logs d'erreurs lors du chargement SQL Server

11 Conclusion

Le projet **FINAL-BI Northwind** aboutit à une solution BI complète, fonctionnelle et entièrement automatisée. Les données brutes (Access + SQL Server) sont transformées en un Data Warehouse fiable, puis analysées pour produire des KPI pertinents et un dashboard professionnel en HTML & PNG.

La solution est réplicable, documentée, et peut évoluer : Power BI, Streamlit, automatisation Airflow, SCD2, etc. Ce travail démontre la maîtrise du pipeline BI complet :

ETL → DW → KPI → Dashboard