

Rapport Projet BI

Nom : KHOUATMI BOUKHATEM

Prenom : Hanna

Matricule : 232332414317

Groupe : 4

1. Introduction du Projet

Dans un contexte où les données occupent une place centrale dans la prise de décision stratégique, les systèmes de **Business Intelligence (BI)** jouent un rôle fondamental pour transformer des données brutes en informations exploitables.

Ce projet a pour objectif de concevoir et de réaliser **une solution BI complète**, depuis la source de données jusqu'à la visualisation analytique, en s'appuyant sur la base de données **Northwind**.

L'approche adoptée repose sur :

- un **ETL développé en Python**,
- une **architecture modulaire et reproductible**,
- et un **dashboard interactif** orienté aide à la décision.

2. Objectifs du projet

Les objectifs principaux de ce projet sont :

- Concevoir une **solution BI de bout en bout**
- Intégrer des données issues de **SQL Server (Northwind)**
- Mettre en place un **processus ETL fiable et justifié**
- Créer un **tableau de bord analytique interactif**
- Réaliser une **analyse multidimensionnelle (OLAP 3D)**
- Justifier les **choix techniques et architecturaux**

3. Présentation de la base de données Northwind

La base **Northwind** est une base de données relationnelle classique utilisée à des fins pédagogiques.

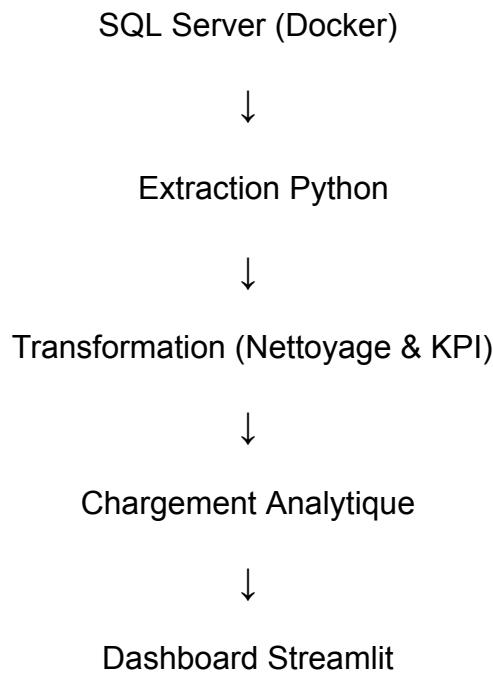
Elle représente le système d'information d'une entreprise commerciale et contient notamment :

- Les commandes (**Orders**)
- Les clients (**Customers**)
- Les employés (**Employees**)
- Les produits
- Les dates de commande et de livraison

Cette base est particulièrement adaptée aux projets BI grâce à sa richesse analytique et à sa structure relationnelle claire.

4. Architecture globale de la solution BI

L'architecture mise en place suit une logique **BI moderne**, structurée en plusieurs couches :



Cette séparation garantit :

- la maintenabilité
- la scalabilité
- la traçabilité des données

4.1. Structure du Projet

```
└── /data
    ├── /raw          # Données brutes extraites (Raw Material)
    ├── /processed    # Données nettoyées (Staging Area)
    └── /external_files # Exports finaux pour l'analyse

    └── /scripts
        ├── 1_extract.py   # Extraction SQL vers CSV
        ├── 2_transform.py  # Nettoyage et enrichissement (Logique BI)
        ├── 3_load.py      # Chargement et tri final
        └── dashboard.py   # Application Streamlit

    └── /reports         # Documentation et rapports
    └── /figures          # Captures des visualisations
    └── README.md        # Documentation technique
```

5. Mise en place de l'environnement technique

5.1 Docker & SQL Server

Docker est utilisé pour lancer une instance de **SQL Server**

Avantages :

- Environnement isolé
- Facilité de déploiement
- Reproductibilité du projet

5.2 DBeaver

DBeaver est utilisé pour :

- Explorer la base Northwind
- Vérifier les tables
- Valider les données extraites

Cet outil facilite la compréhension du schéma relationnel.

Note spécifique sur l'environnement de développement :

Le développement a été effectué sur un environnement macOS. En raison de problèmes de compatibilité et d'instabilité rencontrés avec Jupyter Notebooks sur cette plateforme (notamment liés aux noyaux et à la gestion des environnements virtuels graphiques), il a été décidé de privilégier des scripts Python purs (.py).

6. Conception et réalisation de l'ETL

L'ETL est implémenté en **Python**, avec une architecture claire en trois étapes distinctes.

6.1 Phase d'Extraction (Extract)

L'extraction vise à récupérer l'ensemble des commandes disponibles depuis plusieurs sources, afin de créer un dataset unique et complet.

Sources et traitements :

1. **SQL Server** : extraction des commandes historiques (~830 commandes), enrichissement avec les noms des clients et des employés à partir des tables `Customers` et `Employees`.
2. **Excel – Orders.xlsx** : lecture des commandes récentes, correction et harmonisation des colonnes (`OrderID`, `CustomerID`, `EmployeeID`, `OrderDate`, `ShippedDate`, `CompanyName`), jointure avec les référentiels SQL pour compléter les informations manquantes.
3. **Fusion finale et dédoublonnage** : concaténation de toutes les sources et suppression des doublons sur `OrderID`, en donnant priorité aux données SQL en cas de conflit.
4. **Résultat** : sauvegarde du fichier consolidé `raw_merged_data.csv` avec traçabilité de la provenance des données (`Source`).

Cette étape garantit un jeu de données complet, fiable et prêt pour la phase de transformation.

6.2 Phase de Transformation (Transform)

Cette étape est centrale dans une logique BI, car elle permet de transformer les données brutes en données analytiques.

- **Traitements réalisés :**
- Chargement du fichier `raw_merged_data.csv`

- Conversion des champs :
 - `OrderDate`
 - `ShippedDate`
- Création d'un **KPI métier** :
 - `Livré si ShippedDate existe`
 - `Non Livré sinon`

Gestion intelligente des valeurs manquantes :

Contrairement à une suppression brutale, une approche BI raisonnée a été adoptée :

- `CompanyName` manquant → "**Client Inconnu**"
- `EmployeeName` manquant → "**Non Assigné**"

Cette approche permet de **ne pas perdre des commandes valides**, ce qui est essentiel pour l'analyse décisionnelle.

Suppression ciblée :

- ❖ Suppression **uniquement** des lignes sans `OrderDate`
- ❖ Justification :
 - Impossible de réaliser une analyse temporelle sans date
 - Suppression légitime dans un contexte BI

Résultat :

- Sauvegarde dans `staging_data.csv`
- Traçabilité du nombre de lignes avant/après transformation

6.3 Phase de Chargement (Load)

L'étape Load permet de préparer les données pour la visualisation.

Actions réalisées :

- Lecture de `staging_data.csv`
- Tri des données par `OrderDate` décroissante
- Export vers `Northwind_Final_Analytics.csv`

Cette séparation entre données de **staging** et données **analytiques finales** respecte les bonnes pratiques BI.

7. Modélisation et logique Business Intelligence

La logique BI repose sur :

- **Dimensions :**
 - Date
 - Client
 - Employé
 - Statut de livraison
- **Mesures :**
 - Nombre de commandes
 - Volume de livraisons
 - Taux de livraison

Cette modélisation permet des analyses :

- temporelles
- par acteur (client / employé)
- par statut logistique

8. Conception du Dashboard Analytique

Le dashboard a été développé avec **Streamlit**, en mettant l'accent sur l'**UX/UI** et la lisibilité décisionnelle.



8.1 Filtres dynamiques

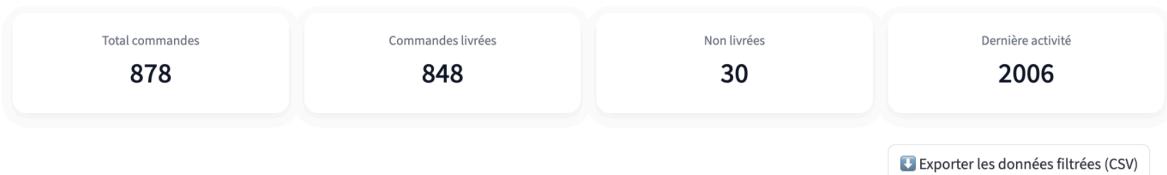
- Filtrage par **période**
- Filtrage par **statut de livraison**
- Mise à jour instantanée des indicateurs

8.2 Indicateurs clés (KPI)

- Total des commandes
- Nombre de commandes livrées
- Nombre de commandes non livrées
- Dernière année d'activité

Northwind Business Intelligence

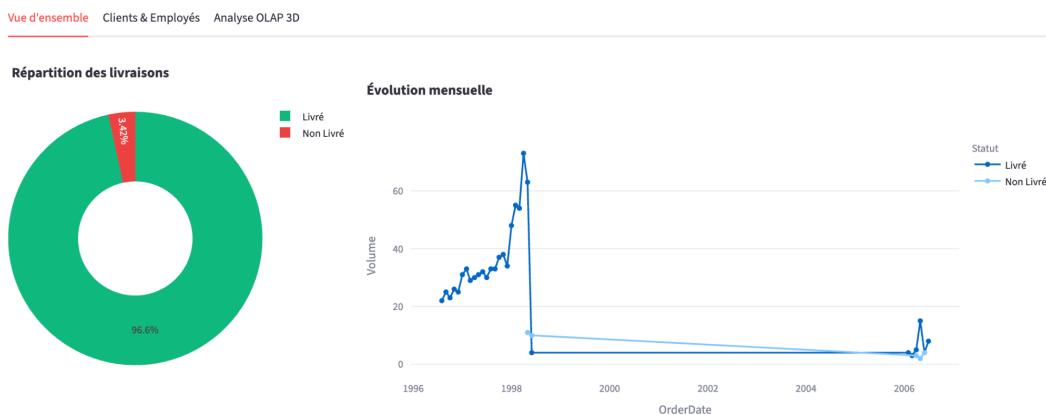
Analyse des commandes du 04/07/1996 au 23/06/2006



Ces KPI offrent une vision synthétique immédiate.

8.3 Visualisations analytiques

- Diagramme circulaire : répartition Livré / Non Livré
- Courbe temporelle : évolution mensuelle des commandes



- Histogramme :
 - Top 10 clients
 - Performance des employés



9. Analyse OLAP Multidimensionnelle (3D)

Une analyse **OLAP 3D** a été intégrée afin d'explorer les données selon :

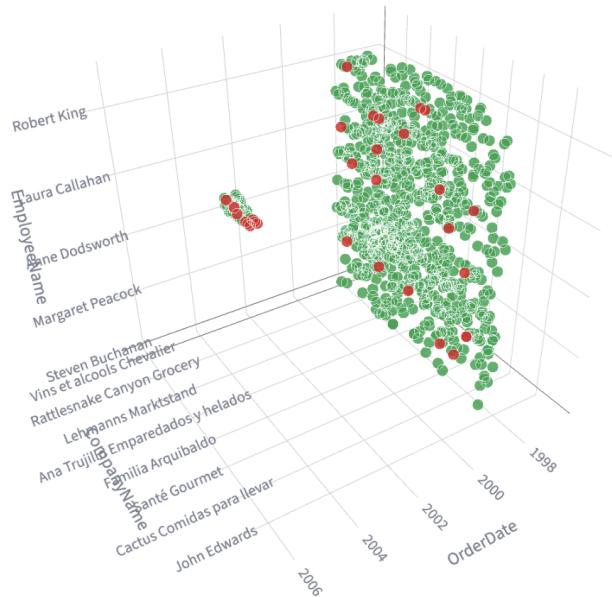
- Date
- Client
- Employé
- Statut

Cette visualisation permet :

- d'identifier les périodes critiques
- d'analyser la performance individuelle
- de détecter des anomalies logistiques

Le code couleur renforce la lisibilité :

- Vert → Livré
- Rouge → Non livré



10. Justification des choix techniques

Pourquoi Docker ?

- Déploiement rapide
- Isolation
- Reproductibilité

Pourquoi Python ?

- Flexibilité
- Écosystème BI riche
- Automatisation facile

Pourquoi Pandas ?

- Manipulation efficace des données
- Nettoyage avancé
- Performance adaptée au projet

Pourquoi Streamlit ?

- Rapidité de développement
- Interactivité native
- Idéal pour des dashboards BI académiques

Pourquoi Plotly ?

- Visualisations interactives
- Support natif de la 3D
- Intégration parfaite avec Streamlit

11. Conclusion et perspectives

Ce projet a permis de mettre en œuvre une **solution BI complète**, depuis la source de données jusqu'à l'analyse décisionnelle avancée.

Les objectifs ont été atteints avec :

- une architecture claire,
- un ETL robuste,
- un dashboard interactif et exploitable.

Perspectives d'amélioration :

- Intégration d'un Data Warehouse
- Ajout de prédictions (Machine Learning)
- Gestion des rôles utilisateurs
- Automatisation via pipelines CI/CD

Conclusion finale

Ce travail illustre une maîtrise des concepts BI, une rigueur méthodologique, et une bonne capacité de justification technique, conformes aux attentes d'un projet ING3 Business Intelligence.