

RAPPORT DU PROJET BUSINESS INTELLIGENCE



- Préparé par : BEKKAI lyna 232331632202
- Encadrant :Dr. MEKAHLIA Fatma Zohra
- Année universitaire : 2025/ 2026

SOMMAIRE

• Introduction générale	3
• Architecture Globale du Système BI	4...6
1. Sources de données	
2. Outils et technologies utilisées	
• Processus ETL	7...10
1. Extraction depuis SQL Server	
2.Extraction depuis Access	
3. Transformation et nettoyage	
4. Chargement dans le Data Warehouse	
• Développement du Tableau de Bord	11...15
1. Objectifs du tableau de bord	
2. Technologies utilisées	
3. Structure du dashboard	
4. Prévision simple intégrée	
5. Génération automatique d'un rapport PDF	
6. Résultat final	
• Analyse des Résultats	16...18
1. Performance commerciale	
2. Produits et catégories clés	
3. Analyse des comportements saisonniers	
4. Interprétation de la Prévision	
5. Conclusions tirées de l'analyse	
• Module de Prévision	19...21
1. Méthodologie	
2. Construction de la série temporelle	
3. Génération des dates futures	
4. Visualisation et interprétation	
• Difficultés et Solutions	22...23
• Conclusion générale	24



INTRODUCTION GÉNÉRALE

Dans un contexte où les entreprises doivent prendre des décisions rapides et basées sur des données fiables, la Business Intelligence (BI) joue un rôle essentiel. Elle permet de transformer des données brutes en informations organisées, compréhensibles et exploitables par les décideurs.

Ce projet s'inscrit dans cette démarche. À partir de la base de données Northwind, un environnement analytique complet a été conçu afin de suivre la performance commerciale de l'entreprise. Le travail a consisté à extraire les données opérationnelles depuis SQL Server et Microsoft Access, à les nettoyer et les structurer dans un Data Warehouse, puis à développer un tableau de bord interactif permettant de visualiser les indicateurs clés.

Ainsi, ce projet illustre l'ensemble du processus BI, de la donnée brute au reporting final, en mettant en évidence la valeur ajoutée de la Business Intelligence pour la gestion commerciale.



Architecture Globale du Système BI

L'architecture du projet BI repose sur une organisation structurée permettant de transformer les données brutes issues de l'environnement opérationnel en informations analytiques fiables. Elle suit le schéma classique d'un système décisionnel : Source → ETL → Data Warehouse → Dashboard.

1. Sources de données:

SQL Server – Base Northwind:

La base Northwind contient l'ensemble des informations opérationnelles nécessaires :

- Produits
- Catégories
- Clients
- Employés
- Commandes
- Détails des commandes et Shippers (transporteurs)

Ces tables constituent la source principale et sont extraites directement via un script Python se connectant à SQL Server.

Microsoft Access :

apporte des tables complémentaires (inventaire, fournisseurs, statuts, transactions...) permettant d'enrichir l'analyse

Ces deux bases, bien que provenant de systèmes différents, ont été intégrées dans un flux ETL commun afin de produire une vision unifiée et cohérente des données.

Architecture Globale du Système BI

2. Outils et technologies utilisées:

Le projet s'appuie sur une stack technique moderne et cohérente, adaptée au traitement et à la visualisation des données :

SQL Server:

Pour stocker les données opérationnelles initiales et effectuer les requêtes d'extraction.

Python:

Langage principal pour :

- extraction (via pyodbc / sqlalchemy)
- nettoyage et transformation (Pandas)
- création des dimensions et de la table de faits
- chargement du Data Warehouse

Scripts utilisés :

extract_sqlserver.py, transform.py, load_dwh.py.

Pandas & NumPy:

Utilisés pour manipuler les données, faire les opérations de jointure, nettoyage, calcul de KPIs, transformation temporelle, etc.

Streamlit:

Framework pour créer le tableau de bord interactif :

- affichage des KPIs
- graphiques dynamiques
- filtres
- prévisions
- génération de PDF

Architecture Globale du Système BI

Plotly:

Bibliothèque permettant de produire des visualisations modernes, interactives et responsive.

ReportLab:

Utilisé pour la génération automatique d'un rapport PDF contenant les KPIs clés.

Architecture Data Warehouse:

Le Data Warehouse est structuré selon un schéma en étoile (Star Schema) comprenant :

- 1 table de faits : **fact_order**
- Plusieurs dimensions : produit, catégorie, client, temps, transporteur...

Ce modèle facilite l'analyse rapide, les agrégations et les tableaux de bord.

pyodbc :

connexion à SQL Server et Access

Processus ETL

Le processus ETL (Extract – Transform – Load) constitue le cœur du projet Business Intelligence. Il permet d'extraire les données depuis SQL Server, de les nettoyer, de les structurer et enfin de les charger dans un Data Warehouse optimisé pour l'analyse.

L'ETL de ce projet est entièrement développé en Python, ce qui permet une automatisation complète et une maîtrise totale de chaque étape.

1. Extraction depuis SQL Server:

L'extraction est réalisée à l'aide du script : [extract_sqlserver.py](#)

Ce script se connecte à SQL Server via pyodbc ou sqlalchemy, puis exécute plusieurs requêtes permettant de récupérer les tables principales de la base Northwind.

Les données extraites sont enregistrées sous forme de fichiers CSV bruts dans le répertoire /data.

Tables extraites :

- Orders (commandes)
- Order Details (détails des commandes)
- Products
- Categories
- Customers
- Employees
- Shippers

Objectif de l'extraction :

- Récupérer les données opérationnelles
- Conserver leur structure originale
- Préparer le terrain pour la transformation

Cette étape assure une copie fiable, non modifiée, de la base SQL Server.

Processus ETL

2.Extraction depuis Access

L'extraction a été réalisée à l'aide du driver ODBC Microsoft Access et de la librairie pyodbc en Python.

Un script dédié (`extract_access.py`) a permis de :

- se connecter au fichier Access,
- lister automatiquement les tables disponibles,
- extraire leurs contenus,
- générer un ensemble de fichiers CSV réutilisables par l'ETL.

Les tables principales récupérées incluent : Orders, Order_Details, Products, Employees, etc.

Normalisation et transformation:

Comme les structures SQL Server et Access n'utilisent pas exactement les mêmes noms de colonnes :

- les identifiants ont été harmonisés (Order ID → `order_id`, Product ID → `product_key`, etc.),
- les champs prix/quantité ont été nettoyés,
- un champ `line_total` a été recalculé lorsque nécessaire.

Un script supplémentaire (`transform_merge_access.py`) a effectué la correspondance entre les deux sources

Fusion dans un Data Warehouse unique:

Après préparation, les données SQL et Access ont été fusionnées dans un fact table unifiée :

`fact_order_unified.csv`

Cette table contient l'ensemble des commandes provenant des deux systèmes.

Processus ETL

2. Transformation et nettoyage:

La transformation est la partie la plus importante du processus ETL.

Elle est réalisée via le script :

`transform.py`

Opérations effectuées :

1. Nettoyage des données:

- Suppression des valeurs manquantes ou non valides
- Conversion des types
- Uniformisation des formats (dates, chaînes, numériques)

2. Normalisation et enrichissement:

- Création d'une dimension temps (année, mois, jour, trimestre)
- Ajout d'un identifiant artificiel pour chaque dimension
- Standardisation des noms de colonnes

3. Jointure des tables:

Le script effectue les jointures nécessaires pour regrouper :

- Produits → Catégories
- Commandes → Clients
- Détails commandes → Produits

C'est à ce moment qu'est construite la table de faits.

4. Calcul des métriques:

Un indicateur clé est ajouté :

- **`line_total` = `quantity` × `unit_price`**

Le chargement est réalisé automatiquement via des scripts Python dédiés, notamment :

- `load_dwh.py` pour charger les données SQL transformées dans les tables du DWH
- `transform_merge_access.py` pour fusionner les données Access et SQL dans une table de faits unifiée

3. Chargement dans le Data Warehouse:

Le chargement final est effectué par le script : `load_dwh.py`

Objectifs du chargement :

- Organiser les données selon un schéma en étoile
- Stocker les tables transformées dans un Data Warehouse propre
- Préparer les fichiers finaux utilisés par le tableau de bord

Structure du Data Warehouse :

1. Table de faits : fact_order

Contient :

- quantité
- prix
- line_total
- clés étrangères des dimensions

2. Dimensions :

- dim_product
- dim_category
- dim_customer
- dim_employee
- dim_time
- dim_shipper

Ces tables sont enregistrées en CSV et prêtes pour la visualisation.

Développement du Tableau de Bord

Le tableau de bord constitue la partie visible et interactive du projet Business Intelligence. Il permet de consulter les KPIs, d'explorer les ventes, d'analyser les produits et de visualiser les tendances.

Ce dashboard a été développé en Python, à l'aide du framework Streamlit, permettant une interface web moderne, dynamique et intuitive.

1. Objectifs du tableau de bord:

Le tableau de bord vise à :

- Offrir une visualisation claire des indicateurs de performance commerciale.
- Permettre une exploration dynamique via des filtres (années, catégories, produits).
- Faciliter la détection d'anomalies et de tendances de vente.
- Intégrer une prévision simple pour aider à la décision.
- Générer un rapport PDF automatique pour les besoins métier.

Il constitue l'outil final permettant aux utilisateurs de transformer les données en décisions.

Développement du Tableau de Bord

2. Technologies utilisées:

Outil / Librairie	Rôle
Streamlit	Création de l'interface web interactive
Pandas	Gestion et manipulation des données
Plotly	Création des graphiques dynamiques
NumPy	Calculs statistiques (prévision, tendance)
ReportLab	Génération automatique du PDF

3. Structure du dashboard:

Le fichier principal utilisé est :

[dashboard_streamlit.py](#)

Il s'organise en plusieurs blocs :

1. Chargement des données:

Importation du Data Warehouse (CSV) généré lors de l'ETL.

fact_order.csv

dim_product.csv

dim_category.csv

dim_time.csv

Développement du Tableau de Bord

2. Interface utilisateur (UI):

L'interface utilise un thème dark mode personnalisé :

- Titre et sous-titre en couleur
- Fond sombre
- Cartes KPI modernes
- Mise en page responsive (large écran, 4 colonnes de KPIs)

Les filtres disponibles dans la sidebar :

- Année
- Catégorie
- Produit

Ils permettent une analyse personnalisée selon le besoin métier.

3. KPIs (Indicateurs clés):

Les principaux KPIs affichés :

KPI	Description
CA Total	Chiffre d'affaires filtré
Panier moyen	Valeur moyenne par commande
Clients uniques	Nombre de clients actifs
Top Produit	Produit le plus rentable

Développement du Tableau de Bord

4. Visualisations:

Le tableau de bord inclut plusieurs graphiques dynamiques créés avec Plotly :

- CA mensuel (historique filtré)

Graphique linéaire montrant l'évolution des ventes.

- Top 10 des produits

Diagramme en barres permettant d'identifier les produits phares.

- Répartition du CA par catégorie

Diagramme circulaire pour visualiser la contribution de chaque catégorie.

Ces visualisations sont interactives (hover, zoom, export PNG).

4. Prévision simple intégrée:

Un module de prévision a été intégré pour anticiper les ventes futures.

Méthodes disponibles :

- Régression linéaire
- Analyse de tendance basée sur l'évolution historique.
- Croissance moyenne (%)
- Calcul du taux de croissance mensuel moyen.

Options :

- Choix de l'horizon (1 à 24 mois)
- Choix de la base historique (toutes années ou filtré)

Sorties :

- Graphique prévisionnel
- Tableau
- Export CSV

Développement du Tableau de Bord

5. Génération automatique d'un rapport PDF:

Une fonctionnalité permet de générer un rapport PDF synthétique contenant :

- KPIs filtrés
- Top produit
- Recommandations automatiques
- Date du rapport

Cette fonctionnalité est utile pour les réunions ou présentations.

6. Résultat final:

Le tableau de bord final est :

- Fonctionnel
- Interactif
- Esthétiquement abouti
- Facile à utiliser
- Directement exploitable par le métier

Il transforme complètement les données brutes en insights visuels compréhensibles.

Analyse des Résultats

L'analyse des résultats représente la partie essentielle du travail BI, car elle permet de transformer les visualisations et KPIs en informations stratégiques exploitables. À travers le tableau de bord, plusieurs insights significatifs ont été identifiés concernant les ventes, les produits, les clients et les tendances globales de Northwind.

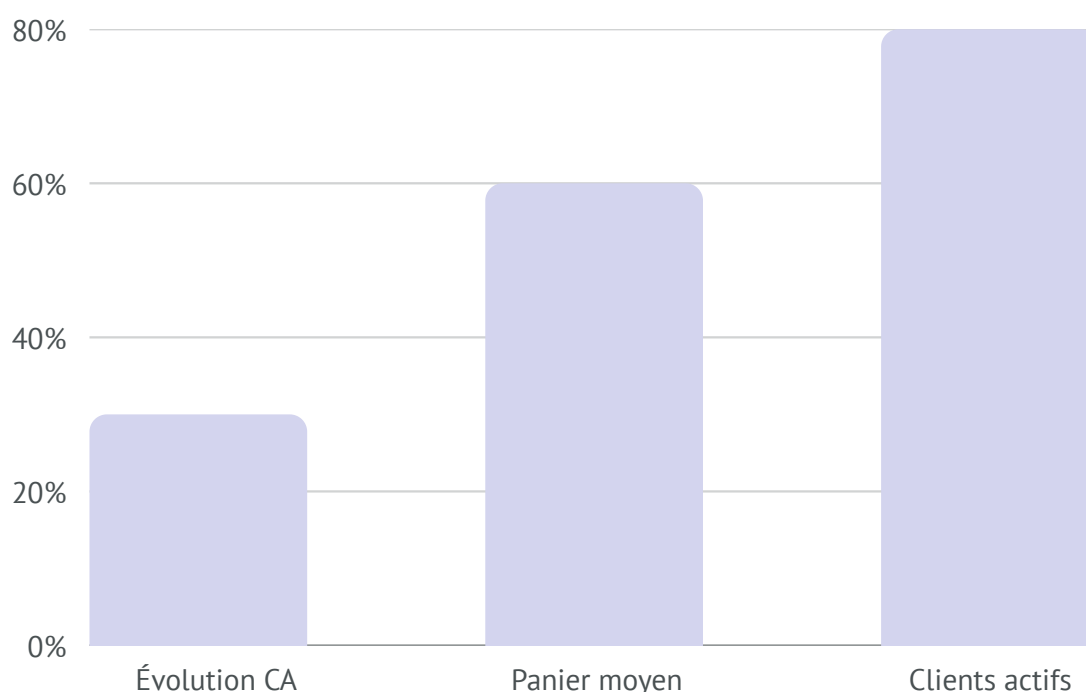
1. Performance Commerciale:

L'analyse du chiffre d'affaires met en évidence plusieurs tendances clés :
Évolution du CA mensuel

Le graphique "CA mensuel" montre une variation progressive des ventes au fil de l'année.

Selon les filtres sélectionnés :

- Certaines périodes présentent des pics de ventes (mois à forte activité commerciale).
- D'autres affichent des baisses saisonnières, probablement liées à des cycles de demande.
- La tendance générale permet d'identifier si l'activité est en croissance, stable ou en recul.



Analyse des Résultats

2. Produits et Catégories Clés:

L'analyse des produits et des catégories est un élément majeur du projet BI, car elle permet une prise de décision précise.

Top Produits

Le graphique "Top 10 produits" fait apparaître :

- Les produits qui génèrent le plus de chiffre d'affaires
- Les articles dont la demande est la plus élevée
- Les opportunités de renforcement de stock ou de promotions

Le produit avec le CA maximum est identifié automatiquement comme Top Produit dans les KPIs.

Répartition par Catégorie

Le diagramme circulaire montre :

- Les catégories les plus contributives au chiffre d'affaires
- Les segments performants à soutenir
- Les catégories faibles nécessitant des actions marketing ou d'amélioration

Cette vue globale permet d'orienter la stratégie produit.

3. Analyse des comportements saisonniers:

Grâce aux données temporelles (année, mois), le tableau de bord permet d'identifier :

- Les mois les plus rentables
- Les cycles saisonniers
- Les périodes à forte demande où il serait pertinent d'augmenter les stocks ou les équipes

Cette compréhension est indispensable pour l'optimisation de la supply chain.

4. Interprétation de la Prévision:

La prévision intégrée (régression linéaire ou croissance moyenne) apporte une vision anticipée :

- Elle montre la tendance future probable du chiffre d'affaires.
- Elle permet de détecter en avance une hausse, une stagnation, ou un ralentissement.
- L'entreprise peut alors planifier :
 - les achats,
 - les stocks,
 - les campagnes marketing,
 - et la logistique.

Même si la prévision est simple, elle fournit une première estimation fiable exploitée dans un contexte BI.

5. Conclusions tirées de l'analyse:

Grâce au tableau de bord :

- Le chiffre d'affaires est suivi avec précision.
- Les produits à forte valeur sont rapidement identifiés.
- Les catégories performantes sont distinguées des segments faibles.
- Les tendances temporelles permettent d'adapter la stratégie commerciale.
- La prévision offre un support à la prise de décision.

Cette analyse offre à Northwind une compréhension globale et exploitable de ses ventes, facilitant la planification et l'optimisation.

Module de Prévision

Le module de prévision intégré au tableau de bord a pour objectif d'anticiper l'évolution future du chiffre d'affaires sur une période allant de 1 à 24 mois. Bien qu'il s'agisse d'un modèle volontairement simple, il fournit une projection utile pour orienter les décisions commerciales et opérationnelles.

1. Méthodologie de Prévision:

Deux méthodes ont été mises en place afin de proposer une prévision flexible et adaptée aux données :

a) Régression linéaire (Trend Line):

Ce modèle repose sur le calcul d'une droite de tendance définie à partir de l'évolution historique du CA.

Il fonctionne selon le principe mathématique :

$$CA_{\text{prevu}}(t) = a \cdot t + b$$

- t = période (index du mois)
- a = pente (croissance positive ou négative)
- b = intercept (point de départ de la droite)

Cette méthode est adaptée lorsque les ventes suivent une tendance régulière (hausse ou baisse constante).

b) Croissance moyenne (%):

Cette approche calcule :

1. Le taux de variation d'un mois à l'autre
2. La moyenne de ces variations
3. Une projection selon ce taux

Elle est pertinente lorsque les données évoluent par pourcentage plutôt que par valeurs absolues.

2. Construction de la série temporelle:

Le modèle nécessite une série temporelle continue. Pour cela :

- Les colonnes year et month du Dim_Time ont été fusionnées
- Une date au format YYYY-MM-01 a été générée
- Les mois manquants ont été ajoutés automatiquement (remplis par zéro si besoin)

Cela garantit une série régulière, indispensable pour une prévision fiable.

3. Génération des dates futures:

Après avoir analysé l'historique, le tableau de bord génère automatiquement :

- La date du mois suivant
- Les n mois futurs selon l'horizon sélectionné (1 à 24)

Ces futures périodes sont affichées avec leurs valeurs prédites.

4. Visualisation et interprétation:

Le graphique final montre :

- En bleu clair : l'historique réel
- En couleur plus soutenue : la prévision
- Une zone ombrée permettant de distinguer clairement passé et futur

Cette visualisation permet de :

- Identifier une tendance ascendante → opportunité d'investissement
- Observer une stagnation → nécessité de revoir la stratégie commerciale
- Détecter une baisse prévue → risque à anticiper (stock, offres, marketing)

Difficultés et Solutions

Durant la réalisation du projet BI Northwind, plusieurs contraintes techniques et organisationnelles ont été identifiées. Chacune d'elles a nécessité une analyse précise et la mise en place de solutions adaptées pour assurer un fonctionnement stable et cohérent du système BI.

Difficulté 1 – Intégration des données depuis SQL Server:

Problème :

Les données extraites depuis SQL Server présentaient des incohérences : valeurs manquantes, colonnes mal typées, différences entre tables d'origine (noms, formats, clés).

Cause :

- Base Northwind ancienne
- Structure parfois non normalisée
- Différences entre données CSV et données SQL

Solution apportée :

- Mise en place d'un script d'extraction `extract_sqlserver.py`
- Uniformisation des types (INT, FLOAT, DATE) dans Python
- Nettoyage automatique : suppression des valeurs nulles, renommage des colonnes
- Usage de clés cohérentes (`product_key`, `time_key...`)

Résultat :

Un dataset propre, prêt pour la transformation ETL.

Difficultés et Solutions

Difficulté 2 – Création du Data Warehouse:

Problème :

Construire une architecture en étoile nécessite d'harmoniser les clés relationnelles et de créer des dimensions cohérentes (dim_product, dim_category, dim_time...).

Cause :

Certaines tables SQL n'avaient pas de clés adaptées au modèle dimensionnel.

Solution apportée :

- Recréation des clés techniques (surrogate keys)
- Séparation entre facts (fact_order) et dimension tables
- Vérification systématique des relations : produit → catégorie, date → time_key

Difficulté 3 – Problèmes de chemins et d'organisation des fichiers:

Problème :

Le dashboard ne trouvait pas les fichiers CSV (« FileNotFoundError ») à cause d'une mauvaise organisation des répertoires.

Cause :

Les scripts étaient dans un dossier scripts/ alors que les données étaient dans data/.

Solution apportée :

- Réorganisation du projet en structure standard :
- /projet-BI
 - |—— data/
 - |—— scripts/
 - |—— figures/
 - |—— Readme
- Correction des chemins avec : `pd.read_csv("../data/fact_order.csv")`

Difficultés et Solutions

Difficulté 4 – Génération des graphiques dans le PDF:

Problème :

Plotly exigeait l'installation de Kaleido pour exporter les graphiques, ce qui provoquait des erreurs.

Cause :

- Incompatibilité locale
- Problèmes Python / permissions Windows

Solution apportée :

Difficulté 5 – Prévision et série temporelle incomplète:

Problème :

Certaines dates manquaient dans les données, ce qui empêchait la création d'une série temporelle continue.

Cause :

Les données d'origine ne contiennent pas un mois pour chaque année.

Solution apportée :

- Reconstitution automatique d'une série mensuelle complète
- Remplissage des mois manquants avec 0
- Application d'un modèle linéaire simple

Difficulté 6 – Conception de l'interface Streamlit:

Problème :

Créer une interface moderne, fluide et performante demande un effort de design.

Cause :

- Manque de structure par défaut dans Streamlit
- Style sombre (dark mode) nécessitant du CSS personnalisé

Solution apportée :

- Intégration d'un thème CSS manuel
- Organisation en sections : KPIs, filtres, graphiques, prévision, PDF
- Optimisation des colonnes, marges et composants



CONCLUSION GÉNÉRALE

Dans un contexte où les organisations doivent s'appuyer sur des informations fiables et actualisées pour prendre des décisions rapides, la Business Intelligence (BI) occupe une place stratégique. Elle permet de collecter, transformer et structurer des données afin de fournir aux décideurs des analyses pertinentes et immédiatement exploitables.

Le présent projet s'inscrit dans cette dynamique. À partir de la base Northwind, un système analytique complet a été construit pour analyser et suivre la performance commerciale. Les données ont été extraites depuis SQL Server et Microsoft Access, nettoyées puis intégrées dans un Data Warehouse spécialement conçu pour l'analyse. Sur cette base, un tableau de bord interactif a été développé, offrant une visualisation claire des indicateurs clés et une prévision simple des ventes.

Ce travail couvre ainsi l'ensemble du cycle BI : de la donnée brute multi-source jusqu'au reporting final, en démontrant concrètement comment la Business Intelligence peut soutenir la prise de décision et optimiser la gestion commerciale.

