

Projet Business Intelligence Northwind

1) Présentation du projet

Objectif

Construire une solution Business Intelligence complète basée sur la base Northwind (SQL Server + Access "excel") incluant :

- ✓ ETL complet en Python
- ✓ Data Warehouse (Star Schema)
- ✓ Visualisations : KPI, graphes for analysis
- ✓ Tableau de bord BI
- ✓ Vidéo démonstrative

Outils utilisés

- Python (Pandas, SQLAlchemy, PyODBC, Plotly, Geopandas...)
- SQL Server ou SQLite (selon les facilités)
- Jupyter Notebook

2) Arborescence du projet (conforme au prof)

```
project_folder/
|
|   └── README.md
|
|   # Description du projet, installation et exécution.
|
|   └── requirements.txt
|
|   # Liste des dépendances Python (pandas, pyodbc, sqlalchemy, etc.)
|
|   └── data/
|
|       └── access.accdb
|
|           # Base de données source Northwind (Access) or Excel
|
|           └── sqlserver.sql
|
|               # Script SQL pour la base de données cible/connexion
```

```
|   └── raw/
|   |   # Étape 1: Extraction → Données brutes des sources
|   └── processed/
|   |   # Zone intermédiaire → Données après le nettoyage initial
|   └── warehouse/
|   |   # Étape 3: Chargement → Entrepôt de Données Final
|   |   |   # Schéma détaillé du modèle en étoile (DWH)
|   |   |   # Script SQL pour créer les tables Dim/Fact
|
|   └── scripts/
|       # Scripts Python de PRODUCTION (code réutilisable et automatisable)
|       └── database_connect.py
|           # Centralisation des connexions (Access, DWH)
|       └── extract_data.py
|           # Logique d'extraction (E du ETL)
|       └── transform_data.py
|           # Logique de nettoyage et transformation (T du ETL)
|       └── load_dwh.py
|           # Logique de chargement (L du ETL)
|       └── etl_main.py
|           # gère l'exécution complète
|
|   └── notebooks/
|       # Notebooks Jupyter pour l'exploration, le développement et le dashboard
|       └── explor.ipynb (bdlo le nom)
|           # EDA: Analyse exploratoire des données sources
```

```
|   └── ELT_dev.ipynb (bdlo le nom)
|   |   # Développement/Test interactif des fonctions ETL
|   ├── modelling.ipynb (bdlo le nom)
|   |   # Construction et prototypage du modèle (Star Schema)
|   ├── verification.ipynb (bdlo le nom)
|   |   # Tests de cohérence et de qualité des données (DWH)
|   └── dashboard_analysis.ipynb (bdlo le nom)
|       # Visualisation des KPI, y compris le Graphe OLAP 3D
|
|   └── reports/
|       └── rapport_BI.pdf
|           # Rapport final obligatoire
|
|   └── figures/
|       # Images et Graphiques pour le Rapport et la Présentation
|       └── kpi_overview.png
|           # Graphique des KPI principaux
|       └── 3d_scatter.png
|           # Le graphique OLAP 3D (Date, Client, Employé)
└── video/
    └── presentation.mp4
        # Démo écran + voix
```

3) Contenu du Data Warehouse (DWH)

Dimensions (tables Dim)

- **DimDate**
- **DimClient**
- **DimEmployee**

Fact table : FactSales

4) Processus ETL (Extraction → Transformation → Load)

- **extract_data.py** (bdlo le nom)
 - Connecter à Access (.accdb)
 - Connecter à SQL Server
 - Extraire les tables : Orders, Customers, Employees, Order Details, Products
 - Sauvegarder dans data/raw/
- **transform_data.py** (bdlo le nom)
 - Nettoyer les textes
 - Harmoniser villes et pays
 - Calculer total_amount
 - Générer : DimDate, DimClient, DimEmployee, DimProduct, FactSales
 - Sauvegarder dans data/cleaned/
- **load_dwh.py** (bdlo le nom)
 - Charger cleaned → staging
 - Insérer staging → dwh
 - Créer fichiers finaux CSV/Parquet dans data/dwh/

5) Notebooks

- **exploration.ipynb**(bdlo le nom)
 - Afficher un aperçu des tables raw
 - Statistiques simples
- **modelling.ipynb**(bdlo le nom)
 - Explication du Star Schema
 - Création des dimensions et table de faits
 - schéma_etoile.png
- **dashboard.ipynb**(bdlo le nom)
 - Calcul des KPI
 - Graphique 3D
 - Carte géographique
 - Heatmaps
- **verification.ipynb**(bdlo le nom)
 - Vérifier les valeurs nulles
 - Vérifier les doublons
 - Comparer nombre de lignes raw vs DWH

6) Visualisations

- **Graphe 3D(OLAP) (visualize_3d.py)**

- Axes :
 - X = Date (year/month)
 - Y = Clients
 - Z = Employés
- Couleur = Total ventes
- Librairie : Plotly

- **livrée/non livrée graphe**

- **KPI principaux**

- Chiffre d'affaires total
- Nombre de commandes
- Commande livrée/non livrée

- **Autres graphes nécessaire**

9) requirements.txt

pandas

numpy

sqlalchemy

pyodbc

matplotlib

plotly

geopandas

jupyter

seaborn

10) README.md (structure prête)

```
# Northwind Business Intelligence Project
```

```
## Description
```

Projet BI basé sur Northwind utilisant Python (ETL), Data Warehouse et Dashboard analytique.

Structure du projet

(arborescence)

Comment exécuter

1. Installer requirements
2. Lancer scripts d'extraction
3. Lancer transformation
4. Lancer load
5. Ouvrir notebooks pour visualisation