

**Université des Sciences et de la Technologie
Houari Boumediene (USTHB)**

Faculté d'Informatique

Master 2 Big Data

Conception d'un entrepot de données dans power bi

Module : Business Intelligence (BI)

Réalisé par :

IHADDADENE Chakib 181831091825

Guide Complet : Power BI Desktop

14 décembre 2025

Table des matières

1	Installation de Power BI Desktop	3
1.1	Prérequis Système	3
1.2	Étapes d'Installation	3
1.2.1	Méthode 2 : Téléchargement Direct	3
2	Premier Lancement et Configuration	4
2.1	Configuration Initiale	4
2.2	Interface Utilisateur	4
3	Architecture de l'Entrepôt de Données Northwind	6
3.1	Modèle en Étoile (Star Schema)	6
3.2	Composants de l'Architecture	7
3.3	Flux de Données	7
4	Création du Projet Northwind	7
4.1	Structure du Projet	7
4.2	Importation des Données Sources	7
4.2.1	Connexion à SQL Server	7
4.2.2	Importation des Fichiers Excel	8
4.3	Transformation avec Power Query	8
4.3.1	Accès à l'Éditeur Power Query	8
4.3.2	Renommage des Requêtes	9
5	Implémentation du Modèle de Données	9
5.1	Création des Dimensions	9
5.1.1	Dim_Employee	9
5.1.2	Dim_Client et Dim_Temps	9
5.2	Création de la Table de Faits	9
5.2.1	TF_Commande	9
5.3	Établissement des Relations	9

6	Comparaison Power BI vs Talend pour l'ETL	10
6.1	Tableau Comparatif	10
6.2	Points Forts par Outil	11
6.2.1	Power BI (Power Query)	11
6.2.2	Talend	11
6.3	Exemple de Code Comparé	11
6.4	Recommandation pour le Projet Northwind	12
7	Visualisation avec Python dans Power BI	12
7.1	Configuration de Python pour Power BI	12
7.1.1	Installation des Prérequis	12
7.1.2	Création d'un Visuel Python	12
7.1.3	Exemple Complet : Analyse des Commandes livrées par territoire . .	13
7.2	Exemple 2 : Top 5 des Territoires par Performance	13
7.2.1	Préparation des Données	13
7.2.2	Script Python pour l'Analyse Territoriale	13
7.2.3	Résultat Attendue	14
7.2.4	Explications du Code	14
7.2.5	Exécution et Résultats	14
7.3	Bonnes Pratiques pour les Visuels Python	15
7.4	Avantages du Visuel Python	15
8	Alternative : Python Pur sans Power BI	15
8.1	Approche Directe avec Jupyter Notebooks	15
8.1.1	Principe de l'Approche	15
8.1.2	Exemple de Code Python Autonome	16
8.1.3	Avantages de l'Approche Python Pure	17
8.1.4	Comparaison des Deux Approches	17
8.1.5	Scénarios d'Utilisation Recommandés	18
8.1.6	Workflow Hybride Recommandé	18
8.1.7	Implémentation Concrète du Projet Northwind	18
8.1.8	Résultat : un des graphes Python Autonome	19
8.2	Dashboard Final Northwind	21
9	Résolution des Problèmes Courants	21
9.1	Problèmes d'Installation	21
9.2	Problèmes de Connexion	21
10	Conclusion	22

Téléchargement Rapide

<https://aka.ms/pbidesktopstore>

1 Installation de Power BI Desktop

1.1 Prérequis Système

Composant	Configuration minimale
Système d'exploitation	Windows 10 ou 11 (64-bit)
Processeur	1 GHz ou plus rapide
Mémoire RAM	4 GB (8 GB recommandé)
Espace disque	2 GB d'espace libre
Résolution écran	1440x900 ou supérieure

TABLE 1 – Configuration système requise

1.2 Étapes d'Installation

1.2.1 Méthode 2 : Téléchargement Direct

1. Rendez-vous sur : <https://aka.ms/pbidesktopstore>
2. Cliquez sur "**Télécharger gratuitement**"
3. Exécutez le fichier `PBIDesktopSetup.exe`
4. Suivez l'assistant d'installation
5. Acceptez les conditions d'utilisation

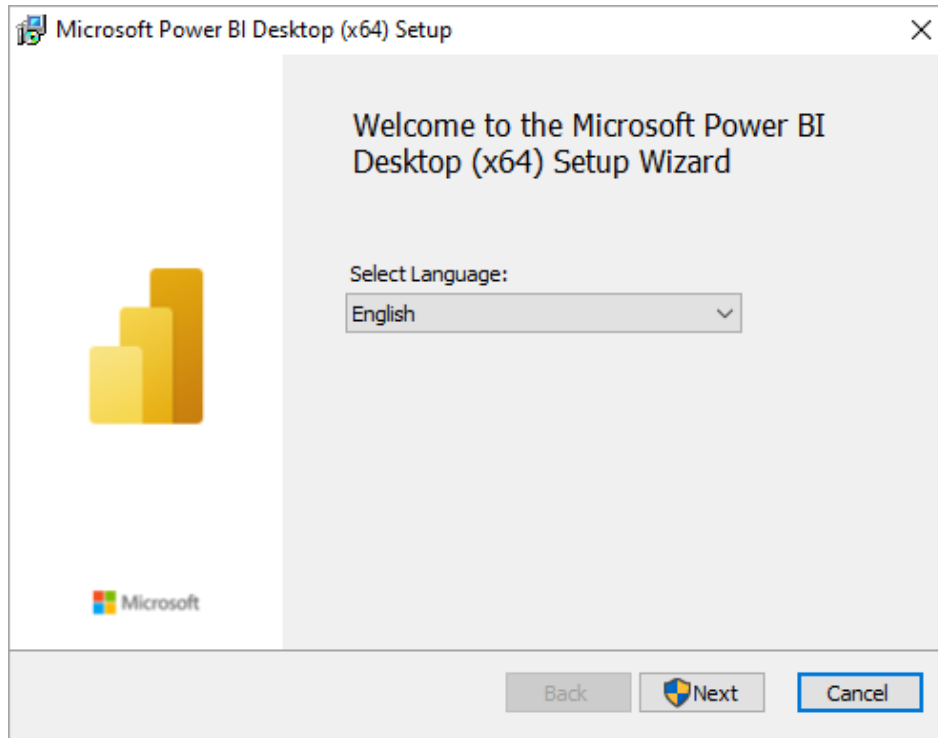


FIGURE 1 – Interface d'installation de power bi desktop

2 Premier Lancement et Configuration

2.1 Configuration Initiale

1. **Écran d'accueil** : Sélectionnez "Commencer"
2. **Connexion** : Connectez-vous avec votre compte Microsoft
3. **Thème** : Choisissez "Clair" ou "Sombre"
4. **Paramètres régionaux** : Sélectionnez "Français (France)"

2.2 Interface Utilisateur

Zone	Fonctionnalité
1. Ruban	Commandes principales (Fichier, Accueil, etc.)
2. Volet Visualisations	Types de graphiques disponibles
3. Volet Champs	Tables et colonnes de données
4. Zone de dessin	Construction des rapports
5. Volet Filtres	Application des filtres
6. Barre d'état	Informations et progression

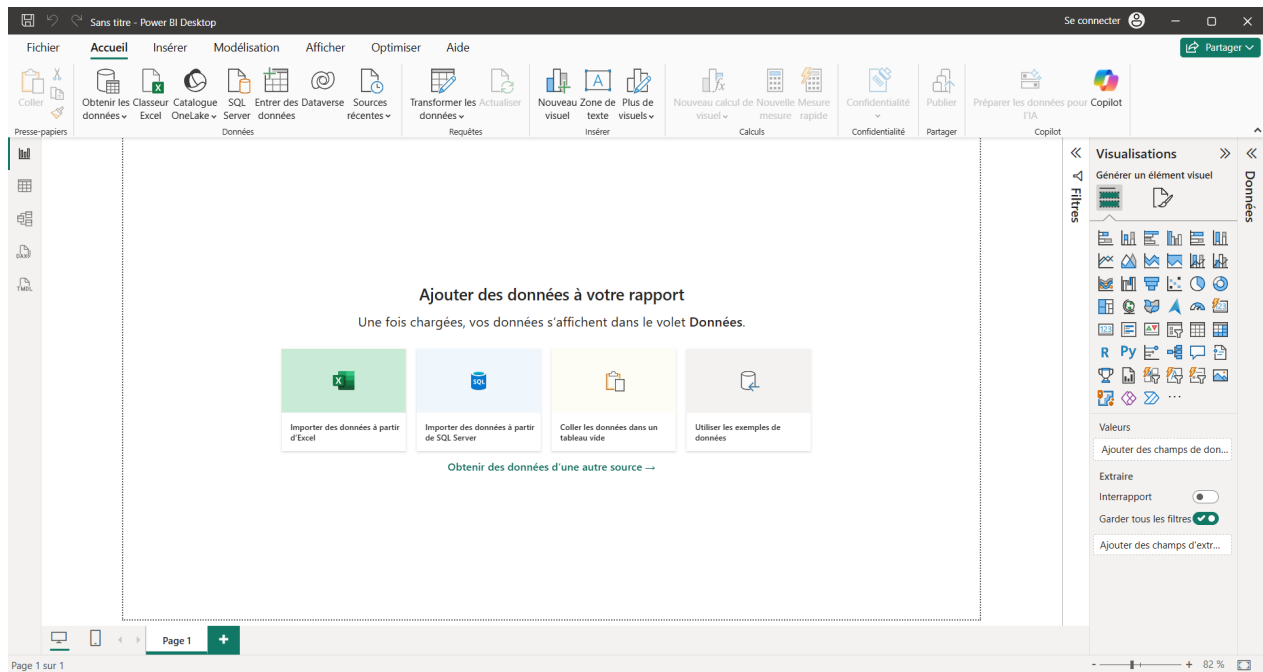


FIGURE 2 – Anatomie de l'interface Power BI Desktop

3 Architecture de l'Entrepôt de Données Northwind

3.1 Modèle en Étoile (Star Schema)

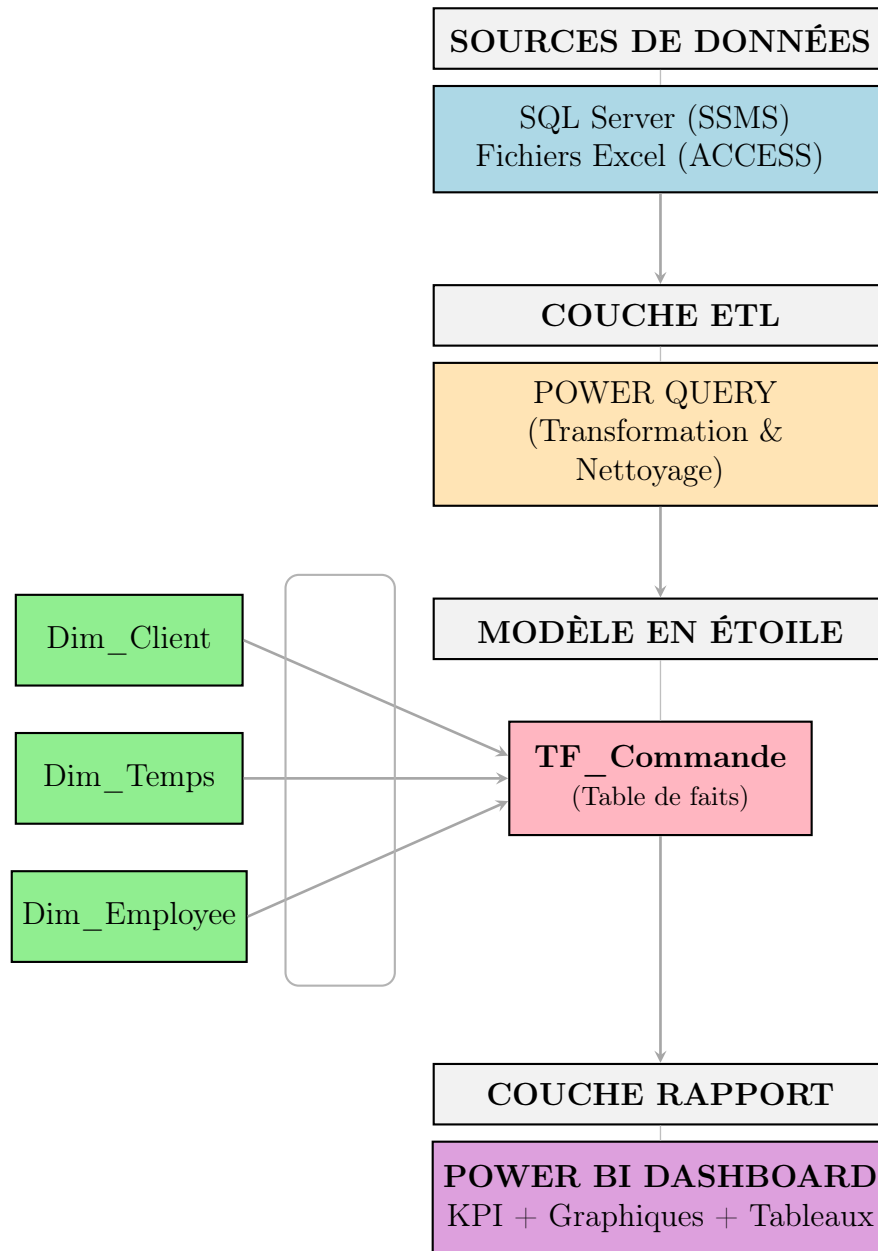


FIGURE 3 – Architecture complète de l'entrepôt de données Northwind avec modèle en étoile à 3 dimensions

3.2 Composants de l'Architecture

Composant	Description
Sources de Données	SQL Server (Northwind) + Fichiers Excel
Couche ETL	Power Query pour l'extraction et transformation
Tables Dimensions	Dim_Employee, Dim_Client, Dim_Temps
Table de Faits	TF_Commande
Couche Présentation	Rapports Power BI + Dashboard

TABLE 2 – Composants de l'architecture de l'entrepôt

3.3 Flux de Données

1. **Extraction** : Données brutes depuis SSMS et Excel
2. **Transformation** : Nettoyage et standardisation via Power Query
3. **Chargement** : Création des dimensions et table de faits
4. **Modélisation** : Établissement des relations 1 :*
5. **Visualisation** : Création des rapports et dashboards

4 Création du Projet Northwind

4.1 Structure du Projet

1. **Créer un dossier projet** :

```
C:\Users\IHADDADENE Chakib\Documents\Power BI Projects\Northwind_DW
    data\                # Données sources
    scripts\             # Codes Power Query
    exports\             # Exports et rapports
```

2. **Lancer Power BI Desktop**
3. **Nouveau fichier** : Fichier → Nouveau
4. **Enregistrer** : Fichier → Enregistrer sous → "Northwind_DW.pbix"

4.2 Importation des Données Sources

4.2.1 Connexion à SQL Server

1. Cliquez sur "**Obtenir des données**" → **Base de données** → **SQL Server**
2. Entrez les informations de connexion :
 - Serveur : DESKTOP-VE01CEQ\SQLCHAKIB
 - Base de données : Northwind
 - Mode de connexion : **Importer** (recommandé)

3. Sélectionnez les tables nécessaires :

- Customers
- Employees
- EmployeeTerritories
- Orders
- Territories

4.2.2 Importation des Fichiers Excel

1. Cliquez sur "Obtenir des données" → Fichier → Excel
2. Naviguez vers : C:\Users\PC\Documents\M2 BIGDATA\tp bi\csv du dm
3. Sélectionnez les fichiers :
 - Customers.xlsx
 - Employees2.xlsx
 - Orders.xlsx

4.3 Transformation avec Power Query

4.3.1 Accès à l'Éditeur Power Query

- Méthode 1 : Accueil → Transformer les données
- Méthode 2 : Clic droit sur une requête → Éditer la requête

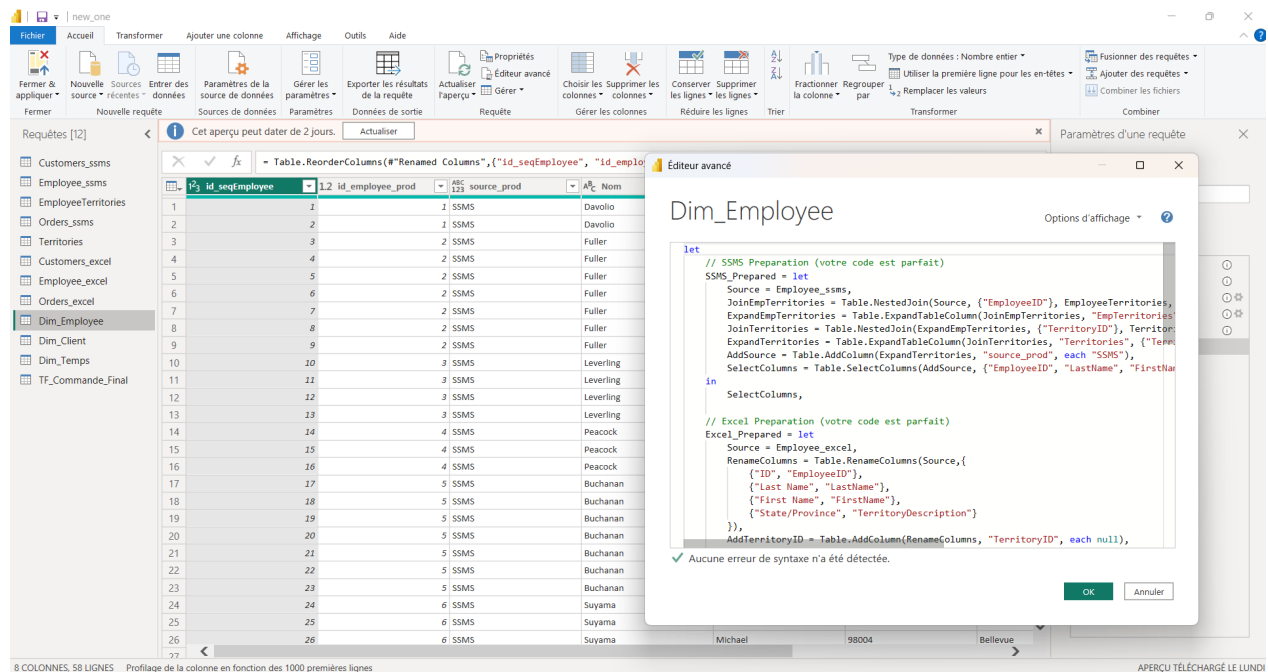


FIGURE 4 – Éditeur Power Query avec requêtes Northwind

4.3.2 Renommage des Requêtes

1. Dans le volet de navigation Power Query :
2. Renommez chaque requête avec le suffixe `_ssms` ou `_excel` :
 - Customers → Customers_ssms
 - Employees → Employee_ssms
 - etc.

5 Implémentation du Modèle de Données

5.1 Création des Dimensions

5.1.1 Dim_Employee

1. **Nouvelle requête** : Accueil → Nouvelle source → Requête vide
2. **Coller le code** : Collez le code M de Dim_Employee
3. **Renommer** : Renommez la requête en Dim_Employee
4. **Fermer et appliquer**

5.1.2 Dim_Client et Dim_Temps

- Répétez le processus pour chaque dimension
- Vérifiez les types de données
- Activez le chargement pour les tables dimensionnelles

5.2 Création de la Table de Faits

5.2.1 TF_Commande

1. Créez une nouvelle requête vide
2. Collez le code complet de TF_Commande
3. Renommez en TF_Commande
4. Vérifiez le nombre de lignes (devrait être 878)

5.3 Établissement des Relations

1. Retournez dans la vue **Modèle** de Power BI
2. Établissez les relations suivantes :
 - TF_Commande[id_temps] → Dim_Temps[id_temps]
 - TF_Commande[id_seqEmployee] → Dim_Employee[id_seqEmployee]
 - TF_Commande[id_seqClient] → Dim_Client[id_seqClient]
3. Vérifiez que toutes les relations sont **1 à plusieurs (*)**

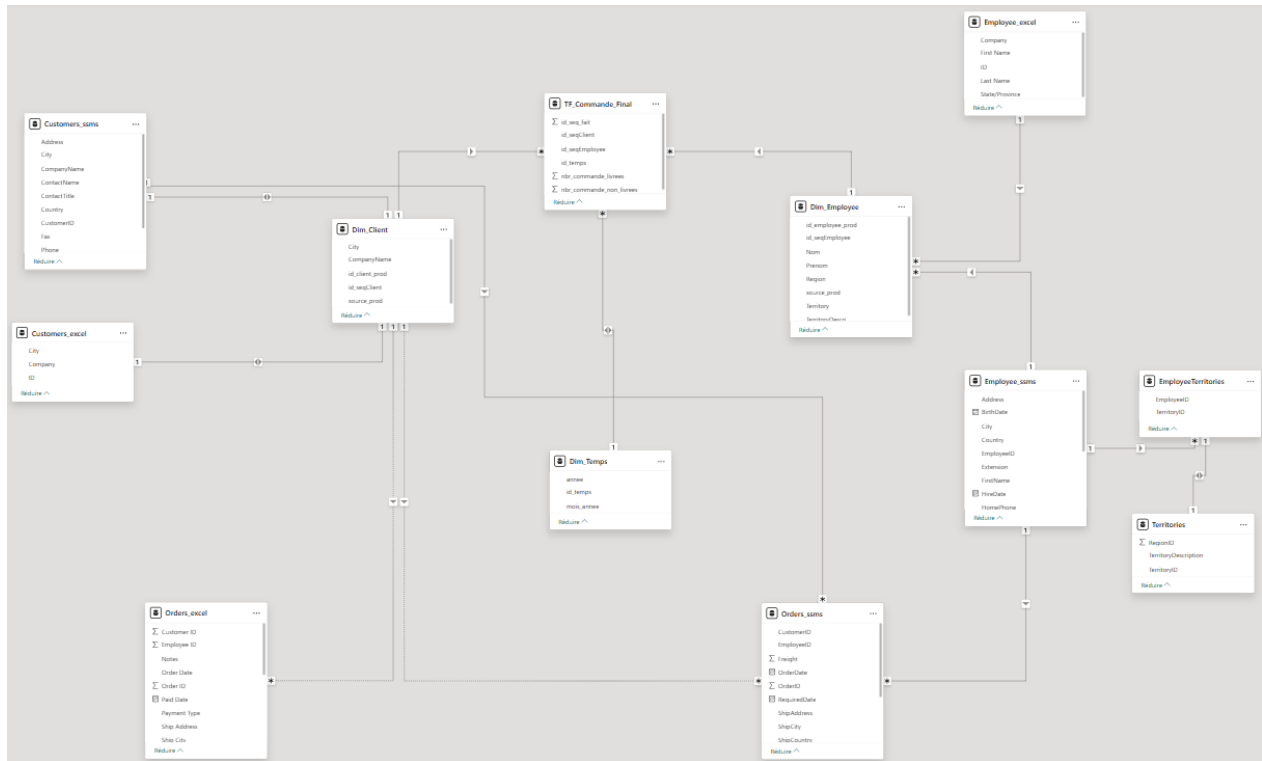


FIGURE 5 – Modèle de données final avec relations

6 Comparaison Power BI vs Talend pour l'ETL

6.1 Tableau Comparatif

Aspect	Power BI (Power Query)	Talend
Type d'outil	Outil de BI avec ETL intégré	Plateforme ETL/ELT dédiée
Complexité	Courbe d'apprentissage douce	Courbe d'apprentissage plus raide
Langage	M (Power Query)	Java + Composants visuels
Connexions	Connecteurs natifs limités	1000+ connecteurs
Transformation	Interface utilisateur + M	Interface graphique + code
Orchestration	Basic (rafraîchissements)	Avancée (workflows complexes)
Coût	Gratuit (Desktop)	Licence payante (Enterprise)
Performance	Optimisé pour données de taille moyenne	Scalable pour gros volumes
Maintenance	Facile (tout intégré)	Complexe (infrastructure séparée)
Utilisation projet Northwind	Parfait pour POC et démonstrations	Surdimensionné pour ce besoin

TABLE 3 – Comparaison Power BI vs Talend pour l'ETL

6.2 Points Forts par Outil

6.2.1 Power BI (Power Query)

- Intégration native avec les rapports
- Interface utilisateur intuitive
- Pas de contexte switching entre ETL et reporting
- Version Desktop gratuite
- Fonctionnalités de modélisation intégrées
- Support DAX pour calculs avancés

6.2.2 Talend

- Orchestration de workflows complexes
- Support de gros volumes de données
- Connecteurs très variés
- Qualité et gouvernance des données
- Métadonnées et documentation
- Intégration avec l'écosystème Big Data

6.3 Exemple de Code Comparé

Power Query (M)	Talend (Java)
<pre>let Source = Orders_ssms, #"Filtered Rows" = Table.SelectRows(Source, each [ShippedDate] <> null) in #"Filtered Rows"</pre>	<pre>// Composant tFilterRow row1.ShippedDate != null // Ou en Java dans tJavaRow if(input_row.ShippedDate != null) { output_row = input_row; }</pre>

TABLE 4 – Comparaison de syntaxe pour un filtre simple

6.4 Recommandation pour le Projet Northwind

Choix Optimal pour Northwind

Power BI avec Power Query est le meilleur choix pour ce projet car :

- Volume de données modeste (878 lignes)
- Besoin d'intégration directe avec les rapports
- Temps de développement réduit
- Aucune infrastructure supplémentaire nécessaire
- Coût nul avec Power BI Desktop gratuit
- Facilité de maintenance et de partage

7 Visualisation avec Python dans Power BI

7.1 Configuration de Python pour Power BI

7.1.1 Installation des Prérequis

1. **Installer Python** : Téléchargez Python 3.8+ depuis <https://python.org>
2. **Installer les packages** :

```
pip install pandas matplotlib
```
3. **Configurer Power BI** :
 - Fichier → Options et paramètres → Options
 - Scripting Python → Spécifier le chemin d'installation Python
 - Exemple : C:\Users\IHADDADENE Chakib\AppData\Local\Programs\Python\Python39

7.1.2 Création d'un Visuel Python

1. **Sélectionner le visuel Python** :
 - Dans le volet Visualisations, cliquez sur l'icône **Python**
 - Une zone de script vide apparaît sur la page
2. **Glisser les champs nécessaires** :
 - Depuis le volet Champs, glissez-déposez les colonnes dans le visuel Python
 - Exemple pour notre projet Northwind :
 - Depuis TF_Commande : `nbr_commande_livrees`
 - Depuis Dim_Employee : `TerritoryDescri`
3. **Écrire le script Python** :
 - Le champ de script s'active automatiquement
 - Power BI crée automatiquement un DataFrame `dataset` contenant vos données

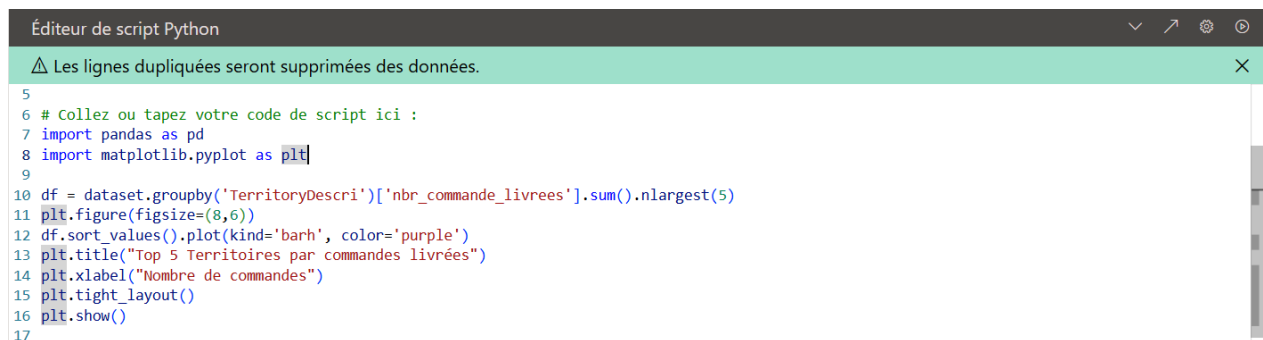


FIGURE 6 – Interface du visuel Python dans Power BI

7.1.3 Exemple Complet : Analyse des Commandes livrées par territoire

7.2 Exemple 2 : Top 5 des Territoires par Performance

7.2.1 Préparation des Données

1. Sélectionner le visuel Python dans le volet Visualisations
2. Glisser les champs nécessaires :
 - Depuis Dim_Employee : TerritoryDescri
 - Depuis TF_Commande : nbr_commande_livrees
3. Power BI crée automatiquement le DataFrame dataset

7.2.2 Script Python pour l'Analyse Territoriale

```
import pandas as pd
import matplotlib.pyplot as plt

df = dataset.groupby('TerritoryDescri')['nbr_commande_livrees'] \
    .sum() \
    .nlargest(5)

plt.figure(figsize=(8,6))
df.sort_values().plot(kind='barh', color='purple')
plt.title("Top 5 Territoires par commandes livrees")
plt.xlabel("Nombre de commandes")
plt.tight_layout()
plt.show()
```

7.2.3 Résultat Attendue

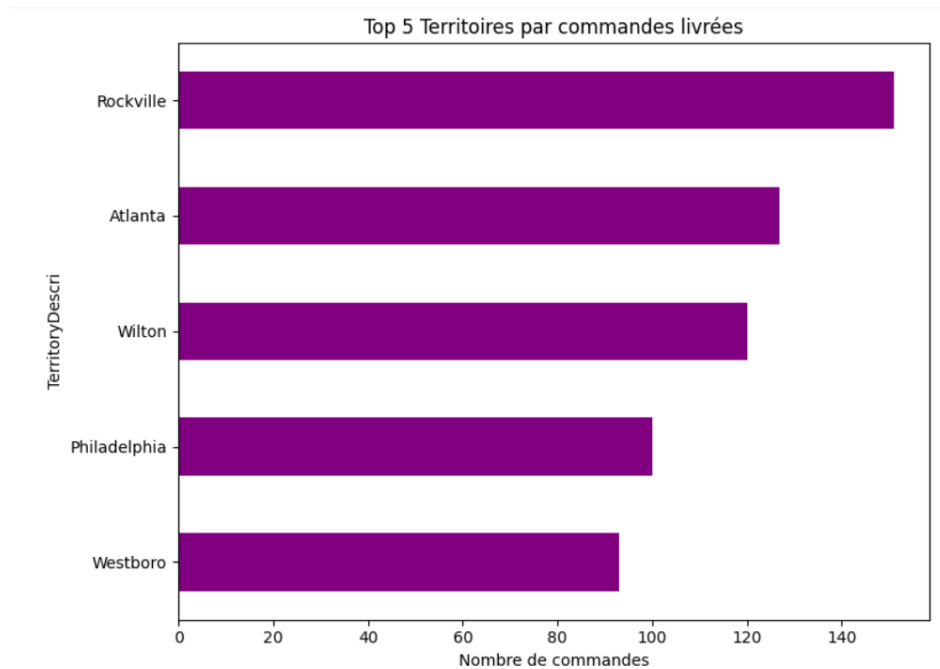


FIGURE 7 – Visualisation du top 5 des territoires générée par Python

7.2.4 Explications du Code

Ligne de code	Explication
<code>dataset.groupby()</code>	Regroupe les données par territoire
<code>nlargest(5)</code>	Sélectionne les 5 territoires avec le plus de commandes
<code>kind='barh'</code>	Crée un diagramme en barres horizontal
<code>color='purple'</code>	Définit la couleur des barres
<code>plt.title()</code>	Ajoute un titre au graphique
<code>plt.text()</code>	Ajoute les valeurs numériques sur les barres
<code>plt.grid()</code>	Ajoute une grille pour meilleure lisibilité
<code>plt.tight_layout()</code>	Optimise l'espace des éléments
<code>plt.show()</code>	Affiche le graphique dans Power BI

TABLE 5 – Explications détaillées du code Python

7.2.5 Exécution et Résultats

1. Cliquez sur "Exécuter le script" (icône play)
2. Attendez le chargement (première exécution peut prendre quelques secondes)
3. Visualisez le résultat : Le graphique apparaît dans le visuel
4. Interagissez : Utilisez les filtres Power BI pour mettre à jour dynamiquement le graphique Python

7.3 Bonnes Pratiques pour les Visuals Python

Pratique	Explication
Glisser d'abord les champs	Power BI crée <code>dataset</code> avec seulement ces colonnes
Vérifier les colonnes	Utilisez <code>print(dataset.columns)</code> pour debugger
Gérer les NaN	Utilisez <code>dropna()</code> ou <code>fillna()</code>
Optimiser les performances	Limitez le nombre de lignes avec des filtres
Utiliser des styles	<code>plt.style.use('seaborn')</code> pour de beaux graphiques
Sauvegarder les scripts	Copiez vos scripts dans un fichier <code>.py</code> externe

TABLE 6 – Bonnes pratiques pour les visuals Python

7.4 Avantages du Visuel Python

- **Intégration native** : Pas besoin d'exporter/importer
- **Interactivité** : Réagit aux filtres Power BI
- **Flexibilité** : Toute bibliothèque Python compatible
- **Mise à jour automatique** : Rafraîchit avec les données

Limitations à connaître

- **Performance** : Les scripts longs peuvent ralentir Power BI
- **Mémoire** : Limité par les ressources de Power BI
- **Dépendances** : Doivent être installées sur chaque machine
- **Version Python** : Doit correspondre à celle configurée

8 Alternative : Python Pur sans Power BI

8.1 Approche Directe avec Jupyter Notebooks

Alternative Complémentaire

Après avoir exploré l'intégration Python-Power BI, il est important de noter qu'on peut également travailler avec Python de manière autonome, sans passer par Power BI, en utilisant directement les tables exportées.

8.1.1 Principe de l'Approche

- **Export des tables** : Les tables dimensionnelles et de faits créées dans Power BI peuvent être exportées au format CSV
- **Chargement direct en Python** : Utilisation de pandas pour charger les fichiers CSV
- **Visualisation autonome** : Création de dashboards complets avec matplotlib/seaborn

— **Avantage majeur** : Indépendance totale de Power BI

8.1.2 Exemple de Code Python Autonome

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
    _DASHBOARD_NORTHWIND_-_ANALYSE_COMPLTE
    _6_Visualisations_Python_avec_les_vraies_donn_ees
"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# 1. CHARGEMENT DES DONNES
print("    _Chargement_des_tables_Northwind...")

# Chargement direct depuis les CSV exports
dim_client = pd.read_csv('data/Dim_Client.csv')
dim_employee = pd.read_csv('data/Dim_Employee.csv')
dim_temps = pd.read_csv('data/Dim_Temps.csv')
tf_commande = pd.read_csv('data/TF_Commande.csv')

# 2. FUSION DES TABLES (similaire au mod le Power BI)
print("    _Fusion_des_tables...")
df = pd.merge(tf_commande, dim_temps, on='id_temps', how='left')
df = pd.merge(df, dim_employee, on='id_seqEmployee', how='left')
df = pd.merge(df, dim_client, on='id_seqClient', how='left')

# 3. CRATION DES 6 VISUALISATIONS
print("    _G n_ration_des_6_visualisations...")

# Graphique 1: Volume des commandes par mois
commandes_par_mois = df.groupby('mois_annee').agg({
    'nbr_commande_livrees': 'sum',
    'nbr_commande_non_livrees': 'sum'
}).reset_index()

plt.figure(figsize=(14, 7))
# ... code pour le premier graphique ...

# Graphique 2: Top 10 clients
top_clients = df.groupby('CompanyName')['nbr_commande_livrees'] \
    .sum().nlargest(10)
```

```
plt.figure(figsize=(10, 6))
top_clients.sort_values().plot(kind='barh', color='skyblue')
plt.title("Top_10_Clients_-_Python_Pur")
plt.xlabel("Commandes_livrées")
plt.tight_layout()
plt.show()

# ... suite des 4 autres visualisations ...
```

8.1.3 Avantages de l'Approche Python Pure

Avantage	Description
Indépendance	Pas besoin de licence Power BI
Portabilité	Fonctionne sur tout OS (Windows, Linux, Mac)
Reproductibilité	Scripts versionnables avec Git
Automatisation	Intégration facile dans des pipelines CI/CD
Personnalisation	Contrôle total sur les graphiques
Libraries avancées	Accès à seaborn, plotly, bokeh, etc.
Export multiple	PDF, HTML, PNG, SVG, etc.
Cloud ready	Déploiement sur serveur Python

TABLE 7 – Avantages de l'approche Python pure

8.1.4 Comparaison des Deux Approches

Critère	Python dans Power BI	Python Pur (Notebook)
Interactivité	Excellente (filtres Power BI)	Limitée (sauf avec widgets)
Intégration	Native avec rapports Power BI	Séparé du dashboard
Apprentissage	Nécessite Power BI	Seulement Python
Portabilité	Windows seulement	Multi-plateforme
Licence	Gratuit mais limité	100% gratuit
Versionnement	Complexe (.pbix)	Facile (.py avec Git)
Automatisation	Limitée	Excellente (scripts)
Cloud	Power BI Service	Tout serveur Python

TABLE 8 – Comparaison des deux approches Python

8.1.5 Scénarios d'Utilisation Recommandés

Quand utiliser quelle approche ?

- **Python dans Power BI :**
 - Dashboards interactifs pour les utilisateurs métier
 - Intégration avec d'autres visuels Power BI
 - Partage via Power BI Service
 - Environnements où Power BI est déjà adopté
- **Python pur (Notebooks) :**
 - Analyses reproductibles pour data scientists
 - Versionnement avec Git
 - Déploiement sur serveurs Linux
 - Intégration dans pipelines de données
 - Besoin de bibliothèques spécifiques (scikit-learn, etc.)

8.1.6 Workflow Hybride Recommandé

Workflow Hybride Recommandé

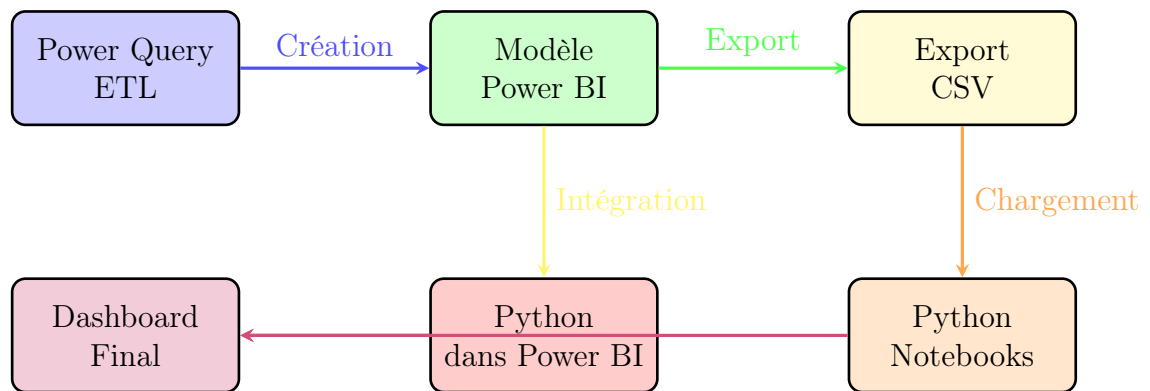


FIGURE 8 – Workflow hybride combinant Power BI et Python pur

8.1.7 Implémentation Concrète du Projet Northwind

1. Étape 1 - Export depuis Power BI :

- Dans Power BI Desktop, sélectionnez chaque table
- Clic droit → **Exporter les données** → **CSV**
- Sauvegardez dans le dossier data/

2. Étape 2 - Création du script Python :

- Créez un fichier `dashboard_northwind.py`
- Copiez le code fourni précédemment

— Installez les dépendances : `pip install -r requirements.txt`

3. Étape 3 - Exécution :

```
# Installation des dépendances
pip install pandas matplotlib numpy seaborn

# Exécution du dashboard
python dashboard_northwind.py

# Ou en mode Jupyter
jupyter notebook notebooks/dashboard_northwind.ipynb
```

4. Étape 4 - Versionnement :

```
# Initialisation Git
git init
git add .
git commit -m "Dashboard_Northwind_en_Python_pur"

# Push sur GitHub
git remote add origin https://github.com/votre-compte/northwind-python.git
git push -u origin main
```

8.1.8 Résultat : un des graphes Python Autonome

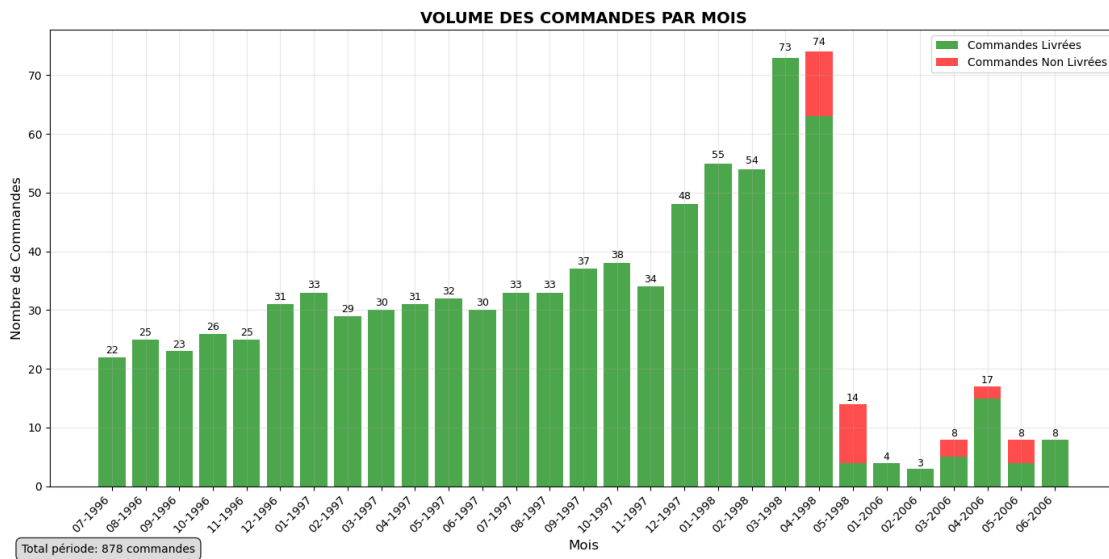


FIGURE 9 – Exemple d'un des graphes générés en Python pur

Conclusion sur les Deux Approches

Pour le projet Northwind, les deux approches sont valables :

- **Python dans Power BI** : Parfait pour une démonstration rapide avec interactivité
- **Python pur** : Idéal pour une solution reproductible, versionnable et déployable

Recommandation : Commencez avec Power BI pour le prototypage rapide, puis exportez vers Python pour la production et le versionnement.

8.2 Dashboard Final Northwind

Notre dashboard final intègre 6 visuals Python , les voici :



FIGURE 10 – Dashboard Northwind avec 6 visuals Python

9 Résolution des Problèmes Courants

9.1 Problèmes d'Installation

Problème	Solution
Échec d'installation Microsoft Store	Téléchargez la version hors Store
Erreur "Accès refusé"	Exécutez en tant qu'administrateur
Manque de dépendances	Installez .NET Framework 4.8
Antivirus bloque l'installation	Désactivez temporairement l'antivirus

9.2 Problèmes de Connexion

Problème	Solution
SQL Server inaccessible	Vérifiez le nom du serveur et les permissions
Fichier Excel non trouvé	Vérifiez le chemin et les permissions
Erreur d'authentification	Utilisez l'authentification Windows
Limite de mémoire	Augmentez la mémoire allouée à Power BI

10 Conclusion

Ce guide vous a accompagné de l'installation de Power BI Desktop à la création d'un projet d'entrepôt de données complet avec Northwind, incluant une analyse comparative avec Talend et deux approches Python. Vous disposez maintenant de toutes les ressources nécessaires pour :

- Installer et configurer Power BI Desktop
- Comprendre l'architecture d'un entrepôt de données
- Comparer Power BI et Talend pour l'ETL
- Créer un projet structuré Northwind
- Importer et transformer des données multi-sources
- Construire un modèle de données optimisé
- Utiliser Python dans Power BI pour des visualisations avancées
- Développer des dashboards Python autonomes sans Power BI
- Choisir l'approche adaptée à vos besoins
- Publier et partager vos analyses