

**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

13 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 Visuals Python	15
7.4 Avantages du Visuel Python	15
7.5 Dashboard Final Northwind	16
8 Résolution des Problèmes Courants	16
8.1 Problèmes d'Installation	16
8.2 Problèmes de Connexion	16
9 Conclusion	17

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

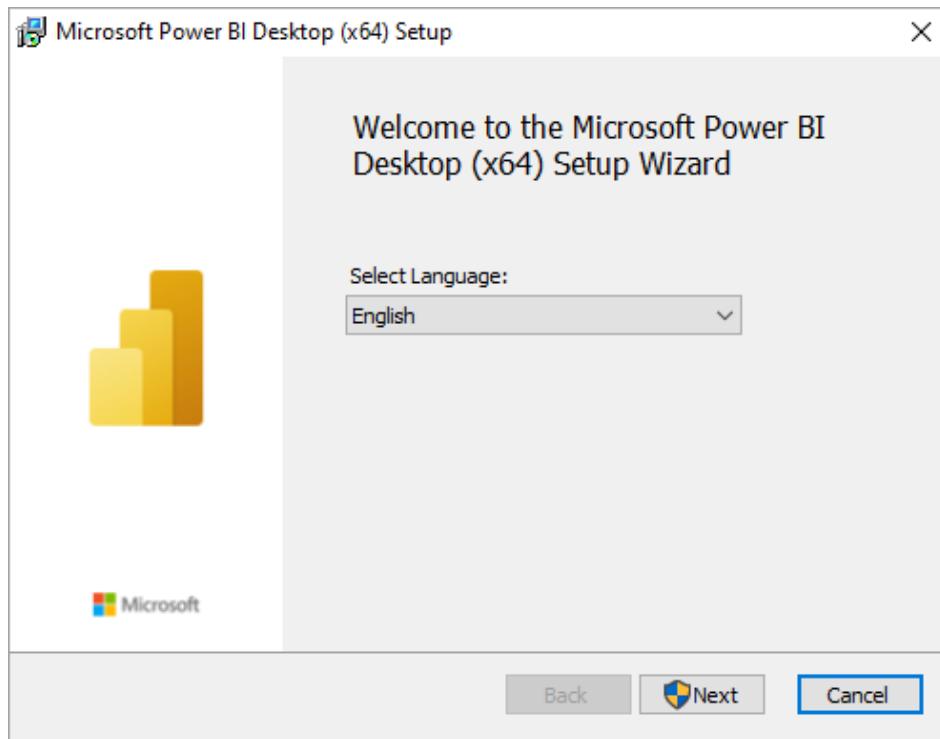


FIGURE 1 – Interface d'installation de power bi desktop

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

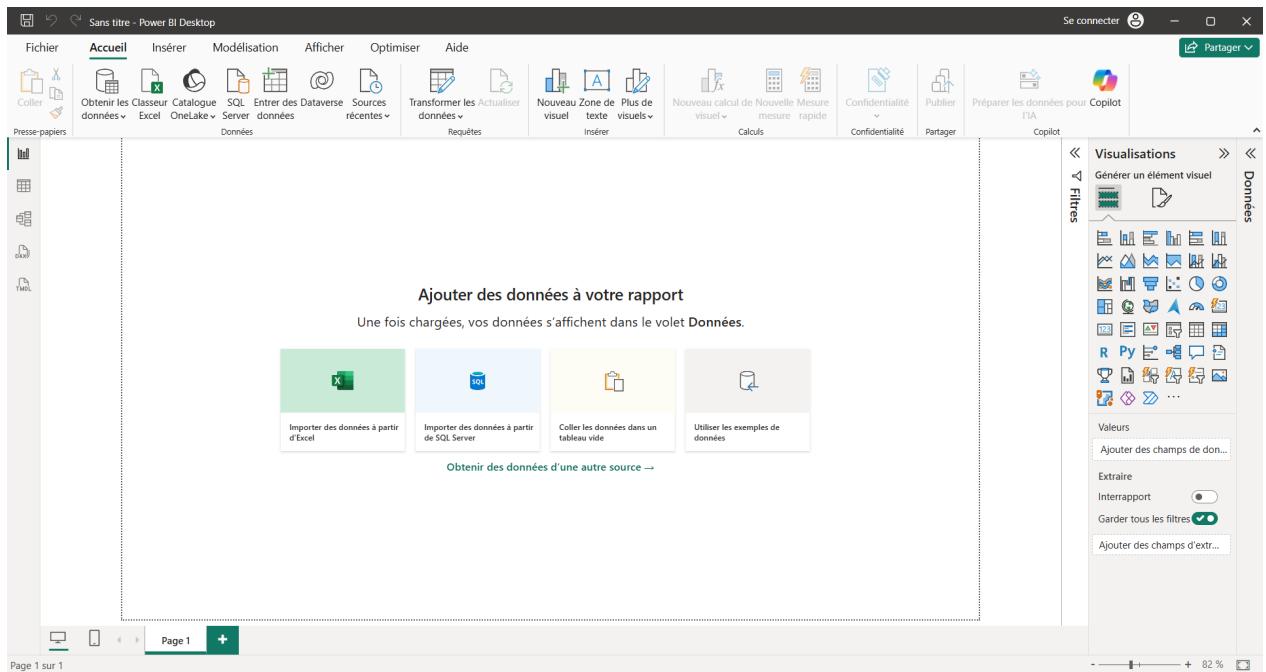


FIGURE 2 – Anatomie de l’interface Power BI Desktop

3 Architecture de l'Entrepôt de Données Northwind

3.1 Modèle en Étoile (Star Schema)

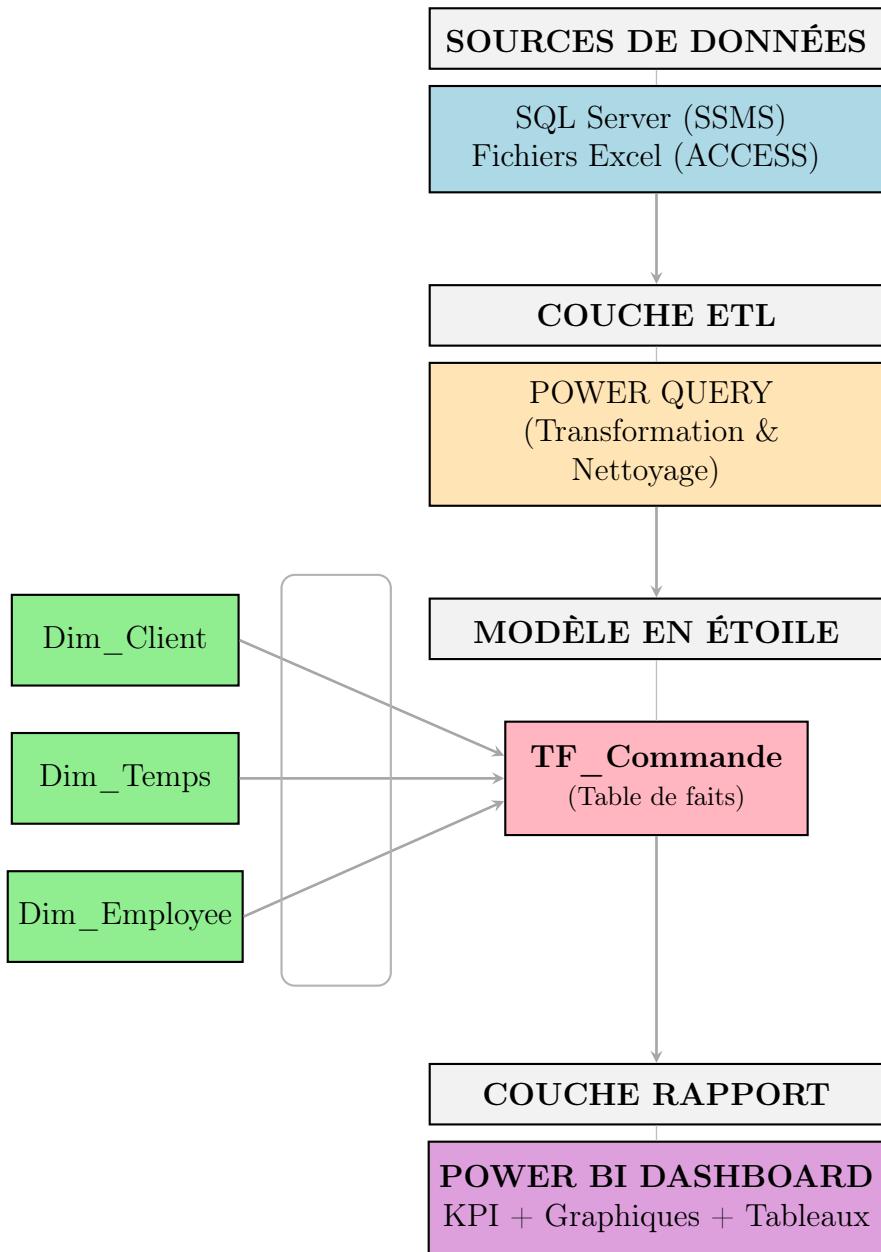


FIGURE 3 – Architecture complète de l'entrepôt de données Northwind avec modèle en étoile à 3 dimensions

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

Ficher Accueil Transformé Ajouter une colonne Affichage Outils Aide

Fermé & appliquer Nouvelle source... Entrer des données Paramètres de la source de données Gérer les paramètres Exporter les résultats de la requête Actualiser l'appareil Propriétés Editeur avancé Choisir les colonnes Supprimer les colonnes Gérer les colonnes Conserver les lignes Supprimer les lignes Réduire les lignes Trier Transformer Type de données : Nombre entier... Utiliser la première ligne pour les en-têtes... Fusionner des requêtes Ajouter des requêtes Combiner Combiner

Nouvelle requête Sources de données Paramètres Données de sortie Requête Gérer les colonnes Réduire les lignes Trier Transformer Paramètres d'une requête

Requêtes [12]

Customers_ssms Employee_ssms EmployeeTerritories Orders_ssms Territories Customers_excel Employee_excel Orders_excel Dim_Employee Dim_Client Dim_Temps TF_Commande_Final

Dim_Employee

Cet aperçu peut durer de 2 à 10 secondes. Actualiser

	id_seqEmployee	id_employee_prod	source_prod	Nom
1	1	1	SSMS	Davolio
2	2	1	SSMS	Davolio
3	3	2	SSMS	Fuller
4	4	2	SSMS	Fuller
5	5	2	SSMS	Fuller
6	6	2	SSMS	Fuller
7	7	2	SSMS	Fuller
8	8	2	SSMS	Fuller
9	9	2	SSMS	Fuller
10	10	3	SSMS	Leverling
11	11	3	SSMS	Leverling
12	12	3	SSMS	Leverling
13	13	3	SSMS	Leverling
14	14	4	SSMS	Peacock
15	15	4	SSMS	Peacock
16	16	4	SSMS	Peacock
17	17	5	SSMS	Buchanan
18	18	5	SSMS	Buchanan
19	19	5	SSMS	Buchanan
20	20	5	SSMS	Buchanan
21	21	5	SSMS	Buchanan
22	22	5	SSMS	Buchanan
23	23	5	SSMS	Buchanan
24	24	6	SSMS	Suyama
25	25	6	SSMS	Suyama
26	26	6	SSMS	Suyama

8 COLONNES, 58 LIGNES Profilage de la colonne en fonction des 1000 premières lignes

Editeur avancé

Dim_Employee Options d'affichage

```

let
    // SSMS Preparation (votre code est parfait)
    SSMS_Prepared = let
        Source = Employee_ssms,
        JoinEmpTerritories = Table.NestedJoin(Source, {"EmployeeID"}, EmployeeTerritories, ExpandTerritories = Table.ExpandTableColumn(JoinEmpTerritories, "EMTerritories"),
        JoinTerritories = Table.NestedJoin(ExpandEmpTerritories, {"TerritoryID"}, Territories, ExpandTerritories = Table.ExpandTableColumn(JoinTerritories, "Territories", {"TerritoryName"}),
        AddSource = Table.AddColumn(ExpandTerritories, "source_prod", each "SSMS"),
        SelectColumns = Table.SelectColumns(AddSource, {"EmployeeID", "LastName", "FirstName", "Title", "HireDate", "Address", "City", "Region", "PostalCode", "Country", "Phone", "Fax", "Email", "TerritoryID", "TerritoryDescription", "source_prod"})
    in
        SelectColumns,
    
    // Excel Preparation (votre code est parfait)
    Excel_Prepared = let
        Source = excel,
        RenameColumns = Table.RenameColumns(Source, {
            {"ID", "EmployeeID"}, {"Last Name", "LastName"}, {"First Name", "FirstName"}, {"State/Province", "TerritoryDescription"}},
        AddTerritoryID = Table.AddColumn(RenameColumns, "TerritoryID", each null),
        AddTerritoryID
    in
        AddTerritoryID

```

Aucune erreur de syntaxe n'a été détectée.

OK Annuler

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** (*)

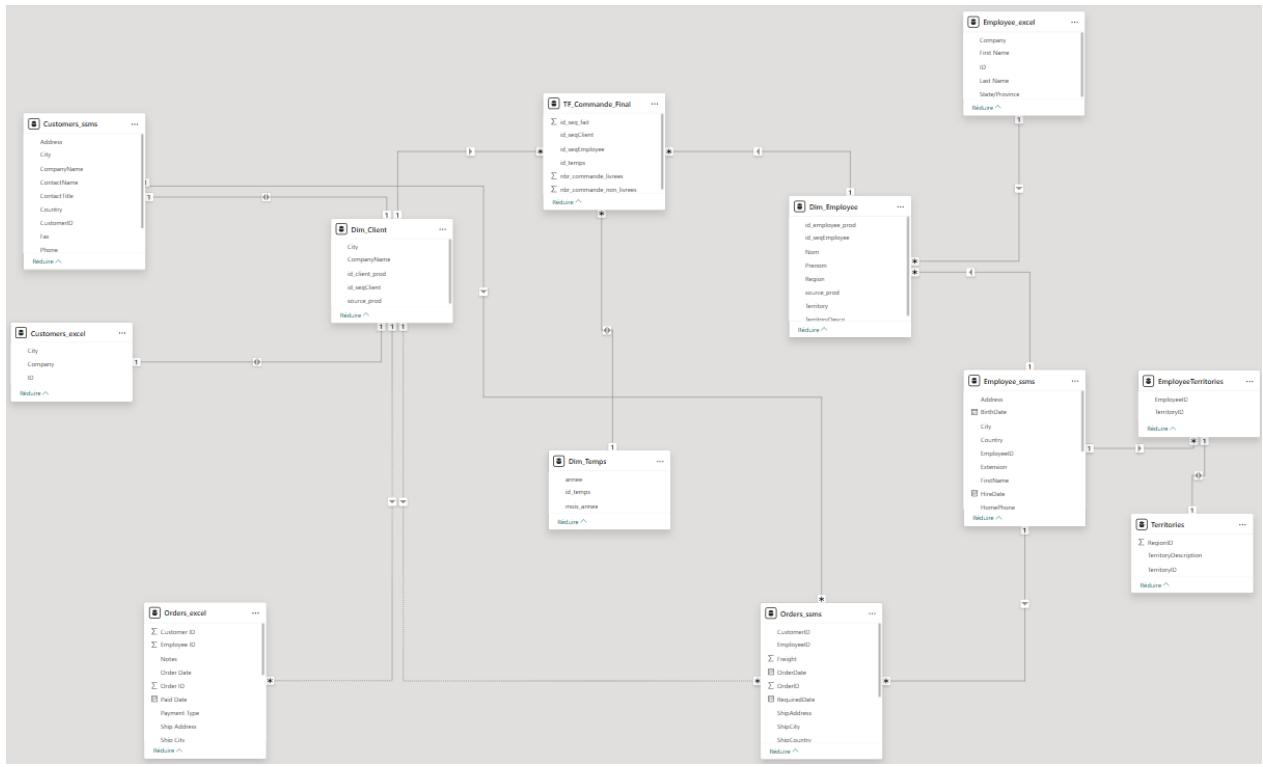


FIGURE 5 – Modèle de données final avec relations

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)	Complexé (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

```

Éditeur de script Python
⚠️ Les lignes dupliquées seront supprimées des données.

5
6 # Collez ou tapez votre code de script ici :
7 import pandas as pd
8 import matplotlib.pyplot as plt
9
10 df = dataset.groupby('TerritoryDescri')[['nbr_commande_livrees']].sum().nlargest(5)
11 plt.figure(figsize=(8,6))
12 df.sort_values().plot(kind='barh', color='purple')
13 plt.title("Top 5 Territoires par commandes livrées")
14 plt.xlabel("Nombre de commandes")
15 plt.tight_layout()
16 plt.show()
17

```

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 livrées")
plt.xlabel("Nombre de commandes")
plt.tight_layout()
plt.show()

```

7.2.3 Résultat Attendue

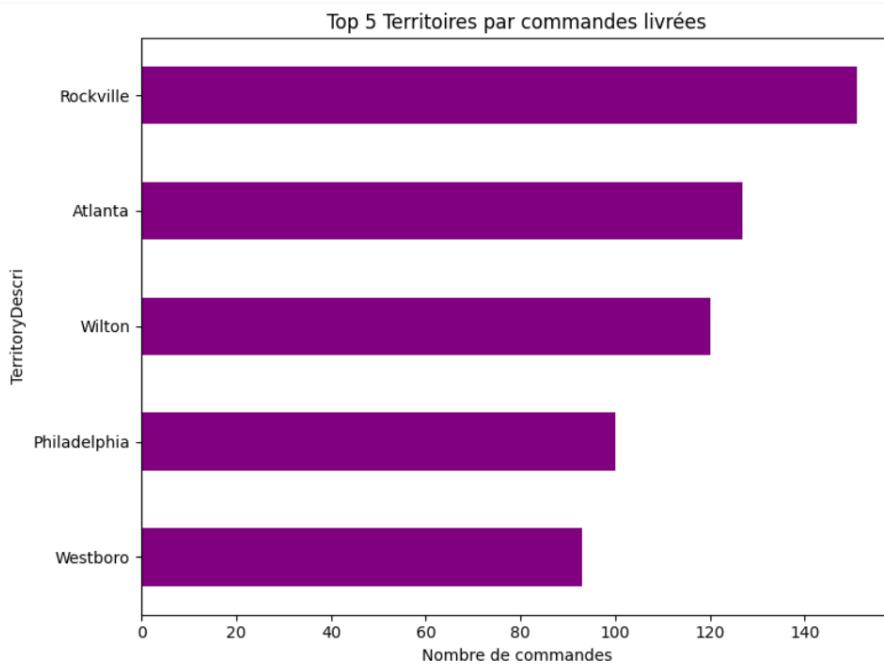


FIGURE 7 – Visualisation du top 5 des territoires générée par Python

7.2.4 Explications du Code

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

7.5 Dashboard Final Northwind

Notre dashboard final intègre 6 visuals Python , les voici :



FIGURE 8 – Dashboard Northwind avec 6 visuals Python

8 Résolution des Problèmes Courants

8.1 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

8.2 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

9 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. 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
- Publier et partager vos analyses