

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE



FACULTÉ D'INFORMATIQUE
DÉPARTEMENT SIQ
SPÉCIALITÉ : BIG DATA

Rapport de TP

TBI – SIG

Présenté par :

– Zouarqui Aya

Encadré(e) par :

– Mme Mekahlia

Année universitaire : 2025/2026

Introduction

Power BI est une plateforme complète d'analyse décisionnelle proposée par Microsoft. Elle permet de se connecter à différentes sources, de transformer les données, de construire un modèle analytique et de créer des visualisations interactives. Contrairement à des outils ETL spécialisés comme Talend, Power BI regroupe l'acquisition, la transformation, la modélisation et la visualisation dans un environnement unique.

Principaux composants de Power BI

- **Power Query** : module ETL dédié à la préparation et la transformation des données.
- **Power Pivot** : moteur de modélisation permettant de créer des relations et des mesures DAX.
- **Power View / Power Map** : outils de visualisation pour concevoir des tableaux de bord interactifs.
- **Power BI Service** : espace cloud pour publier, partager et collaborer autour des rapports.

Power BI vs Talend

Talend est un outil ETL orienté développement, adapté aux pipelines d'intégration complexes. Power BI, en revanche, offre une approche plus intuitive : il intègre un ETL (Power Query), un moteur de modélisation (Power Pivot) et des outils de visualisation. Pour un projet académique, Power BI constitue un choix optimal : il couvre l'ensemble du processus décisionnel dans un seul logiciel gratuit (version Desktop).

Première partie

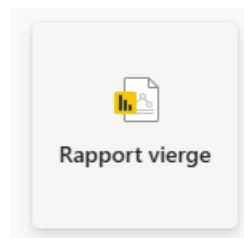
Processus ETL

1. Construction de la table Dim_Employee

Étape 1 : Importation des données sources

A. Source SQL Server

— Accueil → Obtenir des données → SQL Server



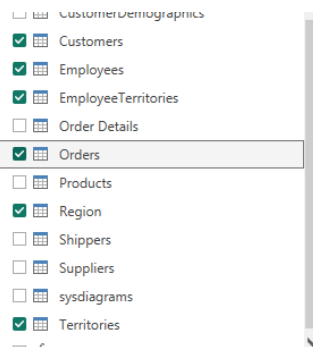
— Connexion au serveur Northwind



FIGURE 1 – Connexion SQL Server



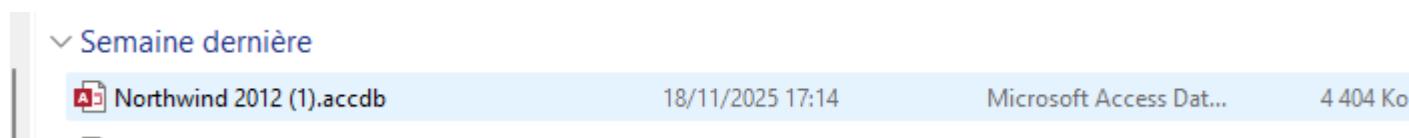
— Importation des tables : Employees, EmployeeTerritories, Territories, Region



B. Source Access

- Import via : Obtenir des données → Access
- Chargement du fichier accdb

Obtenir des données pour commencer →



Résultat

- 4 tables SQL Server chargées

- 1 table access importée
- Les données sont prêtes pour la transformation

Étape 2 : Nettoyage des données

A. Nettoyage des employés SQL Server

- Sélectionner uniquement les colonnes nécessaires : EmployeeID, Nom, Prenom.
Clic droit → *Supprimer les autres colonnes*.

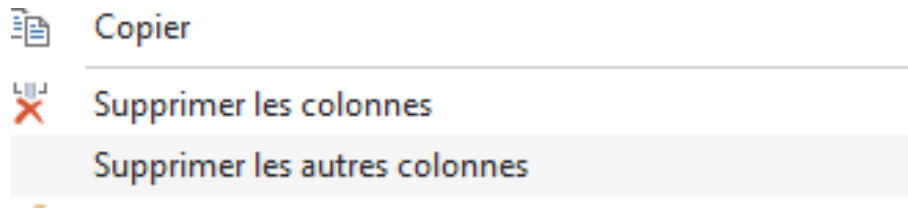
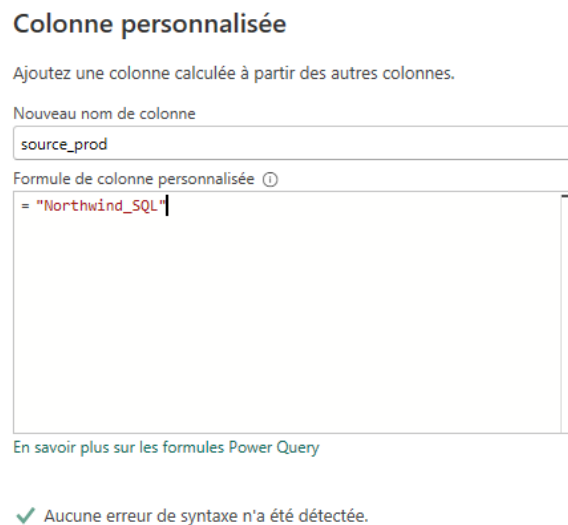


FIGURE 2 – Nettoyage de la table Employees (SQL Server)

- Ajouter une colonne personnalisée **source_prod** : Accueil → Ajouter une colonne
→ Colonne personnalisée Formule utilisée : "Northwind_SQL"



B. Nettoyage des employés Access

- Conserver uniquement les colonnes : ID, Last_Name, First_Name.
- Renommer les colonnes pour harmoniser avec SQL Server :
- ID → id_employee
- Last_Name → Nom
- First_Name → Prenom
- Ajouter une colonne personnalisée **source_prod** contenant : "Northwind_Access"

Colonne personnalisée

Ajoutez une colonne calculée à partir des autres colonnes.

Nouveau nom de colonne

source_prod

Formule de colonne personnalisée ⓘ

= "Northwind_Access"

Colonnes disponibles

ID
Company
Last Name
First Name
E-mail Address
Job Title
Business Phone

<< Insérer

En savoir plus sur les formules Power Query

✓ Aucune erreur de syntaxe n'a été détectée.

OK

Annuler

Étape 3 : Union des deux sources

- Accueil → Combiner → Ajouter des requêtes.
- Union de : **Employees_SQL** + **Employees_Access** (colonnes harmonisées pour permettre la concaténation).

= Table.Combine({Employees_ssms, #"Employees _Access"})				
	EmployeeID	Nom	Prenom	source_prod
1	1	Davolio	Nancy	Northwind_SQL
2	2	Fuller	Andrew	Northwind_SQL
3	3	Leverling	Janet	Northwind_SQL
4	4	Peacock	Margaret	Northwind_SQL
5	5	Buchanan	Steven	Northwind_SQL
6	6	Suyama	Michael	Northwind_SQL
7	7	King	Robert	Northwind_SQL
8	8	Callahan	Laura	Northwind_SQL
9	9	Dodsworth	Anne	Northwind_SQL
10	1	Freehafer	Nancy	Northwind_Access
11	2	Cencini	Andrew	Northwind_Access
12	3	Kotas	Jan	Northwind_Access
13	4	Sergienko	Mariya	Northwind_Access
14	5	Thorpe	Steven	Northwind_Access
15	6	Neipper	Michael	Northwind_Access
16	7	Zare	Robert	Northwind_Access
17	8	Giussani	Laura	Northwind_Access
18	9	Hellung-Larsen	Anne	Northwind_Access

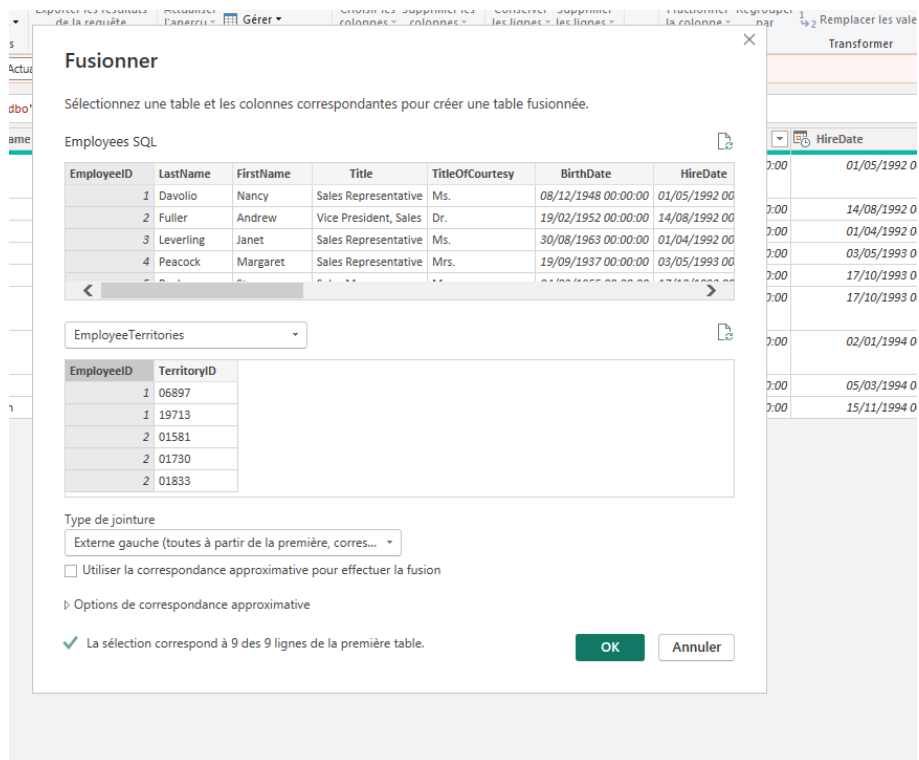
Étape 4 : Réalisation des jointures graphiques

A. Jointure : Employees + EmployeeTerritories

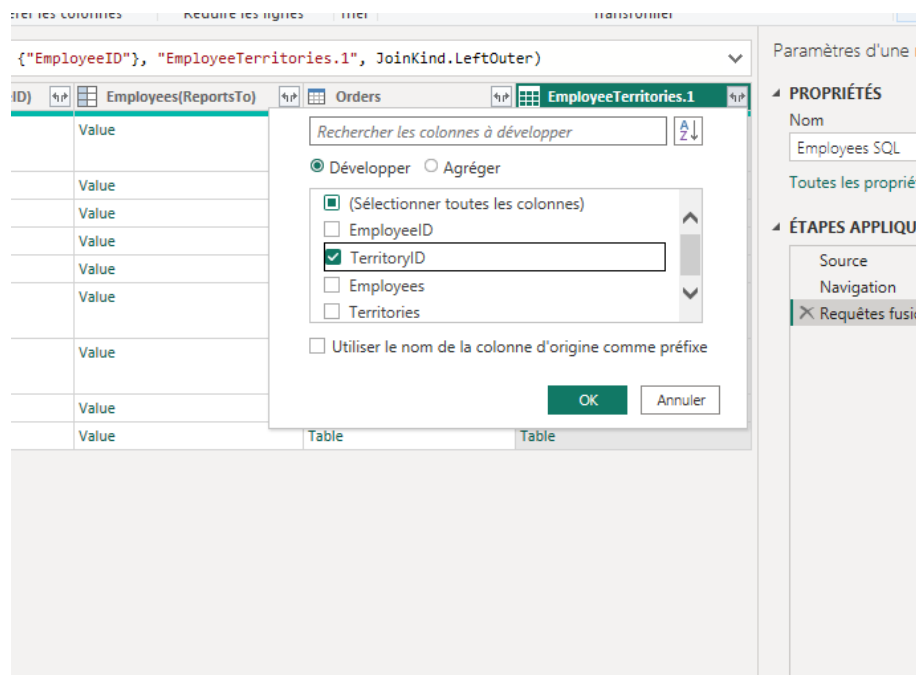
- Fusionner les requêtes : Employees_Union avec EmployeeTerritories



— Clé de jointure : EmployeeID (gauche)



— Développer uniquement la colonne : TerritoryID



B. Jointure avec la table Territories

— Fusion sur la colonne TerritoryID.

Fusionner

Sélectionnez des tables et les colonnes correspondantes pour créer une table fusionnée.

Fusionner1

EmployeeID	Nom	Prenom	source_prod	TerritoryID
1	Davolio	Nancy	Northwind_SQL	06897
1	Davolio	Nancy	Northwind_SQL	19713
2	Fuller	Andrew	Northwind_SQL	01581
2	Fuller	Andrew	Northwind_SQL	01730
2	Fuller	Andrew	Northwind_SQL	01833

Territories

TerritoryID	TerritoryDescription	RegionID
01581	Westboro	1
01730	Bedford	1
01833	Georgetow	1
02116	Boston	1
02139	Cambridge	1

Type de jointure
Externe gauche (toutes à partir de la première, corres...)

☐ Utiliser la correspondance approximative pour effectuer la fusion

Options de correspondance approximative

✓ La sélection correspond à 98 des 98 lignes de la première table.

OK Annuler

— Colonnes développées : TerritoryDescription, RegionID.

Rechercher les colonnes à développer A Z

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)
☐ TerritoryID
☒ TerritoryDescription
☒ RegionID
☐ EmployeeTerritories
☐ Region

☐ Utiliser le nom de la colonne d'origine comme préfixe

OK Annuler

C. Jointure avec la table Region

— Fusion sur la colonne RegionID.

Fusionner ×

Sélectionnez des tables et les colonnes correspondantes pour créer une table fusionnée.

Fusionner2

EmployeeID	Nom	Prenom	source_prod	TerritoryID	TerritoryDescription	RegionID
1	Davolio	Nancy	Northwind_SQL	06897	Wilton	1
1	Davolio	Nancy	Northwind_SQL	19713	Neward	1
2	Fuller	Andrew	Northwind_SQL	01581	Westboro	1
2	Fuller	Andrew	Northwind_SQL	01730	Bedford	1
2	Fuller	Andrew	Northwind_SQL	01833	Georgetow	1

Region

RegionID	RegionDescription
1	Eastern
2	Western
3	Northern
4	Southern

Type de jointure
Externe gauche (toutes à partir de la première, corres...)

☐ Utiliser la correspondance approximative pour effectuer la fusion

Options de correspondance approximative

✓ La sélection correspond à 98 des 98 lignes de la première table.

OK Annuler

— Développer la colonne : RegionDescription.

Rechercher les colonnes à développer A Z ↓

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)

☐ RegionID

☒ RegionDescription

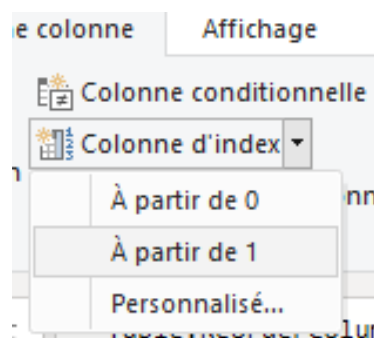
☐ Territories

☐ Utiliser le nom de la colonne d'origine comme préfixe

Étape 5 : Ajouter une colonne d'index

Procédure :

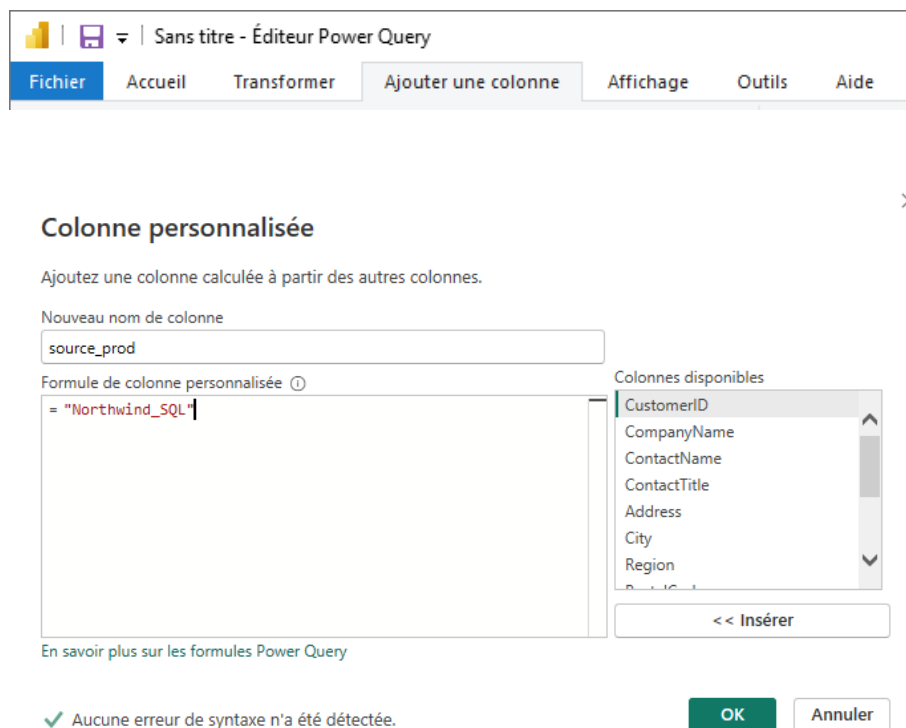
- Accueil → Ajouter une colonne → Colonne d'index
- Choisir : *À partir de 1*
- La colonne générée est ensuite renommée en : `id_seqemploye`



2. Construction de la dimension Client (Dim_Client)

Étape 1 : Préparation côté SQL Server

- Import de la table Customers
- Sélection : CustomerID, CompanyName, City, Country
- Renommages + Ajouter une colonne personnalisée `source_prod` contenant : "Northwind_SQL"



Étape 2 : Préparation de la source Acces

- Import du fichier accddb
- Sélection des CustomerID, CompanyName, City, Country
- Renommage

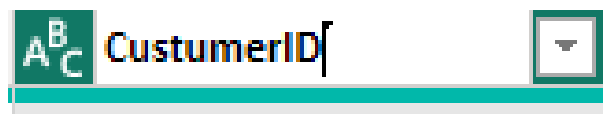
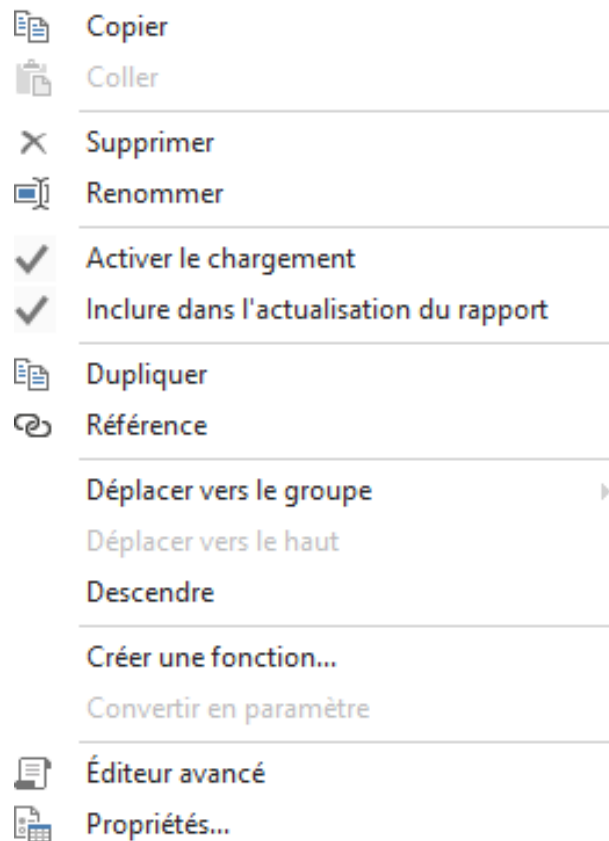


FIGURE 3 – Nettoyage de la table Employees (SQL Server)

— Ajouter une colonne personnalisée `source_prod` contenant : "Northwind_Access"

×

Colonne personnalisée

Ajoutez une colonne calculée à partir des autres colonnes.

Nouveau nom de colonne

source_prod

Formule de colonne personnalisée ⓘ

= "Northwind_Access"

Colonnes disponibles

ID

Company

Last Name

First Name

E-mail Address

Job Title

Business Phone

<< Insérer

En savoir plus sur les formules Power Query

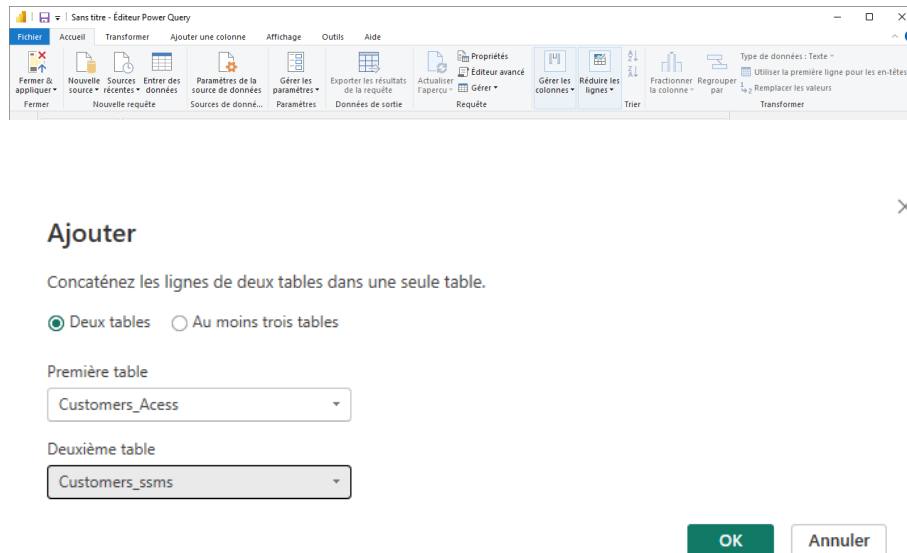
✓ Aucune erreur de syntaxe n'a été détectée.

OK

Annuler

Étape 3 : Union des deux sources

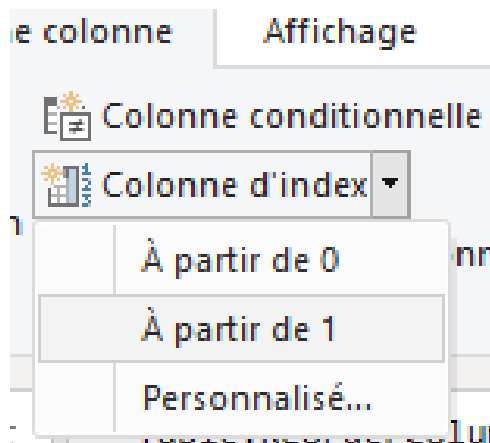
- Combiner → Ajouter des requêtes
- Union Customers_SQL + Customers_Access



Étape 4 : Ajouter une colonne d'index

Procédure :

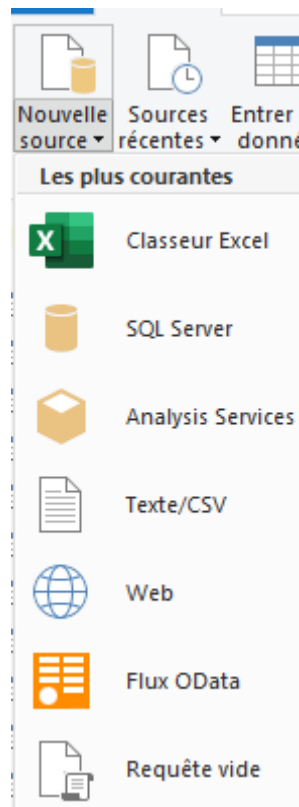
- Accueil → Ajouter une colonne → Colonne d'index
- Choisir : À partir de 1
- La colonne générée est ensuite renommée en : id_sequeclient



3. Table de dimension Temps

Etape 1 :Création d'une requête vierge

Une nouvelle requête a été générée afin de construire une table temporelle couvrant les années 1990 à 2030.



Etape 2 : Code M utilisé

```
let
    DateStart = #date(1990, 1, 1),
    DateEnd   = #date(2030, 12, 31),

    // Liste de toutes les dates
    ListDates = List.Dates(DateStart, Duration.Days(DateEnd - DateStart) + 1,
        #duration(1,0,0,0)),
    TableDates = Table.FromList(ListDates, Splitter.SplitByNothing(), {"Date"},
        type date),
```

```

// Extraire année et mois
AddYear = Table.AddColumn(TableDates, "Annee", each Date.Year([Date]),
Int64.Type),
AddMonth = Table.AddColumn(AddYear, "Mois", each Date.Month([Date]),
Int64.Type),
AddAnneeMois = Table.AddColumn(AddMonth, "AnneeMois", each Date.ToText([Date],
"yyyy/MM"), type text),

// Garder une seule ligne par mois
Grouped = Table.Group(AddAnneeMois, {"Annee", "Mois", "AnneeMois"},
{"MinDate", each List.Min([Date]), type date})),

// Ajouter ID séquentiel
AddIndex = Table.AddIndexColumn(Grouped, "id_temps", 1, 1, Int64.Type),

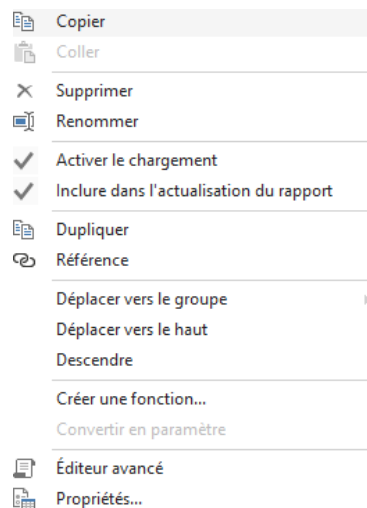
// Réorganiser colonnes
Final = Table.ReorderColumns(AddIndex, {"id_temps", "Annee", "Mois", "AnneeMois"}),
in
Final

```


4. Construction de la Table de Faits

Étape 1 : Nettoyage des données OrderSSMS et OrderAccess

- dupliquer la table orderSQL



- Sélectionner uniquement les colonnes : ShippedDate, OrderDate, IdClient, IdEmployee, IdOrder.
- Clic droit → *Supprimer les autres colonnes.*
- Ajouter une colonne personnalisée **Source_Prod** indiquant l'origine des données :
 - "Northwind_SQL" pour OrderSSMS
 - "Northwind_Access" pour OrderAccess

Etape 2 : Fusion des tables Order Access et Customer Access

Dans la table Order Access, les champs ID et Customer ID sont des identifiants numériques, tandis que dans la table Customer Access, le Customer ID correspond au nom du client. Il a donc été nécessaire d'effectuer cette fusion afin d'harmoniser la structure et d'obtenir une organisation cohérente des données.

- Fusion entre **Order Access** et **Customer Access** sur le **Customer ID** → permet de récupérer les informations client.
- Développement de la colonne issue de la fusion → on ne garde que le nom du client.

- Suppression de l'ancien **Customer ID** numérique → évite les doublons et clarifie la structure.
- Renommage de la colonne **Nom du client** en **Client ID** → harmonisation et cohérence des identifiants.

Cette étape permet de préparer correctement la table **Order Access** avant son intégration dans la table de faits.

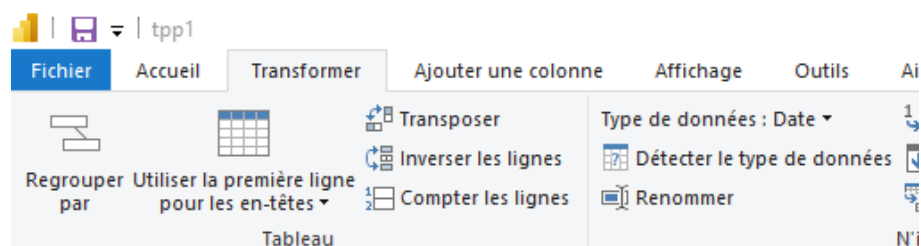
Étape 3 : Union des deux sources

- Accueil → Combiner → Ajouter des requêtes.
- Union des tables : **OrderSSMS** + **OrderAccess**.



Étape 4 : Transformation des dates

- Sélectionner la colonne **ShipDate** → Transformer → Type de données → Date/Heure → Date.
- Même opération pour **OrderDate**.







Type de données : Date ▼

- Nombre décimal
- Nombre décimal fixe
- Nombre entier
- Pourcentage
- Date/Heure
- Date**
- Heure
- Date/Heure/Fuseau horaire
- Durée
- Texte
- Vrai/Faux
- Binaire

Étape 5 : Création des colonnes conditionnelles

Ajouter une colonne Affichage Ou

-  Ajouter une fonction personnalisée
-  Colonne conditionnelle
-  Colonne d'index ▼
-  Duplication de la colonne

Général

- Création de la colonne **Livrée** :
- Condition : si **ShipDate** \neq null \rightarrow 1, sinon 0.

Ajouter une colonne conditionnelle ✕

Ajoutez une colonne conditionnelle calculée en fonction des autres colonnes ou valeurs.

Nouveau nom de colonne
livree

	Nom de la colonne	Opérateur	Valeur	Sortie
Si	ShippedDate	est différent de	null	Alors 1
...				

Ajouter une clause

Autre

ABC 123 * 0

OK Annuler

ABC 123	livree
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1
	1

- Création de la colonne NonLivrée :
- Condition : si ShipDate = null → 1, sinon 0.

Ajouter une colonne conditionnelle

Ajoutez une colonne conditionnelle calculée en fonction des autres colonnes ou valeurs.

Nouveau nom de colonne

	Nom de la colonne	Opérateur	Valeur	Sortie
Si	ShippedDate	égal à	ABC 123 null	Alors ABC 123 1
...				

Ajouter une clause

Autre

ABC 123
of

OK
Annuler

01/05/1998 00:00:00	11044	WOLZA	4	23/04/1998 00:00:00	Northwind_SQL	1	0
	11045	BOTTM	6	23/04/1998 00:00:00	Northwind_SQL	0	1
24/04/1998 00:00:00	11046	WANDK	8	23/04/1998 00:00:00	Northwind_SQL	1	0
01/05/1998 00:00:00	11047	EASTC	7	24/04/1998 00:00:00	Northwind_SQL	1	0

Étape 6 : Fusions avec les dimensions



A. Fusion avec Dim_Employee

— Fusion sur : IdEmployee.

Fusionner

Sélectionnez une table et les colonnes correspondantes pour créer une table fusionnée.

Orders_ssms (2)

ShippedDate	OrderID	CustomerID	EmployeeID	OrderDate	source_prod	livree	nonlivree	id_s
16/07/1996	10248	VINET	5	04/07/1996	Northwind_SQL	1	0	
16/08/1996	10274	VINET	6	06/08/1996	Northwind_SQL	1	0	
10/09/1996	10295	VINET	2	02/09/1996	Northwind_SQL	1	0	
10/07/1996	10249	TOMSP	6	05/07/1996	Northwind_SQL	1	0	

dim_employee_final

id_seqEmployee	EmployeeID	Nom	Prenom	source_prod	TerritoryID	TerritoryDescription	Region
1	1	Davolio	Nancy	Northwind_SQL	06897	Wilton	Eastern
2	1	Davolio	Nancy	Northwind_SQL	19713	Neward	Eastern
3	2	Fuller	Andrew	Northwind_SQL	01581	Westboro	Eastern
4	2	Fuller	Andrew	Northwind_SQL	01730	Bedford	Eastern

Type de jointure

Externe gauche (toutes à partir de la première, corres...

☐ Utiliser la correspondance approximative pour effectuer la fusion

Options de correspondance approximative

✓ La sélection correspond à 878 des 878 lignes de la première table.

OK

Annuler

— Développer uniquement la colonne : id_seqemploye.

Rechercher les colonnes à développer

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)

☒ id_seqEmployee

☐ EmployeeID

☐ Nom

☐ Prenom

☐ source_prod

☐ TerritoryID

☐ TerritoryDescription

☐ RegionDescription

☒ Utiliser le nom de la colonne d'origine comme préfixe

OK Annuler

B. Fusion avec Dim_Client

— Fusion sur : IdClient.

Fusionner

Sélectionnez une table et les colonnes correspondantes pour créer une table fusionnée.

Orders_ssms (2)

ShippedDate	OrderID	CustomerID	EmployeeID	OrderDate	source_prod	livree	nonlivree
16/07/1996	10248	VINET	5	04/07/1996	Northwind_SQL	1	0
10/07/1996	10249	TOMSP	6	05/07/1996	Northwind_SQL	1	0
12/07/1996	10250	HANAR	4	08/07/1996	Northwind_SQL	1	0
15/07/1996	10251	VICTE	3	08/07/1996	Northwind_SQL	1	0
11/07/1996	10252	SUPRD	4	09/07/1996	Northwind_SQL	1	0

dim_clientfinal

id_seqclient	CustomerID	CompanyName	City	Country	source_prod
1	Bedecs	Company A	Seattle	USA	Northwind_Access
2	Gratacos Solsona	Company B	Boston	USA	Northwind_Access
3	Axen	Company C	Los Angelas	USA	Northwind_Access
4	Lee	Company D	New York	USA	Northwind_Access
5	O'Donnell	Company E	Minneapolis	USA	Northwind_Access

Type de jointure

Externe gauche (toutes à partir de la première, corres...

☐ Utiliser la correspondance approximative pour effectuer la fusion

Options de correspondance approximative

✓ La sélection correspond à 830 des 878 lignes de la première table.

OK Annuler

— Développer uniquement : id_seqclient.

Rechercher les colonnes à développer A Z ↓

☒ Développer ☐ Agréger

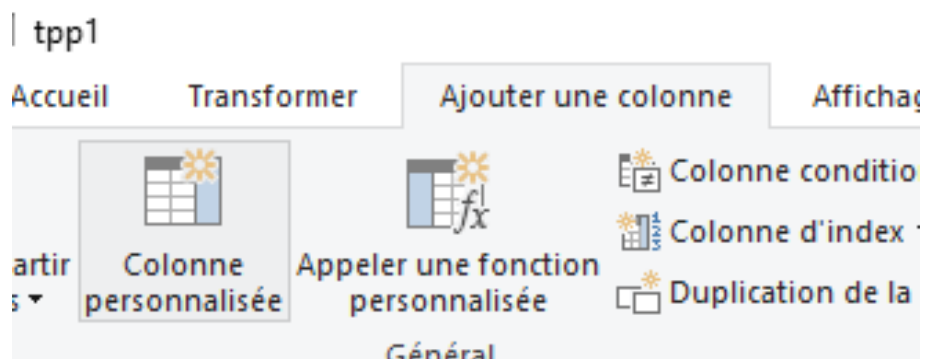
☒ (Sélectionner toutes les colonnes)
 ☒ id_seqclient
 ☐ CustomerID
 ☐ CompanyName
 ☐ City
 ☐ Country
 ☐ source_prod

☒ Utiliser le nom de la colonne d'origine comme préfixe

OK Annuler

C. Création de la colonne MoisAnnée

- Ajouter une colonne → Colonne personnalisée



- Transformer la colonne `OrderDate` en un format : **Mois + Année**.
- Exemple : mars 1997, novembre 1994.

×

Colonne personnalisée

Ajoutez une colonne calculée à partir des autres colonnes.

Nouveau nom de colonne

Mois_Année

Formule de colonne personnalisée ⓘ

= Date.ToText([OrderDate], "yyyy-MM")

Colonnes disponibles

ShippedDate

OrderID

CustomerID

EmployeeID

OrderDate

source_prod

livree

..

< < Insérer

En savoir plus sur les formules Power Query

✓ Aucune erreur de syntaxe n'a été détectée.

OK

Annuler

- Dans la table **Dim__Temps**, la colonne MoisAnnée est au format : AAAA/ MM (exemple : 2000/01).
- Dans la table de faits, la colonne MoisAnnéeProduit était au format : AAAA-MM (exemple : 2000-01).
- Pour harmoniser les deux formats avant la fusion, j'ai effectué une transformation :
 - clic droit sur moisannée → Remplacer les valeurs →
 - Valeur à rechercher : -
 - Remplacer par : /

×

Remplacer les valeurs

Remplacez une valeur dans les colonnes sélectionnées par une autre.

Valeur à rechercher

-

Remplacer par

/

Options avancées

OK

Annuler

D. Fusion avec Dim__Temps

- Fusion sur la colonne Mois_Année.

Fusionner

Sélectionnez une table et les colonnes correspondantes pour créer une table fusionnée.

Orders_ssms (2)

employeeID	OrderDate	source_prod	livree	nonlivree	id_seqclient	id_seqEmployee	Mois_Année
5	04/07/1996	Northwind_SQL	1	0	114	17	1996/07
5	04/07/1996	Northwind_SQL	1	0	114	18	1996/07
5	04/07/1996	Northwind_SQL	1	0	114	19	1996/07
5	04/07/1996	Northwind_SQL	1	0	114	20	1996/07

dim_tempsfinal

id_temps	annee	annee_mois
1	1990	1990/01
2	1990	1990/02
3	1990	1990/03
4	1990	1990/04
5	1990	1990/05

Type de jointure

Externe gauche (toutes à partir de la première, corres...

☐ Utiliser la correspondance approximative pour effectuer la fusion

Options de correspondance approximative

✓ La sélection correspond à 8356 des 8356 lignes de la première table.

OK

Annuler

— Développer uniquement la colonne : id_temps.

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)

☒ id_temps

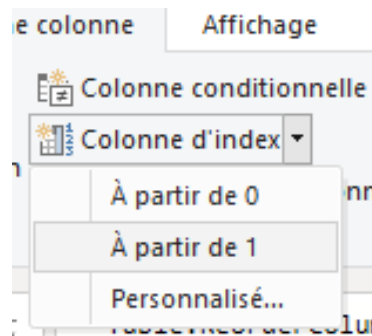
☐ annee

☐ annee_mois

☒ Utiliser le nom de la colonne d'origine comme préfixe

Étape 7 : Création d'une colonne d'index

— Ajouter une colonne → Colonne d'index → À partir de 1.

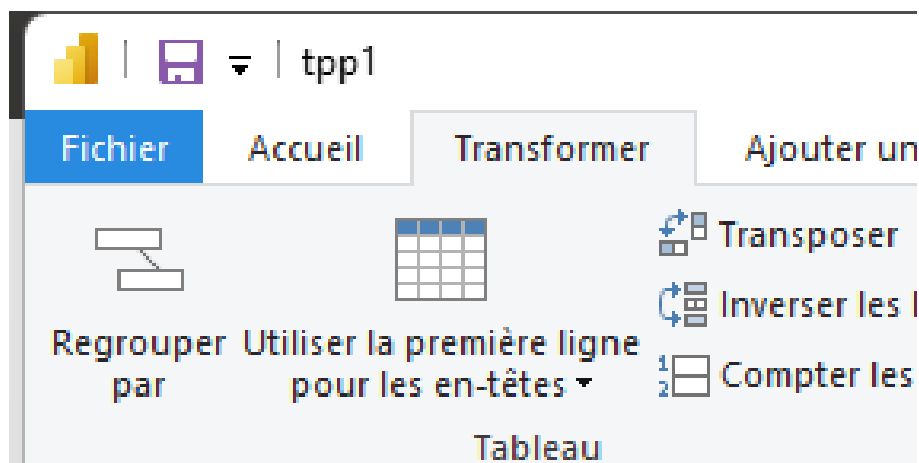


- Renommer cette colonne en : `id_seqfait`.



Étape 8 : Groupement avancé

- Accueil → Transformer → Grouper par.



- Choisir : **Groupement avancé**.
- Colonnes de groupement :
 - `id_seqemploye`
 - `id_seqclient`
 - `id_temps`
- Création des colonnes d'agrégation :
 - **NombreCommandesLivrées** : somme de la colonne `Livrée`

- **NombreCommandesNonLivrées** : somme de la colonne NonLivrée
- **Détail** : opération *Toutes les lignes*

×

Regrouper par

Spécifiez les colonnes de regroupement et une ou plusieurs sorties.

☐ De base
 ☒ Avancé

id_seqclient

id_seqEmployee

id_temps

Ajouter un regroupement

Nouveau nom de colonne	Opération	Colonne
nbr_commandel_livree	Somme	livree
nbr_commandel_nonlivree	Somme	nonlivree
details	Toutes les lignes	

Ajouter une agrégation

OK

Annuler

Étape 9 : Finalisation

- Développer la colonne **Détail**.

2 nbr_commandel_nonlivree

details

Rechercher les colonnes à développer

☒ Développer
 ☐ Agréger

☒ (Sélectionner toutes les colonnes)
 ☒ id_seqfait
 ☐ ShippedDate
 ☐ OrderID
 ☐ CustomerID
 ☐ EmployeeID
 ☐ OrderDate
 ☐ source_prod
 ☐ livree
 ☐ nonlivree
 ☐ id_seqclient
 ☐ id_seqEmployee
 ☐ Mois_Année
 ☐ id_temps

☒ Utiliser le nom de la colonne d'origine comme préfixe

OK

Annuler

- Conserver uniquement : **id_seqfait**.
- Fermer et appliquer.

Resultats

	1.2 id_seqfait	1 ² ₃ id_seqclient	1 ² ₃ id_seqEmployee	1 ² ₃ id_temps	1.2 nbr_commandel_livree	1.2 nbr_commandel
1	1	114	17	79	1	
2	2	114	18	79	1	
3	3	114	19	79	1	
4	4	114	20	79	1	
5	5	114	21	79	1	
6	6	114	22	79	1	
7	7	114	23	79	1	
8	8	114	58	79	1	
9	9	114	62	79	1	
10	10	114	63	79	1	
11	11	114	65	79	1	
12	12	114	67	79	1	
13	13	114	68	79	1	
14	14	114	70	79	1	
15	15	43	17	79	1	
16	16	43	18	79	1	
17	17	43	19	79	1	
18	18	43	20	79	1	
19	19	43	21	79	1	
20	20	43	22	79	1	
21	21	43	23	79	1	
22	22	43	58	79	1	
23	23	43	62	79	1	
24	24	43	63	79	1	
25	25	43	65	79	1	
26	26	43	67	79	1	
27	27	43	68	79	1	
28	28	43	70	79	1	
29	29	118	17	79	1	
30	30	118	18	79	1	
31	31	118	19	79	1	
32	32	118	20	79	1	
33	33	118	21	79	1	
34	34	118	22	79	1	
35	35	118	23	79	1	
36	36	118	58	79	1	
37	37	118	62	79	1	
38	38	118	63	79	1	

Explication des choix techniques

1. Choix des jointures

Les jointures utilisées sont principalement de type "**Externe gauche**" (Left Outer Join). Ce choix permet de :

1.1 Conservation des données principales

- Conserver toutes les lignes de la table principale (ex. **Employees**, **Orders**), même si certaines n'ont pas de correspondance dans les tables liées.
- Éviter la perte de données lors de la fusion de sources hétérogènes (SQL Server et Access).

1.2 Traçabilité des données orphelines

- Garder une trace des données orphelines pour analyse ultérieure (ex. employés sans territoire, commandes sans client).

2. Regroupement avancé dans la table de faits

Le **groupement avancé** a été appliqué sur :

- `id_seqemployee`
- `id_seqclient`
- `id_temps`

2.1 Agrégations appliquées

- `nbr_commandelivree` : somme des commandes livrées.
- `nbr_commandenonlivree` : somme des commandes non livrées.
- `details` : conservation de toutes les lignes brutes pour analyse détaillée.

2.2 Justification du regroupement

- Réduire la granularité au niveau **employé** × **client** × **période** pour des analyses agrégées.
- Permettre des calculs de **KPI** (taux de livraison, performance par employé) sans surcharger le modèle.
- Garder la possibilité de détailler chaque commande via la colonne `details`.

3. Harmonisation des formats de date

Avant la fusion avec `Dim_Temps`, la colonne `MoisAnneeProduit` a été transformée :

3.1 Formats utilisés

- **Format initial** : AAAA-MM (ex. 2000-01)
- **Format cible** : AAAA/MM (ex. 2000/01)

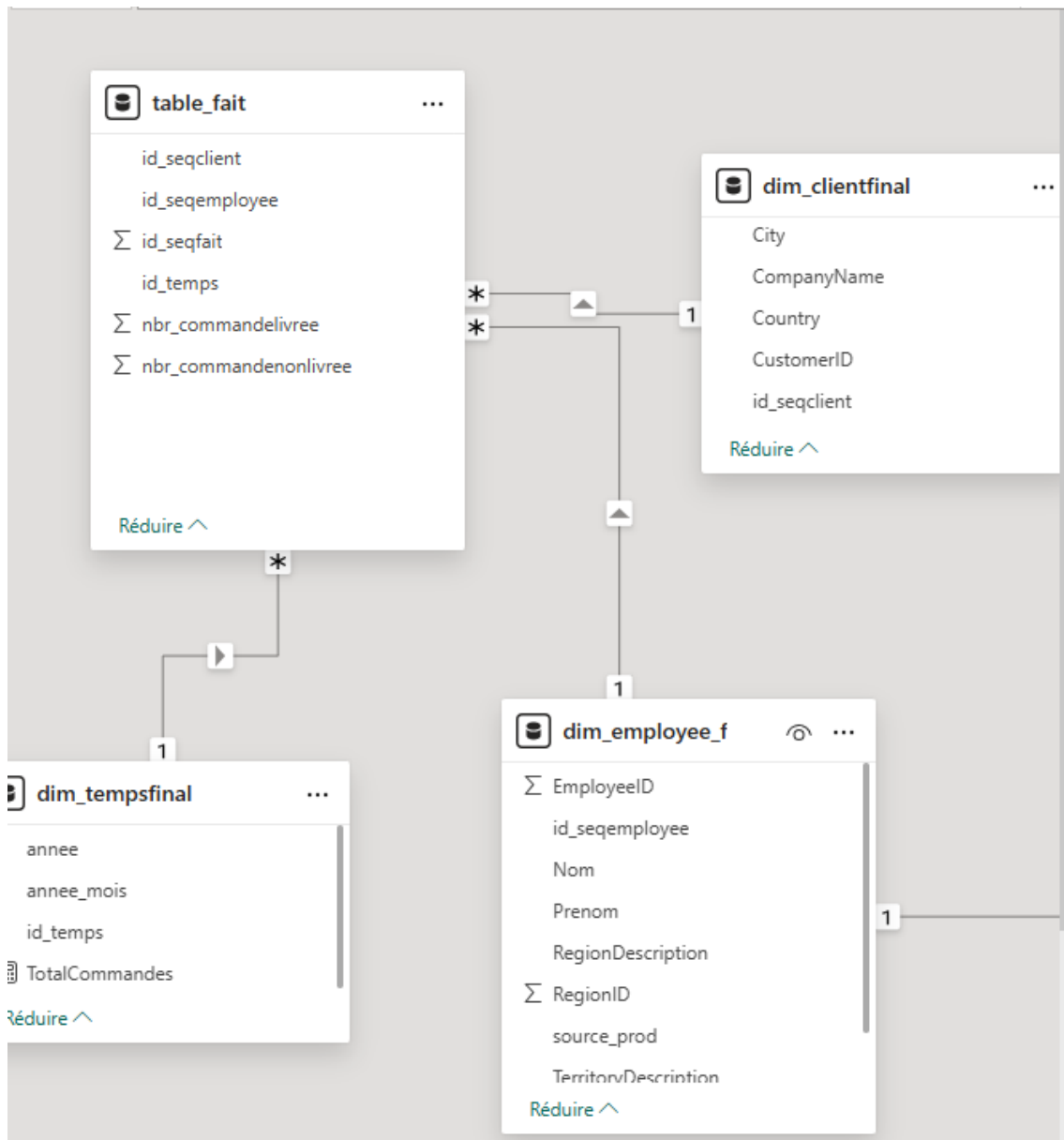
3.2 Raison de la transformation

- Assurer la **cohérence** avec le format de `Dim_Temps[annee_mois]`.
- Permettre une **jointure exacte** sans erreur de type ou de format.

Deuxième partie

Dashbaord

1. Model Etoile



1.1 Modèle Étoile – Explication des relations

Le modèle adopté suit une **architecture en étoile (star schema)**, optimisée pour les requêtes analytiques dans Power BI. Il se compose d'une **table de faits** centrale et de **tables de dimension** reliées par des clés étrangères.

1.1.1 Structure du modèle

Table de faits centrale :

- `table_fait` – Contient les indicateurs mesurables :
 - `nbr_commandelivree` : nombre de commandes livrées
 - `nbr_commandenonlivree` : nombre de commandes non livrées
- Clés étrangères vers les dimensions :
 - `id_seqclient` → `dim_clientfinal`
 - `id_seqemployee` → `dim_employee_f`
 - `id_temps` → `dim_tempsfinal`

Tables de dimension :

- `dim_clientfinal` – Décrit les clients :
 - `CustomerID`, `CompanyName`, `City`, `Country`
 - Clé primaire : `id_seqclient`
- `dim_employee_f` – Décrit les employés :
 - `Nom`, `Prenom`, `RegionDescription`, `TerritoryDescription`
 - Clé primaire : `id_seqemployee`
- `dim_tempsfinal` – Décrit la dimension temporelle :
 - `annee`, `annee_mois`
 - Clé primaire : `id_temps`

1.1.2 Types de relations

Toutes les relations sont de type "**un-à-plusieurs**" (One-to-Many) :

- Une ligne dans une dimension peut correspondre à plusieurs lignes dans la table de faits
- Direction des filtres : **De la dimension vers la table de faits**
- Intégrité référentielle assurée par les clés de substitution (`id_seq*`)

1.1.3 Avantages de ce modèle

- **Performance** : requêtes plus rapides grâce au schéma dénormalisé
- **Simplicité** : navigation intuitive pour les utilisateurs
- **Maintenabilité** : modifications facilitées sur les dimensions sans affecter les faits
- **Compatibilité** : optimisé pour les mesures DAX et les filtres croisés

2. Visualisation

2.1. Création d'une Nouvelle Mesure

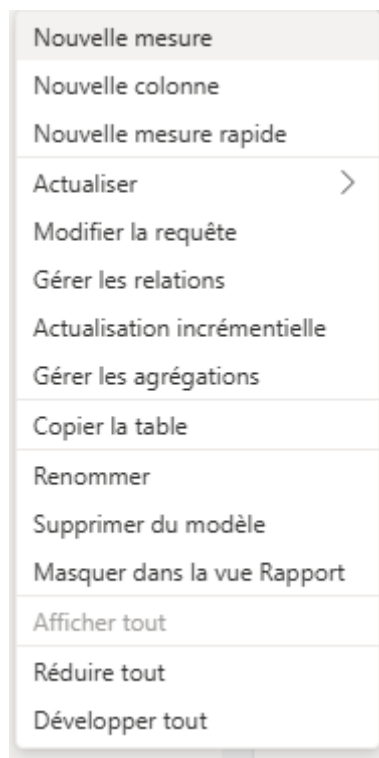
Étape 1 : Accéder à l'interface Power BI

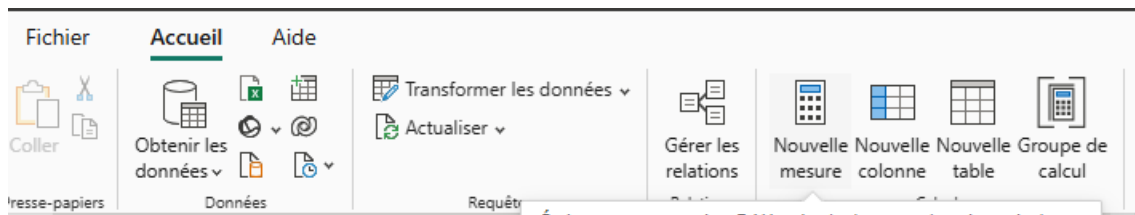
1. Ouvrir **Power BI Desktop**.
2. Cliquer sur l'onglet **Données** dans la barre latérale gauche.
3. Dans le volet **Champs** à droite, sélectionner la table cible.

Étape 2 : Créer une nouvelle mesure

Action :

- Clic droit sur le nom de la table
- Menu contextuel apparaît
- Sélectionner "Nouvelle mesure"





Étape 3 : Validation et sauvegarde

- Appuyer sur **Entrée** pour valider la mesure.
- Vérifier que l'icône verte apparaît (syntaxe correcte).
- La mesure s'affiche automatiquement sous la table sélectionnée.

Mesures DAX créées

Les mesures DAX ont été créées pour enrichir l'analyse

3.1 Pourquoi DAX plutôt que Power Query ?

- **Dynamisme** : les mesures sont recalculées selon les filtres appliqués.
- **Performance** : évite le stockage de colonnes agrégées en aval.
- **Flexibilité** : création d'indicateurs réutilisables dans plusieurs visuels.

2.2. Mesures Fondamentales

2.1.1. Commandes livrées

Commandes livrées = SUM(table_fait[nbr_commandelivree])

2.2.2. Commandes non livrées

Commandes non livrées = SUM(table_fait[nbr_commandenonlivree])

2.3 Visualisations

Étapes d'utilisation du visuel Python dans Power BI

L'utilisation du visuel Python dans Power BI s'est déroulée selon les étapes suivantes :

- **Accueil** → Accès à l'onglet *Visualisations*.
- **Visualisation Python** → Sélection du visuel *Python* depuis le panneau des visualisations.
- **Sélection des données** → Cochage des colonnes nécessaires depuis le panneau *Données* (dimensions temps, employé, client et mesures *commande_livree*, *commande_nonlivree*).
- **Génération du dataset** → Power BI crée automatiquement un dataset contenant les champs sélectionnés.
- **Script Python** → Saisie du code Python utilisant *Pandas* et *Matplotlib* dans la zone dédiée.

- **Exécution du script** → Lancement du code Python et génération du visuel.
- **Affichage du rapport** → Intégration automatique du graphique dans la page du rapport Power BI.

Dans cette section, nous présentons les différentes visualisations réalisées pour analyser les commandes livrées et non livrées, par employés, clients et régions. Les visualisations ont été réalisées à l'aide de Python intégré dans Power BI.

Visual 1 : Commandes non livrées par employé

- **Description** : Ce visuel permet de voir le nombre de commandes non livrées par chaque employé, ce qui aide à identifier les écarts de performance.

- **Type** : Histogramme

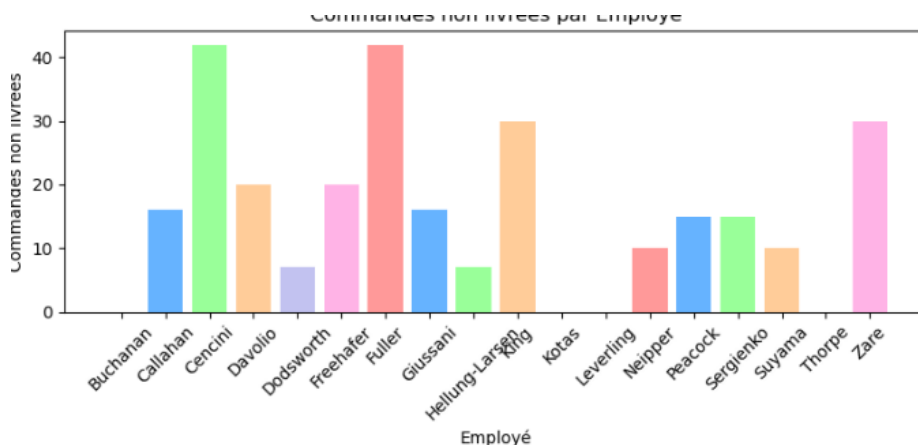
- **Code Python** :

```
import matplotlib.pyplot as plt
```

```
df_emp = dataset.groupby('LastName', as_index=False)['commande_nonlivree'].sum()
```

```
plt.bar(df_emp['LastName'], df_emp['commande_nonlivree'], color='green')
plt.title("Commandes non livrées par employé")
plt.xlabel("Employé")
plt.ylabel("Commandes non livrées")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

- **Capture d'écran** :



Visual 2 : Évolution des commandes livrées

- **Description** : Ce visuel montre l'évolution des commandes livrées dans le temps, permettant d'observer les tendances et les pics d'activité.

- **Type** : Courbe

- **Code Python** :

```
import matplotlib.pyplot as plt
```

```
df_time = dataset.groupby('AnneeMois', as_index=False)['commande_livree'].sum()
```

```
plt.plot(df_time['AnneeMois'], df_time['commande_livree'], color='darkgreen', ma
plt.title("Évolution des commandes livrées")
plt.xlabel("Année-Mois")
plt.ylabel("Commandes livrées")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

— Capture d'écran :



Visual 3 : Répartition des commandes livrées vs non livrées

— **Description** : Ce visuel montre la proportion de commandes livrées par rapport aux commandes non livrées, ce qui permet de visualiser rapidement le taux de livraison.

— **Type** : Camembert

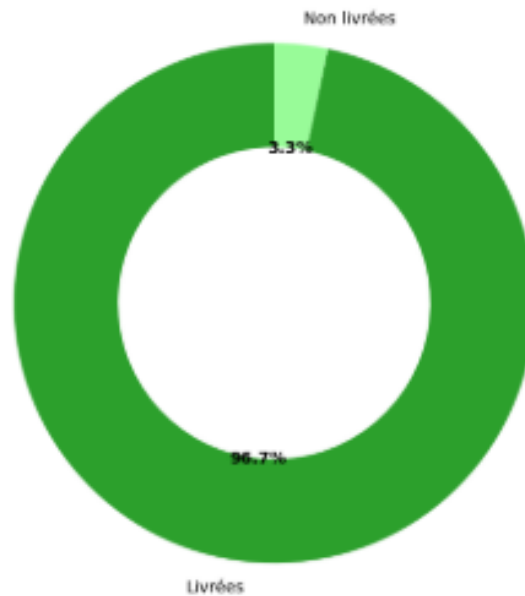
— **Code Python** :

```
import matplotlib.pyplot as plt

totals = [dataset['commande_livree'].sum(), dataset['commande_nonlivree'].sum()]
plt.pie(totals, labels=['Livrées', 'Non livrées'], autopct='%1.1f%%',
        colors=['#2ca02c', '#98fb98'], startangle=90)
plt.title("Répartition commandes livrées vs non livrées")
plt.tight_layout()
plt.show()
```

— Capture d'écran :

Répartition des commandes livrées vs non livrées (compact)



Visual 4 : Répartition des commandes livrées par client

- **Description :** Ce visuel montre les commandes livrées par les principaux clients, permettant d'identifier les clients les plus actifs.

- **Type :** Camembert

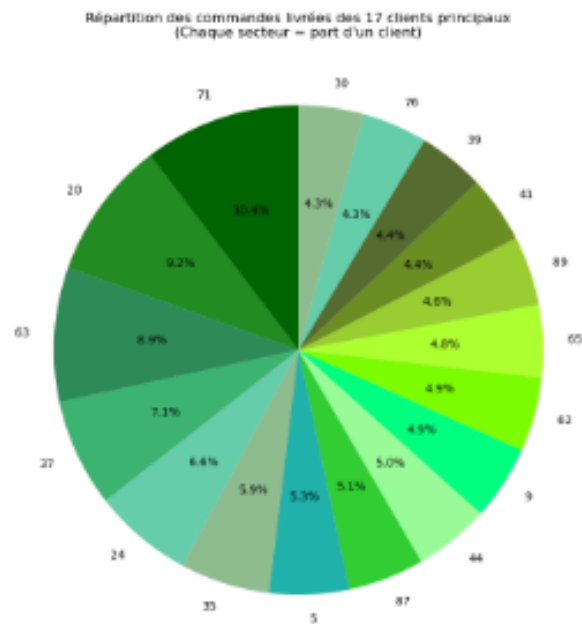
- **Code Python :**

```
import matplotlib.pyplot as plt
```

```
df_clients = dataset.groupby('id_seqclient', as_index=False)['commande_livree']  
df_top17 = df_clients.sort_values('commande_livree', ascending=False).head(17)
```

```
colors = plt.cm.Greens(range(50, 200, 10))  
plt.pie(df_top17['commande_livree'],  
        labels=df_top17['id_seqclient'].astype(str),  
        autopct='%1.1f%%', colors=colors, startangle=90)  
plt.title("Commandes livrées des 17 clients principaux")  
plt.tight_layout()  
plt.show()
```

- **Capture d'écran :**



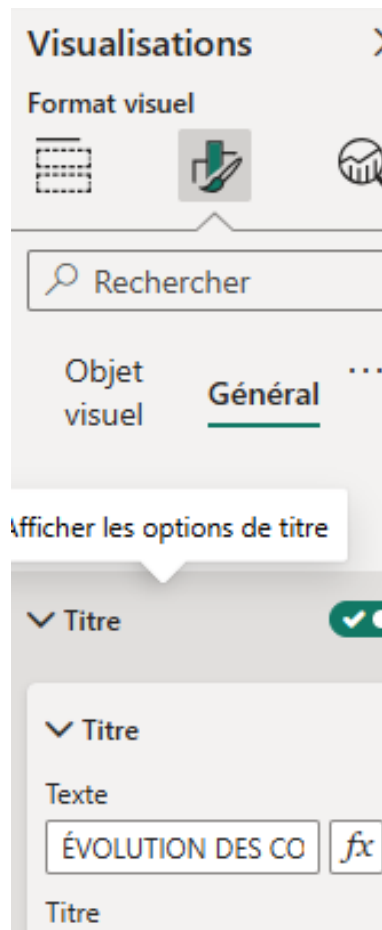
2.4 Choix des bibliothèques

- **Pandas** : pour la manipulation et l'agrégation des données. Permet de grouper par clients, employés ou dates et de calculer facilement des sommes ou d'autres mesures. Simplifie l'intégration avec Power BI.
- **Matplotlib** : pour la visualisation graphique. Permet de créer différents types de graphiques (histogrammes, courbes, camemberts) et de personnaliser couleurs, titres et axes, obtenant des visualisations claires et professionnelles.

2.5 Méthodologie pour les titres des visuels

Étape 1 : Accéder aux paramètres du visuel

1. Sélectionner le visuel
2. Panneau droit → Onglet **Visualisation**
3. Chercher l'option **Titre** ou **En-tête**

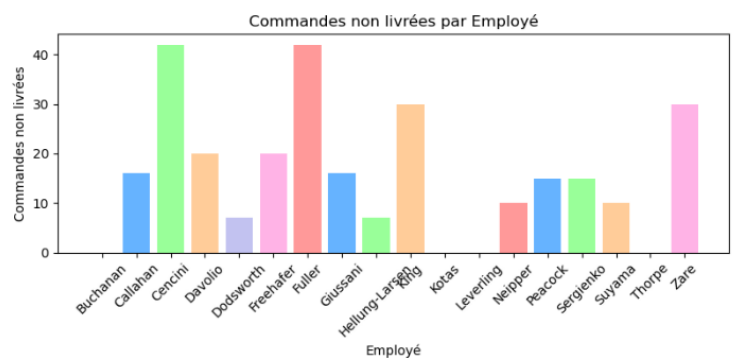
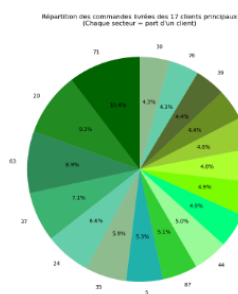
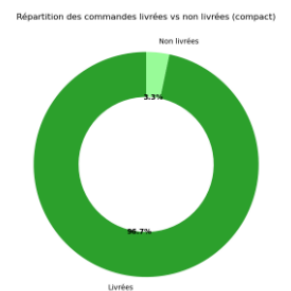


Étape 2 : Configurer le titre

- Activer : ON
- Saisir le texte du titre
- Formatage :
 - Taille police : 12-14 px
 - Style : Gras
 - Aligement : Centre
 - Couleur : Noir (#000000)

3. Resultats

Tableau de Bord Northwind – Analyse des Commandes et Clients



Conclusion

Ce projet a permis de mettre en place une chaîne complète allant de l'intégration des données (ETL) jusqu'à la création d'un tableau de bord interactif dans Power BI. La première partie nous avons travaillé sur l'intégration et l'analyse de données provenant de deux sources distinctes : une base de données SQL Server (SSMS) et une base Microsoft Access. La première étape a consisté à extraire et nettoyer les données de ces sources hétérogènes, en supprimant les colonnes inutiles, en renommant les champs pour harmonisation et en ajoutant des indicateurs de provenance. Nous avons ensuite fusionné ces tables, réalisé les jointures nécessaires et construit les dimensions (Dim_Employee, Dim_Client, Dim_Temps) ainsi que la table de faits, en appliquant des transformations et des calculs conditionnels pour rendre les données prêtes à l'analyse.

La deuxième partie du projet a porté sur la création du tableau de bord Power BI, avec des visualisations claires et interactives : cartes géographiques, histogrammes, graphiques en donut et courbes temporelles. Ces visualisations permettent de suivre les commandes, leur répartition géographique, l'évolution dans le temps et la performance des employés, offrant ainsi une vue décisionnelle complète du fonctionnement commercial de Northwind.

Ce projet démontre la capacité de Power BI à unifier des sources de données différentes, à automatiser le processus de transformation et à fournir des outils d'analyse puissants. Il constitue une base solide pour des projets décisionnels plus avancés et pour l'exploitation de données en contexte professionnel ou académique.