

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 d'Informatique  
Spécialité : Big Data Analytics

# Rapport Projet TBI

**Préparé par :**

**KHETTAB Wissam**

**Matricule : 212131035628**

**Travail présenté à Mme MEKAHLIA Fatma Zohra**

**Année Universitaire : 2025 / 2026**

---

## Introduction à Power BI

Power BI est une suite complète d'outils d'analyse décisionnelle développée par Microsoft. Elle permet de connecter, transformer, modéliser, visualiser et partager des données au sein d'un environnement intégré. Contrairement aux outils ETL traditionnels tels que Talend, Power BI regroupe dans une même plateforme l'acquisition des données, leur transformation, la modélisation analytique ainsi que la création de rapports interactifs.

### Composants principaux de Power BI

- **Power Query** : Composant ETL permettant l'extraction, la transformation et le chargement des données.
- **Power Pivot** : Moteur de modélisation permettant de construire des modèles relationnels et des mesures DAX.
- **Power View / Power Map** : Outils de visualisation interactive pour la création de tableaux de bord.
- **Power BI Service** : Plateforme cloud dédiée à la publication, au partage et à la collaboration autour des rapports.

### Différence entre Talend et Power BI

Talend est un outil ETL spécialisé, principalement orienté développement, permettant de créer des flux d'intégration complexes pour des environnements data warehouses. Power BI, quant à lui, est une plateforme BI complète intégrant dans un même espace un ETL (Power Query), un moteur de modélisation (Power Pivot) et des outils de visualisation. Alors que Talend exige une expertise technique plus élevée et une approche procédurale, Power BI offre une interface visuelle intuitive, plus accessible et orientée analyse. Pour notre projet académique, Power BI constitue ainsi le choix le plus pertinent : il réunit extraction, transformation, modélisation et visualisation dans un seul outil, simplifiant le processus et permettant de démontrer efficacement le cycle complet d'un mini data warehouse, tout en étant gratuit dans sa version Desktop.

## Introduction à la Phase d'Extraction des Données

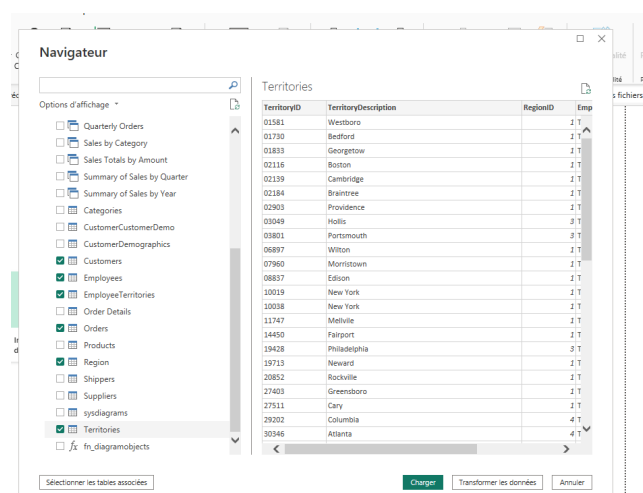
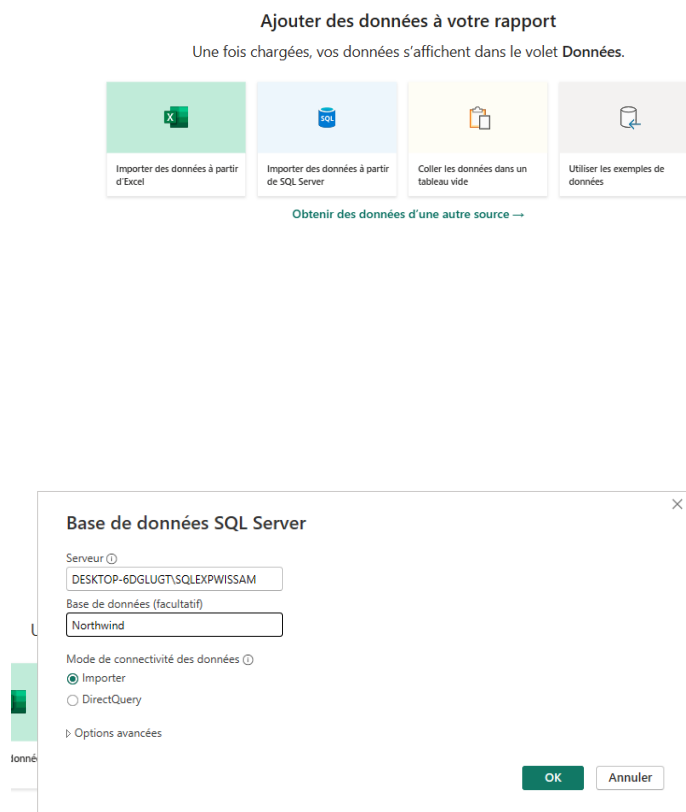
Dans le cadre de ce projet d'Intelligence d'Affaires, l'objectif principal est de mettre en œuvre un processus ETL complet permettant la conception d'un mini entrepôt de données basé sur un modèle en étoile. Ce modèle a pour but de faciliter l'analyse décisionnelle à travers l'intégration, la structuration et l'exploitation de données provenant de sources hétérogènes, en vue de produire des indicateurs pertinents et des tableaux de bord interactifs sous Power BI.

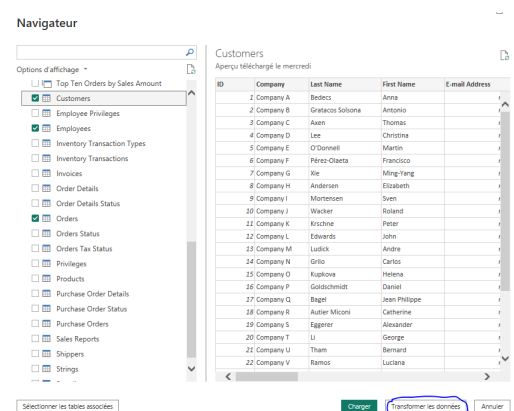
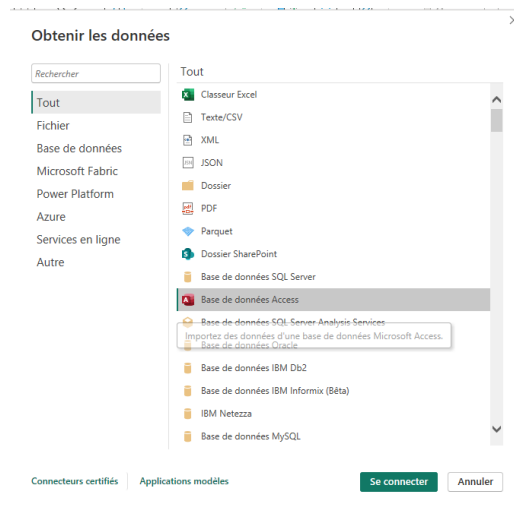
Avant de lancer le processus ETL, une étape préliminaire a été réalisée afin d'identifier et d'importer uniquement les tables nécessaires à la réalisation de notre projet. Cette approche permet de réduire le volume de données traitées, d'optimiser les performances des transformations et d'assurer la cohérence du modèle décisionnel.

Dans ce contexte, deux sources de données ont été exploitées :

- Une base de données relationnelle hébergée sur *SQL Server*, utilisée comme source principale de production.
- Une base de données *Access*, exportée sous *ACCESS*, servant de source complémentaire.

Les tables importées ont été sélectionnées en fonction de leur pertinence métier et de leur contribution directe à la construction des dimensions et de la table de faits du modèle en étoile. Cette phase d'extraction ciblée constitue le socle du processus ETL et garantit une transformation, une intégration et une modélisation des données efficaces et maîtrisées dans Power BI.





## Partie 1 : Construction de la Table Dim\_Employee

### Étape 1 : Importation de la Table *Employees* depuis SQL Server et Access

**Objectif :** Importer les données des employés à partir des deux sources de production.

#### Résultat :

- Deux tables *Employees* importées (SQL Server et Access).
- Données prêtes pour transformation dans Power Query.

### Étape 2 : Sélection et Renommage des Colonnes

**Objectif :** Harmoniser la structure des deux sources.

#### Procédure :

- Conservation des colonnes suivantes dans les deux tables :
  - EmployeeID
  - FirstName
  - LastName
- Renommage des colonnes :

- EmployeeID → id\_employee\_prod
- FirstName → Nom
- LastName → Prenom

### Étape 3 : Ajout de la Colonne source\_prod

**Objectif :** Identifier l'origine de chaque enregistrement.

#### A. Source SQL Server

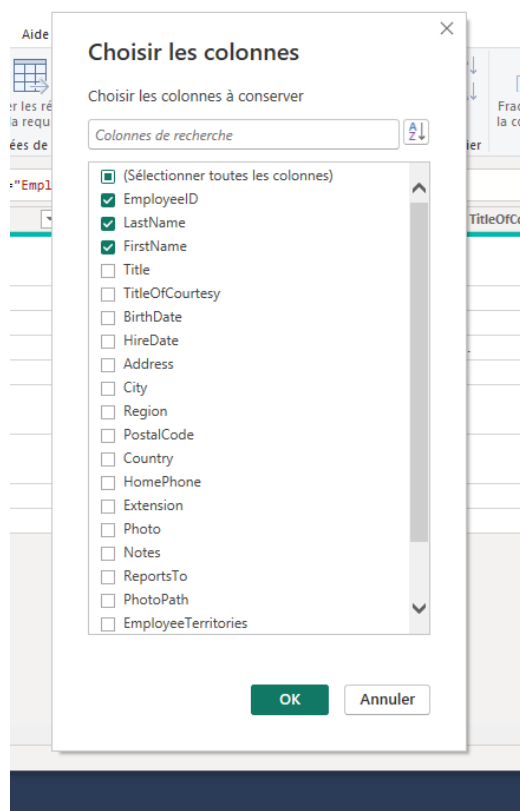
**Procédure :**

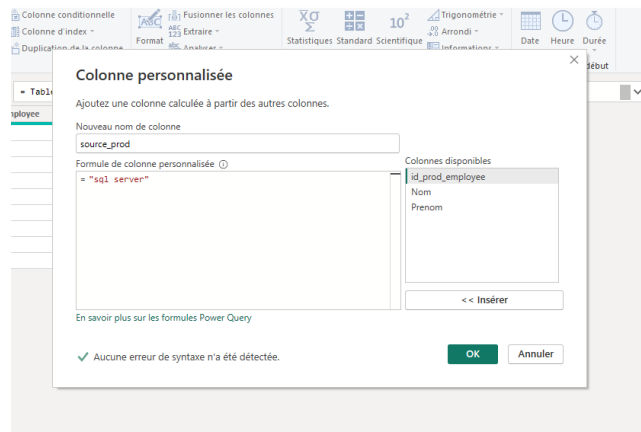
- Ajout d'une colonne personnalisée.
- Formule : "SQL\_SERVER".

#### B. Source Access

**Procédure :**

- Ajout d'une colonne personnalisée.
- Formule : "ACCESS".





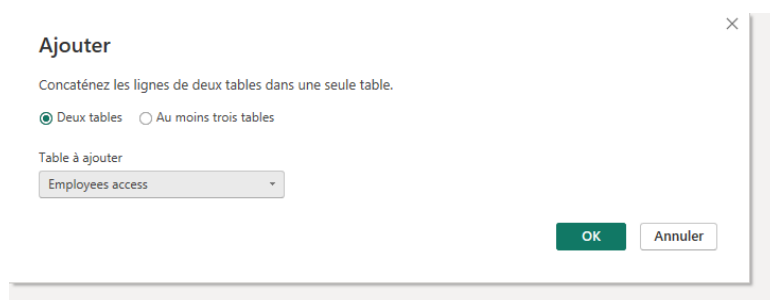
personnalisée Duplication de la colonne Analyser À partir d'un fichier texte À partir d'un nombre Informations

Général

fx = Table.AddColumn("#Colonnes renommées", "source\_prod", each "access")

	id_employee_prod	Nom	Prenom	source_prod
1	1	Freehafer	Nancy	access
2	2	Cencini	Andrew	access
3	3	Kotas	Jan	access
4	4	Sergienko	Mariya	access
5	5	Thorpe	Steven	access
6	6	Neipper	Michael	access
7	7	Zare	Robert	access
8	8	Giussani	Laura	access
9	9	Hellung-Larsen	Anne	access

**En suite on fait l union des deux table**

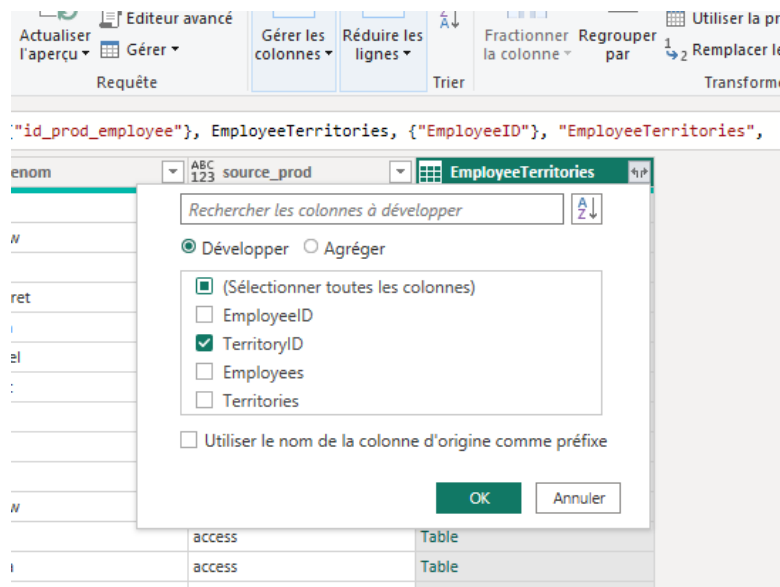
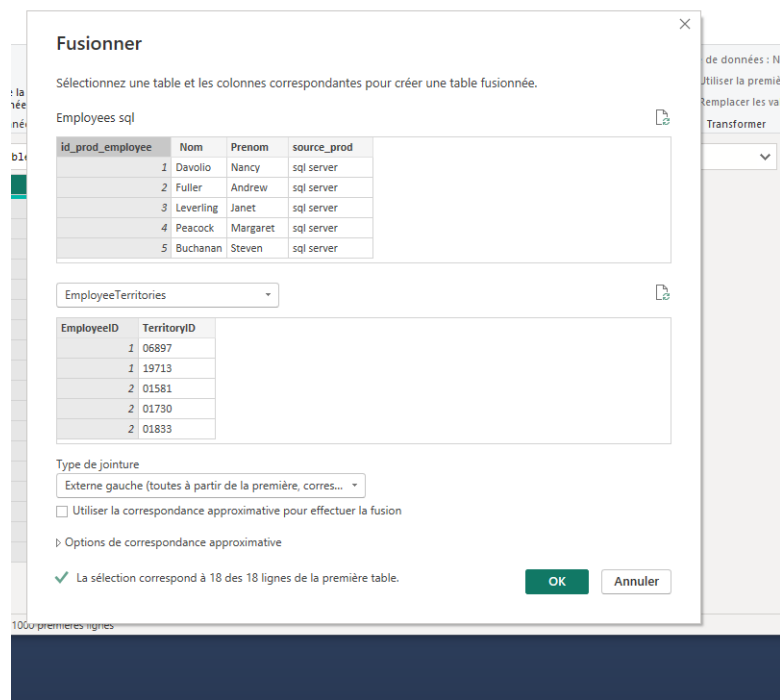


## Étape 4 : Jointure avec la Table *EmployeeTerritories*

**Objectif :** Associer chaque employé à ses territoires.

### Procédure :

- Sélection de la table *Employees* (SQL Server).
- Fusion avec la table *EmployeeTerritories*.
- Colonne de jointure : EmployeeID.
- Type de jointure : Externe gauche.
- Développement de la colonne fusionnée :
  - Conservation de la colonne *TerritoryID*.



## Étape 5 : Jointure avec la Table *Territories*

**Objectif :** Obtenir les informations géographiques des territoires.

### Procédure :

- Fusion avec la table *Territories*.
- Colonne de jointure : TerritoryID.
- Type : Externe gauche.
- Colonnes conservées :
  - TerritoryDescription
  - RegionID

**Fusionner**

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

Employees sql

id_prod_employee	Nom	Prenom	source_prod	TerritoryID
1	Davolio	Nancy	sql server	06897
1	Davolio	Nancy	sql server	19713
2	Fuller	Andrew	sql server	01581
2	Fuller	Andrew	sql server	01730
2	Fuller	Andrew	sql server	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

Rechercher les colonnes à développer

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)

☐ TerritoryID

☒ TerritoryDescription

☒ RegionID

☐ EmployeeTerritories

☐ Region

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

OK Annuler

33607	Table
20852	Table
27403	Table
27511	Table

## Étape 6 : Jointure avec la Table *Region*

**Objectif :** Ajouter l'information régionale.

### Procédure :

- Fusion avec la table *Region*.
- Colonne de jointure : RegionID.
- Type : Externe gauche.
- Colonne conservée :
  - RegionDescription



**Fusionner**

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

Employees sql

id_prod_employee	Nom	Prenom	source_prod	TerritoryID	TerritoryDescription	RegionID
1	Davolio	Nancy	sql server	06897	Wilton	1
1	Davolio	Nancy	sql server	19713	Neward	1
2	Fuller	Andrew	sql server	01581	Westboro	1
2	Fuller	Andrew	sql server	01730	Bedford	1
2	Fuller	Andrew	sql server	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

Rechercher les colonnes à développer

☒ Développer ☐ Agréger

☒ (Sélectionner toutes les colonnes)

☐ RegionID

☒ RegionDescription

☐ Territories

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

OK Annuler

## Étape 7 : Renommage Final des Colonnes

**Objectif :** Clarifier la sémantique des attributs.

### Procédure :

- Renommage :
  - RegionDescription → Region
  - TerritoryDescription → Territory

## Étape 8 : Ajout de la Clé Substitut

**Objectif :** Créer une clé technique pour la dimension.

### Procédure :

- Ajout d'une colonne index.
- Nom de la colonne : id\_seq\_employee.

Table: RenameColumns(#'Index #fourt4', (('Index', "id\_seq\_employee")))

source_prod	TerritoryID	Territory	RegionID	Region	id_seq_employee
sql server	06897	Watson	1	Eastern	1
sql server	19713	Neward	1	Eastern	2
sql server	01581	Westboro	1	Eastern	3
sql server	01730	Bedford	1	Eastern	4
sql server	01833	Georgetown	1	Eastern	5
sql server	02116	Boston	1	Eastern	6
sql server	02139	Cambridge	1	Eastern	7
sql server	02184	Braintree	1	Eastern	8
sql server	40222	Louisville	1	Eastern	9
sql server	30946	Atlanta	4	Southern	10
sql server	31406	Savannah	4	Southern	11
sql server	32859	Orlando	4	Southern	12
sql server	33607	Tampa	4	Southern	13
sql server	20852	Rockville	1	Eastern	14
sql server	27403	Greensboro	1	Eastern	15
sql server	27511	Cary	1	Eastern	16
sql server	02903	Providence	1	Eastern	17
sql server	07960	Morrisown	1	Eastern	18
sql server	08837	Edison	1	Eastern	19
sql server	10019	New York	1	Eastern	20
sql server	10038	New York	1	Eastern	21
sql server	11747	Melville	1	Eastern	22
sql server	14450	Fairport	1	Eastern	23
sql server	85014	Phoenix	2	Western	24
sql server	85251	Scottsdale	2	Western	25
sql server	98004	Bellevue	2	Western	26
sql server	98052	Redmond	2	Western	27
sql server	98104	Seattle	2	Western	28
sql server	60179	Hoffman Estates	2	Western	29

## Étape 9 : Finalisation et Chargement

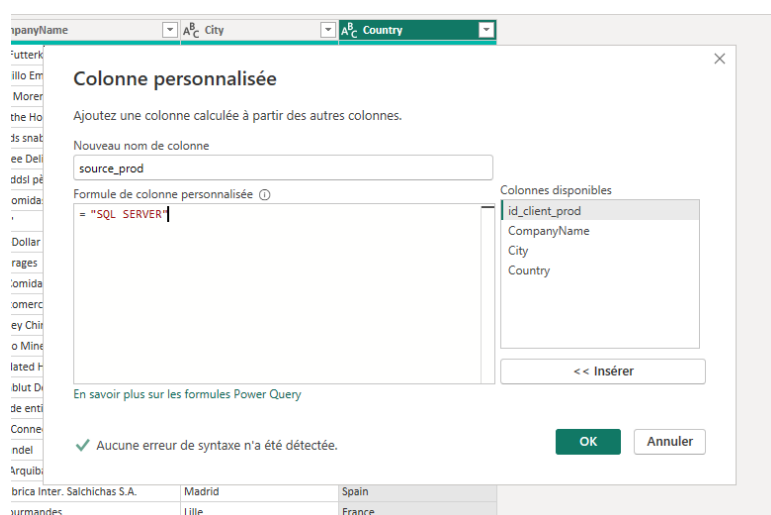
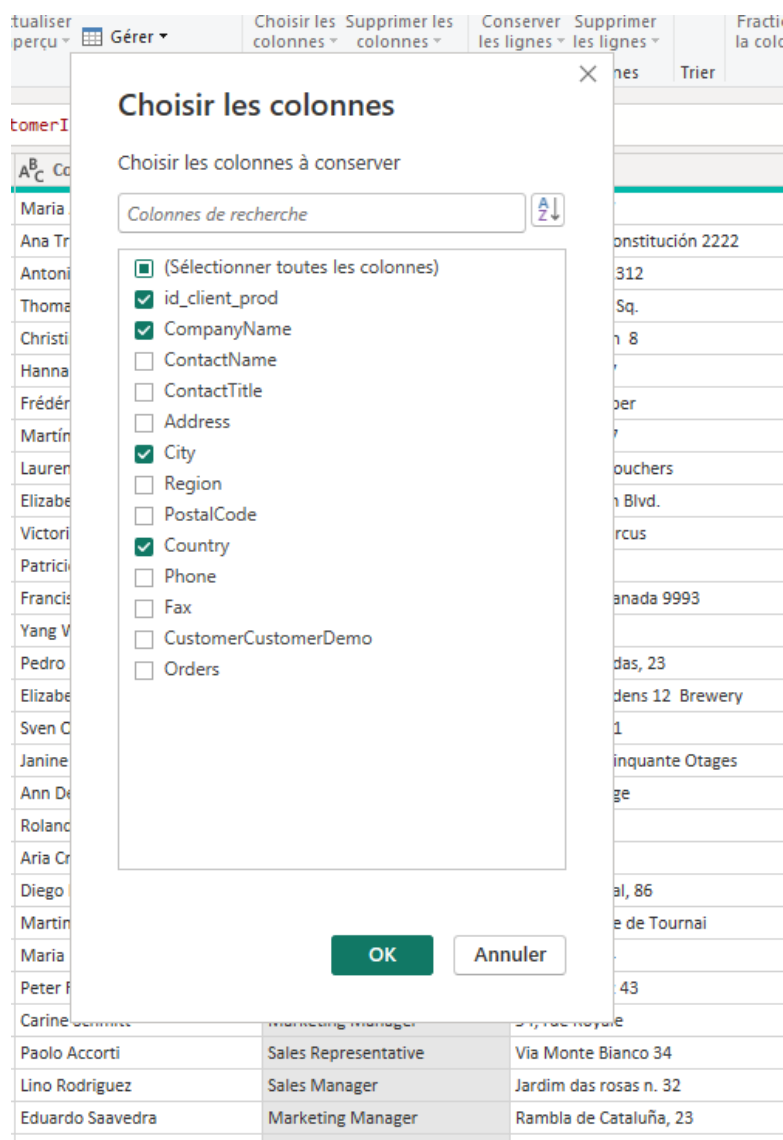
- Vérification de la cohérence des données.
- Fichier → Fermer et appliquer.
- Chargement de la table Dim\_Employee dans le modèle Power BI.

## Construction de la Dimension Client (Dim\_Client)

La dimension **Dim\_Client** a été élaborée à partir de deux sources hétérogènes (SQL Server et ACCESS). L'objectif était d'obtenir une table propre, harmonisée et prête à être intégrée dans le modèle décisionnel.

### Étape 1 : Préparation de la Source SQL Server

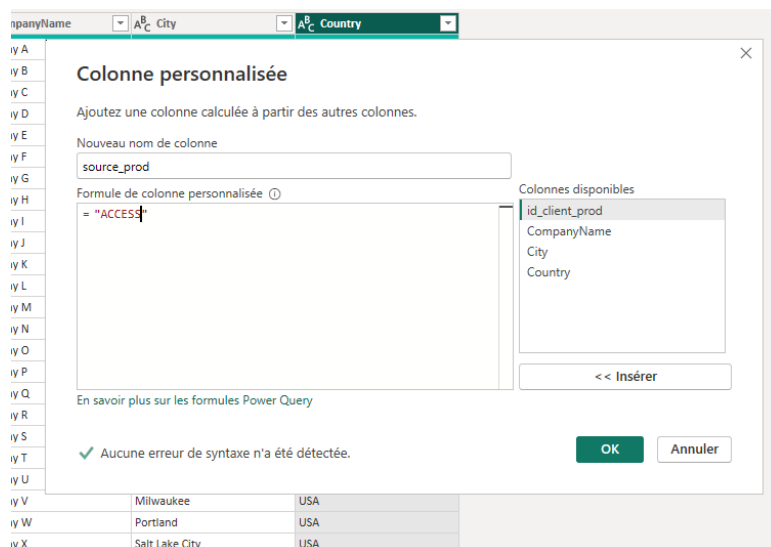
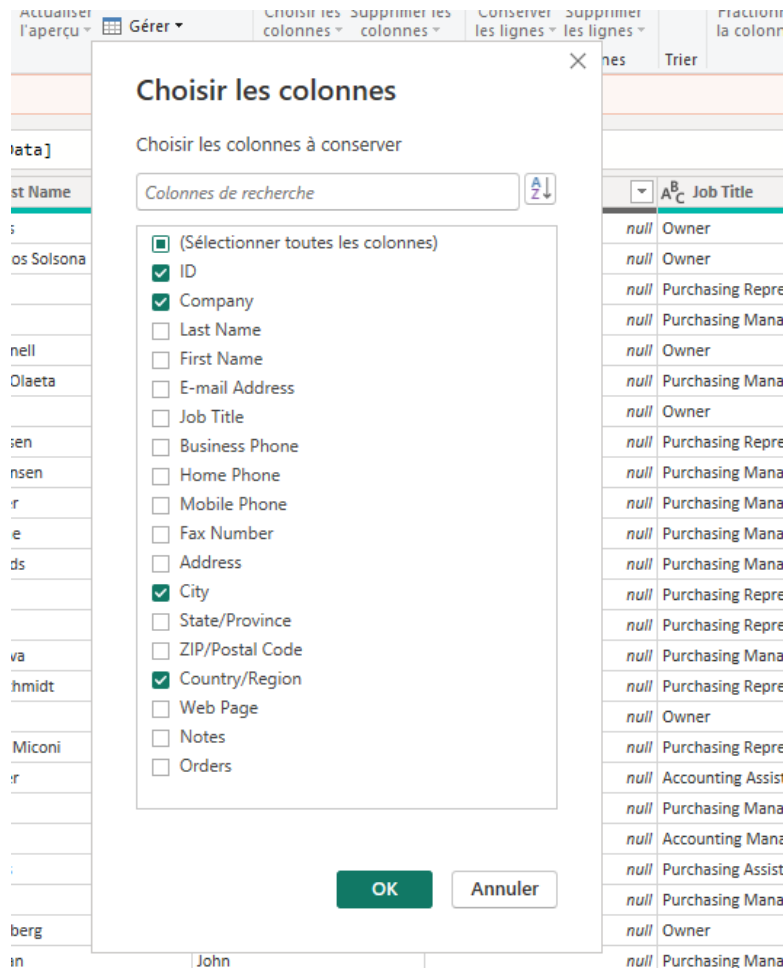
Les données clients ont été extraites depuis la base Northwind via *Accueil → Obtenir des données → SQL Server*. La table **Customers** a été importée dans Power Query, puis épurée afin de conserver uniquement les attributs nécessaires : *CustomerID*, *CompanyName*, *City*, *Country*. été renommées pour adopter une nomenclature uniforme (*CustomerID* → *id\_client\_prod*) et une colonne supplémentaire *source\_prod* contenant la valeur "SQL\_SERVER" a été créée. La requête résultante a été nommée **Customers\_SQL**.



## Étape 2 : Préparation de la Source ACCESS

Un fichier ACCESS complémentaire a été importé via *Obtenir des données* → *ACCESS*. Après sélection des colonnes pertinentes (*ID*, *Company*, *City*, *Country*, *Region*), une harmonisation structurelle

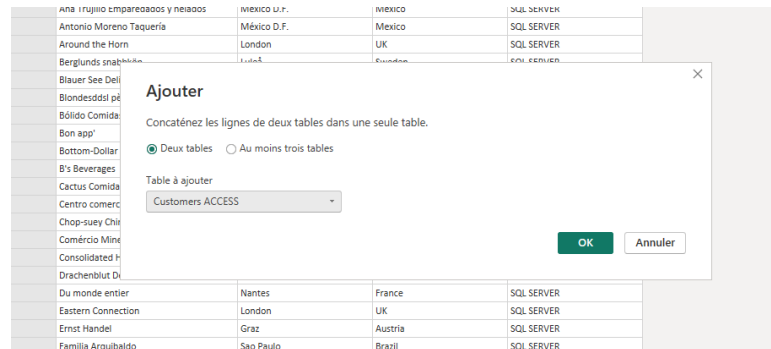
a été réalisée :  $ID \rightarrow id\_client\_prod$ ,  $Company \rightarrow CompanyName$ ,  $Country\_Region \rightarrow Country$ . Une colonne  $source\_prod = "ACCESS"$  a été ajoutée afin de tracer l'origine de chaque enregistrement, et la requête a été renommée **Customers\_ACCESS**.



### Étape 3 : Union des Deux Sources

Une référence de **Customers\_SQL** a été créée sous le nom **Dim\_Client**. L'union des deux sources a ensuite été effectuée via *Accueil* → *Combiner* → *Ajouter des requêtes*, en ajoutant la table **Custo-**

**mers\_CSV**. Cette opération a permis de consolider les données SQL et ACCESS dans une structure unifiée, avec alignement automatique des colonnes harmonisées.



## Étape 4 : Ajout d'un Index Unique

Pour générer une clé de substitution stable indispensable au modèle décisionnel, une colonne d'index a été ajoutée via *Ajouter une colonne* → *Colonne d'index* (à partir de 1). Cette colonne, renommée *id\_dim\_client*, constitue la clé primaire de la dimension.

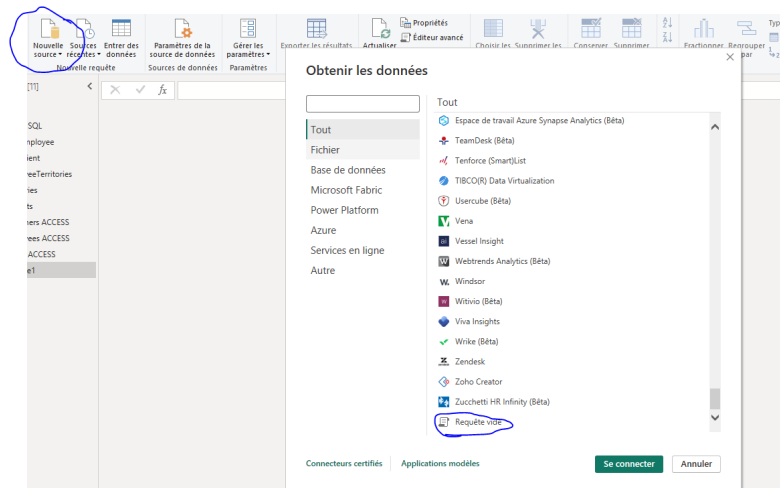
	id_seq_client	id_client_prod	CompanyName	City	Country	source_prod
1	1	1	Alfreds Futterkiste	Berlin	Germany	SQL SERVER
2	2	2	Ana Trujillo Emparedados y helados	México D.F.	Mexico	SQL SERVER
3	3	3	Antonio Moreno Taquería	London	UK	SQL SERVER
4	4	4	Arround the Horn	London	UK	SQL SERVER
5	5	5	Berglunds snabbköp	Luleå	Sweden	SQL SERVER
6	6	6	Blauer See Delikatessen	Mannheim	Germany	SQL SERVER
7	7	7	Blondies père et fils	Strasbourg	France	SQL SERVER
8	8	8	Bólido Comidas preparadas	Madrid	Spain	SQL SERVER
9	9	9	Bon app'	Marseille	France	SQL SERVER
10	10	10	Bottom-Dollar Markets	Toronto	Canada	SQL SERVER
11	11	11	B's Beverages	London	UK	SQL SERVER
12	12	12	Cactus Comidas para llevar	Buenos Aires	Argentina	SQL SERVER
13	13	13	Centro comercial Moctezuma	México D.F.	Mexico	SQL SERVER
14	14	14	Chop-suey Chinese	Bern	Switzerland	SQL SERVER
15	15	15	Comércio Mineiro	Sao Paulo	Brazil	SQL SERVER
16	16	16	Consolidated Holdings	London	UK	SQL SERVER
17	17	17	Drachenblut Delikatessen	Aachen	Germany	SQL SERVER
18	18	18	Du monde entier	Nantes	France	SQL SERVER
19	19	19	Eastern Connection	London	UK	SQL SERVER
20	20	20	Ernst Handel	Graz	Austria	SQL SERVER
21	21	21	Familia Arquibaldo	Sao Paulo	Brazil	SQL SERVER
22	22	22	FISSA Fabrica Inter. Saichicas S.A.	Madrid	Spain	SQL SERVER
23	23	23	Foiles gourmandes	Liège	France	SQL SERVER
24	24	24	Folk och fä HB	Bräcke	Sweden	SQL SERVER
25	25	25	Frankenversand	München	Germany	SQL SERVER
26	26	26	France restauration	Nantes	France	SQL SERVER
27	27	27	Franchi S.p.A.	Torino	Italy	SQL SERVER
28	28	28	Furia Bacalhou e Frutos do Mar	Lisboa	Portugal	SQL SERVER

## Nettoyage et Finalisation de Dim\_Client

Après l'union des données SQL Server et ACCESS, un nettoyage complet a été réalisé en supprimant les doublons (sur *id\_client\_prod* et *source\_prod*), en corrigeant les valeurs nulles, en normalisant les formats de *City* et *Country*, et en validant les types de données afin d'assurer la cohérence du modèle. Les colonnes ont ensuite été réorganisées avant de charger la table dans le modèle via *Fermer et appliquer*, permettant à **Dim\_Client** d'apparaître dans la zone de modélisation. Une vérification finale a confirmé l'intégrité de la clé *id\_dim\_client*. La dimension obtenue est ainsi propre, structurée et parfaitement prête pour les analyses Power BI.

## Table de Dimension Temps : Création d'une nouvelle requête

Nous avons créé une nouvelle requête dans Power Query afin de générer la table de dimension temps. Cette requête a été initialisée comme une requête vide.



## Écriture du code M

Le code M a été écrit dans la barre de formule avancée pour générer la table de dates souhaitée :



✕ ✓ *fx* = Table.RemoveColumns(#"Removed Duplicates",{ "Date"})

	A <sub>C</sub> année	A <sub>C</sub> mois_annee
1	1990	1990-01
2	1990	1990-02
3	1990	1990-03
4	1990	1990-04
5	1990	1990-05
6	1990	1990-06
7	1990	1990-07
8	1990	1990-08
9	1990	1990-09
10	1990	1990-10
11	1990	1990-11
12	1990	1990-12
13	1991	1991-01
14	1991	1991-02
15	1991	1991-03
16	1991	1991-04
17	1991	1991-05
18	1991	1991-06
19	1991	1991-07
20	1991	1991-08
21	1991	1991-09
22	1991	1991-10
23	1991	1991-11
24	1991	1991-12
25	1992	1992-01
26	1992	1992-02
27	1992	1992-03
28	1992	1992-04
29	1992	1992-05
30	1992	1992-06

de la colonne en fonction des 1000 premières lignes

Cette requête permet de créer automatiquement une table contenant toutes les dates comprises entre le 1er janvier 1990 et le 31 décembre 1999, ce qui servira de dimension temporelle pour nos analyses.

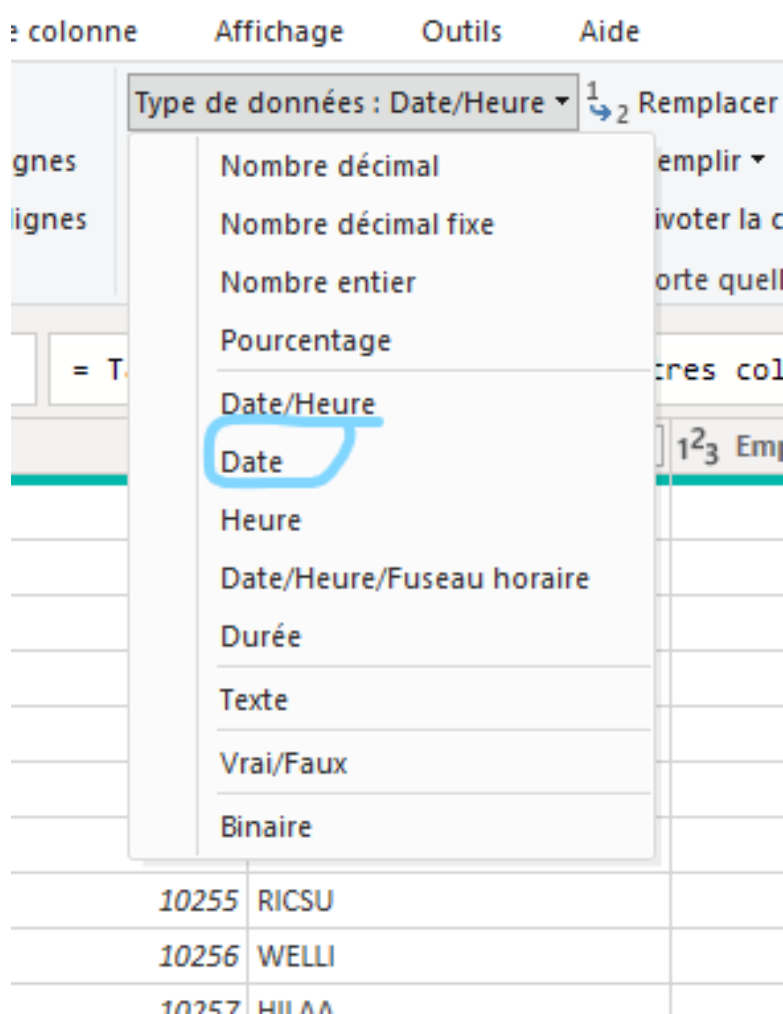
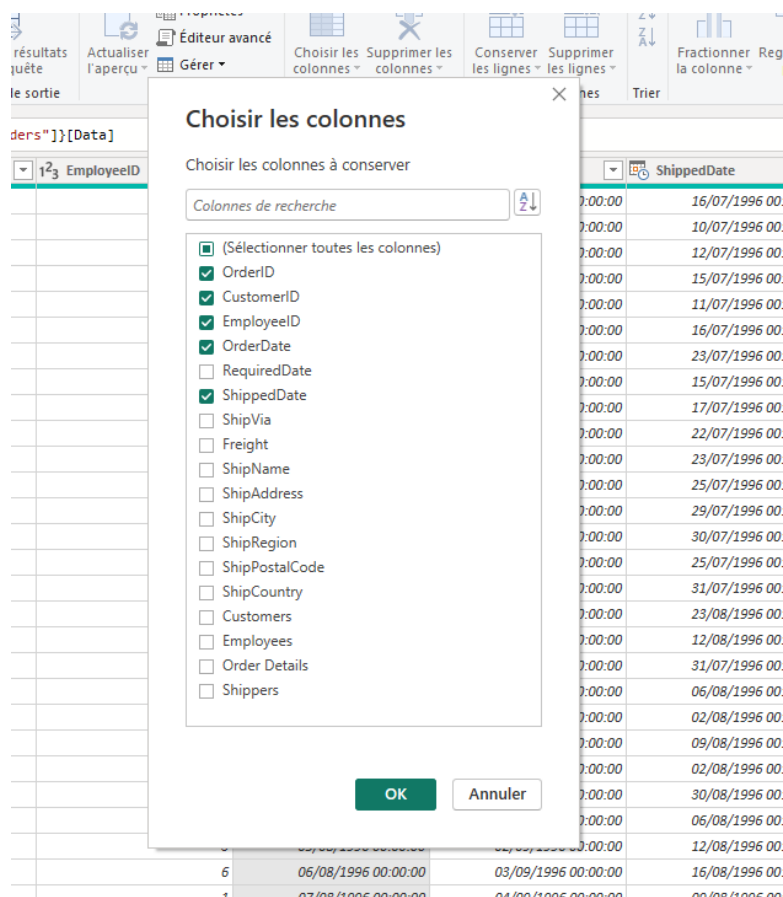
## Partie 2 : Construction de la Table de Faits TF\_Commande

### Procédure de Préparation des Données

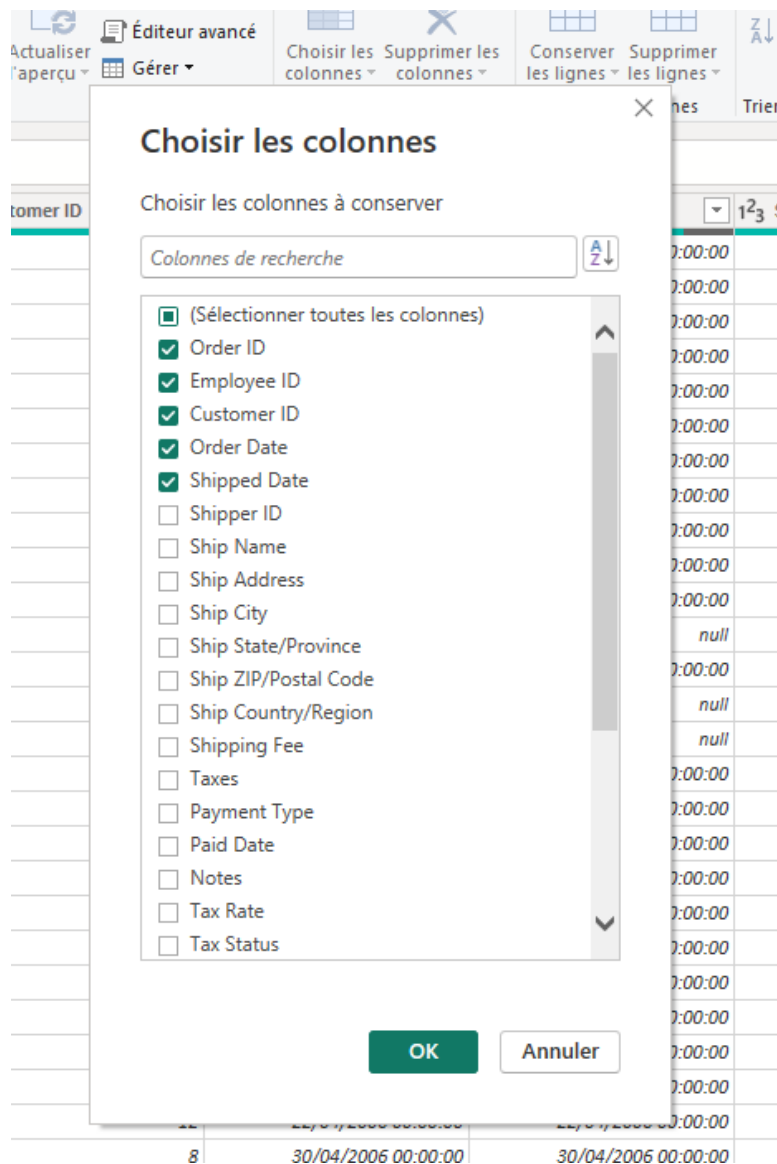
#### Connexion et Importation des Sources

Deux sources distinctes ont été utilisées pour récupérer les commandes :

- **SQL Server** : Connexion au serveur SQL et importation de la table *Orders* de la base *Northwind*. La requête obtenue a été renommée en **Orders\_SQL**.
- **Fichier ACCESS** : Importation du fichier *ACCESS* contenant les commandes exportées depuis Access. La requête correspondante a été renommée en **Orders\_ACCESS**.







## Union des Deux Tables

Une nouvelle requête vide, nommée **Faits\_Source**, a été créée pour centraliser les données. La fusion a été réalisée en utilisant l'opération *Ajouter des lignes*, permettant de :

- sélectionner les tables **Orders\_SQL** et **Orders\_ACCESS**,
- combiner leurs enregistrements en une seule table,
- garantir un alignement automatique des colonnes communes.

Le résultat final est une table unifiée regroupant l'ensemble des commandes provenant à la fois de SQL Server et du fichier ACCESS.

NET	5	04/07/1996	16/07/1996
IMSP	6	05/07/1996	10/07/1996
INAR	4	08/07/1996	12/07/1996
CTE	3	08/07/1996	15/07/1996
IPRD	4	08/07/1996	11/07/1996
INAR			
IOPS			
CSU			
ELU			
LAA			
INSH			
INTC			
ITIK			
JEDE			
ITTC			
INSH			
ILKO			
ONP	2	25/07/1996	12/08/1996
ARTH	3	26/07/1996	31/07/1996
ANK	4	29/07/1996	06/08/1996
IOSR	8	30/07/1996	02/08/1996
HITC	5	31/07/1996	09/08/1996

Ajouter

Concaténez les lignes de deux tables dans une seule table.

☒ Deux tables
☐ Au moins trois tables

Table à ajouter

Orders Accesss

OK
Annuler

## Étape 2 : Création de la Clé Temporelle

### Objectif

L'objectif de cette étape est de générer une clé temporelle au format YYYY-MM, indispensable pour assurer la jointure entre la table des faits et la dimension temporelle *Dim\_Temps*.

### Procédure

Une colonne personnalisée a été ajoutée dans Power Query afin d'extraire le mois et l'année de la colonne *OrderDate*. La formule M utilisée est la suivante :

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

La colonne obtenue a été nommée **mois\_annee**. Elle jouera le rôle de clé de liaison entre la table *Faits\_Source* et la dimension temporelle *Dim\_Temps*.

{#"Colonnes supprimées", #"Orders Accesss"}

CustomerID
EmployeeID
OrderDate
ShippedDate

Colonne personnalisée

Ajoutez une colonne calculée à partir des autres colonnes.

Nouveau nom de colonne

mois\_annee

Formule de colonne personnalisée

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

Colonnes disponibles

OrderID
CustomerID
EmployeeID
OrderDate
ShippedDate

< < Insérer

En savoir plus sur les formules Power Query

✓ Aucune erreur de syntaxe n'a été détectée.

OK
Annuler

# les jointure

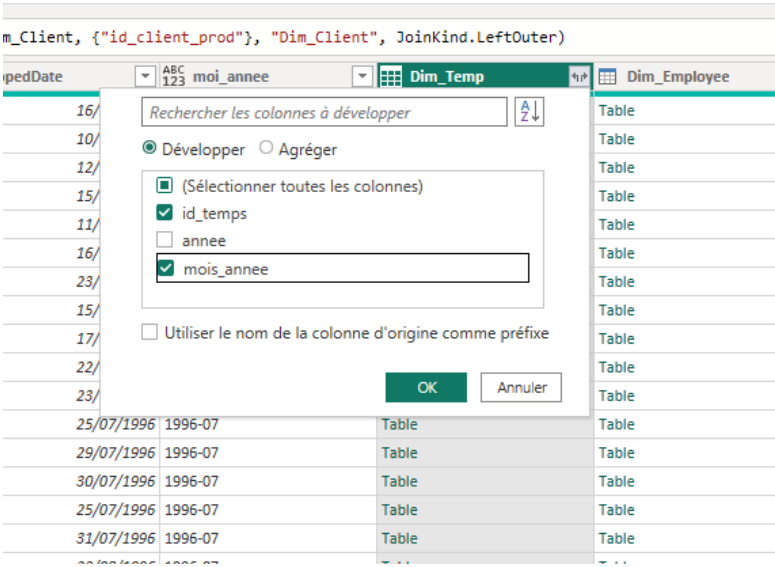
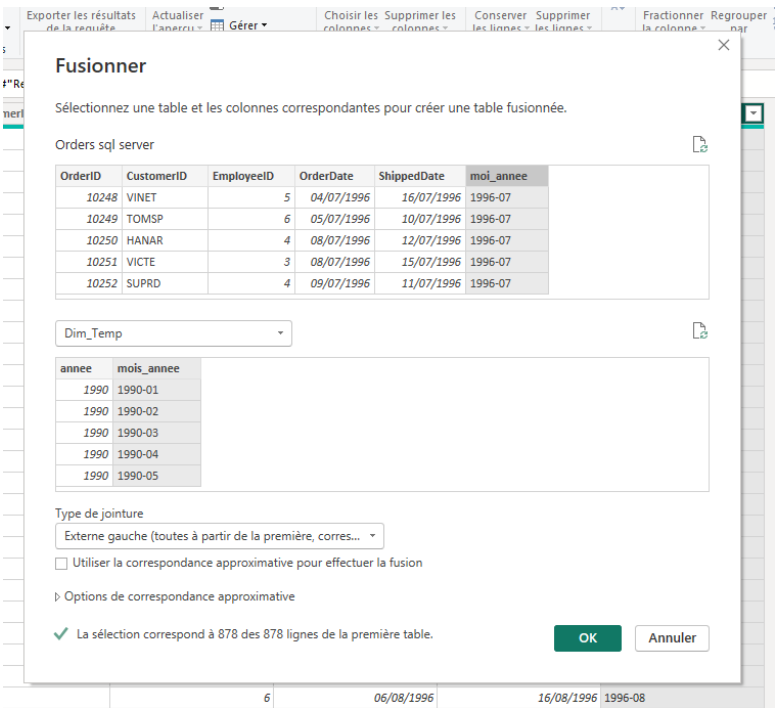
## Étape 3 : Intégration des Dimensions

### Jointure avec la Dimension Temps

La première fusion a été réalisée entre la table *Faits\_Source* et la dimension *Dim\_Temps*. La jointure a été effectuée sur la colonne temporelle :

— **Faits\_Source.mois\_annee = Dim\_Temps.MoisAnnee.**

Le type de jointure utilisé est une **jointure gauche externe**, garantissant la conservation de toutes les lignes de la table des faits. La colonne issue de cette fusion est : **id\_temps**.



## Jointure avec la Dimension Client

Une deuxième fusion a été réalisée avec la dimension *Dim\_Client*. La correspondance a été faite sur la clé fonctionnelle :

— **CustomerID = id\_client\_prod.**

Ici également, une **jointure gauche externe** a été appliquée. La colonne ajoutée à la table des faits est : **id\_seqClient**.

**Fusionner**

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

Orders sql server

OrderID	CustomerID	EmployeeID	OrderDate	ShippedDate	moi_annee
10248	VINET	5	04/07/1996	16/07/1996	1996-07
10249	TOMSP	6	05/07/1996	10/07/1996	1996-07
10250	HANAR	4	08/07/1996	12/07/1996	1996-07
10251	VICTE	3	08/07/1996	15/07/1996	1996-07
10252	SUPRD	4	09/07/1996	11/07/1996	1996-07

Dim\_Client

id_seq_client	id_client_prod	CompanyName	City	Country	source_prod
1	ALFKI	Alfreds Futterkiste	Berlin	Germany	sql server
2	ANATR	Ana Trujillo Emparedados y helados	México D.F.	Mexico	sql server
3	ANTON	Antonio Moreno Taquería	México D.F.	Mexico	sql server
4	AROUT	Around the Horn	London	UK	sql server
5	BERGS	Berglunds snabbköp	Luleå	Sweden	sql server

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

Réduire les lignes Trier Transformer

.d\_seq\_employee"}}

Rechercher les colonnes à développer

● Développer ○ Agréger

(Sélectionner toutes les colonnes)

☒ id\_seq\_client

☐ id\_client\_prod

☐ CompanyName

☐ City

☐ Country

☐ source\_prod

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

OK Annuler

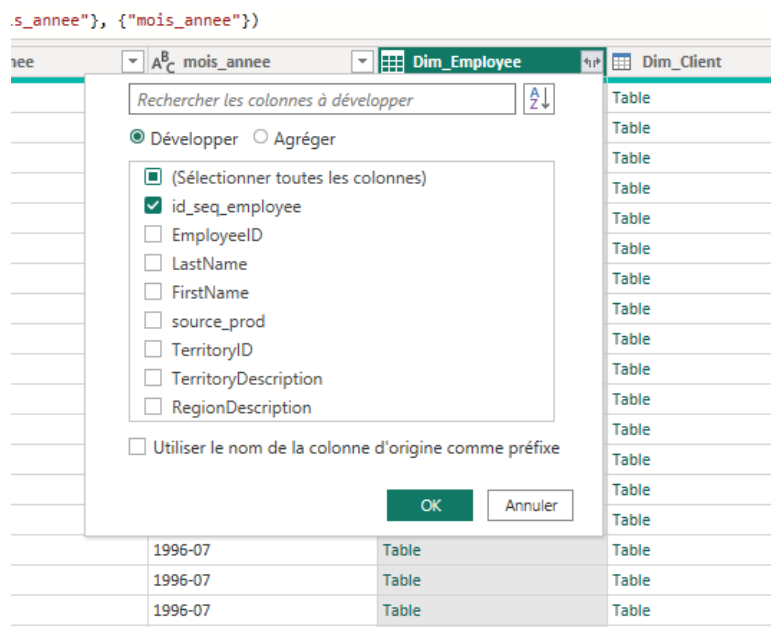
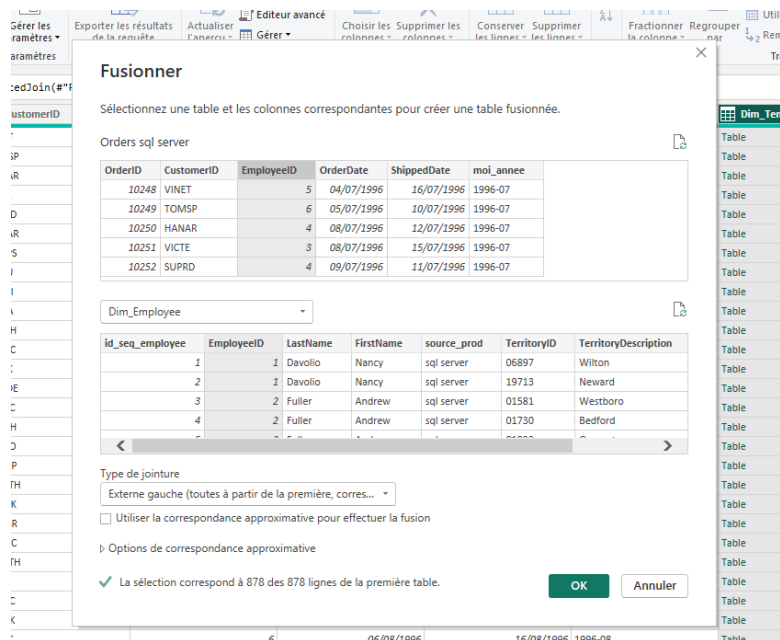
## Jointure avec la Dimension Employé

Enfin, la table *Faits\_Source* a été enrichie par une fusion avec la dimension *Dim\_Employee*. La jointure repose sur deux colonnes :

— **EmployeeID = id\_employee\_prod,**

— **TerritoryID = Territory.**

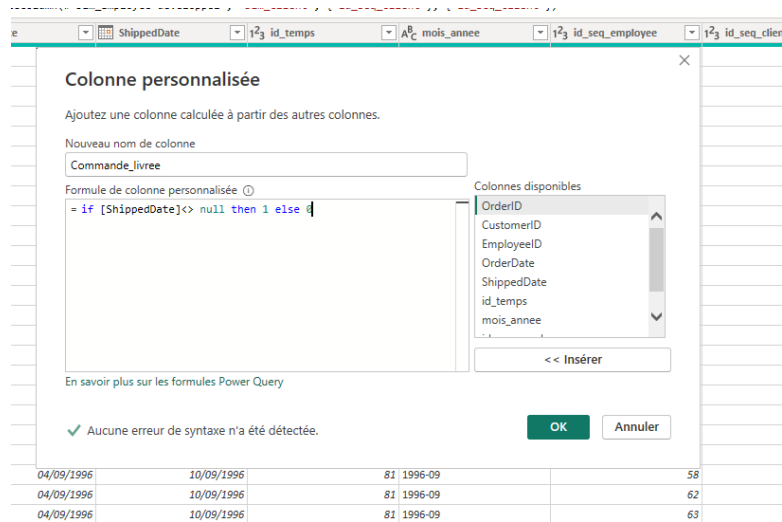
Une **jointure gauche externe** a été appliquée, permettant d'ajouter la clé de substitution associée à l'employé : **id\_seqEmployee**.



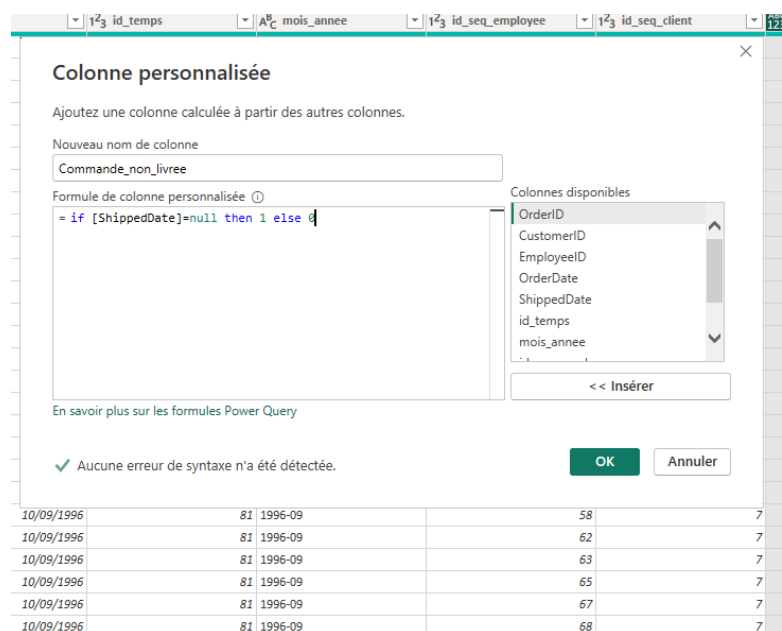
## Calcul des indicateurs binaires

Pour chaque commande, on détermine si elle est livrée ou non en fonction de la colonne ShippedDate.

**Colonne Commande\_livree :**



## Colonne Commande\_Non\_livree :



## Agrégation des données

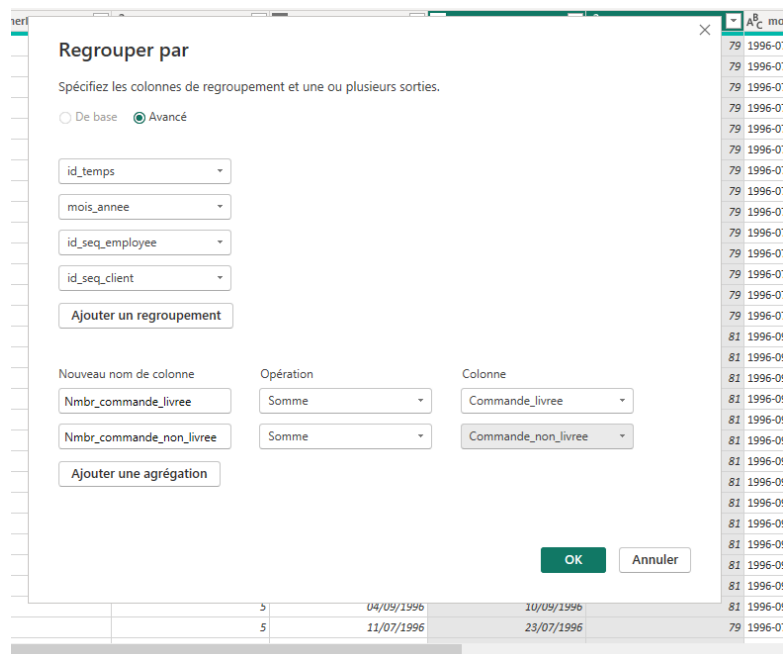
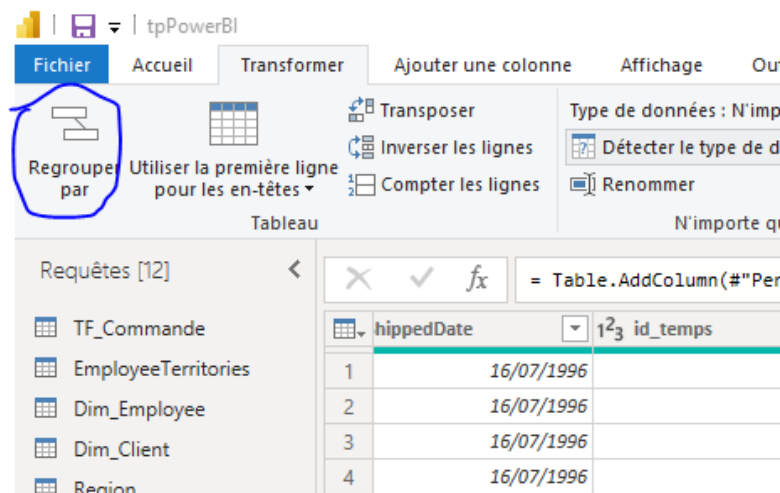
Utilisation de l'outil **Regrouper par** (Table.Group) pour agréger les données à la granularité souhaitée :

### — Clés de regroupement :

- CustomerID
- EmployeeID
- TerritoryID
- MoisAnneeProd

### — Mesures agrégées :

- nbr\_commandes\_livrees = Somme de Commande\_Livree
- nbr\_commandes\_non\_livrees = Somme de Commande\_Non\_Livree



**Résultat :** Une table de faits propre, agrégée, et reliée aux dimensions via des clés séquentielles.

Table

id_seq_client	id_seq_employee	id_seq_commande	Nbr_commande_livree	Nbr_commande_non_livree	mois_annee
1	79	17	85	1	0 1996-07
2	79	18	85	1	0 1996-07
3	79	19	85	1	0 1996-07
4	79	20	85	1	0 1996-07
5	79	21	85	1	0 1996-07
6	79	22	85	1	0 1996-07
7	79	23	85	1	0 1996-07
8	79	24	85	1	0 1996-07
9	79	25	85	1	0 1996-07
10	79	26	85	1	0 1996-07
11	79	27	85	1	0 1996-07
12	79	28	85	1	0 1996-07
13	79	29	85	1	0 1996-07
14	79	30	85	1	0 1996-07
15	81	17	7	1	0 1996-09
16	81	18	7	1	0 1996-09
17	81	19	7	1	0 1996-09
18	81	20	7	1	0 1996-09
19	81	21	7	1	0 1996-09
20	81	22	7	1	0 1996-09
21	81	23	7	1	0 1996-09
22	81	24	7	1	0 1996-09
23	81	25	7	1	0 1996-09
24	81	26	7	1	0 1996-09
25	81	27	7	1	0 1996-09
26	81	28	7	1	0 1996-09
27	81	29	7	1	0 1996-09
28	81	30	7	1	0 1996-09
29	79	17	14	1	0 1996-07

## Partie 4 : Création du Tableau de Bord et Visualisation des Données

### Modélisation et gestion des relations

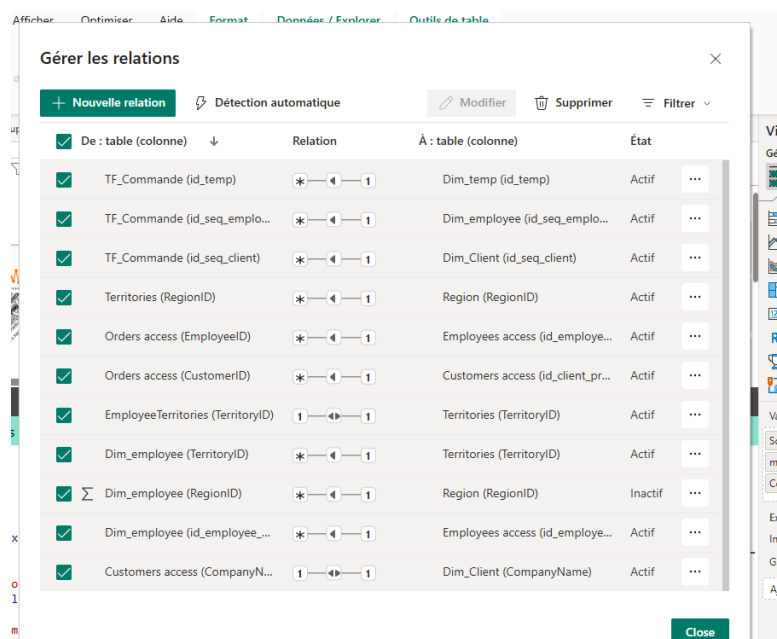
Avant la conception des visualisations, une étape fondamentale a consisté à définir et gérer les relations entre la table de faits et les tables de dimensions dans le modèle Power BI.

La table de faits **TF\_Commande** a été reliée aux différentes tables de dimensions selon un schéma en étoile :

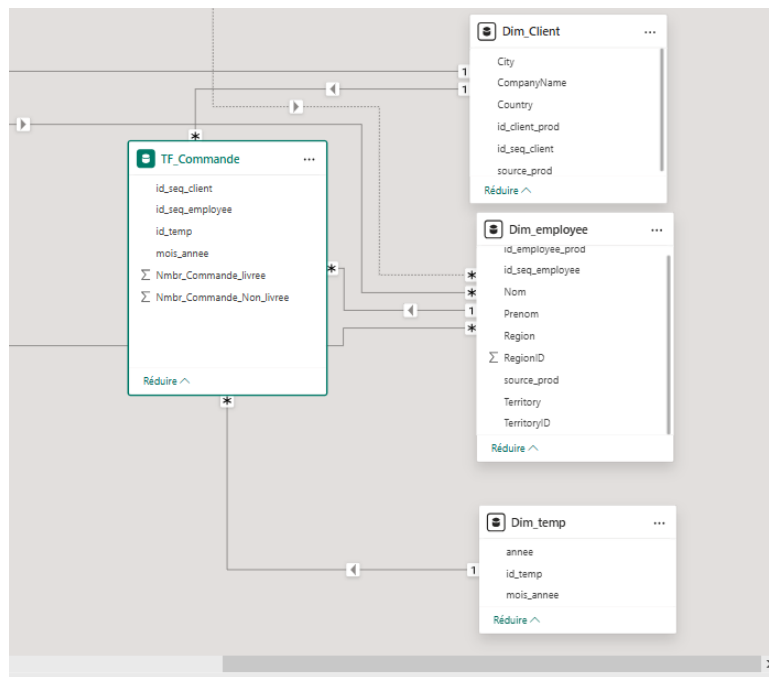
- **Dim\_Client** via la clé `id_seq_client`,
- **Dim\_Employee** via la clé `id_seq_employee`,
- **Dim\_Temps** via la clé `id_temps`.

Les relations définies sont de type *un-à-plusieurs*, orientées des dimensions vers la table de faits. Cette modélisation garantit la cohérence des agrégations, la fiabilité des indicateurs calculés ainsi qu'une navigation analytique fluide selon les axes temporel, géographique et métier.

Une fois le modèle relationnel validé, les données ont été exploitées à travers des visualisations personnalisées développées à l'aide de scripts Python intégrés dans Power BI.







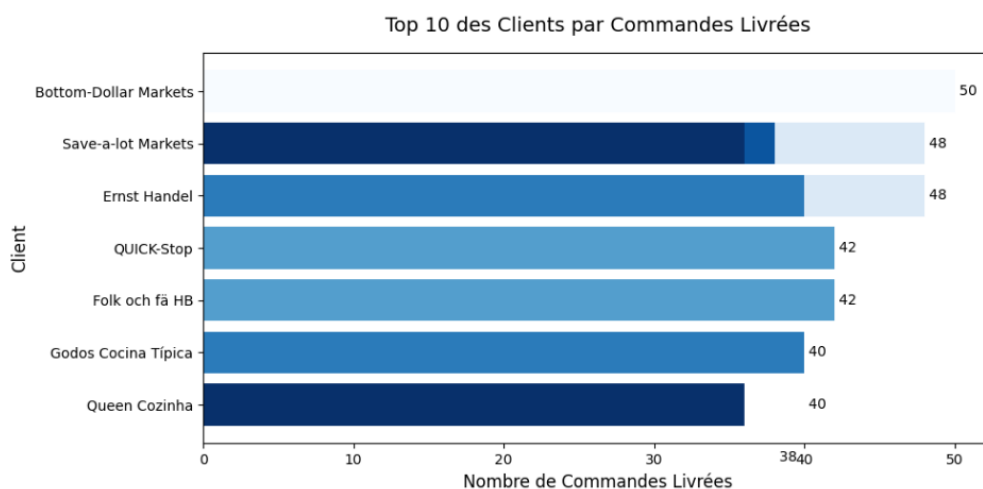
## Visualisations réalisées par scripts Python

### Top 10 des clients par commandes livrées

Cette visualisation représente les dix clients ayant enregistré le plus grand nombre de commandes livrées. Le graphique est construit sous forme de barres horizontales, triées par ordre décroissant afin de mettre en évidence les clients les plus performants.

Un dégradé de couleur est appliqué aux barres pour renforcer la lisibilité visuelle, tandis que les valeurs numériques sont affichées directement sur chaque barre.

**Objectif analytique :** identifier les clients stratégiques générant le plus fort volume de commandes livrées.

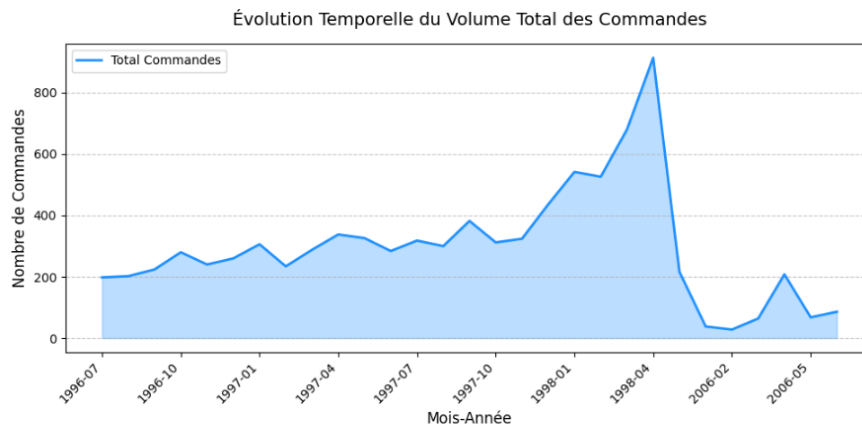


### Évolution temporelle du volume total des commandes

Ce graphique en courbe avec zone remplie illustre l'évolution mensuelle du volume total des commandes. Le volume total est calculé comme la somme des commandes livrées et non livrées.

L'axe horizontal représente la dimension temporelle (mois–année), tandis que l'axe vertical indique le nombre total de commandes. La zone remplie permet de visualiser clairement les tendances globales et les variations dans le temps.

**Objectif analytique :** analyser l'évolution de l'activité commerciale et détecter les périodes de hausse ou de baisse du volume de commandes.

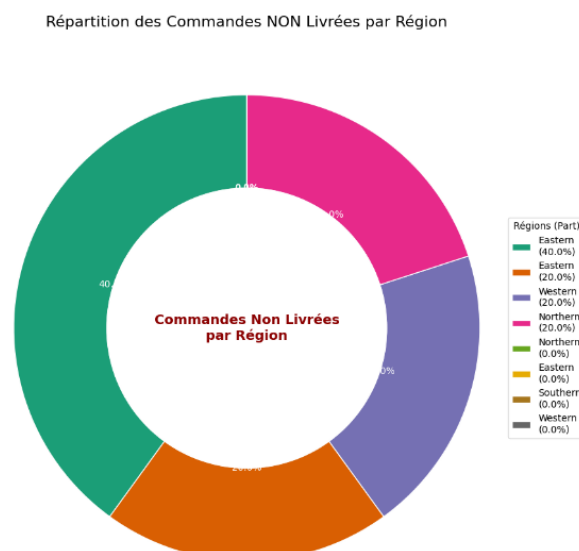


## Répartition des commandes non livrées par région

Cette visualisation prend la forme d'un diagramme en anneau (donut chart) et présente la répartition des commandes non livrées par région.

Chaque segment représente une région, accompagné de son pourcentage par rapport au total des commandes non livrées. Une légende détaillée permet d'identifier précisément la contribution de chaque région.

**Objectif analytique :** identifier les régions présentant le plus de problèmes de livraison afin de cibler les actions correctives.

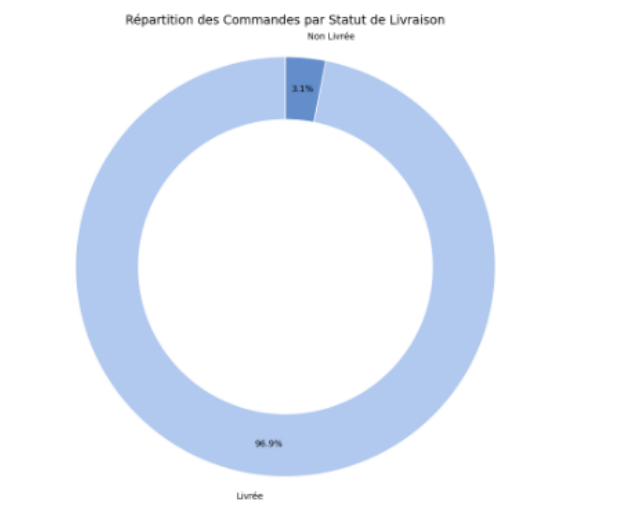


## Comparaison globale des commandes livrées et non livrées

Ce graphique en anneau compare la proportion globale des commandes livrées et des commandes non livrées.

Les données sont agrégées à l'échelle globale, ce qui permet d'obtenir une vision synthétique de la performance du processus de livraison.

**Objectif analytique :** mesurer l'efficacité globale de la chaîne de livraison et évaluer le taux de satisfaction logistique.

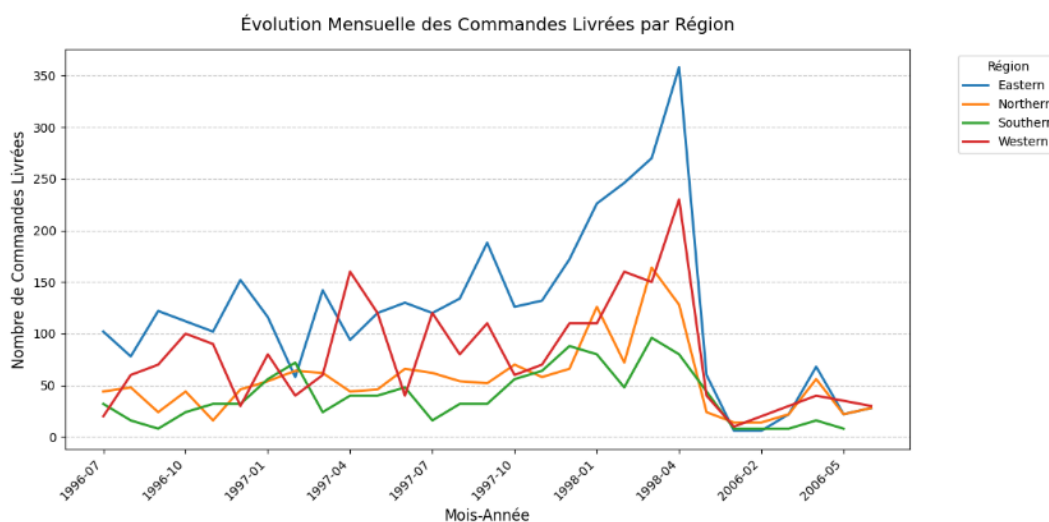


## Évolution mensuelle des commandes livrées par région

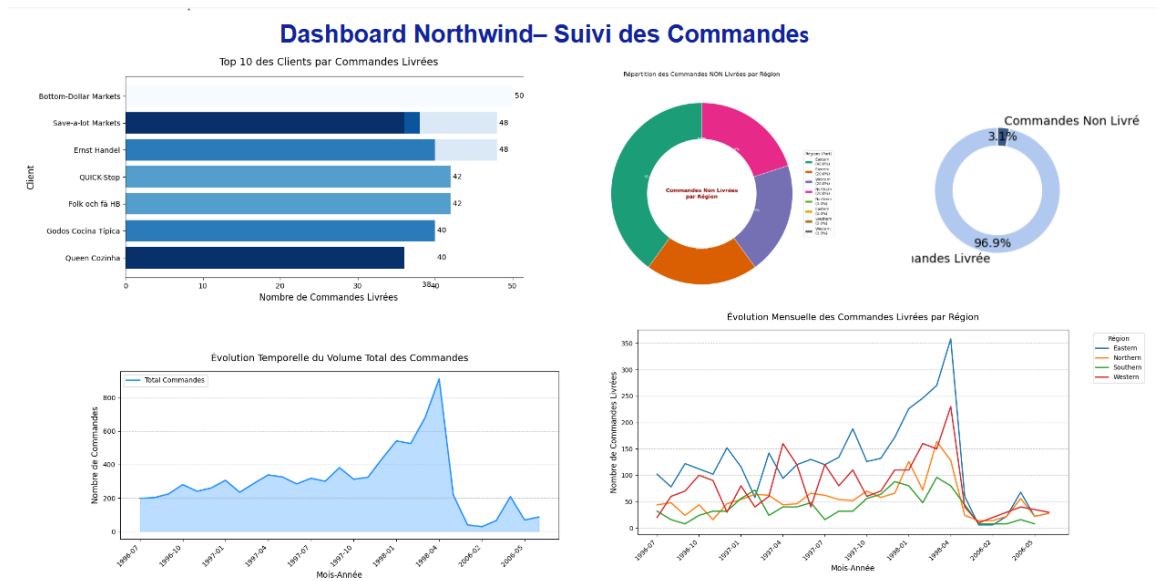
Cette visualisation est constituée de plusieurs courbes, chacune représentant une région. Elle montre l'évolution mensuelle du nombre de commandes livrées pour chaque région.

La comparaison simultanée des courbes permet d'analyser les différences de performance entre régions et d'identifier les tendances spécifiques à chacune d'elles.

**Objectif analytique :** comparer la performance régionale dans le temps et détecter les disparités géographiques.



## Le Dashboard finale



## Conclusion sur la visualisation

L'utilisation de scripts Python intégrés à Power BI a permis de concevoir des visualisations avancées et personnalisées, allant au-delà des graphiques standards proposés par l'outil.

Ces visualisations s'appuient sur un modèle dimensionnel robuste issu de la phase ETL, garantissant la cohérence des indicateurs et la pertinence des analyses. Elles offrent ainsi une lecture claire, synthétique et décisionnelle des données, contribuant efficacement à l'aide à la prise de décision.

## Conclusion

Ce TP a permis de mettre en œuvre un processus **ETL complet** dans Power BI, depuis l'extraction de sources multiples jusqu'à la création d'un modèle dimensionnel optimisé (schéma en étoile). La table de faits TF\_Commande, agrégée et liée aux dimensions, sert de fondement à un tableau de bord interactif et analytique.

Power BI a démontré sa capacité à intégrer en un seul outil les fonctions d'ETL, de modélisation et de visualisation, en offrant une approche plus accessible et visuelle que les outils ETL traditionnels comme Talend.