

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

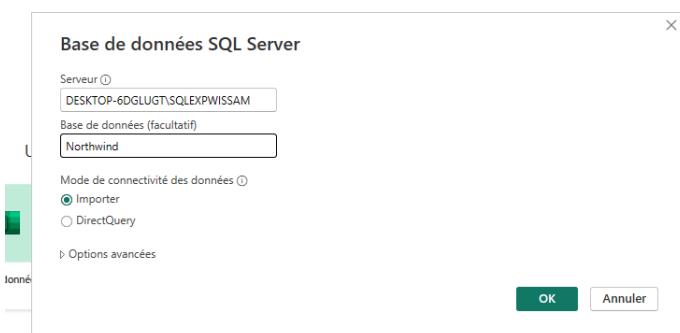
Dans le cadre de ce projet d'Intelligence d'Affaires, l'objectif principal est de mettre en œuvre un processus ETL complet permettant la conception d'un mini entrepôt de données basé sur un modèle en étoile. Ce modèle a pour but de faciliter l'analyse décisionnelle à travers l'intégration, la structuration 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.

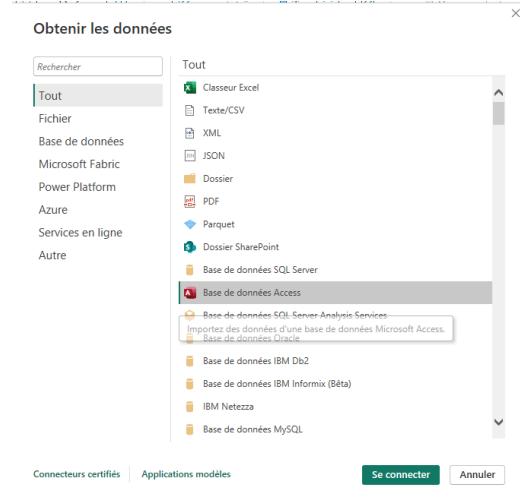
Avant de lancer le processus ETL, une étape préliminaire a été réalisée afin d'identifier et d'importer uniquement les tables nécessaires à la réalisation de notre projet. Cette approche permet de réduire le volume de données traitées, d'optimiser les performances des transformations et d'assurer la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





Navigateur

Options d'affichage *		Customers				
		ID	Company	Last Name	First Name	E-mail Address
<input type="checkbox"/>	Top Ten Orders by Sales Amount	1	Company A	Beddes	Anna	
<input type="checkbox"/>	Customers	2	Company B	Gretacous Solitona	Antonio	
<input type="checkbox"/>	Employee Privileges	3	Company C	Axen	Thomas	
<input checked="" type="checkbox"/>	Employees	4	Company D	Lee	Christine	
<input type="checkbox"/>	Inventory Transaction Types	5	Company E	O'Donnell	Maria	
<input type="checkbox"/>	Inventory Transactions	6	Company F	Pérez-Gatetta	Francisco	
<input type="checkbox"/>	Invoices	7	Company G	Xie	Ming-Yang	
<input type="checkbox"/>	Order Details	8	Company H	Andersen	Elizabeth	
<input type="checkbox"/>	Order Details Status	9	Company I	Mortensen	Sven	
<input checked="" type="checkbox"/>	Orders	10	Company J	Wacker	Roland	
<input type="checkbox"/>	Orders Status	11	Company K	Krischke	Peter	
<input type="checkbox"/>	Orders Tax Status	12	Company L	Edwards	John	
<input type="checkbox"/>	Privileges	13	Company M	Ludick	Andre	
<input type="checkbox"/>	Products	14	Company N	Gilio	Carlos	
<input type="checkbox"/>	Purchase Order Details	15	Company O	Kuhns	Helene	
<input type="checkbox"/>	Purchase Order Status	16	Company P	Coldenbomist	Doris	
<input type="checkbox"/>	Purchase Orders	17	Company Q	Bagel	Jean-Philippe	
<input type="checkbox"/>	Sales Reports	18	Company R	Autier Microni	Catherine	
<input type="checkbox"/>	Shippers	19	Company S	Eggerer	Alexander	
<input type="checkbox"/>	Strings	20	Company T	U	George	
<input type="checkbox"/>		21	Company U	Tham	Bernard	
<input type="checkbox"/>		22	Company V	Ramos	Luciana	

[Sélectionner les tables associées](#) [Charger](#) Transformer les données [Annuler](#)

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

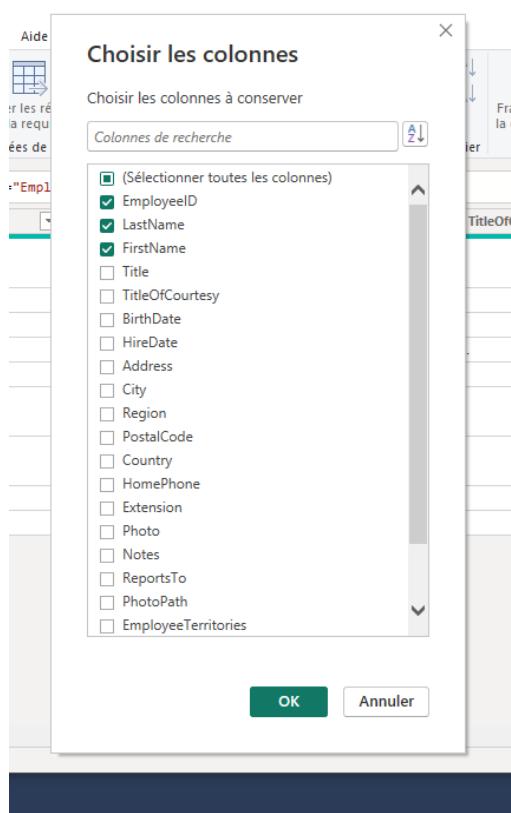
Procédure :

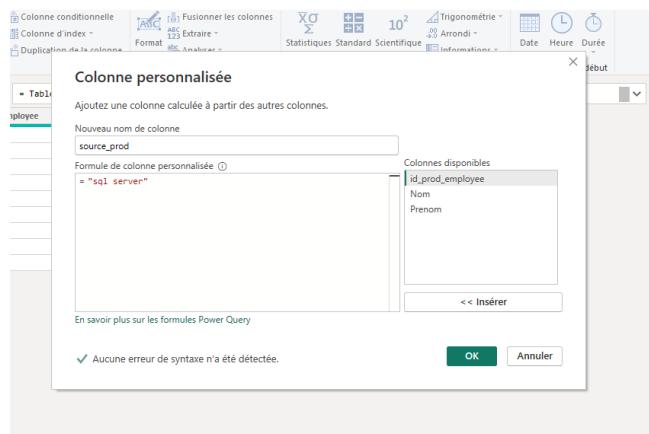
- Ajout d'une colonne personnalisée.
- Formule : "SQL_SERVER".

B. Source Access

Procédure :

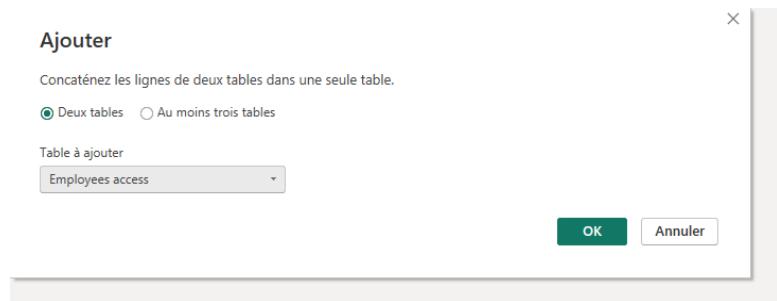
- Ajout d'une colonne personnalisée.
- Formule : "ACCESS".





Général				Analysier	Informations
<code>= Table.AddColumn(#"Colonnes renommées", "source_prod", each "access")</code>					
1	id_employee	Nom	Prenom	source_prod	
1	Freehafer	Nancy		access	
2	Cencini	Andrew		access	
3	Kotas	Jan		access	
4	Sergienko	Mariya		access	
5	Thorpe	Steven		access	
6	Neilper	Michael		access	
7	Zare	Robert		access	
8	Giusani	Laura		access	
9	Hellung-Larsen	Anne		access	

Ensuite on fait l'union des deux table

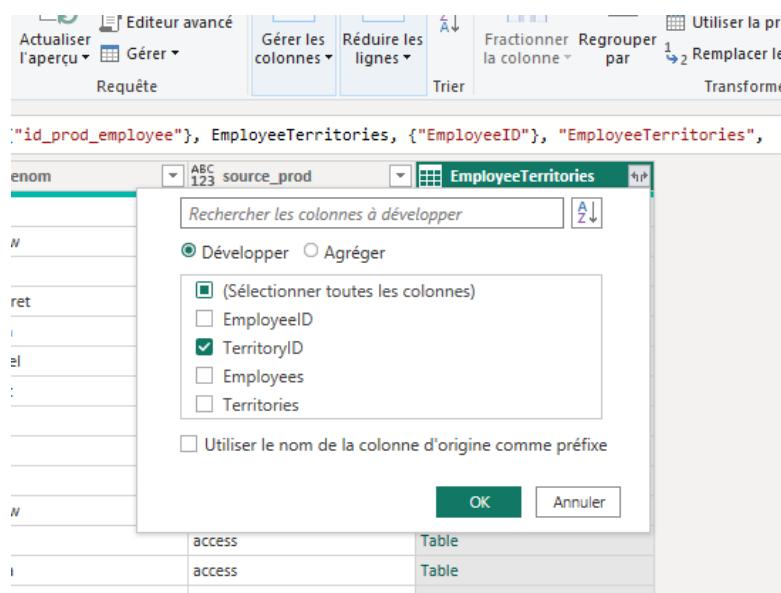
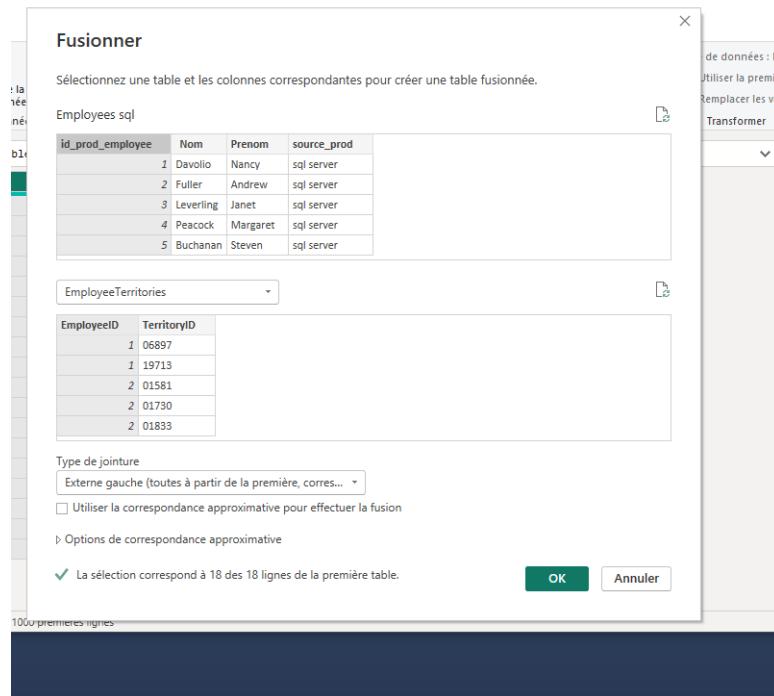


Étape 4 : Jointure avec la Table *EmployeeTerritories*

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*.

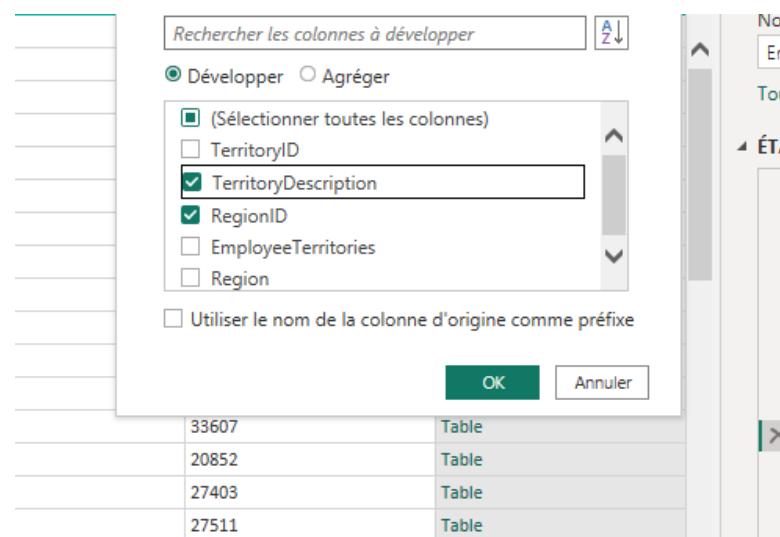
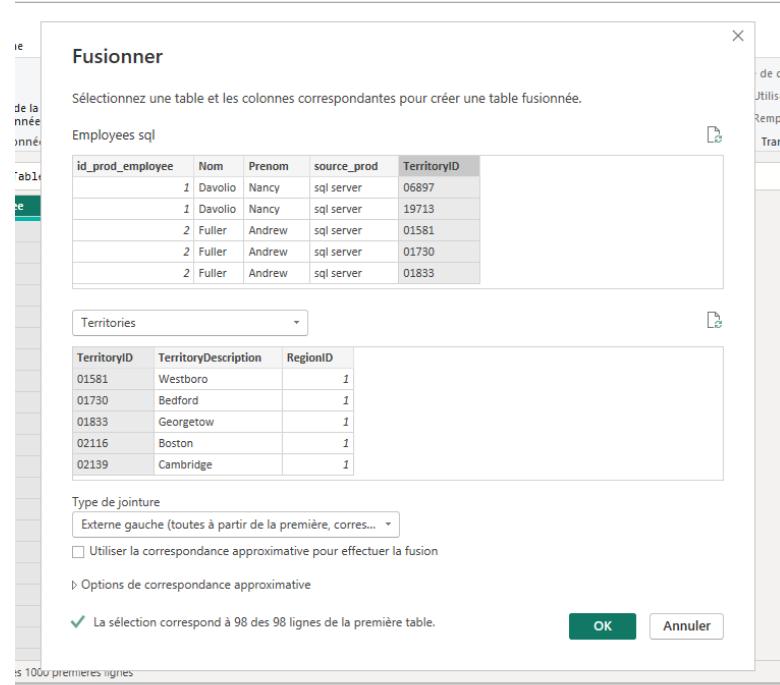


Étape 5 : Jointure avec la Table Territories

Objectif : Obtenir les informations géographiques des territoires.

Procédure :

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

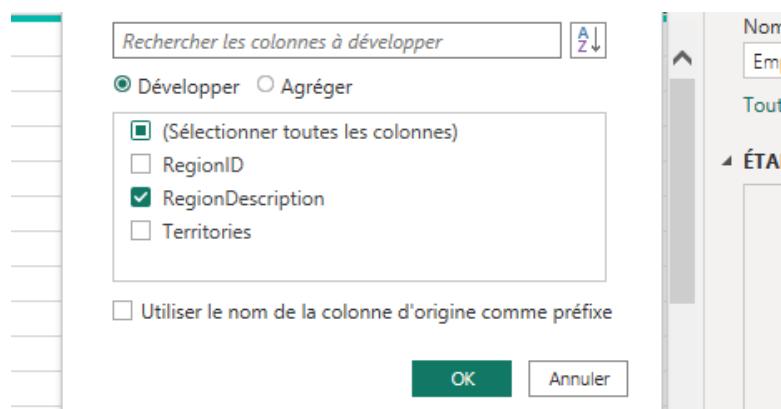
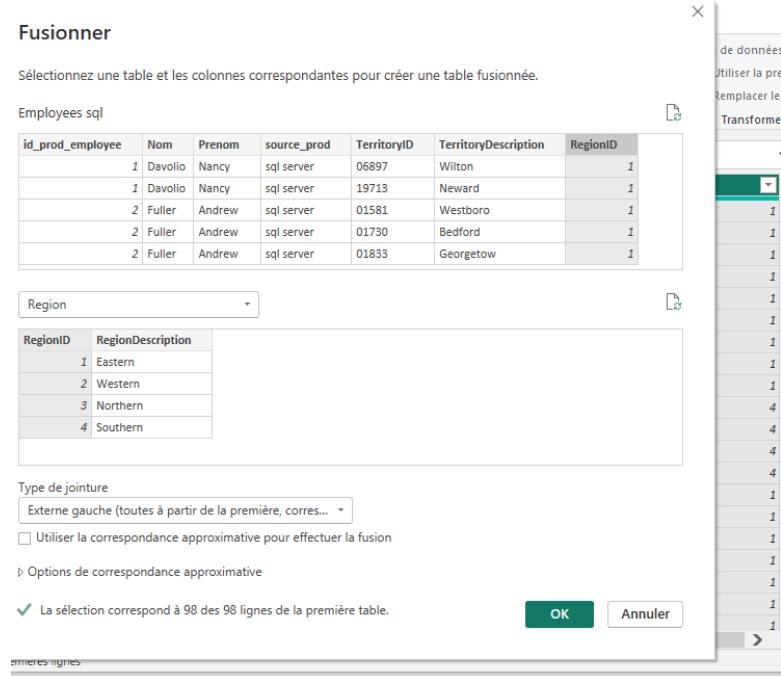


Étape 6 : Jointure avec la Table Region

Objectif : Ajouter l'information régionale.

Procédure :

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



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.

source_prod	TerritoryID	Territory	RegionID	Region	id_seq_employee
sql server	06897	Wilton	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	Brattleboro	1	Eastern	8
sql server	40223	Louisville	1	Eastern	9
sql server	30246	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	02803	Providence	1	Eastern	17
sql server	07960	Morrisons	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	11147	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	98604	Bellevue	2	Western	26
sql server	98652	Redmond	2	Western	27
sql server	98104	Seattle	2	Western	28
sql server	60179	Hoffman Estates	2	Western	29

Étape 9 : Finalisation et Chargement

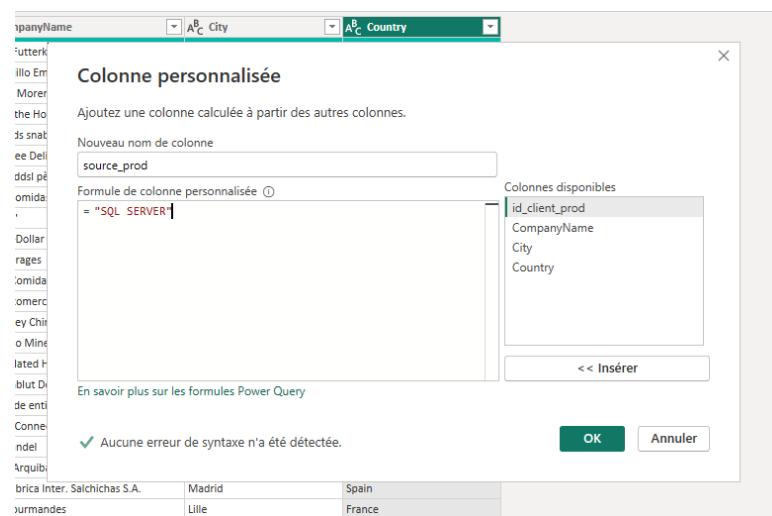
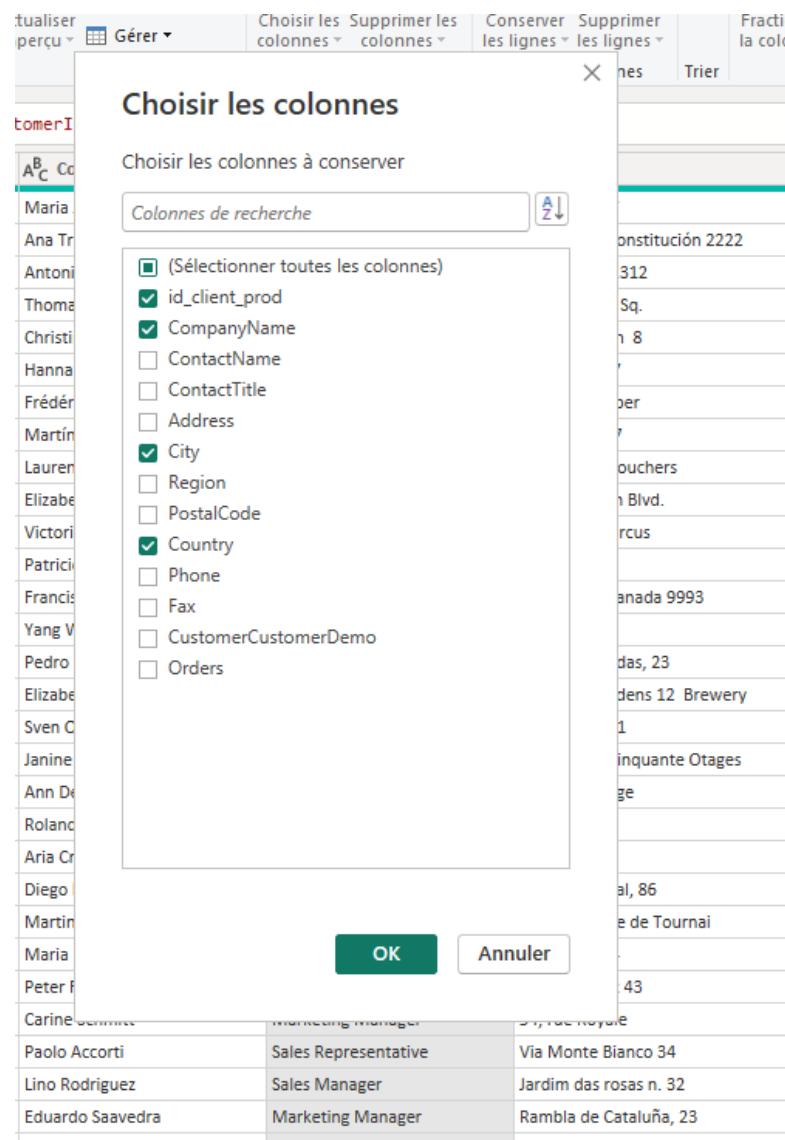
- Vérification de la cohérence des données.
- Fichier → Fermer et appliquer.
- Chargement de la table Dim_Employee dans le modèle Power BI.

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*. ont 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**.

Choisir les colonnes à conserver

Colonnes de recherche	A Z
<input type="checkbox"/> (Sélectionner toutes les colonnes)	
<input checked="" type="checkbox"/> ID	
<input checked="" type="checkbox"/> Company	
<input type="checkbox"/> Last Name	
<input type="checkbox"/> First Name	
<input type="checkbox"/> E-mail Address	
<input type="checkbox"/> Job Title	
<input type="checkbox"/> Business Phone	
<input type="checkbox"/> Home Phone	
<input type="checkbox"/> Mobile Phone	
<input type="checkbox"/> Fax Number	
<input type="checkbox"/> Address	
<input checked="" type="checkbox"/> City	
<input type="checkbox"/> State/Province	
<input type="checkbox"/> ZIP/Postal Code	
<input checked="" type="checkbox"/> Country/Region	
<input type="checkbox"/> Web Page	
<input type="checkbox"/> Notes	
<input type="checkbox"/> Orders	

Choisir les colonnes

Les colonnes à conserver sont : **City**, **Country/Region**

OK Annuler

Colonne personnalisée

Ajoutez une colonne calculée à partir des autres colonnes.

Nouveau nom de colonne : **source_prod**

Formule de colonne personnalisée : **= "ACCESS"**

Colonnes disponibles : **id_client_prod**, **CompanyName**, **City**, **Country**

OK Annuler

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.

	id_seq_client	id_client_prod	CompanyName	City	Country	source_prod
1	1	AUFI	Alfreds Futterkiste	Berlin	Germany	SQL SERVER
2	2	ANATR	Ana Trujillo Emparedados y helados	Mexico D.F.	Mexico	SQL SERVER
3	3	ANTON	Antonio Moreno Taqueria	Mexico D.F.	Mexico	SQL SERVER
4	4	AROUT	Around the Horn	London	UK	SQL SERVER
5	5	BERGS	Berglunds snabbköp	Luleå	Sweden	SQL SERVER
6	6	BLAUS	Blauer See Delikatessen	Mannheim	Germany	SQL SERVER
7	7	BLONP	Blondesdöd père et fils	Strasbourg	France	SQL SERVER
8	8	BOLID	Bólido Comidas preparadas	Madrid	Spain	SQL SERVER
9	9	BONAP	Bon app'	Marseille	France	SQL SERVER
10	10	BOTTM	Bottom-Dollar Markets	Tsawassen	Canada	SQL SERVER
11	11	BSBVE	B's Beverages	London	UK	SQL SERVER
12	12	CACTU	Cactus Comidas para llevar	Buenos Aires	Argentina	SQL SERVER
13	13	CENTC	Centro comercial Móctezuma	Mexico D.F.	Mexico	SQL SERVER
14	14	CHOPS	Chop-suey Chinese	Bern	Switzerland	SQL SERVER
15	15	COMMI	Comércio Mineiro	Sao Paulo	Brazil	SQL SERVER
16	16	CONSH	Consolidated Holdings	London	UK	SQL SERVER
17	17	DRACD	Drachenblut Delikatessen	Aachen	Germany	SQL SERVER
18	18	DUMON	Du monde entier	Nantes	France	SQL SERVER
19	19	EASTC	Eastern Connection	London	UK	SQL SERVER
20	20	ERNSH	Ernst Handel	Graz	Austria	SQL SERVER
21	21	FAMILIA	Familia Arquibaldo	Sao Paulo	Brazil	SQL SERVER
22	22	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain	SQL SERVER
23	23	FOULG	Folies gourmandes	Lille	France	SQL SERVER
24	24	FOKLO	Folk och fä HB	Bräcke	Sweden	SQL SERVER
25	25	FRANK	Frankensversand	München	Germany	SQL SERVER
26	26	FRANR	France restauration	Nantes	France	SQL SERVER
27	27	FRANS	Franchi S.p.A.	Torino	Italy	SQL SERVER
28	28	FURIB	Furia Bacalhau e Frutos do Mar	Lisboa	Portugal	SQL SERVER

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.

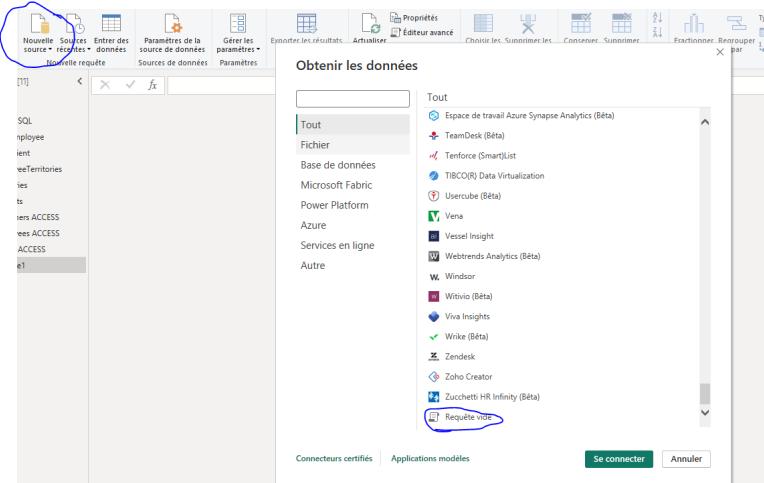
	id_dim_client	id_client_prod	CompanyName	City	Country	source_prod
1	1	AUFI	Alfreds Futterkiste	Berlin	Germany	SQL SERVER
2	2	ANATR	Ana Trujillo Emparedados y helados	Mexico D.F.	Mexico	SQL SERVER
3	3	ANTON	Antonio Moreno Taqueria	Mexico D.F.	Mexico	SQL SERVER
4	4	AROUT	Around the Horn	London	UK	SQL SERVER
5	5	BERGS	Berglunds snabbköp	Luleå	Sweden	SQL SERVER
6	6	BLAUS	Blauer See Delikatessen	Mannheim	Germany	SQL SERVER
7	7	BLONP	Blondesdöd père et fils	Strasbourg	France	SQL SERVER
8	8	BOLID	Bólido Comidas preparadas	Madrid	Spain	SQL SERVER
9	9	BONAP	Bon app'	Marseille	France	SQL SERVER
10	10	BOTTM	Bottom-Dollar Markets	Tsawassen	Canada	SQL SERVER
11	11	BSBVE	B's Beverages	London	UK	SQL SERVER
12	12	CACTU	Cactus Comidas para llevar	Buenos Aires	Argentina	SQL SERVER
13	13	CENTC	Centro comercial Móctezuma	Mexico D.F.	Mexico	SQL SERVER
14	14	CHOPS	Chop-suey Chinese	Bern	Switzerland	SQL SERVER
15	15	COMMI	Comércio Mineiro	Sao Paulo	Brazil	SQL SERVER
16	16	CONSH	Consolidated Holdings	London	UK	SQL SERVER
17	17	DRACD	Drachenblut Delikatessen	Aachen	Germany	SQL SERVER
18	18	DUMON	Du monde entier	Nantes	France	SQL SERVER
19	19	EASTC	Eastern Connection	London	UK	SQL SERVER
20	20	ERNSH	Ernst Handel	Graz	Austria	SQL SERVER
21	21	FAMILIA	Familia Arquibaldo	Sao Paulo	Brazil	SQL SERVER
22	22	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain	SQL SERVER
23	23	FOULG	Folies gourmandes	Lille	France	SQL SERVER
24	24	FOKLO	Folk och fä HB	Bräcke	Sweden	SQL SERVER
25	25	FRANK	Frankensversand	München	Germany	SQL SERVER
26	26	FRANR	France restauration	Nantes	France	SQL SERVER
27	27	FRANS	Franchi S.p.A.	Torino	Italy	SQL SERVER
28	28	FURIB	Furia Bacalhau 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 : Crédation 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 :

```

let
    StartDate = #date(1900,1,1),
    EndDate = #date(2024,12,31),
    NumberOfDay = Duration.Days((EndDate - StartDate) + 1),
    Dates = List.Dates(StartDate, NumberOfDay, #duration(1,0,0,0)),
    #"Converted to Table" = Table.FromList(Dates, Splitter.SplitByNothing(), {"Date"}, null, ExtraValues.Error),
    #"Changed Type" = Table.TransformColumnTypes(#"Converted to Table",{{"Date", type date}}),
    #"Added Year" = Table.AddColumn(#"Changed Type", "annee", each Date.Year([Date]), type text),
    #"Added MonthYear" = Table.AddColumn(#"Added Year", "mois_annee", each Date.ToText([Date], "yyyy-MM"), type text),
    #"Removed Duplicates" = Table.Distinct(#"Added MonthYear", {"annee", "mois_annee"}),
    #"Removed Date Column" = Table.RemoveColumns(#"Removed Duplicates",{"Date"})
in
    #"Removed Date Column"

```

= Table.RemoveColumns(#"Removed Duplicates", {"Date"})

	A ^B _C annee	A ^B _C 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

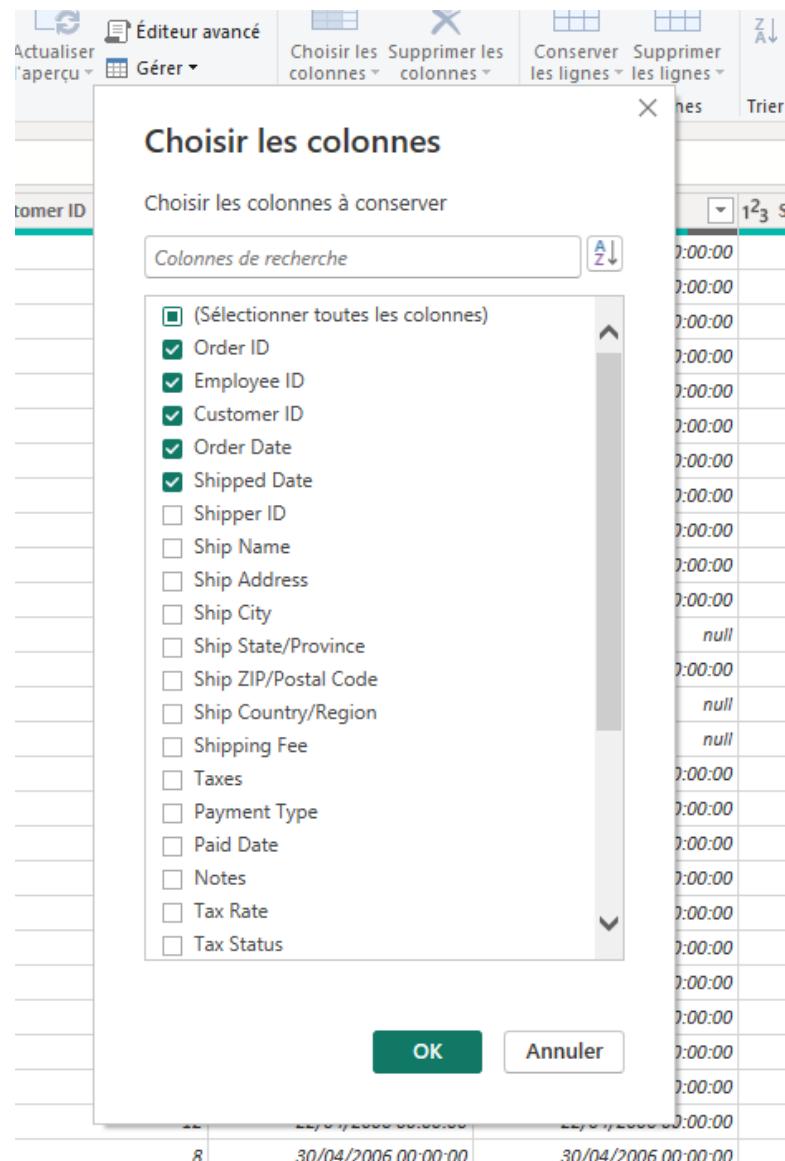
Connexion et Importation des Sources

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**.

The screenshot shows the Microsoft Power BI desktop application. At the top, there's a ribbon with tabs like 'Filtres', 'Actualiser l'aperçu', 'Éditeur avancé', 'Gérer', 'Choisir les colonnes', 'Supprimer les colonnes', 'Conserver les lignes', 'Supprimer les lignes', 'Fractionner la colonne', and 'Trier'. Below the ribbon, a large title 'Choisir les colonnes' is displayed, followed by a subtitle 'Choisir les colonnes à conserver'. A search bar labeled 'Colonnes de recherche' and a sorting icon ('AZ') are present. A list of columns is shown with checkboxes next to them. The checked columns are: OrderID, CustomerID, EmployeeID, OrderDate, ShippedDate, and ShipVia. The unchecked columns are: RequiredDate, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry, Customers, Employees, Order Details, and Shippers. At the bottom right of the dialog are 'OK' and 'Annuler' buttons. In the background, a preview pane shows a table with columns 'OrderID', 'CustomerID', 'EmployeeID', 'ShippedDate', and 'ShipVia' with some sample data rows.

Type de données : Date/Heure	Remplacer
Nombre décimal	Remplir
Nombre décimal fixe	Choisir la format
Nombre entier	Choisir le format
Pourcentage	Choisir le format
Date/Heure	Choisir le format
Date	Choisir le format
Heure	Choisir le format
Date/Heure/Fuseau horaire	Choisir le format
Durée	Choisir le format
Texte	Choisir le format
Vrai/Faux	Choisir le format
Binaire	Choisir le format



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.

The screenshot shows the 'Ajouter' (Add) dialog in Power Query. It displays two tables of data from the 'Orders Accesss' source. The dialog has a radio button for 'Deux tables' (Two tables) selected. A dropdown menu 'Table à ajouter' (Table to add) is set to 'Orders Accesss'. At the bottom right are 'OK' and 'Annuler' (Cancel) buttons.

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.ToString([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*.

The screenshot shows the 'Colonne personnalisée' (Custom Column) dialog in Power Query. It displays a formula `= Date.ToString([OrderDate], "yyyy-MM")` for a new column named 'mois_annee'. A list of available columns is shown on the right, including OrderID, CustomerID, EmployeeID, OrderDate, and ShippedDate. At the bottom right are 'OK' and 'Annuler' (Cancel) buttons.

les jointure

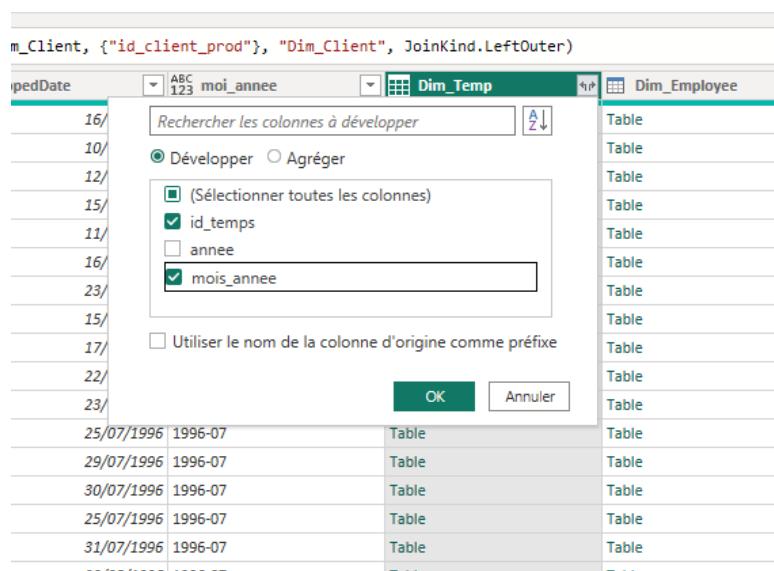
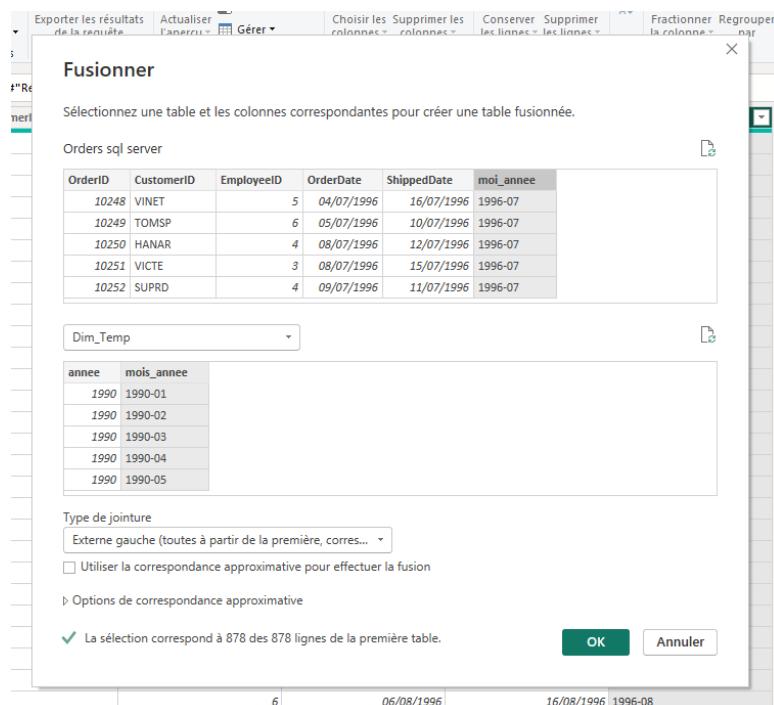
Étape 3 : Intégration des Dimensions

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**.

id	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

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

Paramètres d'application

PROPRIÉTÉS

Nom Orders sql server

Toutes les propriétés

ÉTAPES APPLIQUÉES

Source

Navigatio

Autres co

Type mod

Personnal

Colonnes

Requête a

Personnal

Requêtes

Requêtes

Requêtes

Dim_Tem

id_seq_employee")
nnée i23 id_seq_employee Dim_Client

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

70 Table
17 Table
18 Table

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 :

Nouveau nom de colonne
Commande_livree

Formule de colonne personnalisée ⓘ
= if [ShippedDate]<> null then 1 else 0

Colonnes disponibles
OrderID
CustomerID
EmployeeID
OrderDate
ShippedDate
id_temps
mois_annee

En savoir plus sur les formules Power Query

OK Annuler

04/09/1996	10/09/1996	81 1996-09	58
04/09/1996	10/09/1996	81 1996-09	62
04/09/1996	10/09/1996	81 1996-09	63

Colonne Commande_Non_livree :

Nouveau nom de colonne
Commande_non_livree

Formule de colonne personnalisée ⓘ
= if [ShippedDate]=null then 1 else 0

Colonnes disponibles
OrderID
CustomerID
EmployeeID
OrderDate
ShippedDate
id_temps
mois_annee

En savoir plus sur les formules Power Query

OK Annuler

10/09/1996	81 1996-09	58	7
10/09/1996	81 1996-09	62	7
10/09/1996	81 1996-09	63	7
10/09/1996	81 1996-09	65	7
10/09/1996	81 1996-09	67	7
10/09/1996	81 1996-09	68	7

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

The screenshot shows the 'Transformer' tab in Power BI. A blue circle highlights the 'Regrouper par' button in the top-left corner of the main area. Below it, there's a note: 'Utiliser la première ligne pour les en-têtes' (Use the first line for headers). To the right, there are several buttons: 'Transposer', 'Inverser les lignes', 'Compter les lignes', 'Type de données : N'importe quoi', 'Déterminer le type de données', and 'Renommer'. At the bottom, there's a section titled 'Tableau' with a preview of a table containing four rows of data.

This screenshot shows the 'Regrouper par' dialog box. It has two tabs: 'De base' (selected) and 'Avancé'. Under 'Avancé', four columns are selected for grouping: 'id_temps', 'mois_annee', 'id_seq_employee', and 'id_seq_client'. Below this, two aggregations are defined: 'Nmr_commande_livree' is aggregated by 'Somme' and assigned to the new column 'Commande_livree'; 'Nmr_commande_non_livree' is also aggregated by 'Somme' and assigned to the new column 'Commande_non_livree'. At the bottom, there are 'OK' and 'Annuler' buttons.

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

This screenshot shows the Power BI Query Editor with the 'H_Sec_Fait' table selected. The table contains data from four dimensions: 'T2_Commande' (id_seq_fait, id_temps, id_seq_employee, id_seq_client), 'Dim_Employee' (id_seq_employee), 'Dim_Client' (id_seq_client), and 'Dim_Temp' (mois_annee). There is also a 'Colonne renommée' column. The 'Paramètres d'une requête' pane on the right shows the applied steps: 'T2_Commande' and 'Colonne renommée'.

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

Modélisation et gestion des relations

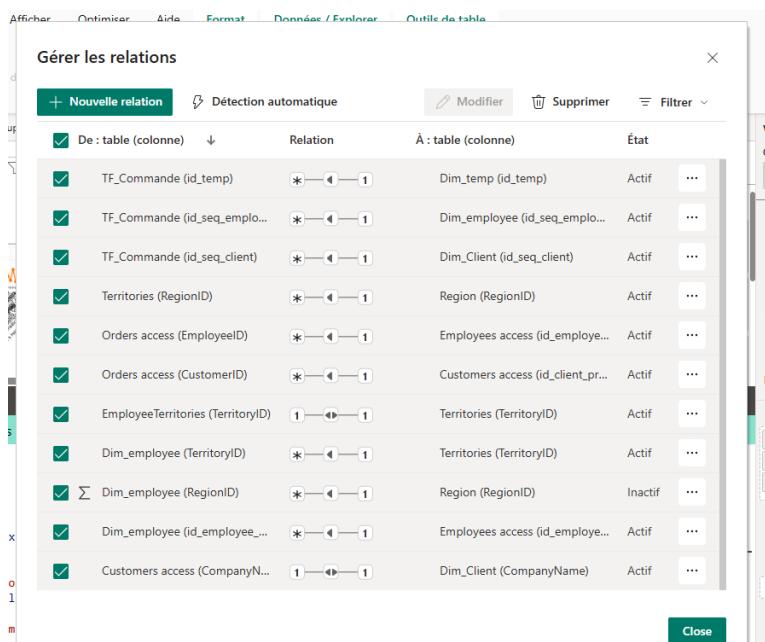
Avant la conception des visualisations, une étape fondamentale a consisté à définir et gérer les relations entre la table de faits et les tables de dimensions dans le modèle Power BI.

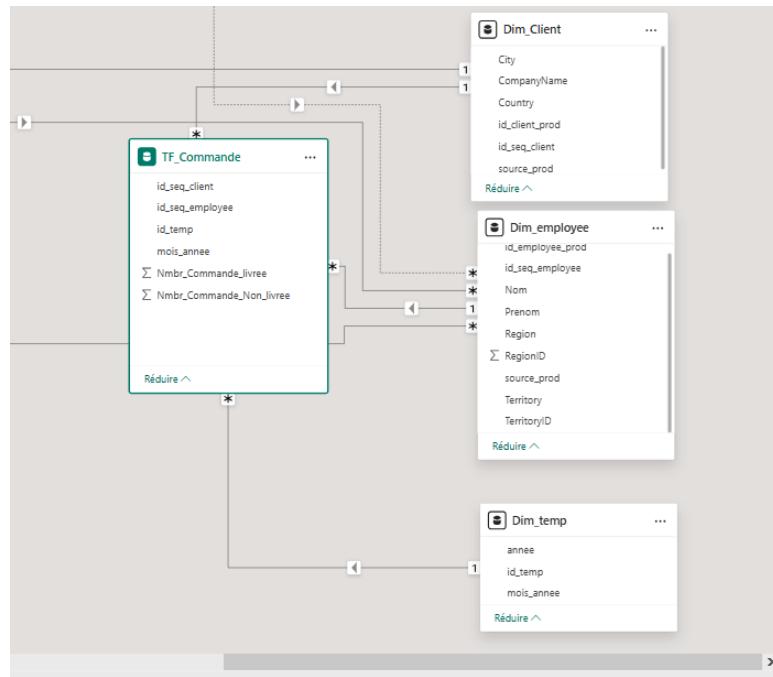
La table de faits **TF_Commande** a été reliée aux différentes tables de dimensions selon un schéma en étoile :

- **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.





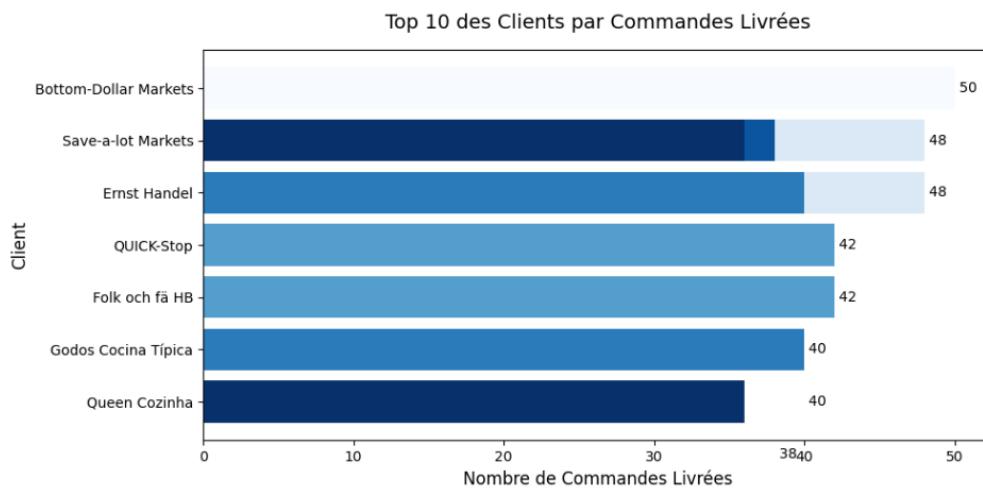
Visualisations réalisées par scripts Python

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.

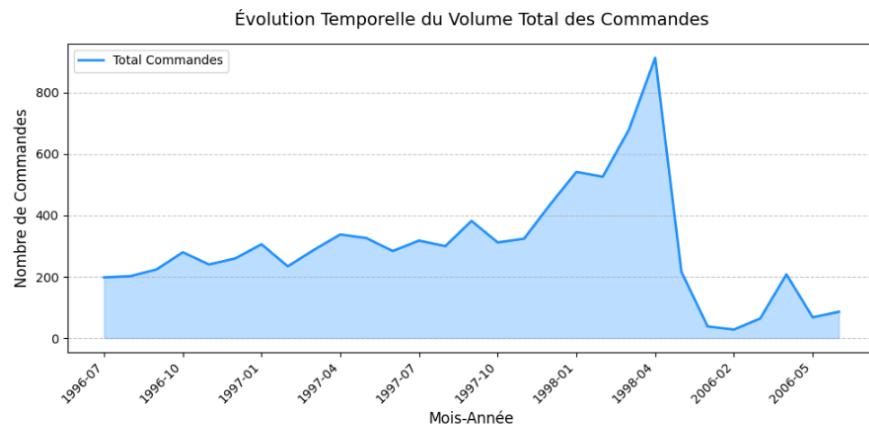


Évolution temporelle du volume total des commandes

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.

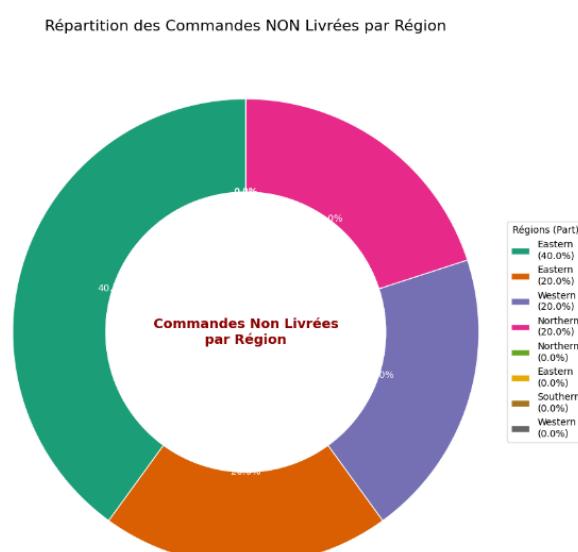


Répartition des commandes non livrées par région

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.

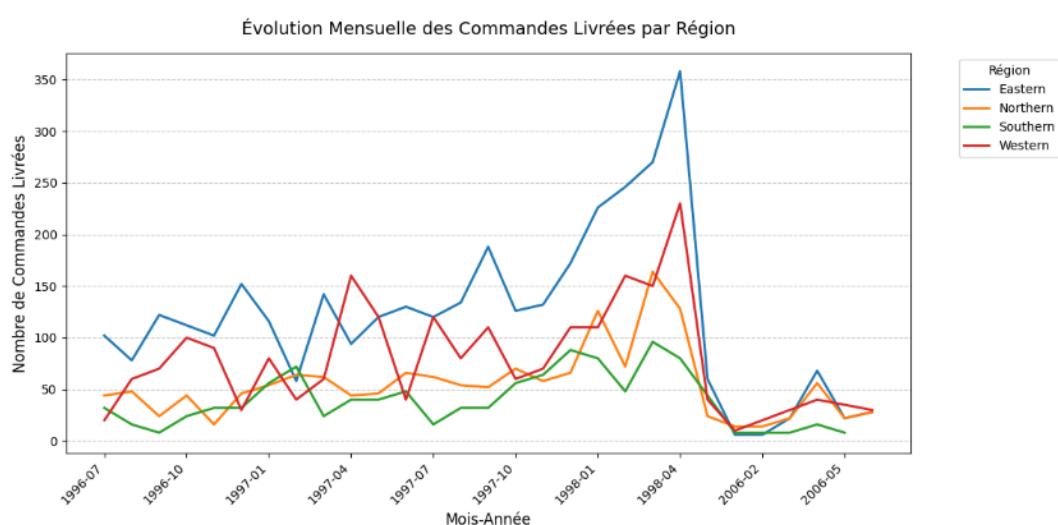


Évolution mensuelle des commandes livrées par région

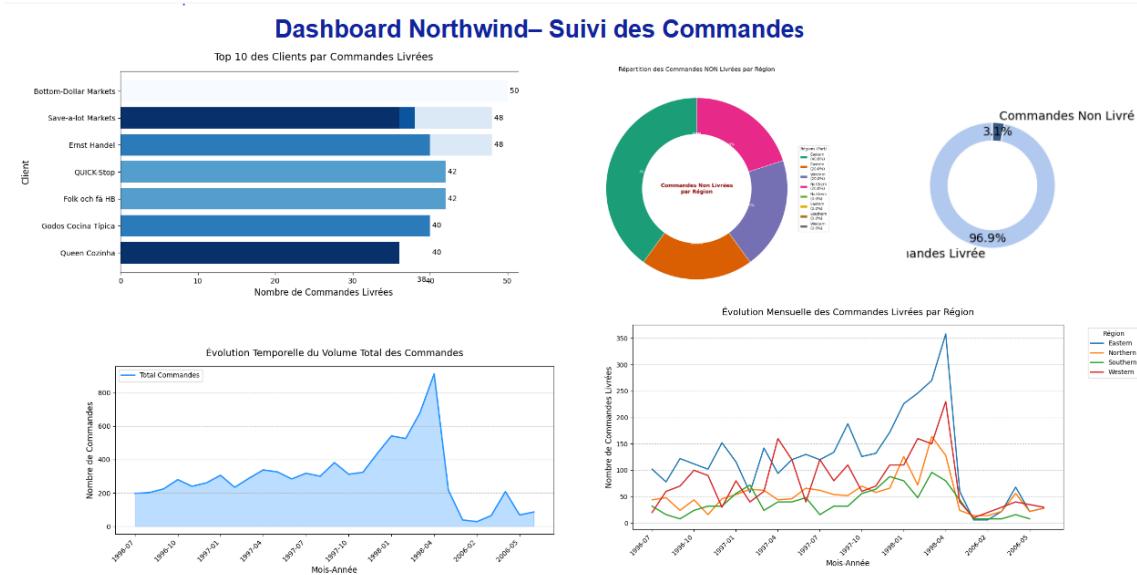
Cette visualisation est constituée de plusieurs courbes, chacune représentant une région. Elle montre l'évolution mensuelle du nombre de commandes livrées pour chaque région.

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.