



# Power BI

Formatrice : BERENGUER Laure



# Introduction

**Power BI**, la solution de Business Intelligence de Microsoft permet aux entreprises **d'agréger, d'analyser et de visualiser** les données en provenance de sources multiples.

À l'issue de cette formation, vous serez en mesure de **créer des tableaux de bord complexes** en maîtrisant les fonctionnalités **avancées** de ce logiciel.

## Public concerné et prérequis

### ► **Public concerné**

Toute personne souhaitant consolider et approfondir ses connaissances dans l'utilisation de Power BI afin de concevoir des tableaux de bord graphiques et interactifs de niveau avancé.

### ► **Prérequis**

Connaissances des bases de Power BI ou connaissances équivalentes à celles apportées par le stage Power BI, concevoir des tableaux de bord.

# La Formatrice

- **Laure BERENGUER**, consultante et formatrice indépendante
- **Site Internet** (cours et exercices) :  
<http://berenguer-formation-conseil.fr/power-bi-perfectionnement-orsys/>
- **Email** : [laure@berenguer.onmicrosoft.com](mailto:laure@berenguer.onmicrosoft.com)
- **LinkedIn** : [www.linkedin.com/in/laure-berenguer38/](http://www.linkedin.com/in/laure-berenguer38/)

## Objectifs de la formation

- Exploiter le langage M pour réaliser des modèles de données optimisés
- Concevoir des mesures sophistiquées avec le langage DAX
- Créer des rapports complexes avec Power BI Desktop
- Créer une présentation et un scénario adapté à votre message

# Plan de formation

- ▶ Présentation, rappels et bonnes pratiques Page 7
- ▶ Concevoir un modèle adéquat et performant Page 26
- ▶ Travailler avec la dimension Temps Page 71
- ▶ Exploiter les fonctions DAX avancées Page 132
- ▶ Affichage, navigation et visualisations Page 171
  
- ▶ Annexes Page 212

# Présentation, rappels et bonnes pratiques

7

# Présentation, rappels et bonnes pratiques

- Rappel du cycle de conception d'un rapport Power BI :
- ✓ Expression du besoin et identification des sources de données.
- ✓ Se connecter et préparer les données.
- ✓ Organiser les tables et les relations, définir des mesures, visualisation.
- Comprendre l'organisation d'un modèle en étoile, des tables de faits, des tables de dimensions.
- Comprendre l'intérêt et l'utilisation d'une table de Date / Calendrier



# La Business Intelligence

## Qu'est-ce que c'est ?

### ► L'informatique décisionnelle ou la BI

Ensemble des moyens, outils et méthodes qui permettent de collecter, de consolider, de modéliser et de restituer les données, matérielles ou immatérielles, d'une entreprise en vue d'offrir une aide à la décision.

# Rappel du cycle de conception d'un rapport Power BI

# Le cycle de conception :

## Expression besoins & sources

- **Expression du besoin / Travail d'enquêteur :**
  - ✓ Dans mon métier, **quelles sont les indicateurs dont j'ai besoin** pour faire mes reportings ?
  - ✓ **Lister tous les besoins** du process métier et/ou service : point avec toutes les personnes concernées (utilisateurs des rapports).
  - ✓ **Centraliser tous les besoins** afin de créer modèle de données Power BI réutilisable par le plus grand nombre : ne pas créer un modèle de données pour un rapport mais **un modèle pour tous les rapports** du service/process métier.
  
- **Identification des sources de données :**
  - ✓ **Quelles sont les sources** qui contiennent les données qui vont nous permettre de créer nos indicateurs ?
  - ✓ **Les accès** : demande des identifiants et mots de passe !

## Le cycle de conception : Connexion aux sources

- ✓ Les **connexions aux sources** sont possibles depuis les fenêtres Power Bi et Power Query,
- ✓ Il existe **différents types de connecteurs** en fonction **du type de sources** de données: Fichier (Excel, texte-csv, etc.), base de données (Oracle, Microsoft, etc.), Azure, Services en ligne (Dynamics 365, Google Analytics, etc.), etc.
- ✓ La connexion au jeu de données plus du tout indispensable car **accès au modèle de données** (*modèle sémantique –nouveau terme été 2023*) directement sur le site en ligne,

# Le cycle de conception :

## Préparation de données

- ✓ **Interface** : la majorité des transformations se font par le biais **d'assistants** ne nécessitant pas d'utiliser le langage M,
- ✓ **Étapes appliquées** : reprend l'ordre de modification des données. On peut modifier et/ou supprimer chacune en revenant dessus ou en modifiant le code correspondant.
- ✓ **Editeur avancé** : permet de visualiser et récupérer le code complet de chaque requête.
- ✓ **Requête vide** : permet **de commencer à partir d'une requête vierge** et de coller facilement le code d'une autre requête (partage d'une table d'un service à un autre).
- ✓ **Actualisation des données** : pour récupérer les nouvelles données des différentes sources.

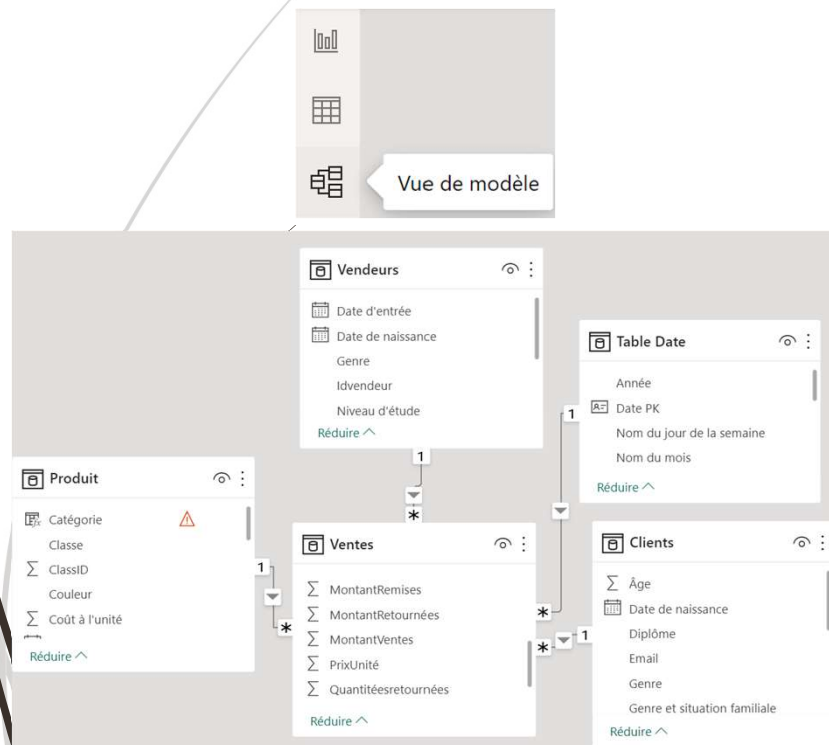
## Le cycle de conception : Organisation des tables

- ✓ **Réduction** du nombre de tables,
- ✓ **Eviter** la table unique
- ✓ Positionner la table contenant les **indicateurs** / la plus importante, **au centre** des autres.
- ✓ Organisation en **schéma en étoile** (Cf. partie suivante)

# Le cycle de conception : Organisation des relations

- ✓ Conditions des **relations automatiques** : Si vos colonnes de correspondances entre les tables ont le même nom et le même type de données.
- ✓ **Créer & modifier et/ou contrôler** : l'outil permet de modifier et contrôler aisément.
- ✓ **Relations actives et inactives** : Une seule relation active est possible. Si d'autres relations doivent être utilisées, elles seront créées en inactive puis on les sollicitera à l'aide d'une fonction DAX.
- ✓ **Cardinalités** : les cardinalités souhaitées sont 1-N dans schéma BI. *Pour un client, il peut y avoir plusieurs commandes (N). Pour une commande, il ne peut y avoir qu'un seul client associé (1).*
- ✓ **Direction du filtrage croisé** : si 1-N, toujours « à sens unique » d'un élément en 1 vers N.
- ✓ Vérifier avec **TCD** : Afin de **contrôler** le bon fonctionnement des relations.

# Le cycle de conception : Modèle de données



- Icône de gauche « Modèle ».
- Il est nécessaire de « **relier vos sources** » et donc vos tables importées afin de pouvoir utiliser des colonnes qui se trouvent dans des tables différentes dans un même visuel.
- Le but n'est pas de tout rassembler avec des RecherchesV mais juste de donner les moyens à l'outil **de naviguer correctement entre les tables** comme si une énorme table de correspondance existait en arrière-fond.
- **Il faut donc lui donner une colonne de correspondance qui fasse le lien entre deux tables comme pour la fonction RechercheV dans Excel.**



# Le cycle de conception : Gestion des relations

- Toujours au niveau de « **la vue de modèle** », cliquer sur « Gérer les relations » dans le ruban « Accueil ».
- La fenêtre **de gestion des relations** apparaît.
- On peut cliquer sur une des **relations pour la modifier, la supprimer et/ou en rajouter** des nouvelles.
- On peut aussi **cliquer directement sur un champ** du modèle (Schéma des tables), rester appuyé et **venir glisser** le premier champ sur le champ de correspondance d'une autre table pour créer la relation.

## Gérer les relations

Active	De : Table (Colonne)	À : Table (Colonne)
<input checked="" type="checkbox"/>	Clients (idgeo)	Géographie (idgeo)
<input checked="" type="checkbox"/>	Souscategorie (Idcatégorie)	Categorie (Idcategorie)
<input checked="" type="checkbox"/>	Ventes (Date de vente)	Table Date (Date PK)
<input checked="" type="checkbox"/>	Ventes (idclient)	Clients (Idclient)
<input checked="" type="checkbox"/>	Ventes (idproduit)	Produit (Idproduit)
<input checked="" type="checkbox"/>	Ventes (idvendeur)	Vendeurs (idvendeur)

### Explication de la première ligne :

Sur la table client, la colonne IDGEO est égale à la colonne IDGEO dans la table Géographie. Cette relation est active donc Power BI peut l'utiliser pour naviguer dans le modèle de données.

# Le cycle de conception : Définition mesures et autres

- ✓ **Intérêt des mesures** : filtrer et bloquer un calcul, créer un titre dynamique, gérer les relations inactives, créer une table temporaire/en mémoire, etc.
- ✓ **Identification** des mesures à créer : est-ce juste une colonne à créer dans Power Query ou cela nécessite-t-il la création d'une mesure dans Power BI? En termes de volumétrie : faut-il mieux créer une mesure qui agrège une colonne en mémoire (SUMX) ou créer la colonne dans Power Query ?
- ✓ **Organisation** et rangement des mesures : possibilité de créer des tables de mesures puis de créer des dossiers dans les tables pour une meilleure prise en main des concepteurs de rapports.
- ✓ **Gestion de la table Date**
- ✓ **Gestion avec les outils de table et de colonnes** (format, résumé, trier par, création de hiérarchies, etc.)

# Le cycle de conception : Visualisation

- ✓ **Sélection du visuel** le plus pertinent/ adapté pour mettre en valeur vos données / indicateurs
- ✓ **Mise en forme** globale et par visuel. Gestion de thème personnalisé et/ou modèle de rapports.
- ✓ **Réflexion et gestion sur les filtres et segments**
- ✓ **Éléments de datavisualisation** : aérer les visuels, homogénéiser les couleurs et police, mise en avant d'indicateurs clés, aide à la navigation avec les signets, etc.

# Comprendre l'organisation d'un modèle en étoile

# Comprendre le modèle en étoile

## Tables de faits et de dimensions

### ➤ Intérêts et explication :

- ✓ Un schéma en étoile est un modèle de données multidimensionnel qui permet d'organiser une base de données afin de faciliter **sa compréhension et son analyse**.
- ✓ Il est orienté **processus métier**.
- ✓ Jointures/rerelations restantes sur le modèle sont **simples et performantes** car le nombre de jointures pour accéder aux données est **limitée**.

### ► Le schéma en étoile est constitué :

- ✓ Table(s) de faits : D'une(De) **table(s) centrale(s) contenant les données quantitatives et mesurables** à positionner en « valeurs » dans les visuels.
- ✓ Table(s) de dimension : De tables qui gravitent autour **contenant les données qualitatives** à positionner en « lignes, colonnes, axes et légendes » dans les tableaux et graphiques croisés dynamiques ou autres visuels.

# Comprendre le modèle en étoile

## Vue globale et questionnements ?

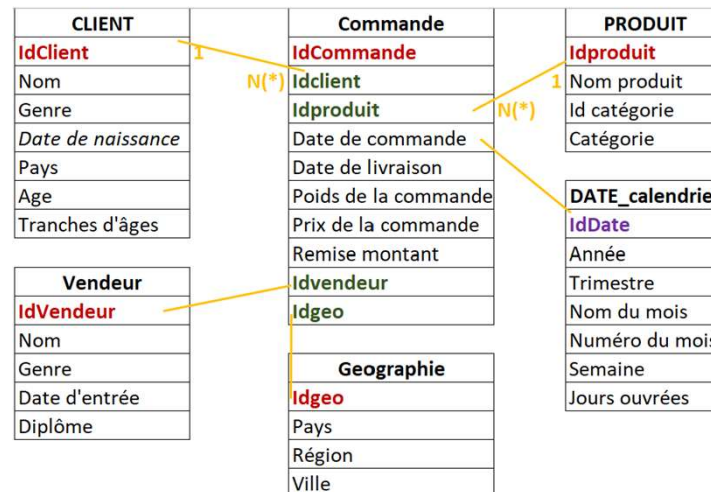
### ➤ Quelles questions pour le construire ? Quel contenu ? :

- ✓ **Table de faits/mesures – table centrale** : Qu'est-ce que je souhaite analyser ?

Regroupe les données quantitatives à mettre en valeurs dans les visuels.

- ✓ **Table de dimensions – tables qui gravitent autour** : à travers quoi ?

Concerne les données qualitatives positionnées en lignes/colonnes TCD et en axe/légende graphique.



# Comprendre le modèle en étoile

## Construction : matrice de bus

### ➤ Intérêts et explication :

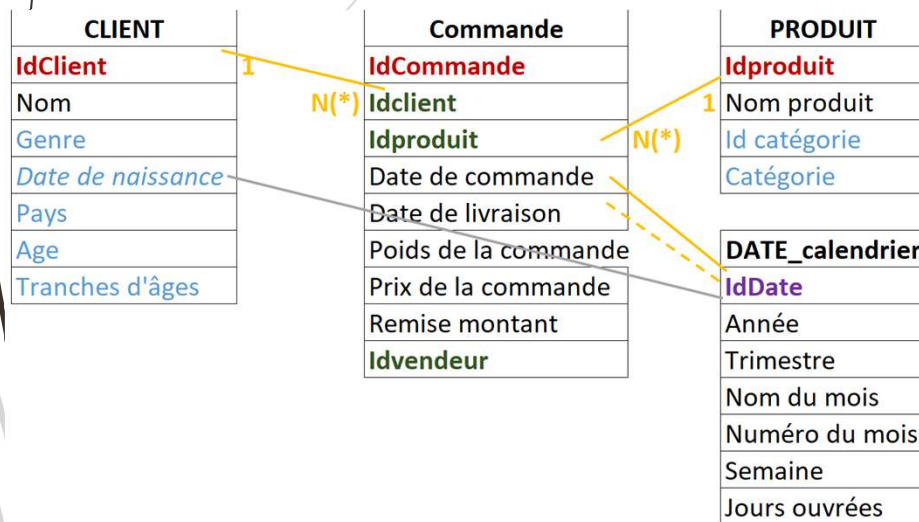
- ✓ Permet de représenter de manière globale chaque **processus métier** pour construire correctement son(s) schéma(s) en étoile.
- ✓ Pour chaque *processus métier* se dégage **plusieurs faits** (indicateurs) qui sont organisés dans des **tables de faits** en fonction de leurs **liens** et **granularité** (mode de calcul – niveau de détail le plus bas)

Processus métier	Table de Fait	Granularité	Fait	Dimensions			
				Client	Produit	Promotion	Date
Gestion des commandes	FactCommande	Une ligne par commande	Montant des commandes	✘			✘
			Quantité des commandes				
			Coût totaux				
Gestion des commandes	FactDetailsCommande	Une ligne par commande à un client par produit	Prix unitaire	✘	✘	✘	✘
			Coût unitaire				
			Nombre produits commandés				
Satisfaction globale	SatisfactionGlobale	Une ligne par client	Note satisfaction globale	✘			✘
			Note temps de livraison				

# Comprendre l'intérêt et l'utilisation d'une table de Date / Calendrier



# Comprendre l'intérêt d'une table de date / calendrier



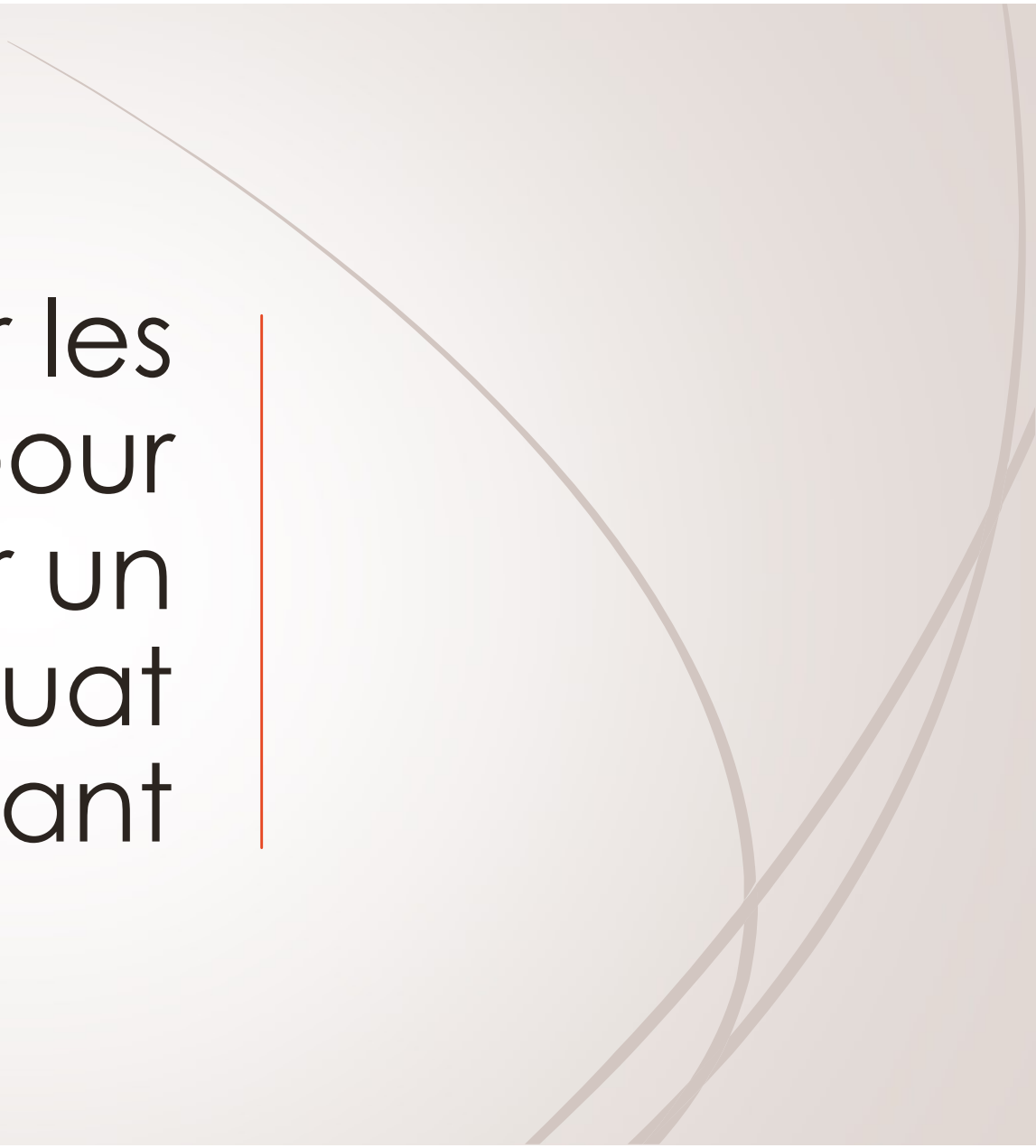
- Un trait plein signifie une relation active (toujours utilisée pour les visuels)
- Un trait avec des tirets signifie une relation inactive (à activer via une fonction DAX).

## ➤ Ajout de la table & intérêt :

- ✓ Toutes les temporalités regroupées au même endroit,
  - ✓ Faire des relations entre chaque date et la clé primaire/date de la table Date-Calendarier.
  - ✓ Spécifier la table Date-Calendarier à Power BI.
  - ✓ Utilisation des fonctions Dax Time Intelligence possible seulement avec cette table.
- ## ✓ Pour la créer correctement dans Power BI:
- ✓ Chaque date est unique,
  - ✓ Chaque ligne de date est renseignée,
  - ✓ Pas de dates manquantes entre une date de début et une date de fin.

26

Utiliser les  
requêtes pour  
concevoir un  
modèle adéquat  
& performant



## Présentation, rappels et bonnes pratiques

- Rappels :
  - ✓ Les différents types de transformation
  - ✓ Activer le chargement et inclure dans l'actualisation du rapport
  - ✓ Dupliquer & Référence
- Fusionner les requêtes pour réduire les flocons
- Agréger et empiler les sources
- Importer des données à partir d'une liste de sources
- Personnaliser des fonctions et passage de paramètres en langage M
- Gérer les erreurs

# Rappels

# Rappels : Power Query

## Les différents types de transformations

### ➤ Transformations de base :

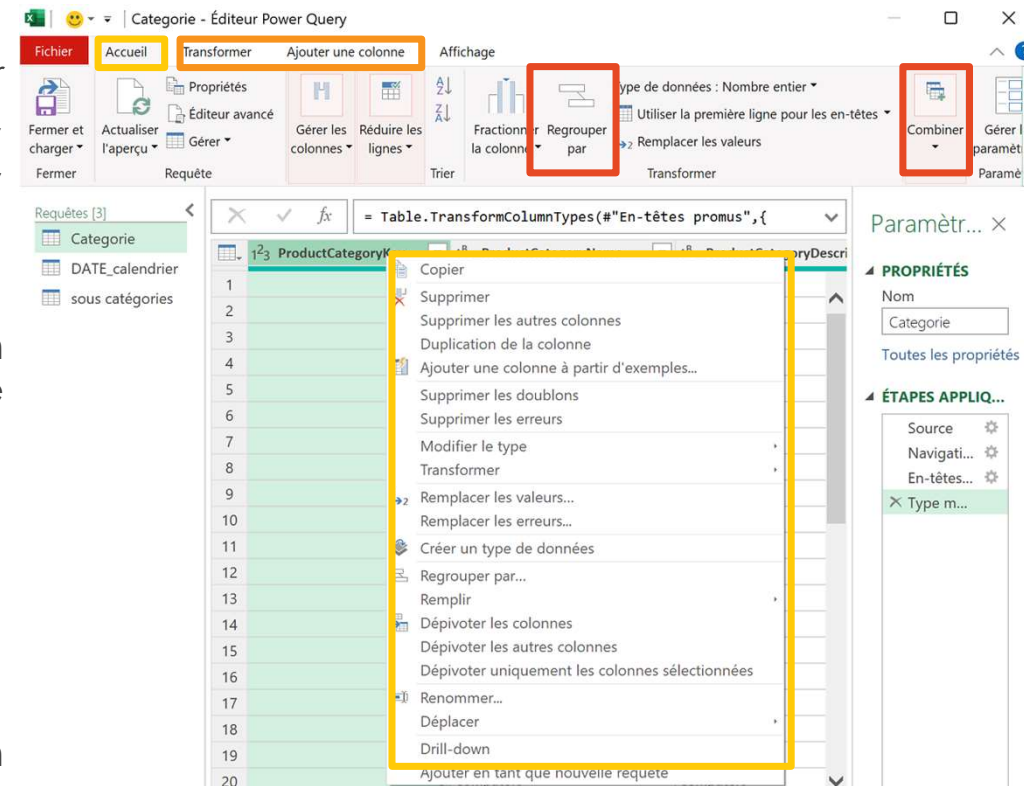
- ✓ Renommer des colonnes, supprimer des colonnes/doublons, remplacer les valeurs, dépivoter, transformer le type de données, fractionner une colonne.

### ➤ Transformations intermédiaires :

- ✓ Création de colonnes calculées simples, création du mois/année/jour et de l'âge en fonction d'une colonne de date existante, etc.

### ➤ Transformations avancées :

- ✓ Regrouper par : calculer (somme, moyenne, etc.) par un ou plusieurs autres champs,
- ✓ Fusion de lignes entre deux tables
- ✓ Fusion de requêtes : jointures entre deux tables afin de récupérer des colonnes et filtrer les lignes en fonction.





# Rappels : Power Query

Quels sont tous les outils possibles ?

- ✓ **Garder en mémoire les connexions aux sources de données** : *chemin d'accès vers vos fichiers Excel, CSV, etc.*
- ✓ **Préparer les données** : *renommer les colonnes, créer des colonnes, filtrer, etc.*
- ✓ **Automatiser la préparation** : tout ce qu'on fait une fois, il le reproduit de manière autonome à chaque actualisation.
- **C'est l'avantage principal de l'outil : il ne faut pas tout refaire pour chaque rapport** hebdomadaire/mensuel ou annuel. De plus, les performances de l'outil sont très élevées et permettent de gérer des fichiers contenant plus d'un million de lignes.
- ✓ **Charger les données finalisées dans Power BI Desktop.**

# Rappels : Power Query

## La préparation : les modifications de base

- Vérifier que pour chaque colonne, l'outil a bien **identifier le bon type de données** :
- **Organiser l'affichage** de la table :
  - **Transposer** (lignes / colonnes), **dépivoter les colonnes**, **trier** afin de réorganiser la table.
- Les différents « **renommage** » :
  - Je souhaite renommer **la table, les colonnes ou bien des valeurs**, c'est ici que je vais le faire en cliquant droit sur la colonne.
- **Nettoyer les données** :
  - **Supprimer les colonnes, les doublons et supprimer les lignes et ou les filtrer** afin de ne garder que ce que l'on souhaite dans la table.
  - **Fractionner les colonnes** afin de séparer des éléments dans des colonnes différentes.



# Rappels : Power Query

## Les transformations/création de colonnes

- Les **transformations/création de colonnes** sur des colonnes de type **texte** : Tout en majuscule/minuscule, supprimer les espaces, etc.
- Les **transformations/création de colonnes** sur des colonnes de type **numérique** : soustraction, addition, multiplication et division simple mais également les arrondis.
- Les **transformations/création de colonnes** sur des colonnes de type **dates**:
  - ✓ À partir d'une date, je **peux récupérer l'année, le mois, le jour**, etc. très facilement sans taper du code comme sur Excel (fonction ANNEE par exemple).
  - ✓ À partir d'une date, je peux également **calculer des âges et/ ou délai** : âge du client jusqu'à la date du jour, délai de livraison, etc.





# Rappels : Power Query

## Les fusions de tables / sources de données

➤ Les **fusions de lignes** :

➤ Si j'ai deux tables contenant les mêmes colonnes (ayant le même nom) et que je souhaite **compiler/concaténer/ajouter les lignes** d'une table en dessous des lignes de l'autre table alors je peux le faire très facilement sur l'outil, sans **formule/sans copier/coller** comme sur Excel.

➤ Exemple :

- J'ai deux tables de ventes, l'une pour la partie Nord de la France et une autre pour la partie Sud.
- Je vais regrouper toutes les lignes des deux fichiers/tables afin d'avoir tout au même endroit.

# Rappels : Power Query

## Les fusions de tables / sources de données

### ➤ Les **fusions de colonnes** :

- Si j'ai deux fichiers différents mais dont une colonne est identique dans les deux et permet de faire **une correspondance**, alors je peux fusionner deux tables pour **recupérer des colonnes** grâce à la colonne de correspondance. Plus simple et plus facile qu'un **RechercheV** sur Excel.

### ➤ Exemple :

- J'ai une table de commande avec toutes les informations en lien avec la commande mais je souhaite récupérer le nom du client qui est une colonne qui se trouve dans la table client.
- Je vais utiliser le 'Numéro du client' présent dans les deux tables pour ramener dans la table commande, le nom du client 😊

# Rappels : Power Query

## Vigilance : Colonnes conditionnelles

### 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 ①		Sortie ①	
Si	Âge	égal à	ABC 123 null	Alors	ABC 123 Non renseigné	...
Autre...	Âge	est inférieur à	ABC 123 20	Alors	ABC 123 Moins de 20 ans	
Autre...	Âge	est inférieur à	ABC 123 40	Alors	ABC 123 20 - 39 ans	
Autre...	Âge	est inférieur à	ABC 123 60	Alors	ABC 123 40 - 59 ans	

Autre ①




S'il doit gérer des valeurs numériques qui contiennent des Null (pas de valeur)

**Il est important de gérer les NULL en premier**

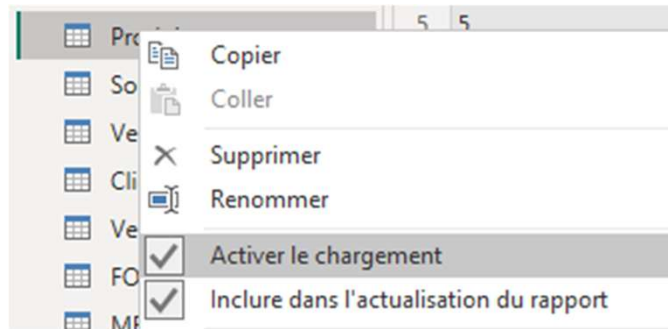
Sinon, cela créera des **ERREURS.**

# Rappels : Power Query

## Désactiver/activer le chargement

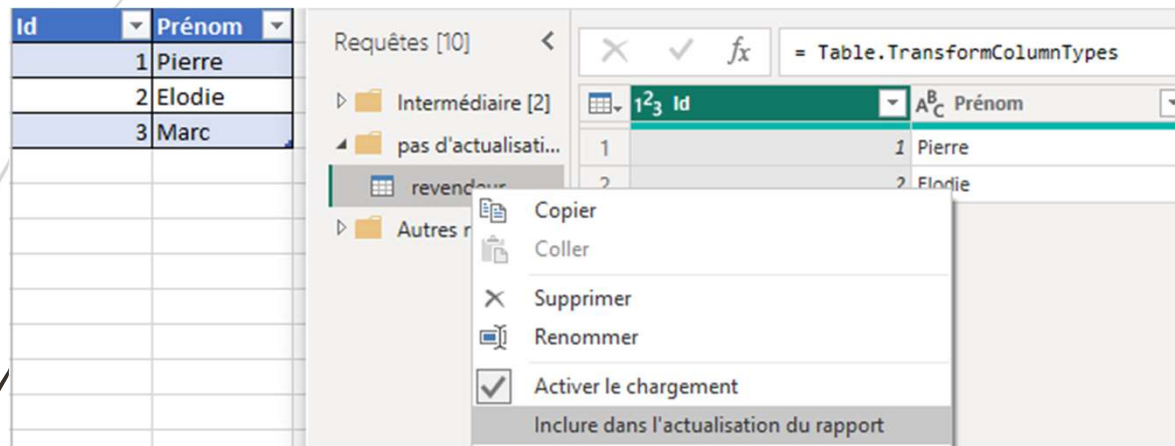
- Par défaut, chaque requête/table à laquelle on s'est connecté via Power Query, **est chargée automatiquement dans Power BI.**

Afin de **désactiver le chargement**, il suffit de cliquer sur la case cochée par défaut.



# Rappels : Power Query

## Inclure dans l'actualisation du rapport



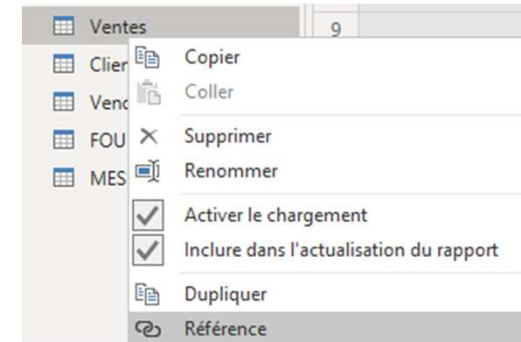
*Si plus tard, une nouvelle ligne est créée (comme dans l'exemple ci-dessus) dans la source et que j'actualise Power BI, il **n'apparaîtra pas dans la fenêtre Power BI et les rapports** utilisant les données correspondantes.*

- On peut aussi **exclure l'actualisation des données** d'une requête.
- Je me connecte une première fois à une source de données, sauf que je souhaite que **les données ne bougent plus même si la source est modifiée**.
- Cela permet d'avoir un **instantanée des données**.

# Rappels : Power Query

## Dupliquer et Référence de requête

- ❖ Il est possible de dupliquer une requête, mais deux possibilités existent dans ce sens.
- **Dupliquer** : permet de **copier/coller l'éditeur avancé** de la requête. Il n'y a donc aucune dépendance entre la requête copiée et celle collée. Chacune va se connecter à la source de données en externe.
- **Référence** : permet de **copier/coller le résultat en lien** avec la dernière étape appliquée de la requête copiée. Il y a donc une dépendance. Si on rajoute une étape dans la requête copiée, cela affectera la requête collée.



Idvendeur	Nom du vendeur	Prénom	Date d'entrée
1	Young	Chloe	02/01/2017 00
2	Gill	Larry	02/01/2017 00
3	Gonzalez	Geoffrey	02/01/2017 00
4	Watson	Alexa	02/01/2017 00

PROPRIÉTÉS  
Nom  
Vendeurs (2)  
Toutes les propriétés

ÉTAPES APPLIQUÉES  
Source

- Il faut **cliquer droit sur la requête** à gauche afin d'y accéder !

Fusionner les requêtes  
pour réduire les flocons

# Concevoir un modèle adéquat

## Fusionner des requêtes pour réduire flocons

- Afin de **créer une table de dimension**, il est souvent nécessaire de récupérer **dans une seule et même table** des informations provenant possiblement **de différentes sources et/ou tables** de données différentes.
- Par exemple, on se connectera à **3 sources** correspondants aux tables **Produit, catégorie et sous-catégorie** mais il n'y a aucun intérêt à garder les 3 tables séparées.
- Il faut donc **récupérer les colonnes liées aux catégories et sous-catégories** directement dans la **table Produit**.



# Concevoir un modèle adéquat

## Fusionner des requêtes pour réduire flocons

- L'outil à utiliser dans Power Query est « **FUSIONNER DES REQUÊTES** »
- Permet de **recupérer des colonnes** grâce à une ou plusieurs colonnes de correspondance (la même dans les deux tables/sources). Proche du **RechercheV** sur Excel.
- Je clique sur la **table/requête** dans laquelle je souhaite rajouter une(des) colonne(s) provenant d'une autre table avant de cliquer sur Fusionner,
- Je spécifie ensuite **l'autre table** en dessous et **la colonne de correspondances** de chaque côté,
- Je choisis **le type de jointure**.

### Fusionner

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

Souscatégorie

Idsouscatégorie	Sous catégories	Idcatégorie
1	MP4 et MP3	1
2	Stylo d'enregistrement	1
3	Ecouteurs Bluetooth	1
4	Televisions	2
5	VCD et DVD	2

Categorie

Idcatégorie	Nom	desc
1	Audio	Audio
2	TV et Vidéo	TV et Vidéo
3	Ordinateurs	Ordinateurs
4	Caméras et caméscopes	Caméras et caméscopes
5	Téléphones portables	Téléphones portables

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 23 des 23 lignes de la première table.

OK

# Concevoir un modèle adéquat

## Fusionner des requêtes pour réduire flocons

- Je sélectionne **le ou les colonnes que je souhaite récupérer** en cliquant sur le bouton à droite de l'intitulé de colonne (qui porte le nom de l'autre table) :

The screenshot shows a table with the following columns: Idsouscatégorie, Sous catégories, Idcatégorie, and Catégorie. The 'Catégorie' column is selected. A context menu is open over the 'Catégorie' column, showing options to 'Développer' (Develop) or 'Agréger' (Aggregate). The 'Développer' option is selected. Below the menu, there are checkboxes for '(Sélectionner toutes les colonnes)', 'Idcatégorie', 'Nom', and 'desc'. The 'Nom' and 'desc' options are checked. There is also a checkbox for 'Utiliser le nom de la colonne d'origine comme préfixe' which is unchecked. At the bottom right, there are 'OK' and 'Annuler' buttons.

- Je clique sur **Développer** si je souhaite uniquement rajouter les colonnes et leurs valeurs correspondantes.

1 <sup>2</sup> 3 Idsouscatégorie	A <sup>B</sup> C Sous catégories	1 <sup>2</sup> 3 Idcatégorie	A <sup>B</sup> C Nom
1	MP4 et MP3		1 Audio
2	Stylo d'enregistrement		1 Audio
3	Ecouteurs Bluetooth		1 Audio
4	Televisions		2 TV et Video

# Concevoir un modèle adéquat

## Fusionner des requêtes pour réduire flocons

- Je sélectionne sur **Agréger** si je souhaite faire des calculs (agrégations : somme, moyenne, etc) sur des colonnes de l'autre table.
- Ici, je suis dans la table catégorie, donc il y a plusieurs sous-catégories pour une même catégorie, donc je souhaite compter combien il y en a pour chaque catégorie.

1 <sup>2</sup> Idcategorie	A <sup>B</sup> <sub>C</sub> Nom	A <sup>B</sup> <sub>C</sub> desc	Souscategorie
1	Audio		
2	TV et Vidéo		
3	Ordinateurs		
4	Caméras et caméscopes		
5	Téléphones portables		
6	Musique, films et livres audios		
7	Jeux et jouets		
8	Electroménager		

1 <sup>2</sup> Idcategorie	A <sup>B</sup> <sub>C</sub> Nom	1.2 Nombre de Sous catégories
1	Audio	3
2	TV et Vidéo	3
3	Ordinateurs	3
4	Caméras et caméscopes	3
5	Téléphones portables	4
6	Musique, films et livres audios	1
7	Jeux et jouets	2
8	Electroménager	4

# Concevoir un modèle adéquat

## Exercices (1/4)

- 1) Dans le **fichier de Formation « Dossier exos »**, ouvrir le dossier « Données BIG » puis importer les fichiers suivants :
  - **« Tous documents »** → Document Excel contenant **4 feuilles à importer** : « Géographie », « VentesAVANT2023 », « VentesDEPUIS2023 », « Livraisons et retours » et **4 tableaux à importer** « Vendeurs », « Catégorie », « Sous-catégorie » et « Produit ».
  - **« Clients »** → fichier texte à importer,
- 2) **Récupérer les lignes** de la table *VentesDEPUIS2023* dans la table *VentesAVANT2023*.  
*Il faut se positionner sur la table qui récupère toutes les lignes donc VentesAVANT2023 puis cliquer sur « Ajouter des requêtes » puis renommer la table « Ventes ».*
- 3) **Désactiver le chargement** dans Power BI de la table *VentesDEPUIS2023*.

# Concevoir un modèle adéquat

## Exercices (2/4)

- 4) **Récupération des colonnes** « Date de livraison » et « Date de retour » de la table *Livraisons et retours* dans la table *Ventes* en utilisant la colonne en commun « IdVente ». Se positionner sur *Ventes* et cliquer sur « Fusionner des requêtes ».
- 5) **Récupération de la colonne** « Catégorie » de la table *Categorie* dans la table *SousCategorie* en utilisant la colonne en commun « IdCategorie ». Se positionner sur *SousCategorie* et cliquer sur « Fusionner des requêtes ».
- 6) **Récupération des colonnes** « Catégorie », « Idcategorie » et « Sous catégories » de la table *SousCategorie* dans la table *Produit* en utilisant la colonne en commun « Idsouscategorie ». Se positionner sur *Produit* et cliquer sur « Fusionner des requêtes ».
- 7) **Désactiver le chargement** dans Power BI des tables *Livraisons et retours*, *Categorie* et *SousCategorie*.

# Concevoir un modèle adéquat

## Exercices (3/4)

**8) Créer une référence de la table produit** en cliquant droit sur la table puis « Référence ». Puis, désactiver le chargement pour cette nouvelle table.

### 9) Regrouper par

- ✓ Regrouper par classe et par couleur,
- ✓ Calculer le nombre de produits, le coût max à l'unité et le prix moyen à l'unité.

#### Regrouper par

Spécifiez les colonnes de regroupement et une ou plusieurs sorties.

De base  Avancé

Classe

Couleur

Ajouter un regroupement

Nouveau nom de colonne

Nombre de produits

Coût max à l'unité

Prix moyen à l'unité

Ajouter une agrégation

Opération

Compter les lignes

Max

Moyenne

Colonne

Coût à l'unité

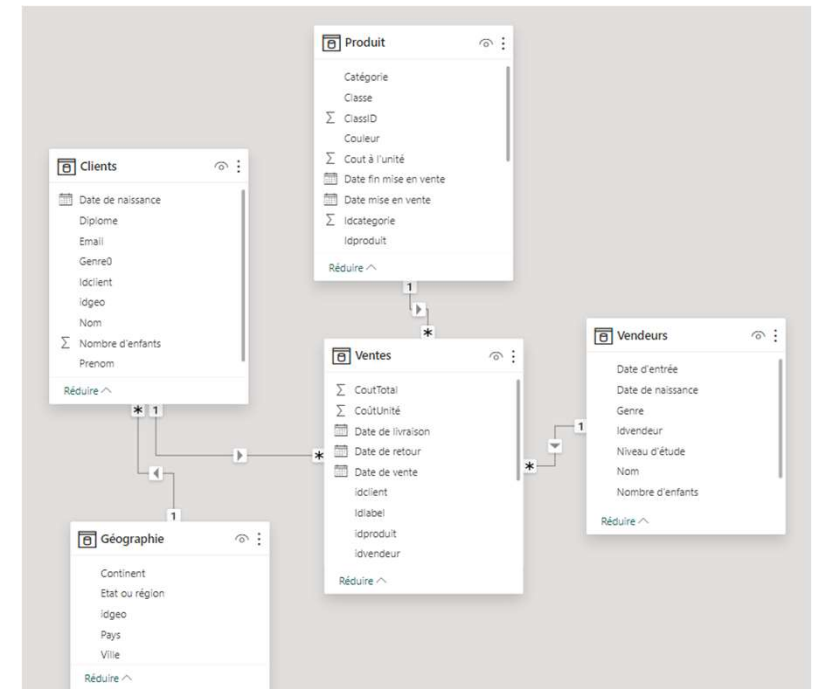
Prix à l'unité

A <sup>B</sup> <sub>C</sub> Classe	A <sup>B</sup> <sub>C</sub> Couleur	1 <sup>2</sup> <sub>3</sub> Nombre de produits	1.2 Coût max à l'unité	1.2 Prix moyen à l'unité
Economique	Argent	7	429,58	247,8342857
Economique	Noir	10	137,63	114,459
Economique	Vert	2	30,58	37,795
Economique	Blanc	6	509,32	261,2683333
Moyen	Rouge	5	187,62	138,342
Moyen	Noir	7	397,25	370,84
De luxe	Bleu	4	220,64	460,99

# Concevoir un modèle adéquat

## Exercices (4/4)

- 4) **Fermer et appliquer Power BI** (message d'erreur – cliquer dessus, laisser de côté puis continuer sur la fenêtre Power BI)
- 5) Dans l'onglet « **Vue de modèle** », vu que les **clés de relations ont le même nom et le même type de données, le modèle se fait tout seul.**
- 4) Il ne restera qu'à organiser tout ça, comme il faut 😊 avec la table ventes au centre.



# Concevoir un modèle adéquat

## Démonstration – Fusion et création de liste

- Dans le **fichier de Formation « Dossier exos »**, ouvrir le Dossier « Partie 2 Listes fonct param BIG » puis le document « Création listes de dates et fusion complexes » à faire ensemble pas à pas.
- Les **données** sont dans le dossier « Dossier Principal » puis fichier Excel « Matrice de bus ».
- Les tableaux à importer s'intitulent sont « Prix » et « Quantité ».

### **Problématique :**

- On a un tableau « Quantité » contenant les dates d'achats, le produit concerné et la quantité vendue mais le prix est inexistant ici. Il faut aller le chercher dans le tableau « Prix ».
- Dans la table « Prix », on a les produits qui se répètent autant de fois qu'il y a eu un prix différent. Il y a une date de début et une date de fin concernant la période pour laquelle chaque prix était en vigueur.
- ✓ Afin de récupérer le bon prix sur la table « quantité », il faut donc créer une liste qui va créer autant de lignes qu'il y a de dates concernées pour chaque prix et ensuite faire une fusion **pour récupérer le bon prix en fonction de la date et du produit**.



# Agréger et empiler les sources

# Concevoir un modèle adéquat

## Agréger et empiler les sources

- Afin de **créer une table de faits**, il est souvent nécessaire de récupérer et **regrouper dans une seule et même table** des informations provenant possiblement **de différentes sources et/ou tables** de données différentes.
- La **table finale contiendra** alors :
  - ✓ Les **identifiants** permettant de faire des relations entre la table de faits et les autres tables environnantes,
  - ✓ Les **dates** (de commande, de livraison, etc.) afin de faire le lien avec la table de Date / Calendrier dont une sert d'IDDate,
  - ✓ Les **données quantitatives et mesurables** reprenant toutes les colonnes contenant les indicateurs dont j'ai besoin : le montant des ventes, les quantités, les coûts totaux, etc.

FactDetailsCommande  
(Matrice bus p21)

IdClient
IdProduit
IdPromotion
Date de commande
Prix unitaire
Coût unitaire
Nombre de produits commandés

# Concevoir un modèle adéquat

## Regrouper pour agréger le fichier final

**Regrouper par** ×

Spécifiez les colonnes de regroupement et une ou plusieurs sorties.

De base  Avancé

Id vente ▾

IdClient ▾

IdProduit ▾

IdPromotion ▾

Date de commande ▾

Ajouter un regroupement

Nouveau nom de colonne	Opération	Colonne
Nombre	Nombre de lignes distinctes ▾	▾
Prix unitaire	Somme ▾	Prix unitaire ▾
Coût unitaire	Somme ▾	Coût unitaire ▾
Nb prod commandés	Somme ▾	Nombre de produits comm... ▾
Ajouter une agrégation		

OK Annuler

1. Cliquer sur « **Avancé** » afin de pouvoir créer autant de regroupements et agrégations que souhaité,
2. **Regroupement** : mettre les **identifiants et dates**,
3. **Agrégation** : utiliser les **données quantitatives et mesurables** pour créer les agrégations souhaitées (le plus souvent des sommes).

# Concevoir un modèle adéquat

## Regrouper pour agréger le fichier final

<b>Id vente</b>	<b>IdClient</b>	<b>IdProduit</b>	<b>IdPromotion</b>	<b>Date de commande</b>	<b>Prix unitaire</b>	<b>Coût unitaire</b>	<b>Nombre de produits commandés</b>
1	2	3	1	23/01/2024	15	3	1
1	2	2	1	23/01/2024	20	5	1
1	2	3	1	23/01/2024	15	3	3
2	1	1	1	24/01/2024	10	3	1
2	1	3	2	24/01/2024	15	3	2

<b>Id vente</b>	<b>IdClient</b>	<b>IdProduit</b>	<b>IdPromotion</b>	<b>Date de commande</b>	<b>Nombre de lignes</b>	<b>Prix unitaire</b>	<b>Coût unitaire</b>	<b>Nb prod commandés</b>
1	2	3	1	23/01/2024	2	30	6	4
1	2	2	1	23/01/2024	1	20	5	1
2	1	1	1	24/01/2024	1	10	3	1
2	1	3	2	24/01/2024	1	15	3	2

# Importer des données à partir d'une liste de sources

# Concevoir un modèle adéquat

## Importer liste de sources : Connexion Dossier

- Afin de se **connecter à plusieurs fichiers**, il est possible de se connecter au **Dossier** dans lequel ils sont enregistrés.
- Que ce soit dans **un dossier de votre PC/dossier partagé** mais également dossier **SharePoint**.

### Obtenir les données

Tout

Fichier

Base de données

Microsoft Fabric

Power Platform

Azure

Services en ligne

Autre

Tout

Classeur Excel

Texte/CSV

XML

JSON

Dossier

PDF

Parquet

Dossier SharePoint

# Concevoir un modèle adéquat

## Importer liste de sources : Connexion Dossier

- Je sélectionne l'emplacement du fichier :  
→ *Partie 2\Dossier principal\Combiner n fichiers*
- Puis cliquer sur « **Combiner et charger** »  
-> s'il n'y a pas de transformation identique à faire sur chaque fichier avant de les combiner.
- A partir d'un **fichier exemple** : le premier qu'il trouve ou celui que l'on détermine pour être le fichier d'exemple,
- Lui expliquer s'il faut se connecter au tableau1 (tableau excel) ou Feuil1 (la feuille qui contient le tableau).
- Il **reproduira la procédure** pour chaque fichier qu'il trouve dans le dossier.

### Dossier

Chemin du dossier

C:\Users\Utilisateur\Desktop\Combiner n fichiers

Parcourir...

OK

### Combiner les fichiers

Sélectionnez l'objet à extraire de chaque fichier. [En savoir plus](#)

Exemple de fichier :

Premier fichier

Premier fichier Avril.xlsx

Avril.xlsx C:\Users\Utilisateur\Desktop\Combiner n fichiers\

Février.xlsx C:\Users\Utilisateur\Desktop\Combiner n fichiers\

Mars.xlsx C:\Users\Utilisateur\Desktop\Combiner n fichiers\

Options d'affichage

Paramètre1 [2]

Tableau1

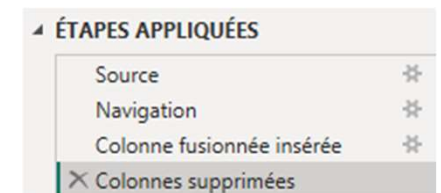
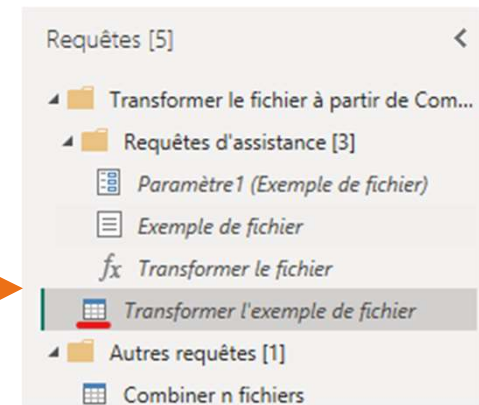
Feuil1

Paul	Barre de cér
Emmanuelle	Tomates séc
Julie	Poisson

# Concevoir un modèle adéquat

## Importer liste de sources : Connexion Dossier

- Je sélectionne l'emplacement du fichier :
- Puis cliquer sur « **Combiner et transformer les données** »  
-> Pour rajouter transformations avant de combiner/concaténer les lignes de tous les fichiers du dossier.
- La fenêtre Power Query s'ouvre et il y a **une requête en plus** dans les requêtes d'assistance intitulée «**transformer l'exemple de fichier** ».
- *Par défaut, il n'existe que Source et Navigation dans les étapes appliquées de base de cette requête.*
- On va ensuite sur la requête finale qui affiche **les données du fichier exemple** et on rajoute les transformations que l'on souhaite qu'il applique sur chaque requête avant de combiner.





# Concevoir un modèle adéquat

## Importer liste de sources : Connexion Dossier

- On clique ensuite **sur la requête finale** (dans « autres requêtes ») : Combiner n fichiers. Les transformations sont prises en compte.

Requêtes [5]

- Transformer le fichier à partir de Com...
- Requêtes d'assistance [3]
  - Paramètre1 (Exemple de fichier)
  - Exemple de fichier
  - fx Transformer le fichier
  - Transformer l'exemple de fichier
- Autres requêtes [1]
  - Combiner n fichiers

Expression.Error : Désolé... Nous n'avons pas trouvé la colonne « Produit commandé » de la table.  
Détails :  
Produit commandé

Paramètres d'une requête

PROPRIÉTÉS

Nom  
Combiner n fichiers

Toutes les propriétés

ÉTAPES APPLIQUÉES

- Source
- Fichiers masqués filtrés1
- Appeler une fonction person...
- Colonnes renommées1
- Autres colonnes supprimées1
- Colonne de tables développée1
- Type modifié

= Table.TransformColumnTypes("#Colonne de tables développée1",{{"Source.Name", type text}, {"Prénom du client", type text}, {"Produit commandé", type text}, {"Livraison effective", type date}})


- *S'il y a une suppression de colonne, Power Query est embêté car il ne la retrouve plus et cela crée une erreur* au niveau de « Type modifié ».
- Il faut supprimer la référence à la colonne.

```
= Table.TransformColumnTypes("#Colonne de tables développée1",{{"Source.Name",
type text}, {"Prénom du client", type text}, {"Produit commandé", type
text}, {"Livraison effective", type date}})
```

# Concevoir un modèle adéquat

## Editeur avancé

- On peut y accéder sur le **ruban du haut**, soit depuis l'onglet accueil, soit affichage.
- Il reprend **chaque étape de transformation**, de la connexion au fichier source à la dernière étape de transformation :



The screenshot shows the 'Éditeur avancé' (Advanced Editor) window in Power BI. The title bar reads 'Éditeur avancé' and the window title is 'Expl\_fctTXT'. There is an 'Options d'affichage' dropdown menu and a help icon in the top right corner. The main area contains the following DAX code:

```
let
    Source = Excel.Workbook(File.Contents("E:\FORMATIONS\SUPPORT\Power BI\Power BI - Niveau 2\Matrice de bus.xlsx"), null, true),
    Expl_fctTXT_Table = Source{[Item="Expl_fctTXT",Kind="Table"]}[Data],
    #"Type modifié" = Table.TransformColumnTypes(Expl_fctTXT_Table,{{"ID", Int64.Type}, {"Texte à générer", type text}})
in
    #"Type modifié"
```

# Personnaliser des fonctions et passage de paramètres en langage M

# Concevoir un modèle adéquat

## Transformations personnalisées Fonctions M

- Le lien vers le site Microsoft : Les différents types de fonction <https://learn.microsoft.com/fr-fr/powerquery-m/date-monthname>

### Exemples dans « colonne personnalisée » :

- Text.From pour combiner texte et numérique `= Text.From([ID]) & " " & [Texte à gérer]`
- Text.BetweenDelimiters pour prendre le texte entre deux délimiteurs spécifiques, `= Text.BetweenDelimiters([Texte à gérer], "[", "]")`
- Text.Reverse pour renvoyer la valeur inversée `= Text.Reverse("#ID et entre[] de texte à gérer")`
- Concaténer l'ID avec la colonne Texte à gérer (seulement entre crochets) : `= Text.From([ID]) & " " & Text.BetweenDelimiters([Texte à gérer], "[", "]")`

1 <sup>2</sup> <sub>3</sub> ID	A <sup>B</sup> <sub>C</sub> Texte à gérer	ABC 123 Concaténation
1	bla[ok]lol	1 ok
2	bla[Non]lol	2 Non
3	bla[oui]lol	3 oui

# Concevoir un modèle adéquat

## Fonctions personnalisées et paramètres

- Créer une **première requête (connexion à la source)** qui fera office d'**exemple pour l'itération/compilation des sources**,
- Créer des **paramètres** (si besoin),
- **Transformer la première requête** en **fonction personnalisée** utilisant les paramètres créés précédemment,
- **Créer la requête finale à charger** dans Power BI qui va **appeler la fonction personnalisée** à partir de **colonnes utilisées comme paramètres** dans cette dernière et alimenter la concaténation des différentes sources.
- **Démonstrations en groupe** 😊

# Concevoir un modèle adéquat

## Fonctions personnalisées et paramètres

### ➤ Démonstrations en groupe :

1. Créer une fonction personnalisée avec ses paramètres qui permettent de concaténer **une liste de fichiers** provenant de dossiers différents, ne comprenant pas les mêmes types de fichiers de la même structure, les mêmes noms de dossiers tout en filtrant sur les dossiers concernant l'année en cours et l'année précédente.

→ Dossier Exos\Dossier principal\ -- > [Pour les Données](#)

→ Partie 2 listes fonct param BIG\ fichier « 2. Création fonction et paramètre simple » → [Fichier pas à pas](#)

1. Créer une fonction personnalisée qui va **créer un tableau à partir de toutes les pages du site web correspondant**. Il faudra pouvoir récupérer les numéros de pages qui seront ensuite gérés via le paramètre pour alimenter le fichier final.

→ Partie 2 listes fonct param BIG\ fichier « 3. EXO paramètres et Fonction » [Fichier pas à pas](#)

# Gérer les erreurs

# Concevoir un modèle adéquat

## Les ERREURS (1/3)

Lors de l'application des requêtes dans le cube, il se peut qu'il trouve des erreurs dans les données, il y aura donc un message avec le nombre d'erreurs :

Appliquer les modifications de la requête



✓ Customers  
 ▶ 18 869 lignes chargées. 385 erreurs.

1. Cliquer sur AFFICHER LES ERREURS

1. Power Query crée un fichier Erreur des requêtes, avec la date et l'heure du chargement de l'erreur. Dans ce fichier, il mettra la table où se trouve les Erreurs et seulement les lignes où il y a des erreurs.

Requêtes [2]

- Erreurs des requêtes – 11/11/2019 16:14:30 [1]
  - Erreurs dans Customers
- Autres requêtes [1]
  - Customers

Formulaire: = Table.ReorderColumns("#Conserver les erreurs", {"Numéro de ligne", "CustomerKey", "GeographyKey", "CustomerLabel"})

1.2	Numéro de ligne	CustomerKey	GeographyKey	CustomerLabel
1	18485	18761	424	Error
2	18486	18762	430	Error
3	18487	18763	431	Error
4	18488	18764	432	Error



# Concevoir un modèle adéquat

## Les ERREURS (2/3)

3. Les erreurs sont **à visualiser dans le fichier d'erreur**. On va essayer de comprendre pourquoi il y a une erreur.

4. Les erreurs sont **à corriger et donc à MODIFIER** (Jamais à supprimer) dans la **requête d'origine** qui contient toutes les lignes et les étapes de modifications.

On se positionne sur la table/requête d'origine : on sélectionne la colonne avec des erreurs et on fait **REPLACER LES ERREURS** → si on a fait un calcul sur un champ avec des valeurs NULL et qu'il ne peut pas les convertir en texte, alors on remplace par 'Non renseigné'

**OU on change le type de données** → s'il a mal été défini par Power BI car il le définit sur les 200 premières lignes ; S'il a défini les données en numériques car il n'a pas vu les valeurs caractères plus loin dans la 1000<sup>ème</sup> lignes, etc.

# Concevoir un modèle adéquat

## Les ERREURS (3/3) et **exercice**

5. Allez voir dans le fichier d'erreur. S'il n'y a plus que les entêtes de colonnes, c'est qu'on a bien supprimer les lignes avec des erreurs.

Il n'y a **donc plus d'erreurs, on peut réappliquer les requêtes.**



Requêtes [2]

- Erreurs des requêtes - 11/11/2019 16:14:30 [1]
  - Erreurs dans Customers
- Autres requêtes [1]
  - Customers

Formula Bar: = Table.ReorderColumns(#"Conserver les erreurs", {"Numéro de ligne", "CustomerKey", "GeographyKey"})

Columns: 1.2 Numéro de ligne, 1^2\_3 CustomerKey, 1^2\_3 GeographyKey

6. On supprime **le fichier « Erreurs des requêtes »**

**Le faire ensemble avec notre fichier d'erreur sur la table de vente 😊**

# Concevoir un modèle performant (perf+)

# Concevoir un modèle performant (1/3)

## Connexion à la table

1 Fichier Excel  
(vendeurs Excel)

2. Connexion au tableau  
« Vendeurs » et non pas à la  
Feuille Excel « Feuil1 »

3 Ecrire le code dans l'éditeur  
Avancé d'une « Requête  
vide »

The screenshot shows the Power BI Desktop interface with the 'Création de tableau' ribbon active. The ribbon includes options like 'Tableau croisé dynamique', 'Supprimer les doublons', 'Insérer un segment', 'Exporter', 'Actualiser', 'Propriétés', 'Ouvrir dans le navigateur', and 'Supprimer la liaison'. The 'Options de style de tableau' section is also visible, with checkboxes for 'Ligne d'en-tête', 'Première colonne', 'Bouton de filtre', 'Ligne Total', 'Dernière colonne', 'Lignes à bandes', and 'Colonnes à bandes'. Below the ribbon, a data table is displayed with the following columns and rows:

	A	B	C	D	E	F	G	H
1	<b>Idvendeur</b>	<b>Nom</b>	<b>Prénom</b>	<b>Date d'entrée</b>	<b>Niveau d'étude</b>	<b>Situation familiale</b>	<b>Genre</b>	<b>Nombre d'enfants</b>
2	1	Young	Chloe	02/01/2017 00:00	Bac +2	Mariée	Femme	0
3	2	Gill	Larry	02/01/2017 00:00	Bac +2	Célibataire	Homme	0
4	3	Gonzalez	Geoffrey	02/01/2017 00:00	Bac +3	Marié	Homme	1
5	4	Watson	Alexa	02/01/2017 00:00	Bac +5	Pacsée	Femme	3
6	5	Dominguez	Jacquelyn	02/01/2017 00:00	Bac +5	Pacsé	Femme	2

Let - **Début du code**

Source = Excel.Workbook(File.Contents("E:\FORMATIONS\SUPPORT\Power BI\Power BI - Niveau 2\Exercices et données - ancien (à reprendre pour PERF)\Vendeurs Excel.xlsx"), null, true) – **Chemin d'accès vers le fichier Excel**

{[Item="Vendeurs",Kind="Table"]}[Data] – **Choix de l'item s'intitulant Vendeurs qui est de type Tableau et ouvrir les données correspondantes (Data)**

in

Source – **Fin du code toujours positionné sur la dernière étape. Ici qu'une seule qui est la source.**

# Concevoir un modèle performant (2/3)

## Choix des colonnes avant l'import

**1 Préciser quelles colonnes,** nous souhaitons **sélectionner avant d'importer** les données dans Power BI

**3 Ecrire le code dans l'éditeur Avancé d'une « Requête vide »**

Let - **Début du code**

Source = Excel.Workbook(File.Contents("E:\FORMATIONS\SUPPORT\Power BI\Power BI - Niveau 2\Exercices et données - ancien (à reprendre pour PERF)\Vendeurs Excel.xlsx"), null, true) - **Chemin d'accès vers le fichier Excel**  
 {[Item="Vendeurs",Kind="Table"]}[Data] - **Choix de l'item s'intitulant Vendeurs qui est de type Tableau et ouvrir les données correspondantes (Data)**

[ [Idvendeur], [Nom], [Situation familiale], [Genre] ] → **Choix des colonnes avant l'import sur Power Query**  
 in

Source - **Fin du code toujours positionné sur la dernière étape. Ici qu'une seule qui est la source.**

	ABC 123	Idvendeur	ABC 123	Nom	ABC 123	Situation familiale	ABC 123	Genre
1		1		Young		Mariée		Femme
2		2		Gill		Célibataire		Homme
3		3		Gonzalez		Marié		Homme
4		4		Watson		Pacsée		Femme
5		5		Dominguez		Pacsé		Femme

# Concevoir un modèle performant (3/3)

## Filtres avant l'import

**1 Préciser quel(s) filtre(s) on applique sur la source avant d'importer** les données dans Power BI

**3 Ecrire le code dans l'éditeur Avancé d'une « Requête vide »**

Let

Source = **Table.SelectRows**( → **spécification d'un (ou de) filtre(s) appliqué(s) sur la source**

Excel.Workbook(File.Contents("E:\FORMATIONS\SUPPORT\Power BI\Power BI - Niveau 2\Exercices et données - ancien (à reprendre pour PERF)\Vendeurs Excel.xlsx"), null, true)

{[Item="Vendeurs",Kind="Table"]}[Data]

[ [Idvendeur],[Nom], [Situation familiale], [Genre] ] ] ]

**, each ([Genre] = "Femme")** → **pour chaque ligne, faire un filtre sur la valeur femme de la colonne Genre**

**]** → **fin du (ou des) filtre(s) appliqué(s) sur la source**

in

Source – **Fin du code toujours positionné sur la dernière étape. Ici qu'une seule qui est la source.**

# Travailler avec la dimension Temps



# Travailler avec la dimension Temps

- Utiliser les listes et les fonctions pour réaliser un calendrier en M
- Rappels outils de table et création/identification table de Date/Calendrier.
- Intérêt des mesures, création de table de mesures et rangement de ces dernières dans des dossiers.
- Utiliser les variables en DAX pour réaliser un calendrier.
- Utiliser les fonctions Time Intelligence pour créer des comparaisons de périodes, des cumuls et des périodes glissantes.
- Travailler avec plusieurs dates et plusieurs relations avec USERELATIONSHIP



Utiliser les listes et les fonctions pour réaliser un calendrier en M

# Créer un calendrier sur M

## Créer table de dates sur table de faits (1/6)

La **table de dates doit être constituée** de la manière suivante :

- ✓ Elle débute à la **première date de la table de faits** (ventes par exemple),
- ✓ Elle se termine à **la dernière date de la table de faits** (ventes par exemple),
- ✓ Elle contient l'ensemble des jours entre ces 2 dates

### Conséquence :

- ✓ A chaque **mise à jour de la table de faits** (ventes, inventaire, planning), la table de dates **sera donc recalculée** et contiendra ainsi toutes les dates concernées par l'activité de votre entreprise/organisation/etc.
- ✓ **Pour des dates de début et dates de fin dynamiques !**

# Créer un calendrier sur M

## Fonction List.Dates (2/6)

### ➤ Utilisation de la fonction List.Dates composée de 3 arguments :

- ✓ **Start** (date) : la date de début du calendrier,
  - ✓ **Count** (décimal) : le nombre de dates de la liste. Généralement un calcul entre la date de début et la date de fin. Il faut donc toujours rajouté +1 au résultat pour bien avoir toutes les dates (du 01 au 03 janv : compte 2 mais il y a bien 3 jours),
  - ✓ **Step** (durée) : l'incrément choisi entre chaque date de la liste.
- 
- ✓ **La syntaxe de l'argument step est la suivante :**  
*#duration(days as number, hours as number, minutes as number, seconds as number)*
  - ✓ Pour notre exemple : *#duration(1,0,0,0)* car nous voulons avoir chaque date (d'un jour sur l'autre)

# Créer un calendrier sur M Code avec dates fixes (3/6)

- ✓ **Ecrire directement du code dans l'éditeur avancé :**
- ✓ Spécifier que la **date de début** est 01 janvier 2010,
- ✓ Spécifier que la **date de fin** est le 31 décembre 2030,
- ✓ Créer variable qui **compte le nombre de jours** entre les deux,
- ✓ Puis création de la **liste de dates** avec fonction List.Dates

```
let
    // Création listing de dates sur dates fixes
    DateDebut=#date(2010, 1, 1),
    DateFin=#date(2030, 12, 31),
    NbJour = Duration.Days(DateFin-DateDebut),
    Source = List.Dates(DateDebut, NbJour+1, #duration(1, 0, 0 ,0))
```

*in Source*

- ✓ Dans l'onglet **Outils de liste > Transformer**, cliquer sur **Convertir > Vers la table** puis dans la fenêtre, cliquez sur **OK**

# Créer un calendrier sur M

## Code avec dates dynamiques (4/6)

- ✓ **Ecrire directement du code dans l'éditeur avancé :**
- ✓ Spécifier que la **date de début** est la date la plus petite de la table de ventes,
- ✓ Spécifier que la **date de fin** est la date la plus grande de la table de ventes,
- ✓ Créer variable qui **compte le nombre de jours** entre les deux,
- ✓ Puis création de la **liste de dates** avec fonction List.Dates,

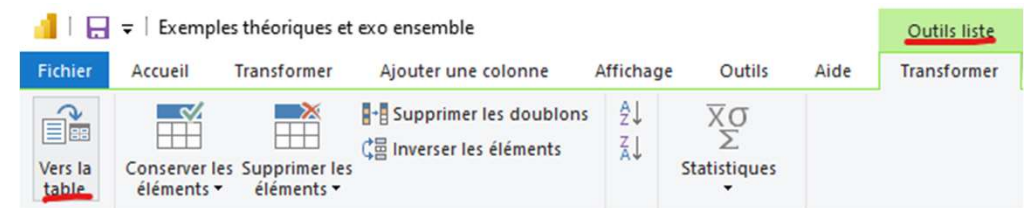
```
let
    // Création listing de dates sur dates dynamiques
    DateDebut=List.Min(Ventes[Date de vente]),
    DateFin=List.Max(Ventes[Date de vente]),
    NbJour = Duration.Days(DateFin-DateDebut),
    Source = List.Dates(DateDebut, NbJour+1, #duration(1, 0, 0 ,0))

in Source
```

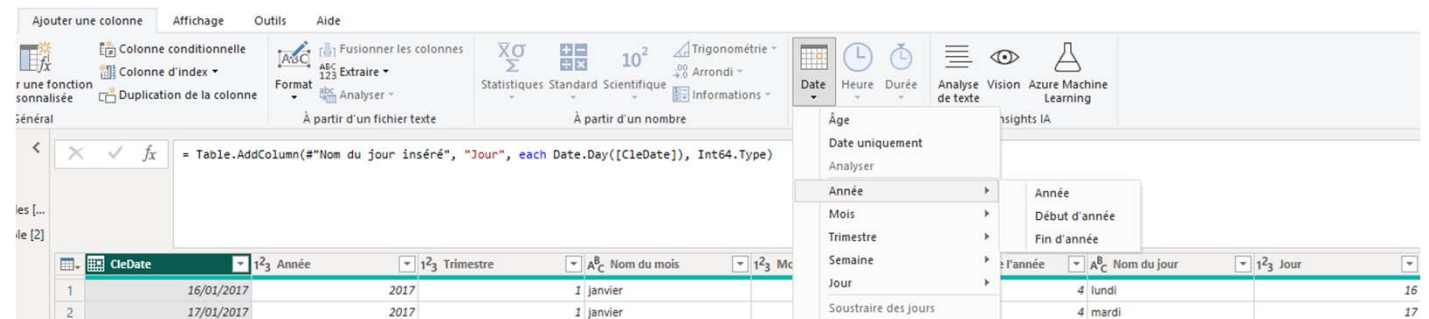
# Créer un calendrier sur M

## Génération table finale (5/6)

- ✓ **Transformer la liste en table** : **Clique-droit sur colonne (vers la table) puis ok**
- ✓ Dans l'onglet **Outils de liste** > **Transformer**, cliquer sur **Convertir** > **Vers la table** puis dans la fenêtre, cliquez sur **OK**
- ✓ **Changer** le type de données
- ✓ **Renommer** en « CleDate »



- ✓ **Ajouter les colonnes d'attributs de date (année, mois, etc.)** :
- ✓ Dans l'onglet **Ajouter une colonne** > **Date** > **Année/Mois/Trimestre/Semaine/Jour**



# Créer un calendrier sur M

## Numéro du jour de la semaine (6/6)

- **Fonction jour de la semaine : Vigilance** = attribue des numéros de 0 à 6.
- ✓ Toujours préciser : `Date.DayOfWeek([CleDate])+1` : 0 devient 1 et 6 = 7 pour le 7ème jour.
- Afin d'éviter les problèmes de version, même s'il compte bien **le premier jour de la semaine pour lundi** (car version Française dans paramètres régionaux), mieux vaut **le rajouter en dur dans le code** :  
`Date.DayOfWeek([CleDate], Day.Monday)+1`
- ✓ Signifie que le premier jour de la semaine est bien le lundi (il commence à compter ce jour-là. Mets 0 par défaut pour le premier jour).

# Créer un calendrier sur M

## Gestion des jours fériés (France) (1/2)

- **Depuis un navigateur Internet : Connexion au site d'API\* du gouvernement !**
  - ✓ Se connecter au **site** <https://api.gouv.fr/documentation/jours-feries>
  - ✓ Cliquer sur **GET** /{zone}.json (liste sur 20 ans en arrière et 5 ans dans le futur)
  - ✓ Par défaut, la zone est la métropole, cliquer sur « **Try it out** » en haut à gauche,
  - ✓ Cliquer sur **Exécute**,
  - ✓ Copier le résultat de la requête URL récupérée : Request URL <https://calendrier.api.gouv.fr/jours-feries/metropole.json>
  
- **Depuis Power BI : Connexion à une nouvelle source : WEB**
  - ✓ Il crée 3 étapes : **ne garder que « Source »** (car il a créé par défaut une table au mauvais format),
  - ✓ Cliquer en haut dans l'onglet « Convertir » sur « En table » puis attribuer les types de données.
  - ✓ Renommer les colonnes : « Date » et « Jours Fériés »
  - ✓ Trier la colonne « Date »

Date	Jours Fériés
01/01/2003	1er janvier
21/04/2003	Lundi de Pâques
01/05/2003	1er mai
08/05/2003	8 mai

\* Un API est une interface de données en code universelle (ici JSON)



# Créer un calendrier sur M

## Gestion des jours fériés (France) (2/2)

- **Fusion des requêtes afin de récupérer la colonne Jour fériés dans la table de dates :**
- Type de jointure : **Externe gauche** car je veux garder tous les jours de ma table de dates même si ce n'est pas un jour férié.
- **Correspondance via la cleDate** dans la table de dates et Date sur table des jours fériés.
- **Création d'une colonne conditionnelle pour colonne finale :**  
**Jours fériés OUI/NON**
  - ✓ S'il y a une valeur dans la colonne récupérée : c'est un jour férié,
  - ✓ S'il n'y a pas : ce n'est pas un jour férié.

A <sup>B</sup> <sub>C</sub> Jours Fériés	A <sup>B</sup> <sub>C</sub> Jours Fériés O/N
Ascension	OUI
	<i>null</i> NON
Lundi de Pentecôte	OUI

## Créer un calendrier sur M – Perf + Gestion des jours ouvrés (France) 1/5 (Perf +)

### ➤ Création des colonnes binaires pour jours ouvrés/fermés :

- ✓ Création de la colonne « **WE** » : si le numéro du jour de semaine est 6 ou 7 alors 1 sinon 0, `#"Création colonne WE" = Table.AddColumn(Source, "WE", each if [Jour de la semaine] = 6 or [Jour de la semaine] = 7 then 1 else 0),`
- ✓ Création de la colonne « **férié** » : si la valeur de la colonne « Jours fériés » est égale à null alors 0 sinon 1 (il y a un nom de jour férié donc c'est férié),

```
#"Création colonne férié 0-1" = Table.AddColumn("#Création colonne WE", "Férié", each if [Jours Fériés] = null then 0 else 1),
```

### ➤ Création des colonnes « jours fermés » et « jours ouvrés » dans table de dates :

- ✓ Création de la colonne « **jours fermés** » : calcul des valeurs des 2 colonnes précédemment créées, (0 ce n'est ni un jour de week-end ni jour férié, 1 = soit l'un soit l'autre, 2 : jour férié et jour de WE) `#"Création jours fermés" = Table.AddColumn("#Type modifié1", "Jours fermés", each [WE]+[Férié]),`
- ✓ Création de la colonne « **jours ouvrables** » : c'est un jour hors WE et jours fériés, (Condition sur la colonne précédente : si 0 alors 1 donc oui, sinon 0)

```
#"Création jours ouvrables" = Table.AddColumn("#Type modifié2", "Jours ouvrables", each if [Jours fermés] = 0 then 1 else 0),
```

# Créer un calendrier sur M – Perf + Gestion des jours ouvrés (France) 2/5 (Perf +)

- **Chargement de la table « livraisons et retours »**
- **Création d'une référence de la « Ventes »** (qui contient toutes les ventes –avant et depuis 2023) afin de construire la table finale avec colonnes calculées.
- ✓ Utiliser « Fusionner les requêtes » pour récupérer les colonnes « date de livraison » et « date de retour » (colonnes de la table « livraisons et retours ».) *Si pas fait au préalable*
- **Création et suppression des colonnes dans la table référence à Ventes :**
- ✓ Création de la colonne « **Nb jours livraison globale** » : mettre des conditions (si pas de date de livraison alors 0 sinon faire le calcul entre les deux dates),
- ✓ **Suppression de toutes les colonnes exceptées** : « Idvente », « Date de vente », « Date de livraison » et « Nb jours livraison globale »,

```

let
    Source = Ventes,
    // il faut bien lui dire que si la date de livraison n'est pas connue, il faut mettre 0 à la place. Si on oublie, cela ne génère pas d'erreur mais on sera bloqué pour créer la liste de dates sans le paramètres sur 0.
    #"Création nb jours livraison global" = Table.AddColumn(Source, "Soustraction", each if [Date de livraison] = null then 0
else Duration.Days([Date de livraison] - [Date de vente])),
    #"Colonnes renommées" = Table.RenameColumns(#"Création nb jours livraison global",{{"Soustraction", "Nb jours livraison globale"}}),
    #"Autres colonnes supprimées" = Table.SelectColumns(#"Colonnes renommées",{"Idvente", "Date de vente", "Date de livraison", "Nb jours livraison globale"}),

```

## Créer un calendrier sur M – Perf + Gestion des jours ouvrés (France) 3/5 (Perf +)

- ✓ **Création d'une colonne de liste :**

- ✓ Il est impératif de **distribuer/développer toutes les dates entre la date de vente et la date de livraison** afin de bien compter ensuite le **nombre de jours ouvrés**.

On part à chaque fois de la date de vente et on rajoute le nombre de jours comptabilisés entre les deux

*List.Dates([Date de vente],[Nb jours livraison globale]+1,#duration(1,0,0,0))*

- ✓ Cliquer sur l'icône à droite de l'entête de « **liste de Dates** » puis « **Développer sur de nouvelles lignes** ».
- ✓ Changer le type de données en « Date »

Idvente	Date de vente	Date de livraison	Nb jours livraison globale	Liste de Dates
1	16/01/2017	19/01/2017	3	16/01/2017
1	16/01/2017	19/01/2017	3	17/01/2017
1	16/01/2017	19/01/2017	3	18/01/2017
1	16/01/2017	19/01/2017	3	19/01/2017
2	17/01/2017	19/01/2017	2	17/01/2017
2	17/01/2017	19/01/2017	2	18/01/2017
2	17/01/2017	19/01/2017	2	19/01/2017

## Créer un calendrier sur M – Perf + Gestion des jours ouvrés (France) 4/5 (Perf +)

### ➤ Connexion à la table de dates :

- ✓ **Connexion à notre table de dates** complète,
- ✓ Récupération de **la colonne « Jours ouvrables »**,

*Ce qui nous permet de savoir, pour chaque jour entre la date de vente incluse et la date de livraison incluse, si ce sont des jours ouvrés (pas WE et pas fériés) ou non.*

### ➤ Regroupement et calcul colonnes sur le nombre de jours de livraison :

- ✓ **Regrouper la table** : Pour chaque vente (regroupement), je souhaite calculer la somme des jours de livraison au global ainsi que la somme des jours de livraisons ouvrés. Je renomme les colonnes.
- ✓ Je **modifie le code du regroupement** : il calcul le nombre de jours ouvrés mais en comptant également le jour de départ (de vente) donc il faut soustraire 1 jour à chaque calcul.

```
= Table.Group("#Récupération jours ouvrables", {"Idvente"}, {"Nb jours livraisons global", each Table.RowCount(_)-1, Int64.Type}, {"Nb jours livraisons ouvrés", each List.Sum([Jours ouvrables])-1, type nullable number}})
```

## Créer un calendrier sur M – Perf + Gestion des jours ouvrés (France) 5/5 (Perf +)

- **Afin d'avoir toutes les colonnes quantitatives dans la table vente**, il me faut réunir toutes les colonnes au même endroit (colonnes de ventes et jours ouvrés).
- Cependant, je ne peux pas récupérer les colonnes de la table de référence de ventes créée précédemment dans la table de vente car cela crée une boucle (**la référence étant liée et dépendante de la table vente**) et bloque le processus.
- On peut néanmoins le faire depuis la **table de référence de ventes**.
- ✓ **Fusion des requêtes** afin de récupérer les colonnes de la table de ventes dans la table de référence en utilisant « idvente » comme colonne de correspondance,
- ✓ Récupérer toutes les colonnes de la table « ventes » exceptée « idvente » déjà présente dans la table de réf.
- ✓ **Créer une colonne conditionnelle : le nombre de jours ouvrés** n'est **pas bon** car il se calcule même quand la date de livraison est inconnue. Il faut donc lui dire de ne rien calculer si il n'existe pas de date de livraison.

```
= Table.AddColumn("#Ventes développé", "Nb jours livraisons ouvrés ok", each if [Date de livraison] <> null then [Nb jours livraisons ouvrés] else null)
```

*Ensuite, bien penser à désactiver le chargement de la table « Ventes » initiale et renommer la nouvelle table créée contenant les notions de jours ouvrés, « ventes ouvrés Pquery ».*

# Rappels outils de table et création/identification table de Date/Calendrier

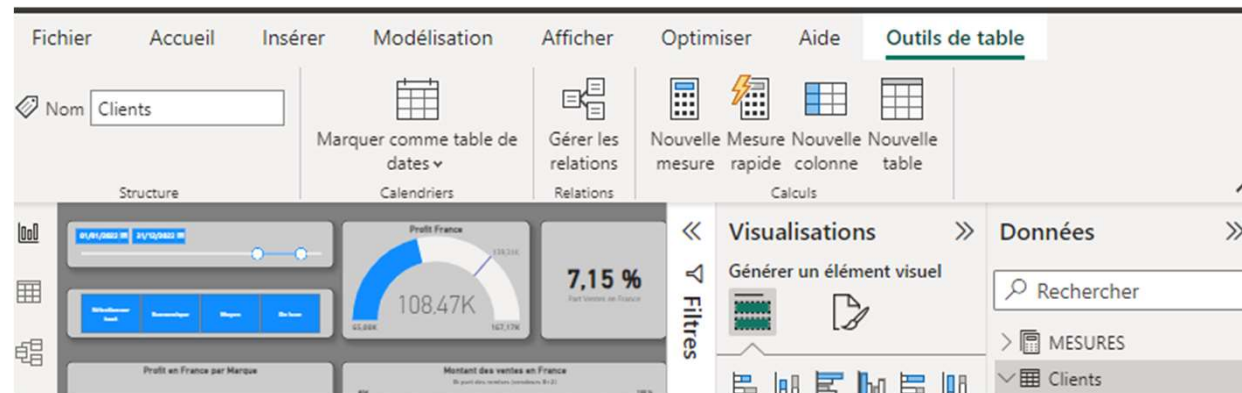
# Les outils de table

## Possibilités

En cliquant sur une table, dans la zone des données à droite, un nouvel onglet en haut apparaît : **“Outils de table”**

**On peut**

- **« Marquer comme table de date »,**
- **avoir accès à la création de mesures, de colonnes et d'autres tables.**





# Les outils de table

## Notions de temporalité et table DATE

La **notion de temporalité et de DATE** est indispensable a tout procces **décisionnel** car :

- On **souhaite récupérer toutes les notions de temporalités** (années, mois, jours, etc. à partir d'une date de vente ou autre transaction a analyser),
- Elle nous permet d'utiliser sans problème **les fonctions Dax Time Intelligence** ensuite dans les mesures DAX.

# Les outils de table

## Création et détermination de la table DATE

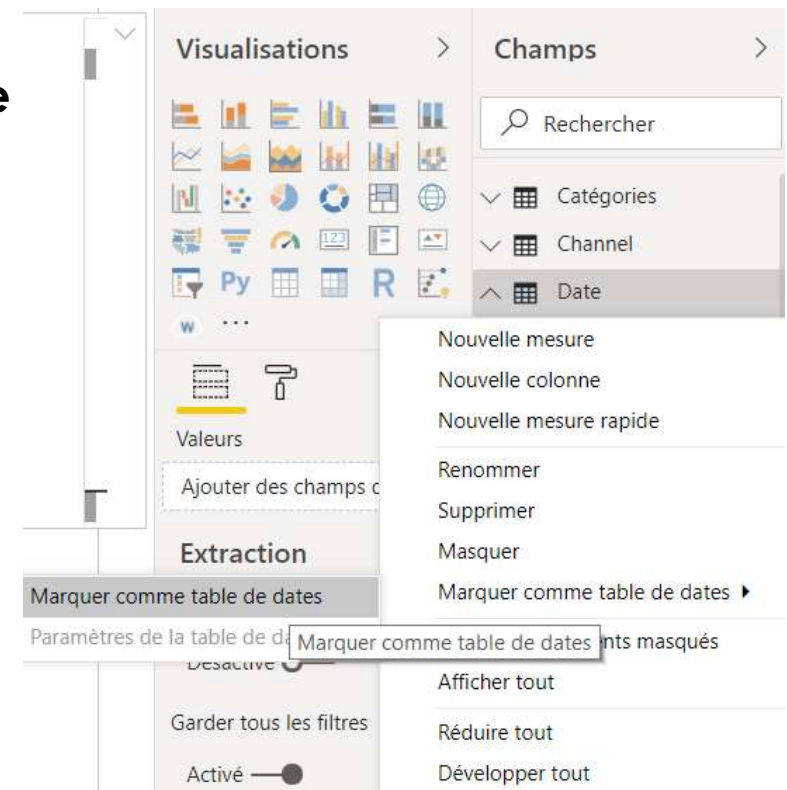
La **table de date doit donc être créée si elle n'existe pas**

**ET**

Il faut dire à Power BI, quelle est notre table de Date.

Il va falloir cliquer droit sur la table de date :

Cliquer sur **Marquer comme table de date**





# Intérêt des mesures Création de table de mesures et rangement

# Mesures DAX

## Intérêt ?

- ▶ **Pour filtrer un calcul sur une ou plusieurs conditions** spécifiques lorsqu'on ne peut pas directement appliquer un filtre sur le visuel (car d'autres mesures utilisées n'ayant pas les mêmes filtres, etc.)
- ▶ Pour **faire des titres dynamiques** qui vont s'actualiser en fonction des filtres choisis par l'utilisateur sur la page du rapport
- ▶ **Pour créer des tables temporaires / en mémoire** afin de faire des calculs très complexes

# Fonctions DAX

## Fonctions statistiques les plus courantes

### ➤ **Plusieurs types de fonctions DAX :**

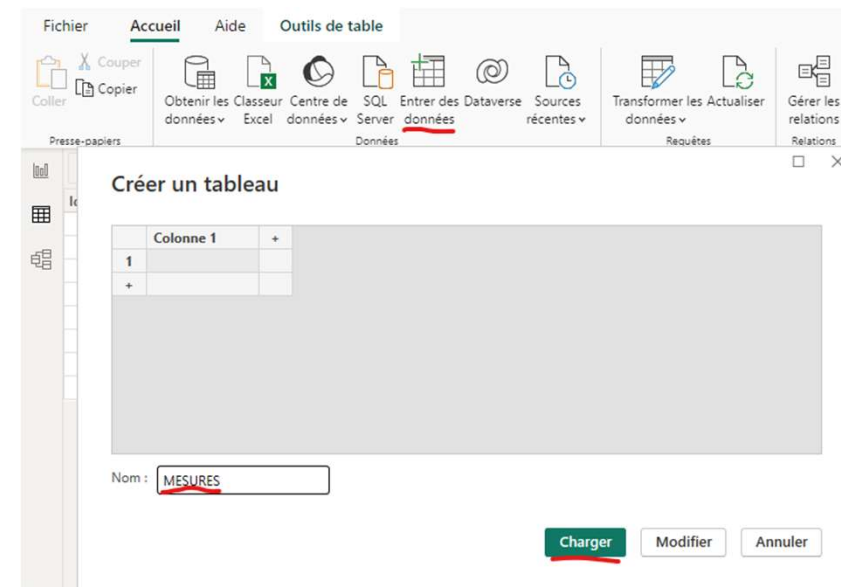
- ✓ Les fonctions d'agrégation (AVERAGE, SUM, SUMX, MAX, DISTINCTCOUNT),
- ✓ Les fonctions de date et heure (CALENDAR, TODAY, MONTH, YEAR, WEEKDAY, TIME),
- ✓ Les fonctions de filtrage (SELECTEDVALUE, ALL, EARLIER, RANK, FILTER, CALCULATE),
- ✓ Les fonctions d'information (ISBLANK, ISERROR, ISFILTERED, ISINSCOPE, CONTAINSSTRING),
- ✓ Les fonctions logiques (AND, IF, NOT, OR),
- ✓ Les fonctions mathématiques (DIVIDE, ROUND, TRUNC),
- ✓ Les fonctions de relation (RELATED, RELATEDTABLE, USERELATIONSHIP, CROSSFILTER),
- ✓ Les fonctions statistiques (MEDIAN, RANKX, CHISQ.INV),
- ✓ Les fonctions de manipulation de tables (ADDCOLUMNS, SUMMARIZECOLUMNS, DISTINCT),
- ✓ Les fonctions de texte (CONCATENATE, LEN, LEFT, REPLACE, TRIM),
- ✓ Les fonctions d'intelligence temporelle (SAMEPERIODLASTYEAR, DATEADD, DATESYTD, DATESINPERIOD).

Les fonctions statistiques : <https://learn.microsoft.com/fr-fr/dax/dax-function-reference>

# Mesures DAX

## Création table de mesures (1/2)

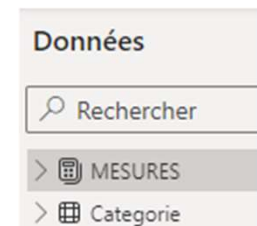
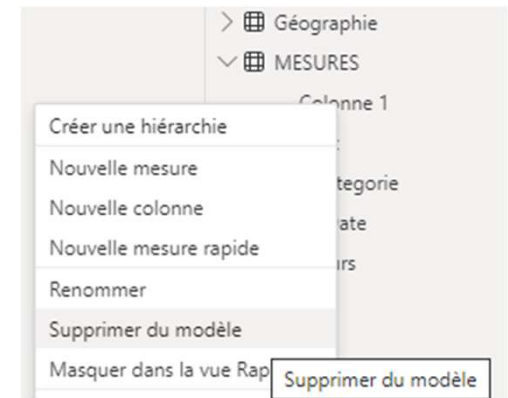
- Les mesures doivent être rangées par type d'indicateurs / thématique afin d'éviter qu'elles soient éparpillées dans toutes les tables. Le but n'est pas de faire perdre du temps au concepteur du rapport mais de lui en faire gagner.
- La méthode la plus simple consiste à cliquer sur « **Entrer des données** » dans l'onglet « accueil » afin de créer une table intitulée « **MESURES** » :



# Mesures DAX

## Création table de mesures (2/2)

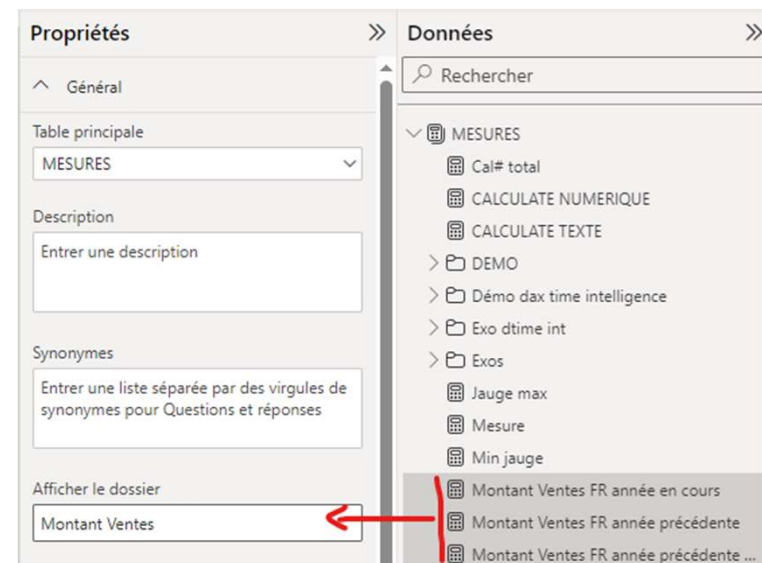
- ▶ La nouvelle table apparaît à droite et ne contient qu'une seule colonne =
- ▶ Après avoir **créé une première mesure**, on peut **supprimer la colonne** inutile.
- ▶ La table remonte tout en haut, au-dessus de la liste des tables afin d'être bien visible.



# Mesures DAX

## Rangement des mesures (1/2)

- ▶ Cliquer sur l'onglet de gauche « **Vue de modèle** » puis ouvrir **la fenêtre « propriétés » à droite** en plus de la fenêtre « données » si elle n'est pas déjà ouverte.
- ▶ **Sélection des mesures** à ranger dans un dossier à droite, dans **la fenêtre « données »** puis dans la fenêtre « propriétés », marquer le nom du dossier avant d'appuyer sur **Entrée sur le clavier**.

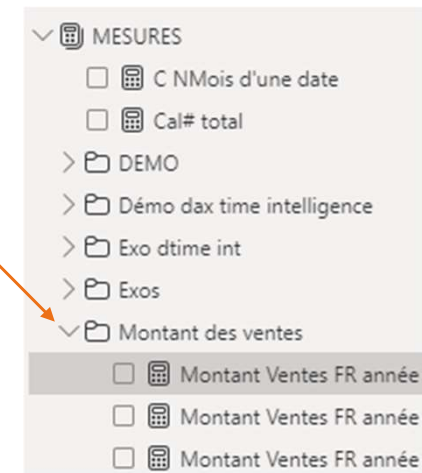




# Mesures DAX

## Rangement des mesures (2/2)

- ▶ Le dossier « **Montant des ventes** » contenant les indicateurs/mesures en lien avec la thématique a été créé.
- ▶ Si j'ai oublié d'y insérer des mesures ou que j'en ai mis en trop, je peux cliquer sur une mesure et **venir la glisser et la déposer dans le dossier simplement avec la souris.**



# Utiliser les variables en DAX pour réaliser un calendrier

# Créer un calendrier avec DAX

## Code standard (1/2)

- **Créer une table (je vais créer ma table de dates dans la partie DAX :**
  - ✓ Spécifier tout d'abord qu'il y aura **plusieurs colonnes** = `ADDCOLUMNS`
  - ✓ Créer une **1<sup>ère</sup> colonne qui crée un listing dynamique** entre une date de début et une date de fin = `CALENDAR`
  - ✓ Créer ensuite l'année, le trimestre, les mois, les jours, etc.
- **Spécificités du code standard :**
  - ✓ Concernant la fonction `CALENDAR`, on va chercher la 1<sup>ère</sup> Date de vente et la dernière date de vente = **mise à jour automatique quand actualisation de la table ventes**,
  - ✓ Concernant la fonction `WEEKNUM` (numéro de semaine) : le **bon numéro ISO** est obtenu en indiquant **21** dans la fonction,
  - ✓ Concernant la fonction `WEEKDAY` (numéro du jour de la semaine), mettre **2** pour indiquer que l'on souhaite le format Français (le 1<sup>er</sup> jour est bien le lundi)

# Créer un calendrier avec DAX

## Code standard (2/2)

```
Table Date = ADDCOLUMNS ( -- création table
CALENDAR (FIRSTDATE(Ventes[Date de vente]);
LASTDATE(Ventes[Date de vente]));
```

```
"Année" ; YEAR ( [Date] );
```

```
"Numéro du mois" ; FORMAT ( [Date]; "MM" );
```

```
"Nom du mois" ; FORMAT ( [Date]; "mmmm" );
```

```
"Numéro de la semaine"; weeknum([Date] ; 21);
```

```
"Numéro du jour du mois"; DAY([Date]);
```

```
"Numéro du jour de la semaine"; WEEKDAY ( [Date] ; 2 );
```

```
"Nom du jour de la semaine"; FORMAT ( [Date]; "dddd" );
```

```
"Trimestre"; "Trimestre " & FORMAT ( [Date]; "Q" );
```

**FIRSTDATE** ou **MIN** pour aller chercher la plus petite valeur d'une colonne  
**LASTDATE/MAX** = +haute

Si **fixe** : calendar( DATE(2015;09;15) ; DATE(2024 ;12 ;31) )

**YEAR** identique à la fonction **ANNEE** dans EXCEL

**MM** : numéro, **MMM** : nom abrégé, **MMMM** : nom entier (Pareil pour les jours DD, DDD, DDDD)

**Concaténation** entre le mot Trimestre suivi d'un espace puis le numéro de trimestre obtenu.

# Créer un calendrier avec DAX

## Ajout WE et jours fériés (1/3) (Perf +)

### ➤ Création de colonnes conditionnelles dans la table date :

- ✓ Création **de la colonne « WE »** en fonction de la colonne N° du jour de la semaine

```
Week-end = IF('Table Date'[Numéro du jour de la semaine]=6 || 'Table Date'[Numéro du jour de la semaine]=7; 1 ; 0) OU en DAX
```

- ✓ Création **de la colonne « férié »**. Après **avoir fait une relation** entre la table de DATE et la table des jours fériés, on va récupérer le nom du jour férié en utilisant **RELATED** :

```
Férié = RELATED('3 Jours feriés metropole'[Jours Feriés])
```

- ✓ Amélioration de la colonne « férié » : si le related ne remonte pas de valeur alors 0 car pas férié sinon 1

```
Férié = IF(ISBLANK(RELATED('3 Jours feriés metropole'[Jours Feriés])); 0; 1)
```

ISBLANK signifie est vide // donc ici: si le résultat du related est vide

# Créer un calendrier avec DAX

## Ajout WE et jours fériés (2/3) (Perf +)

### ➤ Création de colonnes conditionnelles dans la table date:

✓ Création de la colonne « jours fermés » : Si WE ou Férié alors 1 sinon 0

Jours fermés = `if('Table Date'[Week-end] = 1 || 'Table Date'[Férié]=1; 1;0)`

✓ Création de la colonne « jours ouvrés » : si Jours fermés =1 alors 0 sinon 1

Jours ouvrés = `IF('Table Date'[Jours fermés]=1;0;1)`

### ✓ Création des colonnes finales dans la table de ventes :

✓ Création de la colonne « nb jours livraison globale DAX » :

NB jours livraison DAX = `DATEDIFF('Ventes'[Date de vente]; 'Ventes'[Date de livraison]; DAY)`

# Créer un calendrier avec DAX

## Ajout WE et jours fériés (3/3) (Perf +)

- ✓ Création de la colonne « nb jours livraison ouvrés » :

NB jours livraison ouvrés DAX = NETWORKDAYS('Ventes'[Date de vente];  
'Ventes'[Date de livraison]; 1) -1

Mettre 1 ou rien pour we = samedi et dimanche // rajout -1 car la fonction compte le jour de départ

- ✓ Rajout **des jours fériés** dans le calcul :

NB jours livraison ouvrés DAX = NETWORKDAYS('Ventes'[Date de vente];  
'Ventes'[Date de livraison]; 1; 'Jours fériés pour networkdays')-1

Il faut impérativement que la table des jours fériés ne contienne qu'une seule colonne pour que ça fonctionne et qu'il la considère comme une liste.

- ✓ Rajout condition = si pas de date de livraison alors pas de calcul (sinon 32201 à chaque fois)

```
IF(ISBLANK('Calcul jours ouvrés'[Date de livraison])); BLANK();  
NETWORKDAYS('Calcul jours ouvrés'[Date de vente]; 'Calcul jours ouvrés'[Date de  
livraison]; 1; '3 Jours feriés POUR TABLE DAX')-1 )
```

# Créer un calendrier avec DAX

## Autres possibilités sur fonction NetworkDays

### ✓ Utilisation de la fonction NetworkDays :

NB jours livraison ouvrés DAX = NETWORKDAYS('Calcul jours ouvrés'[Date de vente]; 'Calcul jours ouvrés'[Date de livraison]; 1) -1

Mettre 1 ou rien pour we = samedi et dimanche // rajout -1 car la fonction compte le jour de départ

Selon ce que l'on considère jours ouvrés ou non, il y a plusieurs possibilités à part le 1 →

- 1 ou omis : samedi, dimanche
- 2 : dimanche, lundi
- 3 : lundi, mardi
- 4 : mardi, mercredi
- 5 : mercredi, jeudi
- 6 : jeudi, vendredi
- 7 : vendredi, samedi
- 11 : dimanche uniquement
- 12 : lundi uniquement
- 13 : mardi uniquement
- 14 : mercredi uniquement
- 15 : jeudi uniquement
- 16 : vendredi uniquement
- 17 : samedi uniquement



# Fonctions Time Intelligence : Comparaisons, cumuls et périodes glissantes

# Fonctions DAX Time Intelligence

## Importance des filtres pour ces fonctions (1/2)

- **Les fonctions Dax Time Intelligence** aident au quotidien afin de filtrer sur des temporalités spécifiques, afficher des périodicités antérieures ou postérieures, de calculer des cumulatifs, d'évaluer des évolutions etc.
- Néanmoins, ces fonctions ne peuvent pas être utilisées seules et **sans contexte de filtre**.
- **Rappels** : un **filtre peut être 3 choses différentes** dans Power BI :
  - ✓ Un filtre de visuel, de page ou de rapport (Fenêtre des filtres à droite) qui peuvent être considérés plutôt comme des filtres de conception,
  - ✓ Un segment : c'est un visuel qui permet à l'utilisateur d'appliquer des filtres pour actualiser une ou plusieurs pages de rapport. Peut être considéré comme filtre de visualisation / utilisateur final,
  - ✓ Les éléments qualitatifs des visuels : les éléments en lignes ou colonnes dans une matrice, les éléments en axe ou légende dans des graphiques. Les catégories de produits dans un graphique vont filtrer la somme des ventes pour chaque histogramme concerné : c'est bien un filtre.

# Fonctions DAX Time Intelligence

## Importance des filtres pour ces fonctions (2/2)

- Afin qu'une **fonction Dax TIME INTELLIGENCE puisse fonctionner**, il est nécessaire d'être attentif à tous les éléments suivants :
- ✓ Avoir une **table de date identifiée** comme telle dans Power BI (et **reliée** aux autres tables 😊 ),
- ✓ **Utiliser un filtre** (un des 3 explicités dans la diapositive précédente) afin que la fonction puisse se baser dessus pour calculer correctement  
*Si filtre sur 2023 et qu'on utilise Sameperiodlastyear, la fonction calculera l'indicateur sur 2022.*
- ✓ Le **filtre utilisé** doit concerner une **colonne** de la **table de date**,
- ✓ La **colonne utilisée** dans la formule pour créer la **mesure** Dax TIME INTELLIGENCE doit provenir également de la **table de date**.  
*Si on calcule, à partir de la date de vente : on a créé au préalable, une relation active entre cette colonne et la colonne Date de la table date. Cette dernière décompose bien notre date de vente. Il faut donc bien sélectionner la date de la table date.*

# Fonctions DAX Time Intelligence

## Réutilisation de mesures existantes (1/2)

Il est intéressant de décomposer les calculs afin de vérifier que chaque partie de l'indicateur final fonctionne bien et de gagner en lisibilité des mesures.

➤ Création de la **mesure 1** :

Somme montants ventes = `SUM('Ventes'[MontantVentes])`

➤ Création de la **mesure 2** : *(même calcul pour la même période l'année dernière)*

Somme montants ventes Y-1 = `CALCULATE( SUM('Ventes'[MontantVentes]); SAMEPERIODLASTYEAR('Date'[Cledate]))`

Ou ré-utiliser la mesure 1 (*Somme montants des ventes*)

Somme montants ventes Y-1 = `CALCULATE( [Somme montants ventes]; SAMEPERIODLASTYEAR('Date'[Cledate]))`

# Fonctions DAX Time Intelligence

## Réutilisation de mesures existantes (1/2)

Je souhaite connaître ce que représente le montant de l'année en cours par rapport au volume de l'année dernière (comparaison) :

➤ Tout réécrire :

= SUM('Ventes'[MontantVentes])

/

CALCULATE(SUM('Ventes'[MontantVentes]);  
SAMEPERIODLASTYEAR ('Date'[Cledate]))

Ou ré-utiliser les mesures

= [Somme montants ventes]/[Somme montants ventes Y-1]

- ✓ Compréhension plus rapide du calcul
- ✓ Facilité d'écriture

Année	Somme montants ventes	Somme montants ventes Y-1	ratio
2018	3 964 184,67	1 172 134,36	338,20 %
2019	2 737 707,80	3 964 184,67	69,06 %
2020	2 938 696,92	2 737 707,80	107,34 %
2021	2 613 513,61	2 938 696,92	88,93 %
2022	2 620 676,25	2 613 513,61	100,27 %
2023	2 877 253,21	2 620 676,26	109,79 %
2024	249 214,89	298 421,00	83,51 %
<b>Total</b>	<b>18 001 247,35</b>	<b>16 345 334,62</b>	<b>110,13 %</b>

À l'actualisation des données en 2024, on avait réalisé que 83,5 % du montant des ventes réalisé sur la même période l'année précédente.

# Fonctions DAX Time Intelligence

## Création de variables après contrôle (Perf +)

Après avoir séparé chaque mesure et vérifié chaque calcul, créer une seule mesure contenant les parties du calcul en variables :

- Je souhaite calculer le **pourcentage d'évolution d'une année sur l'autre.**
- Je crée **deux variables** qui correspondent à mes deux mesures contrôlées en amont

Evolution d'une année sur l'autre =

```
VAR montantventes = SUM('Ventes'[MontantVentes])
```

```
VAR MontantventesY1 = CALCULATE (SUM('Ventes'[MontantVentes]);  
SAMEPERIODLASTYEAR('Date'[Date]))
```

```
RETURN -- calcul final utilisant les variables créées précédemment  
(montantventes - MontantventesY1)/MontantventesY1
```

- ✓ Gain en performance et stockage sur de grandes volumétries
- ✓ Réutilisation des variables uniquement dans le return (pas d'une variable sur l'autre) si fonction Dax Time Intelligence

# Fonctions DAX Time Intelligence SAMEPERIODLASTYEAR/DATEADD

Année	Somme montants ventes	Somme montants ventes Y-1	Somme montants ventes Y-2 DA
2021	2 613 513,61	2 938 696,92	2 737 707,80
2022	2 620 676,25	2 613 513,61	2 938 696,92
2023	2 877 253,21	2 620 676,26	2 613 513,61
2024	249 214,89	298 421,00	350 884,98
<b>Total</b>	<b>8 360 657,96</b>	<b>8 471 307,79</b>	<b>8 640 803,31</b>

Actualisation au **13 février 2024** :

Et on est à 249 214,89 Euros de montant des ventes.

**Fin février** en 2023, on était à 298 421

## ➤ SAMEPERIODLASTYEAR :

Montant ventes n-1 = `CALCULATE( [Somme montants ventes] ; SAMEPERIODLASTYEAR('Date'[Cledate]))`

Fonctionne pour toutes les périodicités (année, trimestre, mois, semaines, jours), **sur la même période l'année précédente**

✓ Si l'année en cours n'est pas finie, et que les données courent jusqu'au 13 février, alors il compare avec le volume total de février de l'année dernière (et non année entière).

## ➤ DATEADD :

`CALCULATE( [Somme montants ventes] ; DATEADD('Date'[Cledate] ; -2; YEAR))`

Fonctionne à l'identique de Sameperiodlastyear mais on **peut sélectionner la périodicité principale** : il y a deux ans, il y a 3 mois, etc. puis il calcul les hiérarchies inférieures.

# Fonctions DAX Time Intelligence

## PARALLELEPERIOD = La moins utilisée

### ➤ PARALLELEPERIOD :

```
CALCULATE( [Somme montants ventes] ;  
PARALLELEPERIOD('Date'[Cledate]; -1;YEAR))
```

✓ On peut sélectionner la périodicité de calcul voulue comme pour DATEADD = année, trimestre, mois, etc. et l'intervalle (-2 ans, + 2 mois, etc.)

✓ Si l'année en cours n'est pas finie, et que les données courent jusqu'au 13 février, alors ici, il donne la volumétrie totale de l'année précédente et non pas jusqu'à la fin février comme DATEADD et SAMEPERIODLASTYEAR.

✓ IL ne sait pas calculer sur les périodicités inférieures (hiérarchies) : si je lui demande de calculer l'année dernière et que j'ouvre les mois, il ne calculera pas par mois de l'année dernière.

Année	Somme montants ventes	Somme montants ventes Y-1 PP	Somme montants ventes Y-1
2023	2 877 253,21	2 620 676,26	2 620 676,26
2024	249 214,89	2 877 253,21	298 421,00
janvier	191 708,21	2 877 253,21	158 469,25
février	57 506,68	2 877 253,21	139 951,75
01/02/2024 00:00:00	1 274,57	2 877 253,21	847,75
02/02/2024 00:00:00	6 825,90	2 877 253,21	7 799,90

Actualisation au **13 février 2024** :

Et on est à 249 214,89 Euros de montant des ventes.

Qu'importe la périodicité sélectionnée (hiérarchie du tableau), il me donne la **valeur totale de l'année précédente**.



# Fonctions DAX Time Intelligence DATESYTD et les divers cumuls

Année	Somme montants ventes	Ventes CumulYTD
2023	2 877 253,21	2 877 253,21
Trimestre 1	490 231,80	490 231,80
janvier	158 469,25	158 469,25
février	139 951,75	298 421,00
mars	191 810,80	490 231,80
Trimestre 2	836 181,66	1 326 413,46
avril	279 944,80	770 176,60
mai	223 012,34	993 188,94
juin	333 224,52	1 326 413,46

Sur **la seconde colonne**, nous pouvons voir le **calcul cumulatif sur le mois de février** : 298 421 correspondants **à la somme du mois de janvier** (158 469,25) **et du mois de février** (139 951,75)

➤ **DATESYTD** : permet de faire des cumuls dans l'année

```
CALCULATE( [Somme montants ventes] ;  
            DATESYTD('Date'[Cledate] ) )
```

➤ Il en existe 3 différents pour 3 niveaux de temporalités de départ :

- ✓ **DATESYTD** : Year-to-date : Calcule le cumul de début de l'année jusqu'à la dernière date. Se calcule pour chaque niveau de hiérarchie inférieur trimestre, mois, etc.
- ✓ **DATESQTD** : Quarter-to-date : Fonctionne de même mais ne calcule qu'à l'intérieur d'un trimestre
- ✓ **DATESMTD** : Month-to-date : Fonctionne de même mais ne calcule qu'à l'intérieur d'un mois

# Fonctions DAX Time Intelligence

## DATESBETWEEN : cumul entre deux dates

### ➤ DATESBETWEEN :

DATESBETWEEN(<dates>, <start\_date>, <end\_date>)

**DATESBETWEEN dates précises/fixes** = CALCULATE([Somme montants ventes];  
DATESBETWEEN('Date'[Cledate] ; DATE(2024;02;01); DATE(2024;02;10)))

- ✓ Elle est utilisée spécifiquement pour annoncer des dates très précises. On sort donc des temporalités proposées habituellement : année, trimestre, mois, etc.
- ✓ Elle fonctionne également avec du dynamique comme DATESINPERIOD mais en incluant date de fin période précédente

**DATESBETWEEN dynamique** = CALCULATE( [Somme montants ventes];  
DATESBETWEEN('Date'[Cledate] ;  
DATEADD(LASTDATE('Date'[Cledate] );-1; year);  
LASTDATE('Date'[Cledate] )))

Année	Somme montants ventes	DATESBETWEE N dynamique	DATESBETWEEN dyn 1ere date	DATESBETWEEN dyn last date
2024	249 214,89	2 902 491,72	13/02/2023	13/02/2024
janvier	191 708,21	2 912 092,17	31/01/2023	31/01/2024
février	57 506,68	2 902 491,72	13/02/2023	13/02/2024
01/02/2024 00:00:00	1 274,57	2 911 766,74	01/02/2023	01/02/2024
02/02/2024 00:00:00	6 825,90	2 917 744,89	02/02/2023	02/02/2024

# Fonctions DAX Time Intelligence

## DATESINPERIOD : cumul sur période glissante

➤ **DATESINPERIOD** : DATESINPERIOD(dates, start\_date, number\_of\_intervals, interval)

Calculer la somme du montant des ventes sur les 12 derniers mois

```
DATESINPERIOD exp = CALCULATE( [Somme montants ventes];
DATESINPERIOD('Date'[Cledate] ;
LASTDATE('Date'[Cledate]) ; -1; year))
```

- ✓ Fonctionne pour spécifier une périodicité spécifique et dynamique et permet de calculer un cumulatif sur plusieurs mois (trimestres, années) glissants
- ✓ Attention, si on est en cours du mois comme le 13 février alors la date de début sur le mois sera le 01 mars de l'année précédente et non le 14 février. Il ne descend pas sur la date exacte.

Année	Somme montants ventes	DATESINPERIOD exp	DATESINPERIOD 1ère date	DATESINPERIOD last date
2024	249 214,89	2 828 047,10	01/03/2023	13/02/2024
janvier	191 708,21	2 910 492,17	01/02/2023	31/01/2024
février	57 506,68	2 828 047,10	01/03/2023	13/02/2024
01/02/2024 00:00:00	1 274,57	2 910 918,99	02/02/2023	01/02/2024
02/02/2024 00:00:00	6 825,90	2 909 944,99	03/02/2023	02/02/2024

# Fonctions DAX Time Intelligence

## OPENING /CLOSINGBALANCEMONTH

➤ **Sur le cumulatif, je souhaite connaître la valeur cumulée au début du mois et à la fin du mois :**

➤ **Ventes cumulées au début du mois :**

VentesCUMUL DEBUTmois = **OPENINGBALANCEMONTH** ([Ventes Cumu]YTD) ;  
'Date'[Cledate] )

*Permet d'avoir le dernier cumul connu avant la première date du mois (donc valeur pour le mois précédent par exemple)*

➤ **Ventes cumulées à la fin du mois :**

VentesCUMUL FINmois = **CLOSINGBALANCEMONTH** ([Ventes Cumu]YTD) ; 'Date'[Cledate] )

*Permet d'avoir le dernier cumul connu à la fin du mois*

*Attention, cette fonction n'a de sens que pour la gestion des stocks. Là on aura la même valeur que pour le cumul mais en décalé.*

✓ *Des filtres peuvent être ajoutés dans la fonction après la DATE.*

☐ janvier	158 469,25	158 469,25	2 620 676,26
☐ février	139 951,75	298 421,00	158 469,25
☐ mars	191 810,80	490 231,80	298 421,00

# Fonctions DAX Time Intelligence

## Cumul à l'année VS cumul mois glissants

Année	Somme de MontantVentes	Ventes CumulYTD	Ventes 12lastmonth Matrice
2023	2 877 253,21	2 877 253,21	
janvier	158 469,25	158 469,25	2 577 704,76
février	139 951,75	298 421,00	2 568 212,27
mars	191 810,80	490 231,80	2 472 789,31
avril	279 944,80	770 176,60	2 511 186,69
mai	223 012,34	993 188,94	2 534 610,08
juin	333 224,52	1 326 413,46	2 693 547,63
juillet	351 764,14	1 678 177,60	2 898 526,42
août	248 086,59	1 926 264,19	3 068 311,64
septembre	311 803,55	2 238 067,74	3 002 512,18
octobre	265 190,03	2 503 257,77	2 886 860,01
novembre	175 423,03	2 678 680,80	2 881 213,58
décembre	198 572,41	2 877 253,21	2 877 253,21
2024	249 214,89	249 214,89	
janvier	191 708,21	191 708,21	2 910 492,17
février	57 506,68	249 214,89	2 828 047,10
<b>Total</b>	<b>3 126 468,10</b>	<b>249 214,89</b>	

### ➤ DATESYTD pour le cumul à l'année :

Ventes CumulYTD = `CALCULATE([Somme montants ventes]; DATESYTD('Date'[Cledate] ))`

✓ *Fonction déjà calculée ensemble. Elle commence à calculer en janvier donc plus il y a de mois cumulés, plus la valeur augmente forcément.*

### ✓ DATESINPERIOD pour le cumul sur les mois glissants :

Ventes 12lastmonth = `CALCULATE([Somme montants ventes]; DATESINPERIOD('Date'[Cledate] ; LASTDATE('Date'[Cledate] ); -12; MONTH)) = (-1, Year)`

✓ *Fonction déjà calculée ensemble. La période de calcul change chaque mois en fonction du mois sur lequel on est. La valeur du cumul n'augmente donc pas car le premier mois et le dernier mois pris en compte ne sont plus les mêmes.*

# Fonctions DAX Time Intelligence

## Préciser calcul selon niveau hiérarchie (Perf +)

Année	Somme de MontantVentes	Ventes CumulYTD	Ventes 12lastmonth Matrice
2023	2 877 253,21	2 877 253,21	
janvier	158 469,25	158 469,25	2 577 704,76
février	139 951,75	298 421,00	2 568 212,27
mars	191 810,80	490 231,80	2 472 789,31
avril	279 944,80	770 176,60	2 511 186,69
mai	223 012,34	993 188,94	2 534 610,08
juin	333 224,52	1 326 413,46	2 693 547,63
juillet	351 764,14	1 678 177,60	2 898 526,42
août	248 086,59	1 926 264,19	3 068 311,64
septembre	311 803,55	2 238 067,74	3 002 512,18
octobre	265 190,03	2 503 257,77	2 886 860,01
novembre	175 423,03	2 678 680,80	2 881 213,58
décembre	198 572,41	2 877 253,21	2 877 253,21
2024	249 214,89	249 214,89	
janvier	191 708,21	191 708,21	2 910 492,17
février	57 506,68	249 214,89	2 828 047,10
<b>Total</b>	<b>3 126 468,10</b>	<b>249 214,89</b>	

➤ Si je laisse **le calcul sans précision et que j'utilise une matrice**, il va afficher le total pour l'année car n'arrive pas à calculer sur mois glissants. Pour lui l'année, c'est l'année, ce n'est pas les 3 premiers mois en cours + les 9 mois précédents.

➤ Il est donc nécessaire de lui dire sur **quels niveaux de hiérarchies** faire le calcul et l'afficher avec **IF & ISINSCOPE**

➤ **Je lui dis de calculer que si il est sur le niveau des mois ou des trimestres :**

```
= IF( ISINSCOPE('Date'[Nom du mois]) || ISINSCOPE('Date'[Trimestre]); CALCULATE([Somme montants ventes];DATESINPERIOD('Date'[Cledate]; LASTDATE('Date'[Cledate]);-12;MONTH)) )
```

*Ne rien mettre à la fin du IF )) correspond à )Blanck() ) = vide/rien faire*



# Fonctions DAX Time Intelligence

## Cumul sans fin (Perf + )

- Je souhaite que le cumul se fasse sur toutes les années ?  
Ne s'arrête jamais ? Voilà la formule :

Cumul sans fin = `CALCULATE( [Somme montants ventes] ;`

`FILTER( ALL('Date'[Cledate] );`

- je positionne un filtre pour le calcul
- Je lui dit de prendre en compte toutes les dates existantes

`'Date'[Cledate] <= MAX('Date'[Cledate] )))`

- Si elles sont inférieures ou égales à la valeur en ligne (si 2018 : calcul toutes les dates inférieurs ou égales à 2018).

Année	Cumul sans fin
2017	1 172 134,36
2018	5 136 319,03
2019	7 874 026,83
2020	10 812 723,75
2021	13 426 237,36
2022	16 046 913,62
2023	18 924 166,83
2024	19 173 381,72
<b>Total</b>	<b>19 173 381,72</b>

# Concevoir un modèle adéquat

## Exercices

Année	Somme Montant Coûts	Somme Montant Coûts AnnéePrécédente	Marge Progression montant des coûts	Cumul année Montant Coûts	Cumul 6moisglissants Montant Coûts
2022	1 057 677,13	1 046 124,34	1,10 %	1 057 677,13	557 099,19
2023	1 174 222,11	1 057 677,13	11,02 %	1 174 222,11	605 260,57
janvier	64 828,40	79 714,99	-18,67 %	64 828,40	559 923,71
février	57 972,28	66 825,73	-13,25 %	122 800,68	585 719,38
mars	81 040,15	109 431,42	-25,94 %	203 840,83	511 349,46
avril	124 861,41	94 363,16	32,32 %	328 702,24	483 832,00
mai	105 844,02	79 063,54	33,87 %	434 546,26	518 127,36
juin	134 415,28	71 179,10	88,84 %	568 961,54	568 961,54
juillet	133 682,87	62 003,88	115,60 %	702 644,41	637 816,01
août	95 065,93	32 176,61	195,45 %	797 710,34	674 909,66
septembre	119 164,73	155 410,07	-23,32 %	916 875,07	713 034,24
octobre	106 454,23	152 378,87	-30,14 %	1 023 329,30	694 627,06
novembre	69 645,65	71 548,66	-2,66 %	1 092 974,95	658 428,69
décembre	81 247,16	83 581,10	-2,79 %	1 174 222,11	605 260,57

➤ Créer les mesures suivantes dans une matrice contenant une hiérarchie **Année/Trimestre/Mois/Jour en ligne** :

- 1) Créer la somme des montants des coûts au total,
- 2) Créer la somme du montant des coûts de l'année précédente,
- 2) Calculer la marge de progression du montant des coûts (calcul général année sur l'autre),
- 4) Créer le cumul à l'année du montant des coûts,
- 5) Créer le cumul sur 6 mois glissants du montant des coûts.



# Concevoir un modèle adéquat

## Exercices avec corrections

Année	Somme Montant Coûts	Somme Montant Coûts AnnéePrécédente	Marge Progression montant des coûts	Cumul année Montant Coûts	Cumul 6moisglissants Montant Coûts
2022	1 057 677,13	1 046 124,34	1,10 %	1 057 677,13	557 099,19
2023	1 174 222,11	1 057 677,13	11,02 %	1 174 222,11	605 260,57
janvier	64 828,40	79 714,99	-18,67 %	64 828,40	559 923,71
février	57 972,28	66 825,73	-13,25 %	122 800,68	585 719,38
mars	81 040,15	109 431,42	-25,94 %	203 840,83	511 349,46
avril	124 861,41	94 363,16	32,32 %	328 702,24	483 832,00
mai	105 844,02	79 063,54	33,87 %	434 546,26	518 127,36
juin	134 415,28	71 179,10	88,84 %	568 961,54	568 961,54
juillet	133 682,87	62 003,88	115,60 %	702 644,41	637 816,01
août	95 065,93	32 176,61	195,45 %	797 710,34	674 909,66
septembre	119 164,73	155 410,07	-23,32 %	916 875,07	713 034,24
octobre	106 454,23	152 378,87	-30,14 %	1 023 329,30	694 627,06
novembre	69 645,65	71 548,66	-2,66 %	1 092 974,95	658 428,69
décembre	81 247,16	83 581,10	-2,79 %	1 174 222,11	605 260,57

- ✓ Somme Montant Coûts =  $SUM('2Ventes\ ouvertes\ DAX'[CoutTotal])$
- ✓ Somme Montant Coûts AnnéePrécédente =  $CALCULATE([Somme\ Montant\ Coûts];\ SAMEPERIODLASTYEAR('Date'[Cledate] ))$
- ✓ Marge Progression montant des coûts =  $([Somme\ Montant\ Coûts]-[Somme\ Montant\ Coûts\ AnnéePrécédente])/[Somme\ Montant\ Coûts\ AnnéePrécédente]$
- ✓ Cumul année Montant Coûts =  $CALCULATE([Somme\ Montant\ Coûts];\ DATESYTD('Date'[Cledate]))$
- ✓ Cumul 6moisglissants Montant Coûts =  $CALCULATE([Somme\ Montant\ Coûts];\ DATESINPERIOD('Date'[Cledate];\ LASTDATE('Date'[Cledate] );-6;MONTH))$

122

122

# filtrer les données sur la période en cours

# Fonctions DAX Time Intelligence

## Filtrer sans DAX TIME

➤ Je souhaite **afficher la somme des ventes pour l'année en cours et l'année précédente** (sans l'aide d'un filtre dans la page du rapport) :

- Effectivement, si on utilise la mesure « sommes des ventes » et « sommes des ventes année précédente », il nous faut utiliser un filtre sur l'année en cours sur le visuel, sur la page ou dans un segment.
- Dans les cas où nous ne pouvons ni utiliser de segment car le client final n'en veut pas et/ou nous ne pouvons pas utiliser de filtre car il y a plusieurs indicateurs avec filtres différents sur le même visuel, je n'ai d'autre choix que d'appliquer cette méthode.

✓ **Année en cours sans filtre** = `CALCULATE([Somme montants ventes]; year('Date'[Cledate])= Year(Today()))` - *quand l'année de la date principale est égale à l'année de la date du jour (today)*

✓ **Année précédente sans filtre** = `CALCULATE([Somme montants ventes]; year('Date'[Cledate])= Year(Today())-1)`



Travailler avec plusieurs  
dates et plusieurs relations  
avec USERELATIONSHIP

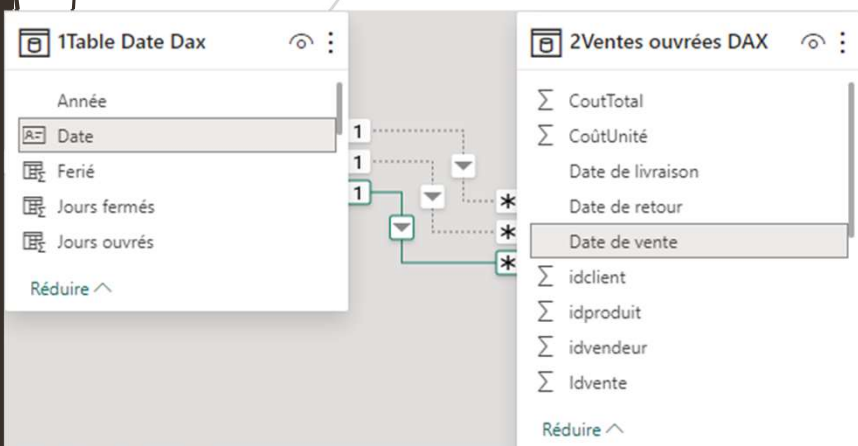
# Les multiples relations de dates

## Fonction USERELATIONSHIP (1/3)

- **Une relation active** est une relation qui n'a pas besoin d'être redéfinie ou spécifiée dans une mesure. *Elle est représentée par un trait fin.*
  - ✓ Il faut imaginer qu'elle représente un lien entre les deux tables comme si sur Excel, on avait fait une RECHERCHEV et **rassemblé toutes les colonnes des deux tables dans une seule** et même table. Il nous suffit de piocher dedans et tout est bien calculé.
  - ✓ Ce type de relation est **toujours « effective »**, pris en compte dans chaque manipulation, dans chaque visuel de nos rapports.
- ❖ **Il ne peut y en avoir qu'une** donc si nous avons besoin de calculer le montant des ventes à date de commande mais aussi à date de livraison, il est nécessaire de créer ce qu'on appelle **des relations inactives pour lier les autres dates à la table de date.**
- **Une relation inactive** est une relation qui a besoin d'être spécifiée dans une mesure/indicateur pour être activée dans la mesure /l'indicateur correspondant. *Elle est représentée par des tirets.*

# Les multiples relations de dates

## Fonction USERELATIONSHIP (2/3)



- Dans notre exemple, j'ai **3 dates** : **une de vente, une de livraison et une de retour.**
- Les questions à se poser sont « **Quelle est la date que je vais le plus utilisée ?** » et/ou « **La grande majorité de mes indicateurs seront basés et calculés sur quelle date ?** »
- ✓ La **relation active** étant toujours effective autant la placer sur la date la plus utilisée. Pour nous, ce sera la date de vente.
- ✓ Les deux autres dates seront positionnées sur **des relations inactives** et il sera **nécessaire de créer des mesures pour pouvoir les utiliser.**
- ✓ Réaliser les relations ensemble pour faire l'exemple suivant.

# Les multiples relations de dates

## Fonction USERELATIONSHIP (3/3)

Année	Somme de MontantVentes	Somme montants ventes dateLIV
2022	2 620 676,25	2 643 295,25
2023	2 877 253,21	2 889 394,40
Trimestre 1	490 231,80	478 739,58
Trimestre 2	836 181,66	825 381,28
Trimestre 3	911 654,28	921 816,00
Trimestre 4	639 185,47	663 457,54
<b>Total</b>	<b>5 497 929,46</b>	<b>5 532 689,66</b>

➤ **Relation active** : Pour calculer « la somme des montants des ventes » en fonction de la date de vente, je peux tout simplement utiliser la colonne « Montant ventes » de la table des ventes et le visuel calculera une somme par défaut car en lien avec une relation active.

➤ **Relation inactive** :

➤ Pour calculer « la somme des montants des ventes » en fonction **de la date de livraison**, je passe par une **relation inactive**, je dois donc **créer une mesure** qui va activer la relation pour elle-même :

```
=CALCULATE(SUM('Ventes'[MontantVentes]);
USERELATIONSHIP('Ventes'[Date de livraison];'Date'[Cledate] ))
```

■ Je calcule la somme des montants des ventes,

■ Mais en utilisant la relation inactive (userelationship) entre la colonne « date de livraison » de la table ventes et la colonne « Date » de la table de Dates.

# Concevoir un modèle adéquat

## Exercices (en plus – si le temps)

Année	Somme Montant Coûts	Somme Montant Coûts Date Livraison	Moyenne Quantités Retournées DateRetour
2022	1 057 677,13	1 064 854,59	2,29
2023	1 174 222,11	1 179 301,52	2,16
janvier	64 828,40	66 490,75	2,22
février	57 972,28	57 747,18	2,03
mars	81 040,15	75 311,82	2,19
avril	124 861,41	115 946,26	1,99
mai	105 844,02	117 449,69	2,15
juin	134 415,28	126 235,26	2,30
juillet	133 682,87	143 822,16	2,27
août	95 065,93	92 959,01	2,15
septembre	119 164,73	116 560,47	2,17
octobre	106 454,23	114 584,87	2,12
novembre	69 645,65	56 203,09	2,25
décembre	81 247,16	95 990,96	1,98
2024	101 144,67	91 275,75	1,49

➤ **Créer les mesures suivantes** dans **une matrice** contenant **une hiérarchie Année/Trimestre/Mois/Jour en ligne** :

- 1) Créer la somme des montants des coûts à date de livraison,
- 2) Créer la moyenne des quantités retournées à date de retour,



# Concevoir un modèle adéquat

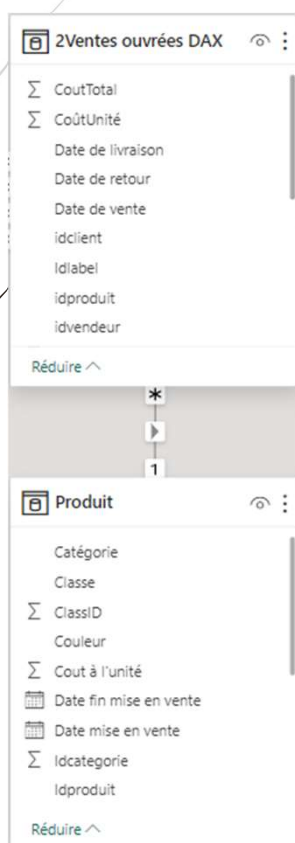
## Exercices avec corrections

Année	Somme Montant Coûts	Somme Montant Coûts Date Livraison	Moyenne Quantités Retournées DateRetour
2022	1 057 677,13	1 064 854,59	2,29
2023	1 174 222,11	1 179 301,52	2,16
janvier	64 828,40	66 490,75	2,22
février	57 972,28	57 747,18	2,03
mars	81 040,15	75 311,82	2,19
avril	124 861,41	115 946,26	1,99
mai	105 844,02	117 449,69	2,15
juin	134 415,28	126 235,26	2,30
juillet	133 682,87	143 822,16	2,27
août	95 065,93	92 959,01	2,15
septembre	119 164,73	116 560,47	2,17
octobre	106 454,23	114 584,87	2,12
novembre	69 645,65	56 203,09	2,25
décembre	81 247,16	95 990,96	1,98
2024	101 144,67	91 275,75	1,49

- ✓ Somme Montant Coûts Date Livraison = `calculate(SUM('Ventes'[CoutTotal]) ; USERRELATIONSHIP('Ventes'[Date de livraison]; 'Date'[Cledate] ))`
- ✓ Moyenne Quantités Retournées DateRetour = `CALCULATE(AVERAGE('Ventes'[Quantitésretournées]); USERRELATIONSHIP('Ventes'[Date de retour]; 'Date'[Cledate]))`

# Les multiples relations de dates

## Fonction CROSSFILTER (1/2) (Perf +)



- La **fonction CROSSFILTER** permet de **modifier le filtre entre deux tables** juste pour le calcul en question.
- Dans le modèle suivant, la table *Produit* et la table *2Ventes Ouvrées DAX* présentent une relation de direction unique: cardinalité plusieurs vers 1.
- Les **agrégations sont donc tenues d'être calculées** sur des champs de la table sur laquelle est accolée la relation \* donc N (**la table de faits** au centre)
- Si l'on souhaite calculer et la somme des ventes et le nombre de produits, il va être **nécessaire d'activer un filtre à double sens mais uniquement pour ce calcul.**

# Les multiples relations de dates

## Fonction CROSSFILTER (2/2) (Perf +)

Année	Somme montants ventes	Nombre de Idproduit - Colonne normale	Nombre de produits CROSSFILTER
2017	1 172 134,36	120	104
2018	3 964 184,67	120	118
2019	2 737 707,80	120	118
2020	2 938 696,92	120	118
2021	2 613 513,61	120	116
2022	2 620 676,25	120	115
2023	2 877 253,21	120	117
2024	249 214,89	120	95
<b>Total</b>	<b>19 173 381,71</b>	<b>120</b>	<b>120</b>

- Je crée une **matrice avec en ligne la hiérarchie temporelle** et en valeurs la somme des montants des ventes et le nombre de produits. Le second résultat ne fonctionne pas car la relation entre la table de *Ventes* et la table *Produit* est unique.
- Je crée une mesure avec **USERELATIONSHIP** pour avoir le **bon calcul** :  
 Nombre de produits CROSSFILTER =  

```

CALCULATE(DISTINCTCOUNT(Produit[Idproduit]);
CROSSFILTER( Produit[Idproduit];
'Ventes'[idproduit];
Both)) -- Filtre des deux côtés 😊
    
```

# Exploiter les fonctions DAX avancées

132

# Exploiter les fonctions DAX avancées

- La fonction CALCULATE VS SUMX / AVERAGEX etc + FILTER.
- Le calcul de cumul avec la fonction EARLIER.
- La fonction de gestion des valeurs vides : ISBLANK.
- Les fonctions DAX de création de tables.
- Les fonctions de filtres : SELECTEDVALUE, ISFILTERED et ISINSCOPE.

➤ La fonction CALCULATE VS  
Sumx/Averagex etc + FILTER.

# Fonction CALCULATE

## Rappel : Filtres et gestion des ET et OU

- **Le CALCULATE** permet de faire une agrégation en fonction d'un ou plusieurs filtres.
- **Syntaxe**: `Calculate( Expression = calcul entier ou mesure ; filtre1 fin ou ; filtre 2 ; etc.)`
- Exemple avec plusieurs filtres : La somme des montants des ventes pour BAC+3 et Mariés

```
= CALCULATE([Somme montants ventes]; 'Vendeurs'[Niveau d'étude]= "Bac +3";
'Vendeurs'[Situation familiale] ="Marié" )
= CALCULATE([Somme montants ventes]; 'Vendeurs'[Niveau d'étude]= "Bac +3" &&
'Vendeurs'[Situation familiale] ="Marié" )
```
- Exemple avec le OU : Mariée ou Célibataire :
  - ✓ Je souhaite calculer la somme des montant des ventes réalisées par des personnes mariées ou célibataires.

```
CALCULATE( [Somme montants ventes] ;
'Vendeurs'[Situation familiale] IN {"Mariée";"Célibataire"} ) - IN : liste de valeurs
```

```
VS 'Vendeurs'[Situation familiale]="Mariée" || 'Vendeurs'[Situation familiale]="Célibataire") -- OU
```

## Fonctions DAX avec filtre en ligne SUMX / AVERAGEX / MINX, etc. = MESURE

- **SUMX** : Il permet de faire un calcul (agrégation) en fonction d'un critère/expression en ligne
- **Deux étapes :**
  - ✓ **Le X** : Pour calculer une **expression** : addition, multiplication, etc. entre plusieurs colonnes, **ligne par ligne**, sans créer de colonne dans la table.
  - ✓ **Le SUM** : Pour calculer la **somme totale** de cette **colonne temporaire**/en mémoire.
- **Exemple :**
  - ✓ Il y a les colonnes MontantVentes et CoutTotal dans la table des ventes.
  - ✓ Je souhaite créer la mesure **profit**, qui est la **somme de toutes les soustractions**, ligne par ligne **entre le cout et le montant**.
  - ✓ *Créer une colonne dans une table de ventes/transactionnelle peut être très lourd en termes de stockage et de performance. Priorisez-le SUMX/COUNTX/ etc.*

```
Profit SUMX = SUMX('Ventes'; -- sur quelle table  
'Ventes'[MontantVentes]-'Ventes'[CoutTotal])  
-- expression/calcul ligne par ligne
```



# Fonctions DAX avec filtre en ligne

## SUMX / AVERAGEX / MINX, etc. = COLONNE

### ➤ **SUMX en colonne :**

- Il permet de rapporter une somme calculée sur une autre table = même résultat sur chaque ligne,
- Mais également, de rapporter une somme calculée sur une autre table alors qu'un **FILTRE a été ajouté** = résultat différent sur chaque ligne en fonction de la colonne de correspondance / relation active.

### Exemple :

- ✓ Dans la table vendeurs, j'ai une ligne par vendeur différent,
- ✓ Je souhaite calculer le total de la somme des ventes pour chaque vendeur,
- ✓ Chaque ligne aura donc un résultat différent : un filtre sera appliqué sur la table vente pour faire la somme sur le vendeur correspondant avant de le renvoyer dans la table vendeurs

= SUMX

( RELATEDTABLE ( 'Ventes' ) ; -- spécification de la table MAIS également que pour faire le calcul, il doit utiliser la relation de table pour filtrer la table Ventes avant de faire le calcul

'Ventes'[MontantVentes] ) -- expression/calcul ligne par ligne / on fera la somme de cette colonne

# Fonctions DAX avec filtre en ligne

## SUMX / AVERAGEX / MINX avec FILTER

### ➤ **SUMX avec des filtres:**

- Avant d'utiliser un filtre, il sera nécessaire d'utiliser la fonction **FILTER** car n'existe pas naturellement dans la fonction comme pourrait l'être pour le Calculate,
- Il sera nécessaire d'utiliser le && pour mettre plusieurs filtres inclusifs (**ET**),
- Il sera nécessaire d'utiliser le IN pour la liste de valeurs ou || pour mettre plusieurs filtres exclusifs (**OU**).

Reprises des exemples du Calculate :

```
Bac+3 ET marié = SUMX(  
  FILTER('Vendeurs';  
    'Vendeurs'[Niveau d'étude]= "Bac +3" && 'Vendeurs'[Situation familiale] ="Marié");  
  -- Deux filtres sur colonnes différentes donc &&/Et  
  [Somme montants ventes])
```

```
Mariée OU Célibataire = SUMX(  
  FILTER('Vendeurs';  
    'Vendeurs'[Situation familiale]="Mariée" || 'Vendeurs'[Situation familiale]="Célibataire");  
  [Somme montants ventes]) -- Deux filtres sur même colonne donc IN ou ||
```

# Fonctions DAX avec filtre en ligne

## Perfectionnement du OU et performances

- **Le || peut être modifié par un IN afin d'éviter de tout réécrire :**

```
SUMX Mariée OU Célibataire IN = SUMX(  
  FILTER('Vendeurs';  
    'Vendeurs'[Situation familiale] IN {"Mariée" ; "Célibataire"});  
  [Somme montants ventes])  
-- Deux filtres sur même colonne donc IN  
-- Pas de répétition de colonne pour chaque valeur proposée)
```

- **Pour le Gain en performance et en écriture, utilisation du TREATAS :**

```
SUMX Mariée OU Célibataire IN TREATAS = SUMX(  
  TREATAS({"Mariée" ; "Célibataire"}; Vendeurs[Situation familiale]);  
  [Somme montants ventes])  
-- Au lieu d'utiliser le FILTER : utilisation du TREATAS  
-- Beaucoup moins à écrire et plus performant
```

# Fonction CALCULATE vs SUMX

## Exercices

2023

Somme montant des ventes Audio et Ecouteurs bluetooth	Somme montant des ventes Classes Deluxe et moyen
39,94K	2,51M
Somme montant des ventes Classes Deluxe et moyen SUMX	Profit Moyen
2,51M	901,07

➤ Créer les mesures suivantes dans **une carte à plusieurs lignes** :

- 1) Créer la somme des montants des ventes pour la catégorie « Audio » et la sous-catégorie « Ecouteurs Bluetooth » (*CALCULATE*),
- 2) Créer la somme des montants des ventes pour les classes De luxe et Moyen (*CALCULATE + IN*),
- 3) Créer la somme des montants des ventes pour les classes De luxe et Moyen (*SUMX et TREATAS*),
- 4) Calculer le profit moyen (*Agrégation en X : SUMX, AVERAGEX, etc.*),
- 5) Créer une nouvelle colonne dans la table vendeur qui va calculer la moyenne des remises pour chaque vendeur/chaque ligne (*SUMX + RELATEDTABLE*)

# Fonction CALCULATE vs SUMX

## Exercices avec corrections

- ✓ Somme montant des ventes Audio et Ecouteurs bluetooth  
= `CALCULATE( [Somme montants ventes] ; Produit[Catégorie]="Audio"; Produit[Sous catégories]="Ecouteurs bluetooth")`
- ✓ Somme montant des ventes Classes Deluxe et moyen  
= `CALCULATE( [Somme montants ventes] ; Produit[Classe] IN {"De luxe";"Moyen"})`
- ✓ Somme montant des ventes Classes Deluxe et moyen SUMX  
= `SUMX( TREATAS({"De luxe" ; "Moyen"};Produit[Classe]); [Somme montants ventes])`
- ✓ Profit Moyen  
= `AVERAGEX( '2Ventes ouvrées DAX'; '2Ventes ouvrées DAX'[MontantVentes]- '2Ventes ouvrées DAX'[CoutTotal])`
- ✓ Moyenne remise EXO  
= `AVERAGEX( RELATEDTABLE('2Ventes ouvrées DAX'); '2Ventes ouvrées DAX'[MontantRemises])`

# Le calcul de cumul avec la fonction EARLIER

# Les fonctions avancées itératives

## Fonction EARLIER (1/3)

date entrée	NB meme dates ligne
02/01/2017	2
02/01/2017	2
03/01/2017	1
04/01/2017	2
04/01/2017	2

- **EARLIER** signifie la **ligne courante**.
- Cette fonction va permettre de **comparer chaque ligne courante** avec **les autres lignes**.

### ➤ 1<sup>er</sup> exemple :

*Pour chaque ligne de la table vendeurs, on va compter le nombre de ligne ayant la même date que celle sur laquelle on est positionné, en incluant cette dernière dans le calcul.*

```
COUNTROWS(
    FILTER(Vendeurs; on filtre sur tous les vendeurs même
    si on est positionné que sur une seule ligne
    EARLIER(Vendeurs[Date d'entrée])= Vendeurs[Date d'entrée]
    )
)
```

# Les fonctions avancées itératives

## Fonction EARLIER (2/3)

➤ Avec la fonction **EARLIER**, je peux également créer un rang.

➤ 2<sup>nd</sup> exemple :

*Pour chaque ligne de la table vendeurs, on a rapporté la somme du montant des ventes . Je souhaite maintenant créer une colonne qui me ramène le numéro de rang du vendeur en fonction du montant des ventes réalisé.*

```
COUNTROWS(
  FILTER(Vendeurs;
    Vendeurs[SUMX colonne]<= EARLIER(Vendeurs[SUMX colonne])
    -- nb de lignes quand le MontantVentes est inférieur à la valeur de la ligne en
    cours en incluant cette dernière (=)
  )
)
```

SUMX colonne ↕	Rang Vendeur par MontantVentes/sumx colonne ▾
3 630 631,68	1
3 676 920,97	2
3 872 634,02	3
3 972 735,46	4
4 020 459,60	5



# Les fonctions avancées itératives

## Fonction EARLIER (3/3)

➤ **Avec la fonction EARLIER**, je peux également créer un cumul sur le rang créé.

➤ 3ème exemple :

*Pour chaque ligne de la table vendeurs, on a créé le rang des vendeurs. Je souhaite maintenant créer une colonne qui calcule le cumulatif des montants des ventes en fonction du rang du vendeur.*

Cumul\_rang Vendeurs =

```
CALCULATE( SUM(Vendeurs[SUMX colonne]);
```

```
  FILTER(Vendeurs; -- FILTER obligatoire ici, sinon le filtre naturel du CALCULATE
    ne fonctionne pas pour un calcul en ligne (colonne) mais seulement pour les mesures
```

```
      Vendeurs[SUMX colonne]<= EARLIER(Vendeurs[SUMX colonne])
```

```
  -- somme des valeurs plus petites et incluant la ligne en cours
```

```
  ))
```

Rang Vendeur par MontantVentes/sumx colonne	Cumul rang Vendeurs
1	3 630 631,68
2	7 307 552,64
3	11 180 186,66
4	15 152 922,12
5	19 173 381,72

# Les fonctions avancées VS Ranks

➤ **Avec la fonction Ranks**, on peut également créer un classement/rang.

➤ Exemple :

*Pour chaque ligne de la table vendeurs, on a rapporté la somme du montant des ventes . Je souhaite maintenant créer une colonne qui me ramène le numéro de rang du vendeur en fonction du montant des ventes réalisé.*

Rang Vendeur RANKS =

```
RANKX('Vendeurs'; -- sur quelle table on fait le rang
      'Vendeurs'[SUMX colonne];; -- sur quelle colonne/valeur
      asc; Dense) - tri ascendant ou descendant // + dense (Ou SKIP), si même
valeur même numéro de rang (opposé de SKIP par défaut pour EARLIER)
-- somme des valeurs plus petites et incluant la ligne en cours
))
```

Rang Vendeur par MontantVentes/sumx colonne	Cumul rang Vendeurs
1	3630632
2	7307553
3	11180187
4	15152922
5	19173382

# La fonction de gestion des valeurs vides : ISBLANK

# Fonctions avancées

## Précautions : ISBLANK et autres

➤ **Condition sur une seule colonne avant de faire un calcul :**

= IF( **ISNUMBER**( Sales[Price] ); Sales[Quantity] \* Sales[Price], BLANK() )

-- s'il y a un numéro renseigné dans la colonne PRICE alors je fais ma multiplication sinon je ne fais rien pour ces lignes.

= IF( **ISBLANK**( Sales[Price] ) ; BLANK() ; Sales[Quantity] \* Sales[Price] )

-- Si la colonne PRICE n'est pas renseignée (vide/Null) alors je ne fais rien, sinon je fais ma multiplication.

**Condition sur les deux colonnes avant de faire un calcul :**

= IF( ISBLANK( Sales[Price] ) || ISBLANK Sales[Quantity] ; BLANK() ; Sales[Quantity] \* Sales[Price] )

-- Si la colonne Price ou ( || ) la colonne Quantity n'est pas renseignée, alors je ne fais rien, sinon je fais le calcul.

= IF( **ISERROR**( Sales[Quantity] \* Sales[Price] ); BLANK(); Sales[Quantity] \* Sales[Price] )

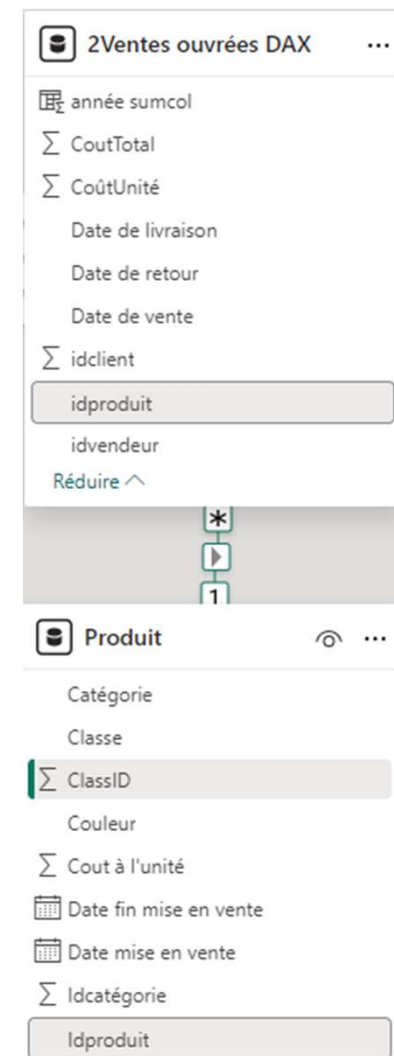
-- S'il y a une erreur dans le calcul entre les deux colonnes, je ne fais rien, sinon je fais le calcul.

# Les fonctions avancées

## Fonction RELATED (Perf +)

- La **fonction RELATED** correspond au RECHERCHEV dans EXCEL.
- Elle permet de **recupérer des colonnes** dans d'autres tables en fonction **d'une colonne de correspondance**.
- Dans Power BI, elle utilise **la relation active** créée dans le modèle par défaut pour récupérer des colonnes. Elle ne peut récupérer qu'une valeur donc la relation doit avoir une **cardinalité de 1** (dans la table où on souhaite récupérer la colonne).
- Elle est donc très simple, il suffit d'écrire **la fonction** puis entre parenthèses **le champ** que l'on souhaite **recupérer**.
- ✓ Dans la table des ventes, je peux récupérer la classe du produit en utilisant la relation active (Idproduit). Je crée une **nouvelle colonne** :

Récupération classe du produit = `RELATED(Produit[Classe])`



# Les fonctions avancées

## Fonction LOOKUPVALUE (1/2) (Perf +)

- La **fonction LOOKUPVALUE** correspond au RECHERCHEV dans EXCEL mais elle permet d'utiliser **plusieurs colonnes** pour faire la **correspondance** et **recupérer des colonnes**.
- Ici, il est impératif de **préciser les colonnes de correspondances** donc de détailler la formule.
- ✓ Chargement **des tables VLUP\_TAB1 et VLUP\_TAB2** dans le fichier Excel « Matrice de Bus » :
- ✓ Dans la table **VLUP\_TAB2**, je souhaite **recupérer le prix** en utilisant les colonnes Mois et Item en correspondances avec VLUP\_TAB1.

Je crée donc une **nouvelle colonne** :

Récupération PRIX

= LOOKUPVALUE( VLUP\_TAB1[Prix];

-- *Colonne à récupérer = prix*

VLUP\_TAB1[Mois];VLUP\_TAB2[Mois];

-- *1<sup>ère</sup> colonne de correspondance dans les deux tables = mois*

VLUP\_TAB1[Item];VLUP\_TAB2[Item]) -- *2<sup>nde</sup> colonne de correspondance dans les deux tables = item*

Mois	Item	Récupération PRIX
Janvier	Vélo	100
Janvier	Skate	130
Février	Vélo	110
Février	Skate	120

# Les fonctions DAX de création de tables

# Les fonctions de création de tables SUMMARIZECOLUMNS (1/2)

- **SUMMARIZECOLUMNS** permet de créer **une table** qu'on va utiliser ensuite de la même manière qu'une table du modèle de données et permet de réaliser **des regroupements et agrégations** (comme la fonctionnalité « Regrouper » dans Power Query »).
- On ne peut pas l'utiliser **directement pour une mesure**. *Sumx (summarizecolumns) impossible*
- Intéressant si on n'a **pas la main sur Power Query**
- ET pour **utiliser sur certains visuels** qui demandent une **préparation** spécifique des données en amont.

- **La syntaxe générale :**

```
SUMMARIZECOLUMNS(  
  <groupBy_columnName> ; -- regroupement 1 / spécification de la colonne  
  <groupBy_columnName> ; -- regroupement 2 / etc...  
  <filterTable>; -- le filtre se fera sur les colonnes à regrouper en amont du calcul  
  <name>; -- nom de la nouvelle colonne correspondant au calcul suivant (expression)  
  <expression> ; -- calcul permettant de créer cette colonne  
  <name>; -- nom de la seconde nouvelle colonne  
  <expression> ) -- calcul permettant de créer cette seconde colonne
```





## Les fonctions de création de tables CALCULATE TABLE avec SummarizeColumns

- **CALCULATETABLE** permet de créer des tables physiques et/ou **virtuelles** que l'on peut filtrer en utilisant plusieurs **conditions/filtres** et d'utiliser cette table pour effectuer d'autres calculs.
- Si on souhaite regrouper au préalable, il faudra utiliser un SUMMARIZECOLUMNS à l'intérieur du CALCULATETABLE.

- Même exemple que pour SUMMARIZE mais le filtre dans CALCULATETABLE:

```
CALCULATETABLE( --premier argument « ce que j'affiche » (colonnes)
SUMMARIZECOLUMNS(Produit[Catégorie];
                  'Date'[Année];
                  "Montant ventes sumcol"; [Somme montants ventes]);
TREATAS({"Audio";"TV et Video"}; Produit[Catégorie])
)
```

-- je filtre sur les catégories « audio et TV/Video » dans le CALCULATETABLE

# Les fonctions de création de tables

## UNION : ajout de lignes

ID PROD	VALEUR
3	30
1	50
4	10
2	20

➤ **UNION** permet de **créer une nouvelle table** qui **associe les lignes d'une table avec celles d'une autre** à condition qu'elles aient les mêmes colonnes (même nombre, mêmes noms et dans le même ordre).

➤ Permet de faire comme « **Ajouter des colonnes** » dans Power Query

➤ Exemple:

*Pour faire l'exemple, charger dans le fichier Excel « Données démo simple », les tables « prod1 » et « prod2 ».*

UNION DES PROD = UNION(PROD1;PROD2)

- Possibilité de mettre autant de tables que souhaité
- Intéressant si on n'a pas la main sur Power Query

# Les fonctions de filtres : SELECTEDVALUE, HASONEFILTER et ISFILTERED

# Les fonctions avancées de filtrage SELECTEDVALUE pour les Titres dynamiques

- **SELECTEDVALUE** permet de créer des mesures pour gérer des **titres** et sous-titres **dynamiques**.

- Exemple :

*J'ai un segment sur la classe de produit. On souhaite que le sous-titre du visuel s'actualise en fonction de la classe sélectionnée. Je crée dans une mesure = « Titre dynamique classe »*

Titre dynamique CLASSE = `SELECTEDVALUE(Produit[Classe]; "Plusieurs classes sélectionnées")`

*Affiche la valeur sélectionnée dans le segment CLASSE. Si aucune valeur ou plusieurs valeurs sélectionnées alors affiche « Plusieurs classes sélectionnées »*

- Aller dans la fenêtre « Visualisation » > « Mise en forme votre visuel » > « Titre » > « Sous-titre » > Texte = Cliquer sur « FX » > « Sur quel champ devons-nous nous baser » : Sélectionner la mesure créée. Le titre s'actualise en fonction du segment.

# Les fonctions avancées de filtrage CONCATENEX et multi sélection

- **SELECTEDVALUE** est très pratique pour notre **titre dynamique** cependant, si je sélectionne deux classes, j'aurais « Plusieurs classes sélectionnées » qui s'affichent et non les noms des classes sélectionnées en question. Pour rectifié, il faut utiliser **CONCATENEX**.

- Exemple :

*Je souhaite que mon titre affiche Economique et De Luxe si je sélectionne ses classes spécifiques.*

Titre dynamique CLASSE multi sélection =

```
CONCATENATEX ( VALUES ( Produit[Classe] ) ; Produit[Classe] ; " & "
```

*Je concatène toutes les valeurs de la colonne « Classe » = valeurs uniques car utilisation de la fonction VALUES (et non pas table complète)*

*Je ne souhaite pas concaténer toutes les valeurs «Moyen » répétées plusieurs fois dans la même colonne !*

*Et j'affiche le nom des classes sélectionnées séparées par un espace + le & et un autre espace.*

# Les fonctions avancées de filtrage

## ISFILTERED : calcul différent si utilisation filtre

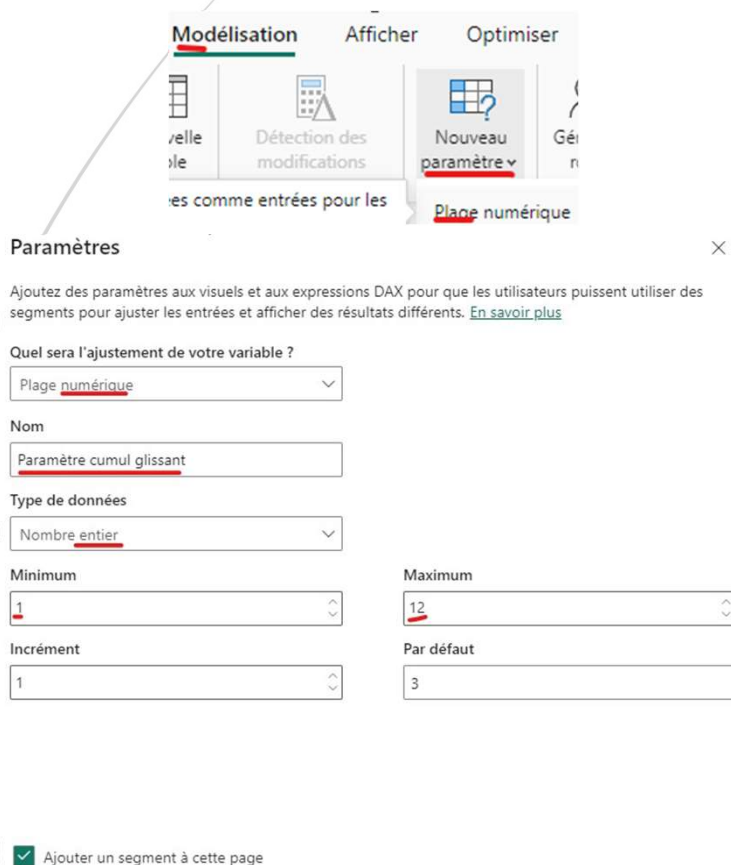
- **ISFILTERED** permet de mettre des conditions selon ce qui est filtré ou non, pour gérer toutes les possibilités d'un titre dynamique par exemple.
- Même Exemple que dans la diapositive précédente :
- ✓ *Si les 3 classes de produit sont sélectionnées, je ne veux pas les concaténer mais afficher « Toutes classes confondues » comme pour le cas, ou rien n'est sélectionné.*

```
VAR VCount = COUNTROWS( VALUES (Produit[Classe]) )
                - compte Le nombre de valeurs sélectionnées dans Le segment
VAR VConcat = CONCATENATEX ( VALUES ( Produit[Classe] ) ; Produit[Classe] ; " & ")

RETURN        -- affichage final de La mesure
    IF( ISFILTERED( Produit[Classe] ) && VCount < 3 ; Vconcat ; "Toutes classes
    confondues" ) - si Le segment classe est utilisé et que Le nombre de valeurs
    sélectionnées est inférieur à 3, alors faire la concaténation, sinon mettre «toutes
    classes confondues »
```

# Les fonctions avancées de filtrage

## Paramétrage : choix nombre mois glissants



**Modélisation** | Afficher | Optimiser

Nouvelle table | Détection des modifications | **Nouveau paramètre** | Général

Paramètres

Ajoutez des paramètres aux visuels et aux expressions DAX pour que les utilisateurs puissent utiliser des segments pour ajuster les entrées et afficher des résultats différents. [En savoir plus](#)

Quel sera l'ajustement de votre variable ?  
 Plage numérique

Nom  
 Paramètre cumul glissant

Type de données  
 Nombre entier

Minimum  
 1

Maximum  
 12

Incrément  
 1

Par défaut  
 3

Ajouter un segment à cette page

➤ **Les paramètres de plages numériques** permettent de créer des **segments** qui vont pouvoir être **intégrés** dans des **titres dynamiques et des mesures** pour agir sur les visuels directement.

➤ Exemple :

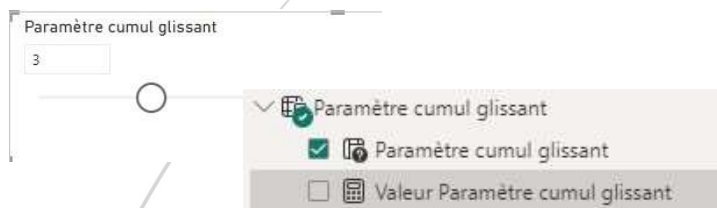
*On a créé une mesure pour calculer des cumuls de vente sur 12 mois glissants. Imaginons que l'utilisateur final souhaite changer le nombre de mois de cumul ?*

- 1) Créer un **paramètre de plage numérique (onglet modélisation)**
- 2) **Paramétrage** : Lui donner un nom, mettre 1 à 12 pour les 12 mois de l'année. Un incrément de 1 pour avoir les 12 valeurs et une valeur par défaut (3 par exemple).



# Les fonctions avancées de filtrage

## Paramétrage : choix nombre mois glissants



```
DatesBetween -12 mois =
CALCULATE( [Somme montants ventes];
DATESINPERIOD('Date'[Cledate] ;
LASTDATE('Date'[Cledate] );
- [Valeur Paramètre cumul glissant] ;
MONTH))
```

➤ **Conséquence** : cela crée une table avec le paramètre (itération) et la valeur à utiliser dans une formule (calcul du cumul)

3) Dans le code permettant de créer les 12 mois glissants, remplacer le « -12 » par « - ([Valeur Paramètre cumul]) »

**Conséquence** : la mesure se calculera en fonction du numéro sélectionné par l'utilisateur final dans le segment.

4) Pour le titre du visuel, cliquer sur FX, mais au lieu de s'embêter avec SELECTEDVALUE, mettre uniquement la valeur de paramètre plus les mots du titre souhaités :  
= [Valeur Paramètre cumul] & " mois glissants«

Si je clique sur 6 mois, le titre devient : **6 mois glissants**

# Les fonctions avancées de filtrage

## Paramétrage : choix colonnes via segment

**Paramètres**

Ajoutez des paramètres aux visuels et aux expressions Delle segments pour ajuster les entrées et afficher des résultats

Quel sera l'ajustement de votre variable ?

Champs

Nom

Choix colonnes

Ajouter et réorganiser des champs

Classe	X
Marque	X
Catégorie	X
Sous catégories	X
Nom du produit	X
Couleur	X
Type de Stock	X

Ajouter un segment à cette page

**Modélisation** Afficher Optimiser

Détection des modifications Actualisation de la page

Nouveau paramètre

Plage numérique

**Champs**

Rechercher

- $\Sigma$  ClassID
- Couleur
- $\Sigma$  Cout à l'unité
- >   $\Sigma$  Date fin mise en vente
- >   $\Sigma$  Date mise en vente
- $\Sigma$  Idcatégorie
- Idproduit
- $\Sigma$  Idsouscatégorie
- Marque
- $\Sigma$  Nb stocks
- Nom du produit
- $\Sigma$  Poids

➤ **Les paramètres de champ** permettent de laisser la main libre à l'utilisateur final sur les **colonnes** qu'ils souhaitent **afficher dans ses visuels**.

➤ Exemple :

*Une équipe souhaite avoir un tableau complet mais tous ne souhaitent pas afficher les mêmes colonnes.*

1) Créer un **paramètre de champ**

**2) Paramétrage :** Lui donner un nom.

Sélectionner les champs que les utilisateurs veulent utiliser/choisir pour leur visuel et les organiser et les ranger selon l'ordre voulu.

Cliquer sur CREER

# Les fonctions avancées de filtrage

## Paramétrage : choix colonnes via segment

```
Choix colonnes = {
  ("Classe"; NAMEOF('Produit'[Classe])); 0);
  ("Marque"; NAMEOF('Produit'[Marque])); 1);
  ("Catégorie"; NAMEOF('Produit'[Catégorie])); 2);
  ("Sous catégories"; NAMEOF('Produit'[Sous catégories])); 3);
  ("Nom du produit"; NAMEOF('Produit'[Nom du produit])); 4);
  ("Couleur"; NAMEOF('Produit'[Couleur])); 5);
  ("Type de Stock"; NAMEOF('Produit'[Type de Stock])); 6)
}
```

Choix colonnes

- Classe
- Marque
- Catégorie
- Sous catégories
- Nom du produit
- Couleur
- Type de Stock

Classe	Marque
De luxe	A. Datum
De luxe	Adventure Works
De luxe	Contoso
De luxe	Fabrikam
De luxe	Litware
De luxe	Northwind Traders
De luxe	Proseware
De luxe	Southridge Video
De luxe	Tailspin Toys
De luxe	The Phone Company

- **Conséquence** : cela crée une **table** avec la formule correspondante et le **segment associé**.
- 1) Je crée un tableau simple et je glisse « **Choix colonnes** » à l'intérieur.
- ✓ Si aucun champ n'est sélectionné dans le segment, par défaut, cela signifie qu'il affiche toutes les colonnes.
- ✓ Si un ou plusieurs champs sont sélectionnés, alors il affiche ceux spécifiés dans le segment.

## Les fonctions avancées de filtrage

### ISINSCOPE : calcul différent niveau du visuel

- **ISINSCOPE** permet de différencier le calcul selon le niveau de regroupement / de hiérarchie / filtre de la ligne de valeur par rapport à un autre niveau de hiérarchie et/ou un total. *Ce n'est plus en lien avec le segment mais les filtres lignes et colonnes du visuel :)*

Le HASONFILTER permet de faire la même chose avec moins de performance.

Exemple :

- Je fais une matrice avec les années en lignes, les mois en colonnes et le nombre de ventes en valeurs. Je rajoute un filtre pour exclure les années antérieures à 2020.
- Le total par mois, est le total sur toutes les années du tableau mais j'aimerais que la ligne de total affiche **la moyenne par mois !**

Année	janvier	février	mars	avril	mai	juin	juillet	août	septembre	octobre	novembre	décembre	Total
2020	157	148	168	150	156	135	181	153	138	171	148	186	1891
2021	136	143	125	138	141	143	163	140	168	154	126	157	1734
2022	135	131	164	144	154	151	127	64	266	257	106	128	1827
2023	132	87	129	195	166	203	187	154	179	173	126	159	1890
2024	130	60											190
<b>Total</b>	<b>690</b>	<b>569</b>	<b>586</b>	627	617	632	658	511	751	755	506	630	7532

## Les fonctions avancées de filtrage

### ISINSCOPE : calcul différent niveau du visuel

- Je vais donc lui dire que s'il est sur un filtre de ligne (colonne en ligne du tableau = Année) alors il calcule le total des ventes
- Sinon (il ne reste que la ligne total), il calcule la moyenne sur toutes les années depuis 2020.

Calcul # pour total =

```
var NBVentestotalannee = COUNT( 'Ventes'[Idvente] )
```

```
var moyenneSfiltre = COUNT( 'Ventes'[Idvente] ) /DISTINCTCOUNT( 'Date'[Année] )
```

```
return if (ISINSCOPE( 'Date'[Année])); NBVentestotalannee ; moyenneSfiltre )
```

Année	janvier	février	mars	avril	mai	juin	juillet	août	septembre	octobre	novembre	décembre
2020	157	148	168	150	156	135	181	153	138	171	148	186
2021	136	143	125	138	141	143	163	140	168	154	126	157
2022	135	131	164	144	154	151	127	64	266	257	106	128
2023	132	87	129	195	166	203	187	154	179	173	126	159
2024	130	60										
<b>Total</b>	<b>138</b>	<b>114</b>	<b>147</b>	157	154	158	165	128	188	189	127	158

# Les fonctions avancées de filtrage

## Synthèse des fonctions de filtre

- **SELECTEDVALUE** : pour générer un titre dynamique affichant la valeur sélectionnée (segment – fonctionne si plusieurs valeurs sélectionnées).
- **CONCATENEX** : pour générer un titre dynamique affichant les valeurs sélectionnées (segment – fonctionne si plusieurs valeurs sélectionnées).
- **ISFILTERED** : pour générer un titre dynamique en fonction de conditions pour l'affichage des valeurs sélectionnées (segment – fonctionne avec plusieurs valeurs).
- **PARAMETRES DE PLAGE NUMERIQUE** : permet de gérer via un segment le paramétrage d'une mesure.  
(ex : nombre de mois glissants)
- **PARAMETRE DE CHAMP** : permet de gérer via un segment les colonnes à afficher dans un visuel.
- **HASONFILTER** : pour calculer différemment selon valeur du segment sélectionnée (une seule possible) et/ou niveau de hiérarchie/regroupement/filtre de ligne du visuel lui-même.  
*Moins performant qu'ISINSCOPE.*
- **ISINSCOPE** : pour calculer différemment selon le niveau de hiérarchie/regroupement/filtre de ligne du visuel lui-même où est intégré la mesure. (ex : calcul du total différent des valeurs de classe de produits)

# Fonctions de filtre

## Exercices (1/2)

1. **Dupliquer la page de démonstration réalisée en cours** (*Montant ventes par classes – Selectedvalue titre dynamique*),
  - **Modifier le champ du segment : mettre les catégories de produit à la place des classes,**
  - **Modifier les titres dynamiques en créant les mesures nécessaires :**
    - a) Créer une mesure qui affiche la catégorie sélectionnée (si une seule sélectionnée). Si plusieurs catégories sélectionnées ou aucune : il faut qu'elle affiche « plusieurs catégories sélectionnées » dans le visuel « Carte » sous le segment et l'intituler « titre dynamique Catégorie » (*SELECTEDVALUE*),
    - b) Créer une mesure qui affiche la catégorie sélectionnée (si une seule sélectionnée). Si plusieurs catégories sélectionnées et/ou toutes : il faut qu'elle affiche les noms des catégories sélectionnées dans le visuel « Carte » sous le graphique et l'intituler « titre dynamique Catégorie multi sélection » (*CONCATENATEX + VALUES*),



# Fonctions de filtre

## Exercices (2/2)

- c) Créer une mesure qui affiche la catégorie sélectionnée (si une seule sélectionnée). Si plusieurs catégories sélectionnées mais pas toutes: il faut qu'elle affiche les noms des catégories sélectionnées. Si aucune et donc toutes les catégories sélectionnées : il faut qu'elle affiche «plusieurs catégories sélectionnées » dans le sous-titre du graphique et l'intituler « TitreDynamique catégorie ISFILTERED») – *s'aider de l'exemple fait ensemble en démonstration en lien avec diapositive 159,*

### 2) Dans une autre page, créer une matrice avec les années en lignes.

- Créer une mesure qui affiche « année » si on est sur une ligne des années, sinon « Total » si on est sur une ligne de total

Année	Calcul différent selon niveau
2020	année
2021	année
2022	année
2023	année
2024	année
<b>Total</b>	<b>total</b>



# Fonctions de filtre

## Exercices avec corrections

### 1. Mesures SELECTEDVALUE, CONCATENEX VALUES et ISFILTERED

✓ Titre dynamique Catégorie

= `SELECTEDVALUE(Produit[Catégorie]; "Plusieurs catégories sélectionnées")`

✓ Titre dynamique Catégorie multi sélection

= `CONCATENATEX( VALUES(Produit[Catégorie]); Produit[Catégorie]; " & ")`

✓ TitreDynamique catégorie ISFILTERED =

`VAR VCount = COUNTROWS( VALUES (Produit[Catégorie]) )`

`VAR VConcat = CONCATENATEX ( VALUES ( Produit[Catégorie] ) ; Produit[Catégorie] ; " & ")`

`RETURN`

`IF( ISFILTERED( Produit[Catégorie] ) && VCount < 3 ; Vconcat ; "Toutes catégories confondues" )`

### 2. Mesure ISINSCOPE

✓ Calcul différent selon niveau

✓ = `if (ISINSCOPE('1Table Date Dax'[Année] ) ; "année"; "total")`

# Affichage, navigation et visualisation

170

# Rappels : Synchroniser des segments entre les pages

The screenshot shows the Power BI Desktop interface with the 'Outils visuels' ribbon selected. The 'Synchroniser les segments' checkbox is checked. A pink arrow points from this checkbox to the 'SYNCHRONISER LES SEGMENTS' panel on the right. The panel includes a section for 'Ajouter et synchroniser avec toutes les pages, ou sélectionner des pages spécifiques :', a list of page names, and a section for 'Options avancées' with a checkbox for 'Synchroniser les changements de champ avec d'autres segments'.

**SYNCHRONISER LES SEGMENTS**

Ajouter et synchroniser avec toutes les pages, ou sélectionner des pages spécifiques :

Nom de la page

**Vue d'ensemble**

- Germany
- Canada
- France
- Great Britain
- Sales Amount tooltip
- Brand & Class tooltip
- Satisfaction Comparison
- Details Page
- Demo page for other features

Options avancées

Entrer un nom de groupe pour synchroniser la sélection avec d'autres visuels associés à

NSAT

Synchroniser les changements de champ avec d'autres segments

**CHAMPS**

**VISUALISATIONS**

**FILTRES**

PowerBI conseils

PAGE 1 SUR 10

# Rappels : Synchroniser des segments entre les pages

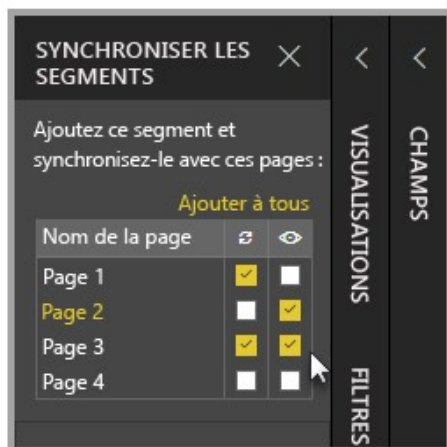
Dans le volet **Synchroniser les segments**, vous pouvez spécifier comment le segment doit être synchronisé entre les pages du rapport.

Vous pouvez spécifier si chaque segment doit être **appliqué** à chaque page du rapport, et si le segment doit être **visible** sur chacune de ces pages.

Par exemple, vous pouvez placer un segment sur la **page 2** de votre rapport, comme illustré dans l'image suivante.

Vous pouvez ensuite spécifier si ce segment doit *s'appliquer* à chaque page sélectionnée, et si ce segment doit être *visible* sur chaque page du rapport sélectionné.

Vous pouvez appliquer n'importe quelle combinaison de ces options, pour chaque segment. Cliquer sur **Ajouter à tous** à la fin.



# Rappels : Modifier les interactions

The screenshot shows the Microsoft Power BI interface. The ribbon is set to 'Outils pour les visuels' (Tools for Visuals), with the 'Format' tab selected. The 'Modifier les interactions' (Modify Interactions) button is highlighted in the ribbon. A tooltip for this button is displayed, containing the text: 'Modifier les interactions des éléments visuels. Changez la façon dont les visuels interagissent quand des points de données sont sélectionnés. En mode Édition, sélectionnez le visuel source et choisissez les comportements en utilisant les icônes qui apparaissent.' Below the ribbon, a 'Datekey' visual is shown with a date range from 27/12/2006 to 04/12/2009. A 'SalesAmount' visual is also shown, displaying '2,4bn€'. A red arrow points from the 'Modifier les interactions' button to the 'Datekey' visual, and another red arrow points from the tooltip to a filter icon on the 'Datekey' visual.

Tous les visuels d'une page sont interconnectés.

Si on sélectionne une valeur ou une partie d'un graphique sur le rapport, les autres éléments zooment sur cette partie.

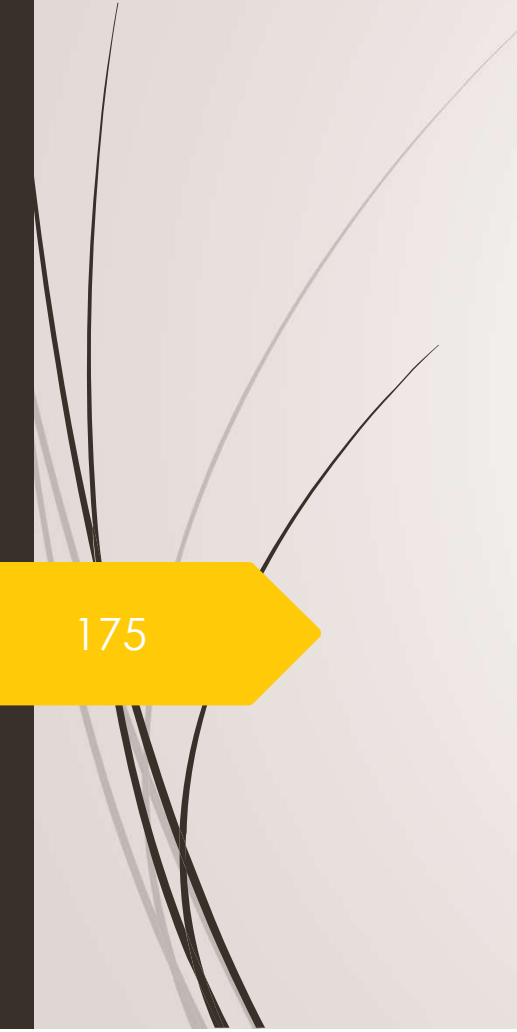
Si l'on souhaite qu'un visuel du rapport ne change pas en fonction des autres visuels et des segments, il faut le sélectionner, puis Cliquer sur Format dans l'onglet « outils pour les visuels ».

Puis cliquer sur « Modifier les interactions ».

Ensuite, des petites icônes apparaissent avec un rond barré à droite : si l'on clique dessus, celui-ci ne se filtrera plus en fonction des autres visuels ou filtres.

## Affichage, navigation et visualisation

- Créer un fond de page, un thème, un modèle de rapport.
- Créer des rapports "Maître-Détail" en utilisant les filtres d'extractions.
- Utiliser les signets pour la navigation et le Storytelling.
- Utiliser l'onglet des filtres et l'onglet sélection.
- Représenter avec des images. Préparer des commentaires dynamiques et personnaliser les infobulles.
- Définir un visuel personnalisé avec le Synoptic Panel

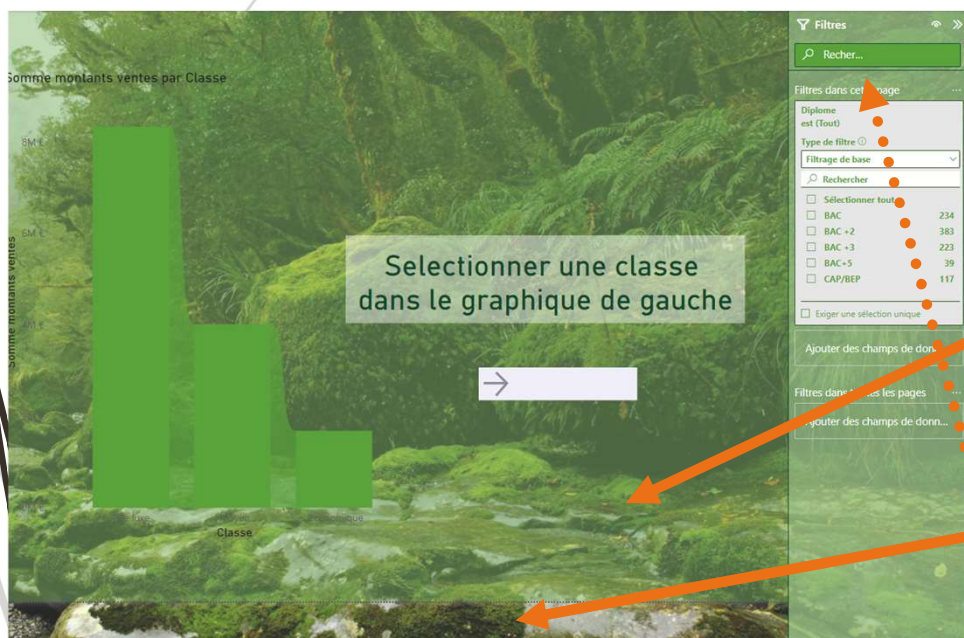


Créer un fond de  
page, un thème, un  
modèle de rapport



# Affichage, navigation et visualisation

## Créer un fond de page



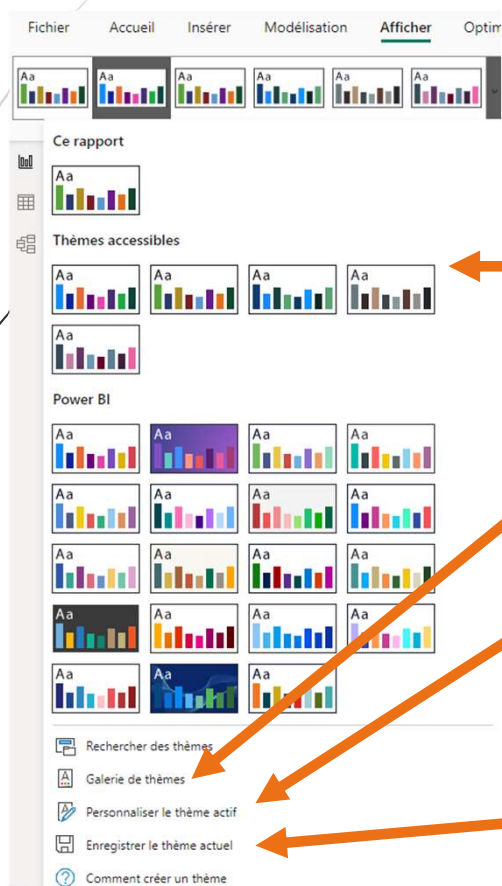
➤ **Le fond de page se crée au tout début** afin de pouvoir dupliquer la page et par conséquent, la mise en forme de celle-ci.

On peut gérer la mise en forme de divers éléments :

- ✓ **Paramètres du canevas** : taille et forme (paysage, portrait etc.),
- ✓ **L'arrière-plan de canevas** correspond à la zone de la page de rapport sur laquelle on peut positionner des visuels. Il est au **premier plan** derrière les graphiques,
- ✓ **Le papier peint** est **derrière le canevas** de la page et correspond à toute la page de rapport,
- ✓ **Le volet et les cartes de filtres** pour ajuster la mise en forme de la fenêtre de filtres si besoin.



# Affichage, navigation et visualisation Utilisation et création de thèmes

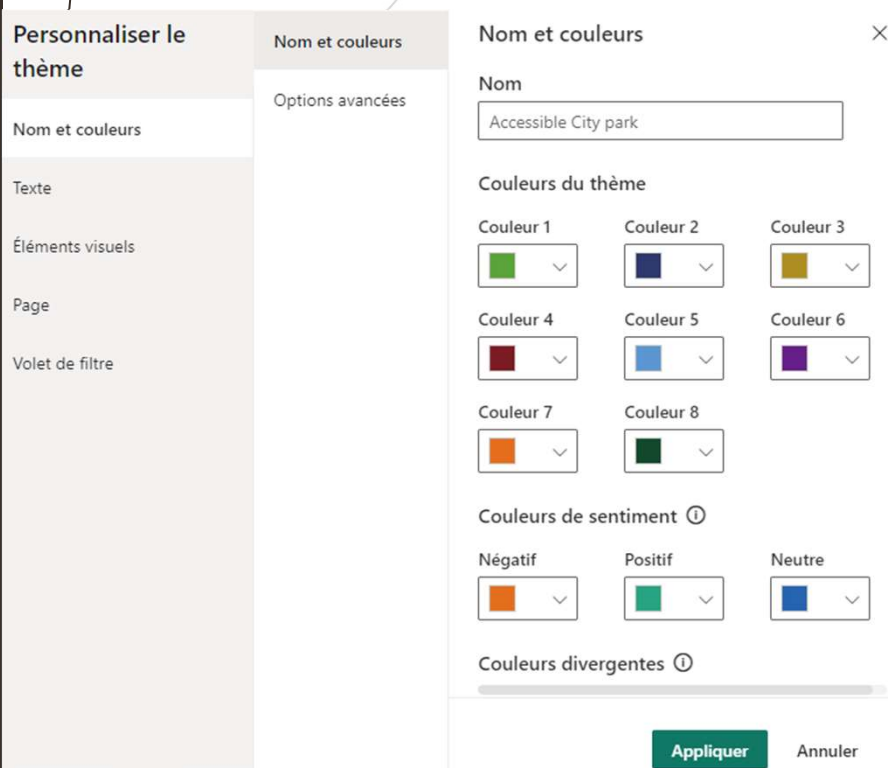


## ➤ Concernant les thèmes, voici les 3 options :

- ✓ **Les thèmes de base intégrés** que l'on peut directement sélectionner depuis Power BI Desktop,
  - ✓ **Les thèmes complexes préconçus** que l'on peut importer au format JSON depuis le site de Microsoft et modifiable au besoin sur Power BI Desktop,
  - ✓ **Les thèmes personnalisés à créer** que l'on construit pour notre utilisation dans notre entreprise.
- Pour accéder à ces possibilités, cliquer sur « Afficher » dans le ruban, puis la petite flèche à droite des premiers thèmes proposés.
- Pour enregistrer un thème d'entreprise : cliquer sur « Enregistrer le thème actif ».

# Affichage, navigation et visualisation

## Personnaliser un thème



- **1 Partir du thème par défaut de Power BI et/ou sélectionner un thème préconçu à améliorer.**
- **Cliquer sur « Personnaliser le thème actif »** et peuvent être modifié divers éléments d'un rapport :
  - ✓ Les couleurs : soit je modifie les couleurs du thème et les concepteurs iront mettre pour chaque visuel ce qu'ils préfèrent parmi la charte graphique spécifiée, soit je vais dans « options avancées » et j'indique pour chaque élément spécifique des visuels, sa couleur associée.
  - ✓ Le texte (général, titre etc.) et les éléments visuels (arrière-plan, bordure, etc.)
  - ✓ Page : arrière-plan et papier peint,
  - ✓ Volet de filtre : volet et carte de filtre.

# Affichage, navigation et visualisation

## Créer un modèle de rapport

### Exporter



- **Les modèles de rapport** permettent une uniformité/fichier type pour tous les rapports ayant besoin du même **jeu/modèle de données**.
- **Cela permet qu'ils aient tous la même conformité** : carte graphique, thème, indicateurs calculés de la même manière, etc.
- De plus, le fichier, un document .pbit est plus léger que le PBIX.
- Pour le créer, aller dans **FICHIER, Exporter** puis cliquer sur « **Modèle Power BI** ».
- Donner un **nom et un emplacement**.
- Vous pourrez ensuite le **partager** à tous ceux qui souhaitent travailler sur le même modèle de données et le même thème d'entreprise.

# Création fond et thème personnalisé

## Exercices

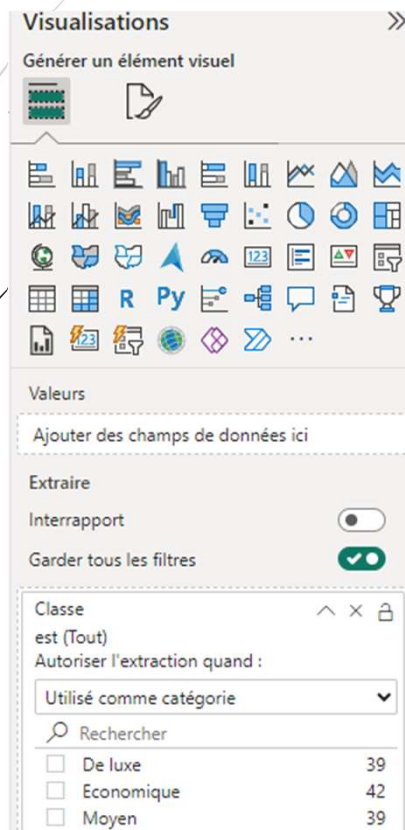
- 1. Copier/coller votre fichier PBIX pour faire l'exercice et supprimer pages existantes sur la copie :**
  - Modifier le « papier peint » de la première page : choisir l'image que vous souhaitez sur ce fond de page,
  - Rajouter sur le « canevas » de la première page, une couleur correspondante à l'image et jouer sur la transparence (autour de 50% pour visualiser la couleur mais l'image également à l'arrière).
- 2. Sur une seconde page, faire un graphique puis personnaliser le thème actif :**
  - Créer un graphique en histogramme, avec les catégories en axe principal, les classes en légende et le montant des ventes en valeurs.
  - Dans « personnaliser le thème », modifier les couleurs principales et appliquer pour voir les différences. Faites de même avec la modification du texte, la bordure (éléments visuels) et le volet de filtre.
- 3. Enregistrer le fichier comme « Modèle » :**
  - Cliquer sur Exporter, puis Modèle et lui donner un nom.



Créer des rapports  
"Maître-Détail" en  
utilisant les filtres  
d'extractions

# Affichage, navigation et visualisation

## Les filtres d'extractions (1/3)



➤ **Les filtres d'extractions** permettent d'atteindre une **page cible filtrée** en fonction d'un élément sélectionné dans une autre page / rapport.

➤ Exemple :

On a une page de départ avec un graphique par classe de produits

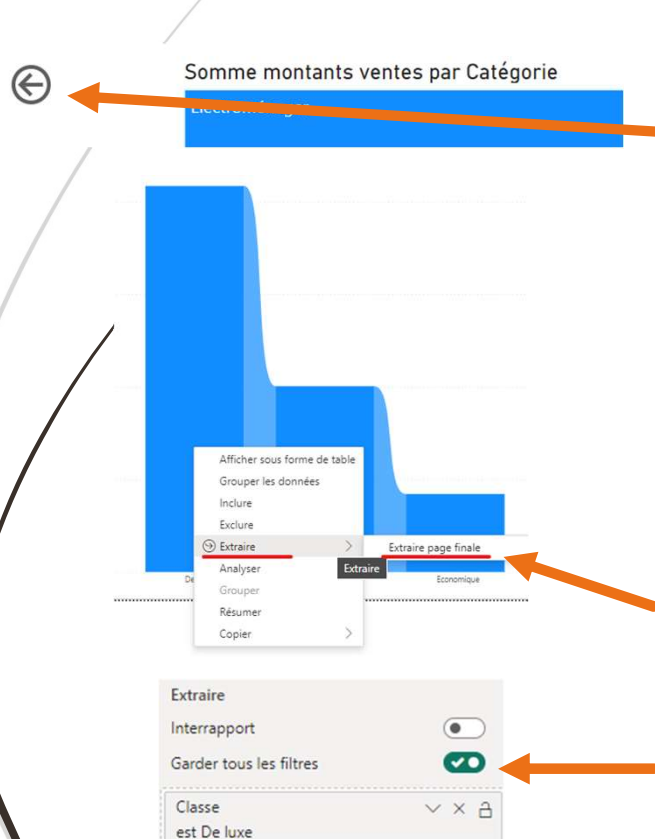
On a une page finale avec une treemap par catégorie.

Quand on clique sur une classe de produit dans la page de départ, on veut atterrir sur la page finale filtrée sur la classe sélectionnée.

- 1) Créer les deux pages et **se positionner sur la page finale.**
- 2) Ne pas cliquer/sélectionner aucun visuel de la page.
- 3) **Glisser le champ d'extraction « Classe » dans la partie « Ajouter des champs d'extractions »** de la fenêtre « Visualisations » de la page.

# Affichage, navigation et visualisation

## Les filtres d'extractions (2/3)



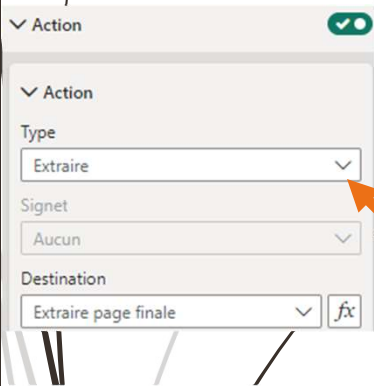
➤ Une flèche précédente apparaît automatiquement, tout en haut à gauche de la page finale. Elle permet **de retourner automatiquement sur la page précédente/de filtrage** de la page actuelle. *Il faut d'abord avec utiliser un filtre sur une autre page pour qu'elle fonctionne.*

- 1) **Aller sur la page de départ :**
- 2) **Cliquer-droit sur une des barres** de l'histogramme « De luxe » par exemple.
- 3) Cliquer sur « **Extraire** » : toutes les pages ayant un champ d'extraction sur la colonne « classe » seront proposées.
- 4) Cliquer sur « **Extraction page finale** » : on navigue directement vers la page finale qui est filtrée sur la classe sélectionnée dans la page de départ.

*Si des segments ou filtres étaient utilisés dans la page de départ, ils filtrent également la page finale si la case est cochée dans cette dernière.*

# Affichage, navigation et visualisation

## Les filtres d'extractions (3/3)



Selectionner une classe  
dans le graphique de  
gauche

Cliquer sur la flèche  
pour continuer la  
navigation



- **Le clique-droit + Extraction + sélection de la page**, n'est pas des plus simples et des plus rapides pour les utilisateurs finaux. Le mieux est de leur **incorporer des boutons** 😊
1. Sur la page de départ, **insérer un bouton « flèche droite »**
  2. Dans **la mise en forme du bouton**, activer **l'action**. Sélectionner le type « **Extraire** » et la destination « **Extraire page finale** » (sélection de la page finale).
  3. Afin que le bouton puisse extraire et donc nous emmener sur la page finale filtrée, il **faut cliquer sur une classe dans le graphique** de la page de départ (pour la sélectionner).
  4. On peut informer de la procédure à l'utilisateur dans un visuel au-dessus de la flèche : = `IF( ISFILTERED( Produit[Classe] ) ; "Cliquer sur la flèche pour continuer la navigation" ; "Sélectionner une classe dans le graphique de gauche"`)




# Création d'une extraction

## Exercices

### 1. Créer des deux pages d'exercices, page d'extraction et page de destination :

- Création d'une « Treemap » avec en catégorie de visuel la « marque » et en valeur « montant des ventes » et l'intitulée « Extraction exo »,
- Dans une seconde page, créer un graphique en histogramme, avec les catégories en axe principal, les classes en légende et le montant des ventes en valeurs. Laisser de la place en haut à gauche et renommer la page en « extraction exo 2 »,
- Toujours dans cette seconde page, créer une extraction sur le champ « marque », ce qui crée par défaut une flèche « page précédente » en haut à gauche.

### 2. Sur la page principale « extraction exo », création du texte interactif en fonction de l'action et du bouton pour être dirigé vers la page de destination filtrée :

- Création d'un visuel s'intitulant « Carte »  et créer une mesure pour l'affichage du texte d'aide à l'utilisateur. Celui-ci indiquera à l'utilisateur de cliquer sur une marque, s'il ne l'a pas encore fait. Dans le cas échéant, il lui indiquera de cliquer sur la flèche droite à côté du visuel carte, que l'on créera dans un second temps,
- Création d'un bouton « Flèche droite » puis dans sa mise en forme, activer l'action. Puis dans la section « action », sélection du type « extraire » puis « destination » vers la page de destination voulue.

# Utiliser les signets pour la navigation et le Storytelling

# Affichage, navigation et visualisation

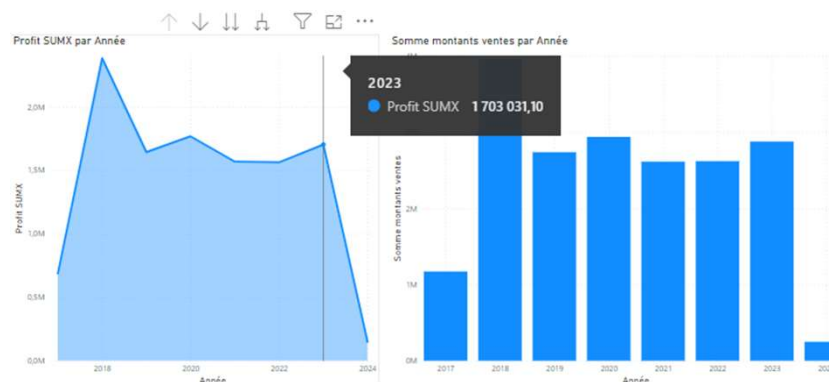
## Utilisation des signets – Cas exemple

➤ **Les signets** permettent de naviguer différemment sur la page et par conséquent, d'afficher les **données différemment selon les choix utilisateurs** (boutons).

➤ Exemple :

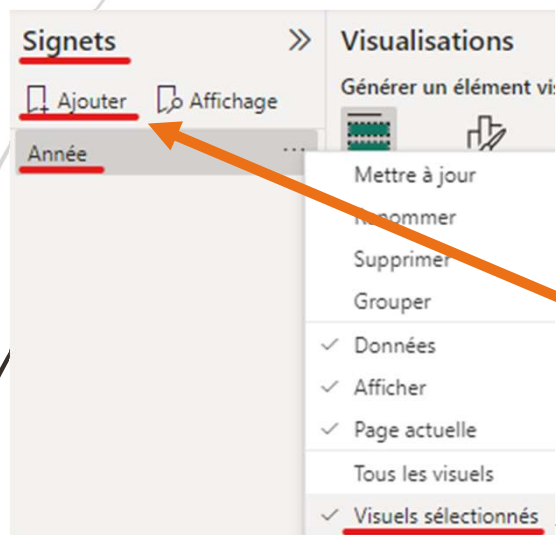
*Sur notre page de rapport, on a deux visuels avec 3 niveaux de hiérarchies : années, trimestres et mois.*

*On aimerait que l'utilisateur - au lieu d'utiliser les différents types de flèches pour descendre dans la hiérarchie de chaque visuel - clique simplement sur les boutons associés à la hiérarchie voulue pour que tous les visuels de la page affichent la hiérarchie souhaitée.*



# Affichage, navigation et visualisation

## Utilisation des signets – Création (1/2)



### ➤ Création du premier signet :

➤ Dans le **ruban en haut**, cliquer sur **afficher**, puis sur **Signets** afin **d'ouvrir la fenêtre « Signets »** à droite.

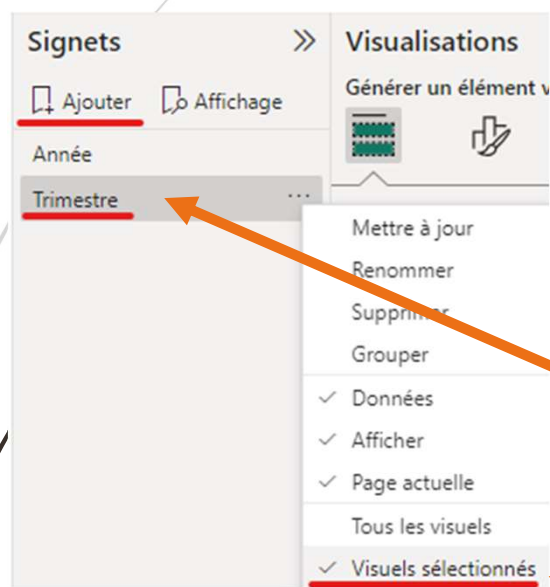
1. **Se positionner sur le premier niveau de hiérarchie (année)** sur tous les visuels,
2. **Sélectionner tous les visuels concernés** par notre premier signet (deux visuels avec CTRL),
3. Cliquer sur **Ajouter** dans partie Signets : renommer le signet **« Année »**,

*Le signet capture l'état de l'affichage au moment de la création du signet.*

3. **Cliquer droit sur le signet** (en ayant toujours en sélection tous les visuels concernés) et **sélectionner « Visuels sélectionnés »**.

# Affichage, navigation et visualisation

## Utilisation des signets - Création (2/2)



### ➤ Création des signets suivants :

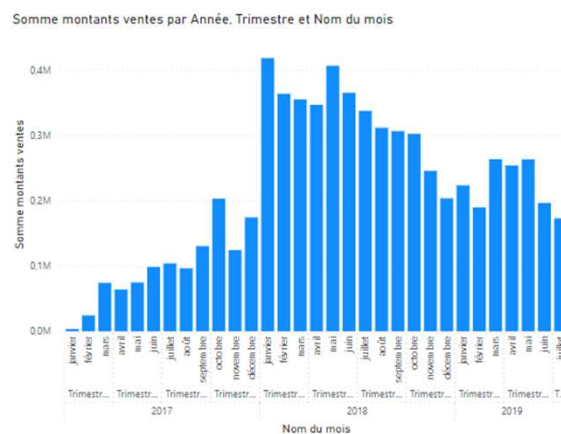
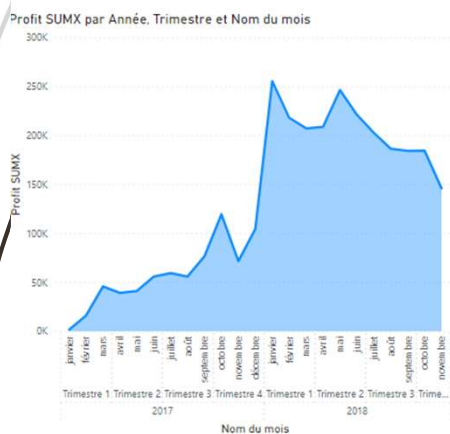
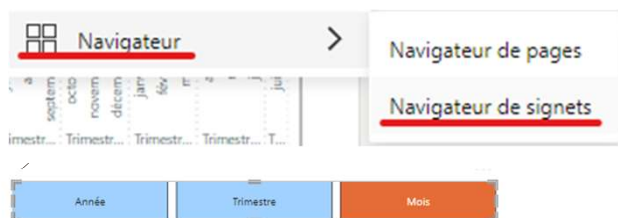
1. **Descendre sur le second niveau de hiérarchie (trimestre)** (puis « **mois** » quand on répètera la procédure pour le dernier) sur tous les visuels,

*Même procédure que pour le premier signet :*

2. **Sélectionner tous les visuels concernés** par notre premier signet (deux visuels avec CTRL),
3. Cliquer sur **Ajouter** dans partie Signets : renommer le signet « **Trimestre** » (puis « **mois** » quand on répètera la procédure pour le dernier),
4. **Cliquer droit sur le signet** (en ayant toujours en sélection tous les visuels concernés) et **sélectionner « Visuels sélectionnés »**.

# Affichage, navigation et visualisation

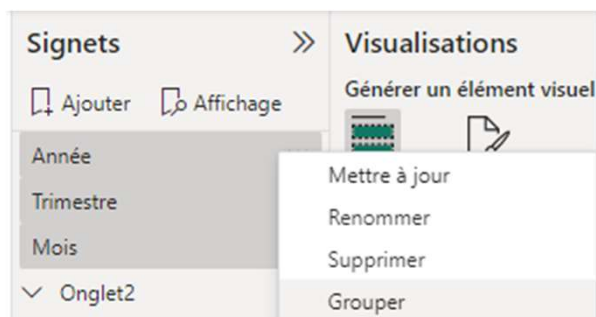
## Utilisation des signets - Boutons (1/3)



- **Création des boutons pour naviguer dans les signets:**
- Dans le **ruban en haut**, cliquer sur **Insérer**, puis sur **Boutons** puis **Navigateur** « **Navigateur de signets** ».
- **Des boutons affichant tous les signets apparaissent.** En faisant un clic + CTRL sur mois : tous les visuels se positionnent sur l'affichage mois.

# Affichage, navigation et visualisation

## Utilisation des signets - Boutons (2/3)



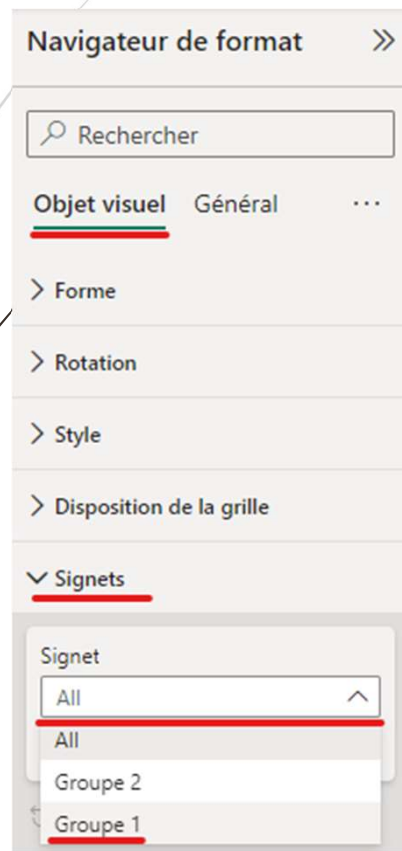
➤ **Créer des groupes de signets pour les utiliser dans les boutons :**

Si j'ai **plusieurs signets** correspondant à **des pages différentes**, il me faut les **grouper** pour que le **navigateur de signets** ne m'affiche pas tous les signets de toutes les pages.

➤ **Renommer avec le nom de la page/onglet du rapport.**

# Affichage, navigation et visualisation

## Utilisation des signets - Boutons (3/3)



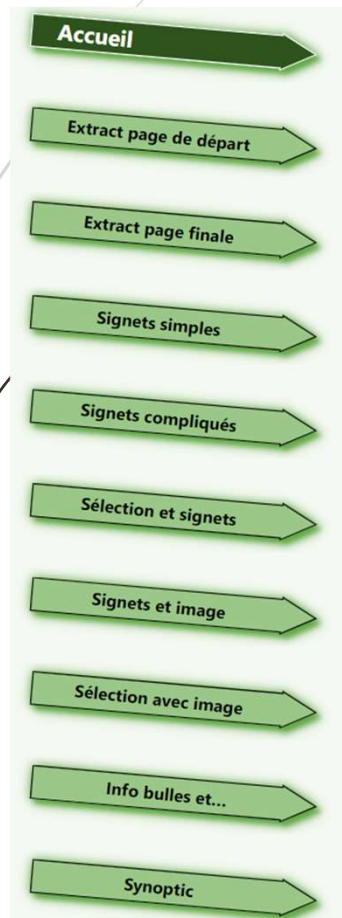
- Après avoir créé mes groupes de signets, je vais dans la **mise en forme** de mon **Bouton de signet**.
- Dans la dernière partie « **Signets** », il est positionné sur All, cliquer sur la flèche et sélectionner le bon groupe de signets à afficher dans le bouton de la page en cours. Le bouton s'affiche ensuite correctement :





# Création d'un plan (navigateur de signets)

## Exercices



### 1. Création des signets :

- Création d'un signet général pour chaque page du rapport (dans l'ordre),
- Création d'un groupe de signets s'intitulant « plan »

### 2. Création de la page de plan / Sommaire du rapport :

- Création d'un navigateur de signets (on aurait pu créer un navigateur de pages, mais il manquerait la page de sommaire que l'on ait en train de créer) Insérer – Boutons – Navigateur – Navigateur de signets,
- Dans la partie disposition, indiquer l'orientation « verticale »,
- Aller dans le format du navigateur et sélectionner le bon signet (le groupe « plan » créé plus tôt),
- Gérer la mise en forme de l'objet visuel (parties Forme, Style et disposition de grille) puis test du navigateur.

*\*Exemple de restitution quand toutes les démonstrations sont réalisées.*



Utiliser l'onglet des filtres  
et l'onglet sélection

# Affichage, navigation et visualisation

## Fenêtre des filtres (1/2)

The screenshot shows the Power BI interface with the 'Filtres' (Filters) pane on the left and the 'Visualisations' pane on the right. The 'Filtres' pane is divided into three sections: 'Filtres sur ce visuel' (Filters on this visual), 'Filtres dans cette page' (Filters on this page), and 'Filtres dans toutes les pages' (Filters on all pages). The 'Filtres sur ce visuel' section is active, showing a list of fields: 'Continent est (Tout)', 'Pays est (Tout)', 'SalesAmount est (Tout)', and 'Ville est (Tout)'. Below this is a search bar and a button 'Ajouter des champs de do...'. The 'Visualisations' pane shows a grid of visualization options and a 'Filtres' icon. A modal window is open over the 'Filtres' icon, showing the configuration for the 'RegionCountryName' filter. The modal window has a title 'RegionCountryName' and a subtitle '5 premiers par Sales...'. It contains a 'Type de filtre' dropdown menu set to 'N premiers', an 'Afficher les éléments' section with a 'Haut' dropdown and a '5' input field, and a 'Par valeur' section with a 'SalesAmount' dropdown. At the bottom of the modal is an 'Appliquer le filtre' button.

➤ On peut choisir **d'effectuer un filtre**, sur **un seul visuel**, sur **toute la page** ou sur **toutes les pages** du rapport.

➤ Pour les champs de type Texte, nous pouvons réaliser des filtres de bases en cochant les valeurs à sélectionner, ou avancé (contient, ne contient pas, etc.) ou

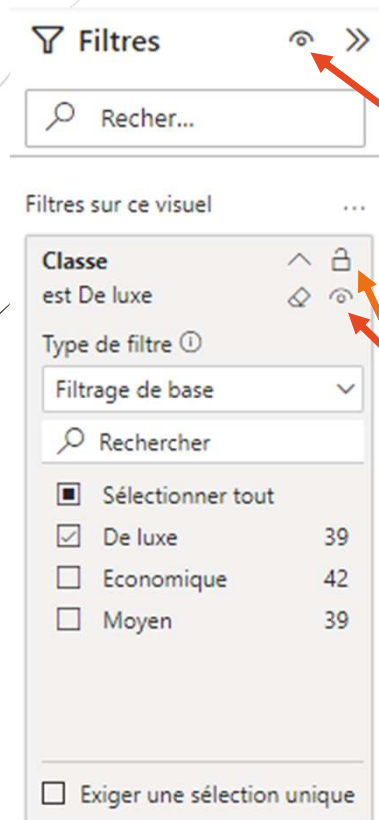
**N premiers :**

Je peux décider d'avoir, par exemple, le **top 5 des pays** en fonction **de leur montant des ventes**.

Pour les pays, je choisis le type de filtre « N premiers », je précise si haut ou bas (top ou flop) et le nombre à afficher puis à trier par la valeur (Sales amount).

# Affichage, navigation et visualisation

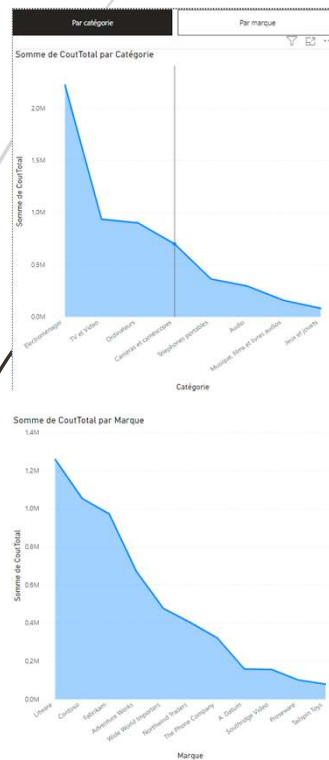
## Fenêtre des filtres (2/2)



- **Les champs qui sont sur un visuel** sont dans les « filtres sur ce visuel » par défaut et ne peuvent être supprimés.
- Nous **pouvons masquer la vue de l'onglet/fenêtre Filtres** afin que les utilisateurs n'y aient pas accès. C'est logique en soit, ce sont **des filtres de conception et non pas, des filtres utilisateurs** comme le sont naturellement les segments.
- Nous pouvons **masquer chaque filtre individuellement** pour l'utilisateur final.
- Nous pouvons laisser afficher un filtre mais **le bloquer** à l'utilisation pour le lecteur/utilisateur final. Il pourra le voir et en être conscient sans pouvoir le modifier (évite les problèmes si beaucoup d'utilisateurs finaux).

# Affichage, navigation et visualisation

## Fenêtre de sélection – Cas exemple



➤ **La fenêtre de sélection** permet de **masquer ou d'afficher** des éléments sur une page de rapport en fonction de spécification par des signets en lien.

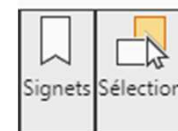
➤ Exemple :

Sur notre page de rapport, on a deux visuels : deux graphiques de même type (graphique de zone empilé) affichant tous deux la somme des coûts. Cependant, un réparti cette valeur par catégorie et l'autre par marque.

On aimerait que l'utilisateur - au lieu d'afficher tous les visuels alors qu'il n'en veut que l'un ou l'autre selon l'utilisation **OU BIEN**, afin d'éviter de créer autant de pages que de répartitions différentes - clique simplement sur les boutons associés à la répartition souhaitée et que seuls les visuels adéquats apparaissent.

# Affichage, navigation et visualisation

## Fenêtre de sélection – Création (1/2)



\* Faire deux graphiques identiques mais pas avec même axe, puis les positionner l'un sur l'autre.

### ➤ Création du premier signet :

➤ Dans le **ruban en haut**, cliquer sur **afficher**, puis sur **Signets** et « **Sélection** » afin **d'activer et donc afficher les fenêtres « Signets » « Sélection »**.

1. Dans la fenêtre **sélection**, dans « **ordre des couches** » **masquer « Par marque »** :
2. Dans la fenêtre Signets, créer un nouveau signet intitulé « par catégorie » et laisser « Tous les visuels » car masque l'autre visuel et donc n'agit pas que sur le visuel sélectionné.

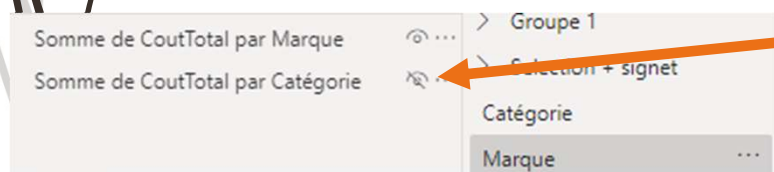
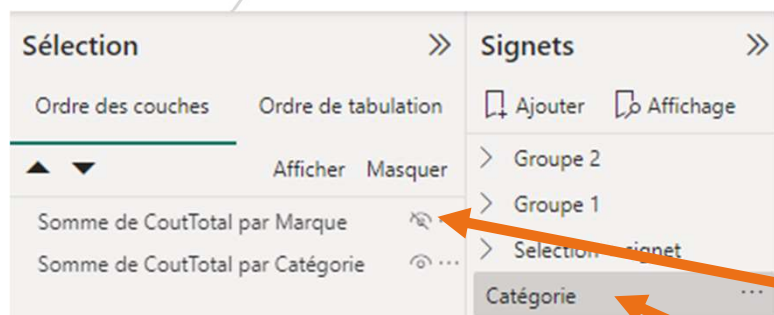
*Le signet capture l'état de l'affichage au moment de la création du signet.*

### 2. Faire de même pour le second signet.

Bien penser à démasquer « Par marque »

Masquer « Par catégorie »

Créer un signet « Par marque » puis grouper les deux signets.



# Affichage, navigation et visualisation Fenêtre de sélection – Création (2/2)



- Dans le **ruban en haut**, cliquer sur **Insérer**, puis sur **Boutons** puis Navigateur « **Navigateur de signets** ». Dans la mise en forme, sélectionner le **bon groupe de signets** comme vu précédemment dans partie « Signets ».
- **Des boutons affichant tous les signets apparaissent**. En faisant un clic + CTRL sur mois : seuls les visuels concernés par le bouton/signet sélectionné s'affichent.
- On peut positionner les deux visuels l'un sur l'autre pour un gain de place. Quand je clique sur « **Marque** », le graphique par marque s'affiche et pareil pour la répartition par catégorie.

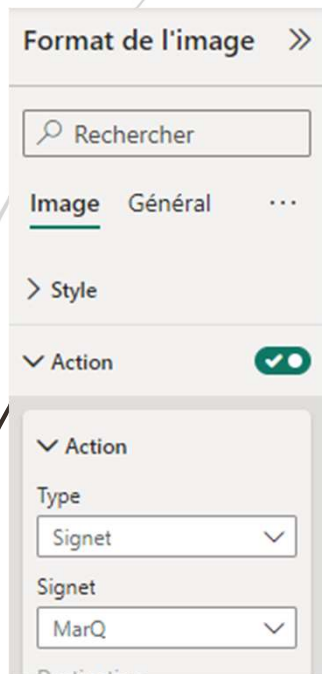


Représenter avec des images. Préparer des commentaires dynamiques et personnaliser info-bulles.



# Affichage, navigation et visualisation

## Insertion d'image et action/sélection

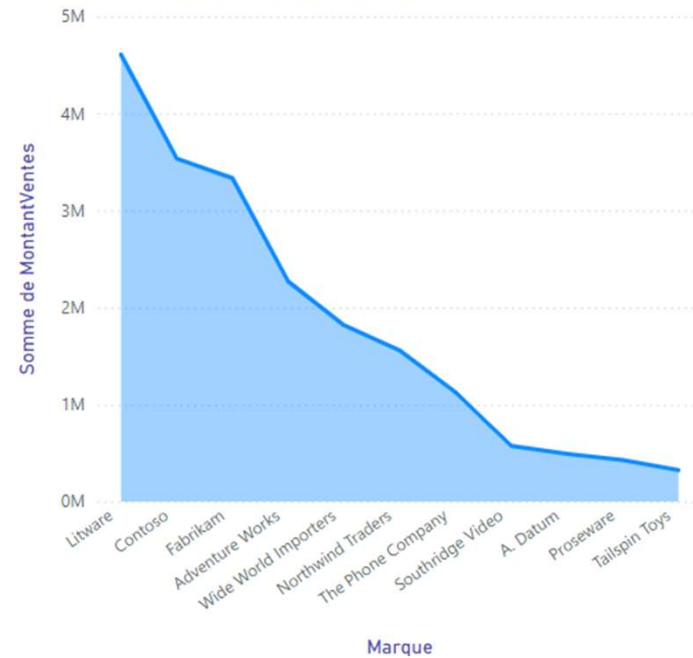


- Dans le **ruban en haut**, cliquer sur **Insérer**, puis sur **Image**. Sélectionner ensuite **l'image sur l'ordinateur**.
- **Sur une page vierge, je crée deux images** : une pour les marques/brand et un autre pour les catégories. → « Dossier exo » → « Partie 5 BIG » → télécharger les images marques et catégories.
- **Je sélectionne l'image des marques (Rose / Brand) :**
  - ✓ Dans format de l'image, j'active Action.
  - ✓ Je clique dessus, et je sélectionne comme type d'action « Signet » puis je sélectionne le signet « MarQ ».
- Quand je clique dessus, **je navigue jusqu'à la page contenant le signet correspondant** « Signets et image » et sont affichés uniquement **le graphique et l'image** correspondants.

# Affichage, navigation et visualisation

## Insertion d'image = résultat

Somme de MontantVentes par Marque



# Affichage, navigation et visualisation

## Insertion de commentaires dynamiques

➤ Dans le **ruban en haut**, cliquer sur **Insérer**, puis sur **Texte**.  
**Une zone de texte apparaît.**

➤ Si je clique dessus, je peux écrire du **texte libre mais non dynamique**.

+ Valeur | Revue

Créer une valeur dynamique qui est mise à jour avec vos données

Comment calculeriez-vous cette valeur

somme montants ventes

Résultat

14 037 062,69 €

\$% Général \$ % Auto

Nommez votre valeur

# Profit

**249 214,89 de Montant des Ventes**

➤ Si je souhaite y ajouter des données dynamiques, je clique sur « **+ Valeur** »

➤ Je vais chercher **le nom de la mesure à afficher** ou les mesures à calculer entre elles.

➤ Je **gère la mise en forme** du résultat

➤ Je donne un **nom à la valeur**.

→ **La valeur s'affiche dans la zone de texte suivi du texte non dynamique que l'on inscrit à la suite.**

# Affichage, navigation et visualisation

## Insertion de commentaires dynamiques

The screenshot shows a two-step process in a dynamic text editor:

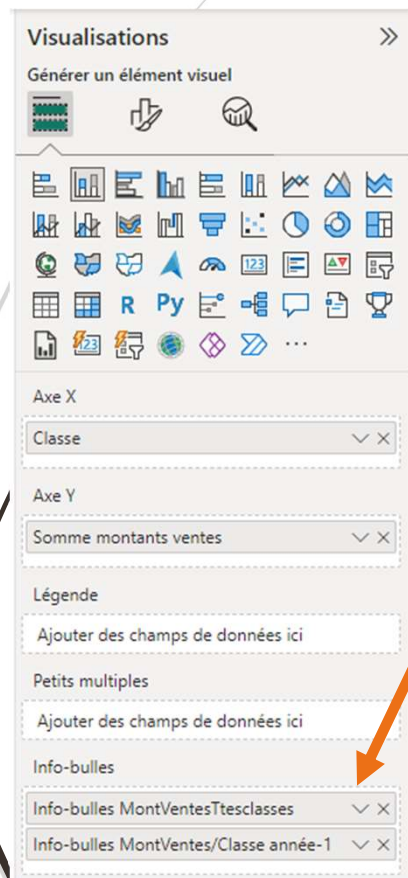
- Step 1 (Top):** A header with a plus icon and the word "Valeur" next to a list icon and the word "Revue". Below it, the text "Créer une valeur dynamique qui est mise à jour avec vos données" and "Comment calculeriez-vous cette valeur" is followed by a text input field containing "année en cours".
- Step 2 (Bottom):** A header with a plus icon and "Valeur" next to a list icon and "Revue". Below it, the text "Vérifier et modifier les valeurs de votre narration" is followed by a list of three items: "#MontantVentes", "#AnnéePrécédente", and "#Evolution". Each item has a dropdown arrow, a pencil icon, a plus icon, and a trash icon.

Orange arrows point from the text on the right to the "année en cours" input field, the "249 214,89 de Montant des Ventes en 2024" text, and the "Revue" button in the second step.

- Création d'une **mesure** qui récupère l'**année en cours** : = `year(today())`
- Mettre un **filtre sur la page entière** sur l'année en cours.
- Rajouter « en » puis aller chercher la valeur de la mesure correspondante pour afficher l'**année en dynamique** dans la zone de texte.
- Pour retrouver les **valeurs créées précédemment** et les rajouter dans le texte et/ou les modifier, cliquer sur **Revue**.

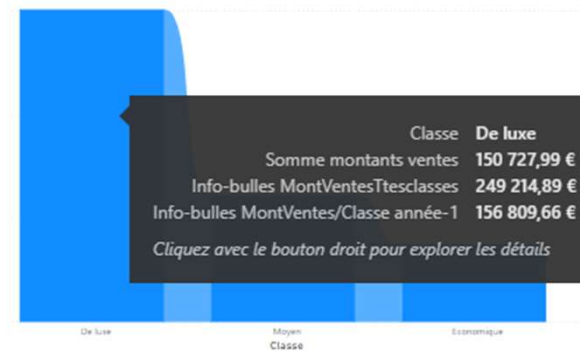
# Affichage, navigation et visualisation

## Personnaliser info-bulles



- Création d'une **mesure** qui **calcule la somme du montant des ventes toutes classes de produit réunies** :  
= `CALCULATE([Somme montants ventes]; ALL(Produit[Classe]))`
- Récupération de la **mesure** qui **calcule la somme du montant des ventes de l'année dernière** : (calcule pour la classe sur laquelle on passe avec la souris)

➤ **Faire glisser** les deux mesures dans « **Info-bulles** » de la fenêtre « Visualisations ». **L'info-bulle fait apparaître ses deux mesures en + :**



# Texte dynamique et info-bulle

## Exercices

### 1. Créer un commentaire dynamique et une info-bulle :

- Création d'un texte sur la page « Extraction exo 2 » indiquant « Le profit correspondant est de 'Valeur dynamique' »,
- Rajouté le nom de la marque concernée. Il faut donc créer une mesure qui va chercher la marque filtrée sur la page (révision 😊 selectedvalue).
- Création d'une info-bulle en plus reprenant sur le graphique la valeur du profit. (Faire glisser le champ dans partie « info-bulles » du visuel).

Le profit correspondant est de  
€ 1 386 911,50 pour Fabrikam



# Définir un visuel personnalisé avec le Synoptic Pannel

# Définir un visuel personnalisé avec le Synoptic Panel

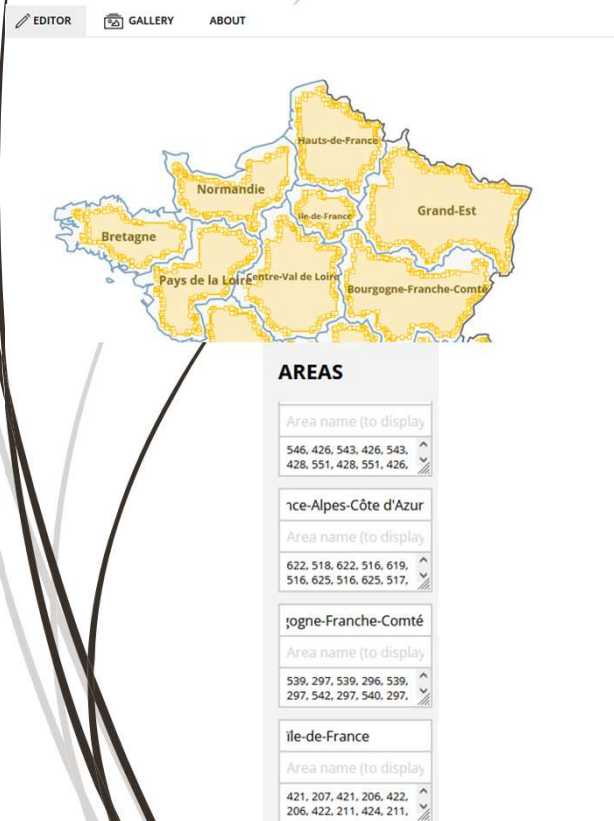
- ▶ Le **Synoptic Panel** est un visuel personnalisé de **Power BI** qui permet de **cartographier des zones personnalisées sur une image**, puis **de les lier à des données**.

Cela signifie que vous pouvez transformer **n'importe quelle image statique en une visualisation interactive**. Que vous ayez besoin de représenter des données sur un plan d'étage, une carte géographique ou un schéma de processus, le **Synoptic Panel** offre une solution.

- ▶ **Pourquoi utiliser le Synoptic Panel ?**
- ▶ Il permet de représenter les données de manière spatiale, ce qui facilite l'interprétation des relations et des tendances. Cela peut être particulièrement utile dans des secteurs comme la logistique où la visualisation de l'espace et du placement peut avoir une grande influence sur les décisions.



# Définir un visuel personnalisé avec le Synoptic Pannel



1. **Au préalable** : Avoir une image (PNG de préférence) et un jeu de données associé à l'image.

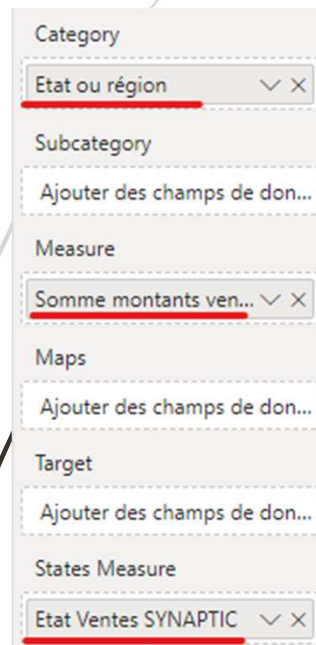
2. **Mettre des repères dans l'image** :

- ✓ Ouvrir le site Synoptic : <https://synoptic.design/>,
- ✓ Créer des repères dans l'image : soit zone carrées/rectangulaire avec le viseur sinon avec la baguette magique (régions – qui remplit la zone).

*Pour chaque zone, changer le numéro par le nom correspondant aux données : Si Île de France dans les données, changer 1 et mettre le nom exactement comment il est écrit dans les données. Si on laisse 1, il ne fera pas la correspondance entre 1 et Île de France.*

- ✓ Exporter au format SVG.

# Définir un visuel personnalisé avec le Synoptic Pannel



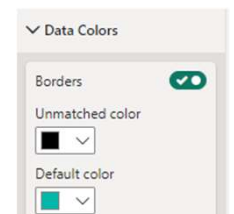
### 3. Dans Power BI : Associer l'image et le jeu de données dans Power BI :

- ✓ Intégrer le jeu de données (Import si données pas chargées),
- ✓ Ajouter le visuel Synoptic et Importer à l'intérieur le visuel avec les repères créés (SVG enregistré sur l'ordinateur),

Ajout Catégorie et mesure avant de pouvoir sélectionner image créée !

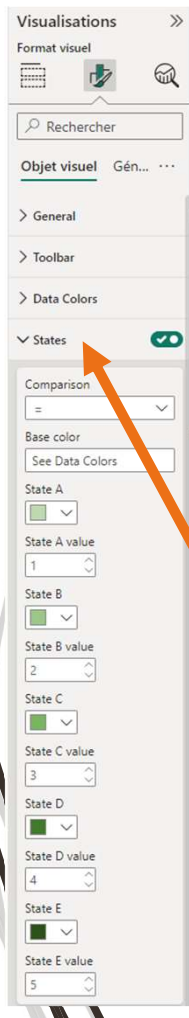
### 4. Insérer des données dans le visuel (fenêtre visualisations) :

- ✓ Dans la zone catégorie : insérer le champ région,
  - ✓ Dans la zone mesure : insérer la mesure (MontantVentes),
- Dans la zone de mise en forme du visuel, dans la partie « Data Colors » :  
S'il y a des données correspondantes à la région, alors couleur désignée (verte)  
sinon autre couleur (ici noir pour unmatched color).



- ✓ Dans la zone « Etats de mesures »\* : mettre une mesure qui donne le numéro de couleur (1 pour clair → 5 pour foncé). → **Explication sur la diapositive suivante.**

# Définir un visuel personnalisé avec le Synoptic Pannel



## ➤ Créer deux mesures :

- Une mesure qui crée **la part des ventes** sur le global, toutes région confondue.

= `[Somme montants ventes]/CALCULATE([Somme montants ventes]; all('Géographie'[Etat ou région]))`

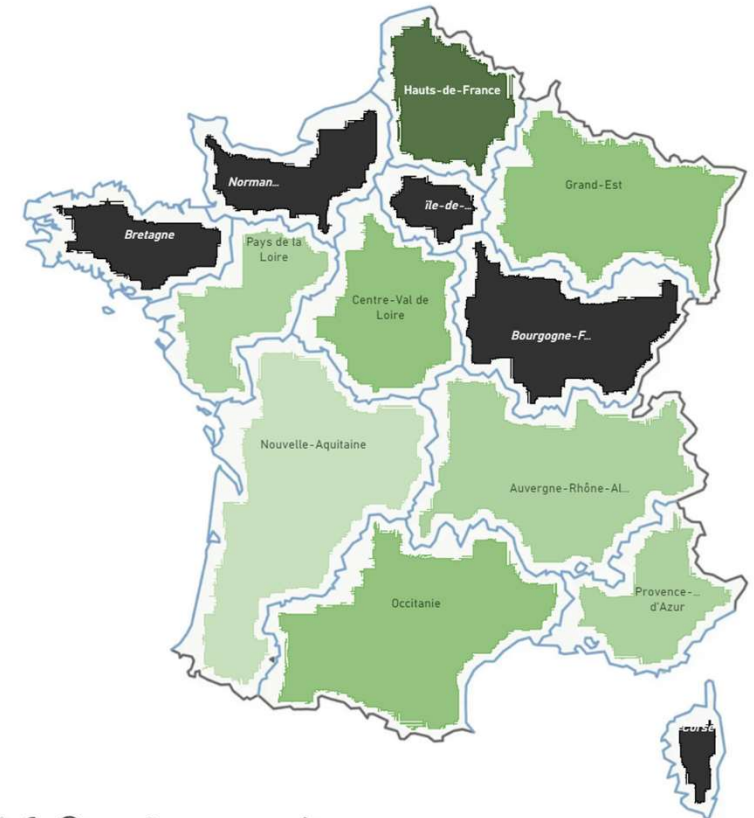
- Une mesure qui crée **l'état des ventes\*** :

= `IF([Part Ventes région] < 0,02; 1;  
IF([Part Ventes région] < 0,04; 2;  
IF([Part Ventes région]<0,08; 3;  
IF([Part Ventes région]<0,35; 4;5))))`

## ➤ Dans la partie « Visualisations », « format visuel », « States » :

**Ajuster les couleurs** pour chaque valeur de la mesure état des ventes.

Si la valeur est égale à 1, alors vert très clair, etc.



212

# Annexes

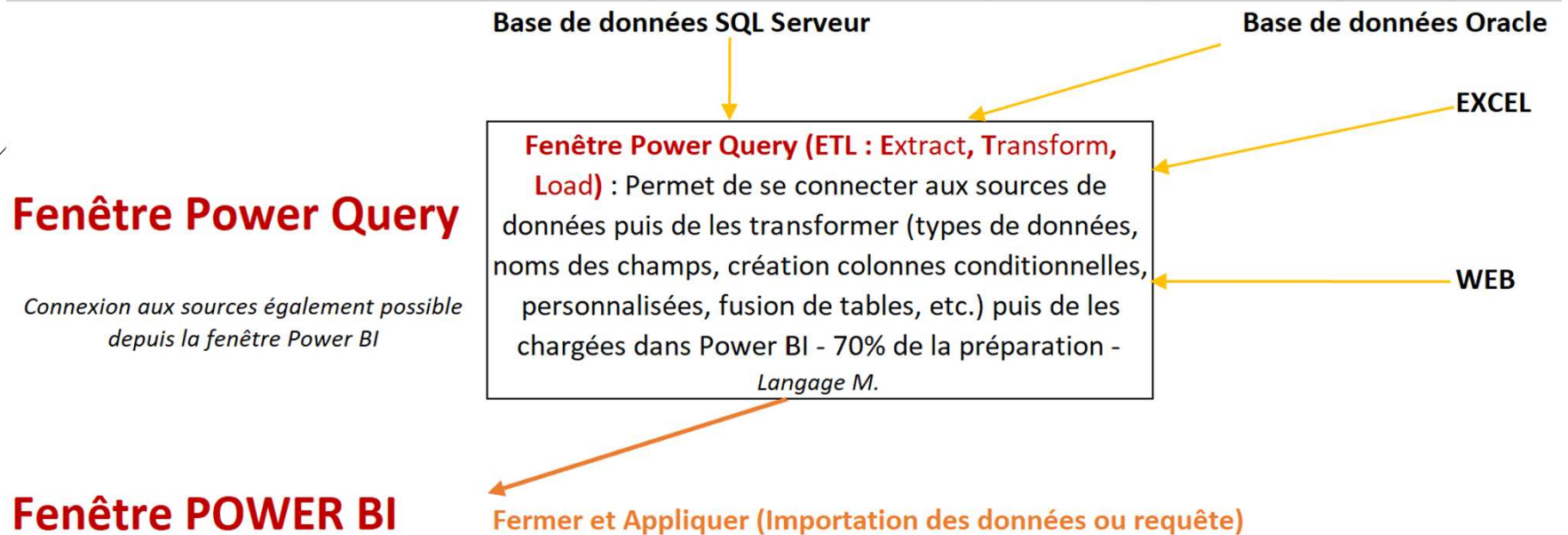


# Annexes

Partie 1 : Rappel du cycle sur Power BI et de préparation des données  
(éléments de formation débutant – niveau 1)

# Interface Power BI : Où et quoi ?

## Chargement et transformation



# Interface Power BI : Où et quoi ?

## Optimisation du jeu de données et création des rapports

### Fenêtre POWER BI

*Onglets de gauche*

--Vue de modèle

Power Pivot (vue de diagramme) --> EXCEL

-- Affichage table (et ou Affichage du rapport)

Si DIRECT QUERY n'existe pas  
disponible dans Onglet Rapport

Power Pivot (vue de données)--> EXCEL

--Affichage du Rapport

Visuels Excel et Power Map --> EXCEL

Jeux de données -- CUBE TABULAIRE (Fonctionne en In-mémoire) -> Préférable d'avoir minimum 8 Go RAM

- Vérifier les relations entre les tables --> Ce n'est pas parce qu'une relation est créée par Power Bi qu'elle est juste !
- Créer les relations afin d'optimiser le modèle --> vérification des relations dans une matrice (TCD)

-- Optimiser le modèle en supprimant/masquer des tables (fusions de colonnes)

-- Créer les hiérarchies

-- Marquer/spécifier la table date ou la créer (Mesures de type DAX TIME INTELLIGENCE)

-- Créer des nouvelles colonnes et créer des mesures DAX / Indicateurs statistiques avec le langage DAX. (20%)

**Langage DAX => Data analysis expressions -- cube tabulaire**

-- Créer nos visuels et mise en forme du Tableau de bords / reportings.

-- Segments : filtres utilisateurs

-- Filtres (filtres de conception)



# La Business Intelligence

## Création d'un bon jeu de données

- Les bases de données traditionnelles dites « **transactionnelles** » ont **4 points « négatifs »** à effacer pour tendre vers une bonne base de données décisionnelles dites « **jeu de données** » dans **Power BI** :

- **Axes négatifs du transactionnel :**

- Noms des tables / champs / modalités pas clairs.
- Trop de tables à gérer et à connaître.
- Beaucoup trop de champs fonctionnels sans intérêt pour l'analyse finale.
- Beaucoup de données brutes à retravailler avant l'analyse.

- **Axes d'améliorations dans Power BI :**

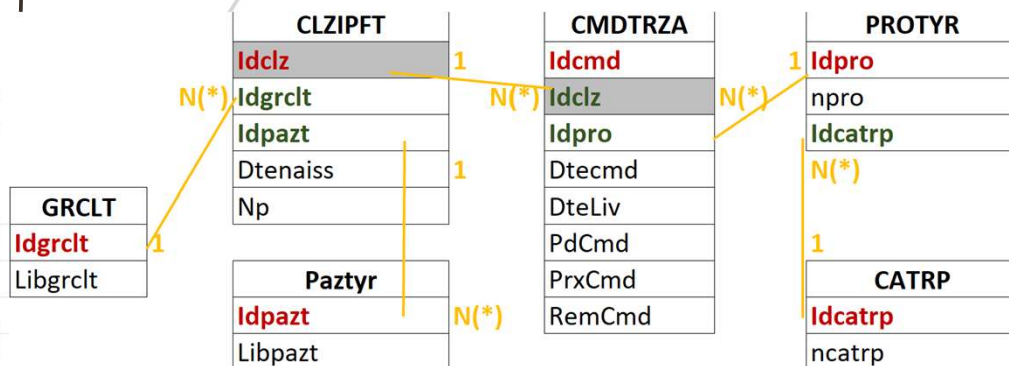
- ✓ **Modification des noms** des tables/champs/modalités afin qu'ils soient lisibles et clairs pour tous et dans les visuels.
- ✓ **Réduction du nombre de tables**
- ✓ **Suppression des colonnes** dispensables pour les besoins de l'analyse finale.
- ✓ **Créer variables d'analyses** : des nouvelles colonnes à partir des données brutes.



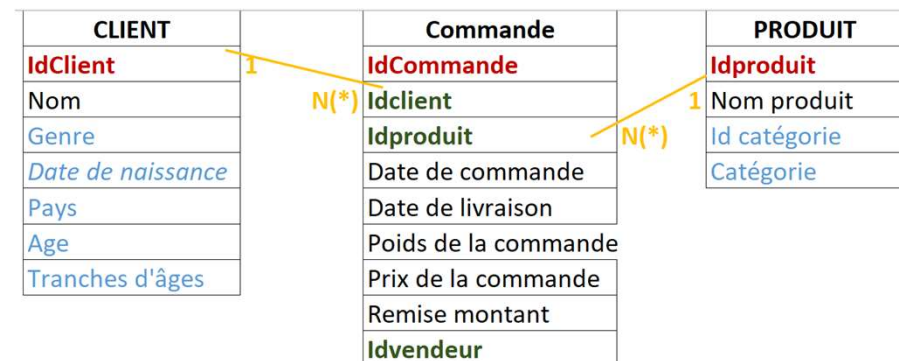
# La Business Intelligence

## Création d'un bon jeu de données

### ➤ BDD transactionnelle :



### ➤ Jeu de données Power BI :






- Libpazt correspondant au « libellé du genre » renommé « genre » et récupéré via un rechercheV dans la table Client.
- Par conséquent, la table Paztyr n'a plus d'intérêt, on gagne une table.
- Le numéro de téléphone du client intitulé « Np » n'étant pas nécessaire pour l'analyse et les tableaux de bord, il a été supprimé.
- À partir de la « dtenaiss », la date de naissance du client, on a créé deux nouvelles colonnes : âge et tranches d'âge.

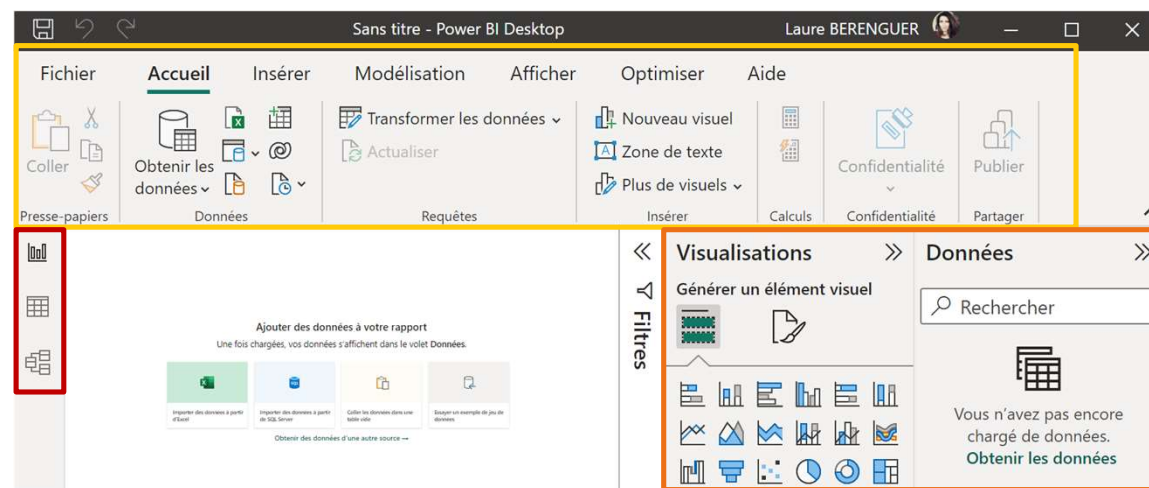
# Annexes

Partie 1 : Rappel sur interface Power BI et connexion aux données  
(éléments de formation débutant – niveau 1)

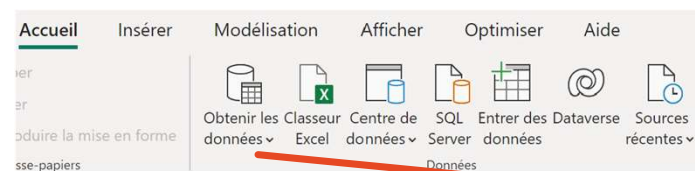
# Interface Power BI : Ruban, onglets de gauche, etc.

## ➤ L'interface est constituée :

1. D'un **ruban d'onglets** en haut, 
2. De **fenêtres à droite** : 
  - Visualisation : création et modification des visuels,
  - Données : affichage des tables et des champs.
3. De **3 icônes de gauche** (de haut en bas) : 
  - l'onglet affichage du rapport : onglet de conception de rapports,
  - L'onglet affichage table : onglet de visualisation des données,
  - L'onglet vue de modèle : onglet de conception du schéma de données et des relations.



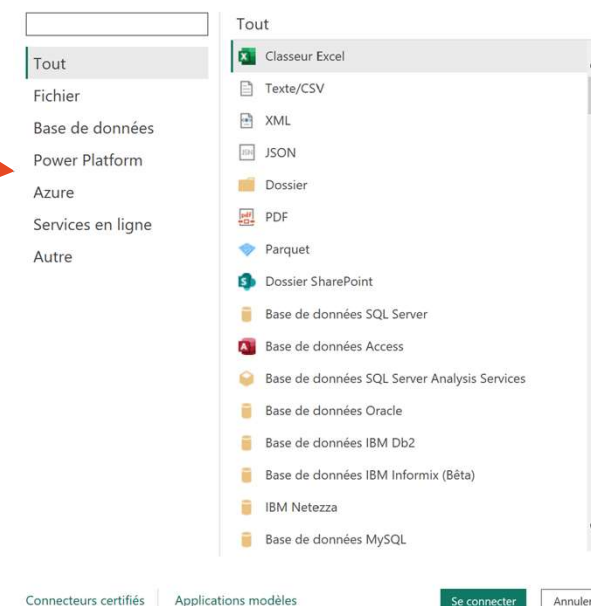
# Obtenir des données : Connexion aux sources



➤ Dans l'onglet du ruban du haut « **Accueil** », plusieurs choix de connecteurs aux données sont proposés :

- Les plus fréquents, visibles directement,
- Pour les autres : cliquer sur « Obtenir les données » puis « Plus » en bas de la liste de choix pour obtenir la fenêtre ci-contre.
- Un large panel de connexions sont possibles, allant du **fichier Excel et/ou CSV aux différentes Bases de Données**, en passant par les services en **lignes et en cloud**.

## Obtenir les données



# Annexes

Partie 3 : Création d'une table de Dates/Calendrier dans Power Query pas à pas sans passer par l'éditeur avancé directement

# Créer un calendrier sur M

## Créer table de dates sur table de faits (1/6)

### ➤ Préparer la table :

- ✓ Créer une référence de la table de faits (Table de ventes),
- ✓ Garder uniquement la colonne de dates (Date de vente),
- ✓ Supprimer les doublons et renommer cette étape « **Base** ».

### Pourquoi les lier ?

- ✓ A chaque **mise à jour de la table de faits** (ventes, inventaire, planning), la table de dates **sera donc recalculée** et contiendra ainsi toutes les dates concernées par l'activité de votre entreprise/organisation/etc.

✓ **Pour des dates de début et dates de fin dynamiques !**

The screenshot shows a table with the following data:

	Date de vente
1	16/01/2017
2	17/01/2017
3	18/01/2017
4	21/01/2017
5	23/01/2017
6	30/01/2017
7	02/02/2017
8	09/02/2017

On the right, the 'PROPRIÉTÉS' panel shows:

- Nom: 1 Ventes pour table de Dates
- Toutes les propriétés
- ÉTAPES APPLIQUÉES:
  - Source: Garder uniquement col Date ... \*
  - Base

# Créer un calendrier sur M

## Déterminer dates de début et de fin (2/6)

La **table de dates doit être constituée** de la manière suivante :

- ✓ Elle débute à la **première date de la table de faits** (ventes par exemple),
- ✓ Elle se termine à **la dernière date de la table de faits** (ventes par exemple),
- ✓ Elle contient l'ensemble des jours entre ces 2 dates

### ➤ Création StartDate:

- ✓ Utiliser un Filtre chronologique : Est la plus ancienne
- ✓ Clic-droit sur l'unique valeur de la colonne : sélectionner « Drill Down » créant un seul résultat,
- ✓ Renommer l'étape StartDate

### ➤ Passage vers EndDate :

- ✓ Clic-droit dessus et sélectionner « **Insérer l'étape après** »

	Liste
1	16/01/2017

ÉTAPES APPLIQUÉES

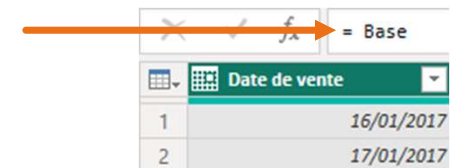
- Source
- Garder uniquement col Date ... ✱
- Base ①
- Filtre ancien ✱
- ✱ StartDate
  - ✱ Modifier les paramètres
  - 📄 Renommer
  - ✕ Supprimer
  - Supprimer jusqu'à la fin
  - Insérer l'étape après**

# Créer un calendrier sur M

## Déterminer dates de début et de fin (3/6)

### ✓ Modifier le code de la nouvelle étape :

En changeant StarDate dans la formule par Base afin de revenir au résultat avant la création de StartDate.



	Date de vente
1	16/01/2017
2	17/01/2017

### Création EndDate à la suite :

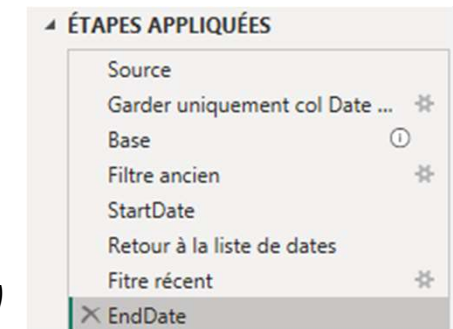
- ✓ Utiliser un Filtre chronologique : Est la plus récente
- ✓ Clic-droit sur l'unique valeur de la colonne : sélectionner « Drill Down » créant un seul résultat,

**Modifier le code fixe en dynamique** (si actualisation données) :  
 := #"Filtre Récent"{0}[Date de vente] (prend la date sur la 1<sup>ère</sup> ligne =0)

Au lieu de

= #"Filtre Récent"{[#"Date de vente"=#date(2024, 02, 13)]}[Date de vente]  
 (prend la date trouvée lors de l'import des données et non pas actualisée)

- ✓ Renommer l'étape **EndDate**



ÉTAPES APPLIQUÉES	
Source	
Garder uniquement col Date ...	✳
Base	①
Filtre ancien	✳
StartDate	
Retour à la liste de dates	
Filtre récent	✳
✕ EndDate	



# Créer un calendrier sur M

## Fonction List.Dates (4/6)

➤ **Utilisation de la fonction List.Dates composée de 3 arguments :**

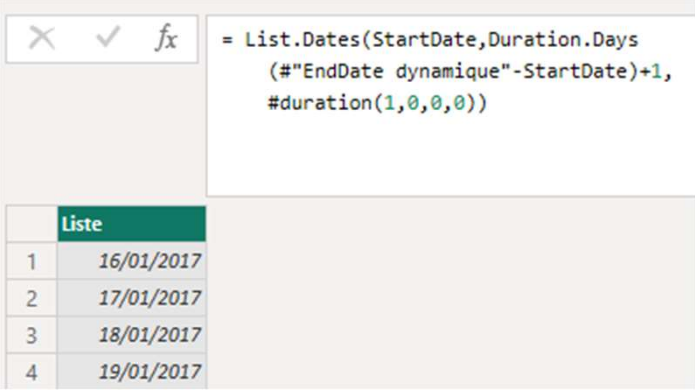
- ✓ **Start** (date) : la date de début du calendrier,
- ✓ **Count** (décimal) : le nombre de dates de la liste,
- ✓ **Step** (durée) : l'incrément choisi entre chaque date de la liste.

- ✓ **La syntaxe de l'argument step est la suivante :**  
*#duration(days as number, hours as number, minutes as number, seconds as number)*
- ✓ Pour notre exemple : *#duration(1,0,0,0)* car nous voulons avoir chaque date (d'un jour sur l'autre)

# Créer un calendrier sur M

## Génération de la liste de dates (5/6)

- ✓ **Dernières étapes de préparation :**
- ✓ **Clic-droit sur étape « EndDate »** et sélectionner « **Insérer l'étape après** »
- ✓ Dans le code, remplacer EndDate par la fonction List.Dates :  
`= List.Dates(StartDate,Duration.Days(EndDate-StartDate)+1,#duration(1,0,0,0))`
- ✓ *Une liste de dates est ainsi créée, avec toutes les dates entre la première date et la dernière date de vente.*
- ✓ *Afin de bien avoir la dernière date :*  
Rajouter +1 (après avoir calculer le Nombre de jours entre début et fin.



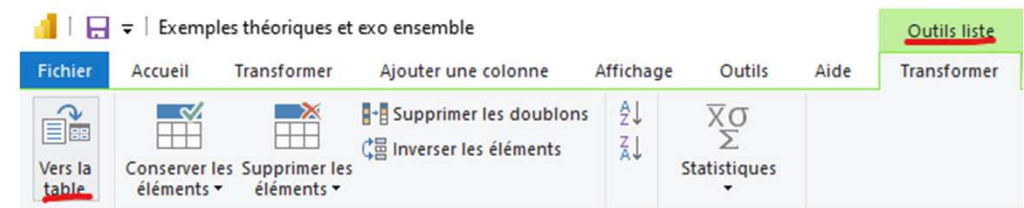
The screenshot shows a spreadsheet interface. At the top, a formula bar contains the following formula: `= List.Dates(StartDate,Duration.Days(#"EndDate dynamique"-StartDate)+1,#duration(1,0,0,0))`. Below the formula bar, a table with the title "Liste" is displayed. The table has two columns: an index column and a date column. The index column contains the numbers 1, 2, 3, and 4. The date column contains the dates 16/01/2017, 17/01/2017, 18/01/2017, and 19/01/2017.

Liste	
1	16/01/2017
2	17/01/2017
3	18/01/2017
4	19/01/2017

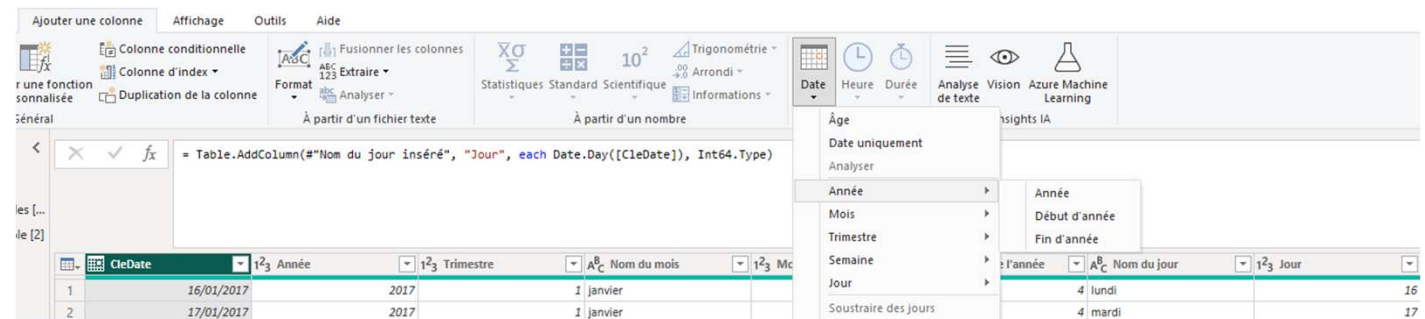
# Créer un calendrier sur M

## Génération table finale (6/6)

- ✓ **Transformer la liste en table :**
- ✓ Dans l'onglet **Outils de liste** > **Transformer**, cliquer sur **Convertir** > **Vers la table** puis dans la fenêtre, cliquez sur **OK**
- ✓ **Changer** le type de données
- ✓ **Renommer** en « CleDate »



- ✓ **Ajouter les colonnes d'attributs de date (année, mois, etc.) :**
- ✓ Dans l'onglet **Ajouter une colonne** > **Date** > **Année/Mois/Trimestre/Semaine/Jour**



# Annexes

Partie 3 : Rappels les outils de table et de colonnes + hiérarchie  
(éléments de formation débutant – niveau 1)

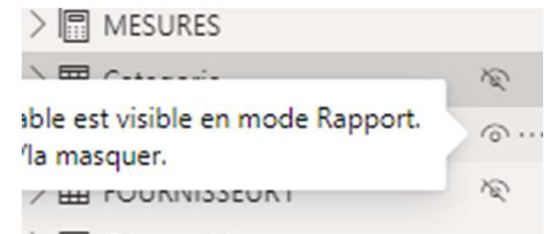
## Onglet Données/Rapport

### Masquer des éléments

Il est possible de **masquer une table et/ou colonne et/ou mesure** pour la vue Rapport.

Effectivement, on peut avoir besoin d'une table et d'autres éléments pour construire et optimiser notre jeu de données dans Power BI mais ne pas le rendre disponible, car dispensable, aux concepteurs de rapport.

Il suffit de **cliquer droit** sur la table/la colonne/la mesure dans la zone de données (à droite) puis sélectionner « **Masquer** » et/ou cliquer sur l'icône Œil, positionné à droite de chaque élément.



# Onglet Données/Rapport

## Les outils de colonne

En cliquant sur un champ et/ou mesure, dans la zone des données à droite, un nouvel onglet en haut apparaît : **“Outils de colonne”**

On peut

- **Gérer le format,**
- **Gérer le résumé et la catégorie de données,**
- **Effectuer des Trier Par.**

Fichier Accueil Aide Outils de table **Outils de colonne**

Catégorie Format Texte Résumé Ne pas résumer Trier par colonne

Texte \$ % Autom... Catégorie de données Aucune catégorie Groupes de données

Structure Mise en forme Propriétés Trier Groupes Relations Calculs

Idcategorie	Catégorie
1	Audio
2	TV et Video
3	Ordinateurs
4	Caméras et caméscopes
5	Téléphones portables

Données

Rechercher

MESURES

Catégorie

Catégorie

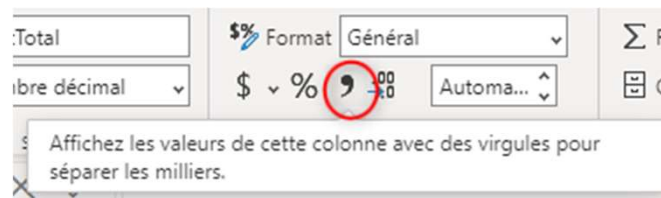
# Onglet Données/Rapport

## Format – Devise et séparateur de milliers

Dans la partie « Mise en forme », il est possible d'effectuer deux actions liées au format, possible uniquement sur Power BI :

- La devise :

- Le séparateur de milliers qui se cache derrière l'apostrophe, à droite du pourcentage :



# Onglet Données/Rapport

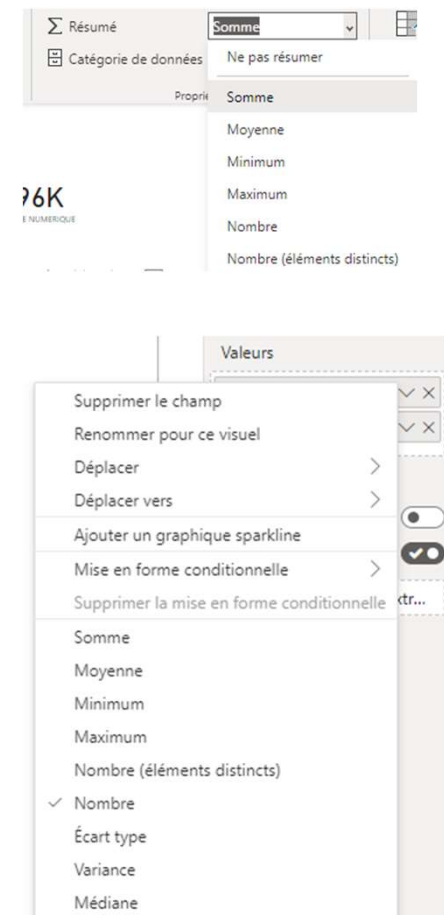
## Le résumé

Dans la partie « Propriétés », « Résumé » permet de stipuler l'utilisation par défaut d'une colonne dans tous les visuels :

- Si le champ **est de type « Texte »**, il sera en « Ne pas résumer » par défaut. En cochant le champ/la colonne, il sera toujours positionné automatiquement en « ligne » ou « axe des x » dans les visuels.
- Si le champ est de **type « Nombre entier »**, et semble être utilisé pour **relier des tables** entre elles, il sera considéré en « Ne pas résumer » également, car soupçonné d'être une clé primaire/secondaire. Si on le coche, il se positionnera automatiquement en ligne.
- Pour finir, si le champ est de **type « Nombre entier »** et ne semble pas être une clé, il aura un résumé « Somme ». En cochant le champ, il se positionnera toujours en « valeurs », et **ses valeurs seront sommées** dans les visuels. Tout ce qui est quantitatif sera par défaut sommer.

Si on souhaite sommer le champ dans les visuels, il ne faudra pas changer son utilisation par défaut.

Si on souhaite changer de temps en temps, l'opération que doit réaliser le visuel sur le champ en question, il faudra changer l'opération au niveau du visuel.





# Onglet Données/Rapport

## Catégoriser une donnée :

1. Cliquer sur un champ de type texte/caractère.
2. Dans l'onglet « **Outils de colonne** », cliquer sur **Catégorie des données**.
  - ✓ **Sélection de localité pour ville**
  - ✓ **Sélection de région pour les Etats ou les régions,**
  - ✓ **Etc.**

La catégorisation permet à l'outil de classier et **utiliser vos colonnes correctement** et de les associer aux tables de correspondances géographiques de Microsoft pour mieux les positionner sur les visuels géographiques.

idgeo	Continent	Etat ou région	Pays	Ville
17	North America	Washington	United States	Bellevue
18	North America	California	United States	Beverly Hills
19	North America	Montana	United States	Billings
20	North America	Mississippi	United States	Biloxi
21	North America	Alabama	United States	Birmingham
22	North America	South Carolina	United States	Bluffton
23	North America	Washington	United States	Bothell
24	North America	Utah	United States	Bountiful
25	North America	Massachusetts	United States	Braintree
26	North America	Missouri	United States	Branson
27	North America	Washington	United States	Bremerton
28	North America	California	United States	Burbank
29	North America	Ohio	United States	Burbank
30	North America	California	United States	Burlingame
31	North America	Georgia	United States	Byron
32	North America	Kentucky	United States	Campbellsville

# Onglet Données/Rapport

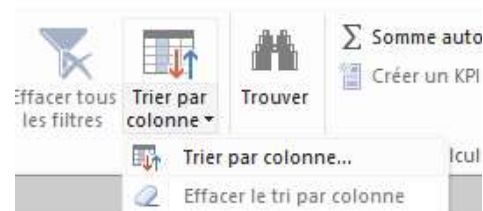
## Trier les mois

**Les mois ne sont jamais affichés de janvier à décembre sur les visuels car l'outil les trie par ordre alphabétique !**

Afin de contourner ce problème, il faut utiliser le TRIER PAR dans outils de Colonne :

- Cliquer sur le champ « Nom du mois » (janvier, février, etc.)
- Cliquer sur TRIER PAR COLONNE et sélectionner « Numéro de mois »

Power BI va trier le « Nom du mois » par le « numéro du mois »



# Onglet Données/Rapport

## Les hiérarchies : définition

Il y a deux notions de hiérarchies dans les données décisionnelles :

- **Les hiérarchies naturelles : Table date** ( Dans une année, il y a des trimestres. Dans les trimestres, il y a des mois. Dans les mois, il y a des semaines, etc...), **table géographie** (Dans un continent, il y a des pays, ensuite des Etats, ensuite des régions, puis des départements puis des villes.)
- Les autres hiérarchies propres à l'entreprise : Dans une catégorie de produit, il y a des sous-catégories, et pour chaque sous-catégories il y a des produits.

# Onglet Données/Rapport

## Les hiérarchies : à quoi ça sert ?

Pourquoi faire des hiérarchies ?

Les hiérarchies vont permettre **de choisir un axe d'analyse à plusieurs niveaux** (années, trimestres, mois par exemple) sans avoir à faire l'effort de définir le chemin d'analyse à chaque fois.

On crée en avance la hiérarchie dans le modèle de données et on aura plus qu'à la choisir dans un visual afin d'avoir toutes les dimensions de la hiérarchie.

**C'est donc un gain de temps et cela permet de développer un visuel sur différents niveaux de détails !**

# Onglet Données/Rapport

## Les hiérarchies dans un tableau

**Exemple** : Je sélectionne ma hiérarchie pour un tableau :



Je peux directement avoir accès à deux niveaux dans mon tableau croisé dynamique

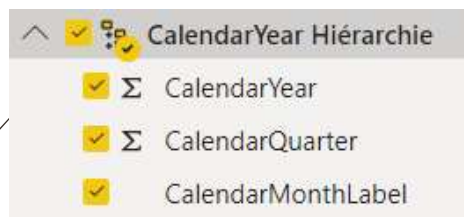


Catégories
<b>Audio</b>
Bluetooth
Headphones
MP4&MP3
Recording Pen
<b>Cameras and camcorders</b>
Camcorders
Cameras & Camcorders
Accessories
Digital
Cameras

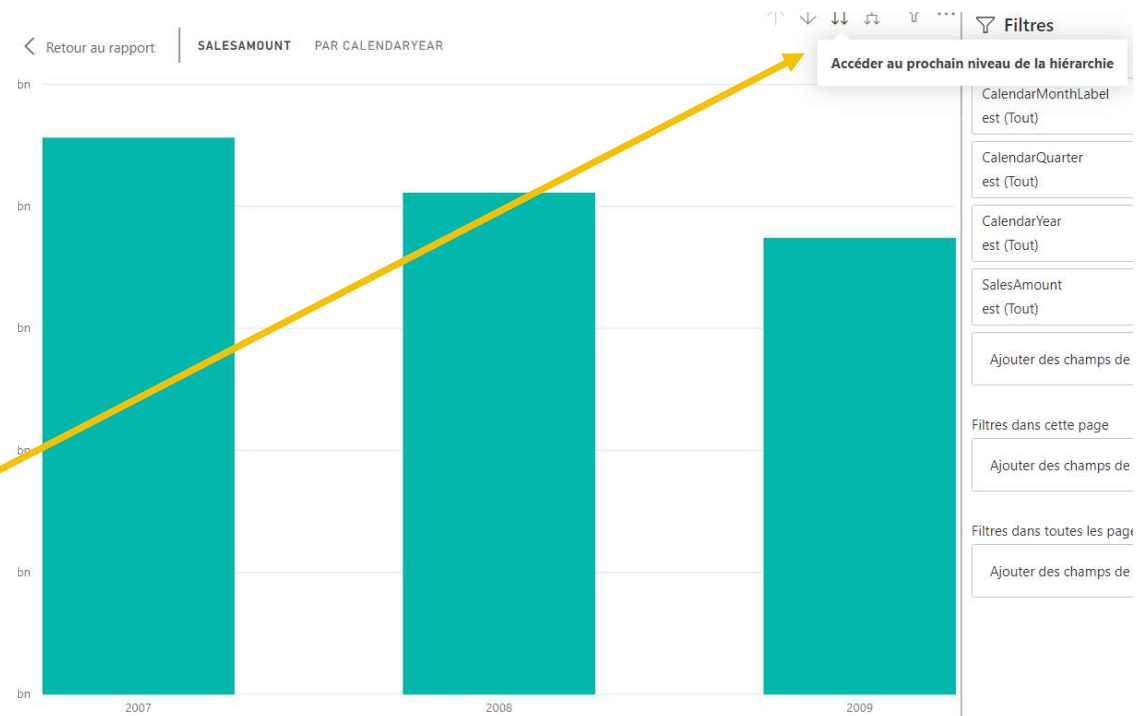
# Onglet Données/Rapport

## Les hiérarchies dans un graphique

**Exemple 2 :** Je sélectionne ma hiérarchie pour un graphique :

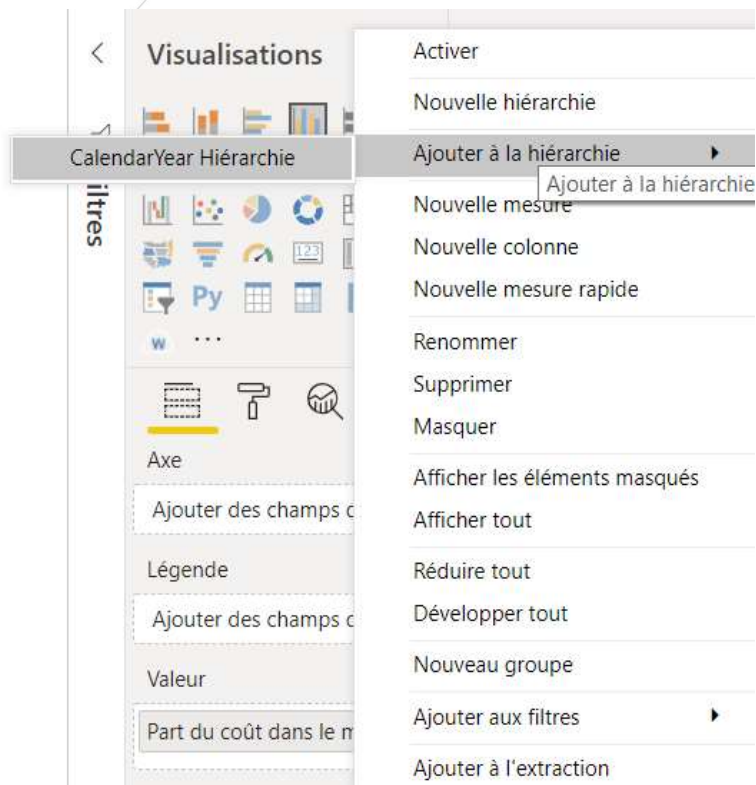


Les résultats vont être affichés sur le 1er niveau de Hiérarchie mais on va pouvoir descendre sur les niveaux, trimestre, mois.



# Onglet Données/Rapport

## Les hiérarchies : les créer !!!



Je clique sur le **1er champ que je souhaite mettre dans ma hiérarchie :**

- L'année par exemple et je clique droit puis **NOUVELLE HIERARCHIE**,
- La hiérarchie vient de se créer avec un seul champ : je pense à la renommer,

Je clique droit sur le **second champ** que je souhaite ajouter (TRIMESTRE par exemple) et je sélectionne **AJOUTER A LA HIERARCHIE**, puis choisir la hiérarchie dans laquelle l'incorporer.

Je peux ensuite **monter ou descendre** mes champs dans la hiérarchie.

## Liens utiles

- **Club Power BI sur Youtube** → conférences sur les bonnes pratiques et nouveautés Power BI.
- **Le blog du CFO masqué** → très bonnes explications sur les fonctions Dax et utilisation de Power Query.
- **Le guide du langage Dax** : <https://dax.guide/functions/aggregation/>
- **Fonctions Power Query M** : <https://docs.microsoft.com/fr-fr/powerquery-m/power-query-m-function-reference>
- **Forum Power Bi sur l'outil** (fenêtre jaune qui apparaît au démarrage) - Anglais
- **Autres sites en général** : anglais et français en tapant mots clés dans google. Exemples : « Dax SUM » ou « Power Query performances » ...



## Sources du support

- <https://docs.microsoft.com/fr-fr/power-bi/desktop-shape-and-combine-data>
- <https://docs.microsoft.com/fr-fr/power-bi/power-bi-visualization-slicers>
- <https://docs.microsoft.com/fr-fr/power-bi/power-bi-visualization-kpi>
- <https://docs.microsoft.com/fr-fr/power-bi/desktop-getting-started>
- <http://www.lecfomasque.com/power-bi-et-power-query-limportance-des-parametres-regionaux/>
- <https://docs.microsoft.com/fr-fr/power-bi/service-tips-and-tricks-for-color-formatting>
- <https://docs.microsoft.com/fr-fr/power-bi/sample-datasets#get-and-open-a-sample-content-pack-in-power-bi-service>