
SOMMAIRE GENERAL

1	AVANT DE COMMENCER	3
1.1	OBJECTIF DE LA FORMATION	3
1.2	DESCRIPTION DU JEU DE DONNEES UTILISE POUR LES CAS PRATIQUES	3
2	LES CONCEPTS DE BASE DE LA BUSINESS INTELLIGENCE	4
2.1	BUSINESS INTELLIGENCE ET ANALYSE DE DONNEES	4
2.2	FAITS (EVENEMENTS), MESURES (RESUMER), DIMENSIONS (FILTRE), GRANULARITE (GROUPE)	5
2.3	LA MODELISATION EN ETOILE	6
2.4	A EVITER : LE MODELE DIT DE LA « GRANDE TABLE A PLAT »	7
3	PRESENTATION DE POWER BI	8
3.1	LES ETAPES DE CREATION D'UN RAPPORT POWER BI	8
3.2	LE FONCTIONNEMENT INTERNE DE POWER BI DESKTOP	8
3.3	LES LANGAGES UTILISES DANS POWERBI	9
4	POWER QUERY : ACQUISITION ET TRANSFORMATION DES DONNEES	10
4.1	DESCRIPTION ET FINALITE DE POWER QUERY	10
4.2	PREPARER NOS DONNEES POUR CONSTRUIRE NOTRE MODELE POWER BI	15
4.3	AUTRES CAS CONCRETS D'UTILISATION DE POWER QUERY ET M	19
4.4	CHANGER DE NIVEAU DE GRANULARITE (REGROUPEMENT)	22
4.5	DEPENDANCES DES REQUETES	22
5	DAX : CALCULS ET RELATIONS	24
5.1	CREER QUATRE MESURES DE BASE	24
5.2	CREER UNE MESURE AVEC PLUSIEURS COLONNES D'UNE MEME TABLE : LES ITERATEURS	25
5.3	LES RELATIONS ENTRE LES TABLES AVEC RELATED	25
5.4	AFFINER ET FILTRER NOS CALCULS EN DAX	26
5.5	LES FONCTIONS DE CLASSEMENT ET DE PALMARES	37
5.6	CALENDRIER ET FONCTIONS D'INTELLIGENCE TEMPORELLE	42
5.7	CAS PARTICULIERS SUR LA MODELISATION	46
5.8	REGROUPER A TRAVERS DES DIMENSIONS AVEC SUMMARIZE (GROUP BY)	50
5.9	LA PROBLEMATIQUE EN-TETE / DETAIL : LE POIDS DE LA COMMANDE	52
5.10	UTILISER DES VARIABLES	52
5.11	MESURES ET COLONNES CALCULEES : DEUX NOTIONS DIFFERENTES ET COMPLEMENTAIRES	53
5.12	LA SECURITE AU NIVEAU DES LIGNES (RLS)	54
5.13	LES GROUPES DE CALCUL (AVEC TABULAR EDITOR2)	65
5.14	EXPORTER LES MESURES DE NOTRE MODELE	55
5.15	CREER DES MESURES A LA VOLEE DANS UN EDEITEUR (OUTIL DAX ET DEFINE)	55
5.16	CREER UN FICHIER MODELE (.PBIT)	55
5.17	AUTRES CAS D'USAGE	56
6	CREER LE RAPPORT	65
6.1	FINALISER LA MODELISATION AVANT DE COMMENCER	66
6.2	COMMENT CHOISIR LE VISUEL ADAPTE : LA MATRICE DE ZELAZNY	66
6.3	LES OPTIONS GENERALES D'UN RAPPORT	68
6.4	LES OPTIONS DANS LES VISUELS	68
6.5	SEGMENTS, FILTRES CACHES ET INTERACTIONS ENTRE LES VISUELS	69
6.6	EXPLORATION DES DONNEES	70
6.7	LES PARAMETRES DE CHAMPS	71

1 AVANT DE COMMENCER

1.1 Objectif de la formation

À l'issue de cette formation, vous serez en mesure de réaliser l'**acquisition** de vos données dans PowerQuery, de procéder à leur **nettoyage** et **transformation**. Vous saurez créer un **modèle en étoile**, comprendre la logique du **DAX**. Vous serez ainsi à même de produire des **tableaux de bord**, grâce à la compréhension du modèle sous-jacent et à l'application des bonnes pratiques. Vous saurez enfin publier vos rapports dans **Power BI Services**.

1.2 Description du jeu de données utilisé pour les cas pratiques

La formation est basée **sur plus de 90 cas pratiques** qui alterneront avec des messages forts en termes de bonnes pratiques et de compréhension des concepts et des formules. Elle se déroulera à travers **une étude de cas décrivant l'activité d'une société de négoce de vêtements répartie sur plusieurs continents**. Les différentes tables qui vont alimenter notre étude sont :

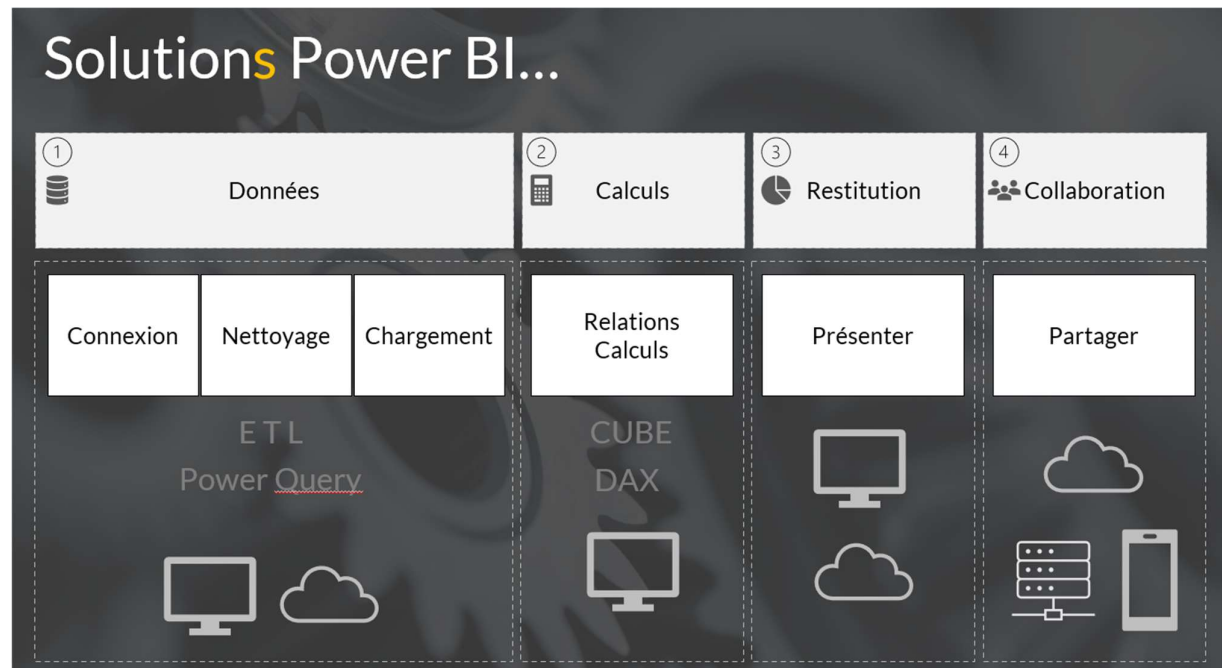
- La table des **entêtes de commandes** (6572 lignes)
- La sous-table **détail des commandes** (les lignes de chaque commande, 20246 lignes)
- La table **clients** (92 sociétés)
- La table **Zones** des clients (table interne à saisir, 4 régions)
- La table **articles** (78 articles)
- La table **famille articles** (8 familles)
- Le jeu de données s'étend du **19/04/2019** au **31/03/2024**

➔ **Merci de bien vouloir créer un dossier [FormationPBI](#) et d'y copier l'intégralité des éléments ci-dessous que vous venez de télécharger par le lien fourni par votre formateur. Vous y trouverez :**

- A la racine : les fichiers de données
- Un répertoire CommandeDossiers
- Un répertoire Supports Apprenants avec vos supports

3 PRESENTATION DE POWER BI

3.1 Les étapes de création d'un rapport Power BI



3.2 Le fonctionnement interne de Power BI Desktop

Selon les deux modes de connexion aux données (Direct Query ou Import), le chargement de données diffère. **VERTIPAQ** est la base interne de Power BI, **indexée** et **compressée**.

Les conséquences des deux modes de connexions ?

Pour faire très simple, on peut dire :

- En mode import tout se passe dans Power BI
- En mode Direct Query tout se passe dans le système source (SQL, Azure, etc.)
- Le mode Import est plus rapide, le mode DirectQuery est plus dynamique.

3.2.1 En mode import

- A. Connexion à la source** : Requête SQL, lecture fichier, API...
- B. Aperçu dans Power Query** : Les données sont chargées en RAM uniquement pour afficher l'aperçu.
- C. Fermer et charger** : Power Query renvoie les données transformées → VertiPaq les charge en RAM, les compresse et construit ses index.
- D. Enregistrer le PBIX** : Le modèle VertiPaq (compressé + métadonnées) est stocké sur le disque dans le fichier PBIX.
- E. Réouverture du PBIX** : Le fichier est relu et VertiPaq recharge le modèle en RAM.

4 POWER QUERY : ACQUISITION ET TRANSFORMATION DES DONNEES

4.1 Description et finalité de Power Query

4.1.1 A quoi sert Power Query ?

Power Query est l'outil de préparation des données : extraire, nettoyer et structurer, puis charger les données. C'est un ETL. (Extraction, Transform, Load)

Power Query va automatiser la préparation des données, à savoir essentiellement :

1. Réaliser une connexion avec vos données source
2. Modifier le type des données
3. Nettoyer les données des caractères invisibles
4. Supprimer des colonnes
5. Filtrer les lignes des tables
6. Remplacer des valeurs
7. Remplacer ou supprimer les données en erreur
8. Ajouter des colonnes personnalisées et conditionnelles et à partir d'exemples
9. Dépivoter ou pivoter des requêtes
10. Réaliser des ajouts de tables (ajout de requête)
11. Réaliser des jointures ou fusions de tables (fusion de requêtes)

4.1.2 Créer une première requête sur un fichier Excel

Onglet **Accueil** / nouvelle source et choisir le fichier Commandes.xlsx.

Le formateur vous guidera pour la première requête sur le fichier commandes.xlsx

Répéter cette action pour chaque fichier

2. TP : Créer la connexion aux tables commandes (en-têtes) , dans Power Query

Créer la requête Commandes sur la table **Commandes.xlsx**.

Remarques pratiques

On voit en bas à gauche : 7 colonnes, +999 lignes. On peut aussi compter les lignes dans transformer / compter les lignes.

*Le bouton **atteindre la colonne** permet de retrouver rapidement la colonne sans défiler car on obtient une liste verticale*

*Le Bouton **paramètres de la source de données** permet : de modifier les chemins aux sources de données utilisées, de gérer la **confidentialité** (Aucun, public, professionnel, privé), exporter les **fichiers de connexion (pbids)**.*

Si besoin, on peut copier la table entière (en haut à gauche)

4.1.3 Les requêtes : organisées en étapes

- A chaque opération réalisée sur une requête, Power Query inscrit la modification opérée dans la fenêtre **Etapes appliquées**.

4.1.8 Distribution et profil d'une colonne

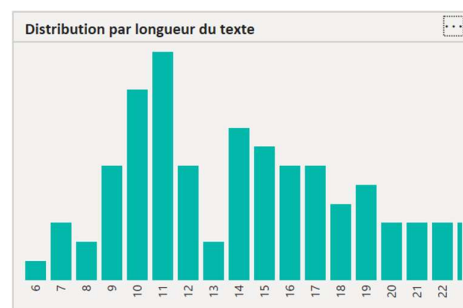
Attention : le profilage ne se fait par défaut que sur les 1000 premières lignes.

- Permet de donner les valeurs distinctes et uniques
- Ainsi que la distribution statistique (moyenne, nombre de valeurs, min, etc)
- On peut remplacer les valeurs à partir des barres de distribution
- NaN : not a number (ex 0/0)
- Dans le profil d'une colonne, pour les caractères alphanumériques, leurs min et max sont dans l'ordre alphabétique



Remarque

Pour les colonnes contenant du texte, on peut dans le profil de la colonne activer « regrouper par », et obtenir une **distribution par longueur de texte**, comme ci-dessous avec le nom de la société



4. TP : Profilage des données

Objectif : expliquer les **valeurs uniques, distinctes, valides, erreur, vide**

- Créer une requête pour chaque onglet du fichier « **qualité de la colonne.xlsx** »
- Activer la distribution et la qualité de la colonne
- Expliquer pour chaque requête (Qualite cas 1 et Qualite cas 2 les notions en objet

4.1.9 Qualité de la colonne et correction des erreurs

Les erreurs de type de données concernent surtout les fichiers plats. Les fichiers plats n'imposent aucun schéma : tout arrive comme du texte, avec souvent des valeurs

8. Supprimer une ligne contenant la valeur null dans l'IDCommande

Supprimez la ligne de la table Commandes dont le champs IDCommande n'est pas renseigné (Null).
Commencez par filtrer la table par ordre croissant d'IDCommande
On va utiliser la fonctionnalité supprimer les lignes du haut du groupe réduire les lignes (1 ligne)
On peut aussi utiliser la fonctionnalité supprimer les éléments vides dans la liste déroulante à droite du champ.

4.2.3 Créer une table directement dans Power Query

Si vous souhaitez créer une table dans Power BI (par exemple pour créer des tables de correspondance), sous l'onglet **Accueil** cliquez sur le bouton **Entrer des données**. Dans la boîte de dialogue renseignez le nom de la table, le nom des colonnes et les valeurs de la table. Dans le modèle que nous sommes en train de construire, cette table servira de table de correspondance.

9. Créer une table de donnée interne contenant les zones des clients

Choisir le bouton « entrer des données » du menu accueil de Power Query
Nommez cette requête **Zones**

IDZone	NomZone
1	EMEA
2	Nord Amérique
3	Asie
4	Sud Amérique

4.2.4 Fusionner des colonnes de plusieurs tables (requête fusion)

4.2.4.1 Rappels sur les différents types de jointures (fusion de requêtes)

Nous appellerons première requête la requête faisant allusion à la première table ou table de gauche, ou table précédente.



- **Interne (Inner)** : seules les données communes aux deux tables sont renvoyées.
- **Externe gauche (Left outer)** : les données de la première requête sont rapatriées avec les correspondances lorsqu'elles existent dans la deuxième requête, « null » sinon.
- **Externe droite (Right outer)** : les données de la deuxième requête sont rapatriées avec les correspondances lorsqu'elles existent dans la première requête, « null » sinon.
- **Externe entière (Full outer)** : toutes les lignes des deux requêtes sont rapatriées valeur « null » dans la table ne possédant pas l'article correspondant.

5 DAX : CALCULS ET RELATIONS

(DAX = **D**ata **A**nalysis **E**Xpressions)

Nous venons de transformer nos données, réalisé les fusions, des ajouts, dans PowerQuery. Nous allons créer une **mesure** en DAX dans le cas pratique ci-dessous et vérifier que nous obtenons tous le même résultat ☺, avant de continuer.

28. Créer un dossier des mesures et une page de rapport mesures

Suivre attentivement la manipulation du formateur

5.1 Créer quatre mesures de base et appliquer le bon format

29. SUM - On crée ici la mesure _CA (chiffre d'affaires)

Objectif

Calculer le chiffre d'affaires, le nombre de commandes et le montant moyen commandé par ligne et le stocker dans un objet appelé mesure, puis donner appliquer le bon formatage de la mesure (devise euro)

Conseil

Préfixer le nom des mesures (underscore par exemple), afin d'éviter les redondances avec une colonne calculée du même nom, et appeler les mesures directement.

```
_CA = SUM (DetailCommandes[MontantLigne])  
_NBCommandes = DISTINCTCOUNT (DetailCommandes[IDCommande])  
_NBLignesCommandes = COUNTROWS (DetailCommandes)  
_MontantMoyenLigne = AVERAGE (DetailCommandes[MontantLigne])
```

Remarque : la fonction d'agrégation COUNT (DetailCommandes[IDCommande]) aurait produit le même résultat si IDCommande ne contient pas de valeurs nulles

30. Créer la mesure _Poids et la formater en kg.

Objectif

Appliquer un format spécifique

On utilisera le format dynamique en appliquant cette expression : "#,### kg"

```
Poids = SUM ((DetailCommandes[PoidsCargaison])
```

36. Calculer le nombre de commandes pour les clients dont le CA>1M

Calculer le nombre de commandes pour les clients dont le CA>1M

On doit trouver 1340 commandes

Explication de la formule

1. FILTER retourne la table des clients dont le CA est > 1M
2. CALCULATE recalcule le nombre de commandes pour ces clients filtrés

_NBCommandesClientsCA>1M = CALCULATE([_NBCommandes],FILTER(Clients,[_CA]>1000000))

NomSociete	_NBCommandesClientsCA>1M
Boleros	383
Grunewald	285
Th Fashing	336
The Corner Store	336
Total	1340

Bonus ! Nombre de clients dont le CA est > 1000 000€

On appellera cette mesure _NBClientsActifsCA>1M

37. TP – Plusieurs filtres « ET » : CA>1M ET zone EMEA

Calculer le nombre de commandes réalisé par les clients dont le CA>1M et de la Zone EMEA

On doit trouver 1004 commandes

Explication de la formule

3. FILTER retourne la table des clients dont le CA est > 1M et zone = « EMEA »
4. CALCULATE recalcule le nombre de commandes pour ces clients filtrés

_NBCommandesClientsCA>1M_Et_ZoneEMEA =

CALCULATE (
[_NBCommandes],
FILTER (Clients, [_CA] > 1000000 && Clients[NomZone] = "EMEA")
)

NomSociete	_NBCommandesClientsCA>1M_Et_ZoneEMEA
Boleros	383
Grunewald	285
Th Fashing	336
Total	1004

Remarque ! : pour la condition ou (inclusif) il faut utiliser le double « pipe » ||

5.5.1 Attribuer un rang dynamique avec RANKX

50. Créer une mesure pour classer les clients en fonction du CA avec RANKX

Objectif

Obtenir un rang de classement dynamique des clients en fonction de leur chiffre d'affaires ([_CA])

Explication de la mesure _Classement_NomSociete_CA

1. ALL(Clients) : Supprime tous les filtres de la table Clients, incluant tous les clients dans le classement, même ceux non visibles dans le contexte actuel de filtre.
2. [_CA] : c'est sur cette base que le classement est établi.
3. DESC : Indique que le classement doit être fait en ordre décroissant, donc le client avec le plus haut CA recevra le rang le plus bas (c'est-à-dire, rang 1).
4. Dense : Spécifie le mode de classement. Contrairement au mode par défaut qui pourrait sauter des rangs en cas d'égalité (par exemple, passer de 1 directement à 3 si deux clients sont à égalité pour la première place), le mode Dense assure que les rangs suivent de manière continue sans sauter de nombres après une égalité.

_Classement_Clients_CA_Dynamique =
RANKX(ALL(Clients) , [_CA],,DESC,Dense)

NomSociete	_CA	_Classement_Clients_CA_Dynamique
Boleros	1 440 794,66 €	1
Th Fashing	1 252 341,31 €	2
The Corner Store	1 149 422,50 €	3
Grunewald	1 084 000,79 €	4
Eintrach GS	990 383,32 €	5
Rode & Vite	799 566,14 €	6
Warp AG	784 316,78 €	7
Don Balón	690 189,08 €	8
Boombastic	670 648,60 €	9
Champes	642 068,98 €	10

51. Classement par client et par zone

Objectif

Modifier la mesure **_Classement_Clients_CA_Dynamique** afin d'obtenir un rang de classement des pays des clients en fonction de leur chiffre d'affaires ([_CA], à l'intérieur de chaque zone. (C'est-à-dire que pour chaque zone nous aurons un classement qui commence par 1)

NomZone	_CA	Classement
Nord Amerique		
The Corner Store	1 149 422,50 €	1
Bond Ltd	579 581,38 €	2
Davenport Fashion	375 154,54 €	3
Millenium	353 522,33 €	4
Nirvana Stores	340 397,88 €	5
Too Hot 4U	203 078,43 €	6
Big Foot Shoes	180 363,49 €	7
Macalena	138 700,68 €	8

66. Calculer le montant moyen des commandes avec SUMMARIZE

Objectif

Déterminer la moyenne des montants des commandes, où le montant de chaque commande est la somme totale de toutes ses lignes.

Explication de la mesure

Cette mesure calcule le montant moyen des commandes en suivant deux étapes principales :

1. Regroupement par commande : SUMMARIZE pour regrouper les lignes de détail des commandes (DetailCommandes) par IDCommande. Pour chaque commande, elle calcule le montant total de la commande en sommant (SUM) les montants de toutes ses lignes (DetailCommandes[MontantLigne]). Le résultat est une nouvelle table temporaire contenant chaque IDCommande et son MontantCommande correspondant.
2. Calcul de la moyenne : Applique ensuite AVERAGEX sur cette table temporaire pour calculer la moyenne des montants des commandes. AVERAGEX parcourt chaque commande dans la table temporaire et calcule la moyenne de [MontantCommande], qui est le montant total de chaque commande.

_Montant_Moyen_Commande =

AVERAGEX (

SUMMARIZE (

DetailCommandes,

DetailCommandes[IDCommande],

"MontantCommande", SUM (DetailCommandes[MontantLigne])

),

[MontantCommande]

)

NomSociete	_Montant_Moyen_Commande	_Montant_Moyen_Ligne
TTT-The Ticky Tie	6 972 €	1 877 €
Fawtly Towers	5 949 €	1 983 €
Alles Lusekofter	5 508 €	1 916 €
Leningrad Cowboys Shop	5 002 €	1 213 €
Eye Fashion	4 970 €	1 420 €
Pulp Toxedos	4 909 €	1 403 €

5.16.5 Le calcul des moyennes mobiles

Une moyenne mobile, c'est comme une fenêtre glissante qui parcourt les données et, à chaque pas, calcule la moyenne des valeurs dans cette fenêtre. Elle lisse les variations à court terme pour révéler la **tendance** à long terme des données. C'est un outil clé pour voir au-delà des fluctuations quotidiennes et comprendre la direction générale des données.

76. Calculer une moyenne mobile

Objectif

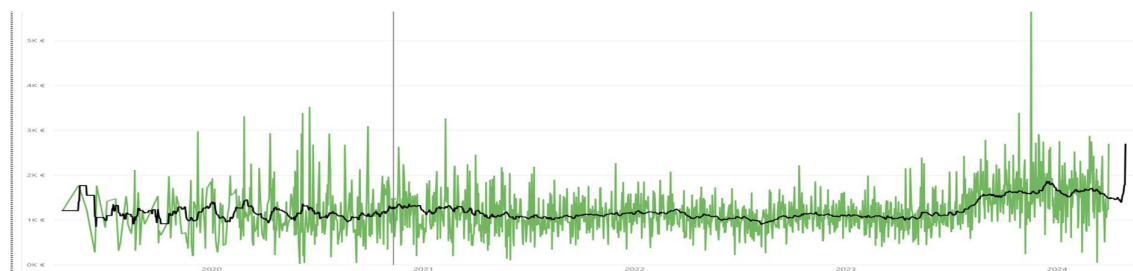
Calculer une moyenne mobile des ventes quotidiennes sur 30 jours

Explication de la mesure

1. DATESINPERIOD : retourne une table contenant une colonne de dates qui se trouvent dans une période donnée.
2. Calendrier[Date] : Désigne la colonne de dates dans une table de calendrier
3. LASTDATE(Calendar[Date]) : Renvoie la dernière date dans le contexte actuel de la colonne de dates du calendrier. Cela permet de déterminer le point de fin pour le calcul de la période.
4. -30, DAY : Spécifie la période à calculer comme étant les 30 derniers jours précédant la dernière date du contexte actuel.

_MoyenneMobile30J =

```
CALCULATE(  
  AVERAGE(DetailCommandes[MontantLigne]),  
  DATESINPERIOD(  
    Calendrier[Date],  
    LASTDATE(Calendar[Date]),  
    -30,  
    DAY  
  )  
)
```



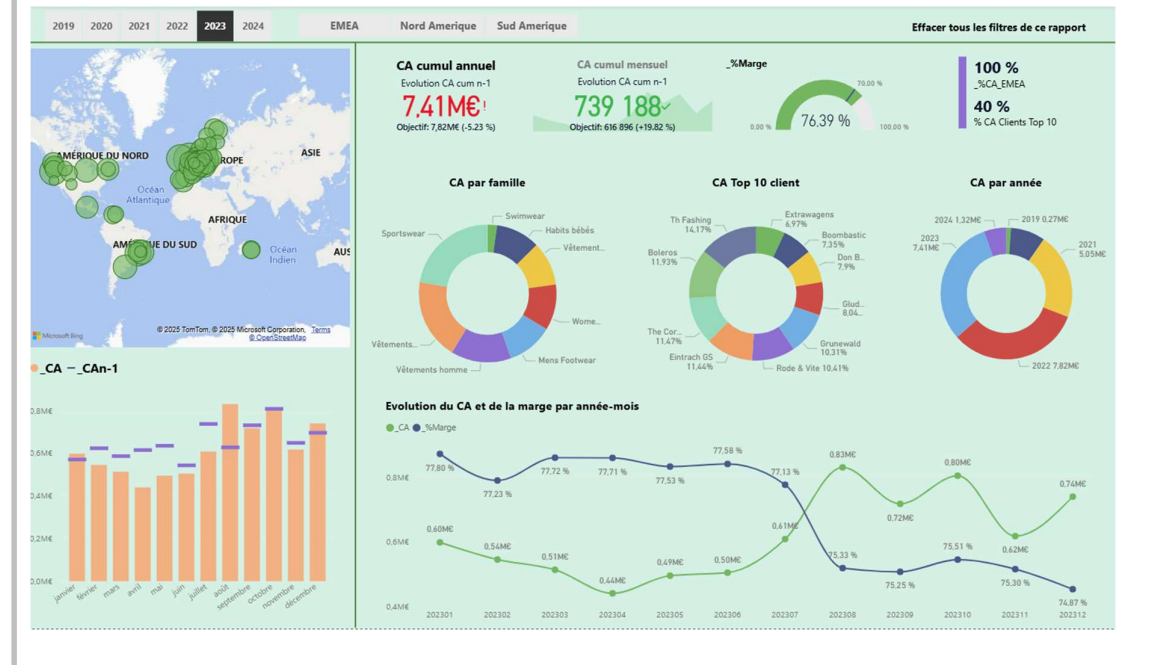
5.16.6 La contribution : la loi des 80/20

La loi des 80/20, connue sous le nom de principe de Pareto, est un concept économique qui stipule qu'environ **80% des effets proviennent de 20% des causes**. C'est constater, par exemple que 80% des ventes proviennent de 20% des clients. C'est une observation empirique plutôt qu'une règle stricte, mais elle est fréquemment utilisée pour souligner l'importance qu'il y a de se concentrer sur les segments les plus productifs ou rentables d'une activité.

6.9 Cas de synthèse n°1 : le tableau de bord de notre modèle

88. Cas pratique de synthèse n°1

A partir du modèle de données réalisé jusque là, nous allons créer la page de rapport ci-dessous. Toutes les consignes des visualisations seront données par votre formateur 😊



SUMX

SUMX(<table>, <expression>)

Retourne la somme d'une expression évaluée pour chaque ligne d'une table.

AVERAGEX

AVERAGEX(<table>, <expression>)

Calcule la moyenne (arithmétique) d'un ensemble d'expressions évaluées sur une table.

RANKX

RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])

Retourne le classement d'un nombre dans une liste de nombres pour chaque ligne de l'argument table.

CONCATENATEX

CONCATENATEX(<table>, <expression>[, <delimiter> [, <orderBy_expression> [, <order>]]...])

Concatène le résultat d'une expression évaluée pour chaque ligne d'une table.

9.2.2 Fonctions de filtrage

CALCULATE

CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])

Évalue une expression dans un contexte de filtre modifié.

CALCULATETABLE opère exactement de la même façon, sauf qu'elle modifie le contexte de filtre appliqué à une expression qui retourne un objet de table.

ALL

ALL([<table> | <column>[, <column>[, <column>[, ...]]]])

Retourne toutes les lignes d'une table, ou toutes les valeurs d'une colonne, en ignorant les filtres qui ont été éventuellement appliqués. Cette fonction est utile pour effacer les filtres et créer des calculs sur toutes les lignes d'une table.

ALLEXCEPT

ALLEXCEPT(<table>, <column>[, <column>[, ...]])

Supprime tous les filtres de contexte de la table, à l'exception de ceux qui ont été appliqués aux colonnes spécifiées.

ALLSELECTED

ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[, ...]]]])

Supprime les filtres de contexte des colonnes et des lignes dans la requête actuelle, tout en conservant tous les autres filtres de contexte ou filtres explicites.

FILTER

FILTER(<table>, <filter>)

Retourne une table qui représente un sous-ensemble d'une autre table ou expression.