

1 Avant de commencer

1.1 Objectif de la formation

À l'issue de la 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** et de la fonction **CALCULATE**. Vous serez ainsi à même de produire des **rapports efficaces et justes**, grâce à la compréhension du modèle sous-jacent et à l'application des bonnes pratiques. Vous saurez enfin partager vos rapports dans **Power BI Services**.

1.2 Outils et méthode pédagogiques

La formation se déroulera à travers **une étude de cas décrivant l'activité d'une société de négoce de vêtements**. Les différentes données qui vont alimenter notre étude, et que vous trouverez dans le **dossier DataFiles**.

La formation est basée **sur 60 cas pratiques**, qui alterneront avec des messages forts en termes de bonnes pratiques et de compréhension des concepts.

.. / ..

2 LES CONCEPTS DE BASE EN BI

2.1 Faits, mesures, agrégations

Fait, mesure, agrégat

1. L'activité est caractérisée par des **faits** qui se produisent (événements).
2. Les faits peuvent être faits **quantifiés** par des valeurs numériques (prix, quantité, montant)
3. Les faits peuvent être **non quantifiés** : survenance d'une panne informatique.
4. Les faits sont résumés par une **fonction d'agrégat** appelée mesure (somme des ventes)
5. Pour les faits non quantifiés l'agrégation sera le comptage des faits (nombre de pannes).

Les différents types de mesure

1. **Les mesures additives** : l'agrégation peut être réalisée selon n'importe quelle dimension. (montant de vente, quantité commandée, etc.)
2. **Les mesures semi-additives** : l'agrégation ne peut être réalisée sur l'axe temporel (snapshot).
3. **Les mesures non-additives** : l'agrégation n'est pas possible (ratios)

.. / ..

3 POWER QUERY : ACQUISITION ET TRANSFORMATION DES DONNEES

3.1.1 A quoi sert Power Query ?

Power Query est l'outil de préparation des données : extraire, nettoyer et structurer les données. **En termes de modélisation, Power Query établit la structure initiale tandis que DAX enrichit cette structure avec des mesures et des relations.**

La finalité de ce chapitre 3 sera d'apprendre à travailler avec Power Query, mais aussi de construire notre modèle de données. De ce fait, les cas pratiques s'enchaînent et sont donc interdépendants !

PowerQuery va permettre différentes manipulations :

1. Réaliser une connexion avec vos données source
2. Modifier le type des données
3. Supprimer des lignes ou des colonnes
4. Remplacer des valeurs et gérer les données en erreur
5. Ajouter des colonnes personnalisées et conditionnelles
6. Dépivoter des tables
7. Concaténer des tables (ajout de requête)
8. De réaliser des jointures (fusion de requêtes)

3.1.2 Editeur avancé (M)

Vous avez la possibilité de visualiser le langage M généré, associé à votre requête. Sous l'onglet **Affichage**, cliquez sur le bouton **Editeur avancé**.

SalariesAvecErreurs

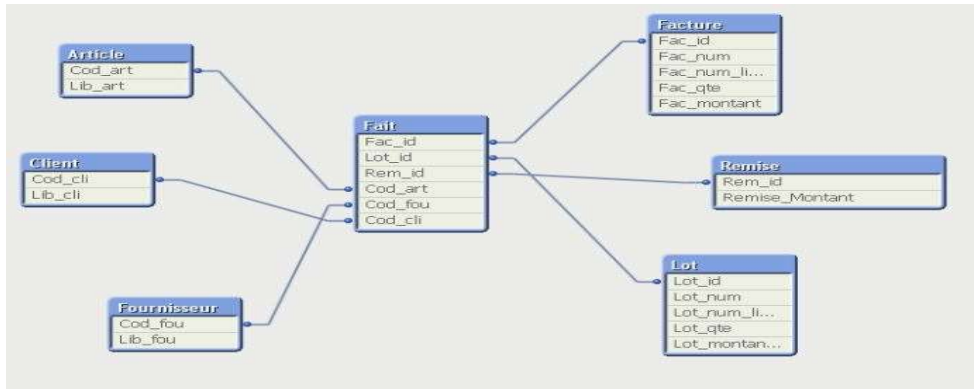
```
let
    Source = Excel.Workbook(File.Contents("C:\Users\Philippe\philippe sarfati\Ressources - Formation\Power BI\Cours Power BI\MonCours\DataFiles\Sale\SalarisAvecErreurs_Sheet", [Item="SalariesAvecErreurs", Kind="Sheet"])[Data],
    #"En-têtes promus" = Table.PromoteHeaders(SalarisAvecErreurs_Sheet, [PromoteAllScalars=true]),
    #"Type modifié" = Table.TransformColumnTypes(#"En-têtes promus",{{"IDSalarie", Int64.Type}, {"Prenom", type text}, {"Nom", type text}, {"Sexe", type text}, {"DateNaissance", #date(1900, 1, 1)}, {"DateDebut", #date(1900, 1, 1)}}),
    #"Lignes filtrées" = Table.SelectRows(#"Type modifié", each true),
    #"Type modifié1" = Table.TransformColumnTypes(#"Lignes filtrées",{{"IDBureau", Int64.Type}},
    #"Erreurs remplacées" = Table.ReplaceErrorValues(#"Type modifié1", {{"DateNaissance", #date(1900, 1, 1)}}),
    #"Erreurs remplacées1" = Table.ReplaceErrorValues(#"Erreurs remplacées", {{"DateDebut", #date(1900, 1, 1)}})
in
    #"Erreurs remplacées1"
```

3.1.3 Rappels sur l'importance d'une modélisation en étoile

C'est l'état de l'art en modélisation BI. Votre objectif doit consister à créer un modèle de données permettant de manipuler les données de façon efficace. Autrement dit :

Une table centrale répertoriant les faits, au centre du modèle, qui contient les clés étrangères (lien avec les dimensions) et des champs utilisés pour calculer les mesures (nombre d'unités, montants des ventes, montants budgétaires, etc.)

Des tables environnantes contenant les dimensions assorties de tous leurs attributs (produits, clients, catégories, calendrier, fournisseurs, etc.).



Modèle en étoile

.. / ..

3.2 Les opérations de bases avec Power Query

3.2.1 Fusionner des requêtes (jointures = joindre des colonnes de plusieurs tables)

Une opération de requête de fusion joint deux tables existantes en fonction des valeurs correspondantes d'une ou plusieurs colonnes. Vous pouvez choisir d'utiliser différents types de jointures, en fonction de la sortie souhaitée.

Onglet Accueil, Fusionner des requêtes (jointure).

Lors de la fusion de deux requêtes, l'on pourra :

- soit développer les colonnes
- soit agréger selon une fonction d'agrégation.

Rechercher les colonnes à développer

Développer Agréger

(Sélectionner toutes les colonnes)
 IDFamille
 NomFamille
 DesignationFamille

Utiliser le nom de la colonne d'origine comme préfixe

OK Annuler

Rechercher les colonnes à agréger

Développer Agréger

(Sélectionner toutes les colonnes)
 Σ Somme de IDFamille
 # Nombre de NomFamille
 # Nombre de DesignationFamille

Utiliser le nom de la colonne d'origine comme préfixe

OK Annuler

3.2.1.1 Rappels sur les différents types de jointures

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



.. / ..

4 DAX : CALCULS ET RELATIONS

4.1 Introduction au DAX

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

Comment lire l'aide des fonctions DAX ?

- | OU logique : les symboles qui se trouvent de l'un ou l'autre côté peuvent être utilisés.
- < > Chevrons : éléments **obligatoires**
- [] Crochets : éléments **facultatifs**.

4.1.1 Les mesures en DAX

Une mesure en DAX est une expression calculée à partir des colonnes (champs) de notre modèle, et qui **n'est pas matérialisée par une colonne**. Une mesure calcule un agrégat (Sum, Count, etc) et renvoie donc une valeur **unique**.

.. / ..

4.2 Modifier les contextes avec les fonctions ALL

La fonction CALCULATE est une fonction centrale dans Power BI

- Dans le cas 26 elle a permis de réaliser **une transition de contexte**
- Nous avons vu également qu'elle a permis avec la fonction CROSSFILTER d'activer le temps d'une mesure une relations bidirectionnelle entre deux table afin de rendre la propagation du filtre possible dans le sens plusieurs à un.
- On va cette fois-ci observer son rôle dans la modification des interactions avec les filtres internes et externes, par **le remplacement, l'ajout et la suppression des contextes**, associés aux modificateurs ALL, ALLSELECTED et ALLEXCEPT.

4.2.1 Les ratios avec ALL

Le modificateur ALL permet **d'effacer tous les filtres** (filtres externes et filtres internes)

- ALL(table) efface tous les filtres portés sur n'importe quel champ de la table
- ALL(Table[Colonne1]) efface seulement les filtres sur la colonne1
- ALL() sans arguments efface les filtres sur toutes les tables

Créer la mesure suivante, et la table ci-dessous

```
_Ratio_une_colonne = DIVIDE([_CA],CALCULATE ( [_CA], ALL ( Articles[NomFamille] ) ))
```

RATIO UNE COLONNE

NomFamille	_CA	_Ratio_une_colonne
Habits bébés	2 244 148 €	9,39 %
Mens Footwear	2 710 356 €	11,34 %
Sportswear	5 931 321 €	24,83 %
Swimwear	453 065 €	1,90 %
Vêtements enfants	2 150 149 €	9,00 %
Vêtements femme	4 573 914 €	19,14 %
Vêtements homme	3 117 496 €	13,05 %
Womens Footwear	2 710 647 €	11,35 %
Total	23 891 097 €	100,00 %

.. / ..

4.3 Changer de niveau de granularité

Nous allons utiliser la fonction SUMMARIZE, afin de déterminer **le panier de commande moyen**.

Syntaxe de SUMMARIZE

Retourne une table récapitulative des totaux demandés sur un ensemble de groupes.

SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>[, <expression>]...)

1er. TP - Changement de granularité avec SUMMARIZE

Nos analyses sont réalisées sur des agrégations du champs **MontantLigne**. Ce niveau de détail, appelé granularité peut être modifié par la création d'une fonction d'agrégation utilisant une table virtuelle pour déterminer le nouveau niveau d'agrégation. Nous souhaitons un niveau de granularité basé sur le montant total de la commande (somme des lignes de commande).

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 €

Créer pour cela la mesure suivante

```
_Montant_Moyen_Commande =  
AVERAGEX (  
    SUMMARIZE (  
        DetailCommandes,  
        DetailCommandes[IDCommande],  
        "MontantCommande", SUM ( DetailCommandes[MontantLigne] )  
    ),  
    [MontantCommande]  
)
```

Remarque

On peut aussi utiliser la mesure suivante

```
NBLignesCommandesClients = COUNTROWS(RELATEDTABLE(DetailCommandes))
```

4.4 Les fonctions DAX de relation entre les tables

4.4.1 Fonction RELATED

Cette fonction peut être comparée à la fonction RechercheV ou RechercheX dans Excel. **Cette fonction ayant besoin d'un contexte de ligne**, elle peut uniquement être utilisée dans une expression de colonne calculée, ou comme fonction imbriquée dans une expression qui utilise une fonction d'analyse de table.

.. / ..

5 CREER LE RAPPORT

Un **rapport Power BI** est un affichage sous plusieurs angles d'un jeu de données, comportant des visualisations de différentes observations et informations.

Un rapport peut comprendre une seule visualisation ou des pages remplies de visualisations.

5.1 La gestion dynamique des titres

On peut gérer les titres dynamiques avec la fonction SELECTEDVALUE.

2e. TP Cas pratique selection

```
"CA pour l'année " & SELECTEDVALUE ( Calendrier[Annee] ) & " "
& SELECTEDVALUE ( PRESTATIONS[Famille] )
```

Annee Selectionnee =

```
VAR Annee = CONCATENATEX(ALLSELECTED(Calendrier[Annee]),Calendrier[Annee]," ")
```

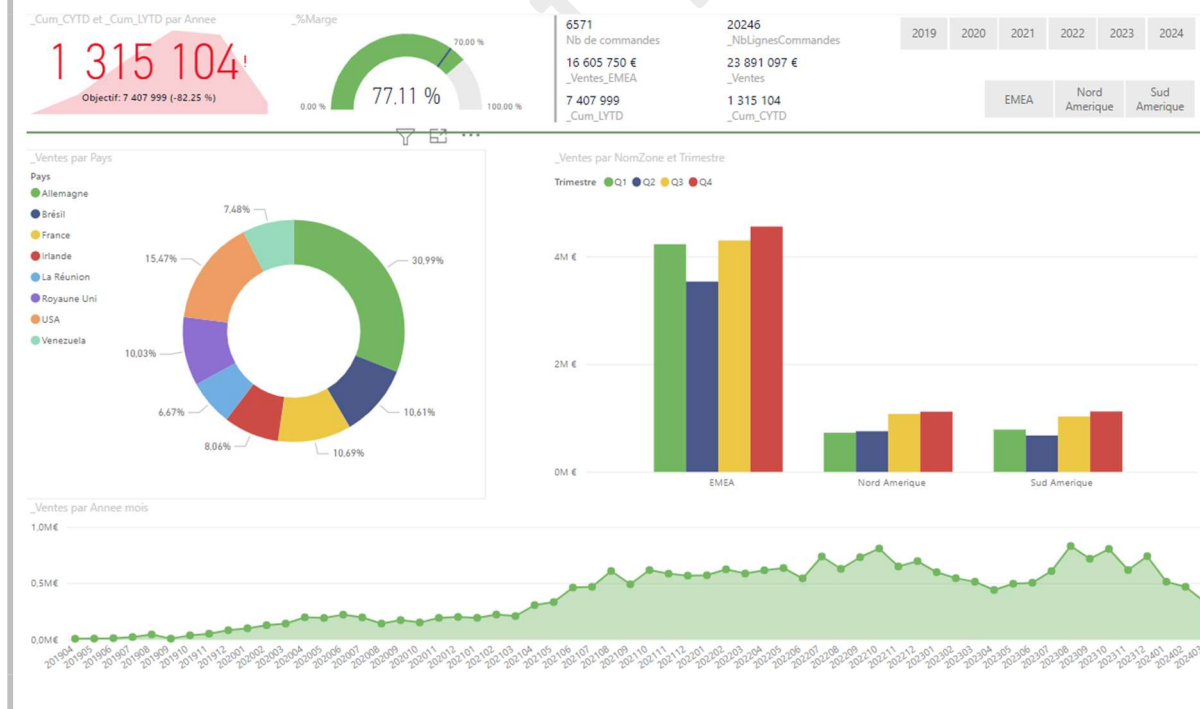
```
RETURN
```

```
IF(
    ISFILTERED(Calendrier[Annee]),
    "Année : "&Annee,
    "Années : Toutes"
)
```

.. / ..

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

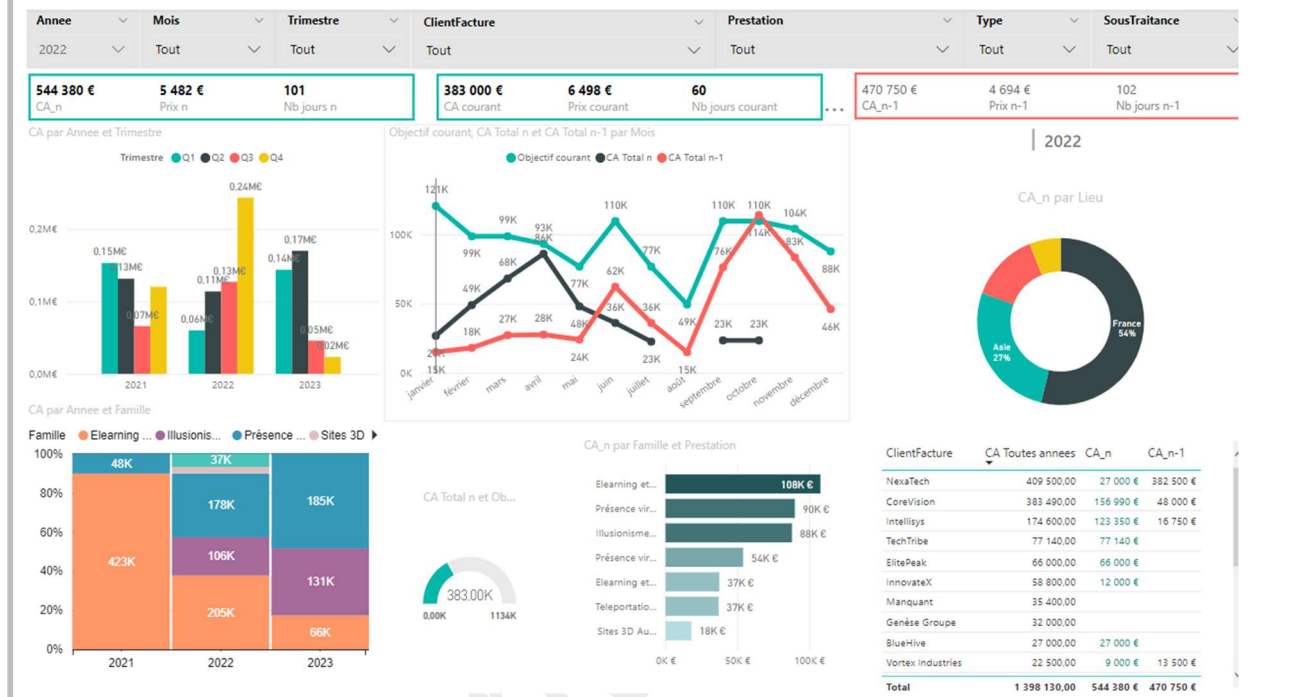
A partir du modèle de données réalisé jusque là, nous allons créer la page de rapport ci-dessous. Toutes les consignes des visualisations seront données par votre formateur 😊



5.3 Cas de synthèse n°3 : construire un tableau de bord avancé

3e. Cas de synthèse n° 3

Pour réaliser cette étude de cas, nous allons utiliser le modèle de données créé dans le cas pratique précédent et nous allons créer de nouvelles mesures pour alimenter les visualisations de ce nouveau tableau de bord. Nous observerons ici les interactions données dans la énoncé pour reproduire les réactions attendues lors des sélections. Comme dans les cas pratiques précédents toutes les consignes vous seront données par votre formateur 😊



6 Annexes

6.1 Glossaire des fonctions

La lecture de l'aide des fonctions DAX

Le formalisme suivant permet de comprendre l'aide sans ambiguïté :

- | **OU** logique : les symboles qui se trouvent de l'un ou l'autre côté peuvent être utilisés.
- < > Chevrons : éléments **obligatoires**
- [] Crochets : éléments **facultatifs**.

ADDCOLUMNS

ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)

Ajoute des colonnes calculées à la table ou à l'expression de table donnée.

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.

La fonction ALLSELECTED obtient le contexte qui représente toutes les lignes et colonnes dans la requête, tout en conservant les contextes et les filtres explicites autres que les filtres de lignes et de colonnes. Cette fonction peut être utilisée pour obtenir des valeurs totales affichées dans des requêtes.

AVERAGEX

AVERAGEX(<table>,<expression>)

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

BLANK

Retourne un espace.

BLANK()



6.2 Index Dax

A	P
ALL 28, 29, 30, 67	PARALLELPERIOD 41
ALLEXCEPT 29, 30	
ALLSELECTED 29, 30, 31	R
AVERAGEX 5, 31, 67, 68, 69	RELATED 36
	RELATEDTABLE 35, 37
C	
CALCULATE 26, 27, 28, 29, 30, 31, 66, 67, 68	S
CONCATENATEX 34, 67	SAMEPERIODLASTYEAR 41
COUNT 5, 23, 63	SUM 5, 23, 26, 32, 63
COUNTRROWS 35, 37, 63	SUMMARIZE 35, 67
CROSSFILTER 28, 29	SUMX 5, 31, 32, 37
	SWITCH 24, 63
D	T
DATESINPERIOD 69	TOPN 34, 67, 68
DATESYTD 41	TRUE 24, 62
DISTINCTCOUNT 5, 23, 63	
DIVIDE 30, 31, 67, 68	U
	USERRELATIONSHIP 40
F	V
FILTER 67	VAR 63, 64, 68, 69
	Y
I	YEAR 68
ISBLANK 68	
ISINSCOPE 68, 69	
M	
MONTH 68, 69	