Power Bi

DAX

Introduction

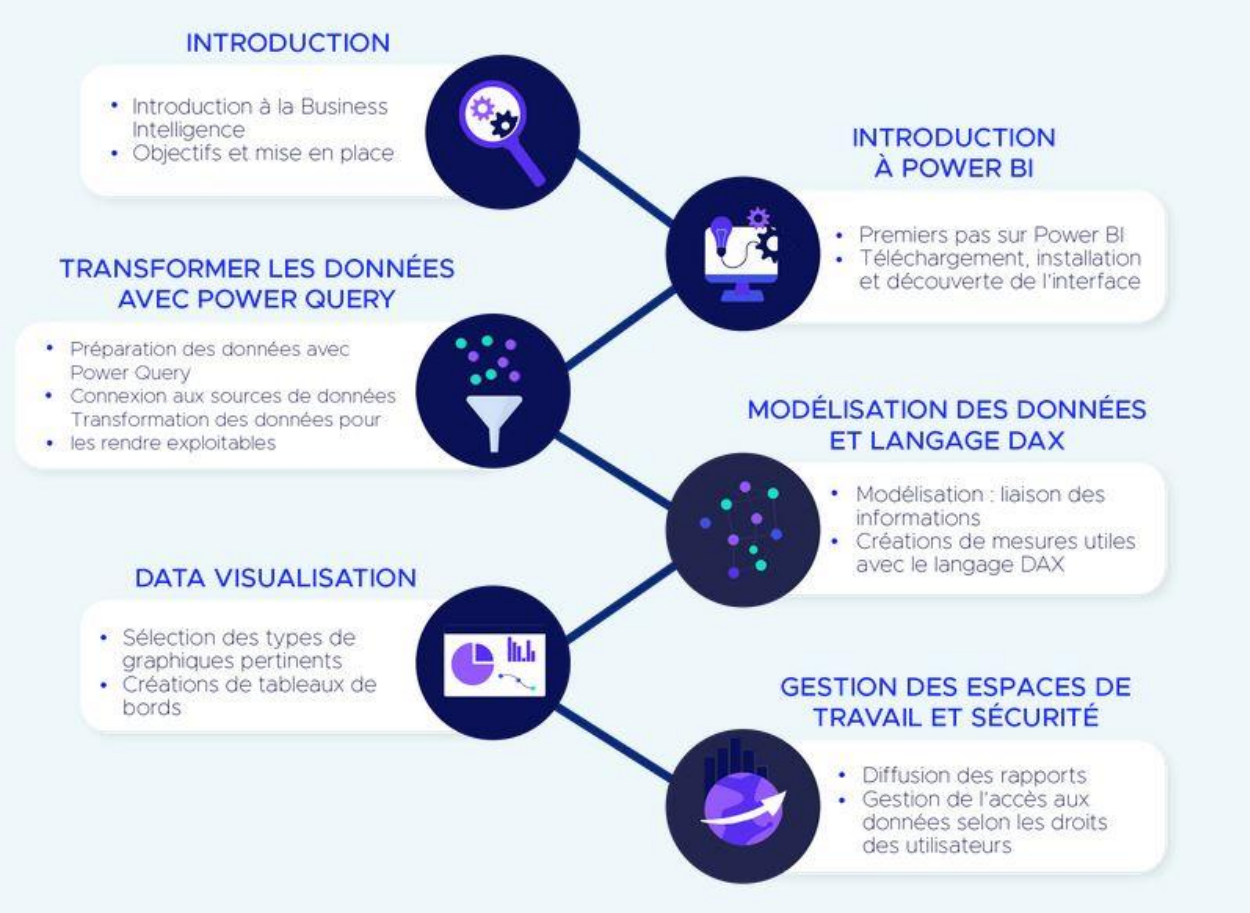


Table des matières

Introduction

- Un modèle de données avec power BI = le développement de **rapport visuel**

- Etapes:
  - Se connecter à des données
  - Transformer et préparer données
  - Définir la « logique métier » en ajoutant des calculs DAX (Data Analysis Expressions)
  - Publier le modèle sur Power BI

Introduction

- Il existe **plusieurs façons** de développer un modèle.

- Objectif d'un modèle le plus optimal:
  - réduire le temps d'actualisation des données
  - réduire l'utilisation des ressources (la mémoire et le processeur)

- Exemple: Conception *schéma STAR*

Introduction

- Avantage :

- une structure conviviale

- requêtes analytiques plus performances.

1) La table de FAIT

Stock une accumulation de lignes qui représentent des **observations** ou des **événements** qui enregistrent une activité métier spécifique

Ex: vente de produit, mouvement de stock, solde boursier, taux change quotidien, …

2) Les tables de DIMENSION

Décrit vos **entités métier**

Permet le filtrage et le regroupement des données de la table de faits.

Ex: des personnes, des lieux, des produits ou des concepts.

Processus de BI avec power Bi

Trois Types de calculs

- Des requêtes analytiques sur le modèle au moyen instructions via DAX (Data Analysis Expressions)

- On peut ajouter trois types de calculs à votre modèle de données :
  - Tables calculées
  - Colonnes calculées
  - Mesures

Tables calculées

- Formule DAX pour ajouter une table calculée à un modèle.
- La formule peut
  - dupliquer
  - transformer des données de modèle existantes
  - créer une série de données pour produire une nouvelle table

- Les données de la table calculée étant toujours importées dans le modèle, elles augmentent la taille de stockage du modèle et peuvent prolonger la durée d'actualisation des données

Trois types de

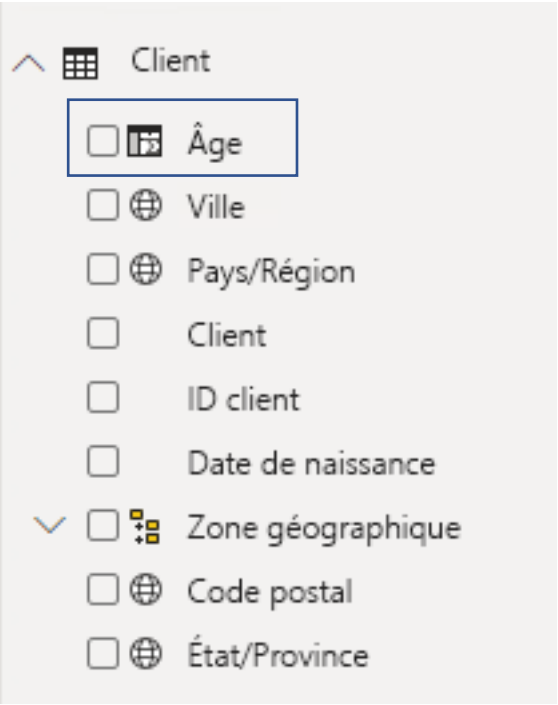
Colonnes calculées

- Formule DAX pour ajouter une colonne calculée à une table de votre modèle.

La formule est évaluée:

- pour chaque ligne de table
- retourne une valeur unique

- Dans le volet **Champs**, les colonnes calculées sont caractérisé par une icône spéciale.

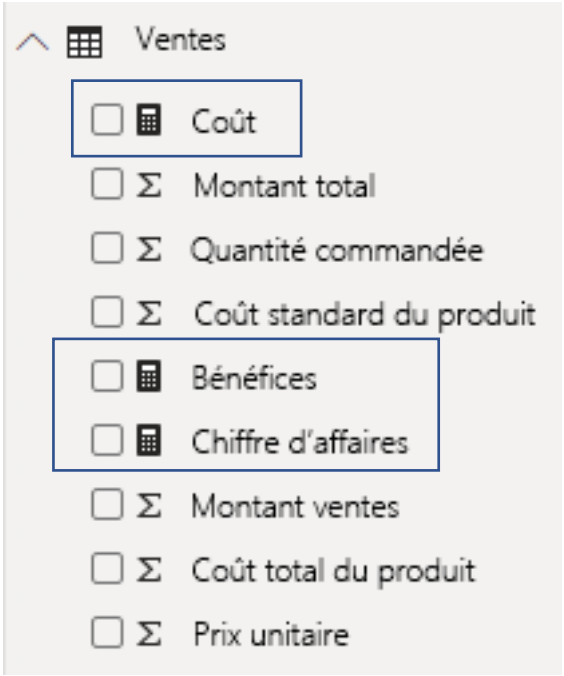


Trois types de

Mesures

- Formule DAX pour ajouter une mesure à une table de votre modèle.

  La formule:
  - Fait un résumé sur les données du modèle
  - Retourne une valeur unique

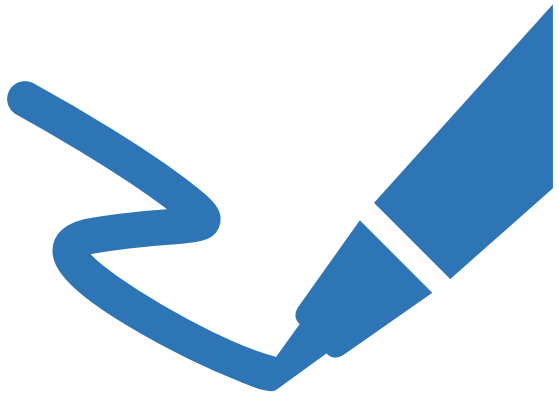


- Contrairement aux colonnes calculées, qui sont évaluées au moment de l'actualisation des données, les mesures sont utilisées au moment de la requête et leurs résultats ne sont jamais stockés dans le modèle.

- Dans le volet Champs, les mesures sont affichées avec l'icône de calculatrice.

Trois types de

Formule DAX



Ecrire une formule DAX

Défini un nom

Suivi du symbole égal (=)

Suivi par une **formule DAX**.

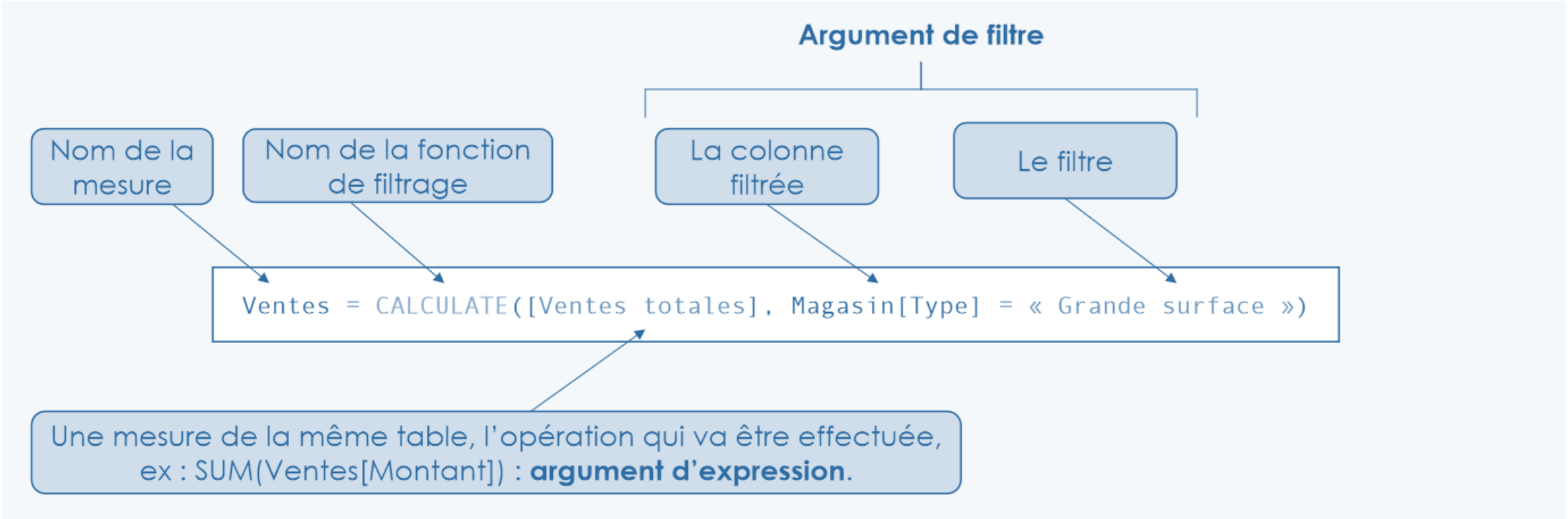
```DAX
<Calculation name> = <DAX formula>
```

- Une formule DAX se compose d'expressions qui retournent un résultat

- Les formules de *table calculée* doivent retourner un objet table
- Les formules de *colonne* et de *mesure calculées* doivent retourner une valeur scalaire ou unique.

Ecrire une formule DAX

- Exemple:



Ecrire une formule DAX

- Les formules sont assemblées à l'aide des éléments suivants :
  1. Fonctions DAX
  2. Opérateurs DAX
  3. Références aux objets de modèle
  4. Valeurs constantes, comme le nombre 24 ou le texte littéral « FY » (pour l'année fiscale)
  5. Variables DAX
  6. Espace blanc

1) Fonctions

- Function_Name(Arg1; Arg2; [Arg3]; ...)
- Les fonctions ont des arguments qui autorisent le passage de variables.
- Elles peuvent utiliser de nombreux appels de fonction et imbriquent souvent des fonctions dans d'autres fonctions.
- Dans une formule, les noms des fonctions doivent être suivis de parenthèses. Entre les parenthèses, les variables sont passées.
- Certaines fonctions n'acceptent pas d'arguments, ou les arguments peuvent être facultatifs.

Ecrire une formule

1) Fonctions

- Il existe une centaine de fonctions DAX, chacune conçue pour atteindre un objectif spécifique.
- Étant donné que DAX est issu du complément Power Pivot pour Microsoft Excel 2010, plus de 80 de ces fonctions sont également disponibles dans Excel.

- Toutefois, il existe des fonctions que vous ne trouverez pas dans Excel, car elles sont spécifiques à la modélisation de données :
  - Fonctions de navigation entre les relations
  - Fonctions de modification du contexte de filtre
  - Fonctions d'itérateur
  - Fonctions Time Intelligence
  - Fonctions Path

Ecrire une formule

1) Fonctions

- Exemples de fonctions provenant d'Excel:
  - IF teste si une condition: IF(<logical_test>,<value_if_true>[, <value_if_false>])
  - Fonctions de résumé: SUM, COUNT, AVERAGE, MIN, MAX
  - Autres exemples: ABS, ROUND, SQRT, LEN, LEFT, RIGHT, UPPER, DATE, YEAR, MONTH, NOW, ISNUMBER, TRUE, FALSE, AND, OR, NOT et IFERROR etc.

Ecrire une formule

1) Fonctions

- Exemples de fonctions spécifiques à Power BI:
  - **Fonction DISTINCTCOUNT**
  - compter le nombre de valeurs distinctes dans une colonne (sans doublon)
  - **Fonction DIVIDE**
  - pour effectuer une division
  - gère automatiquement le cas de la division par zéro; Si le dénominateur est zéro ou vide, la fonction retourne une valeur vide

Ecrire une formule

2) Opérateurs

- Les formules s'appuient également sur des *opérateurs DAX*, qui peuvent effectuer des calculs arithmétiques, comparer des valeurs, utiliser des chaînes ou des conditions de test.

Opérateurs arithmétiques

Le tableau ci-dessous répertorie les opérateurs arithmétiques.

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division
^	Élévation à la puissance

Opérateurs de comparaison

Le tableau suivant répertorie les opérateurs de comparaison, qui sont utilisés pour comparer deux valeurs. Le résultat est TRUE ou FALSE.

Opérateur	Description
=	Égal à
==	Strictement égal à
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
<>	Non égal à

Ecrire une formule

Opérateur de concaténation de texte

Utilisez le caractère esperluette (&) pour connecter, ou concaténer, deux valeurs de texte afin de produire une valeur de texte continue. Par exemple, considérez la définition de colonne calculée suivante :

Opérateurs logiques

Utilisez les opérateurs logiques pour combiner des expressions et produire un résultat unique. Le tableau suivant répertorie tous les opérateurs logiques.

Opérateur	Description
&&	Crée une condition AND entre deux expressions où chacune a un résultat booléen. Si les deux expressions retournent TRUE, la combinaison des expressions retourne également TRUE ; sinon, la combinaison retourne FALSE.
|| (double barre verticale)	Crée une condition OR entre deux expressions logiques. Si l'une des expressions retourne TRUE, le résultat est TRUE. Le résultat est FALSE uniquement quand les deux expressions sont FALSE.
IN	Crée une condition OR logique entre chaque ligne comparée à une table. Remarque : La syntaxe du constructeur de table utilise des accolades.
NOT	Inverse l'état d'une expression booléenne (FALSE à TRUE, et vice versa).

2) Opérateurs

- Exemple de colonne calculée avec opérateur:

```
Total Sales Amount = Sales[SalesAmount] + Sales[TaxAmt] + Sales[Freight]
```

```
Customer and Address = Customers[Name] & " : " & Customers[AddressLine1]
```

Ecrire une formule

3) Références aux objets de modèle

- Les formules peuvent uniquement faire référence à trois types d'objets de modèle :
  - des tables
  - des colonnes
  - des mesures

Une formule ne peut pas faire référence à une hiérarchie ou à un niveau de hiérarchie.

4) Variables DAX

- Les formules peuvent déclarer des variables DAX pour stocker les résultats.

- Lorsque de la déclaration d'une variable, une clause RETURN est utilisée pour définir l'expression, qui fait alors référence à la variable

- Avantage utilisation variables:
  - Amélioration de la lisibilité
  - Amélioration des performances, car les variables sont évaluées une seule fois

```
Revenue YoY % =
VAR RevenuePriorYear =
    CALCULATE(
        [Revenue],
        SAMEPERIODLASTYEAR('Date'[Date])
    )
RETURN
    DIVIDE(
        [Revenue] - RevenuePriorYear,
        RevenuePriorYear
    )
```

Ecrire une formule

5) Espace blanc

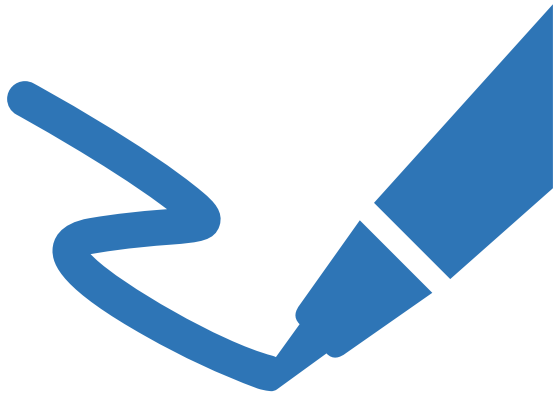
- L’espace blanc fait référence à des caractères que vous pouvez utiliser pour mettre en forme vos formules d’une manière rapide et simple à comprendre. Les caractères d’espace blanc sont les suivants :
  - Espaces
  - Retours chariot (Shift + Enter)

- L’espace blanc est facultatif et ne modifie pas la logique de votre formule, et n’a pas d’impact négatif sur les performances.

6) Types de données DAX

Type de données de modèle	Type de données DAX	Description
Nombre entier	Entier 64 bits	-2^{63} à $2^{63}-1$
Nombre décimal	Réel 64 bits	Négatif : $-1{,}79 \times 10^{308}$ à $-2{,}23 \times 10^{-308}$ Zéro Positif : $2{,}23 \times 10^{-308}$ à $1{,}79 \times 10^{308}$ Limité à 17 chiffres décimaux
Boolean	Boolean	TRUE ou FALSE
Texte	String	Chaîne de caractères Unicode
Date	Date/heure	Les dates valides sont toutes les dates après le 1er mars 1900
Devise	Devise	$-9{,}22 \times 10^{14}$ à $9{,}22 \times 10^{14}$ Limité à quatre chiffres décimaux de précision fixe
N/A	BLANK	Dans certains cas, il s'agit de l'équivalent de NULL dans une base de données (SQL)

Création Colonne calculée



Créer une colonne calculée

- Les colonnes calculées dans les modèles d'importation ont un coût.

  Elles augmentent la taille de stockage du modèle et peuvent allonger le délai d'actualisation des données.

  Cela est dû au fait que les colonnes calculées sont recalculées lorsqu'elles contiennent des dépendances de formules liées à des tables actualisées.

- Ex

```
Due Fiscal Year =
"FY"
    & YEAR('Due Date'[Due Date])
        + IF(
            MONTH('Due Date'[Due Date]) > 6,
            1
        )
```

& - concaténation
YEAR - retourne la valeur numérique entière de l'année d'échéance
IF - retourne la valeur lorsque le numéro du mois de la date d'échéance est compris entre 7 et 12 (juillet et décembre) ; dans le cas contraire, elle retourne la valeur BLANK

```
Due Month =
FORMAT('Due Date'[Due Date], "yyyy mmm")
```

Format - convertit la valeur de la colonne Due Date (Date d'échéance) en texte à l'aide d'une chaîne de format. Dans ce cas, la chaîne de format génère une étiquette qui décrit l'année et le nom du mois abrégé

- Exemple de colonne calculée:

```
1 MonthKey =
2 (ANNÉE('Due Date'[Due Date]) * 100) + MOIS('Due Date'[Due Date])
```

Date d'échéance	Année fiscale ...	Trimestre fiscal d'...	Mois d'éch...	Date complè...	MonthKey
01/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	01 juillet 2017	201707
02/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	02 juillet 2017	201707
03/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	03 juillet 2017	201707
04/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	04 juillet 2017	201707
05/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	05 juillet 2017	201707
06/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	06 juillet 2017	201707
07/07/2017 00:00:00	FY2017	FY2017 T1	Juil. 2017	07 juillet 2017	201707

- Exemple colonne calculée:

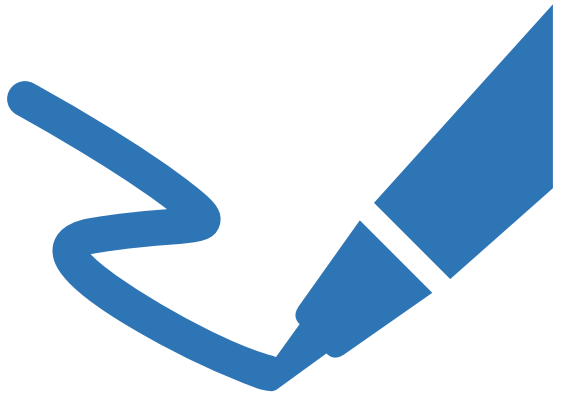
```
Month Number = MONTH('Calendar'[Date])

Quarter = QUARTER('Calendar'[Date])

Year = YEAR('Calendar'[Date])

Delay in Days = DATEDIFF(Sales[OrderDate],Sales[ShipDate], DAY)

Delay in Hours = DATEDIFF(Sales[OrderDate],Sales[ShipDate], HOUR)

End of Month = EOMONTH(Sales[OrderDate],0)

Start of Month = STARTOFMONTH(Sales[OrderDate])

Delay until EoM = DATEDIFF(Sales[OrderDate], EOMONTH(Sales[OrderDate],0), DAY)

Today = TODAY()

Yearly Income Class = IF(Customers[YearlyIncome] <30000, "Low", IF(Customers[YearlyIncome] <50000,"Standard", "High"))
```

- Exemple colonne calculée avec des caractères:

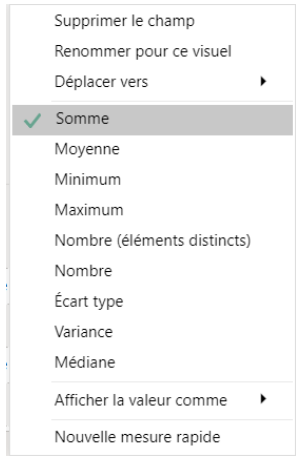
```
Name Cleaned = TRIM(Customers[Name])

Name Upper = UPPER(Customers[Name])

Group new = SUBSTITUTE(Territory[Group], "North America", "America")

Order Number = RIGHT(Sales[SalesOrderNumber],5)

Sales Order Line Number = CONCATENATE(Sales[Order Number],Sales[SalesOrderLineNumber])

First Name = MID(Customers[Name], 1, FIND( " ",Customers[Name]) - 1)

Date Formatted = FORMAT('Calendar'[Date], "dd MMM yyyy")
```

Exercice A

Création Mesure



Créer une mesure



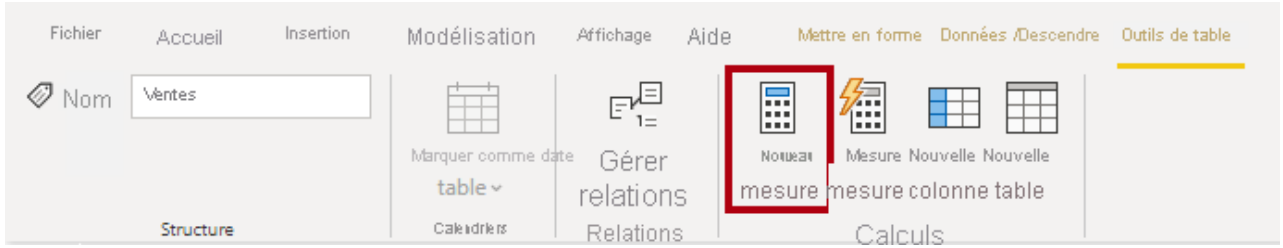
- Deux types de mesures:
  - Les mesures implicites sont des comportements automatiques qui permettent aux visuels de résumer les données des colonnes du modèle.
  - Les mesures explicites, sont des calculs que vous écrivez directement

```
Profit =
SUM(Sales[Profit Amount])
```

- Les mesures ne stockent pas de valeurs. Elles sont utilisées au moment de la requête pour retourner des résumés des données de modèle

- Exemple de mesure simple:

  Une mesure *simple* est une mesure qui agrège les valeurs d'une seule colonne ; elle fait ce que les mesures implicites font automatiquement.

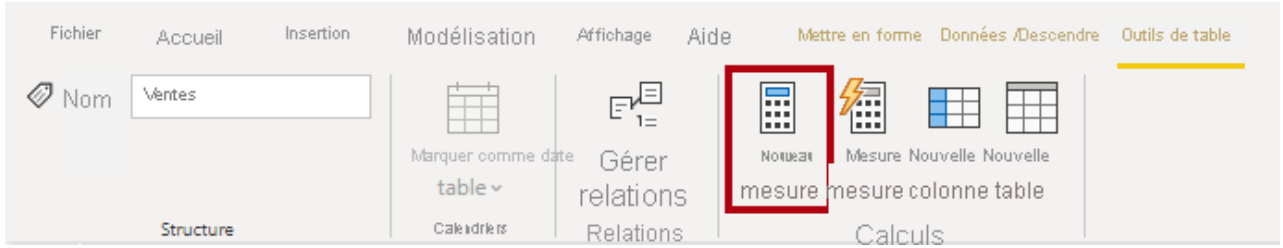



```
Revenue =
SUM(Sales[Sales Amount])
```

```
Cost =
SUM(Sales[Total Product Cost])
```

- Exemple de mesure composé:

Lorsqu'une mesure fait référence à une ou plusieurs mesures, elle est appelée *Mesure composée*.




```
Profit =
[Revenue] - [Cost]
```

Quelques fonctions

- Découverte de quelques fonctions:
  - Agrégation
  - Calculate
  - ALL
  - Time Intelligence

Quelques fonctions agrégations

- Fonction d'agrégation simple:

  Exemple: Sum, Count, DISTINCTCOUNT, Average, Min, Max, etc.

- Fonction d'itérateur d'agrégation:

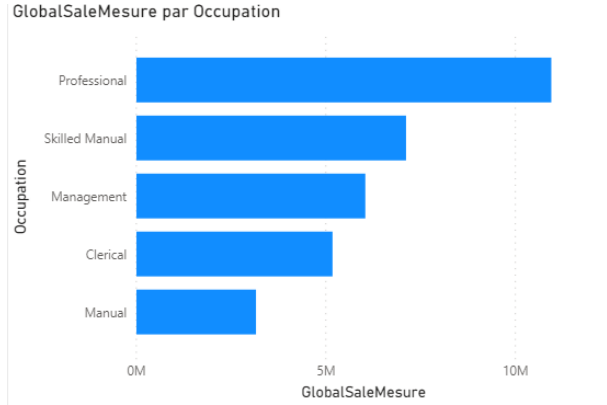
  Les fonctions d'itérateur énumèrent toutes

  les lignes d'une table donnée et évaluent

  une expression donnée pour **chaque ligne**.

  Exemples: SUMX, COUNTX, MINX, MAXX, RANKX et autres.

```
GlobalSaleMesure =
SUMX(sales,Sales[SalesAmount]
+Sales[TaxAmt]+Sales[Freight])
```



Exercice B

Fonction Calculate

- CALCULATE() est la fonction la plus importante et la plus puissante de DAX.
- C'est la seule fonction qui a la possibilité de modifier le contexte du filtre provenant de vos visuels.
- Il modifie le contexte du filtre(qui peut être une mesure ou une autre formule DAX) en appliquant/supprimant/modifiant des filtres.

```
CALCULATE(<expression>, [[<filter1>], <filter2>]…)
```

Fonction Calculate

- Exemple (sous forme Booléen et sous forme de table):

```
Revenue Red = CALCULATE([Revenue], 'Product'[Color] = "Red")
```

```
Revenue Red =
CALCULATE(
    [Revenue],
    FILTER(
        'Product',
        'Product'[Color] = "Red"
    )
)
```

Région	Chiffre d'affaires	Chiffre d'affaires Rouge
Australie	$10,655,335.96	$2,681,324.79
Canada	$16,355,770.46	$3,573,412.99
Centre	$7,909,009.01	$1,585,997.34
France	$7,251,555.65	$1,051,014.15
Allemagne	$4,878,300.38	$670,607.30
Nord-Est	$6,939,374.48	$1,876,016.33
Nord-ouest	$16,084,942.55	$2,292,905.61
Sud-Est	$7,879,655.07	$1,457,221.07
Sud-Ouest	$24,184,609.60	$5,345,637.47
Royaume-Uni	$7,670,721.04	$1,063,753.75
Total	**$109,809,274.20**	**$21,597,890.81**

Fonction Calculate

- Autres exemples:

```
Number of Women = CALCULATE(COUNTROWS(Customers),Customers[Gender] = "F“ )

Number of Clerical = CALCULATE(COUNT(Customers[First Name]),Customers[Occupation] = "Clerical“ )

Number of Clerical Women = CALCULATE(COUNTROWS(Customers), Customers[Occupation] =
                                                    "Clerical", Customers[Gender] = "F")

Number of Rich People = CALCULATE(COUNTROWS(Customers), Customers[YearlyIncome] > 100000)

Number of Manual and Skilled Manual Customers = CALCULATE(COUNTROWS(Customers),

                                        Customers[Occupation] IN  {"Manual", "Skilled Manual"})
```

Exercice C

Fonctions ALL

- La fonction ALL() supprime tous les filtres actuels du contexte de filtre actuel.

- Les fonctions DAX ALL(), ALLEXCEPT() et ALLSELECTED() sont toutes très similaires dans ce qu'elles font.

- TOUTES les fonctions sont très utiles pour calculer le grand total ou le rapport au total

Fonctions ALL

- Exemple ALL()

```
Grand Total = CALCULATE(SUM(Sales[SalesAmount]), ALL(Sales))

Ratio To Grand Total = SUM(Sales[SalesAmount])/[Grand Total]
```

Occupation	SalesAmount	Grand Total	Ratio To Grand Total
Clerical	$4 684 786,64	$29 358 677,22	15,96 %
Management	$5 467 861,54	$29 358 677,22	18,62 %
Manual	$2 857 970,89	$29 358 677,22	9,73 %
Professional	$9 907 977,28	$29 358 677,22	33,75 %
Skilled Manual	$6 440 080,86	$29 358 677,22	21,94 %
Total	**$29 358 677,22**	**$29 358 677,22**	**100,00 %**

Fonctions ALL

Exemple ALLEXCEPT()

```
Grand Total Except for Groups = CALCULATE(SUM(Sales[SalesAmount]), ALLEXCEPT(Territory,Territory[Group]))
```

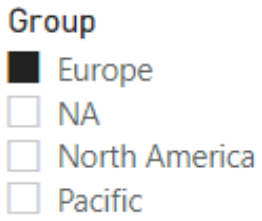
Group	Country	SalesAmount	Grand Total
⊟ Europe	France	$2 644 017,71	$29 358 677,22
	Germany	$2 894 312,34	$29 358 677,22
	United Kingdom	$3 391 712,21	$29 358 677,22
⊟ NA	NA		$29 358 677,22
⊟ North America	Canada	$1 977 844,86	$29 358 677,22
	United States	$9 389 789,51	$29 358 677,22
⊟ Pacific	Australia	$9 061 000,58	$29 358 677,22

Group	Country	SalesAmount	Grand Total Except for Groups
⊟ Europe	France	$2 644 017,71	$8 930 042,26
	Germany	$2 894 312,34	$8 930 042,26
	United Kingdom	$3 391 712,21	$8 930 042,26
⊟ North America	Canada	$1 977 844,86	$11 367 634,37
	United States	$9 389 789,51	$11 367 634,37
⊟ Pacific	Australia	$9 061 000,58	$9 061 000,58

The ALL Function

Exemple ALLSELECTED()

```
Grand Total for Selected Territories = CALCULATE(SUM(Sales[SalesAmount]), ALLSELECTED(Territory))
```



Region	SalesAmount	Grand Total
France	$2 644 017,71	$29 358 677,22
Germany	$2 894 312,34	$29 358 677,22
United Kingdom	$3 391 712,21	$29 358 677,22
Total	**$8 930 042,26**	**$29 358 677,22**

Region	SalesAmount	Grand Total for Selected Territories
France	$2 644 017,71	8 930 042,26
Germany	$2 894 312,34	8 930 042,26
United Kingdom	$3 391 712,21	8 930 042,26
Total	**$8 930 042,26**	**8 930 042,26**

Exercice D

Fonction Time Intelligence

- Time Intelligence concerne les calculs dans le temps. Plus précisément, Time Intelligence est associé aux calculs de dates, de mois, de trimestres ou d'années, et probablement du temps.

- Il est rare que vous deviez calculer au fil du temps en termes d'heures, de minutes ou de secondes.

- Dans les calculs DAX (Data Analysis Expressions), Time Intelligence correspond à la *modification du contexte de filtre pour les filtres de date*.

Spécification de la table de dates

- Respecter la configuration requise d'une *table de dates* :
  - Elle doit avoir une colonne de type de données Date (ou date/heure), connue sous le nom de *colonne de date*.
  - La colonne de dates doit contenir des valeurs uniques.
  - La colonne de dates ne doit pas contenir de valeurs vides.
  - Il ne doit pas y avoir de dates manquantes dans la colonne de dates.
  - La colonne de dates doit couvrir des années entières. Une année n'est pas nécessairement une année civile (janvier-décembre).
  - La table de dates doit être indiquée comme une table de dates.

Time Intelligence

- Synthèses dans le temps:

  DATESYTD, TOTALYTD,DATESBETWEEN, DATESINPERIOD

- Comparaisons dans le temps:

  DATEADD, PARALLELIPERIOD, SAMEPERIODLASTYEAR, NEXTDAY, NEXTMONTH, NEXTQUARTER, NEXTYEAR et PREVIOUSDAY, PREVIOUSMONTH, PREVIOUSQUARTER et PREVIOUSYEAR.

Time Intelligence

Previous Year Sales

- Previous Year Sales = CALCULATE(Sales[Total Sales], PREVIOUSYEAR('Calendar'[Date]))

  Une fois les ventes de l'année précédente générées, il est assez facile de calculer la différence par rapport à l'année précédente et la progression

- Difference to Previous Year = [Total Sales] - [Previous Year Sales]

- Progression to Previous Year = ([Total Sales] - [Previous Year Sales]) / [Previous Year Sales]

CalendarYear	SalesAmount	Previous Year Sales	Progression to Previous Year
2001	3 266 373,66		Infini
2002	6 530 343,53	3 266 373,66	99,93 %
2003	9 791 060,30	6 530 343,53	49,93 %
2004	9 770 899,74	9 791 060,30	-0,21 %
Total	**29 358 677,22**		**Infini**

Time Intelligence

Previous Sales

- Même logique PREVIOUSQUARTER(), PREVIOUSMONTH() and PREVIOUSDAY()

CalendarYear	SalesAmount	Previous Quarter Sales	Difference to Previous Quarter	Progression to Previous Quarte
⊟ **2001**	**3 266 373,66**		**3 266 373,66**	
3	1 453 522,89		1 453 522,89	
4	1 812 850,77	1 453 522,89	359 327,89	24,72 %
⊟ **2002**	**6 530 343,53**	**1 812 850,77**	**4 717 492,76**	**260,23 %**
1	1 791 698,45	1 812 850,77	-21 152,32	-1,17 %
2	2 014 012,13	1 791 698,45	222 313,68	12,41 %
3	1 396 833,62	2 014 012,13	-617 178,52	-30,64 %
4	1 327 799,32	1 396 833,62	-69 034,29	-4,94 %
⊟ **2003**	**9 791 060,30**	**1 327 799,32**	**8 463 260,98**	**637,39 %**
1	1 413 530,30	1 327 799,32	85 730,98	6,46 %
2	1 623 971,06	1 413 530,30	210 440,76	14,89 %
3	2 744 340,48	1 623 971,06	1 120 369,42	68,99 %
4	4 009 218,46	2 744 340,48	1 264 877,98	46,09 %
⊟ **2004**	**9 770 899,74**	**4 009 218,46**	**5 761 681,28**	**143,71 %**
1	4 283 629,96	4 009 218,46	274 411,50	6,84 %
2	5 436 429,15	4 283 629,96	1 152 799,19	26,91 %
3	50 840,63	5 436 429,15	-5 385 588,52	-99,06 %
4		50 840,63	-50 840,63	-100,00 %
Total	**29 358 677,22**		**29 358 677,22**	

Time Intelligence

Year over Year

- Sales Same Period Last Year = CALCULATE([Total Sales], SAMEPERIODLASTYEAR('Calendar'[Date]))

CalendarYear	SalesAmount	Sales Same Period Last Year
2001	**3 266 373,66**	
3	1 453 522,89	
4	1 812 850,77	
2002	**6 530 343,53**	**3 266 373,66**
1	1 791 698,45	
2	2 014 012,13	
3	1 396 833,62	1 453 522,89
4	1 327 799,32	1 812 850,77
2003	**9 791 060,30**	**6 530 343,53**
1	1 413 530,30	1 791 698,45
2	1 623 971,06	2 014 012,13
3	2 744 340,48	1 396 833,62
4	4 009 218,46	1 327 799,32
2004	**9 770 899,74**	**9 791 060,30**
1	4 283 629,96	1 413 530,30
2	5 436 429,15	1 623 971,06
3	50 840,63	2 744 340,48
4		4 009 218,46
Total	**29 358 677,22**	**19 587 777,48**

CalendarYear	SalesAmount	Sales Same Period Last Year
2001	**3 266 373,66**	
July	473 388,16	
August	506 191,69	
September	473 943,03	
October	513 329,47	
November	543 993,41	
December	755 527,89	
2002	**6 530 343,53**	**3 266 373,66**
January	596 746,56	
February	550 816,69	
March	644 135,20	
April	663 692,29	
May	673 556,20	
June	676 763,65	
July	500 365,15	473 388,16
August	546 001,47	506 191,69
September	350 466,99	473 943,03
October	415 390,23	513 329,47
Total	**29 358 677,22**	**19 587 777,48**

Time Intelligence

Running Total par année

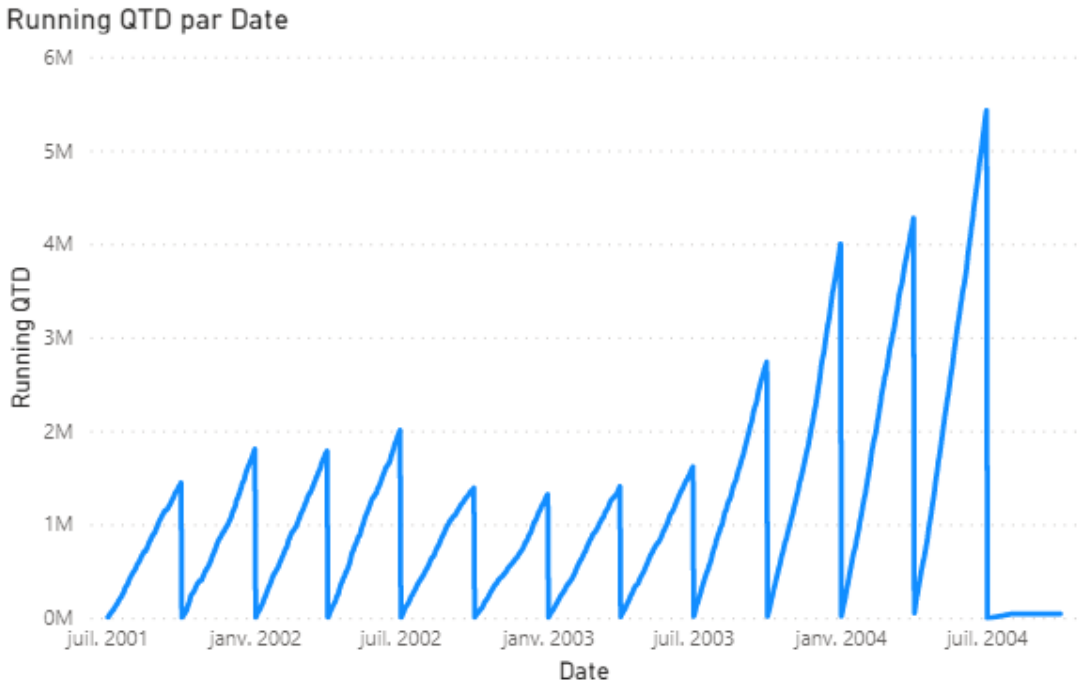
- Running YTD = CALCULATE([Total Sales], DATESYTD('Calendar'[Date]))

CalendarYear	SalesAmount	Running YTD
2001	**3 266 373,66**	**3 266 373,66**
July	473 388,16	473 388,16
August	506 191,69	979 579,85
September	473 943,03	1 453 522,89
October	513 329,47	1 966 852,36
November	543 993,41	2 510 845,77
December	755 527,89	3 266 373,66
2002	**6 530 343,53**	**6 530 343,53**
January	596 746,56	596 746,56
February	550 816,69	1 147 563,25
March	644 135,20	1 791 698,45
April	663 692,29	2 455 390,74
May	673 556,20	3 128 946,94
June	676 763,65	3 805 710,59
July	500 365,15	4 306 075,74
August	546 001,47	4 852 077,21
September	350 466,99	5 202 544,20
October	415 390,23	5 617 934,44
Total	**29 358 677,22**	**9 770 899,74**

Time Intelligence

Running Total (ce qui est en cours jusqu'à la fin de chaque mois ou trimestre)

- Running MTD = CALCULATE([Total Sales], DATESMTD('Calendar'[Date]))
- Running QTD = CALCULATE([Total Sales], DATESQTD('Calendar'[Date]))

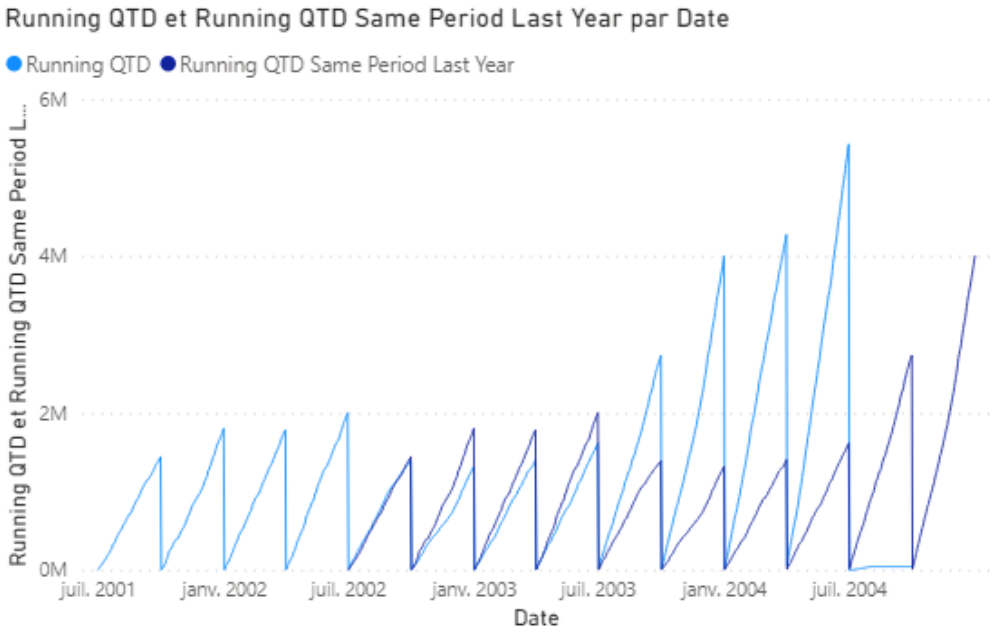


Time Intelligence

Running Total Same Period Last Year

Time Intelligence peut être imbriqué pour effectuer des calculs plus complexes

Running QTD Same Period Last Year = CALCULATE([Total Sales], SAMEPERIODLASTYEAR(DATESQTD('Calendar'[Date])))



Time Intelligence

Opening Balance Month - Closing Balance Month (Existe aussi pour Quarter and Year)

- Opening Balance Month = OPENINGBALANCEMONTH([Total Sales], 'Calendar'[Date])
- Closing Balance Month = CLOSINGBALANCEMONTH([Total Sales], 'Calendar'[Date])

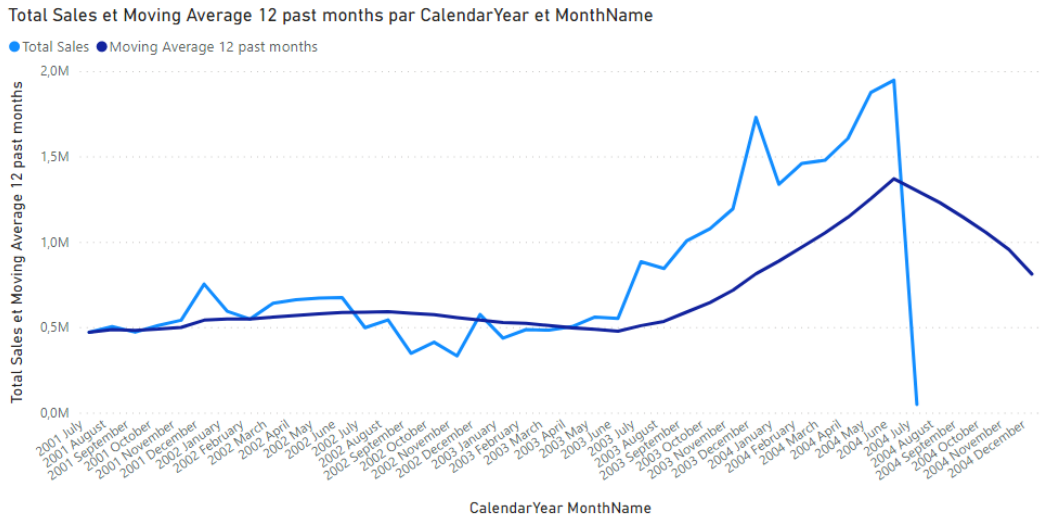
CalendarYear	Opening Balance Month	Closing Balance Month	Difference Opening Closing
2001		**22 168,72**	**-22 168,72**
July		15 012,18	-15 012,18
August	15 012,18	20 859,78	-5 847,60
September	20 859,78	35 782,70	-14 922,92
October	35 782,70	6 749,98	29 032,72
November	6 749,98	18 590,45	-11 840,47
December	18 590,45	22 168,72	-3 578,27
2002	**22 168,72**	**11 311,85**	**10 856,87**
January	22 168,72	21 762,16	406,56
February	21 762,16	17 688,07	4 074,09
March	17 688,07	13 931,52	3 756,55
April	13 931,52	40 174,33	-26 242,81
May	40 174,33	7 652,36	32 521,97
June	7 652,36	18 233,89	-10 581,53
July	18 233,89	19 782,57	-1 548,68
August	19 782,57	23 309,82	-3 527,26
September	23 309,82	13 599,67	9 710,15
October	13 599,67	7 191,33	6 408,34
November	7 191,33	15 652,65	-8 461,32
Total			

Time Intelligence

Rolling Average 12 Previous Months

- Moving Average 12 past months =

```
CALCULATE([Total Sales], DATESINPERIOD('Calendar'[Date], MAX('Calendar'[Date]), -1, YEAR))
/
CALCULATE(DISTINCTCOUNT('Calendar'[MonthName]), DATESINPERIOD('Calendar'[Date], MAX('Calendar'[Date]), -1, YEAR))
```

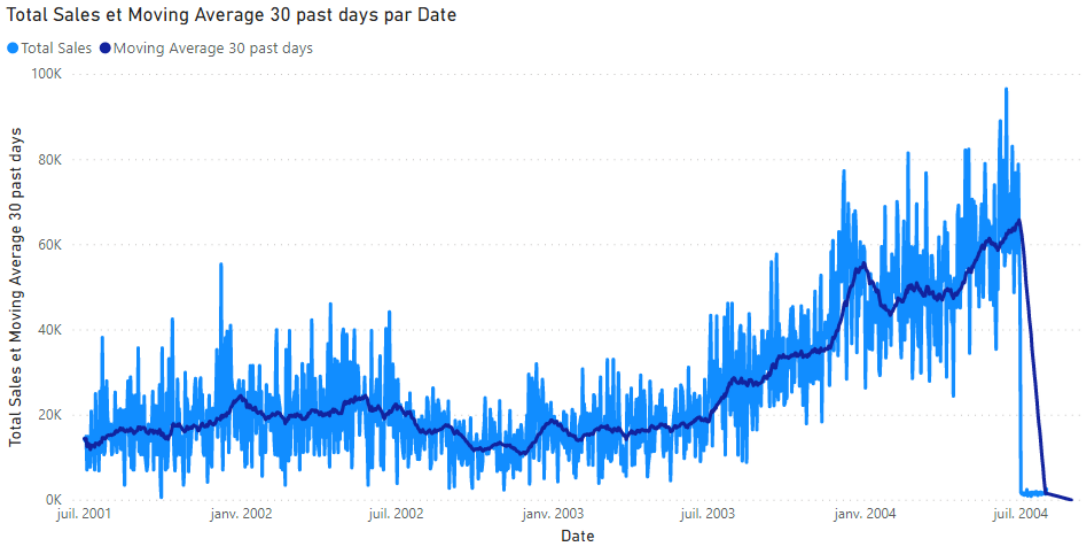


Time Intelligence

Rolling Average 30 Previous Days

- Moving Average 30 past days =

```
CALCULATE([Total Sales], DATESINPERIOD('Calendar'[Date], MAX('Calendar'[Date]), -30, DAY))

/

CALCULATE(DISTINCTCOUNT('Calendar'[Date]), DATESINPERIOD('Calendar'[Date], MAX('Calendar'[Date]), -30, DAY))
```



Exercice D