

# Power BI – Formation utilisateur

Comprendre et utiliser les fonctionnalités et paramètres de Power BI

**Module 6 – Enrichir le modèle de données**

# Objectifs du module

1. Comprendre le langage DAX
2. Découvrir les fonctions DAX
3. Concevoir des tables, colonnes et mesures calculées avec DAX
4. Utiliser les variables

1.

## Comprendre le langage DAX

- a.** Qu'est-ce que DAX et comment ça fonctionne ?
- b.** Syntaxe générale
- c.** Définition des variables et utilisation
- d.** Où utiliser DAX
- e.** Contextes d'évaluation

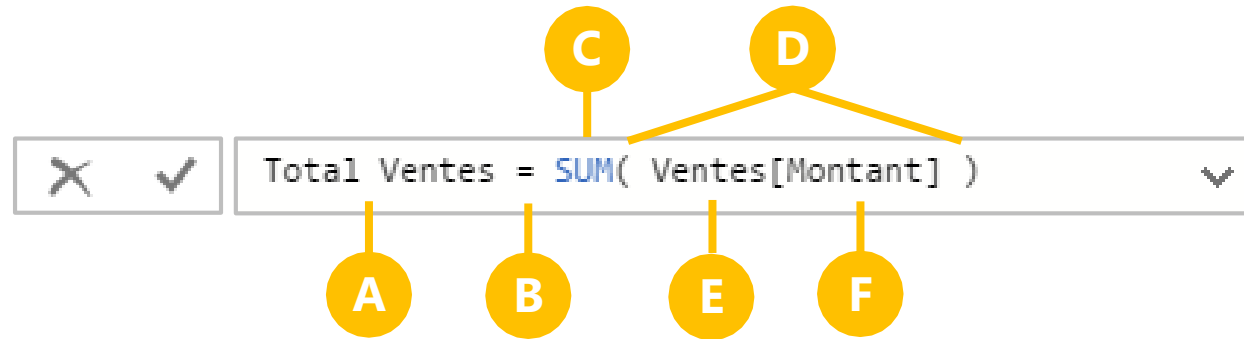
# Qu'est-ce que DAX ?

- Data Analysis Expression
- Collection de fonctions, d'opérateurs et de constantes qui peuvent être utilisés dans une formule ou une expression, pour calculer et retourner une ou plusieurs valeurs
  - Dans Microsoft SQL Server Analysis Services, Power Pivot dans Excel, et Power BI Desktop
- Langage fonctionnel (le code exécuté est contenu dans une fonction)

# Comment fonctionne DAX ?

- DAX fournit des fonctions qui ont les mêmes fonctionnalités et noms que les fonctions Excel
- Conçu pour fonctionner avec des tables et colonnes. Il a deux types principaux de données :
  - **Numérique** : entiers, décimaux, devises, dates et booléens (TRUE/FALSE)
  - **Autre** : chaînes et objets binaires (fichiers image, son ou vidéo)
- Fonctions peuvent contenir d'autres fonctions imbriquées, des instructions conditionnelles et des références à des valeurs

# Syntaxe générale



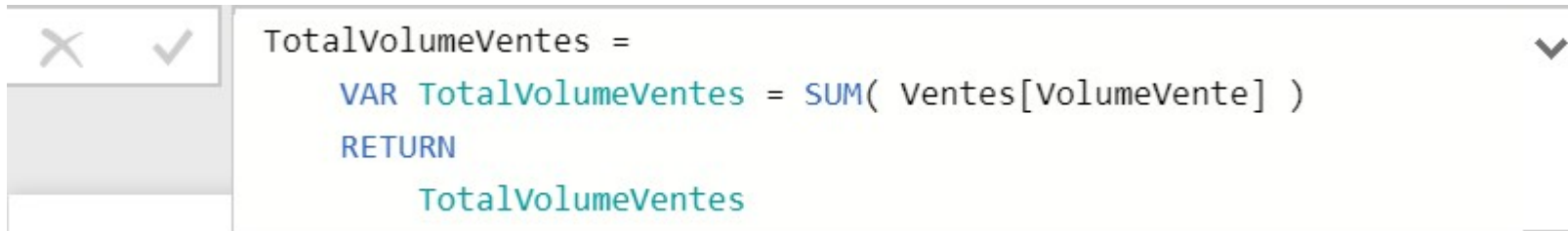
- A. Le nom de la mesure, ici **Total Ventes**
- B. L'opérateur (signe égal =) indique le début de la formule. Après le calcul, un résultat est retourné
- C. La fonction DAX **SUM** additionne tous les nombres figurant dans la colonne **[Montant]** de la table **Ventes**
- D. Des parenthèses **()** entourent l'expression qui peut contenir un ou plusieurs arguments. Toutes les fonctions nécessitent au moins un argument. Un argument passe une valeur à une fonction
- E. La table référencée **Ventes**
- F. La colonne référencée **[Montant]** dans la table **Ventes**. Avec cet argument, la fonction SUM() sait sur quelle colonne agréger une somme

# Variables

- Stocke le résultat d'une expression
- Peut être transmise en tant qu'argument à d'autres expressions de mesure
  - Une fois que les valeurs résultantes ont été calculées pour une expression de variable, ces valeurs ne changent pas, même si la variable est référencée dans une autre expression
  - Utilisent le contexte d'évaluation de la définition au lieu de celui où la variable est utilisée
- Peut être de n'importe quel type de données, notamment une table entière

# Utilisation de variables

- Possibilité de définir une variable (n'importe où dans une expression DAX) à l'aide de la syntaxe suivante :

A screenshot of a DAX formula bar. On the left, there is a small panel with a close button (X) and a checkmark button. The main area contains the following DAX code:

```
TotalVolumeVentes =  
    VAR TotalVolumeVentes = SUM( Ventes[VolumeVente] )  
    RETURN  
        TotalVolumeVentes
```

- Remarques :
  - Les mesures ne peuvent pas faire référence à des variables définies en dehors de l'expression de mesure
  - Les variables peuvent appeler des mesures et se référer à des variables définies précédemment
  - Si, pour une raison quelconque, la variable n'est pas utilisée dans le code, elle n'est pas évaluée



# Où utiliser DAX

## ○ Colonne calculée

- L'expression DAX définie dans une colonne calculée fonctionne dans le contexte de la ligne de la table à laquelle elle appartient
- Les valeurs de la colonne calculée sont recalculées uniquement si les données sous-jacentes sont actualisées ou en cas de recalcul manuel

## ○ Mesure

- L'expression DAX est évaluée dans le contexte de la cellule de la table ou de la requête DAX (ne travaille pas ligne par ligne)

# Colonne calculée ou mesure?

## ○ Colonne calculée

- Une colonne calculée est utile pour segmenter ou filtrer la valeur, ou pour faire un calcul sur chaque ligne de la table
- Une colonne calculée fait partie de la table du modèle de données et consomme de la mémoire

## ○ Mesure

- Une mesure est utile pour calculer des pourcentages, des taux ou pour des agrégations complexes
- Une mesure consomme du CPU car elle est calculée à la volée lorsqu'elle est utilisée dans un visuel

# Contextes d'évaluation

- Changent la façon avec laquelle une expression est évaluée, fournissant des résultats différents pour la même expression
- Reflètent la sélection de ligne ou de cellule actuelle, ainsi que toutes les données associées
- Il existe deux types de contexte :
  - **Le contexte de ligne**
  - **Le contexte de filtre**

# Contexte de ligne

- Provient d'une colonne calculée ou de fonctions spécifiques (FILTER et fonctions avec suffixe en X – SUMX, AVERAGEX, etc.)
- Représente les valeurs dans chaque ligne individuelle et celles dans les colonnes associées à la ligne actuelle
- Ne se propage pas via les relations

DebutSemaine = DimDates[DateComplete] - WEEKDAY(DimDates[DateComplete]; 2 ) +1

aine	Jour de l'année	Nom du jour	Jour	DebutSemaine
0	185	lundi	3	03/07/2000
1	186	mardi	4	03/07/2000
2	187	mercredi	5	03/07/2000
3	188	jeudi	6	03/07/2000
4	189	vendredi	7	03/07/2000
5	190	samedi	8	03/07/2000
6	191	dimanche	9	03/07/2000
0	190	lundi	9	09/07/2001
1	191	mardi	10	09/07/2001
2	192	mercredi	11	09/07/2001
3	193	jeudi	12	09/07/2001

CHAMPS

Rechercher

\_Mesures

DimDates

Σ Année

DateComplete

DebutSemaine

Σ Jour

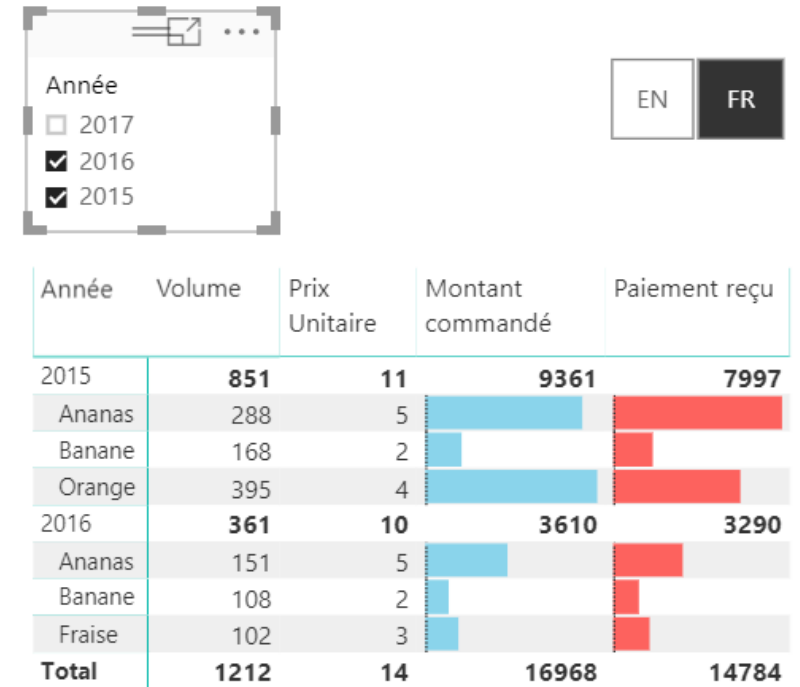
Σ Jour de l'année

Σ Jour de la semaine

Σ Mois

# Contexte de filtre

- Jeu de filtres appliqué à l'évaluation d'une expression DAX (en général une mesure)
- Peut être implicite (filtres visuels, interaction utilisateur) ou explicite (utilisation de CALCULATE ou CALCULATETABLE)
- Se propage dans le sens 1 à n d'une relation
- N'affecte pas le contexte de ligne



# Altération du contexte

- Possibilité de contrôler le contexte (et sa propagation – pour le contexte de filtre) en utilisant certaines fonctions
- Exemples:
  - La fonction **RELATED** développe le contexte de la ligne actuelle pour inclure des valeurs dans une colonne associée. Elle permet d'effectuer des recherches
  - La fonction **FILTER** permet de spécifier les lignes à inclure dans le contexte actuel
  - La fonction **ALL** définit le contexte dans une formule. Elle peut être utilisée pour remplacer les filtres qui sont appliqués comme résultat du contexte de requête
  - La fonction **ALLEXCEPT** permet de supprimer tous les filtres à l'exception d'un filtre à spécifier
  - Les fonctions **EARLIER** et **EARLIEST** permettent d'effectuer une boucle sur des tables en effectuant des calculs, tout en référant une valeur issue d'une boucle interne

## 2.

### Découvrir les fonctions DAX

- a. Fonctions statistiques, d'agrégation et de comptage
- b. Fonctions math. et trigonométriques
- c. Fonctions de filtrage
- d. Fonctions logiques
- e. Fonctions de date et d'heure, de Time Intelligence
- f. Fonctions d'information
- g. Fonctions de texte

# Fonctions statistiques et d'agrégation

- Permettent la création de sommes et de moyennes, la recherche de valeurs maximale et minimale, etc.
- Permettent également de compter différents éléments, tels que des valeurs distinctes, des valeurs non vides et des lignes de table
  - SUM()
  - SUMX()
  - AVERAGE
  - MIN()
  - MAX()
  - COUNT()
  - COUNTA() : compte le nombre de cellules qui ne sont pas vides
  - COUNTBLANK()
  - COUNTROWS(Table)
  - DISTINCTCOUNT()
  - Etc.



# Fonctions math. et trigonométriques

- Très semblables aux fonctions mathématiques et trigonométriques Excel (même syntaxe et comportement)
- Quelques différences mineures dans les types de données numériques utilisés
  - DIVIDE()
  - POWER()
  - RAND()
  - SIGN()
  - ROUND()
  - Etc.

# Fonctions de filtrage

- Permettent de retourner des types de données spécifiques, de rechercher des valeurs dans des tables associées et de procéder à un filtrage par valeurs associées
- S'appuient sur des tables et des relations, comme une base de données
- Permettent de manipuler le contexte de données pour créer des calculs dynamiques
  - CALCULATE()
  - EARLIER()
  - ALL()
  - ALLEXCEPT()
  - ALLSELECTED()
  - CROSSFILTER()
  - USERELATIONSHIP()
  - FILTER()
  - Etc.

# Evaluation des paramètres

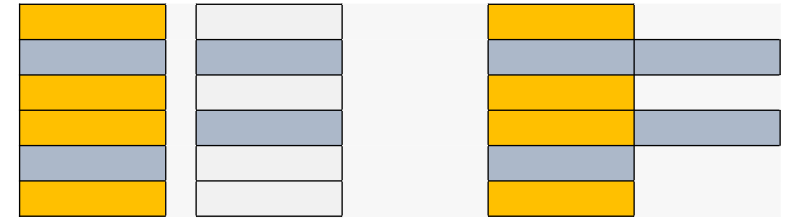
- Paramètres des fonctions évalués dans l'ordre chronologique
  - Sauf pour CALCULATE et CALCULATETABLE
    - Premier paramètre évalué seulement après l'évaluation de tous les autres
- Exemple :

```
VolumeVentes2015 = CALCULATE(  
    SUM( Ventes[VolumeVente] );  
    Ventes[AnneeCommande] = 2015  
)
```

# Fonctions de jointures

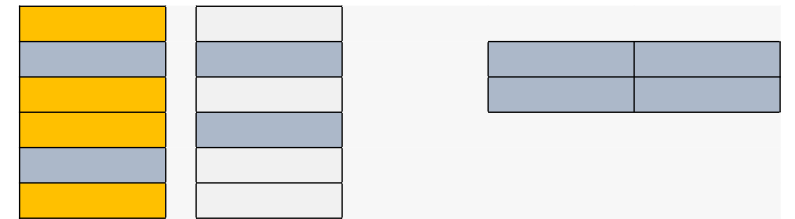
- GENERATEALL et NATURALINNERJOIN

Toutes les lignes de la table 1 et celles apparaissant dans la table 2



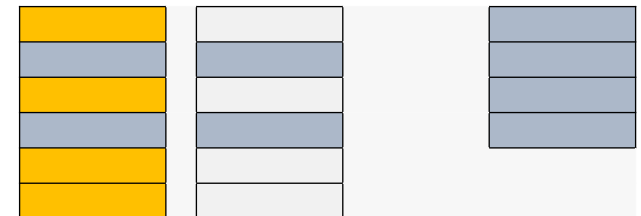
- GENERATE et NATURALINNERJOIN

Uniquement les lignes apparaissant dans les deux tables



- EXCEPT

Lignes de la table 1 n'apparaissant pas dans la table 2



# Fonctions logiques

- Agissent sur une expression pour retourner des informations sur les valeurs ou ensembles au sein de l'expression
- Peuvent également être exprimées sous forme d'*opérateurs*. Par exemple, **AND** peut être tapé sous la forme (remplacé par) **&&** dans l'expression DAX
  - AND
  - OR
  - NOT
  - IF(Test logique, Valeur si vrai, Valeur si faux)
  - IFERROR(Valeur, Valeur si erreur)
  - etc.

# Fonctions de date et d'heure

- Utilisent un type de données **DateTime**
- DateTime est stocké en tant que valeur à virgule flottante
  - Partie entière (avant la virgule) = nombre de jours après le 30/12/1899
  - Partie décimale (après la virgule) = fraction d'une journée, définie en secondes (1 / 24 \* 60 \* 60)
- Heures, minutes et secondes sont converties en fractions décimales d'une journée
- Expression de Date
  - Date + 1 = le jour d'après
  - Date - 1 = le jour précédent

# Fonctions de date et d'heure

- • DATE(Année, Mois, Jour)
- • NOW()
- • TODA
- Y()
- TIME()
- • HOUR()
- • MINUTE()
- DAY()
- MONTH()
- YEAR()
- WEEKDAY()
- WEEKNUM()
- EOMONTH()
- CALENDAR()
- CALENDARAUTO()
- DATEDIFF()
- Etc.

# Fonctions Time Intelligence

- Visent à répondre aux besoins d'analyse décisionnelle en permettant de manipuler des données à l'aide de périodes (jours, mois, trimestres et années, entre autres), puis de générer et comparer des calculs sur ces périodes
  - DATEADD()
  - DATESBETWEEN()
  - DATESINPERIOD()
  - DATESMTD()
  - PARALLELPERIOD()
  - FIRSTDATE()
  - FIRSTNONBLANK()
  - STARTOFMONTH()
  - ENDOFMONTH()
  - PREVIOUSDAY()
  - NEXTDAY()
  - TOTALMTD()
  - Etc.



# Fonctions d'information

- Examinent la cellule ou la ligne qui est fournie comme argument et indique si la valeur correspond au type attendu
- Bien qu'elles puissent s'avérer utiles selon les circonstances, il est utile de connaître le type de données de vos colonnes à l'avance, plutôt que de vous fier uniquement à ces fonctions
  - ISBLANK()
  - ISNUMBER()
  - ISTEXT()
  - ISNONTTEXT()
  - ISERROR()
  - Etc.

# Fonctions de texte

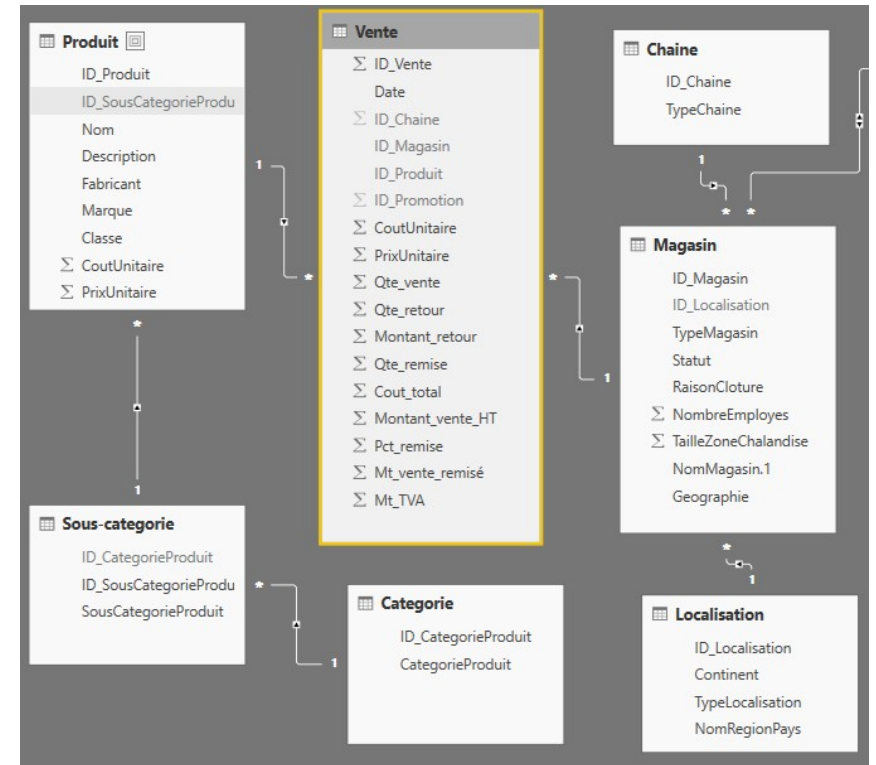
- Permettent de retourner une partie d'une chaîne, de rechercher un texte dans une chaîne ou de concaténer des valeurs de chaîne
- Comprennent également des fonctions pour le contrôle des formats des dates, heures et nombres
  - CONCATENATE()
  - REPLACE()
  - SEARCH()
  - UPPER()
  - LOWER()
  - FIXED()
  - Etc.

# 3/4 Créer des tables, colonnes et mesures calculées (avec et sans variables)

- a. Colonne calculée  
et relations
- b. Mesures  
rapides

# Colonne calculée et relations

- Il est possible de faire des calculs à partir de colonnes externes à la table via :
  - RELATED(table[champs])
- **Attention : ne fonctionne qu'avec les champs en relation DIRECTE dans le sens n à 1**
  - Par exemple pour une colonne calculée dans la table Vente, vous pouvez faire :  
RELATED(Produit[Marque])  
mais pas RELATED(Sous-categorie[SousCategorieProduit])



# Mesures rapides

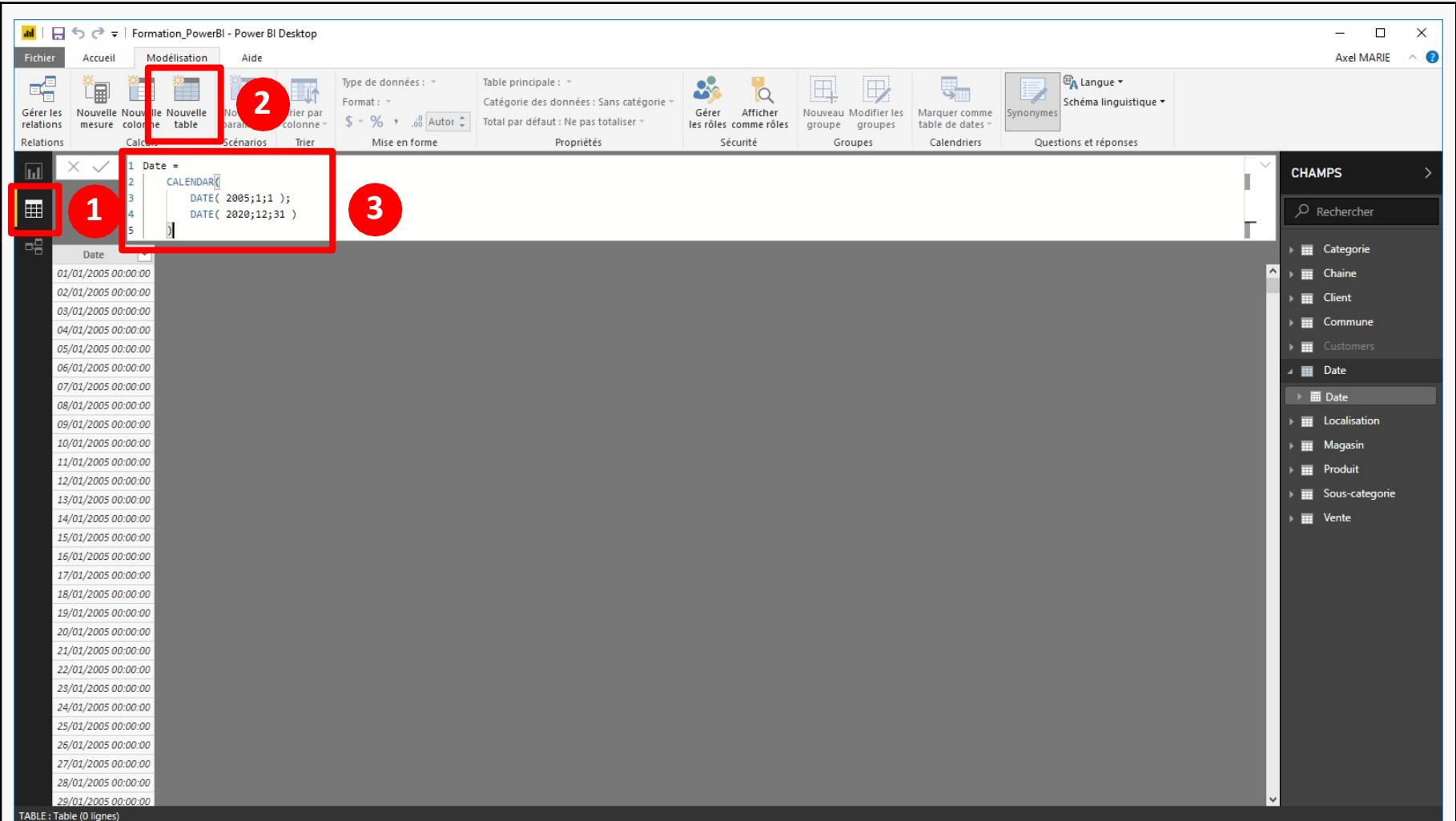
- Une mesure rapide exécute un ensemble de commandes DAX en arrière-plan
  - Il n'y a pas besoin d'écrire de commande DAX
- Cinq groupes de types de calculs de mesure rapide:
  - Agrégat dans une catégorie (moyenne, variance, maximum, etc.)
  - Filtres et bases de référence (mesure filtrée, différence par rapport à la référence, totaux provenant des nouvelles catégories, etc.)
  - Time intelligence (cumul annuel/trimestriel/mensuel jusqu'à ce jour, variation d'une année/trimestre/mois à l'autre, moyenne mobile, etc.)
  - Totaux (résultat cumulé, total de la catégorie, etc.)
  - Opérations mathématiques (addition, soustraction, etc.)
  - Texte (liste concaténée de valeurs)

# Exercices pratiques

- a. Créer une table calculée de type Date
- b. Créer des colonnes calculées
- c. Créer des mesures calculées (mesure rapide ou non, avec et sans variable)

# Ajouter une table calculée

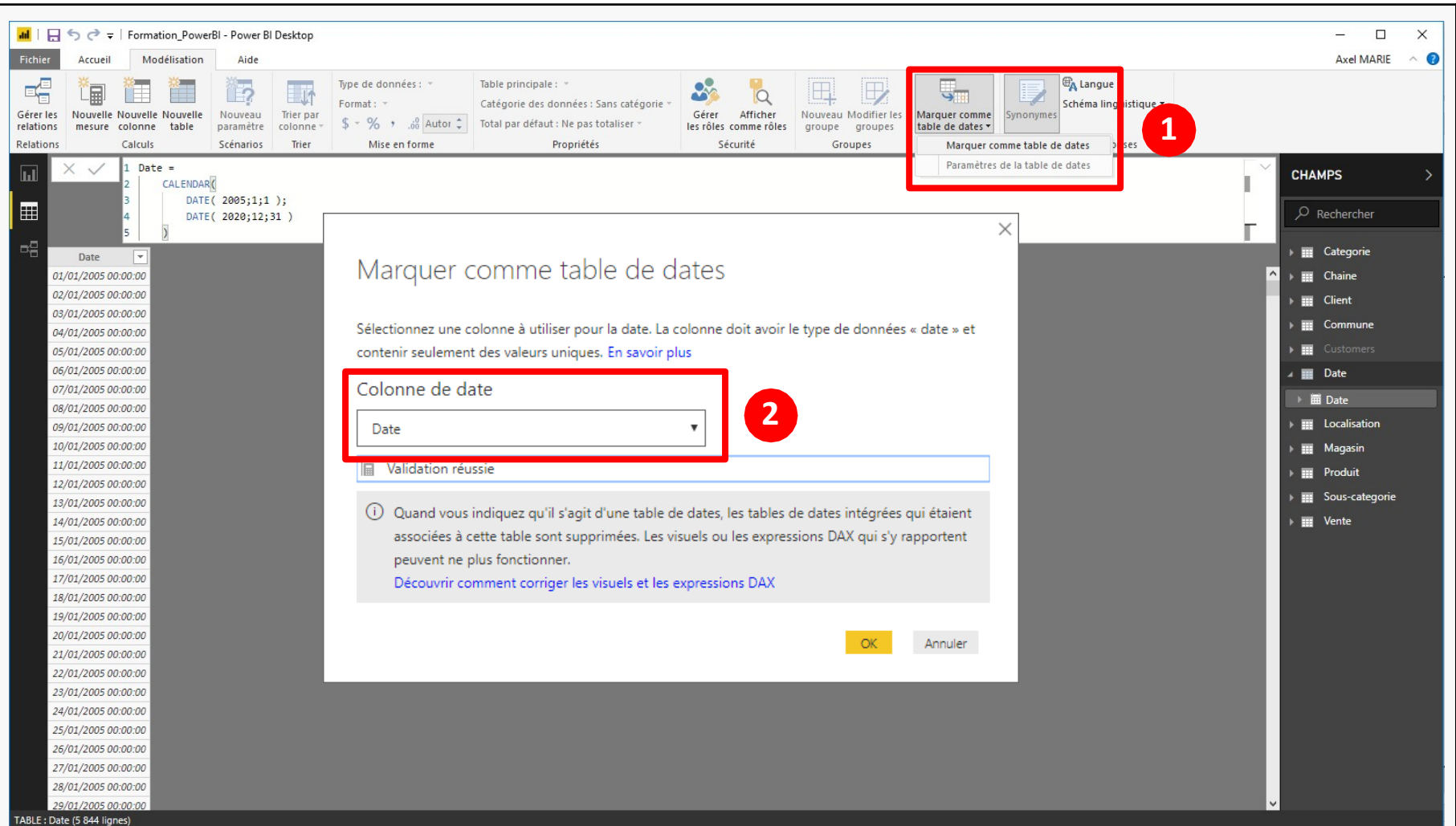
1. Rendez-vous dans la vue **Données**.
2. Dans le ruban **Modélisation**, cliquez sur **Nouvelle table**.
3. Entrez la formule DAX suivante :  
**CALENDAR(**  
    **DATE( 2005;1;1 );**  
    **DATE( 2020;12;31 )**  
**)**



# Marquer comme table de dates

1. Dans le ruban **Modélisation**, cliquez sur **Marquer comme table de dates**.

2. Dans la fenêtre qui s'ouvre, sélectionnez le champ **Date** et cliquez sur OK.





# Ajouter des colonnes calculées

1. Cliquez sur **Nouvelle colonne**.

2. Créez 4 colonnes :

**DebutSemaine** =  
'Date'[Date] -  
WEEKDAY('Date'[Date]; 2  
) +1

**AnneeMois** = FORMAT(  
'Date'[Date]; "yyyymm" )

**NomMois** = FORMAT(  
'Date'[Date]; "mmmm" )

**Annee** = YEAR(  
'Date'[Date] )

Formation\_PowerBI - Power BI Desktop

Fichier Accueil Modélisation Aide

Gérer les relations Nouvelles mesures Nouvelle colonne Nouvelles tables Nouveau paramètre Trier par colonne Scénarios Trier

Type de données : Date/Heure Format : \*14/03/2001 (dd/MM/yyyy) Catégorie des données : Sans catégorie Total par défaut : Ne pas totaliser

Gérer les rôles Afficher comme rôles Nouveaux groupes Modifier les groupes Marquer comme table de dates Synonymes Schéma linguistique

Questions et réponses

DebutSemaine = 'Date'[Date] - WEEKDAY('Date'[Date];2) +1

Date	NomMois	Annee	AnneeMois	DebutSemaine
01/01/2005 00:00:00	janvier	2005	200501	27/12/2004
02/01/2005 00:00:00	janvier	2005	200501	03/01/2005
03/01/2005 00:00:00	janvier	2005	200501	03/01/2005
04/01/2005 00:00:00	janvier	2005	200501	03/01/2005
05/01/2005 00:00:00	janvier	2005	200501	03/01/2005
06/01/2005 00:00:00	janvier	2005	200501	03/01/2005
07/01/2005 00:00:00	janvier	2005	200501	03/01/2005
08/01/2005 00:00:00	janvier	2005	200501	03/01/2005
09/01/2005 00:00:00	janvier	2005	200501	03/01/2005
10/01/2005 00:00:00	janvier	2005	200501	10/01/2005
11/01/2005 00:00:00	janvier	2005	200501	10/01/2005
12/01/2005 00:00:00	janvier	2005	200501	10/01/2005
13/01/2005 00:00:00	janvier	2005	200501	10/01/2005
14/01/2005 00:00:00	janvier	2005	200501	10/01/2005
15/01/2005 00:00:00	janvier	2005	200501	10/01/2005
16/01/2005 00:00:00	janvier	2005	200501	10/01/2005
17/01/2005 00:00:00	janvier	2005	200501	17/01/2005
18/01/2005 00:00:00	janvier	2005	200501	17/01/2005
19/01/2005 00:00:00	janvier	2005	200501	17/01/2005
20/01/2005 00:00:00	janvier	2005	200501	17/01/2005
21/01/2005 00:00:00	janvier	2005	200501	17/01/2005
22/01/2005 00:00:00	janvier	2005	200501	17/01/2005
23/01/2005 00:00:00	janvier	2005	200501	17/01/2005
24/01/2005 00:00:00	janvier	2005	200501	24/01/2005
25/01/2005 00:00:00	janvier	2005	200501	24/01/2005
26/01/2005 00:00:00	janvier	2005	200501	24/01/2005
27/01/2005 00:00:00	janvier	2005	200501	24/01/2005
28/01/2005 00:00:00	janvier	2005	200501	24/01/2005
29/01/2005 00:00:00	janvier	2005	200501	24/01/2005
30/01/2005 00:00:00	janvier	2005	200501	24/01/2005
31/01/2005 00:00:00	janvier	2005	200501	31/01/2005
01/02/2005 00:00:00	février	2005	200502	31/01/2005

TABLE : Date (5 844 lignes) COLONNE : DebutSemaine (836 valeurs distinctes)

CHAMPS

Rechercher

- Categorie
- Chaîne
- Client
- Commune
- Customers
- Date
  - Annee
  - AnneeMois
  - Date
  - DebutSemaine
  - NomMois
- Localisation
- Magasin
- Produit
- Sous-catégorie
- Vente

# Créer une relation entre les dates

1. Allez dans la vue **Données**.
2. Ajoutez une relation entre **Vente[Date]** et **Date[Date]**

The screenshot shows the Power BI Desktop interface in the 'Données' view. The 'Vente' table is selected, and its 'Date' column is highlighted. The 'Date' table is also visible in the background. A dialog box titled 'Modifier la relation' is open, showing the relationship between 'Vente' and 'Date'. The 'Date' column in the 'Vente' table is linked to the 'Date' column in the 'Date' table. The cardinality is set to 'Plusieurs à un (\*:1)' and the cross-filter direction is 'À sens unique'. The 'Rendre cette relation active' checkbox is checked.

**Modifier la relation**

Sélectionnez des tables et des colonnes qui sont liées.

Vente

ID_Vente	Date	ID_Chaine	ID_Magasin	ID_Produit	ID_Promotion	CoutUnitaire	PrixUnitaire
212402	jeudi 13 janvier 2011	1	69	38	2	99,14	
263494	jeudi 13 janvier 2011	1	95	39	2	99,14	
48652	jeudi 13 janvier 2011	1	169	40	2	99,14	

Date

Date	NomMois	Annee	AnneeMois	DebutSemaine
01/01/2005 00:00:00	janvier	2005	200501	27/12/2004
02/01/2005 00:00:00	janvier	2005	200501	27/12/2004
03/01/2005 00:00:00	janvier	2005	200501	03/01/2005

Cardinalité: Plusieurs à un (\*:1)

Direction du filtrage croisé: À sens unique

☒ Rendre cette relation active

☐ Intégrité référentielle supposée

☐ Appliquer le filtre de sécurité dans les deux directions

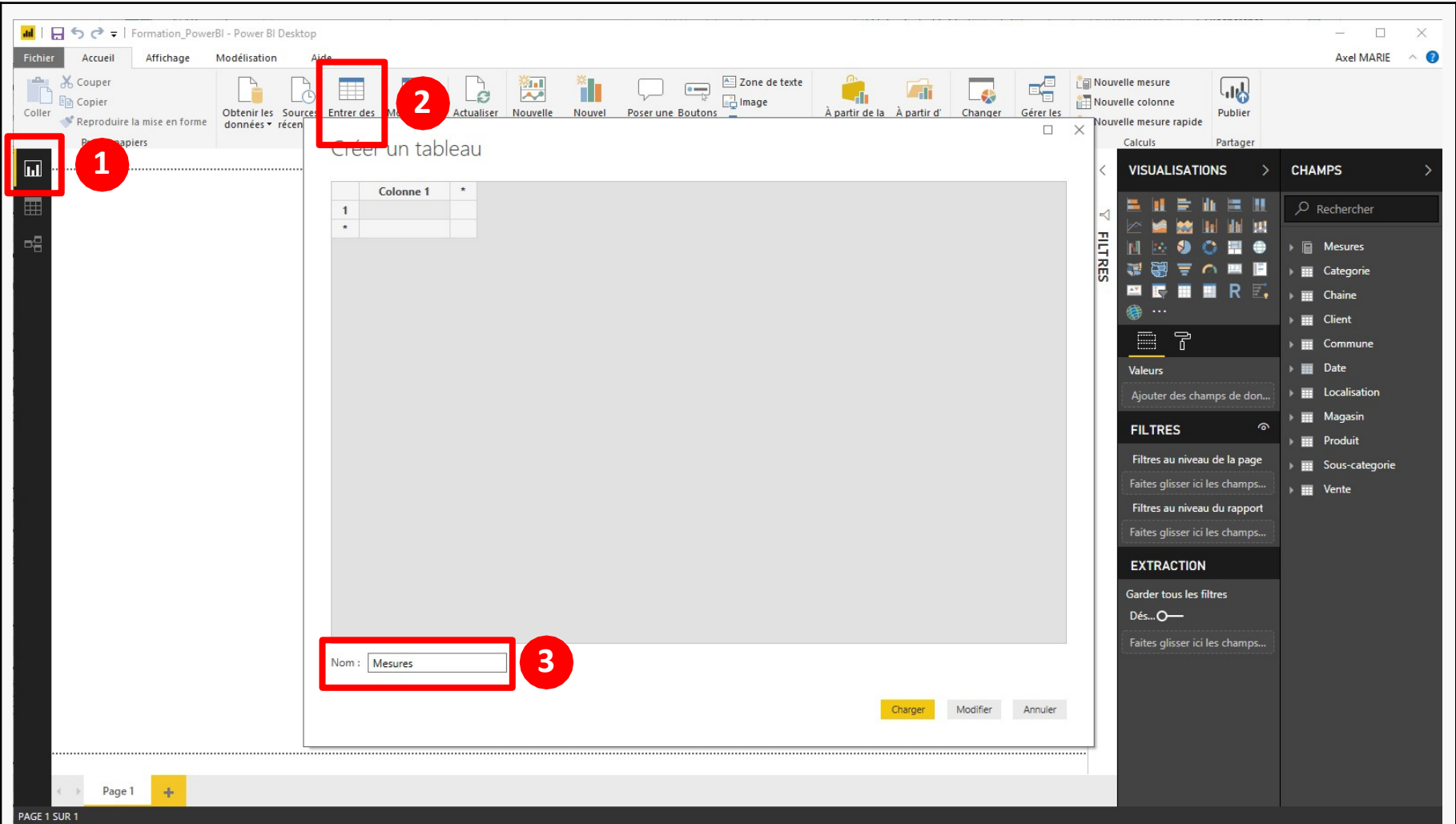
OK Annuler

# Créer une table de mesures

1. Placez-vous sur la vue **Rapport**.

2. Dans le ruban **Accueil**, cliquez sur **Entrer des données**.

2. Nommez la table **Mesures** et cliquez sur **Charger**.



# Créer une mesure Volume vente

1.Sélectionnez la table **Mesure**. **Supprimez la Colonne 1**.

2.Dans le ruban **Accueil**, cliquez sur **Nouvelle mesure**. Entrez la formule suivante :

**VolumeVente** =  
DISTINCTCOUNT(  
Vente[ID\_Vente] )

3.Ajoutez un visuel de type **Carte** et faites un drag and drop de la mesure pour apprécier le résultat.

The screenshot shows the Power BI Desktop interface. The ribbon at the top includes 'Fichier', 'Accueil', 'Affichage', 'Modélisation', 'Aide', 'Format', and 'Données / Explorer'. The 'Données / Explorer' tab is active, showing a table named 'Vente' with columns 'ID\_Vente' and 'ID\_Client'. A new measure is being created, with the formula 'VolumeVente = DISTINCTCOUNT ( Vente[ID\_Vente] )' entered in the 'Formules de mesures' pane. A red box highlights the formula bar, and a red circle with the number '2' is next to it. In the center of the workspace, a card visual displays the value '30000' for 'VolumeVente'. A red dashed box surrounds the card, and a red arrow points from the 'Mesures' section in the 'CHAMPS' pane to the card. The 'CHAMPS' pane on the right shows the 'Mesures' section expanded, with 'VolumeVente' listed under 'Mesures'. A red box highlights 'VolumeVente', and a red circle with the number '1' is next to it. The 'VISUALISATIONS' pane on the left shows the 'Carte' (Card) visual selected. A red circle with the number '3' is next to the 'Carte' icon. The status bar at the bottom indicates 'Page 1 SUR 1'.

# Créer une mesure Vol. vente J-1

1. Dans le ruban **Accueil**, cliquez sur **Nouvelle mesure**.

2. Dans la barre de formule, entrez :

```
VolumeVenteJ-1 =  
CALCULATE(  
    DISTINCTCOUNT(  
        Vente[ID_Vente] );  
    PREVIOUSDAY(  
        Vente[Date] )  
)
```

The screenshot shows the Power BI Desktop interface. The ribbon 'Accueil' is active, and the 'Nouvelle mesure' button is highlighted with a red box and a red circle with the number 1. The formula bar shows the DAX formula for 'VolumeVenteJ-1' with a red box and a red circle with the number 2. The visual shows a table with columns 'Nombre de ID\_Vente', 'Date', and 'VolumeVenteJ-1'. The first row shows 30000 for the date 'samedi 1 janvier 2011'.

Nombre de ID_Vente	Date	VolumeVenteJ-1
2453	samedi 1 janvier 2011	2453
2461	dimanche 2 janvier 2011	2461
2365	lundi 3 janvier 2011	2365
2378	mardi 4 janvier 2011	2378
2287	mercredi 5 janvier 2011	2287
2369	jeudi 6 janvier 2011	2287
2490	vendredi 7 janvier 2011	2369
2231	samedi 8 janvier 2011	2490
2368	dimanche 9 janvier 2011	2231
2381	lundi 10 janvier 2011	2368
2516	mardi 11 janvier 2011	2381
2306	mercredi 12 janvier 2011	2516
1395	jeudi 13 janvier 2011	2306
30000		



# Créer une mesure Evo. vol. vente

1. Cliquez sur **Nouvelle mesure**.

2. Dans la barre de formule, entrez :

**EvoVolVente =**

VAR VolVente = [VolumeVente]

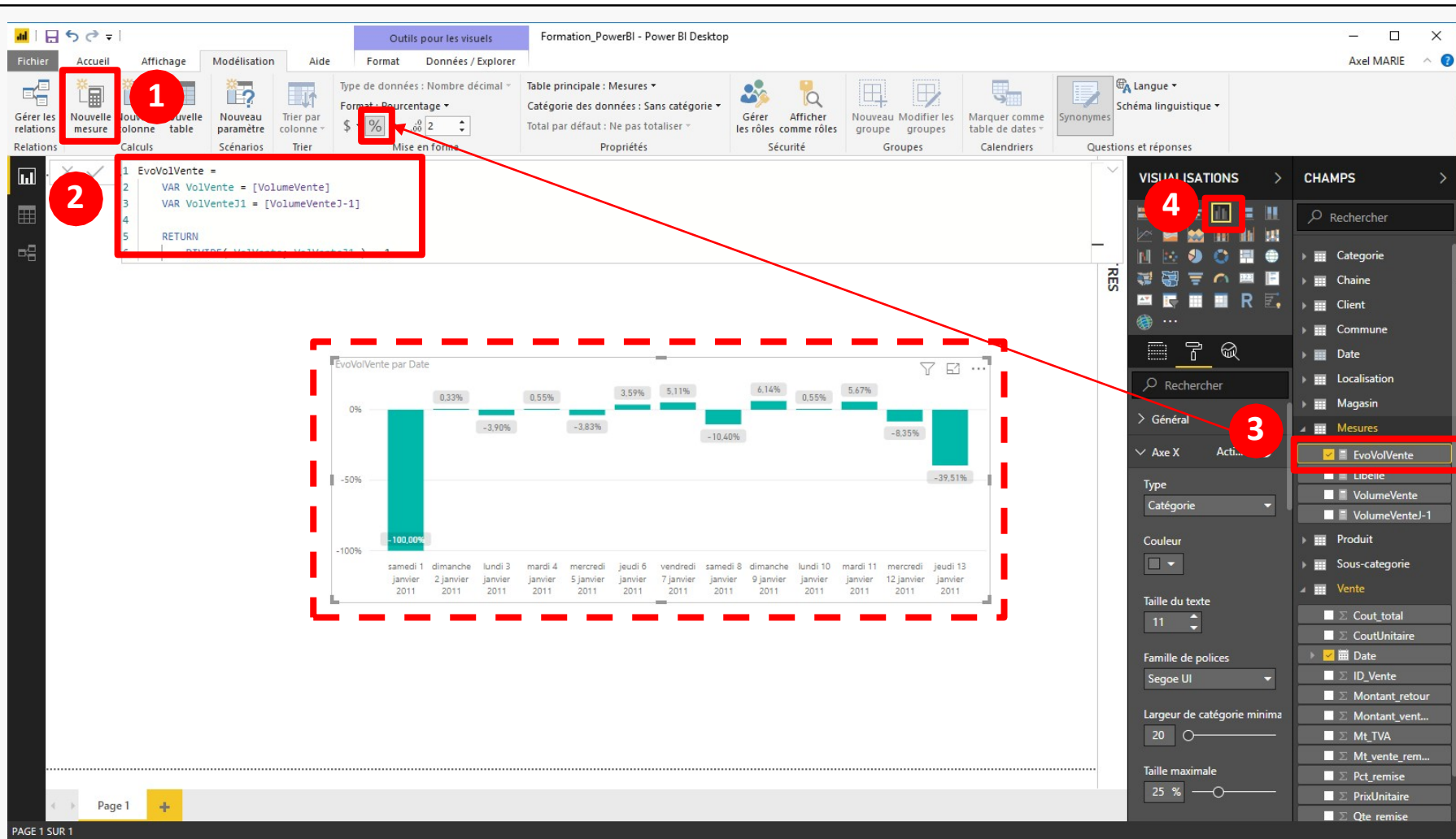
VAR VolVenteJ1 = [VolumeVenteJ-1]

RETURN

DIVIDE( VolVente; VolVenteJ1 ) - 1

3. Sélectionnez la mesure puis, dans le ruban **Modélisation**, cliquez sur l'icône du %.

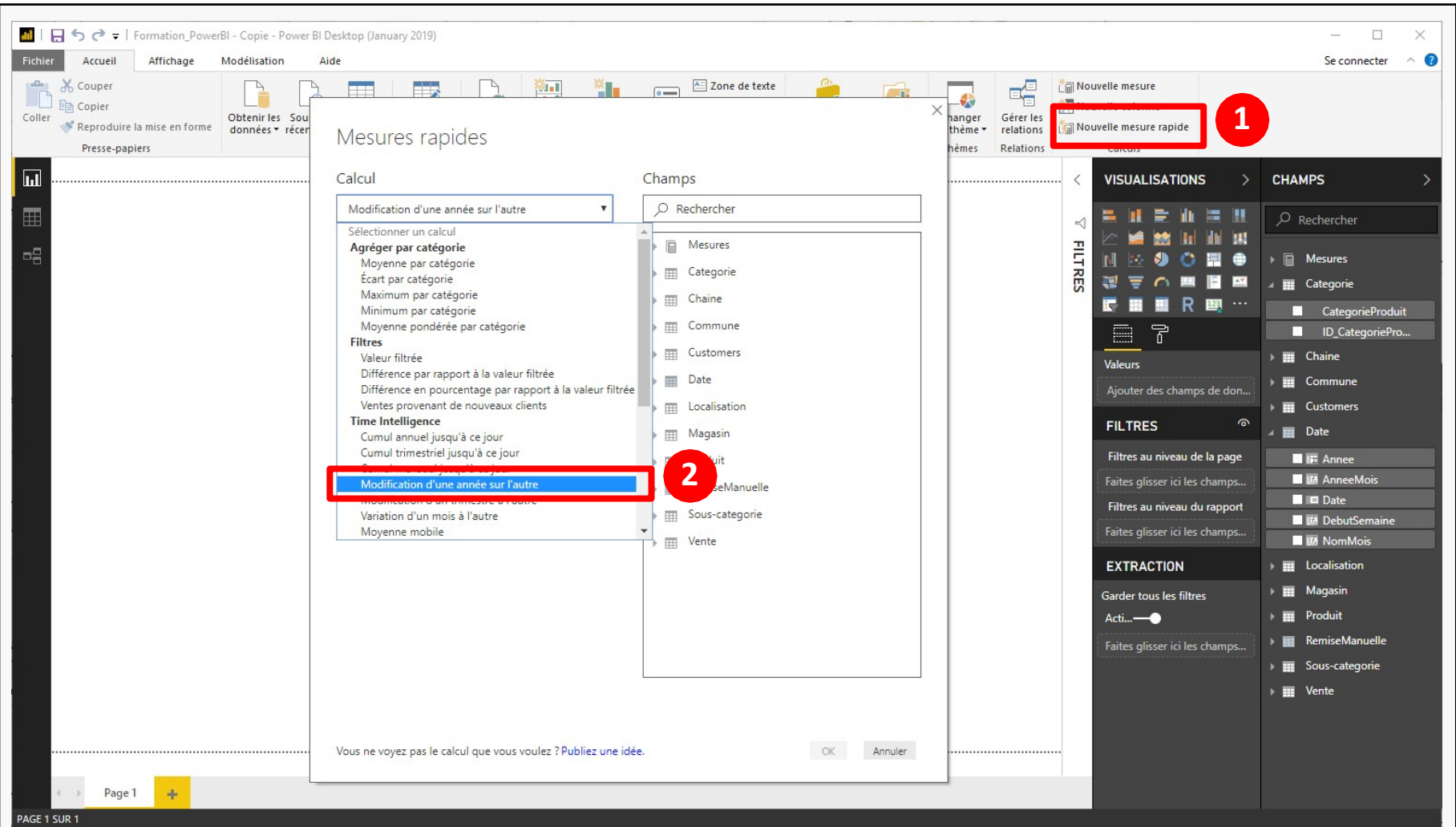
4. Ajoutez un histogramme groupé pour apprécier le résultat.



# Evo. vol. vente avec *une mesure rapide*

1. Dans le ruban **Accueil**, cliquez sur **Nouvelle mesure rapide**.

2. Dans la fenêtre qui s'ouvre, choisissez **Modification d'une année sur l'autre** dans la liste déroulante **Calcul**.

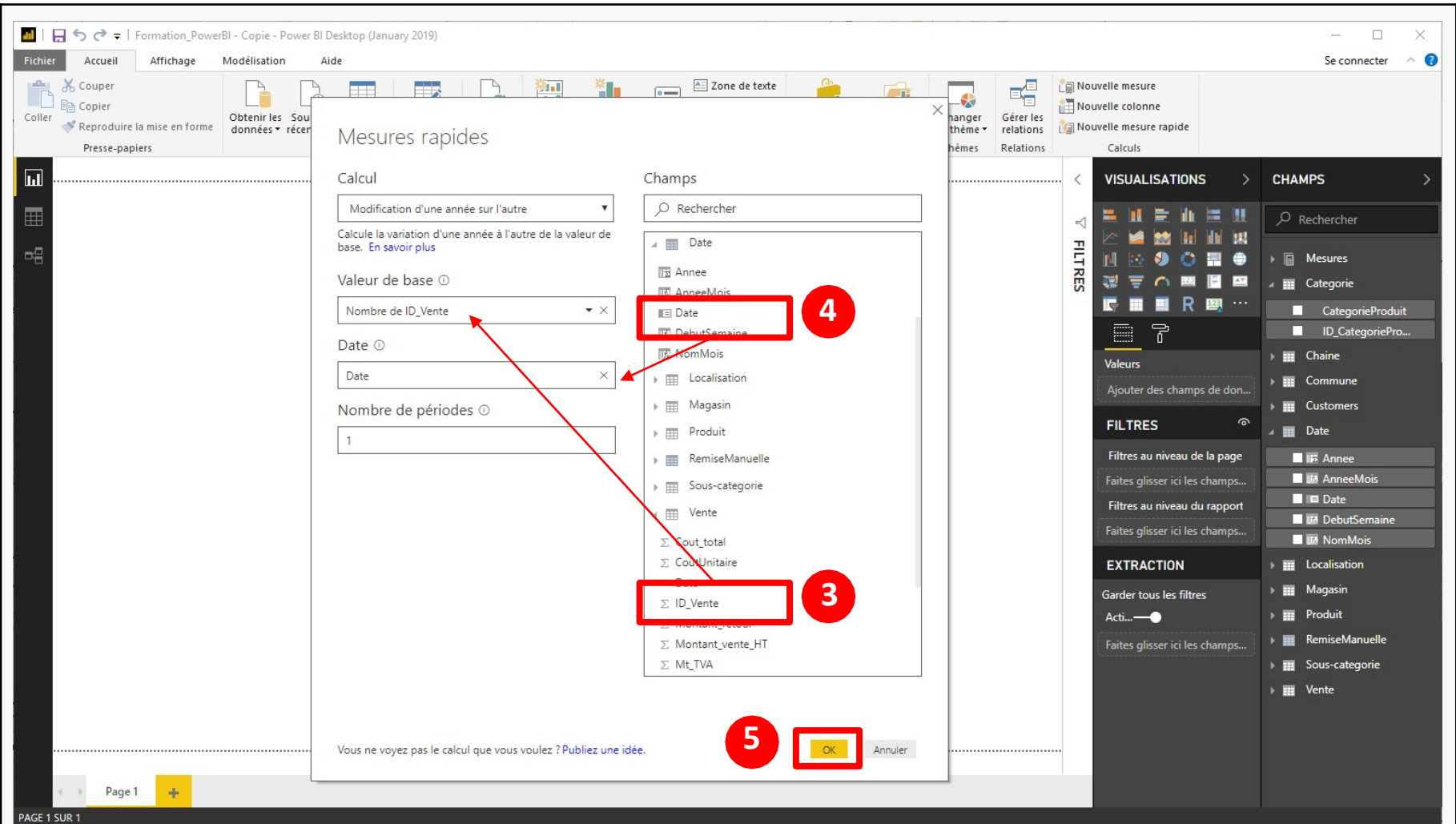


# Evo. vol. vente avec *une mesure rapide*

3. Glissez le champ **Vente[ID\_Vente]** en *Valeur de base*. Faites un clic-droit sur le champ et choisissez **Nombre** (éléments distincts).

4. Glissez le champ **Date[Date]** en *Date*.

5. Acceptez.



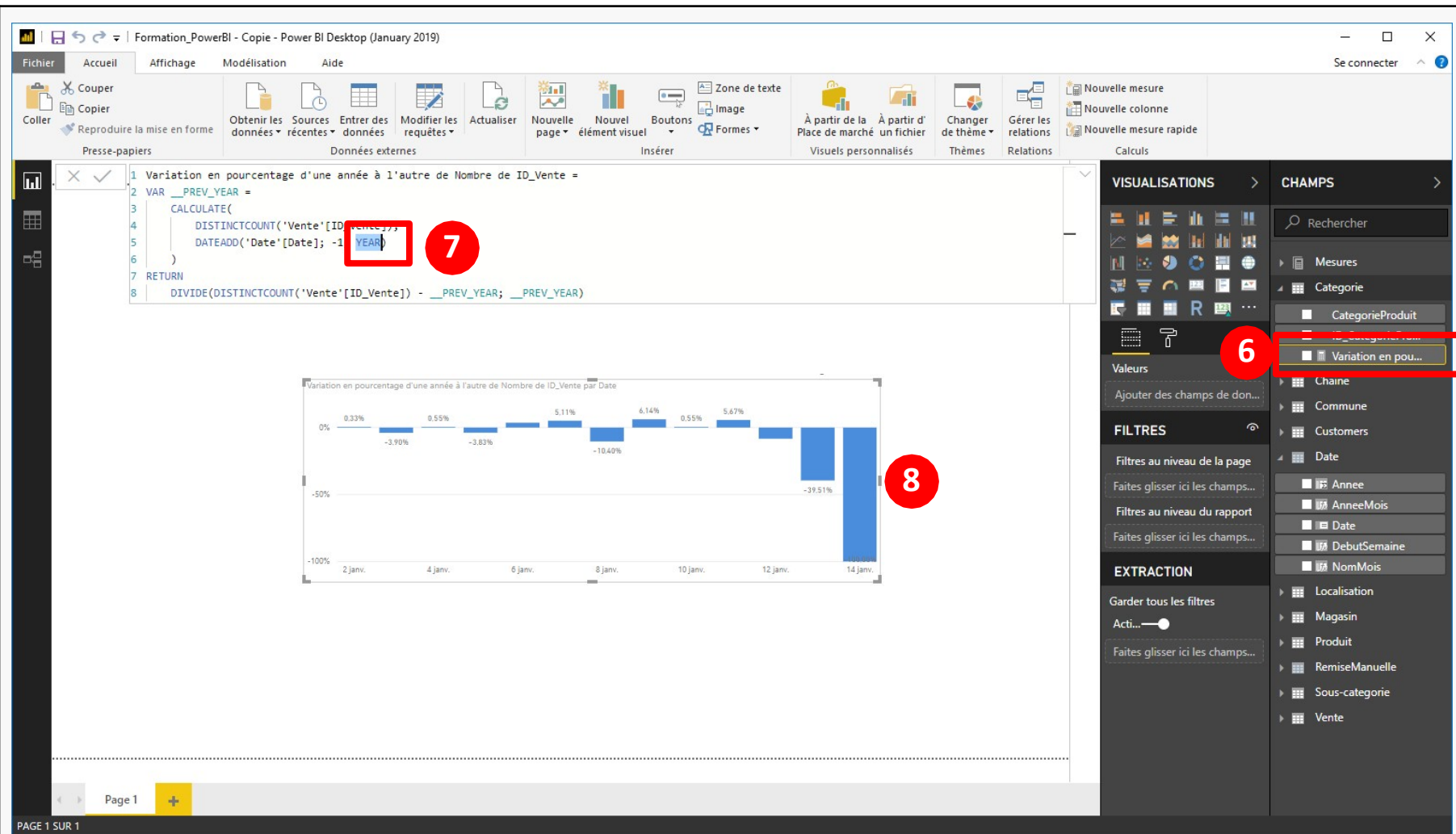


# Evo. vol. vente avec *une mesure rapide*

6. Cliquez sur la mesure nouvellement créée.

7. Dans la barre de formule, modifiez **YEAR** par **DAY**.

8. Créez un visuel pour juger du résultat.



# Récupérer le choix de l'utilisateur

1. Dans le ruban **Accueil**, cliquez sur **Nouvelle mesure**.

2. Dans la barre de formule, créez une mesure **Libelle** utilisant les variables ci-contre.

3. Ajoutez un visuel de type **Carte** pour afficher la mesure.

The screenshot shows the Power BI Desktop interface. The ribbon is set to 'Accueil' (Home). The 'Nouvelle mesure' (New measure) button is highlighted with a red box and a red circle with the number 1. The formula bar shows the following DAX code:

```
1 Libelle =  
2 VAR DateSelect = IF(  
3     ISFILTERED( Vente[Date] );  
4     DISTINCT( ALLSELECTED( Vente[Date] ) );  
5     FIRSTNONBLANK( Vente[Date]; 1 )  
6 )  
7  
8 VAR MontantVente = FORMAT(  
9     SUM( Vente[Mt_vente_remisé] );  
10    "#00,000"  
11 )  
12  
13 RETURN  
14 "A la date du " & DateSelect & " la société a réalisé " & MontantVente & "€ de chiffres d'affaire"
```

The formula bar is highlighted with a red box and a red circle with the number 2. Below the formula bar, a card visualization is shown with a red dashed border. The card has a filter for 'Date' set to 'mercredi 5 janvier 2011'. The main text of the card reads: 'A la date du 05/01/2011 la société a réalisé 5 807 637€ de chiffres d'affaire'. The card is labeled 'Libelle' at the bottom. The right-hand pane shows the 'CHAMPS' (Fields) list, where the 'Libelle' measure is highlighted with a red box and a red circle with the number 3.