



+ QUIZ

Version en ligne

OFFERTE !

pendant 1 an

Power BI Desktop

Renforcer, approfondir, explorer

En téléchargement



source de données
pour les exercices



exemples de rapports
Power BI

André MEYER-ROUSSILHON



Solutions Business



Introduction

Chapitre 1

Récupérer les données

A. Rappels sur l'accès aux données	7
B. Précisions sur la version du logiciel utilisée et les options activées	10
C. Cas complexes d'import de fichier	12
1. Dépivoter un tableau croisé complexe	12
2. Importer des données empilées	22
3. Importer un fichier JSON	24
D. Éléments fondamentaux du langage M	26
1. Objectif et intérêt de cette section	26
2. DAX ou M ?	27
3. Pour aller plus loin	28
4. Aperçu de la structure d'un code M	29
5. Afficher et éditer le code M	31
6. Cas concrets d'utilisation du code M	32
E. Importer des données web : le web-scraping	57
1. Extraire un tableau	57
2. Extraire une liste structurée implicite	58
3. Extraire des données JSON à l'aide d'une API	61
4. Extraire des données sur plusieurs pages à l'aide du code M	63
F. Agréger les tables	69
1. Comprendre l'agrégation	69
2. Mettre en place et utiliser la donnée agrégée	73
G. Utiliser les paramètres pour dynamiser la requête	95
1. Pourquoi utiliser les paramètres de requête	95
2. Mettre en place un paramètre	95
3. Paramètres et modèle de document	102
H. Comprendre l'actualisation incrémentielle	103
1. Comment la mettre en place ?	105
2. Limitations et réserves	109

Chapitre 2

Mettre en place le modèle de données

A. Introduction	113
B. État de l'art et bonnes pratiques	114
1. Choisir les options	114
2. Disposer les tables dans la vue Modèle et mettre en place les relations	116
3. Créer systématiquement des mesures	119
4. Masquer certains champs	121
5. Créer des dossiers pour organiser les champs	122
6. Créer une table du temps	124
7. Ramener à un modèle en étoile	128
C. Comprendre et appliquer le modèle en étoile	132
1. Description du modèle en étoile et de ses composants	132
2. Les autres modèles courants et leurs inconvénients	135
3. Précisions sur la grande table à plat	139
4. Mettre en place le modèle étoile	140

Chapitre 3

Maîtriser CALCULATE

A. Introduction	159
B. Préambule	161
1. Les mesures, leur format, leur nom.	161
2. Le contexte de filtre, le contexte de ligne.	162
3. Mise en forme des formules	162
4. Raccourcis de l'éditeur de formule DAX	163
5. Correction de formule avec les variables	165
6. Méthodologie de validation d'une formule	169
C. CALCULATE, les principes	171
1. La syntaxe	172
2. Les trois façons de modifier le contexte de filtre (remplacer, ajouter, supprimer)	174
3. L'ordre des opérations dans CALCULATE	176
4. Les arguments de filtres complexes sur une colonne	177
5. Les arguments de filtres complexes sur plusieurs colonnes	179

D. Les motifs avec ALL	182
1. #1 ratio toutes catégories	182
2. #2 ratio sur une seule des colonnes	184
3. #3 ratio sur toutes les deux colonnes (tables différentes)	186
4. #4 ratio sur toutes les deux colonnes (même table)	186
5. #5 ratio sur toutes les colonnes de la table dimension	188
6. #6 ratio sur toutes les colonnes de la table des faits	189
7. #7 récapitulatif visuel	190
E. Les autres motifs	193
1. #8 all ... values	193
2. #9 keepfilters	194
3. #10 allselected	195
4. #12 userrelationship	196
F. Les types d'analyses courantes et CALCULATE	196
1. La comparaison	197
2. Le cumul	198
3. La tendance	199
4. Le classement	200
5. La contribution	201
6. La variance (ou variation)	201
7. Pareto	203
8. La moyenne mobile	205
G. La transition de contexte	206
1. Propagation des contextes	206
2. CALCULATE implicite	209
3. Les CALCULATE imbriqués	210
4. Ordre d'évaluation dans CALCULATE	212
5. Risques liés à la transition de contexte	213
6. Une erreur courante : la référence circulaire	215

Chapitre 4

Data science, IA et Machine Learning

A. Introduction	221
B. Personas	222
C. Utiliser R (ou Python) avec Power BI	223
1. Installation (R et R studio)	224
2. R pour importer les données	226
3. R pour transformer les données	230
4. R pour créer des visualisations	233
D. IA et Machine Learning	239
1. Visualisations	239
2. AutoML (Machine Learning)	264
3. Cognitive Services	270
Conclusion	275
Index	277

A. Introduction

Le rôle central du langage DAX dans l'analyse des données n'est plus à démontrer : aussi bien pensée que soit votre source de données, aussi bien construit que soit votre modèle, il ne s'agit que des pierres sur lesquelles vous allez pouvoir fonder vos analyses, et celles-ci seront toujours basées sur des formules DAX.

Ce chapitre est centré sur LA fonction DAX par excellence : **CALCULATE**.

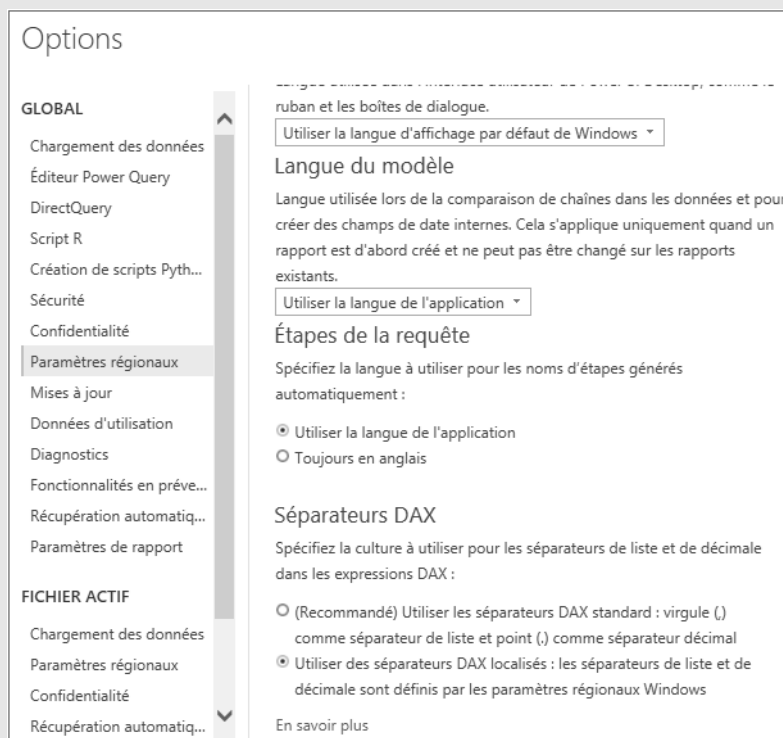
Parce que c'est la fonction pivot, celle qui demande de maîtriser les deux notions clés du DAX, à savoir les fonctions de table et le contexte de filtre et, par conséquent de maîtriser toute une batterie d'autres fonctions considérées à juste titre comme centrales.

Et parce que derrière une apparente simplicité – après tout, **CALCULATE** se contente de modifier le contexte de filtre avant de calculer une expression -, se cache un mécanisme d'une richesse et parfois d'une complexité remarquables.

Après un préambule destiné à poser les bases du travail (rappel des notions clés, debugging, validation d'un calcul), nous aborderons **CALCULATE**, nous évoquerons des modèles de formule, nous entrerons dans les arcanes de la fonction, et nous verrons comment **CALCULATE** permet de réaliser des analyses courantes.



Depuis la mise à jour de mai 2020, les séparateurs DAX sont par défaut la virgule pour séparer les listes et les arguments au sein d'une fonction, et le point comme séparateur décimal. Un système « à l'américaine », donc, par opposition aux paramètres localisés, le point-virgule pour séparer les arguments et virgule comme séparateur décimal, pour ce qui concerne la France. Nous restons dans ce livre fidèle à ces derniers. Vous pouvez passer de l'un à l'autre à l'aide du menu **Fichier - Options et paramètres - Options**, puis dans la rubrique **Global - Paramètres régionaux**, faites votre choix des séparateurs DAX :



Si vous préférez utiliser les (nouveaux) paramètres recommandés par Power BI, il vous suffira, dans tous les exemples de formules qui suivent, de remplacer le point-virgule par la virgule.

B. Préambule



Cette section traite d'un ensemble de points fondamentaux et généraux de DAX. Vous n'y trouverez donc pas d'exercices d'application, mais je vous encourage à appliquer ces concepts et règles dans toutes vos formules DAX. Il est essentiel que vous lisiez très attentivement les pages qui suivent. Et peut-être, que vous les relisiez par la suite !

1. Les mesures, leur format, leur nom

Pour l'essentiel, les formules doivent être utilisées dans le cadre de la création de mesures, et plus rarement pour la création de colonne.

Rappelons en effet que les mesures ne prennent pas de place dans votre modèle (n'utilise pas d'espace de stockage), et garantissent donc un document plus léger et plus performant. Rappelons que les mesures ne sont calculées qu'au moment de leur utilisation, selon le contexte dans lequel elles apparaissent, et pas pour chaque ligne de la table dans laquelle, à l'inverse, une colonne est « physiquement » créée.

Rappelons également que toute donnée numérique issue de la source et destinée à être analysée (montant des ventes, quantité, résultat de test, etc.), doit être masquée, et remplacée par une mesure équivalente (`[Montant] = SUM(Ventes[Montant des ventes])`). Nous verrons plus loin le lien avec CALCULATE.

Il est par ailleurs utile, lors de la création de formule, de prendre le réflexe de formater tout de suite la mesure nouvellement créée. Enfin pour ce qui concerne le nom de la mesure, évitez les lettres accentuées dans les quelques premiers caractères (afin qu'Intellisense retrouve rapidement votre mesure).

2. Le contexte de filtre, le contexte de ligne

Il ne s'agit pas pour nous ici de reprendre intégralement ces notions, mais simplement de rappeler que le contexte de ligne est généré automatiquement lors de la création d'une colonne par le biais d'une formule, ainsi que lors de l'utilisation d'une fonction de type itérateur (en particulier les fonctions X – SUMX, etc. – et la fonction FILTER).

Le contexte de filtre, lui, est défini par les visuels (tables, graphiques, segments) présents sur le rapport (et parfois sur les autres rapports). Mais il peut aussi être manipulé par formule, et c'est précisément le rôle de la fonction CALCULATE.

Le contexte de ligne filtre la table sur laquelle il s'applique, et n'en retient qu'une ligne. L'utilisation des fonctions RELATED ou RELATEDTABLE permet cependant au contexte de ligne de se propager vers d'autres tables.

Le contexte de filtre *filtre le modèle* dans son ensemble, selon le sens de propagation de 1 à N le long des relations. L'utilisation de la fonction CROSSFILTER, ou d'une relation bi-directionnelle, permettent toutefois au filtre de se propager dans le sens N à 1.

Enfin, dans certains cas, le contexte de ligne est transformé en contexte de filtre, selon le principe de la transition de contexte. Nous reviendrons sur cette notion capitale.

3. Mise en forme des formules

Afin d'assurer la lisibilité des formules, et notamment lorsqu'elles deviennent complexes, certaines règles de mises en forme peuvent être suivies :

- Les colonnes référencées le sont toujours en précisant le nom de la table d'abord, alors que les mesures sont simplement référencées entre crochets : cela aide en particulier à mieux « voir » le CALCULATE implicite.
- Ajouter autant de commentaires que nécessaire, précédés d'un //.
- Faciliter la vue des parenthèses ouvrantes et fermantes.
- Aller à la ligne autant de fois que vous le voulez.

```
[Montant moyen facturé] =  
    AVERAGEX (  
        Date[Date] ;  
        [Montant facturé]  
    )
```

Notez aussi qu'une variante, notamment dans le cadre d'un `CALCULATE`, peut consister à décaler le point-virgule, pour mieux être en mesure de mettre en commentaire une des lignes du code :

```
[Mesure] =
    CALCULATE (
        expression
        ; filtre1
        ; filtre2
        ; filtre3
    )
```

Écrite comme ça, la formule permet de mettre en commentaire facilement (ici, le filtre 3) :

```
[Mesure] =
    CALCULATE (
        expression
        ; filtre1
        ; filtre2
//      ; filtre3
    )
```

4. Raccourcis de l'éditeur de formule DAX





L'éditeur de formule fourmille de raccourcis clavier permettant d'accélérer la saisie ou de faciliter la correction des formules DAX.

Avec un minimum de pratique, vous gagnerez un temps précieux pour remettre en forme, corriger à plusieurs endroits en même temps, intervertir les lignes, etc.


Voici quelques-uns de ces raccourcis.

Les cinq raccourcis-clavier les plus utiles

- ▶ Sélectionner toutes les occurrences du terme (ou des caractères) actuellement sélectionné (exemple : une colonne, une table ou une fonction qu'il faut remplacer partout dans la formule).






  (ou   L)

- ▶ Mettre en commentaire (ou à l'inverse, enlever la mise en commentaire)












Sélectionner la ou les lignes et  /

- ▶ Aller à la ligne avec indentation (retrait)














 

- ▶ Copier une ligne au-dessus/en-dessous
   ou 
- ▶ Valider une proposition de fonction d'Intellisense









Manipuler les lignes

- ▶ Déplacer une ligne vers le haut/le bas
  ou 
- ▶ Insérer la ligne ci-dessous
 
- ▶ Insérer la ligne ci-dessus
  
- ▶ Sélectionner la ligne actuelle
 I
- ▶ Naviguer jusqu'à une ligne en indiquant son numéro
 G (taper ensuite  pour masquer la zone de saisie du numéro de ligne)



Saisir du texte

- ▶ Insérer plusieurs curseurs aux endroits choisis en cliquant (utile pour saisir un même texte à plusieurs endroits)
 Clic
- ▶ Activer plusieurs curseurs au même endroit sur plusieurs lignes
   ou   
- ▶ Sélectionner toutes les occurrences de la sélection actuelle (exemple : une colonne, une fonction qu'il faut remplacer partout).
  L ou  
- ▶ Sélectionner le mot entier (là où se trouve le curseur) (Si une portion de texte a déjà été sélectionnée,  D sélectionne la prochaine occurrence de la même sélection, et ainsi de suite : cela permet de modifier l'ensemble des portions sélectionnées d'un coup)
 D

Commentaire et indentation

- ▶ Mettre en commentaire (ou l'inverse)
Sélectionner la ou les lignes et  /
- ▶ Augmenter l'indentation / réduire l'indentation
 / 
- ▶ Aller à la ligne sans indenter
 
- ▶ Aller à la ligne avec indentation
 

Utiliser Intellisense

- ▶ Valider une proposition

- ▶ Pour rappeler la fenêtre Intellisense
 I

5. Correction de formule avec les variables

Il n'existe pas, dans Power BI, de moteur d'analyse des formules permettant d'en arrêter l'exécution au moment où une erreur est détectée (comme c'est le cas dans Excel notamment). Pour des formules complexes, longues, ceci peut être un problème.

C'est là qu'une utilisation astucieuse des variables peut vous aider.



Les variables sont déclarées, et exécutées, au début du script. Elles sont couramment utilisées pour n'exécuter un calcul qu'une fois, même si celui-ci est appelé plusieurs fois dans la formule. Leur première utilité est donc d'améliorer la performance de la formule.

Leur deuxième raison d'être est d'améliorer la lisibilité de la formule.

Leur troisième intérêt est d'effectuer un calcul sur l'état du contexte de filtre au début de l'exécution de la formule (en effet, le calcul n'étant effectué qu'une fois, la valeur de la variable est figée pendant toute la durée de la formule). Ce point, important, est développé à la fin de cette section.



Pour illustrer ce point, je me sers d'un exemple proposé par Marco Russo dans un tutoriel de Guy In A Cube : <https://www.youtube.com/watch?v=9SV2VnYbgg4>

Nous allons commencer par voir les trois états de mise en forme de la formule : d'abord sans mise en forme, puis avec, et enfin avec l'introduction des variables.

Cette formule retourne une liste de couleurs dont le nombre varie en fonction du choix fait par l'utilisateur à l'aide d'un segment *Couleurs_seg*.

Premier état :

```
Couleurs =  
IF (  
    COUNTROWS ( DISTINCT ( 'Produit'[Couleur] ) ) > SELECTEDVALUE  
( 'Couleurs_seg'[Nombre de Couleurs] );  
    CONCATENATEX ( TOPN ( SELECTEDVALUE ( 'Couleurs_seg'[Nombre de  
Couleurs] ); VALUES ( 'Produit'[Couleur] ); 'Produit'[Couleur]; " , "  
& " et plus..." ;  
    CONCATENATEX ( DISTINCT ( 'Produit'[Couleur] ); 'Produit'[Couleur];  
    " , " )  
)
```

Vous remarquerez tout de suite que le code est très confus, donc difficile à lire, et par conséquent difficile à corriger.

Et en effet, ce code renvoie une erreur. Mais où se situe-t-elle ?

Le premier nettoyage consiste à formater le code selon les règles courantes, par exemple en le passant à la moulinette de www.daxformatter.com.