



# Chapitre 4

## VBA, DAX et le logiciel DAX Studio

### 1. Conception d'un projet

La conception d'un projet de tableau de bord à partir de Power Pivot et de DAX doit être gérée comme tout autre projet informatique de type RAD (*Rapid Application Development*).

Il faut bien définir le besoin fonctionnel. Si vous n'êtes pas l'expert métier, il faut éventuellement demander à ce dernier d'établir manuellement des exemples de tableaux qui se rapprochent du résultat attendu. C'est ce qu'on appelle la maquette.

Ensuite, vous devez valider les formules de calcul si le métier est spécifique.

Pour finir, il est nécessaire de valider le modèle de données. N'hésitez pas à faire corriger, quand c'est possible, les sources des données car souvent le pivot contient trop de données et a tendance à ralentir le traitement des requêtes.

### 2. Règles de nommage

L'auteur a déjà rencontré deux types de difficulté quant à la gestion des mesures ou champs calculés. La première concerne le nombre important de mesures. Par exemple, le projet le plus gros contenait 15 tableaux croisés dynamiques, autant de graphiques et environ 50 mesures.

L'autre type de difficulté rencontrée, c'est d'être deux à travailler sur le même tableau de bord.

Dans les deux cas, seule une organisation correcte des mesures permet de s'y retrouver.

Il n'existe pas de règles de nommage universelles. Vous devez trouver celles qui vous conviennent le mieux afin de vous y retrouver.

Les règles de nommage type informatique, par exemple identifier la table d'appartenance à l'aide de trois lettres, donner des noms de zones en ne gardant que les consonnes d'un mot, et d'autres règles de ce type ne permettent pas à l'utilisateur final de comprendre le lien avec son métier.

Il est fortement conseillé de prendre l'habitude de faire précéder les noms des colonnes par leur table d'appartenance. De toute façon, Power Pivot 2016 vous oblige de plus en plus à le faire. Dans le cas contraire, vous aurez des erreurs lors de la rédaction de vos formules.

### 3. Les grandes formules par étapes

Une grande formule est une formule DAX comportant plus de dix lignes ou plus de quatre imbrications de fonctions.

Comme dans tous les langages, mais encore plus avec DAX, il est primordial de mettre au point vos grandes formules étape par étape, c'est-à-dire savoir les découper en plus petites formules et maîtriser leur résultat avant de passer au morceau suivant pour assembler le tout au final.

Les développeurs habitués le savent, le plus dur dans un développement, c'est de pouvoir déboguer son code. Dans tous les langages avancés, il existe un outil qui permet de suivre étape par étape le fonctionnement de son code et de vérifier l'état de certaines variables pendant le déroulement de l'exécution du code. Ceci permet de valider si le déroulement du code suit la logique du développeur.

Avec DAX, il n'existe pas ce genre d'outil et puis beaucoup de fonctions DAX retournent des listes de valeurs, voire des tables entières. Il est donc parfois très difficile de comprendre pourquoi la formule d'une mesure ne renvoie pas le résultat attendu.

Une des manières de faire est de découper le code de sa mesure en petits morceaux pour en maîtriser le comportement et avancer étape par étape dans la mise au point des codes complexes.

Par exemple, la fonction de calcul de la réactivité dans la colonne DIFF1 du dernier exercice de la société CONTROLETOUR ne peut pas être écrit en une seule fois.

Les avancées de la version 2016, permettent de rendre vos grandes formules plus claires en utilisant par exemple la déclaration VAR. Surtout, n'hésitez pas à les utiliser, et à ajouter des commentaires dans vos grandes formules.

Voir aussi les exercices sur l'exemple de la société SA-ELEVATOR.

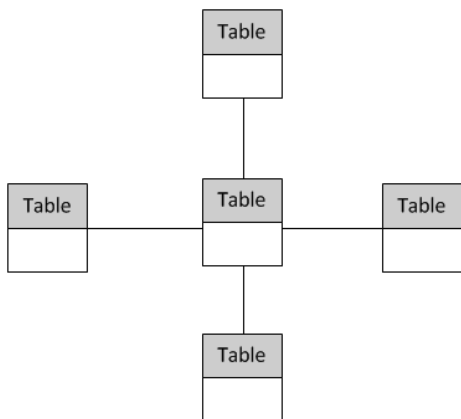
## 4. Les modèles en étoile

Pour des raisons de performance, il est fortement conseillé, sauf exception particulière, de disposer d'un modèle de base de données relationnelle dit en étoile.

Ceci signifie que votre modèle doit contenir une table principale et des tables secondaires. Il faut si possible utiliser le moins possible de tables de troisième niveau.

Vous pouvez avoir aussi dans votre modèle plusieurs étoiles.

Exemple de modèle en étoile :



## 5. Suivi des mesures en VBA

Il a été vu au chapitre Principes et définitions, à la section Organiser les mesures, qu'il est utile de suivre ses mesures à partir d'un tableau croisé pour savoir quel champ calculé utilise quels autres champs.

Dans cette section, vous découvrez succinctement comment un développeur VBA peut construire un dictionnaire de vos champs calculés et de vos tables à partir d'un simple code VBA.

Ce code est une nouveauté Excel 2016.

Pour maîtriser ce code et le faire évoluer, il est indispensable de connaître le langage de programmation d'Excel. Ce langage est appelé VBA. De nombreux ouvrages ont été rédigés pour apprendre ce langage.

### 5.1 Le besoin

Le développeur DAX souhaite avoir une feuille Excel spécifique appelée Dictionnaire dans laquelle il souhaite avoir la liste exhaustive des champs calculés de son projet, la liste des tables et la liste de leurs colonnes qui se trouvent dans le Power Pivot.

### 5.2 Résultat attendu

Le résultat est décrit dans deux tableaux. Le premier liste les mesures du projet et leurs formules et le second liste les tables et leurs colonnes :

Liste des mesures	
C1	1
C2	COUNTA([Remise])
C3	COUNTAX(FILTER(Ventes, [Qté]>5), [Remise])
C4	SUMX(FILTER(Ventes, Ventes[Remise]>3), [Remise])

Liste des tables					
Produits	IDProduct	PU	Type		
Ventes	IdFact	IdProduct	Qté	Type	Remise
Ventes2	IdFact	IdProduct	Qté	Type	Remise
Couleurs	Couleurs				
TypeProduits	Type Produit	Type			

### 5.3 Le code VBA

Mieux qu'un long discours, voici le code qui permet de calculer ces deux tableaux.

```
Sub DictionnaireMesuresetTables()
    '***Déclaration des variables Model
    Dim MonModele As Model
    Dim MesTables As ModelTables
    Dim MaTable As ModelTable
    Dim MesMesures As ModelMeasures
    Dim MaMesure As ModelMeasure
    '*** Instanciation des objets principaux
```

```
Set MonModele = ActiveWorkbook.Model
Set MesTables = MonModele.ModelTables
Set MesMesures = MonModele.ModelMeasures

'*** nettoyage de la feuille Dictionnaire
Sheets("Dictionnaire").Range("A1:Z250").Clear

'*** récupération de toutes les mesures et de leurs formules
Sheets("Dictionnaire").Cells(1, 1) = "Liste des mesures"
n = 2
For Each MaMesure In MesMesures
    NomMesure = MaMesure.Name
    FormuleMesure = MaMesure.Formula
    Sheets("Dictionnaire").Cells(n, 1) = NomMesure
    Sheets("Dictionnaire").Cells(n, 2) = FormuleMesure
    n = n + 1
Next

n = n + 1
Sheets("Dictionnaire").Cells(n, 1) = "Liste des tables"

'*** récupération des tables et de leurs structures
For Each MaTable In MesTables
    n = n + 1
    c = 2
    NomTable = MaTable.Name
    Sheets("Dictionnaire").Cells(n, c) = NomTable

    For Each MaColonne In MaTable.ModelTableColumns
        c = c + 1
        NomColonne = MaColonne.Name
        Sheets("Dictionnaire").Cells(n, c) = NomColonne
    Next

Next

End Sub
```