

A. Introduction

Avec ce nouveau chapitre, nous entrons de plain-pied dans Power Query et ses capacités à nettoyer et à transformer les données issues de la source.

Nous allons voir le fonctionnement général de Power Query, et l'articulation avec les lignes de code M.

Nous aborderons le travail au niveau des requêtes (ou tables, les termes sont synonymes), au niveau des colonnes (modifier, supprimer, ou ajouter des colonnes), la transformation des champs de type texte, numérique ou date et le traitement des tableaux croisés.

La plupart du temps, nous verrons que les opérations peuvent être réalisées à l'aide de l'interface graphique, mais aussi à l'aide du code M, et pourquoi comprendre ce code et en connaître certaines syntaxes simples permet d'accélérer le travail, et souvent de dépasser les limites de l'interface.

Les opérations que nous illustrons dans ce chapitre sont courantes, des opérations que tout utilisateur de Power Query doit connaître.

Le chapitre Exploiter la puissance de Power Query vous permettra de pousser plus loin le travail à l'aide de l'interface, avec entre autres la mise en place de tables d'agrégation, ou encore de paramètres.

Le chapitre Guide pratique du code M propose, lui, une approche plus globale de ce code, avec des éclairages sur sa structure, ses entités et ses concepts, et illustrera ces différents points à l'aide d'un ensemble de cas de mises en œuvre fréquents.

B. Fonctionnement général

L'accès à Power Query est possible à tout moment du travail dans Power BI : soit à partir du navigateur que nous avons vu au chapitre précédent (options **Transformer les données**), en d'autres termes immédiatement après la phase de connexion aux données, soit à partir de Power BI (bouton **Transformer les données** dans le ruban **Accueil**), en cours d'élaboration du rapport.

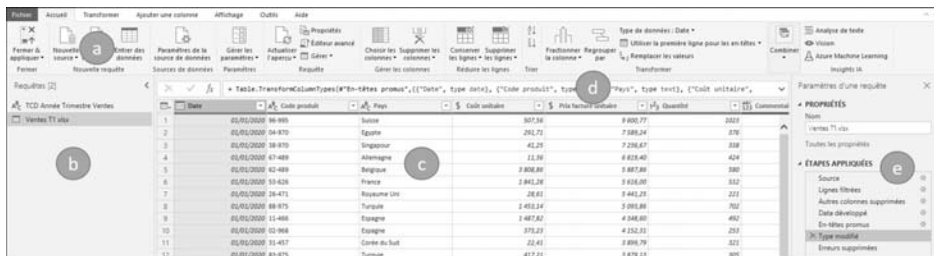
Il faut ici distinguer les sources locales (ou personnelles) et les sources d'entreprise (base de données notamment, mais aussi toute source partagée) : les premières nécessitent le plus souvent un travail de nettoyage et de préparation, ce qui est moins vrai des secondes. Mais il n'existe aucune règle définitive en la matière.

Il est utile de rappeler que le travail dans Power Query est effectué sur un échantillon de données (1000 lignes) : sur la base de cet échantillon, vous allez pouvoir définir un ensemble d'opérations (dont le nom précis est « étapes »). C'est seulement lorsque vous serez satisfait par cet ensemble d'étapes que vous allez envoyer la requête à la source, qui ramènera alors l'intégralité des lignes de la source.

Autrement dit, tout travail dans Power Query se conclut par l'activation de la fonctionnalité **Fermer & appliquer** : c'est à ce moment que vous envoyez la requête.

Il faut souligner un autre aspect de Power Query : toute étape de transformation correspond à une ligne de code M, visible dans la barre de formule. Nous allons y revenir dans la section suivante.

Pour rappel, un aperçu de l'interface graphique de Power Query :



a : l'ensemble de rubans, et dans le ruban **Accueil** la fonctionnalité **Fermer & appliquer** ;

b : les requêtes, ou tables (mais aussi, nous le verrons plus loin, les fonctions personnalisées, les paramètres, etc.).

c : la table active.

d : la barre de formule ; si vous ne voyez pas la barre de formule, rendez-vous dans le ruban **Affichage** et cochez la case **Barre de formule**.

e : les étapes.

C. Lire et comprendre le code M

1. Aperçu de la structure d'un code M

Dans ce chapitre, nous allons souvent alterner entre l'interface graphique et le code M : dans la présente section, je vous propose une petite introduction à ce code, sa structure, et comment l'afficher, mais uniquement dans le but de vous proposer par la suite des petites modifications souvent très simples, en général à partir de la barre de formule – comme vous pourriez le faire dans Excel.

Voici à quoi ressemble un code M résultant de transformations courantes. Les étapes apparaissent en gras dans le code ci-dessous (chacune des étapes correspond à une transformation). Les retours à la ligne ont été ajoutés dans un souci de clarté. Ce code a été généré automatiquement par l'interface graphique.

En deux mots : il s'agit d'abord d'aller chercher les données (étapes **Source** et **Feuill Sheet**), puis d'opérer une série de transformations (de l'étape **En-têtes promus** à **Lignes filtrées**), et enfin de retourner le résultat (après le **IN**) :

```

let
    Source =
Excel.Workbook(File.Contents(ChoixSource), null, true),
    Feuill_Sheet =
Source{[Item="chine",Kind="Sheet"]}[Data],
    "En-têtes promus" =
Table.PromoteHeaders(
    Feuill_Sheet,
    [PromoteAllScalars=true]),
    "Type modifié" =
Table.TransformColumnTypes(
    "En-têtes promus",
    {{"date", type date},
    {"heure", type time},
    {"confirmés", Int64.Type}}),
    "Index ajouté" =
Table.AddIndexColumn("Type modifié", "Index", 1, 1),
    "Personnalisée ajoutée" =
Table.AddColumn(
    "Index ajouté",
    "Date heure",
    each [date] & [heure]),
    "Lignes filtrées" =
Table.SelectRows(
    "Personnalisée ajoutée",
    each ([date] <> null))
in
    "Lignes filtrées"

```

Extraire et préparer les données en vue de leur exploitation dans Excel ou Power BI

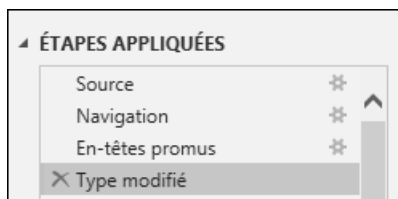
De manière plus générale (et peut-être plus lisible), la structure du code est le plus souvent la suivante :

```
let
    Source = <nom de la source> ,
    // cette ligne est un commentaire
    #"Etape 1" = FonctionM (Source, paramètres) ,
    #"Etape 2" = FonctionM (#"Etape 1", paramètres) ,
    #"Etape finale" = FonctionM (#"Etape 2", paramètres)
in
    #"Etape finale"
```

Le mot-clé LET (en minuscules) permet d'énoncer une série d'opérations tandis que le mot-clé IN (en minuscules) indique ce qui ressortira. C'est le plus souvent la dernière étape, mais pas nécessairement : il est possible d'invoquer une étape antérieure, par exemple pour corriger le code.

Comme vous pouvez le constater, l'élément clé est l'étape.

Elle porte un nom et appelle une fonction à laquelle elle passe des paramètres. Dans l'interface de Power Query, vous retrouvez le nom des étapes dans la rubrique ÉTAPES APPLIQUÉES du volet Paramètres d'une requête :



Dans cet exemple, quatre étapes ont été appliquées

Vous retrouvez le code dans la barre de formule :

Si le nom de l'étape est un seul mot, il apparaît tel quel : *Source*, *Feuil1_Sheet*. S'il est composé de plusieurs termes, il apparaît encadré de guillemets et précédé d'un dièse : *#"Etape 2"*.



Attention, M est sensible à la casse : il faut impérativement respecter les minuscules et les majuscules (le plus souvent au début du nom de la fonction).

Autres remarques sur le code visible ci-dessus :

- ▶ Parmi les paramètres d'une fonction, vous retrouvez quasi systématiquement une référence à une autre étape, le plus souvent celle qui précède.
- ▶ Pour insérer un commentaire, faite précéder la ligne d'un double slash //.
- ▶ Pour insérer un commentaire sur plusieurs lignes, vous pouvez aussi utiliser la syntaxe /* ... */.

2. Afficher et éditer le code M

Bien que vous puissiez éditer le code directement dans la barre de formule, la façon la plus efficace d'afficher et de modifier le code M est d'utiliser l'éditeur avancé (accessible depuis l'onglet Accueil), fenêtre dans laquelle l'intégralité du code (c'est-à-dire toutes les étapes) est visible :



Plutôt que d'un éditeur avancé, il faudrait plutôt parler d'un éditeur rudimentaire pour utilisateur avancé! En effet, vous n'y trouverez que peu de coloration syntaxique (mais tout de même bien utile), et une aide à l'écriture (M IntelliSense).

Moyennant quoi les codeurs **experts** utilisent souvent Notepad++ avec l'extension M de Lars Schreiber, ou encore Visual Studio (version 2017 et suivantes) avec le SDK Power Query.

Extraire et préparer les données en vue de leur exploitation dans Excel ou Power BI

Ce serait trop pour notre approche du M, aussi je me contenterai dans la suite de cet ouvrage de la barre de formule ou de l'éditeur avancé.



En termes de coloration syntaxique, le rouge indique une zone de texte (avec ses guillemets), le bleu turquoise un type de donnée, le bleu foncé un mot-clé. Tout le reste est noir.

D. Transformer les colonnes



Dans chacune des sections de ce chapitre, nous allons reprendre les fichiers que nous avons commencé à créer au chapitre précédent. Vous pouvez repartir de vos fichiers ou les retrouver dans le répertoire `exemples\chapitre3` disponible en téléchargement.

Transformer et nettoyer les colonnes sont les premières actions que vous réaliserez dans Power Query : cela inclut notamment supprimer les colonnes ou les lignes inutiles et vérifier ou changer le type de la colonne.

1. Conserver ou supprimer les colonnes



Dans cette section, nous utilisons le fichier `source02.pbix`.

Il existe une règle générale dans Power BI : ne ramenez que les données dont vous avez besoin. La quantité de données a un impact sur la performance de l'application, en lien avec les capacités de votre PC. Plus le fichier est léger, mieux Power BI réagira.

Conserver uniquement les colonnes nécessaires est donc la première action que vous devez faire. Cela suppose bien sûr que vous ayez une bonne idée de l'objectif que vous fixez à votre rapport. Mais rien n'est non plus définitif : vous pouvez à tout moment rajouter des colonnes ou annuler la suppression d'une colonne.

a. Conserver ou supprimer les colonnes à l'aide de l'interface

- 👉 Ouvrez le fichier `source02.pbix`.
 - 👉 Dans le ruban **Accueil** - groupe **Requêtes** cliquez sur **Transformer les données** pour lancer Power Query.
 - 👉 Dans le volet **Requêtes**, cliquez sur **Ventes** pour faire apparaître la table.
- Sur la droite de la table, des colonnes superflues (**Column10** à **Column14**) ont été extraites : nous allons les supprimer.
- 👉 En maintenant la touche **[Ctrl]** enfoncée, cliquez un à un sur les en-têtes des cinq colonnes concernées.
 - 👉 Puis effectuez un clic droit pour ouvrir le menu contextuel et cliquez sur **Supprimer les colonnes**.

ABC 123	Column10	ABC 123	Column11	ABC 123	Column12	ABC 123	Column13	ABC 123	Column14	PROPRIÉTÉ
	null		null		null		null		null	
	null		null		null		null		null	
	null		null		null		null		null	
	null		null		null		null		null	
	null		null		null		null		null	

- Copier
- ✖ Supprimer les colonnes
- Supprimer les autres colonnes
- ➕ Ajouter une colonne à partir d'ex
- Supprimer les erreurs

Les colonnes disparaissent, et une nouvelle étape (**Colonnes supprimées**) a été créée dans le volet **Paramètres d'une requête**. Le nom de l'étape est précédé d'une croix, qui permet la suppression et le retour à l'état avant transformation.

Il existe différentes manières de supprimer une ou plusieurs colonnes :

- ▶ par clic droit, ainsi que nous venons de le voir,
- ▶ à l'aide de la fonctionnalité **Choisir les colonnes**,
- ▶ en sélectionnant les colonnes à conserver, plutôt que celles à supprimer : lorsque vous supprimez plus de colonnes que vous n'en conservez, cette option peut être plus rapide. Cette opération peut être effectuée par clic-droit (option **Supprimer les autres colonnes** du menu contextuel), ou à l'aide de la fonctionnalité **Supprimer les autres colonnes** du menu déroulant **Supprimer les colonnes**).

Voyons par exemple la deuxième méthode :

- 👉 Dans le volet **Paramètres d'une requête**, cliquez sur la croix à gauche de l'étape **Colonnes supprimées**. Ceci annule l'étape, et les colonnes réapparaissent donc.

ÉTAPES APPLIQUÉES	
Source	✖
Navigation	✖
En-têtes promus	✖
Type modifié	
✖ Colonnes supprimées	