



Documentation Technique – Étapes Power Query M (Power BI)

Bah Cheikh | Responsable Data

13 octobre 2025

Introduction Générale

Cette documentation présente en détail l'ensemble des étapes de traitement et de transformation des données réalisées dans Power BI à l'aide du langage M (Power Query). Chaque table importée depuis Azure Blob Storage est traitée via des requêtes Power Query spécifiques. Ces transformations permettent de :

- Nettoyer, typer et structurer les données.
- Préparer les jeux de données pour la visualisation.
- Consolider les sources HubSpot et Notion.
- Garantir une cohérence entre les jeux de données.
- Optimiser les performances des rapports Power BI.

Chaque section suivante est dédiée à une table, avec une description étape par étape des transformations appliquées, suivie du code M complet.

1. Table **hubspot-data-historique**

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier `hubspot-data.csv`.
- Import du fichier CSV et promotion des en-têtes.
- Remplacement du point par une virgule dans la colonne `Différence entre les appels (secondes)`.
- Transformation du type de cette colonne en `Int64`.
- Fractionnement de la colonne `Date d'activité` sur le caractère "+".
- Transformation des types pour : `id`, `Durée_Secondes`, `Résultat de l'appel`, etc.
- Suppression de la colonne `Date d'activité.2`.
- Renommage de `Date d'activité.1` en `Date d'activité`.
- Remplacement de la valeur "Inconnu" par une chaîne vide dans `Résultat de l'appel`.

Code M

```
let
Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
#"hubspot-data1" = Source{[Name="hubspot-data"]}[Data],
#"https://storagedatagrowthhackerz.blob.core.windows.net/hubspot-data/_hubspot-data.csv" =
#"hubspot-data1"{(#"Folder Path"="https://storagedatagrowthhackerz.blob.core.windows.net/
hubspot-data/",Name="hubspot-data.csv")}[Content],
#"CSV import " = Csv.Document(#"https://storagedatagrowthhackerz.blob.core.windows.net/
hubspot-data/_hubspot-data.csv",[Delimiter=";", Columns=10, Encoding=65001, QuoteStyle=
QuoteStyle.None]),
#"En-t tes promus" = Table.PromoteHeaders(#"CSV import ", [PromoteAllScalars=true]),
#"Valeur remplac e" = Table.ReplaceValue(#"En-t tes promus",".",",",Replacer.ReplaceText
,{"Différence entre les appels (secondes)"}),
#"Type modifi " = Table.TransformColumnTypes(#"Valeur remplac e",{("Différence entre
les appels (secondes)", Int64.Type)}),
#"Fractionner la colonne par d limiteur" = Table.SplitColumn(#"Type modifi ", "Date
d activit ", Splitter.SplitTextByDelimiter("+", QuoteStyle.Csv), {"Date d activit
.1", "Date d activit .2"}),
#"Type modifi 1" = Table.TransformColumnTypes(#"Fractionner la colonne par d limiteur
",{("id", Int64.Type), {"Durée_Secondes", Int64.Type}, {"Résultat de l'appel", type text
}, {"Date d activit .1", type datetime}, {"Date d activit .2", type time}, {"
Activit attribue ", type text}, {"Jour", type text}, {"Heure", Int64.Type}, {"Minute
", type time}, {"Crneau Horaire", type text})),
#"Colonnes supprimees" = Table.RemoveColumns(#"Type modifi 1,{"Date d activit .2"}),
#"Colonnes renomm es" = Table.RenameColumns(#"Colonnes supprimees",{("Date d activit
.1", "Date d activit ")}),
#"Valeur remplac e1" = Table.ReplaceValue(#"Colonnes renomm es","Inconnu","",Replacer.
ReplaceText,{"Résultat de l'appel"}))
in
#"Valeur remplac e1"
```

2. Table **hubspot-data-latest**

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier `hubspot-data-latest.csv`.
- Import du fichier CSV et promotion des en-têtes.
- Transformation des types des colonnes, incluant : `id`, `Durée_Secondes`, `Date d'activité` (au format `datetimezone`), etc.
- Nettoyage de la colonne `Résultat de l'appel` (remplacement de "Inconnu" par une chaîne vide).
- Conversion de la colonne `Date d'activité` en texte, puis découpe sur le séparateur "+" pour séparer date et heure.
- Transformation des colonnes en types `datetime` et `time`.
- Suppression de `Date d'activité.2` et renommage en `Date d'activité`.
- Remplacement du point par une virgule dans `Différence entre les appels (secondes)`.
- Transformation finale de cette dernière colonne en type entier.

Code M

```
let
Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
#"hubspot-data-latest1" = Source[{Name="hubspot-data-latest"}][Data],
#"https://storagedatagrowthhackerz blob core windows net/hubspot-data-latest/_hubspot-data-latest csv" = #"hubspot-data-latest1"({#"Folder Path"="https://storagedatagrowthhackerz.blob.core.windows.net/hubspot-data-latest/",Name="hubspot-data-latest.csv"})[Content],
#"CSV import " = Csv.Document(#"https://storagedatagrowthhackerz blob core windows net/hubspot-data-latest/_hubspot-data-latest csv",[Delimiter=",", Columns=10, Encoding=65001, QuoteStyle=QuoteStyle.None]),
#"En-t tes promus" = Table.PromoteHeaders(#"CSV import ", [PromoteAllScalars=true]),
#"Type modifi " = Table.TransformColumnTypes(#"En-t tes promus",{{"id", Int64.Type}, {"Duree_Secondes", Int64.Type}, {"R sultat de l'appel", type text}, {"Date d activit ", type datetimezone}, {"Activit attribue ", type text}, {"Jour", type text}, {"Heure ", Int64.Type}, {"Minute", type time}, {"Cr neau Horaire", type text}, {"Difference entre les appels (secondes)", type text}}),
#"Valeur remplac e" = Table.ReplaceValue(#"Type modifi ", "Inconnu", "", Replacer.ReplaceText, {"R sultat de l'appel"}),
#"Fractionner la colonne par d limiteur" = Table.SplitColumn(Table.TransformColumnTypes(#"Valeur remplac e", {{"Date d activit ", type text}}, "fr-FR"), "Date d activit ", Splitter.SplitTextByDelimiter("+", QuoteStyle.Csv), {"Date d activit .1", "Date d activit .2"}),
#"Type modifi 1" = Table.TransformColumnTypes(#"Fractionner la colonne par d limiteur ", {{"Date d activit .1", type datetime}, {"Date d activit .2", type time}}),
#"Colonnes supprimees" = Table.RemoveColumns(#"Type modifi 1", {"Date d activit .2"}),
#"Colonnes renommees" = Table.RenameColumns(#"Colonnes supprimees", {{"Date d activit .1", "Date d activit "}}),
#"Valeur remplac e1" = Table.ReplaceValue(#"Colonnes renommees", ".", "", Replacer.ReplaceText, {"Difference entre les appels (secondes)"}),
#"Type modifi 2" = Table.TransformColumnTypes(#"Valeur remplac e1", {{"Difference entre les appels (secondes)", Int64.Type}}),
#"Lignes filtres" = Table.SelectRows(#"Type modifi 2", each true)
in
#"Lignes filtres"
```

3. Table data_Julian

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier data_julian.csv.
- Import du fichier CSV et promotion des en-têtes.
- Transformation des types des colonnes : id, Durée_Secondes, Date d'activité, etc.
- Remplacement du point par une virgule dans la colonne Différence entre les appels (secondes).
- Conversion de cette colonne en Int64.
- Remplacement de la valeur "Sébastien Pichs" par "Julien LE CLOIREC" dans la colonne Activité attribuée à.

Code M

```
let
  Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
  #"hubspot-data1" = Source{[Name="hubspot-data"]}[Data],
  #"https://storagedatagrowthhackerz blob core windows net/hubspot-data/_data_julian csv" =
  #"hubspot-data1"{{#"Folder Path"="https://storagedatagrowthhackerz.blob.core.windows.net/
  hubspot-data/",Name="data_julian.csv"}}[Content],
  #"CSV import " = Csv.Document(#"https://storagedatagrowthhackerz blob core windows net/
  hubspot-data/_data_julian csv",[Delimiter="," Columns=10, Encoding=65001, QuoteStyle=
  QuoteStyle.None]),
  #"En-t tes promus" = Table.PromoteHeaders(#"CSV import ", [PromoteAllScalars=true]),
  #"Type modifi " = Table.TransformColumnTypes(#"En-t tes promus",{{"id", Int64.Type}, {"
  Dure_Secondes", Int64.Type}, {"R sultat de l'appel", type text}, {"Date d activit ",
  type datetime}, {"Activit attribue ", type text}, {"Jour", type text}, {"Heure",
  Int64.Type}, {"Minute", type time}, {"Cr neau Horaire", type text}, {"Diff rence entre
  les appels (secondes)", type text}}),
  #"Valeur remplac e" = Table.ReplaceValue(#"Type modifi ",".",",",Replacer.ReplaceText,{"
  Diff rence entre les appels (secondes)"}),
  #"Type modifi 1" = Table.TransformColumnTypes(#"Valeur remplac e",{{"Diff rence entre
  les appels (secondes)", Int64.Type}}),
  #"Valeur remplac e1" = Table.ReplaceValue(#"Type modifi 1","S bastien Pichs","Julien LE
  CLOIREC",Replacer.ReplaceText,{"Activit attribue "})
in
  #"Valeur remplac e1"
```

4. Table **hubspot-data**

Étapes de transformation

- Fusion des deux tables précédentes : `hubspot-data-historique` et `hubspot-data-latest` à l'aide de `'Table.Combine'`.
- Tri des lignes par la colonne `Date d'activité` en ordre décroissant.
- Ajout des lignes provenant de la table `data_Julian` pour enrichir l'historique.
- Résultat : une table unique, ordonnée, consolidée, prête pour analyse.

Code M

```
let
  Source = Table.Combine({#"hubspot-data-historique", #"hubspot-data-latest"}),
  #"Lignes tri es" = Table.Sort(Source,{{"Date d activit ", Order.Descending}}),
  #"Requ te ajout e" = Table.Combine({#"Lignes tri es", data_Julian})
in
  #"Requ te ajout e"
```

5. Table **rdvs**

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier `RDVs.csv`.
- Import du fichier CSV et promotion des en-têtes.
- Transformation des types des colonnes : `DateTime_RDV`, `Date_generation`, etc.
- Suppression des colonnes non utilisées : `Date_RDV`, `Intitulé_poste`.
- Remplacement des valeurs vides dans la colonne `Source_RDV` par "Inconnu".

- Normalisation de l’orthographe du prénom “Élodie” → “Elodie” dans la colonne Responsable.
- Transformation du champ Date_generation en type date.

Code M

```
let
Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
rdvs1 = Source([Name="rdvs"])[Data],
#"https://storagedatagrowthhackerz.blob.core.windows.net/rdvs/_RDVs.csv" = rdvs1{[#"Folder
Path"="https://storagedatagrowthhackerz.blob.core.windows.net/rdvs/",Name="RDVs.csv"]}[
Content],
#"CSV import " = Csv.Document("#https://storagedatagrowthhackerz.blob.core.windows.net/
rdvs/_RDVs.csv",[Delimiter="," Columns=8, Encoding=65001, QuoteStyle=QuoteStyle.None]),
#"En-t tes promus" = Table.PromoteHeaders("#CSV import ", [PromoteAllScalars=true]),
#"Type modifi " = Table.TransformColumnTypes("#En-t tes promus",{{"Date_generation",
type text}, {"Date_RDV", type text}, {"DateTime_RDV", type datetime}, {"Statut", type text
}, {"Intitul_poste", type text}, {"Source_RDV", type text}, {"Responsable", type text},
{"Nom_Client", type text}}),
#"Colonnes supprime" = Table.RemoveColumns("#Type modifi ","{Date_RDV", "
Intitul_poste"}),
#"Valeur remplac e" = Table.ReplaceValue("#Colonnes supprime","","Inconnu",Replacer.
ReplaceValue,{"Source_RDV"}),
#"Valeur remplac e1" = Table.ReplaceValue("#Valeur remplac e"," lodie ","Elodie",
Replacer.ReplaceText,{"Responsable"}),
#"Type modifi 1" = Table.TransformColumnTypes("#Valeur remplac e1",{{"Date_generation",
type date}})
in
#"Type modifi 1"
```

6. Table notion-data

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier plan_de_charge.csv.
- Import du fichier CSV et promotion des en-têtes.
- Transformation des types : colonnes booléennes, entières, datetimezone.
- Suppression de la colonne End.
- Conversion de Start (datetimezone) en texte, puis séparation par “+” pour isoler la date.
- Transformation de Start.1 et Start.2 en datetime et time, respectivement.
- Suppression de Start.2 et renommage en Date & Heure Session.
- Nettoyage des valeurs dans Nom_Client (remplacement de “KodKodKod Studio Côte d’Ivoire” par “KodKodKod Studio”).
- Transformation finale de la colonne CA_GH_Session en Int64.

Code M

```
let
Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
#"notion-data1" = Source{[Name="notion-data"]}[Data],
#"https://storagedatagrowthhackerz blob core windows net/notion-data/_plan_de_charge csv"
= #"notion-data1"({#"Folder Path"="https://storagedatagrowthhackerz.blob.core.windows.net/
notion-data/",Name="plan_de_charge.csv"}[Content],
#"CSV import " = Csv.Document(#"https://storagedatagrowthhackerz blob core windows net/
notion-data/_plan_de_charge csv",[Delimiter="," Columns=15, Encoding=65001, QuoteStyle=
QuoteStyle.Csv]),
#"En-t tes promus" = Table.PromoteHeaders(#"CSV import ", [PromoteAllScalars=true]),
#"Type modifi " = Table.TransformColumnTypes(#"En-t tes promus",{{"Cible", type text},
{"Commercial", type text}, {"Appels", Int64.Type}, {"Conversations", Int64.Type}, {"RDVs",
Int64.Type}, {"Objectif_RDVs", Int64.Type}, {"Minari", type logical}, {"Bonus", type
logical}, {"Rem_genere", Int64.Type}, {"Rem_honore", Int64.Type}, {"Start", type text}, {"
End", type datetimezone}, {"Nom_Client", type text}}),
#"Colonnes supprime" = Table.RemoveColumns(#"Type modifi ",{"End"}),
#"Type modifi 1" = Table.TransformColumnTypes(#"Colonnes supprime",{{"Start", type
datetimezone}}),
#"Fractionner la colonne par d limiteur" = Table.SplitColumn(Table.TransformColumnTypes
(#"Type modifi 1", {{"Start", type text}}, "fr-FR"), "Start", Splitter.
SplitTextByDelimiter("+", QuoteStyle.Csv), {"Start.1", "Start.2"}),
#"Type modifi 2" = Table.TransformColumnTypes(#"Fractionner la colonne par d limiteur
",{{"Start.1", type datetime}, {"Start.2", type time}}),
#"Colonnes supprime1" = Table.RemoveColumns(#"Type modifi 2",{{"Start.2"}),
#"Colonnes renomme" = Table.RenameColumns(#"Colonnes supprime1",{{"Start.1", "Date &
Heure Session"}}),
#"Lignes filtr es" = Table.SelectRows(#"Colonnes renomme", each true),
#"Valeur remplac e" = Table.ReplaceValue(#"Lignes filtr es","KodKodKod Studio C te
d ivoire ", "KodKodKod Studio", Replacer.ReplaceText, {"Nom_Client"}),
#"Lignes filtr es1" = Table.SelectRows(#"Valeur remplac e", each true),
#"Type modifi 3" = Table.TransformColumnTypes(#"Lignes filtr es1",{{"CA_GH_Session",
Int64.Type}}),
#"Lignes filtr es2" = Table.SelectRows(#"Type modifi 3", each true)
in
#"Lignes filtr es2"
```

7. Table **image-commerciaux**

Étapes de transformation

- Connexion à Azure Blob Storage et récupération du fichier `commerciaux_blob_images.csv`.
- Import du fichier CSV brut avec deux colonnes.
- Transformation du type de chaque colonne en texte.
- Promotion de la première ligne en tant qu'en-têtes.
- Changement du nom des colonnes en `Commercial` et `Image path`.
- Filtrage des lignes valides (non nulles).

Code M

```
let
    Source = AzureStorage.Blobs("storagedatagrowthhackerz"),
    #"image-commerciaux1" = Source{[Name="image-commerciaux"]}[Data],
    #"https://storagedatagrowthhackerz blob core windows net/image-commerciaux/_commerciaux_blob_images csv" = #"image-commerciaux1"{[#"Folder Path"="https://storagedatagrowthhackerz.blob.core.windows.net/image-commerciaux/",Name="commerciaux_blob_images.csv"]}[Content],
    #"CSV import " = Csv.Document(#"https://storagedatagrowthhackerz blob core windows net/image-commerciaux/_commerciaux_blob_images csv",[Delimiter="," , Columns=2, Encoding=65001, QuoteStyle=QuoteStyle.None]),
    #"Type modifi " = Table.TransformColumnTypes(#"CSV import ",{{"Column1", type text}, {"Column2", type text}}),
    #"En-t tes promus" = Table.PromoteHeaders(#"Type modifi ", [PromoteAllScalars=true]),
    #"Type modifi 1" = Table.TransformColumnTypes(#"En-t tes promus",{{"Commercial", type text}, {"Image path", type text}}),
    #"Lignes filtres" = Table.SelectRows(#"Type modifi 1", each true)
in
    #"Lignes filtres"
```

Conclusion

Ce document a présenté en détail les étapes de transformation des données réalisées dans Power BI à l'aide du langage M, à partir des différentes sources disponibles sur Azure Blob Storage. Chaque table a été nettoyée, typée, enrichie et intégrée dans le modèle de données Power BI, garantissant ainsi :

- Une fiabilité des données et une réduction des erreurs liées à l'importation.
- Une cohérence dans les noms, les formats et les types de données.
- Une préparation optimale pour les visualisations dynamiques et les analyses avancées.
- Une extensibilité future en cas d'ajout de nouvelles sources ou de transformations supplémentaires.

L'approche modulaire utilisée permet une maintenance simplifiée et une transparence totale sur l'ensemble du pipeline de traitement. Cette documentation servira de référence pour toute mise à jour ou évolution future de l'architecture Power BI du projet Growth Hackerz.