

Rapport de ventes Heavy Power Nutrition

I. Énoncé du problème

Je suis analyste de données chez HPN, ma mission consiste à concevoir et à fournir des solutions de business intelligence de bout en bout pour l'entreprise.

Mon manager a besoin de suivre les indicateurs clés de performances (KPI) tels que les ventes nettes, les bénéfices, les commandes et retours, pour avoir un aperçu sur les produits avec les taux de retours et de remboursement les plus élevés.

Il souhaite également visualiser des informations sur des bases mensuelles, hebdomadaires, trimestrielles et annuelles.

II. Solution

Afin de répondre à ce besoin crucial, je vais concevoir un tableau de bord interactif sous Power BI. Ce dernier permettra d'effectuer une analyse approfondie de la navigation sur la page en s'attardant sur des indicateurs clés tels que les revenus, les transactions, la quantité commandée, les ventes nettes et la marge bénéficiaire. L'utilisateur aura la possibilité de visualiser ces données selon différentes granularités temporelles, allant du jour de la semaine à l'année entière. Cette flexibilité permettra une exploration et une compréhension accrues des performances commerciales, favorisant ainsi une prise de décision plus éclairée et stratégique.

II.1. Objectif

Créer un tableau de bord interactif sous Power BI pour répondre à un besoin d'analyse des performances commerciales.

Fonctionnalités du tableau de bord

✚ *Ventilation des données de navigation sur la page :*

- Revenus
- Transactions
- Quantité commandée
- Ventes nettes
- Marge bénéficiaire

✚ *Périodes d'analyse :*

- Jour de la semaine
- Mois
- Trimestre
- Année

Avantages :

- Interactivité : Exploration et filtrage des données en temps réel.
- Visibilité : Suivi des indicateurs clés de performance (KPIs) à différents niveaux de granularité.
- Prise de décision : Identification des tendances et des points d'amélioration pour optimiser les performances commerciales.

III. Réalisation

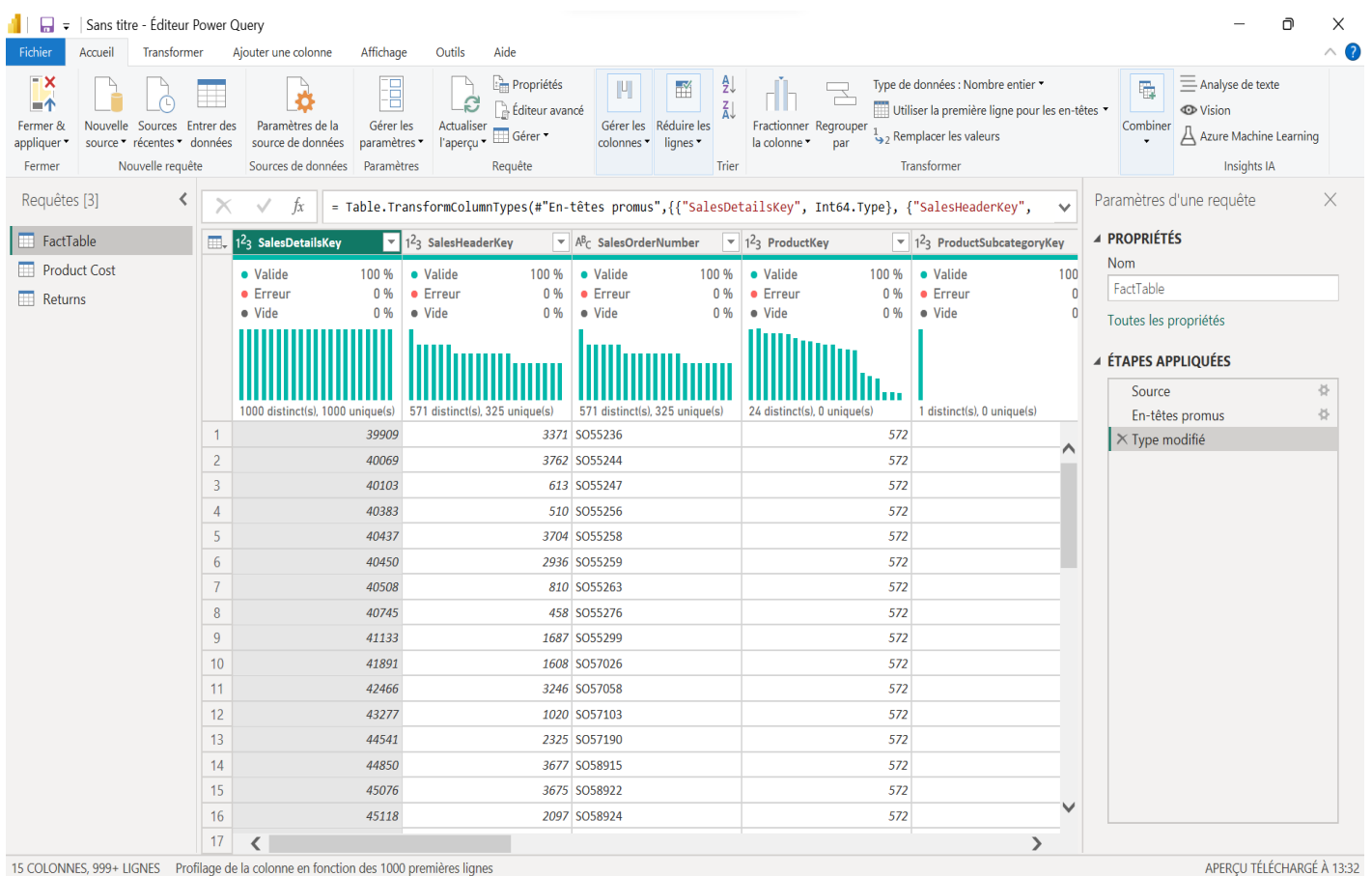
III.1. Importation de données et transformation de données

III.1.1. Les tables de faits

Les données sont toutes au format CSV. Nous disposons de trois tableaux de faits différents (« Factable , Returns et Product Cost »). Je commence par importer ces trois tables dans l'environnement Power BI les uns après les autres.

Pour explorer et procéder au nettoyage nécessaire, je choisis de transformer les données importées dans l'éditeur Power Query.

Tableau 1 : Exploration des tableaux de faits dans Power Query



III.1.2. Les autres tables

Après l'exploration initiale des tables de faits, j'ai importé et exploré les autres tables du jeu de données. Au cours de l'exploration, j'ai remarqué que la table "Products", contenant les informations sur les produits, ne possédait pas de clé primaire. Cette absence de clé primaire pouvait causer des problèmes lors de la modélisation des données et de la création de relations entre les tables, notamment pour l'identification unique de chaque produit.

Pour résoudre ce problème, j'ai procédé à la normalisation de la table "Products" en créant deux tables séparées :

- Products_Category : pour stocker les informations relatives aux catégories de produits.
- Products_Subcategory : pour stocker les informations relatives aux sous-catégories de produits.

Étapes de la normalisation :

🔧 Duplication de la table "Products" :

La table "Products" a été dupliquée deux fois. Le premier duplicata a été renommé "Products_Category" et le second duplicata, "Products_Subcategory".

🔧 Sélection des colonnes :

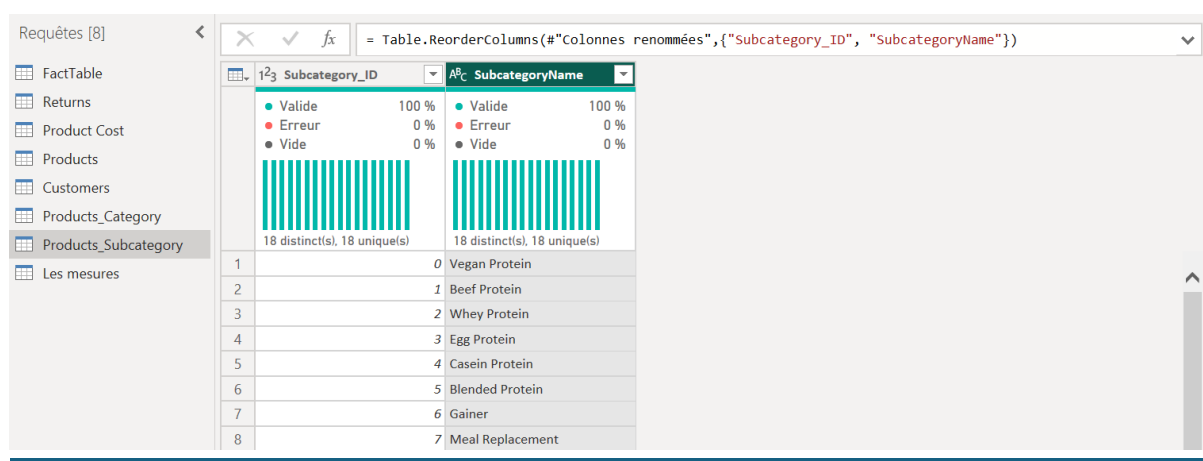
Une requête de sélection de colonne a été utilisée pour sélectionner les colonnes pertinentes dans chaque table : Colonne "CategoryName" pour la table "Products_Category" et Colonne "SubcategoryName" pour la table "Products_Subcategory".

🔧 Suppression des doublons et création de la clé primaire :

Pour chaque nouvelle table, les doublons ont été supprimés et une colonne de clé primaire a été ajoutée.

Cette normalisation a permis de supprimer la redondance des données, d'améliorer la structure du jeu de données et la création de clés primaires dans les nouvelles tables pour une identification unique et précise des catégories et des sous-catégories de produits.

Tableau 2 : Table des sous catégories créée



Subcategory_ID	SubcategoryName
0	Vegan Protein
1	Beef Protein
2	Whey Protein
3	Egg Protein
4	Casein Protein
5	Blended Protein
6	Gainer
7	Meal Replacement
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

Tableau 3 : Table des catégories créées

Requêtes [8]	= Table.ReorderColumns(#"Colonnes renommées",{ "Product_Category_ID", "CategoryName" })	
FactTable	123 Product_Category_ID	ABC CategoryName
Returns	Valide 100 %	Valide 100 %
Product Cost	Erreur 0 %	Erreur 0 %
Products	Vide 0 %	Vide 0 %
Customers	5 distinct(s), 5 unique(s)	5 distinct(s), 5 unique(s)
Products_Category	1 Protein	
Products_Subcategory	11 Carbs	
Les mesures	21 Vitamins	
	31 Clothing	
	41 Amino Acids	

III.1.3. Création de la table calendrier

De nombreuses tables comportent des colonnes de type date. Pour standardiser les dates entre ces différentes tables, j'ai utilisé la fonction CalendarAuto dans Dax pour créer une table nommée "Calendrier", en me basant sur la colonne "OrderDate" de la table "Factable". Voici l'intégralité du code utilisé pour créer cette table.

- Calendrier* =

```

VAR maxYear = YEAR(MAX('FactTable'[OrderDate]))
VAR minYear = YEAR(MIN('FactTable'[OrderDate]))
RETURN
    ADDCOLUMNS(
        FILTER(
            CALENDARAUTO(),
            YEAR([Date]) >= minYear && YEAR([Date]) <= maxYear),
        "Year", YEAR([Date]),
        "Day", DAY([Date]),
        "Month Number", MONTH([Date]),
        "Month", FORMAT([Date], "mmm"),
        "Week Number", WEEKDAY([Date]),
        "Weekday", FORMAT([Date], "ddd"),
        "Quarter", FORMAT([Date], "QQ"))

```

En plus du calendrier, j'ai ajouté une colonne distinguant les jours de semaine des week-ends avec le code suivant :

- Weekend/Semaine* =

```

SWITCH(
    TRUE(),
    'Calendrier'[Numéro de la semaine] = 1, "Weekend",
    'Calendrier'[Numéro de la semaine] = 7, "Weekend",
    "Jour ouvrable")

```

IV. Modélisation des données et création de mesures

Les différentes tâches de nettoyages et de normalisation terminées, j'ai procédé à la création de relations entre les différentes tables en utilisant les clés primaires.

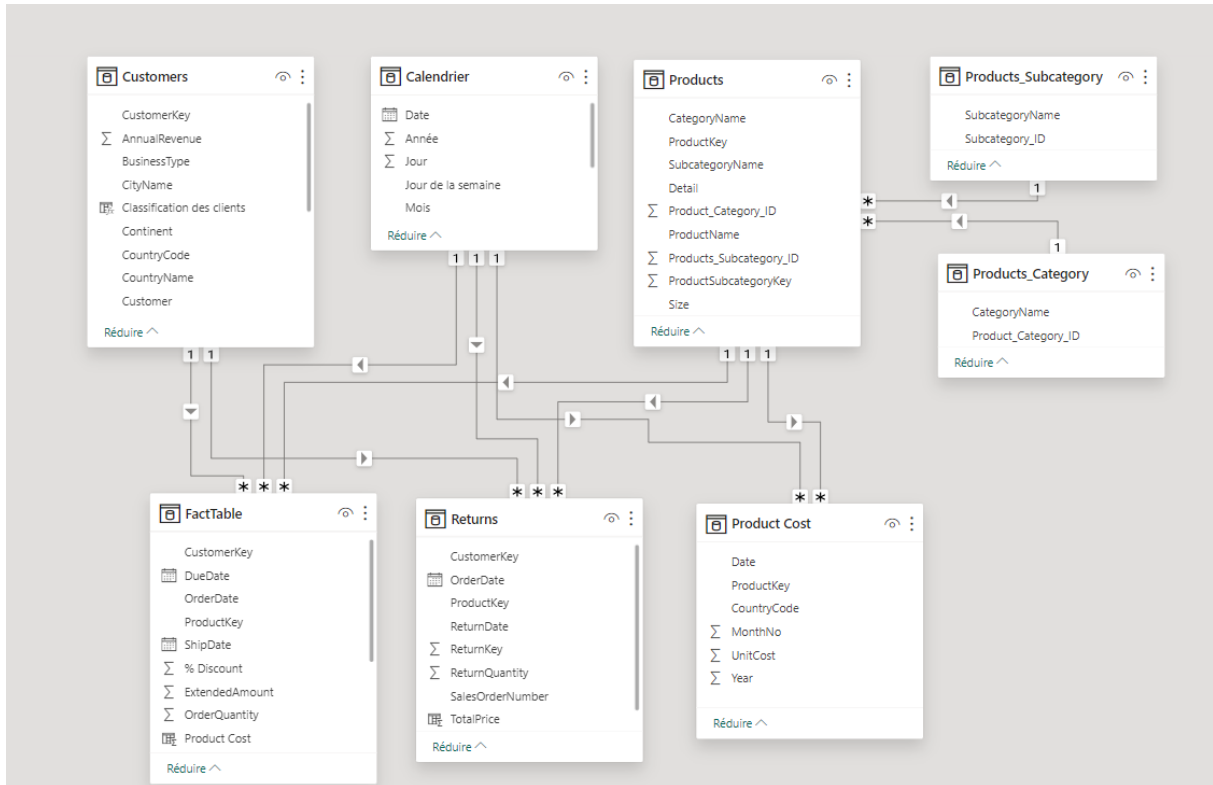


Figure 1 : Relation entre les différentes tables

Étant donné que les données fournies ne contiennent pas toutes les informations nécessaires pour répondre aux exigences du projet, j'ai élaboré diverses mesures, les voici :

Mesures relatives aux quantités

- Quantités commandées = $SUM(FactTable[OrderQuantity])$
- Quantité retournée = $SUM>Returns[ReturnQuantity])$
- Total références vendues = $DISTINCTCOUNT(FactTable[ProductKey])$
- Nombre de transactions = $COUNTROWS(FactTable)$
- Références disponible = $COUNTROWS(Products)$

Mesures relatives aux ventes et revenus

- Revenu Brut = $SUMX('FactTable', FactTable[OrderQuantity]*FactTable[UnitPrice])$
- Remboursement Total = $SUMX>Returns, Returns[ReturnQuantity] * Returns[UnitPrice])$
- Remise totale = $SUMX(FactTable, FactTable[ExtendedAmount]*FactTable[% Discount])$
- Taux de remise = $DIVIDE([Remise totale],[Revenu Brut], 0)$
- Taux de remboursement = $DIVIDE([Remboursement Total], [Revenu Brut], 0)$
- Ventes Nettes = $[Revenu Brut] - [Remboursement Total] - [Remise totale]$

- $COGS^1 = SUMX('FactTable', FactTable[Product Cost]*FactTable[OrderQuantity])$
- $Marge\ Brute = [Ventes\ Nettes] - [COGS]$

Mesures relatives aux clients et produits

- $Nombre\ de\ clients = COUNTROWS(Customers)$
- $Nombre\ de\ clients\ actifs = DISTINCTCOUNT(FactTable[CustomerKey])$
- $Nbre\ de\ transaction\ par\ client = CALCULATE(COUNTROWS(FactTable))$
- $Revenue\ par\ client = CALCULATE(SUMX('FactTable', 'FactTable'[OrderQuantity]*'FactTable'[UnitPrice]))$

A partir de ces mesures calculées sur les clients, j'ai pu les classer les clients en 4 groupes comme suite :

- Groupe Diamant : clients avec Plus de 200 transactions =
- Groupe or : clients avec plus de 100 transactions
- Groupe argent : clients avec 1 à 100 transactions
- Client inactif : si aucune transaction.

Le code utilisé pour ce classement est le suivant :

- *Classification des clients* =
`VAR Transaction_ = Customers[Nbre de transaction par client]`
`VAR Result =`
`SWITCH(TRUE(),`
`Transaction_ = BLANK(), "Client inactif",`
`Transaction_ > 200, "Diamant",`
`Transaction_ > 100, "Or",`
`"Argent")`
`RETURN Result`

Analyse des retours

Ici, j'ai effectué une analyse clé sur les produits retournés par les clients. J'ai créé des visuels dynamiques pour que les utilisateurs finaux sélectionnent les n principaux clients effectuant le plus de retours et les n produits les plus retournés en fonction de leur choix. Pour y parvenir, j'ai utilisé les codes DAX ci-dessous ainsi que la fonctionnalité de plage numérique de Power BI.

- *Classement des clients* =
`VAR Rank_Customers = RANKX(ALL(Customers[Customer]),[Quantité retournée],,DESC)`
`VAR Result =`
`IF(Rank_Customers <= [Valeur Sélectionner], [Quantité retournée])`
`RETURN`
`Result`
- *Classement des produits* =

¹ COGS = Cost Of Goods Sold (coûts des produits vendus)

```

VAR Rank_Products= RANKX(ALL(Products[ProductName]),[Quantité retournée],,DESC)
VAR Result=
IF(Rank_Products <= [Valeur Sélectionner], [Quantité retournée])
RETURN
Result

```

Création de sélections et d'intitulés dynamiques

- Sélectionner un nombre = *GENERATESERIES(1, 50, 1)*
- Valeur Sélectionner = *SELECTEDVALUE('Sélectionner un nombre'[Sélectionner un nombre], 0)*
- Nombre de clients sélectionner =
VAR Top_N = SELECTEDVALUE('Sélectionner un nombre'[Sélectionner un nombre])
VAR Result = "Les " & Top_N & " clients avec les retours les plus élevés"
Return Result
- Nombre de produits sélectionner =
VAR Top_N = SELECTEDVALUE('Sélectionner un nombre'[Sélectionner un nombre])
VAR Result = "Les " & Top_N & " produits les plus retournés"
Return Result

Pour mieux appréhender la dynamique résultante, je vous encourage à consulter le tableau de bord via Power BI Services en cliquant sur [ce lien](#).

V. Design du tableau de bord

Après avoir créé les mesures nécessaires pour concrétiser ce projet, j'ai entamé la conception du tableau de bord. Tout d'abord, j'ai élaboré l'arrière-plan du tableau de bord dans PowerPoint, puis je l'ai intégré dans Power BI. Par la suite, j'ai enrichi l'interactivité du tableau de bord en incorporant des transitions de page ainsi que des transitions de contexte.

Le tableau de bord a été déployé sur Power BI Service. Pour pleinement apprécier le fruit de ce travail et découvrir l'interactivité dynamique du tableau de bord, veuillez cliquer sur le lien ci-dessous.

[Rapport des ventes HPN](#)