

Projet Maven Market

Analyse commerciale de bout en bout avec Power BI Desktop

Un fil rouge complet : préparation des données, modélisation en étoile, mesures DAX & Time Intelligence, puis construction d'un tableau de bord interactif.

Adaptation française du projet open-source

github.com/MaanAkshay/Power-BI-Project-Maven-Market-

Jeu de données : **Maven Market** (Maven Analytics — Data Playground)

Niveau : Intermédiaire → Avancé · Durée estimée : 8 à 12 h

Sommaire

1. Contexte & objectifs pédagogiques
2. Le jeu de données Maven Market
3. Prérequis & mise en route
4. Phase 1 — Préparation des données (Power Query)
5. Phase 2 — Modélisation & relations (schéma en étoile)
6. Phase 3 — Mesures DAX & Time Intelligence
7. Phase 4 — Construction du rapport « Topline Performance »
8. Compétences révisées (correspondance programme)
9. Livrables & grille d'évaluation sur 20

1. Contexte & objectifs pédagogiques

Maven Market est une chaîne de distribution internationale (alimentaire et grande consommation) implantée aux États-Unis, au Canada et au Mexique. La direction souhaite un **tableau de bord de pilotage commercial** permettant de suivre les transactions, le chiffre d'affaires, la marge, les retours produits et la performance par marque, catégorie et région.

Au terme de ce projet, l'étudiant·e saura :

- importer et **nettoyer plusieurs sources** via Power Query (langage M) ;
- construire un **modèle en étoile** propre avec une table Calendrier dédiée ;
- écrire des **mesures DAX** robustes, dont des calculs de **Time Intelligence** ;
- concevoir un **rapport interactif** lisible, avec KPI, cartes, signets et navigation ;
- porter un **regard critique** sur la qualité du modèle et de la restitution.

À propos de ce document

Le dépôt d'origine décrit la structure du projet (4 phases) sans détailler le code. Cette version française reprend fidèlement cette structure et l'enrichit des **mesures DAX et transformations canoniques** du projet Maven Market, afin d'en faire un TP directement exploitable. Les noms de tables/colonnes suivent ceux du jeu de données officiel.

2. Le jeu de données Maven Market

Le jeu de données se télécharge gratuitement sur le **Data Playground** de Maven Analytics (rechercher « *Maven Market* »). Il se compose de plusieurs fichiers CSV :

Fichier / Table	Rôle	Champs principaux
MavenMarket_Transactions	Faits — ventes	transaction_date, product_id, customer_id, store_id, quantity
MavenMarket>Returns	Faits — retours	return_date, product_id, store_id, quantity
MavenMarket_Products	Dimension produit	product_id, product_brand, product_name, product_retail_price, product_cost
MavenMarket_Customers	Dimension client	customer_id, first/last name, birthdate, gender, city, country
MavenMarket_Stores	Dimension magasin	store_id, store_type, store_city, store_country, store_address...
MavenMarket_Regions	Dimension région	region_id, sales_district, sales_region
MavenMarket_Calendar	Dimension temps	date

Analogie NordSupply

Ce modèle est le jumeau de NordSupply : **Transactions** ≈ commandes/lignes_commande , **Products** ≈ produits , **Customers** ≈ clients , **Regions** ≈ régions commerciales. Les compétences sont directement transposables.

3. Prérequis & mise en route

- **Power BI Desktop** installé (version gratuite suffisante).
- Les fichiers CSV Maven Market enregistrés dans un dossier de travail organisé.
- Connaissances de base de l'interface (vues Rapport / Données / Modèle).

Mise en route : créer un nouveau fichier **.pbix** , puis *Accueil* → *Obtenir les données* → *Texte/CSV* pour importer chaque fichier (ou *Dossier* pour combiner les fichiers de même structure).

4. Phase 1 — Préparation des données (Power Query)

Power Query

Langage M

Nettoyage

Étape 1 — Données clients (Customers)

- Promouvoir la première ligne en en-têtes, vérifier/forcer les **types de données**.
- Créer une colonne **nom complet** en fusionnant prénom et nom.
- Extraire l'**année de naissance** à partir de `birthdate`.
- Ajouter une colonne indicatrice « **a des enfants** » (Oui/Non) selon `total_children`.

```
// Colonne personnalisée : Nom complet  
= [customer_first_name] & " " & [customer_last_name]  
  
// Colonne personnalisée : Année de naissance  
= Date.Year([birthdate])
```

Étape 2 — Données produits (Products)

- Vérifier que `product_retail_price` et `product_cost` sont bien numériques (décimal).
- Ajouter une colonne **prix réduit** (ex. remise de 10 %).

```
// Colonne personnalisée : Prix remisé (-10 %)  
= [product_retail_price] * 0.9
```

Étape 3 & 4 — Magasins (Stores) & Régions

- **Fusionner** les composantes d'adresse du magasin en une adresse complète.
- Extraire/normaliser le **code de district / région**.
- Vérifier le typage des données régionales (texte vs nombre).

Étape 5 — Table Calendrier

Enrichir la table de dates avec les colonnes temporelles nécessaires aux analyses :

- **Début de semaine, Nom du jour, Début de mois, Nom du mois, Trimestre, Année.**
- Colonne **Week-end** (« Y » / « N ») pour les analyses de fréquentation.

```
// Exemples de colonnes (éditeur Power Query)  
Début de semaine = Date.StartOfWeek([date], Day.Sunday)  
Nom du jour      = Date.DayOfWeekName([date])  
Début de mois    = Date.StartOfMonth([date])  
Nom du mois      = Date.MonthName([date])  
Trimestre        = "T" & Text.From(Date.QuarterOfYear([date]))  
Année            = Date.Year([date])
```

Étape 6 — Données de retours (Returns)

- Importer et typer la table des retours (`return_date`, `product_id`, `quantity`).

Étapes 7 & 8 — Organisation

- Renommer proprement chaque requête (préfixe cohérent), regrouper en **dossiers** de requêtes.
- *Fermer & appliquer* pour charger le tout dans le modèle.

5. Phase 2 — Modélisation & relations (schéma en étoile)

Schéma en étoile

Relations *→1

Table de dates

En **vue Modèle**, organiser les tables et créer les relations :

Table « plusieurs » (fait)	Clé	Table « un » (dimension)	Type
Transactions	product_id	Products	* → 1
Transactions	customer_id	Customers	* → 1
Transactions	store_id	Stores	* → 1
Transactions	transaction_date	Calendar	* → 1
Returns	product_id	Products	* → 1
Returns	store_id	Stores	* → 1
Returns	return_date	Calendar	* → 1
Stores	region_id	Regions	* → 1

- Marquer la table **Calendar** comme « **table de dates** » (indispensable à la Time Intelligence).
- Masquer les colonnes **clés** de la vue Rapport pour clarifier le modèle.
- Vérifier le **sens de filtrage** (des dimensions vers les faits, en simple sens).

Bonne pratique

Disposer les tables de dimension **autour** des deux tables de faits (Transactions, Returns) pour visualiser nettement le schéma en étoile (ou « flocon » léger avec Stores → Regions).

6. Phase 3 — Mesures DAX & Time Intelligence

DAX

CALCULATE

Time Intelligence

Créer une table dédiée `_Mesures` pour y regrouper toutes les mesures (bonne pratique).

6.1 — Mesures de base

```
Quantity Sold = SUM(Transactions[quantity])

Total Transactions = COUNTROWS(Transactions)

Quantity Returned = SUM>Returns[quantity])

Total Returns = COUNTROWS>Returns)

Return Rate =
DIVIDE([Quantity Returned], [Quantity Sold], "Aucune vente")
```

6.2 — Chiffre d'affaires, coût & marge

```
Total Revenue =
SUMX(Transactions,
    Transactions[quantity] * RELATED(Products[product_retail_price]))

Total Cost =
SUMX(Transactions,
    Transactions[quantity] * RELATED(Products[product_cost]))

Total Profit = [Total Revenue] - [Total Cost]

Profit Margin =
DIVIDE([Total Profit], [Total Revenue], 0)
```

6.3 — Filtres de contexte (CALCULATE / ALL)

```
Weekend Transactions =
CALCULATE([Total Transactions], Calendar[Week-end] = "Y")

% Transactions Week-end =
DIVIDE([Weekend Transactions], [Total Transactions], "n/a")

All Transactions =
CALCULATE([Total Transactions], ALL(Transactions))

% du total =
DIVIDE([Total Transactions], [All Transactions])
```

6.4 — Time Intelligence (cœur du TP)

```
// Chiffre d'affaires glissant sur 60 jours
Revenue 60 jours =
CALCULATE([Total Revenue],
    DATESINPERIOD(Calendar[date], MAX(Calendar[date]), -60, DAY))

// Mois précédent (comparaison M-1)
Last Month Transactions =
CALCULATE([Total Transactions], DATEADD(Calendar[date], -1, MONTH))

Last Month Revenue =
CALCULATE([Total Revenue], DATEADD(Calendar[date], -1, MONTH))

Last Month Profit =
CALCULATE([Total Profit], DATEADD(Calendar[date], -1, MONTH))

// Cumul annuel (Year-To-Date)
Revenue YTD =
TOTALYTD([Total Revenue], Calendar[date])

// Comparaison à l'année précédente (N-1)
Revenue N-1 =
CALCULATE([Total Revenue], SAMEPERIODLASTYEAR(Calendar[date]))
```

6.5 — Objectifs (cibles) & classement

```
// Objectif : +10 % par rapport au mois précédent
Revenue Target = [Last Month Revenue] * 1.1
Profit Target = [Last Month Profit] * 1.1

// Classement des marques par CA
Rang Marque =
RANKX(ALL(Products[product_brand]), [Total Revenue], , DESC)
```

À comprendre, pas seulement à copier

Faire expliciter par l'étudiant·e la différence **contexte de ligne** (SUMX) vs **contexte de filtre** (CALCULATE), et le rôle de **RELATED** pour traverser une relation.

7. Phase 4 — Construction du rapport « Topline Performance »

Visuels

Mise en forme conditionnelle

Signets

Page principale — « Topline Performance »

- **Titre + logo** Maven Market en en-tête (zone de texte + image).
- **Cartes KPI** : Total Transactions, Total Profit, Total Returns, avec **tendance** vs mois précédent.
- **Matrice** : transactions et profit **par marque** de produit, avec **barres de données** / mise en forme conditionnelle.

- **Carte géographique** : répartition des ventes par pays / région.
- **Treemap** : poids des catégories ou des marques.
- **Graphique en courbes / colonnes** : tendance du chiffre d'affaires dans le temps.
- **Jauge** : chiffre d'affaires réel comparé à l'**objectif** (**Revenue Target**).
- **Segments** (slicers) : année, mois, pays, catégorie.

Interactivité & navigation

- **Signets (bookmarks)** pour basculer entre plusieurs vues (ex. CA / Marge / Retours).
- **Boutons de navigation** reliés aux signets ; **info-bulles** personnalisées sur les visuels.
- Vérifier et régler les **interactions entre visuels** (filtrage croisé cohérent).

Compléments à ajouter (pour couvrir 100 % du programme)

Le projet d'origine reste léger sur trois points de ton cours : exiger en plus **(1)** un **drillthrough** vers une page de détail produit/magasin, **(2)** un **rôle RLS** (sécurité au niveau des lignes, ex. filtrage par pays), **(3)** la **publication** sur le service Power BI avec **actualisation planifiée**.

8. Compétences révisées (correspondance programme)

Notion du cours	Couverte dans le TP	Phase
Import multi-sources (CSV / dossier)	Oui	1
Power Query / langage M (nettoyage, colonnes, fusion)	Oui	1
Table Calendrier & colonnes temporelles	Oui	1-2
Modélisation en étoile & relations *→1	Oui	2
DAX de base (SUM, SUMX, COUNTROWS, DIVIDE, RELATED)	Oui	3
Contexte de filtre (CALCULATE, ALL)	Oui	3
Time Intelligence (YTD, M-1, N-1, glissant 60 j)	Oui	3
RANKX / classement & objectifs (cibles)	Oui	3
Visuels variés + mise en forme conditionnelle	Oui	4
Signets, boutons, info-bulles, interactions	Oui	4
Drillthrough	À ajouter (consigne bonus)	4
RLS (sécurité au niveau des lignes)	À ajouter (consigne bonus)	4
Publication & actualisation planifiée	À ajouter (consigne bonus)	4

9. Livrables & grille d'évaluation sur 20

Livrables attendus : le fichier **.pbix** finalisé, et une courte soutenance (5 min) commentant les choix de modélisation et 3 enseignements métier tirés du tableau de bord.

Critère	Détail	Points
Préparation des données	Typage correct, colonnes calculées, requêtes nommées/organisées	/4
Modélisation	Schéma en étoile propre, relations valides, table de dates marquée, clés masquées	/4
Mesures DAX	Mesures de base + ≥ 3 mesures de Time Intelligence fonctionnelles	/5
Rapport & visuels	KPI, carte, matrice mise en forme, lisibilité, cohérence des couleurs	/4
Interactivité	Segments, signets/boutons, interactions maîtrisées	/2
Bonus	Drillthrough · RLS · publication avec actualisation	+1
Total		/20

Document pédagogique — adaptation française du projet open-source github.com/MaanAkshay/Power-BI-Project-Maven-Market-. Jeu de données Maven Market © Maven Analytics (Data Playground, usage pédagogique). Les mesures DAX/M présentées suivent la version canonique du projet Maven Market.