BUSINESS INTELLIGENCE · ÉDITION 2026

Microsoft Power BI

Le guide complet — de la donnée brute à la décision

Collecte, **Power Query**, modélisation en étoile, **DAX moderne**, visualisation, publication — avec les nouveautés **Fabric, Direct Lake, Copilot, PBIR & fonctions DAX**.

7 chapitres progressifs	**60+** exemples DAX, M & SQL prêts à l'emploi	**1** aide-mémoire DAX en annexe	**30 j** feuille de route d'apprentissage





EL KHLIFI Imad
Doctorant en Économie & Gestion · Data & BI
FSJES Tanger · Université Abdelmalek Essaâdi

Power BI · Data & IA · Économétrie

Édition **2026**

AVANT DE COMMENCER

Transformer des données brutes en décisions — méthodiquement.

Power BI est devenu l'outil de référence de la Business Intelligence en libre-service. Ce guide vous accompagne pas à pas, du premier import de fichier jusqu'au rapport publié, sécurisé et commenté par Copilot. Chaque chapitre combine les concepts, des exemples de code commentés et les bonnes pratiques issues du terrain.



Le flux de travail d'un projet Power BI, de la source au décideur

★ Ce qui change dans l'édition 2026

Microsoft Fabric & Direct Lake
Power BI s'inscrit désormais dans Fabric : les données vivent dans OneLake et le mode **Direct Lake** les lit sans les copier.

Copilot dans les rapports
Résumés automatiques, questions en langage naturel, aide à l'écriture de DAX : l'IA devient un assistant d'analyse.

PBIR, PBIP & TMDL
Les rapports deviennent des fichiers texte lisibles : versionnage Git, revue de code et travail en équipe facilités.

DAX nouvelle génération
Calculs visuels, fonctions DAX définies par l'utilisateur (UDF) et intelligence temporelle fondée sur des calendriers personnalisés.

Comment lire ce guide

ASTUCE — Gain de temps

ATTENTION — Piège fréquent

À RETENIR — Concept clé

2026 — Nouveauté

SOMMAIRE

Votre parcours, en 7 étapes

01 **Introduction & écosystème Power BI**
Composants, Fabric, licences, interface et premier projet

02 **Connexion aux sources de données**
Import, DirectQuery, Direct Lake, fichiers, bases, API, actualisation

03 **Transformer avec Power Query**
Nettoyage, langage M, fusions, regroupements, query folding

04 **Modélisation des données**
Schéma en étoile, relations, table de dates, optimisation

05 **Le langage DAX**
Contextes, CALCULATE, time intelligence, variables, UDF, calculs visuels

06 **Visualisation & data storytelling**
Choisir le bon visuel, mise en page, interactivité, accessibilité

07 **Publier, partager & gouverner**
Service, espaces de travail, RLS, Git, Copilot, gouvernance

A **Annexes**
Aide-mémoire DAX · Check-list performance · Feuille de route 30 jours

CHAPITRE 1

01



Introduction & écosystème Power BI

Comprendre ce qu'est Power BI aujourd'hui, comment ses briques s'articulent avec Microsoft Fabric, et réaliser un premier rapport de bout en bout.

DANS CE CHAPITRE

1.1 Qu'est-ce que Power BI ?

1.2 L'écosystème Fabric

1.3 Licences & installation

1.4 Découvrir l'interface

1.5 Concepts fondamentaux

1.6 Premier projet guidé

1.1 Qu'est-ce que Power BI ?

Microsoft Power BI est une plateforme de Business Intelligence qui permet de se connecter à des centaines de sources, de préparer les données, de les modéliser, puis de produire des rapports interactifs consultables sur le web, dans Teams et sur mobile. Sa force : couvrir toute la chaîne analytique dans un seul outil, accessible aux métiers comme aux équipes techniques.

Les composants de la plateforme

Composant	Rôle	Pour qui ?
Power BI Desktop	Application Windows gratuite de conception : connexion, Power Query, modèle, DAX, rapports	Créateurs de rapports
Service Power BI	Plateforme en ligne (SaaS) : publication, partage, actualisation planifiée, applications	Toute l'organisation
Power BI Mobile	Applications iOS / Android pour consulter rapports et alertes, avec Copilot	Décideurs nomades
Report Server	Serveur de rapports sur site pour les contextes sans cloud	DSI, secteurs réglementés
Power BI Embedded	Intégration des rapports dans vos propres applications et portails	Développeurs, éditeurs

Pourquoi Power BI ?

100+
connecteurs natifs

M + DAX
deux langages, deux rôles

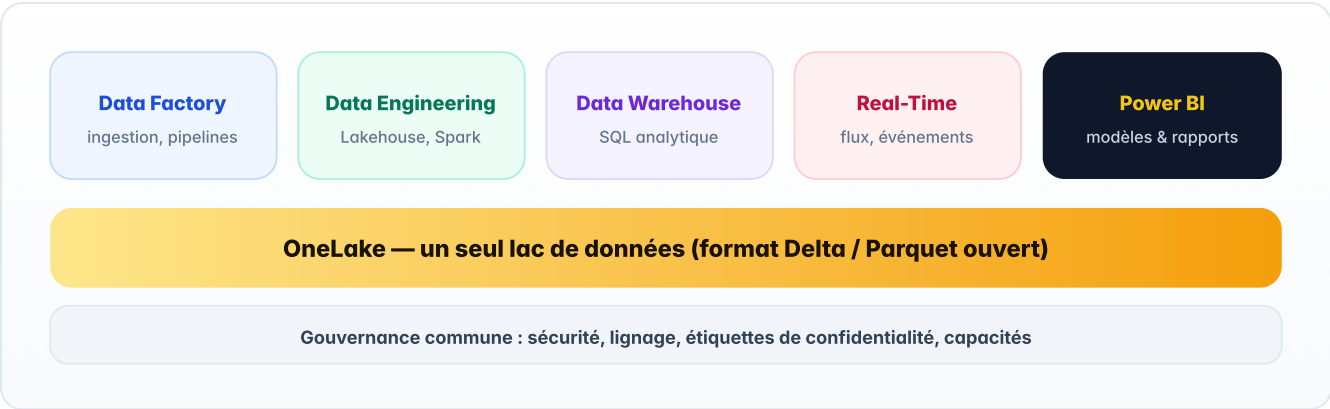
1/mois
mise à jour de Desktop

M365
intégration Teams, Excel, SharePoint

- **Facilité :** une interface accessible aux non-développeurs, et des langages puissants pour aller plus loin.
- **Préparation intégrée :** Power Query nettoie et restructure sans code — ou avec du code M.
- **Modélisation sérieuse :** moteur en mémoire VertiPaq, relations, hiérarchies, mesures DAX.
- **Partage et gouvernance :** espaces de travail, applications, sécurité au niveau des lignes, étiquettes de confidentialité.
- **IA intégrée :** Copilot, narrations automatiques, visuels d'analyse (influenceurs clés, arborescence de décomposition).

1.2 Power BI dans l'écosystème Microsoft Fabric

Microsoft Fabric est la plateforme analytique unifiée de Microsoft. Power BI en est la couche de restitution : il partage avec les autres charges de travail un même lac de données, **OneLake**, et un même modèle de capacité.



À RETENIR

Vous n'avez pas besoin de Fabric pour apprendre Power BI : Desktop reste gratuit et autonome. Fabric devient pertinent lorsque les volumes, le nombre de sources ou les besoins de gouvernance augmentent.

1.3 Licences & installation

Licence	Ce qu'elle permet	Usage typique
Gratuite (Fabric Free)	Créer dans Desktop, publier dans « Mon espace de travail » ; pas de partage	Apprentissage, prototypage
Power BI Pro	Publier, partager et collaborer dans des espaces de travail ; actualisation 8 fois/jour	Équipes et PME
Premium par utilisateur (PPU)	Fonctions avancées (actualisation 48 fois/jour, modèles plus volumineux, pipelines)	Experts BI, projets avancés
Capacité Fabric (F SKU)	Capacité partagée par l'organisation, toutes les charges Fabric, diffusion large de contenus	Grandes organisations

Tableau 1.1 — Panorama des licences (les conditions tarifaires évoluent : vérifiez la page officielle avant tout achat)

Configuration système de Power BI Desktop

Composant	Minimum	Recommandé
Système	Windows 10 ou Windows 11 (64 bits)	Windows 11 à jour
Processeur	x64, 1 GHz	Multicœur récent
Mémoire	2 Go	8 Go et plus pour des modèles confortables
Affichage	1440 × 900	1920 × 1080 ou plus
Composants	.NET Framework 4.7.2+ et Microsoft Edge WebView2	Installés automatiquement via le Microsoft Store

ASTUCE PRO

Installez Power BI Desktop depuis le **Microsoft Store** : les mises à jour mensuelles s'appliquent automatiquement, sans droits administrateur. Sur Mac, utilisez une machine virtuelle Windows ou les fonctions d'édition du service web.

1.4 Découvrir l'interface

Le ruban

- **Accueil** : obtenir des données, transformer, actualiser, publier
- **Insérer** : visuels, zones de texte, boutons, images
- **Modélisation** : mesures, colonnes, tables, paramètres, rôles
- **Affichage** : thèmes, mise en page, volets
- **Optimiser** : analyseur de performances, préréglages
- **Aide** : documentation et formation

Les volets latéraux

- **Données** : tables, colonnes, mesures du modèle
- **Visualisations** : choix et mise en forme des visuels
- **Filtres** : niveau visuel, page ou rapport
- **Signets, Sélection, Performances** : à afficher selon les besoins
- **Copilot** : assistant de création et d'analyse

Les cinq vues de travail

Vue	À quoi elle sert
Rapport	Concevoir les pages, placer et mettre en forme les visuels
Tableau (Données)	Explorer les lignes importées, vérifier les types et les valeurs
Modèle	Gérer relations, hiérarchies, propriétés et dossiers d'affichage
Requête DAX	Écrire et tester des requêtes DAX, avec l'aide de Copilot
TMDL	Éditer le modèle sémantique sous forme de script texte (Tabular Model Definition Language)

1.5 Concepts fondamentaux

Source de données

Origine de l'information : fichiers (Excel, CSV, JSON, PDF), bases (SQL Server, PostgreSQL, Oracle), services cloud (SharePoint, Dataverse, Salesforce), web et API.

Tables & relations

Les données sont organisées en tables (lignes = enregistrements, colonnes = champs), reliées entre elles par des clés communes.

Modèle sémantique

L'ensemble tables + relations + mesures. Il porte la logique métier et alimente un ou plusieurs rapports.

Colonne calculée

Calculée **ligne par ligne** lors de l'actualisation, puis stockée dans le modèle. Elle consomme de la mémoire et sert à catégoriser ou filtrer.

Mesure

Calculée **à la demande**, selon le contexte de filtre du visuel. Non stockée : c'est l'outil privilégié pour tous les indicateurs (KPI).

DAX (Data Analysis Expressions) est le langage des mesures, colonnes et tables calculées. Sa syntaxe rappelle celle d'Excel, mais il raisonne sur des tables et des contextes de filtre plutôt que sur des cellules.

Votre première mesure — DAX

```
Ventes Totales = SUM ( Ventes[Montant] )
```

1.6 Premier projet guidé : un rapport de ventes

Objectif : une page de synthèse affichant le chiffre d'affaires, son évolution mensuelle, la répartition par catégorie et le top 5 des produits.

ÉTAPE 1
Obtenir les données
Accueil → Obtenir des données → Classeur Excel. Sélectionnez les tables Ventes, Produits et Clients.

ÉTAPE 2
Transformer
Cliquez sur « Transformer les données » : vérifiez les types, supprimez les colonnes inutiles, renommez clairement.

ÉTAPE 3
Modéliser
Dans la vue Modèle, vérifiez les relations 1→* entre Produits, Clients et Ventes. Ajoutez une table de dates.

ÉTAPE 4
Créer les mesures
Ventes Totales, Nombre de commandes, Panier moyen — toujours sous forme de mesures explicites.

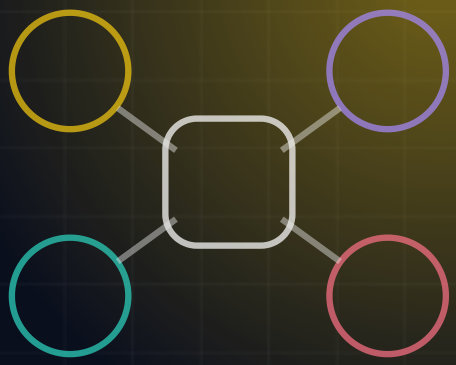
ÉTAPE 5
Visualiser
Cartes KPI en haut, courbe mensuelle, barres par catégorie, visuel Top 5 filtré par rang.

ÉTAPE 6
Enregistrer et publier
Enregistrez le projet, puis Publier → espace de travail. Configurez l'actualisation dans le service.

ASTUCE PRO
Pour vous entraîner, utilisez les jeux d'exemples fournis par Microsoft (Financial Sample, Contoso, AdventureWorks) : ils proposent un modèle réaliste sans risque pour vos données.

CHAPITRE 2

02



Connexion aux sources de données

Choisir le bon mode de stockage, se connecter à des fichiers, des bases de données, des services cloud et des API, puis automatiser l'actualisation.

DANS CE CHAPITRE

2.1 Modes de stockage

2.2 Fichiers

2.3 Bases de données

2.4 Services en ligne

2.5 Web & API REST

2.6 Actualisation des données

2.1 Choisir le bon mode de stockage

Le mode de stockage détermine **où vivent les données** et donc la performance, la fraîcheur et les fonctionnalités disponibles. C'est la première décision d'architecture d'un projet.

Import DÉFAUT

Les données sont compressées et chargées dans le moteur en mémoire VertiPaq.
+ Performances maximales, toutes les fonctions DAX, travail hors ligne
– Données figées entre deux actualisations ; taille limitée (1 Go par modèle en Pro)

DirectQuery

Les données restent à la source ; chaque interaction envoie une requête.
+ Données à jour, pas de duplication, très gros volumes
– Performances dépendantes de la source, certaines limites DAX et Power Query, charge sur le serveur

Direct Lake FABRIC

Lit directement les tables Delta de OneLake, sans copie ni requêtes SQL.
+ Vitesse proche de l'Import et fraîcheur proche du DirectQuery
– Nécessite des données dans Fabric (Lakehouse ou Warehouse) et une capacité Fabric

Composite

Combine plusieurs modes dans un même modèle : par exemple, dimensions en Import et grande table de faits en DirectQuery ou Direct Lake.
+ Flexibilité maximale
– Conception plus exigeante

À RETENIR

Règle de décision : commencez par l'Import. Passez au DirectQuery si la fraîcheur temps réel ou le volume l'imposent, et au Direct Lake si vos données sont déjà dans Microsoft Fabric.

2.2 Se connecter à des fichiers

Excel

Power BI lit les formats `.xlsx`, `.xlsm` et `.xlsb`. Chemin : **Accueil → Obtenir des données → Classeur Excel**, sélection des feuilles ou tables dans le Navigateur, puis **Transformer les données** plutôt que « Charger » pour garder la main.

ASTUCE PRO

Mettez vos plages au format **Tableau** dans Excel (Ctrl + T) et nommez-les. Power BI suivra la table même si des lignes s'ajoutent, ce qui fiabilise l'actualisation.

- Une seule ligne d'en-têtes, sans cellules fusionnées ni lignes de totaux.
- Un type cohérent par colonne (dates, nombres, texte).
- Stockez le fichier sur **OneDrive ou SharePoint** pour permettre l'actualisation dans le cloud sans passerelle.

CSV et fichiers texte

Points de vigilance : **délimiteur** (virgule, point-virgule, tabulation), **encodage** (UTF-8 recommandé) et **paramètres régionaux**.

ATTENTION
Les CSV produits en français ou au Maroc utilisent souvent le point-virgule comme séparateur et la virgule comme décimale. Utilisez « Modifier le type → Utilisation des paramètres régionaux » pour éviter que 1,5 devienne 15.

JSON et dossiers

Les fichiers **JSON** (souvent issus d'API) se transforment en tables en développant les enregistrements et en convertissant les listes. Le connecteur **Dossier** combine automatiquement tous les fichiers de même structure : idéal pour des exports mensuels.

Structure de dossier combinable — TEXTE

```
C:\Donnees\Ventes\
   Ventes_2026_01.xlsx
   Ventes_2026_02.xlsx
   Ventes_2026_03.xlsx      <- tout nouveau fichier est intégré à l'actualisation
```

2.3 Se connecter aux bases de données

SQL Server

1. Obtenir des données → Base de données SQL Server.
2. Saisir le serveur (`serveur.domaine.com`) et, si possible, la base de données.
3. Choisir le mode : Import ou DirectQuery.
4. S'authentifier : Windows, base de données ou compte Microsoft (Entra ID).
5. Sélectionner les tables ou vues nécessaires, puis Transformer.

À RETENIR
Sur de gros volumes, préférez des **vues SQL** préparées côté serveur qui ne renvoient que les colonnes et les lignes utiles. Vous gagnez en performance et en maintenabilité.

Requête SQL personnalisée — SQL

```sql
-- Requête source : seulement les colonnes utiles, filtrées à la source
SELECT  v.DateVente,
        v.MontantVente,
        c.NomClient,
        c.Region,
        p.NomProduit,
        p.Categorie
FROM    Ventes v
INNER JOIN Clients  c ON v.ClientID  = c.ClientID
INNER JOIN Produits p ON v.ProduitID = p.ProduitID
WHERE   v.DateVente >= '2024-01-01'
  AND   v.Statut = 'Valide'
```

ATTENTION
Une requête SQL saisie dans la boîte « Instruction SQL » peut empêcher le **query folding** des étapes suivantes (voir chapitre 3). Une vue côté serveur est généralement préférable.

Autres moteurs pris en charge

PostgreSQL open source avancé	**MySQL** web & applications	**Oracle** systèmes d'entreprise	**SAP HANA** ERP en mémoire
Snowflake entrepôt cloud	**BigQuery** Google Cloud	**Redshift** Amazon Web Services	**Databricks** lakehouse

2.4 Se connecter aux services en ligne

SharePoint & OneDrive
Dossier SharePoint pour combiner les fichiers d'une bibliothèque ; **Liste SharePoint** pour importer une liste comme table. Utilisez l'URL racine du site.

Microsoft Dataverse
Socle de la Power Platform et de Dynamics 365. Connexion via l'URL de l'environnement, prise en charge de l'Import et du DirectQuery.

Dynamics 365
Privilégiez le connecteur Dataverse pour les nouveaux rapports ; les flux OData restent une méthode historique.

Salesforce
Objets ou rapports Salesforce. Chaque actualisation consomme des appels API : vérifiez vos quotas.

À RETENIR
Google Analytics 4 se connecte également via son connecteur dédié (authentification OAuth, choix de la propriété, des dimensions et des métriques).

2.5 Connexion Web & API REST

Le connecteur **Web** détecte automatiquement les tableaux HTML d'une page. Pour une API REST, on écrit quelques lignes de M :

Appeler une API REST dans Power Query — M

```
let
    url     = "https://api.exemple.com/v1/ventes",
    entetes = [ #"Authorization" = "Bearer " & TokenAPI,
                #"Accept"        = "application/json" ],
    reponse = Json.Document( Web.Contents( url, [ Headers = entetes ] ) ),
    lignes  = reponse[data],
    table   = Table.FromRecords( lignes )
in
    table
```

ATTENTION
Pour que l'actualisation fonctionne dans le service, gardez une **URL de base fixe** dans `Web.Contents` et passez les parties variables via les options `RelativePath` et `Query`. Ne stockez jamais un jeton en clair dans le code : utilisez un paramètre ou les informations d'identification du connecteur.

OData (Open Data Protocol) est un standard d'API REST : Obtenir des données → Flux OData, puis sélection des entités.

2.6 Actualisation des données

Type	Où	Principe
Manuelle	Desktop	Bouton Actualiser : utile en développement
Planifiée	Service	Jusqu'à 8 fois par jour en Pro, 48 fois en PPU ou capacité
Incrémentielle	Service	Ne recharge que les périodes récentes d'une grande table
Temps réel	Service / Fabric	DirectQuery, Direct Lake ou flux en temps réel

ATTENTION
Les sources **sur site** (serveur interne, dossier réseau) exigent une **passerelle de données locale** pour être actualisées depuis le cloud.

Configurer l'actualisation incrémentielle

1. Créer deux paramètres de type Date/Heure nommés exactement `RangeStart` et `RangeEnd`.
2. Filtrer la table de faits : date ≥ RangeStart et date < RangeEnd.
3. Clic droit sur la table → Actualisation incrémentielle : définir la période archivée (ex. 5 ans) et la période actualisée (ex. 10 jours).
4. Publier : le service crée et gère automatiquement les partitions.

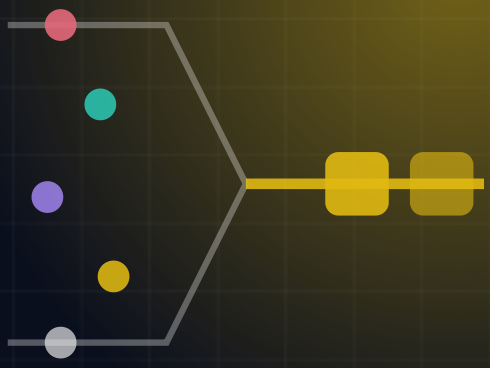
Filtre requis pour l'actualisation incrémentielle — M

```
Ventes_Filtrees = Table.SelectRows(
    Source,
    each [DateVente] >= RangeStart and [DateVente] < RangeEnd
)
```

ASTUCE PRO
Ce filtre doit se « replier » (query folding) vers la source : sinon chaque partition relirait toute la table et le gain disparaîtrait.

CHAPITRE 3

03



Transformer les données avec Power Query

Nettoyer, restructurer et enrichir les données avant de les charger : des transformations en quelques clics, et le langage M pour aller plus loin.

DANS CE CHAPITRE

3.1 Découvrir Power Query

3.2 Transformations de base

3.3 Transformations avancées

3.4 Erreurs & valeurs nulles

3.5 Fusionner & ajouter

3.6 Grouper & agréger

3.7 Paramètres & fonctions

3.8 Query folding & bonnes pratiques

3.1 Découvrir Power Query

Power Query est le moteur de préparation des données de Power BI (on le retrouve aussi dans Excel et dans les Dataflows Gen2 de Fabric). Chaque action génère une **étape** enregistrée : la transformation est rejouée à l'identique à chaque actualisation.

Ouvrir l'éditeur

- Accueil → **Transformer les données**
- Lors d'un import : bouton « Transformer » au lieu de « Charger »
- Clic droit sur une table → Modifier la requête

Anatomie de l'éditeur

1. **Requêtes** (gauche) : la liste des tables
2. **Aperçu** (centre) : les données transformées
3. **Étapes appliquées** (droite) : l'historique
4. **Barre de formule** : le code M de l'étape
5. **Ruban** : les commandes

ASTUCE PRO

Activez **Affichage → Qualité, Distribution et Profil de colonne**, et basculez le profilage sur « l'ensemble du jeu de données » : vous repérez immédiatement les erreurs, les vides et les valeurs aberrantes.

Le langage M en un coup d'œil

Une requête M est une suite d'étapes nommées dans un bloc `let … in` ; chaque étape réutilise la précédente.

Structure d'une requête M — M

```
let
    Source        = Excel.Workbook( File.Contents( CheminFichier ), null, true ),
    Feuille       = Source{[Item = "Ventes", Kind = "Table"]}[Data],
    TypesModifies = Table.TransformColumnTypes( Feuille, {
                        { "Date",    type date },
                        { "Montant", type number } } )
in
    TypesModifies
```

3.2 Transformations de base

Supprimer des colonnes

Sélectionnez les colonnes à garder → **Supprimer les autres colonnes**. Si la source ajoute une colonne plus tard, elle sera ignorée automatiquement.

Renommer

Double-clic sur l'en-tête. Adoptez des noms métier lisibles : « Montant HT » plutôt que « MNT_HT_01 ».

Filtrer

Filtres de texte (contient, commence par), numériques (entre, supérieur à), de date (année dernière, mois en cours), combinés par ET / OU.

Trier

Le tri dans Power Query n'impose pas l'ordre d'affichage dans les visuels : utilisez plutôt « Trier par colonne » dans le modèle.

Les types de données

Type	Exemple	Remarque
Texte	« Casablanca »	Codes postaux, identifiants, numéros de compte
Nombre entier	1 250	Clés techniques, quantités
Nombre décimal fixe	1 250,75	Montants monétaires (4 décimales, sans erreur d'arrondi)
Nombre décimal	0,0375	Ratios, mesures scientifiques
Date / Date-heure	16/11/2026	N'utilisez Date-heure que si l'heure est nécessaire
Vrai/Faux	TRUE	Indicateurs binaires

ATTENTION
La détection automatique des types se trompe régulièrement : un code postal « 01000 » devient 1000. Vérifiez **toujours** les types, et placez l'étape de typage explicitement en fin de nettoyage.

3.3 Transformations avancées

Fractionner et fusionner des colonnes

Méthode	Avant	Après
Par délimiteur	« Rabat, Maroc »	« Rabat » | « Maroc »
Par nombre de caractères	« 20260315 »	« 2026 » | « 03 » | « 15 »
Minuscule → majuscule	« NomPrénom »	« Nom » | « Prénom »
Fusion (séparateur espace)	« Imad » + « El Khlifi »	« Imad El Khlifi »

Pivoter et dépivoter

Le **dépivotage** transforme un tableau « large » (une colonne par mois) en tableau « long » (une ligne par observation), le format idéal pour l'analyse.

Avant : format large

Produit	Jan	Fév	Mar
A	100	150	120
B	200	210	190

Après : format long

Produit	Mois	Ventes
A	Jan	100
A	Fév	150
A	Mar	120
B	Jan	200
…	…	…

ASTUCE PRO
Sélectionnez la colonne **Produit** puis **Dépivoter les autres colonnes** : les mois ajoutés à l'avenir seront traités automatiquement.

Colonnes personnalisées et conditionnelles

Exemples de colonnes personnalisées — M

```
// Chiffre d'affaires de la ligne
= [Quantite] * [PrixUnitaire]

// Préfixe conditionnel
= if [Pays] = "Maroc" then "MA-" & [CodePostal] else [CodePostal]

// Extraction de l'année
= Date.Year( [DateVente] )
```

À RETENIR
La **Colonne à partir d'exemples** permet de taper le résultat attendu sur quelques lignes : Power Query devine la formule M correspondante.

3.4 Gérer les erreurs et les valeurs nulles

Une cellule `Error` provient le plus souvent d'une conversion de type impossible, d'une division par zéro ou d'un format de date invalide.

1. **Supprimer les erreurs** : retire les lignes concernées (à utiliser avec prudence).
2. **Remplacer les erreurs** : par une valeur par défaut.
3. **try ... otherwise** : gestion programmée, au cas par cas.

try ... otherwise et remplacement des nulls — M

```
// Calcul protégé
= try [Montant] / [Quantite] otherwise 0

// Remplacer les nulls par 0 dans la colonne Montant
= Table.ReplaceValue( Source, null, 0, Replacer.ReplaceValue, {"Montant"} )
```

Pour les valeurs **nulles** : supprimer les lignes vides, remplacer par une valeur, ou **Remplissage vers le bas** pour propager une valeur (typique des exports Excel avec cellules fusionnées).

3.5 Fusionner et ajouter des requêtes

Ajouter (UNION)
Empile verticalement des tables de même structure : fichiers mensuels, régions, exports successifs. Les noms et types de colonnes doivent correspondre.

Fusionner (JOIN)
Combine horizontalement deux tables sur une ou plusieurs colonnes communes, comme une jointure SQL.



Figure 3.1 — Les six types de fusion (en orange : les lignes conservées)

À RETENIR
Externe gauche pour enrichir une table principale sans perdre de lignes ; **interne** pour ne garder que les correspondances ; **anti gauche** pour détecter les orphelins (ex. ventes sans client connu) — un excellent contrôle qualité.

3.6 Grouper et agréger

Accueil → Regrouper par résume les données : somme, moyenne, min / max, médiane, nombre de lignes, nombre de valeurs distinctes.

Données brutes

Région	Produit	Qté	Montant
Nord	A	10	100
Nord	B	15	150
Sud	A	20	200
Sud	B	25	250

Après regroupement par Région

Région	Qté	Montant	Nb produits
Nord	25	250	2
Sud	45	450	2

ATTENTION
Ne pré-agrégez pas par réflexe : en conservant le détail, les mesures DAX restent capables de répondre à n'importe quelle question. Regroupez lorsque le grain fin est réellement inutile ou trop volumineux.

3.7 Paramètres et fonctions personnalisées

Les **paramètres** (Accueil → Gérer les paramètres) rendent la solution configurable : chemin de fichier, serveur, date de début, seuil de filtrage.

Utiliser un paramètre — M

```
let
    Source = Excel.Workbook( File.Contents( CheminFichier ), null, true )  // paramètre
in
    Source
```

Une **fonction personnalisée** encapsule une logique réutilisable ; on l'invoque ensuite via « Appeler une fonction personnalisée ».

Fonction de nettoyage de texte — M

```
( texte as text ) as text =>
let
    SansEspaces = Text.Trim( texte ),
    Propre      = Text.Proper( SansEspaces )
in
    Propre
```

3.8 Query folding & bonnes pratiques

Le **query folding** (repli de requête) est la capacité de Power Query à traduire vos étapes en une seule requête exécutée par la source (SQL, par exemple). C'est le facteur de performance n°1 sur les bases de données.

✓ Se replient généralement

Filtrer des lignes, supprimer ou renommer des colonnes, modifier les types, fusions simples, regroupements.

✕ Cassent souvent le repli

Colonnes d'index, certaines colonnes personnalisées complexes, fonctions M sans équivalent SQL, sources non relationnelles (Excel, CSV).

ASTUCE PRO

Clic droit sur une étape → **Afficher la requête native**. Si l'option est grisée, le repli est rompu à partir de cette étape. Placez les étapes « repliables » en premier.

Organisation

- Noms explicites : « Ventes_Nettoyees »
- Groupes de requêtes : Sources, Staging, Dimensions, Faits
- Étapes renommées, commentaires dans le code M

Performance

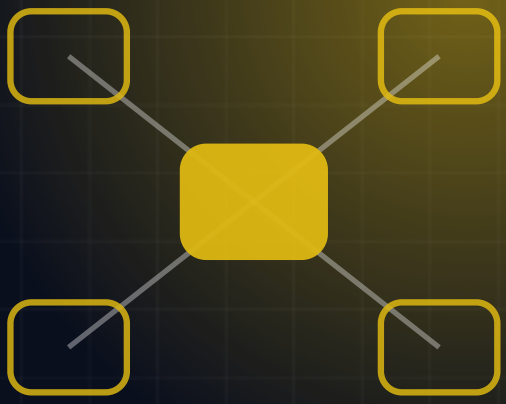
- Filtrer et réduire les colonnes tôt
- Désactiver le chargement des requêtes intermédiaires
- Éviter les tris inutiles
- Préserver le query folding

Maintenance

- Paramétrer chemins et serveurs
- Documenter les règles métier
- Tester avec des données réelles
- Versionner le projet (PBIP)

CHAPITRE 4

04



Modélisation des données

Le modèle est la fondation de tout rapport : un schéma en étoile bien conçu rend le DAX simple, les rapports rapides et les chiffres justes.

DANS CE CHAPITRE

4.1 Schéma en étoile

4.2 Faits & dimensions

4.3 Relations & filtrage

4.4 Hiérarchies

4.5 Table de dates

4.6 Colonnes vs mesures

4.7 Optimiser le modèle

4.1 Schéma en étoile ou en flocon ?



Figure 4.1 — Un schéma en étoile : une table de faits centrale, des dimensions autour, des relations 1 → *

Étoile RECOMMANDÉ

- Une table de faits centrale (transactions, événements)
- Des dimensions dénormalisées autour
- Relations directes faits ↔ dimensions
- DAX plus simple, performances supérieures

Flocon

- Dimensions normalisées en cascade (Produit → Sous-catégorie → Catégorie)
- Relations indirectes
- Plus de jointures, filtres plus lents
- À aplatir dans Power Query quand c'est possible

4.2 Tables de faits et tables de dimensions

Faits

- Les événements mesurables : ventes, commandes, paiements
- Valeurs numériques : quantités, montants, durées
- Clés étrangères vers les dimensions
- Volumétrie forte (millions de lignes), grain le plus fin

Dimensions

- Le contexte descriptif : qui, quoi, où, quand
- Attributs de filtre et de regroupement
- Clé primaire unique
- Volumétrie faible, hiérarchies possibles

Table de faits

VenteID	DateID	ClientID	ProduitID	Montant
1	20260101	1001	5001	150,00
2	20260101	1002	5002	200,00
3	20260102	1001	5001	150,00

Table de dimension

table de dimension

ProduitID	Produit	Catégorie	Marque
5001	Laptop Pro	Informatique	TechBrand
5002	Souris sans fil	Accessoires	TechBrand
5003	Clavier mécanique	Accessoires	GameBrand

4.3 Relations et direction du filtrage

Cardinalité	Exemple	Conseil
Un-à-plusieurs (1 : *)	Un client → plusieurs ventes	Le cas standard, à privilégier
Un-à-un (1 : 1)	Produit ↔ Fiche technique	Signale souvent deux tables à fusionner
Plusieurs-à-plusieurs (* : *)	Budget par catégorie ↔ ventes par produit	Utile pour des grains différents ; tester soigneusement



Le filtre descend de la dimension vers les faits.

Le filtre circule dans les deux sens : risque d'ambiguïté.

ATTENTION

Le filtrage **bidirectionnel** peut créer des chemins ambigus et dégrader les performances. Préférez un filtrage unique, et activez le sens inverse ponctuellement dans une mesure avec `CROSSFILTER`.

Relations actives et inactives

Une seule relation peut être **active** entre deux tables. Si la table Ventes contient une date de commande et une date de livraison, la seconde relation vers le calendrier sera inactive (en pointillés). On l'active à la demande :

Activer une relation inactive — DAX

```
Ventes par Date Livraison =
CALCULATE (
    [Ventes Totales],
    USERELATIONSHIP ( Ventes[DateLivraison], Calendrier[Date] )
)
```

4.4 Hiérarchies

Une hiérarchie permet de naviguer entre niveaux de détail (**exploration descendante** / **remontante**) dans les visuels. Clic droit sur un champ → Créer une hiérarchie, puis ajoutez les niveaux du plus général au plus fin.

Géographique
Pays → Région → Province → Ville

Temporelle
Année → Trimestre → Mois → Jour

4.5 La table de dates

Une table de dates dédiée, continue et marquée comme telle, est indispensable à l'intelligence temporelle (cumuls, comparaisons N-1, moyennes mobiles).

Une table de dates dynamique, bornée par les données — DAX

```dax
Calendrier =
VAR DateDebut = DATE ( YEAR ( MIN ( Ventes[DateVente] ) ), 1, 1 )
VAR DateFin   = DATE ( YEAR ( MAX ( Ventes[DateVente] ) ), 12, 31 )
RETURN
    ADDCOLUMNS (
        CALENDAR ( DateDebut, DateFin ),
        "Année",        YEAR ( [Date] ),
        "N° Mois",      MONTH ( [Date] ),
        "Mois",         FORMAT ( [Date], "MMMM" ),
        "Mois court",   FORMAT ( [Date], "MMM" ),
        "Trimestre",    "T" & QUARTER ( [Date] ),
        "Année-Mois",   FORMAT ( [Date], "YYYY-MM" ),
        "Semaine ISO",  WEEKNUM ( [Date], 21 ),
        "Jour",         FORMAT ( [Date], "DDDD" ),
        "Week-end",     WEEKDAY ( [Date], 2 ) > 5
    )
```

1. Sélectionnez la table → **Outils de table → Marquer comme table de dates**, et choisissez la colonne Date.
2. Reliez Calendrier[Date] aux colonnes de date des faits (relation 1 : *).
3. Triez « Mois » par « N° Mois » (Outils de colonne → Trier par colonne).
4. Désactivez « Date/heure automatique » dans les options : elle crée des tables cachées inutiles.

NOUVEAUTÉ 2026
L'**intelligence temporelle fondée sur des calendriers** permet de définir plusieurs calendriers (civil, fiscal, commercial 4-4-5, scolaire) et débloque des calculs par semaine comme `TOTALWTD`. Elle s'active dans les fonctionnalités en préversion de Desktop puis se configure dans les options de calendrier ou la vue TMDL.

4.6 Colonnes calculées ou mesures ?

Critère	Colonne calculée	Mesure
Moment du calcul	À l'actualisation	À l'affichage, selon les filtres
Stockage	Stockée, consomme de la RAM	Non stockée
Contexte	Ligne par ligne	Contexte de filtre
Utilisation	Axes, segments, filtres, catégories	Valeurs, KPI, ratios, cumuls
Exemple	Tranche d'âge, catégorie client	CA, marge %, croissance N-1

Tableau 4.1 — Choisir le bon type de calcul

Colonne calculée de segmentation — DAX

```dax
Tranche Client =
SWITCH (
    TRUE (),
    Clients[CA Total] > 10000, "Premium",
    Clients[CA Total] > 5000,  "Standard",
    "Basique"
)
```

ASTUCE PRO

Règle d'or : si le calcul peut être fait en amont (source SQL ou Power Query), faites-le en amont ; sinon, préférez une mesure ; utilisez une colonne calculée seulement pour un attribut fixe servant à filtrer ou à regrouper.

4.7 Optimiser le modèle

Réduire la taille

- Supprimer les colonnes inutilisées (surtout les identifiants de transaction)
- Réduire la cardinalité : séparer date et heure, arrondir les décimales
- Types adaptés : entier plutôt que texte
- Désactiver le chargement des requêtes intermédiaires

Optimiser les relations

- Relations 1 : * sur des clés entières
- Filtrage unique par défaut
- Limiter les relations plusieurs-à-plusieurs
- Éviter les chaînes de dimensions (flocon)

Bonnes pratiques de modélisation

Structure
Schéma en étoile, table de dates dédiée, une table de mesures dédiée.

Lisibilité
Noms métier, dossiers d'affichage, colonnes techniques masquées, descriptions renseignées.

Préparation IA
Descriptions, synonymes et schéma simplifié rendent le modèle « prêt pour Copilot ».

CHAPITRE 5

05

Le langage DAX

```
CALCULATE (
    [Ventes],
    DATEADD (
        -1, YEAR
    ) )
```

Écrire des mesures justes et performantes : comprendre les contextes, maîtriser CALCULATE, l'intelligence temporelle, les variables — et les nouveautés 2026.

DANS CE CHAPITRE

5.1 Syntaxe & contextes

5.2 Agrégations & itérateurs

5.3 CALCULATE & filtres

5.4 Intelligence temporelle

5.5 Logique & conditions

5.6 Variables

5.7 Texte, tables & erreurs

5.8 UDF & calculs visuels

5.9 Bonnes pratiques

5.1 Syntaxe et contextes d'évaluation

Structure d'une mesure — DAX

```
Nom de la mesure =
    EXPRESSION_DAX      // une fonction, une référence, une constante…
```

Éléments de syntaxe

- Colonne : `Table[Colonne]`
- Mesure : `[Nom Mesure]` (sans nom de table)
- Fonction : `FONCTION ( arguments )`
- Opérateurs : + – * / & = <> < > && || IN
- Commentaires : `//` ou `/* … */`

Types de données

- Entier (64 bits)
- Décimal (virgule flottante)
- Décimal fixe / Devise (4 décimales)
- Date / Date-heure
- Texte, Booléen (TRUE / FALSE)
- BLANK : l'absence de valeur

Tout le DAX repose sur une idée : **une même formule renvoie des résultats différents selon le contexte dans lequel elle est évaluée.**

Contexte de ligne

« la ligne courante »

Qté 3 × Prix 50

Qté 2 × Prix 80 ← ligne évaluée

Qté 5 × Prix 20

Contexte de filtre

« ce que voit la cellule du visuel »

Année = 2026 | Région = Nord | Catégorie = Audio

Transition de contexte

CALCULATE transforme le contexte de ligne en filtres.

C'est ce qui permet d'appeler une mesure dans un itérateur (SUMX, AVERAGEX…).

5.2 Fonctions d'agrégation et itérateurs

Les agrégations de base — DAX

```
Ventes Totales   = SUM ( Ventes[Montant] )
Montant Moyen    = AVERAGE ( Ventes[Montant] )
Vente Min        = MIN ( Ventes[Montant] )
Vente Max        = MAX ( Ventes[Montant] )
Nb Lignes        = COUNTROWS ( Ventes )
Nb Clients       = DISTINCTCOUNT ( Ventes[ClientID] )
```

Les fonctions **X** (SUMX, AVERAGEX, MAXX, COUNTX…) sont des **itérateurs** : elles parcourent une table ligne par ligne, évaluent une expression, puis agrègent le résultat.

Itérateurs DAX

```dax
CA Total =
SUMX ( Ventes, Ventes[Quantite] * Ventes[PrixUnitaire] )

Prix Moyen Pondéré =
DIVIDE ( [CA Total], SUM ( Ventes[Quantite] ) )

Panier Moyen =
AVERAGEX ( VALUES ( Ventes[CommandeID] ), [CA Total] )   // une ligne par commande
```

À RETENIR

Dans **Panier Moyen**, la mesure [CA Total] est évaluée pour chaque commande grâce à la **transition de contexte** : appeler une mesure revient à l'envelopper dans un CALCULATE implicite.

5.3 CALCULATE, la fonction la plus puissante

`CALCULATE` évalue une expression dans un **contexte de filtre modifié** : elle ajoute, remplace ou retire des filtres.

Filtres booléens dans CALCULATE DAX

```dax
Ventes Maroc =
CALCULATE ( [Ventes Totales], Clients[Pays] = "Maroc" )

Ventes Maroc 2026 =
CALCULATE (
    [Ventes Totales],
    Clients[Pays] = "Maroc",
    Calendrier[Année] = 2026
)

Ventes Cibles =
CALCULATE (
    [Ventes Totales],
    Produits[Catégorie] = "Électronique",
    Clients[Région] IN { "Nord", "Sud" }
)
```

ATTENTION

Plusieurs arguments de filtre se combinent en **ET**. Un filtre booléen ne peut porter que sur **une seule table** : pour un « OU » entre deux tables, utilisez une table de combinaisons ou des mesures séparées.

FILTER : pour les conditions sur des mesures

FILTER et TOPN — DAX

```
Ventes Produits Chers =
CALCULATE ( [Ventes Totales], Produits[Prix] > 100 )      // filtre de colonne : à préférer

Ventes Clients Fidèles =
CALCULATE (
    [Ventes Totales],
    FILTER ( VALUES ( Clients[ClientID] ), [Nb Commandes] >= 5 )   // condition sur une mesure
)

Ventes Top 10 Produits =
CALCULATE (
    [Ventes Totales],
    TOPN ( 10, ALL ( Produits[Nom] ), [Ventes Totales] )
)
```

ALL, ALLSELECTED, ALLEXCEPT, REMOVEFILTERS

Ignorer ou conserver des filtres — DAX

```
% du Total =
DIVIDE ( [Ventes Totales], CALCULATE ( [Ventes Totales], REMOVEFILTERS ( Produits ) ) )

% de la Sélection =
DIVIDE ( [Ventes Totales], CALCULATE ( [Ventes Totales], ALLSELECTED ( Produits ) ) )

Ventes Catégorie (toutes sous-cat.) =
CALCULATE ( [Ventes Totales], ALLEXCEPT ( Produits, Produits[Catégorie] ) )
```

Fonction	Effet
ALL / REMOVEFILTERS	Supprime tous les filtres d'une table ou d'une colonne
ALLSELECTED	Supprime les filtres internes au visuel mais conserve les segments et filtres externes
ALLEXCEPT	Supprime tous les filtres de la table sauf ceux des colonnes indiquées
KEEPFILTERS	Ajoute un filtre sans écraser le filtre existant (intersection)

5.4 Intelligence temporelle

Cumuls à date (YTD, QTD, MTD) — DAX

```
CA Cumul Annuel   = TOTALYTD ( [Ventes Totales], Calendrier[Date] )
CA Cumul Trim.    = TOTALQTD ( [Ventes Totales], Calendrier[Date] )
CA Cumul Mensuel  = TOTALMTD ( [Ventes Totales], Calendrier[Date] )

CA Exercice (fin juin) =
TOTALYTD ( [Ventes Totales], Calendrier[Date], "06-30" )
```

Comparaisons de périodes — DAX

```
CA N-1 =
CALCULATE ( [Ventes Totales], SAMEPERIODLASTYEAR ( Calendrier[Date] ) )

Croissance N/N-1 % =
VAR Actuel   = [Ventes Totales]
VAR Precedent = [CA N-1]
RETURN
    DIVIDE ( Actuel - Precedent, Precedent )

CA Mois Précédent =
CALCULATE ( [Ventes Totales], DATEADD ( Calendrier[Date], -1, MONTH ) )
```

Moyenne mobile sur 3 mois (moyenne de totaux mensuels) — DAX

```
CA Moyenne Mobile 3 Mois =
CALCULATE (
    AVERAGEX ( VALUES ( Calendrier[Année-Mois] ), [Ventes Totales] ),
    DATESINPERIOD ( Calendrier[Date], MAX ( Calendrier[Date] ), -3, MONTH )
)
```

À RETENIR

Toutes ces fonctions exigent une **table de dates continue et marquée** (chapitre 4). Sans elle, les résultats sont faux ou vides.

5.5 Fonctions logiques et conditionnelles

IF et SWITCH (TRUE ()) — DAX

```
Statut Objectif =
IF ( [Ventes Totales] >= [Objectif], "Atteint", "À améliorer" )

Niveau Performance =
SWITCH (
    TRUE (),
    [Ventes Totales] > 100000, "Or",
    [Ventes Totales] > 50000,  "Argent",
    [Ventes Totales] > 10000,  "Bronze",
    "Standard"
)

Nom Trimestre =
SWITCH ( QUARTER ( MAX ( Calendrier[Date] ) ), 1, "T1", 2, "T2", 3, "T3", "T4" )
```

ASTUCE PRO

`SWITCH ( TRUE (), … )` remplace avantageusement les IF imbriqués : la lecture est linéaire et les conditions sont évaluées dans l'ordre.

5.6 Variables : lisibilité et performance

`VAR` stocke un résultat intermédiaire, calculé **une seule fois** et immuable ; `RETURN` renvoie le résultat final.

Variables — DAX

```
Marge % =
VAR CA     = [Ventes Totales]
VAR Couts  = SUM ( Ventes[Coût] )
VAR Marge  = CA - Couts
RETURN
    DIVIDE ( Marge, CA )

Score Client (RFM simplifié) =
VAR Recence   = DATEDIFF ( MAX ( Ventes[DateVente] ), TODAY (), DAY )
VAR Frequence = [Nb Commandes]
VAR Montant   = [Ventes Totales]
RETURN
    IF ( Recence < 30, 10, IF ( Recence < 90, 5, 1 ) )
  + IF ( Frequence > 10, 10, IF ( Frequence > 5, 5, 1 ) )
  + IF ( Montant > 10000, 10, IF ( Montant > 5000, 5, 1 ) )
```

ASTUCE PRO

Pour déboguer, faites temporairement `RETURN` d'une variable intermédiaire : vous inspectez chaque étape sans réécrire la mesure.

5.7 Texte, tables et gestion des erreurs

Fonctions de texte — DAX

```
Nom Complet =
Clients[Prénom] & " " & Clients[Nom]

Code Produit =
UPPER ( Produits[Code] ) & "-" & FORMAT (
Produits[ID], "0000" )

Initiales =
LEFT ( Clients[Prénom], 1 ) & LEFT (
Clients[Nom], 1 )

Email Propre =
SUBSTITUTE ( LOWER ( Clients[Email] ), " ",
"" )
```

Fonctions de table — DAX

```
Ventes par Catégorie =
SUMMARIZECOLUMNS (
    Produits[Catégorie],
    "Total", [Ventes Totales]
)

Top 5 Clients =
TOPN ( 5, VALUES ( Clients[Nom] ), [Ventes
Totales] )

Liste Pays = VALUES ( Clients[Pays] )
```

Gérer les divisions par zéro et les valeurs vides — DAX

```
Marge Moyenne = DIVIDE ( SUM ( Ventes[Marge] ), [Ventes Totales] )   // renvoie BLANK si /0

Statut Livraison =
IF ( ISBLANK ( MAX ( Ventes[DateLivraison] ) ), "En attente", "Livré" )
```

ATTENTION
Évitez `IFERROR` et `ISERROR` dans les mesures : ils désactivent des optimisations du moteur. Utilisez `DIVIDE` et des tests explicites.

5.8 Les nouveautés DAX : fonctions et calculs visuels

Fonctions DAX définies par l'utilisateur (UDF) NOUVEAU

Les UDF permettent d'écrire une logique **une fois**, avec des paramètres typés, puis de la réutiliser dans toutes les mesures du modèle. On les définit dans la vue Requête DAX ou la vue TMDL.

Une fonction réutilisable — DAX

```
DEFINE
    /// Variation relative entre une valeur et sa référence
    FUNCTION VariationPct =
        ( actuel : NUMERIC, reference : NUMERIC ) =>
            DIVIDE ( actuel - reference, reference )

// Utilisation dans des mesures
Croissance CA %     = VariationPct ( [Ventes Totales], [CA N-1] )
Croissance Marge %  = VariationPct ( [Marge], [Marge N-1] )
```

Calculs visuels NOUVEAU

Les **calculs visuels** s'écrivent directement sur un visuel et opèrent sur sa grille de résultats : cumuls, écarts, moyennes mobiles, sans alourdir le modèle.

Calculs visuels — DAX

```
Cumul           = RUNNINGSUM ( [Ventes Totales] )
Écart vs préc.  = [Ventes Totales] - PREVIOUS ( [Ventes Totales] )
Moy. mobile 3   = MOVINGAVERAGE ( [Ventes Totales], 3 )
% du parent     = DIVIDE ( [Ventes Totales], COLLAPSE ( [Ventes Totales], ROWS ) )
```

NOUVEAUTÉ 2026
Copilot sait désormais **écrire et expliquer des requêtes DAX** dans la vue Requête DAX, en s'appuyant sur les descriptions et synonymes du modèle. Relisez toujours le code proposé avant de l'adopter.

5.9 Bonnes pratiques DAX

Nommage

- Mesures en langage métier : « CA Total », « Marge % »
- Jamais de nom de table devant une mesure
- Colonnes toujours qualifiées : `Table[Col]`
- Table de mesures dédiée, dossiers d'affichage

Performance

- Variables pour éviter les recalculs
- Filtres de colonnes plutôt que FILTER sur une table entière
- DIVIDE au lieu de « / »
- Itérer sur des colonnes à faible cardinalité
- Mesurer avec l'analyseur de performances

Lisibilité

- Indentation homogène (DAX Formatter, vue Requête DAX)
- Mesures de base réutilisées par les mesures avancées
- Commentaires pour les règles métier
- Descriptions renseignées pour chaque mesure

ASTUCE PRO

Construisez en couches : d'abord des mesures de base (**Ventes Totales**, **Coût Total**), puis des mesures dérivées (**Marge**, **Marge %**), enfin des mesures temporelles (**Marge % N-1**). Le débogage devient trivial.

CHAPITRE 6

06



Visualisation & data storytelling

Un bon rapport ne montre pas tout : il répond à une question. Choisir le bon visuel, hiérarchiser l'information et guider le regard du décideur.

DANS CE CHAPITRE

6.1 Choisir le bon visuel

6.2 Principes de design

6.3 Mise en page d'un rapport

6.4 Interactivité

6.5 Visuels d'analyse & IA

6.6 Accessibilité & thèmes

6.1 Choisir le bon visuel

Question posée	Visuel recommandé	À éviter
Comment évolue l'indicateur ?	Courbe, histogramme groupé par mois	Secteurs, cartes
Qui contribue le plus ?	Barres horizontales triées	Secteurs à plus de 5 parts
Quelle part du total ?	Anneau (≤ 5 parts), barres 100 %, treemap	Secteurs 3D
Écart à l'objectif ?	Carte KPI, jauge, barres avec ligne de référence	Tableaux bruts
Relation entre deux mesures ?	Nuage de points	Double axe sans légende
Où se situe l'activité ?	Carte choroplèthe, carte Azure Maps	Carte de bulles surchargée
Comment se décompose un écart ?	Cascade (waterfall), arborescence de décomposition	Empilements multiples
Valeurs précises à consulter ?	Tableau ou matrice avec mise en forme conditionnelle	Graphiques illisibles

Tableau 6.1 — Matrice de choix des visuels

6.2 Principes de design

1. Une question par page
Chaque page sert un objectif : pilotage, analyse, détail. Le titre de page énonce la question ou la conclusion.

2. Hiérarchie visuelle
Lecture en « Z » : KPI en haut à gauche, tendance au centre, détail en bas à droite.

3. Couleur avec intention
Une couleur d'accent pour ce qui compte, du gris pour le contexte, vert / rouge réservés aux écarts.

4. Moins d'encre
Supprimez quadrillages, bordures, légendes redondantes et décimales inutiles.

5. Titres qui concluent
« Le CA progresse de 12 % grâce au web » plutôt que « CA par canal ». Les titres dynamiques se créent avec une mesure.

6. Cohérence
Mêmes couleurs, polices, formats et positions de filtres sur toutes les pages.

Un titre de visuel qui s'adapte aux filtres — DAX

```
Titre Dynamique =
"Ventes " & SELECTEDVALUE ( Calendrier[Année], "toutes années" )
    & " : " & FORMAT ( [Croissance N/N-1 %], "+0,0 %;-0,0 %" ) & " vs N-1"
```

6.3 Mise en page d'un rapport

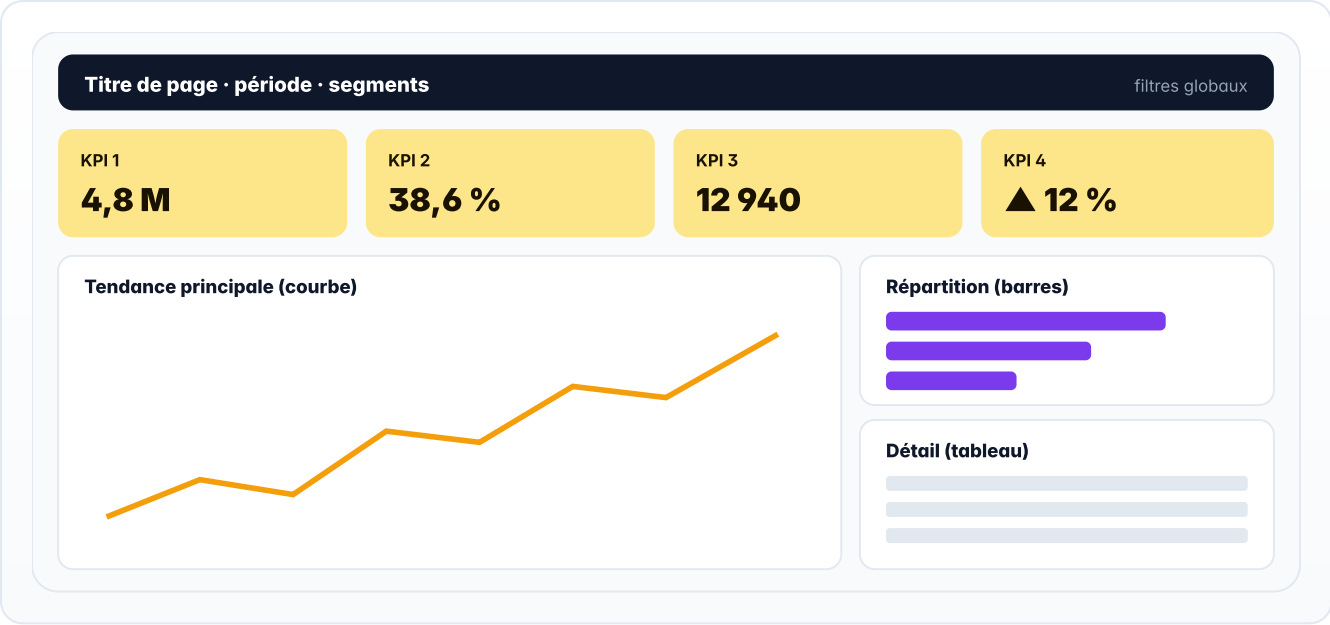


Figure 6.1 — Une grille de lecture éprouvée : contexte, KPI, tendance, répartition, détail

- Format de page 16:9, grille de 8 ou 16 px, marges constantes.
- 4 à 6 visuels par page au maximum ; le détail va sur une page d'extraction.
- Segments regroupés dans un volet de filtres ou un panneau escamotable (signets).
- Une page « À propos » : source, date d'actualisation, définitions des indicateurs.

6.4 Rendre le rapport interactif

Filtrage croisé
Un clic sur un visuel filtre les autres. Ajustez-le via **Format → Modifier les interactions**.

Extraction (drill-through)
Une page de détail accessible par clic droit depuis n'importe quel visuel, filtrée sur l'élément choisi.

Info-bulles de rapport
Une mini-page affichée au survol : contexte supplémentaire sans encombrer la page.

Signets & boutons
Basculer entre vues (graphique / tableau), réinitialiser les filtres, construire une navigation.

Paramètres de champs
Laisser l'utilisateur choisir la mesure ou l'axe affiché (CA, marge, quantité) dans un même visuel.

Synchronisation des segments
Un même segment pilote plusieurs pages (Affichage → Synchroniser les segments).

6.5 Visuels d'analyse et IA

Influenceurs clés
Identifie les facteurs qui expliquent une mesure ou un résultat (ex. ce qui augmente le taux de désabonnement).

Arborescence de décomposition
Décompose une mesure dimension par dimension, avec divisions automatiques « IA » vers la valeur la plus haute ou la plus basse.

Narration intelligente & Copilot
Résumés textuels générés à partir des visuels de la page, mis à jour avec les filtres.

Détection d'anomalies
Signale automatiquement les points inhabituels d'une série temporelle, avec une explication possible.

NOUVEAUTÉ 2026
Le bouton **Résumer** de Copilot, accessible depuis le ruban du rapport et l'en-tête des visuels, génère une synthèse des tendances, points saillants et évolutions à travers les pages.

6.6 Accessibilité et thèmes

- Contraste suffisant (ratio 4,5 : 1 pour le texte) ; ne jamais coder l'information par la seule couleur.
- Texte alternatif sur chaque visuel, ordre de tabulation logique (volet Sélection).
- Taille de police ≥ 12 pt pour les étiquettes, ≥ 18 pt pour les KPI.
- Un **thème JSON** d'entreprise : couleurs, polices et styles par défaut appliqués à tous les visuels.

Fichier de thème (Affichage → Thèmes → Parcourir) TEXTE

```
{
  "name": "Thème Entreprise 2026",
  "dataColors": [ "#F59E0B", "#7C3AED", "#0EA5A4", "#2563EB", "#E11D48", "#64748B" ],
  "foreground": "#0F172A",
  "background": "#FFFFFF",
  "tableAccent": "#F59E0B"
}
```

CHAPITRE 7

07



Publier, partager & gouverner

Du fichier local au contenu d'entreprise : espaces de travail, applications, sécurité des données, travail en équipe avec Git et usage maîtrisé de Copilot.

DANS CE CHAPITRE

7.1 Le service Power BI

7.2 Espaces de travail & applications

7.3 Sécurité au niveau des lignes

7.4 PBIP, PBIR & Git

7.5 Copilot & IA

7.6 Gouvernance

7.1 Le service Power BI

Depuis Desktop, **Accueil → Publier** envoie le modèle sémantique et le rapport dans un espace de travail du service (app.powerbi.com). On y configure ensuite l'actualisation, le partage et les alertes.

Rapports
Les pages interactives conçues dans Desktop.

Tableaux de bord
Une page unique de vignettes épinglées depuis plusieurs rapports, avec alertes sur les valeurs.

Modèles sémantiques
Le modèle publié, réutilisable par d'autres rapports (connexion active) ou dans Excel.

ASTUCE PRO
Séparez **un modèle sémantique partagé** et **plusieurs rapports légers** qui s'y connectent : une seule source de vérité, des chiffres identiques partout et une maintenance centralisée.

7.2 Espaces de travail, rôles et applications

Rôle	Peut…
Administrateur	Tout gérer, y compris les membres et la suppression de l'espace
Membre	Publier, modifier, partager et ajouter des contributeurs
Contributeur	Créer et modifier du contenu, sans gérer les accès
Lecteur	Consulter et interagir, sans modifier

Une **application** regroupe les rapports d'un espace de travail en un produit fini, avec sa navigation et des **audiences** distinctes (direction, managers, opérationnels). C'est le canal de diffusion recommandé à grande échelle.

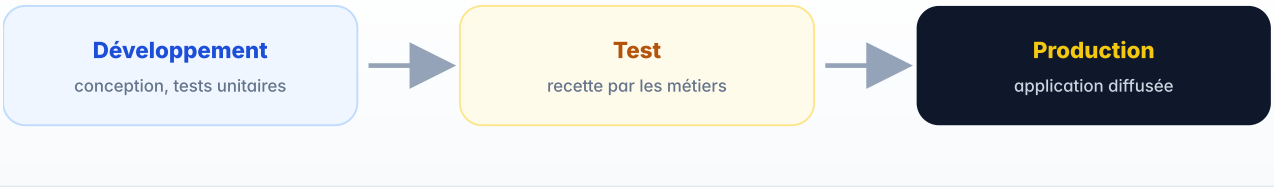


Figure 7.1 — Les pipelines de déploiement : promouvoir le contenu d'un environnement à l'autre

7.3 Sécurité au niveau des lignes (RLS)

La **RLS** restreint les lignes visibles selon l'utilisateur connecté : chaque directeur régional ne voit que sa région, avec un seul rapport.

Expressions de filtre RLS — DAX

```dax
// Rôle statique : Modélisation → Gérer les rôles → table Clients
[Région] = "Nord"

// Rôle dynamique : une table Habilitations (Email, Région) reliée à Clients
Habilitations[Email] = USERPRINCIPALNAME ()
```

1. Créer le rôle et son expression de filtre dans Desktop.
2. Tester avec **Afficher en tant que** (rôle + utilisateur fictif).
3. Publier, puis affecter les utilisateurs ou groupes Entra ID au rôle dans le service.
4. Pour masquer des colonnes ou des tables entières : sécurité au niveau des objets (OLS).

ATTENTION
La RLS ne s'applique pas aux **administrateurs, membres et contributeurs** de l'espace de travail. Les lecteurs finaux doivent avoir le rôle Lecteur ou passer par l'application.

7.4 PBIP, PBIR et travail en équipe avec Git

Le format historique **.pbix** est un fichier binaire unique. Le **projet Power BI (PBIP)** enregistre la solution sous forme de dossiers et de fichiers texte : le rapport au format **PBIR** (JSON) et le modèle au format **TMDL**.

Structure d'un projet PBIP — TEXTE

```
MonProjet.pbip
MonProjet.Report\
    definition\pages\...        <- une page = un fichier JSON (PBIR)
MonProjet.SemanticModel\
    definition\tables\Ventes.tmdl
    definition\relationships.tmdl
```

Une mesure décrite en TMDL — TMDL

```
measure 'Ventes Totales' = SUM ( Ventes[Montant] )
    formatString: #,##0
    displayFolder: Ventes
    description: "Chiffre d'affaires HT, toutes remises déduites"
```

Ce que cela change

- Historique des versions et retour arrière (Git)
- Revue de code : on voit exactement quelle mesure a changé
- Travail parallèle et fusion des modifications
- Modifications en masse par script

Mise en route

- Fichier → Enregistrer sous → Projet Power BI (.pbip)
- Initialiser un dépôt Git (Azure DevOps ou GitHub)
- Ou connecter l'espace de travail Fabric à Git
- Ajouter un `.gitignore` pour les caches locaux

NOUVEAUTÉ 2026

En 2026, **PBIR devient le format de rapport par défaut** dans le service et dans Desktop : les nouveaux rapports sont créés directement dans ce format texte, prêt pour le contrôle de version.

7.5 Copilot et IA : un assistant, pas un pilote automatique

Pour les créateurs

- Suggérer des pages de rapport à partir d'une demande
- Écrire, expliquer et corriger du DAX
- Générer les descriptions des mesures
- Préparer le modèle pour l'IA : schéma simplifié, instructions, réponses vérifiées

Pour les lecteurs

- Résumer un rapport ou une page
- Poser des questions en langage naturel
- Retrouver le bon rapport ou modèle
- Discuter avec ses données sur mobile

ATTENTION

Copilot nécessite une capacité Fabric adaptée (ou PPU selon les fonctions) et son activation par l'administrateur. La qualité des réponses dépend directement de la qualité du modèle : noms clairs, descriptions, synonymes et relations propres.

7.6 Gouvernance et cycle de vie

Certifier

Marquez les modèles validés comme **Promus** ou **Certifiés** pour guider les utilisateurs vers la source de vérité.

Protéger

Étiquettes de confidentialité Microsoft Purview, conservées lors des exports vers Excel, PowerPoint ou PDF.

Surveiller

Vue de **lignage**, analyse d'impact avant modification, journaux d'activité et statistiques d'utilisation.

ANNEXE A

Aide-mémoire DAX

Les fonctions à connaître par cœur, regroupées par usage. À garder sous la main.

AGRÉGATION

`SUM / AVERAGE`	Somme, moyenne d'une colonne
`MIN / MAX`	Valeurs extrêmes
`COUNTROWS`	Nombre de lignes d'une table
`DISTINCTCOUNT`	Nombre de valeurs uniques
`SUMX / AVERAGEX`	Itérer puis agréger une expression

FILTRES & CONTEXTE

`CALCULATE`	Évaluer dans un contexte modifié
`FILTER`	Table filtrée par une condition
`ALL / REMOVEFILTERS`	Retirer des filtres
`ALLSELECTED`	Garder les filtres externes
`KEEPFILTERS`	Intersecter au lieu d'écraser
`USERELATIONSHIP`	Activer une relation inactive

INTELLIGENCE TEMPORELLE

`TOTALYTD / QTD / MTD`	Cumuls à date
`SAMEPERIODLASTYEAR`	Même période N-1
`DATEADD`	Décaler de n jours / mois / années
`DATESINPERIOD`	Fenêtre glissante
`PARALLELPERIOD`	Période complète décalée
`DATESBETWEEN`	Plage de dates explicite

LOGIQUE

`IF`	Condition simple
`SWITCH ( TRUE () )`	Conditions multiples lisibles
`AND / OR / NOT`	Opérateurs logiques (ou && ||)
`IN { … }`	Appartenance à une liste
`COALESCE`	Première valeur non vide
`ISBLANK`	Tester l'absence de valeur

TABLES

`VALUES / DISTINCT`	Valeurs uniques d'une colonne
`SUMMARIZECOLUMNS`	Table agrégée
`ADDCOLUMNS`	Ajouter des colonnes calculées
`TOPN`	N premières lignes
`CALENDAR`	Table de dates
`SELECTCOLUMNS`	Projeter des colonnes

CLASSEMENT & SÉLECTION

`RANKX`	Rang selon une expression
`SELECTEDVALUE`	Valeur unique du filtre courant
`HASONEVALUE`	Une seule valeur filtrée ?
`ISINSCOPE`	Niveau courant d'une hiérarchie
`LOOKUPVALUE`	Rechercher une valeur
`RELATED`	Valeur de la table « un »

MATHÉMATIQUES & TEXTE

`DIVIDE`	Division sûre (BLANK si /0)
`ROUND / INT`	Arrondir, partie entière
`FORMAT`	Mettre en forme un nombre / une date
`CONCATENATEX`	Concaténer les valeurs d'une table
`LEFT / RIGHT / MID`	Extraire du texte
`SUBSTITUTE`	Remplacer du texte

CALCULS VISUELS & SÉCURITÉ

`RUNNINGSUM`	Cumul dans le visuel
`MOVINGAVERAGE`	Moyenne mobile dans le visuel
`PREVIOUS / NEXT`	Ligne précédente / suivante
`COLLAPSE`	Valeur du niveau parent
`USERPRINCIPALNAME`	Utilisateur connecté (RLS)
`CROSSFILTER`	Changer le sens d'une relation

ANNEXE B

Check-list performance & qualité

Avant chaque mise en production, passez en revue ces points. Un rapport rapide est un rapport utilisé.

Données & Power Query

- [x] Seules les colonnes utiles sont chargées
- [x] Les filtres sont appliqués à la source
- [x] Le query folding est préservé
- [x] Les types sont explicites et justes
- [x] Les requêtes intermédiaires ne sont pas chargées
- [x] Date/heure automatique est désactivée

Modèle

- [x] Schéma en étoile, dimensions aplaties
- [x] Relations 1 : * sur clés entières
- [x] Filtrage unique par défaut
- [x] Table de dates marquée et continue
- [x] Colonnes techniques masquées
- [x] Descriptions et dossiers d'affichage renseignés

DAX

- [x] Mesures explicites pour tous les indicateurs
- [x] Variables pour les sous-calculs répétés
- [x] DIVIDE plutôt que « / »
- [x] Pas de FILTER sur des tables entières sans nécessité
- [x] Pas d'IFERROR dans les mesures
- [x] Formules indentées et commentées

Rapport

- [x] 4 à 6 visuels par page au maximum
- [x] Analyseur de performances : aucun visuel > 1 s
- [x] Titres qui énoncent la conclusion
- [x] Couleurs et formats cohérents (thème)
- [x] Texte alternatif et contraste vérifiés
- [x] Page « À propos » avec définitions et date d'actualisation

ASTUCE PRO

Outils complémentaires gratuits de la communauté : **DAX Studio** (mesurer et analyser les requêtes), **Tabular Editor** (édition avancée du modèle, Best Practice Analyzer) et **Measure Killer** (repérer les colonnes et mesures inutilisées).

ANNEXE C

Feuille de route : 30 jours pour maîtriser Power BI

Une heure par jour, un livrable par semaine. À la fin du mois, vous disposez d'un portfolio présentable en entretien.

JOURS 1 – 7 · FONDATIONS

Installer, importer, visualiser

Chapitres 1 et 2. Connectez un fichier Excel et un CSV, créez votre premier rapport d'une page.
Livrable : un tableau de bord de ventes simple.

JOURS 8 – 14 · PRÉPARATION

Power Query et langage M

Chapitre 3. Nettoyez un jeu de données « sale » : dépivotage, fusions, regroupements, fonctions.
Livrable : un pipeline de nettoyage documenté.

JOURS 15 – 21 · MODÈLE & DAX

Schéma en étoile et mesures

Chapitres 4 et 5. Construisez le modèle, la table de dates, 20 mesures dont N-1, YTD et moyenne mobile.
Livrable : un modèle sémantique propre.

JOURS 22 – 30 · RAPPORT PRO

Storytelling, publication, sécurité

Chapitres 6 et 7. Trois pages, navigation par boutons, RLS, publication et projet PBIP sous Git.
Livrable : un projet complet à partager.

PL-300

Microsoft Certified : Power BI Data Analyst Associate — la certification de référence pour valider ces compétences.

DP-600

Microsoft Certified : Fabric Analytics Engineer Associate — l'étape suivante, orientée Fabric et ingénierie analytique.

Jeux de données pour pratiquer

- Exemples Microsoft : Contoso, AdventureWorks, Financial Sample
- Données ouvertes nationales (statistiques, éducation, économie)
- Vos propres exports Excel métier, anonymisés

CONCLUSION

Les fondamentaux d'abord, les nouveautés ensuite.

Félicitations : vous avez parcouru l'ensemble de la chaîne Power BI, de la connexion aux sources jusqu'à la diffusion sécurisée d'un rapport. Les outils évolueront encore — chaque mois apporte son lot de fonctionnalités — mais trois principes resteront vrais.

01

Un bon modèle simplifie tout

Schéma en étoile, table de dates, mesures explicites.

02

Le contexte est la clé du DAX

Comprendre filtres et transitions, c'est maîtriser CALCULATE.

03

Un rapport répond à une question

Moins de visuels, des titres qui concluent, une lecture guidée.

Pour continuer à progresser

- Pratiquez chaque semaine sur des données réelles de votre métier.
- Suivez la synthèse mensuelle des nouveautés publiée par Microsoft et la documentation Microsoft Learn.
- Rejoignez la communauté Fabric & Power BI : forums, groupes d'utilisateurs, défis de visualisation.
- Préparez la certification PL-300, puis DP-600 pour l'écosystème Fabric.

NOTE DE L'AUTEUR

Les noms de fonctionnalités, licences et limites évoluent régulièrement : ce guide reflète l'état de Power BI fin 2026. Pour toute décision d'achat ou d'architecture, vérifiez la documentation officielle de Microsoft.

MERCI POUR VOTRE LECTURE

Ce guide vous a été utile ?

Aidez-le à toucher d'autres analystes, étudiants et décideurs qui veulent transformer leurs données en décisions.

Réagissez
pour que le guide atteigne plus de personnes de votre réseau.

Commentez
votre fonction DAX préférée — ou la question qui vous bloque.

Partagez
avec une personne qui débute en Power BI.

Suivez-moi
pour les prochains guides : Fabric, DAX avancé, IA appliquée à la donnée.

EI

EL KHLIFI Imad
Doctorant en économie & gestion · Analyse de données & Business Intelligence