

Annexe A – Aide-mémoire DAX : les formules indispensables

Référence complète officielle : <https://learn.microsoft.com/fr-fr/dax/> Convention : **[Mesure]** en majuscule = mesure créée au préalable. **T[Col]** = table[colonne]. - -

1. Agrégations

```
Total Ventes = SUM ( T[Montant] ) -- somme simple
Nb Lignes = COUNTROWS ( T ) -- comptage de lignes
Nb Distinct = DISTINCTCOUNT ( T[ID] ) -- valeurs uniques
Moyenne = AVERAGE ( T[Montant] )
Min / Max = MIN ( T[Date] ) / MAX ( T[Date] )
Compte non vide= COUNTNONBLANK ( T[Col] )
```

2. Itérateurs X (expression ligne par ligne)

```
CA = SUMX ( T, T[Qte] * T[PrixUnitaire] )
CA Moyen/Ligne = AVERAGEX ( T, T[Qte] * T[PrixUnitaire] )
Meilleur Jour = MAXX ( VALUES(DimDate[Date]), [Total Ventes] )
Somme Conditionnelle = SUMX ( FILTER(T, T[Categorie]="A"), T[Montant] )
```

SUM(col) ≠ SUMX(t, expr): le premier additionne une colonne existante ; le second évalue une expression par ligne.

3. CALCULATE & contexte de filtre

```
- - Filtre simple
Ventes Web = CALCULATE ( [Total Ventes], Magasins[Canal] = "Web" )
- - Multi-filtres (ET)
Ventes Web 2024= CALCULATE ( [Total Ventes],
  Magasins[Canal] = "Web",
  DimDate[Annee] = 2024 )
- - OU logique
Nord ou Sud = CALCULATE ( [Total Ventes],
  Region IN { "Hauts-de-France", "Grand Est" } )
- - Retirer les filtres
Total Général = CALCULATE ( [Total Ventes], REMOVEFILTERS() )
% du Total = DIVIDE ( [Total Ventes], CALCULATE([Total Ventes], ALL(Produits)) )
- - Garder ses filtres en plus
Part Catégorie = DIVIDE ( [Total Ventes],
  CALCULATE ( [Total Ventes], KEEPFILTERS(...) ) )
```

Fonctions de modification de contexte:

Fonction	Effet
ALL(table/col)	retire tous les filtres
ALLEXCEPT(t, col)	tout retire SAUF col
REMOVEFILTERS()	idem ALL, plus lisible
KEEPFILTERS(expr)	intersecte avec les filtres existants

Fonction	Effet
USERRELATIONSHIP(a,b)	active une relation inactive
CROSSFILTER(dir)	change la direction d'une relation

4. Relations & lookups

```
- - Colonne calculée dans la table many :
Region Client = RELATED ( Clients[Region] )
- - Mesure descendant la relation :
Nb Produits Vendus = COUNTROWS ( RELATEDTABLE ( Ventes ) )
- - Sans relation (dépannage) – à éviter en prod :
Nom Prod = LOOKUPVALUE ( Produits[Nom], Produits[ID], Ventes[IDProduit] )
```

5. Logiques & sécurisation

```
Statut = IF ( [Atteinte %] >= 1, "OK", "KO" )
Tranche = SWITCH ( TRUE(),
  [Remise] >= 0.15, "Forte",
  [Remise] >= 0.05, "Moyenne",
  "Aucune" )
Ratio Sécurisé = DIVIDE ( num, den ) -- gère /0 → BLANK
Ratio + Alt = DIVIDE ( num, den, 0 ) -- alternative au lieu de BLANK
Valeur Testée = COALESCE ( [Mesure], 0 ) -- remplace BLANK
```

6. Variables : écrire comme un senior

```
Croissance % =
VAR _actuel = [Total Ventes]
VAR _n1 = [Total Ventes N-1]
VAR _delta = _actuel - _n1
RETURN
  IF ( NOT ISBLANK(_n1), DIVIDE ( _delta, _n1 ) )
```

Bénéfices : lisibilité • performance (évaluation unique) • débogage étape par étape.

7. Intelligence temporelle

- (Prérequis absolu : table DimDate continue marquée comme table de dates)*

```
CA YTD = TOTALYTD ( [Total Ventes], DimDate[Date] )
CA MTD = TOTALMTD ( [Total Ventes], DimDate[Date] )
CA QTD = TOTALQTD ( [Total Ventes], DimDate[Date] )
CA N-1 = CALCULATE ( [Total Ventes], SAMEPERIODLASTYEAR ( DimDate[Date] ) )
Mois -1 = CALCULATE ( [Total Ventes], DATEADD ( DimDate[Date], -1, MONTH ) )
12 mois glissants = CALCULATE ( [Total Ventes],
  DATESINPERIOD ( DimDate[Date], MAX(DimDate[Date]), -12, MONTH ) )
Cumul depuis début = CALCULATE ( [Total Ventes],
  FILTER ( ALL(DimDate), DimDate[Date] <= MAX(DimDate[Date]) ) )
Moyenne mobile 3 mois =
AVERAGEX (
  DATESINPERIOD ( DimDate[Date], MAX ( DimDate[Date] ), -3, MONTH ),
  [Total Ventes]
)
```

Comparaisons N vs N-1 prêtes à l'emploi

```
Écart N-1 € = [Total Ventes] - [Total Ventes N-1]
Écart N-1 % = VAR a=[Total Ventes] VAR b=[Total Ventes N-1] RETURN DIVIDE(a-b, b)
```

8. Classements & Top N

```
Rang Produit = RANKX ( ALL(Produits[NomProduit]), [Total Ventes],, DESC, DENSE )
Top 5 CA = CALCULATE ( [Total Ventes],
  TOPN ( 5, VALUES(Produits[NomProduit]), [Total Ventes], DESC ) )
Top 5 + Autres = -- pattern senior :
VAR _top = TOPN(5, VALUES(Produits[NomProduit]), [Total Ventes])
RETURN CALCULATE([Total Ventes], KEEPFILTERS(_top))
```

9. Tables virtuelles utiles

Fonction	Rôle
VALUES(Col)	valeurs distinctes visibles (respecte filtres)
DISTINCT(Col)	proche VALUES (sans ligne BLANK ajoutée)
ALLSELECTED()	respecte la sélection utilisateur (segments) — idéal pour %
SUMMARIZE(t, c1, c2)	agrégat groupBy (préférer ADDCOLUMNS+SUMMARIZE pro)
ADDCOLUMNS(tbl, "x", expr)	ajoute colonnes calculées à une table virtuelle
TOPN(n, tbl, ordre)	top n lignes d'une table

```
% sur sélection = DIVIDE ( [Total Ventes],
  CALCULATE ( [Total Ventes], ALLSELECTED ( ) ) )
```

10. Texte & formatage

```
Libellé KPI = FORMAT ( [Taux Marge %], "0.0%" )
Période = FORMAT ( MAX(DimDate[Date]), "MMMM YYYY", "fr-FR" )
Concat = "CA : " & FORMAT([Total Ventes], "#,0 €")
Longueur = LEN / LEFT / RIGHT / MID / SUBSTITUTE / TRIM / UPPER...
```

Mini-quiz express (réponses en bas)

- Différence entre SUM et SUMX?
- Que fait ALL(Produits)?
- Prérequis pour SAMEPERIODYEAR?
- Pourquoi préférer DIVIDE à /?
- À quoi sert une variable _var? **Réponses:**
- SUM additionne une colonne existante ; SUMX évalue une expression par ligne puis somme.
- Il supprime les filtres provenant de la table Produits (dénominateurs de parts, totaux).
- Une table de dates dédiée, continue, sans trou, marquée comme table de dates.
- DIVIDE gère la division par zéro (BLANK/valeur alternative) sans erreur.

10. Stocker un résultat intermédiaire : lisibilité + performance + debug.