



POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Opérateurs de base en DAX

Problème Traité	Expression DAX	Exemple d'Utilisation
Additionner deux nombres	Nombre1 + Nombre2	5 + 3 donne 8
Soustraire un nombre d'un autre	Nombre1 - Nombre2	10 - 2 donne 8
Multiplier deux nombres	Nombre1 * Nombre2	4 * 2 donne 8
Diviser un nombre par un autre (gestion des zéros)	DIVIDE(Nombre1, Nombre2)	DIVIDE(16, 2) donne 8, gestion sûre de la division par 0
Comparer si un nombre est supérieur à un autre	Nombre1 > Nombre2	5 > 3 renvoie VRAI
Nombre supérieur ou égal à un autre	Nombre1 >= Nombre2	5 >= 5 renvoie VRAI
Vérifier si deux valeurs sont égales	Valeur1 = Valeur2	2 = 2 renvoie VRAI
Vérifier si deux valeurs sont différentes	Valeur1 <> Valeur2	2 <> 3 renvoie VRAI
Concaténer des chaînes de texte	Texte1 & Texte2	"Bonjour" & " " & "Monde" donne Bonjour Monde
Calculer le reste de la division	MOD(Nombre1, Nombre2)	MOD(10, 3) donne 1
Élever un nombre à une puissance	POWER(Nombre, Puissance)	POWER(2, 3) donne 8

Fonctions Logiques en DAX

Problème Traité	Expression DAX
Appliquer une condition	IF(Condition, RésultatSiVrai, RésultatSiFaux)
Combinaison de conditions (ET logique)	AND(Condition1, Condition2)
Combinaison de conditions (OU logique)	OR(Condition1, Condition2)
Inverser une condition	NOT(Condition)
Sélectionner parmi plusieurs options	SWITCH(Expression, Valeur1, Résultat1, Valeur2, Résultat2, ..., RésultatParDéfaut)
Gérer les erreurs dans une expression	IFERROR(Expression, ValeurSiErreur)



POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions Mathématiques et Statistiques en DAX

Problème Traité	Expression DAX	Exemple
Calculer la somme d'une colonne	SUM(Colonne)	SUM(Ventes[Montant]) pour calculer la somme totale des ventes
Trouver la moyenne d'une colonne	AVERAGE(Colonne)	AVERAGE(Ventes[Quantité]) pour la moyenne des quantités vendues
Calculer la médiane d'une colonne	MEDIAN(Colonne)	MEDIAN(Ventes[Montant]) pour la médiane des montants de ventes
Calculer la moyenne géométrique	GEOMEAN(Colonne)	GEOMEAN(Ventes[Quantité]) pour la moyenne géométrique des quantités vendues
Compter le nombre de valeurs	COUNT(Colonne)	COUNT(Ventes[ArticleID]) pour le nombre total d'articles vendus
Diviser deux nombres	DIVIDE(Nombre1, Nombre2)	DIVIDE(SUM(Ventes[Montant]), COUNT(Ventes[ArticleID])) pour le montant moyen par article
Trouver la valeur minimale	MIN(Colonne)	MIN(Ventes[Montant]) pour le montant de vente le plus bas
Trouver la valeur maximale	MAX(Colonne)	MAX(Ventes[Montant]) pour le montant de vente le plus élevé
Compter le nombre de lignes	COUNTROWS(Table)	COUNTROWS(Ventes) pour le nombre total de transactions de vente
Compter valeurs distinctes	DISTINCTCOUNT(Colonne)	DISTINCTCOUNT(Ventes[ClientID]) pour le nombre de clients uniques
Classer une valeur	RANKX(Table, Expression)	RANKX(All(Ventes), SUM(Ventes[Montant])) pour classer les ventes par montant total

POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions de Texte en DAX

Problème Traité	Expression DAX	Exemple
Concaténer des chaînes de caractères	CONCATENATE(<Text1>, <Text2>)	CONCATENATE(Client[Nom], Client[Prénom]) pour fusionner nom et prénom
Convertir en majuscules	UPPER(<TextColumn>)	UPPER(Client[Nom]) pour convertir les noms en majuscules
Convertir en minuscules	LOWER(<TextColumn>)	LOWER(Client[Nom]) pour convertir les noms en minuscules
Trouver la longueur d'une chaîne	LEN(<TextColumn>)	LEN(Client[Commentaire]) pour la longueur d'un commentaire
Remplacer une sous-chaîne	REPLACE(<Text>, <Start>, <Length>, <NewText>)	REPLACE(Client[Adresse], 1, 5, "Rue") pour remplacer les premiers caractères de l'adresse
Extraire une sous-chaîne	MID(<Text>, <Start>, <Length>)	MID(Client[Email], 2, 5) pour extraire une partie de l'email
Trouver la position d'une sous-chaîne	SEARCH(<SubText>, <Text>, <Start>, <NotFound>)	SEARCH("@", Client[Email], 1, -1) pour trouver la position du symbole @ dans un email
Enlever les espaces superflus	TRIM(<TextColumn>)	TRIM(Client[Commentaire]) pour enlever les espaces superflus
Vérifier si un texte contient une sous-chaîne	CONTAINSSTRING(<TextColumn>, <SubText>)	CONTAINSSTRING(Client[Commentaire], "urgent") pour vérifier si 'urgent' est dans un commentaire
Concaténer des valeurs avec un séparateur	CONCATENATEX(<Table>, <Expression>, <Delimiter>)	CONCATENATEX(Clients, Clients[Nom], ", ") pour créer une liste de noms séparés par une virgule
Formater une valeur	FORMAT(<Value>, <FormatString>)	FORMAT(DimDate[Date], "DD/MM/YYYY") pour formater les dates de vente



POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions Dates et Temps en DAX

Problème Traité	Expression DAX	Exemple
Créer une séquence de dates	CALENDAR(<StartDate>, <EndDate>)	CALENDAR("2024-01-01", "2024-12-31") pour un calendrier 2024
Convertir année, mois, jour en date	DATE(<Year>, <Month>, <Day>)	DATE(2024, 12, 31) pour le 31/12/2024
Calculer l'année en cours	YEAR(TODAY())	Donne l'année actuelle, ex. 2024
Trouver le premier jour du mois	STARTOFMONTH(<DateColumn>)	STARTOFMONTH(DimDate[Date]) pour le 1er jour du mois
Calculer l'âge à partir de la date de naissance	DATEDIFF(<BirthDateColumn>, TODAY(), YEAR)	DATEDIFF(Clients[DateNaissance], TODAY(), YEAR) pour l'âge des clients
Obtenir le numéro de la semaine	WEEKNUM(<DateColumn>)	WEEKNUM(DimDate[Date]) pour la semaine de chaque vente
Extraire le mois d'une date	MONTH(<DateColumn>)	MONTH(DimDate[Date]) pour le mois de chaque vente
Convertir une chaîne en date	DATEVALUE(<DateString>)	DATEVALUE("2024-12-31") pour la conversion en date
Trouver le dernier jour du mois	EOMONTH(<DateColumn>, <MonthsOffset>)	EOMONTH(DimDate[Date], 0) pour le dernier jour du mois
Ajouter un nombre de mois à une date	EDATE(<DateColumn>, <MonthsOffset>)	EDATE(DimDate[Date], 3) pour ajouter 3 mois à la date
Obtenir la date actuelle	TODAY()	TODAY() pour la date du jour
Extraire le trimestre d'une date	QUARTER(<DateColumn>)	



POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Contexte de Calcul et Filtres en DAX

Problème Traité	Expression DAX	Exemple
Modifier le contexte de filtre pour un calcul	CALCULATE(<Expression>, <Filter>)	CALCULATE(SUM(Ventes[Montant]), Clients[Region] = "Paris") pour le total des ventes à Paris
Obtenir une table filtrée	FILTER(<TableOrColumn>, <Filter>)	FILTER(ALL(Clients), Clients[Region] = « Paris ») pour la table des Clients à Paris
Ignorer les filtres appliqués sur une table	ALL(<TableOrColumn>)	CALCULATE(SUM(Ventes[Montant]), ALL(Clients)) pour les ventes totales sans filtre Clients
Conserver certains filtres tout en ignorant d'autres	ALLEXCEPT(<Table>, <Column>)	CALCULATE(SUM(Ventes[Montant]), ALLEXCEPT(Clients, Clients[Region])) pour les ventes par région
Appliquer un contexte de filtre avec les sélections actuelles	ALLSELECTED(<TableOrColumn>)	CALCULATE(SUM(Ventes[Montant]), ALLSELECTED(Clients[Region])) pour les ventes des régions sélectionnées
Supprimer tous les filtres d'une table ou colonne	REMOVEFILTERS(<TableOrColumn>)	CALCULATE(SUM(Ventes[Montant]), REMOVEFILTERS(Clients)) pour les ventes totales sans filtres Clients

Fonctions Itératives en DAX

Problème Traité	Expression DAX	Exemple
Somme personnalisée sur une table	SUMX(Table, Expression)	SUMX(Ventes, Ventes[Quantité] * Ventes[PrixUnitaire]) pour le total des ventes
Moyenne personnalisée sur une table	AVERAGEX(Table, Expression)	AVERAGEX(Ventes, Ventes[Montant]) pour la moyenne des montants de vente
Minimum personnalisé sur une table	MINX(Table, Expression)	MINX(Ventes, Ventes[Montant]) pour trouver le montant de vente le plus bas
Maximum personnalisé sur une table	MAXX(Table, Expression)	MAXX(Ventes, Ventes[Montant]) pour le montant de vente le plus élevé
Compter avec condition	COUNTX(Table, Expression)	COUNTX(Ventes, IF(Ventes[Montant] > 1000, 1, BLANK())) pour compter les ventes supérieures à 1000 €

POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions de Time Intelligence en DAX

Problème Traité	Expression DAX	Exemple
Comparer avec la période correspondante de l'année précédente	SAMEPERIODLASTYEAR(<DateColumn>)	CALCULATE(SUM(Ventes[Montant]), SAMEPERIODLASTYEAR(DimDate[Date]))
Obtenir une période de 3 mois	DATESINPERIOD(<DateColumn>, -3, MONTH))	CALCULATE (AVERAGEX (VALUES (D_Date[Mois]), [Sales Amount]), DATESINPERIOD (D_Date[Date], MAX (D_Date[Date]), - 3, MONTH)) pour calculer la moyenne mobile de 3 mois
Obtenir la date du dernier jour de l'année en cours	ENDOFYEAR(<DateColumn>)	ENDOFYEAR(DimDate[Date])
Calculer le total depuis le début de l'année	TOTALYTD(<Expression>, <DateColumn>)	TOTALYTD(SUM(Ventes[Montant]), DimDate[Date])
Calculer le total depuis le début du mois	TOTALMTD(<Expression>, <DateColumn>)	TOTALMTD(SUM(Ventes[Montant]), DimDate[Date])
Calculer le total depuis le début du trimestre	TOTALQTD(<Expression>, <DateColumn>)	TOTALQTD(SUM(Ventes[Montant]), DimDate[Date])
Calculer les ventes de l'année précédente	DATEADD(<DateColumn>, <NumberOfIntervals>, <Interval>)	CALCULATE(SUM(Ventes[Montant]), DATEADD(DimDate[Date], -1, YEAR)) pour les ventes de l'année précédente
Calculer les ventes pour une période spécifique	DATESBETWEEN(<DateColumn>, <StartDate>, <EndDate>)	CALCULATE(SUM(Ventes[Montant]), DATESBETWEEN(DimDate[Date], "2024-01-01", "2024-12-31")) pour les ventes de 2024
Calculer les ventes du dernier mois complet	PREVIOUSMONTH(<DateColumn>)	CALCULATE(SUM(Ventes[Montant]), PREVIOUSMONTH(DimDate[Date])) pour les ventes du mois précédent
Calculer les ventes du dernier trimestre complet	PREVIOUSQUARTER(<DateColumn>)	CALCULATE(SUM(Ventes[Montant]), PREVIOUSQUARTER(DimDate[Date])) pour les ventes du trimestre précédent

POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions de manipulation des tables en DAX

Problème Traité	Expression DAX	Exemple
Créer un résumé de table	SUMMARIZE(Table, GroupBy_ColumnName, ...)	SUMMARIZE(Ventes, Ventes[Produit], "TotalVentes", SUM(Ventes[Montant]))
Retirer les doublons d'une table	DISTINCT(Table)	DISTINCT(Clients[Region])
Ajouter des colonnes calculées	ADDCOLUMNS(Table, "NewColumn", Expression)	ADDCOLUMNS(Produits, "PrixTTC", Produits[PrixHT]*1.2)
Sélectionner des colonnes calculées	SELECTCOLUMNS(Table, "NewColumn", Expression)	SELECTCOLUMNS(Clients, "ClientID", Clients[ID], "Age", YEAR(NOW()) - Clients[AnnéeNaissance])
Grouper une table	GROUPBY(Table, GroupBy_ColumnName, ...)	GROUPBY(Ventes, Ventes[Produit], "TotalVentes", SUMX(CURRENTGROUP(), Ventes[Montant]))
Trouver les lignes communes	INTERSECT(Table1, Table2)	INTERSECT(ClientsActuels, ClientsPotentiels)
Joindre deux tables	NATURALINNERJOIN(Table1, Table2)	NATURALINNERJOIN(Ventes, Stock)
Joindre avec préservation des lignes	NATURALLEFTOUTERJOIN(Table1, Table2)	NATURALLEFTOUTERJOIN(Clients, Commandes)
Combiner les lignes de tables	UNION(Table1, Table2, ...)	UNION(TousClients, NouveauxClients)

Fonctions de relations en DAX

Problème Traité	Expression DAX	Exemple et Explication
Relier des tables (1 vers plusieurs)	RELATED(<ColumnName>)	RELATED(Clients[Nom]) : Renvoie la valeur du côté '1' de la relation (ici, nom du client).
Accéder à une table liée (plusieurs vers 1)	RELATEDTABLE(<Table>)	RELATEDTABLE(Clients) : Renvoie une table du côté 'plusieurs' de la relation (données liées Clients).
Utiliser une relation spécifique entre les tables	USERELATIONSHIP(<Column1>, <Column2>)	CALCULATE(SUM(Ventes[Montant]), USERELATIONSHIP(Ventes[ClientID], Clients[ID])) : Utilise une relation spécifique pour calcul.
Modifier le comportement de filtrage entre les tables	CROSSFILTER(<ColumnName1>, <ColumnName2>, <Mode>)	CALCULATE(SUM(Ventes[Montant]), CROSSFILTER(Ventes[ClientID], Clients[ID], "None")) : Désactive la relation entre les tables Ventes et Clients.
Créer une relation virtuelle entre tables	TREATAS(<Table1>, <Table2>)	CALCULATE(SUM(Ventes[Montant]), TREATAS(VALUES(Clients[Region]), Ventes[Region])) : Applique les filtres de 'Clients' à 'Ventes'.
Rechercher une valeur en fonction de critères	LOOKUPVALUE(<ResultColumn>, <SearchColumn>, <SearchValue>)	LOOKUPVALUE(Clients[Nom], Clients[ID], Ventes[ClientID]) : Trouve le nom du client correspondant à l'ID dans 'Ventes'.



POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonctions d'information en DAX

Problème Traité	Expression DAX	Exemple
Vérifier si une valeur est numérique	ISNUMBER(<Value>)	ISNUMBER(Ventes[Quantité]) pour vérifier si 'Quantité' est numérique
Vérifier si une valeur est une erreur	ISERROR(<Value>)	ISERROR(DIVIDE(Ventes[Montant], 0)) pour vérifier la division par zéro
Obtenir le nom d'une colonne	SELECTEDVALUE(<ColumnName>)	SELECTEDVALUE(Ventes[Produit]) pour obtenir le produit sélectionné
Vérifier si un contexte de filtre est appliqué	HASONEVALUE(<ColumnName>)	HASONEVALUE(Clients[Region]) pour vérifier si une seule région est sélectionnée
Retourner l'identifiant principal de l'utilisateur	USERPRINCIPALNAME()	USERPRINCIPALNAME() pour obtenir l'identifiant principal de l'utilisateur actuel
Vérifier si un filtre spécifique est appliqué	HASONEFILTER(<ColumnName>)	HASONEFILTER(Ventes[Catégorie]) pour vérifier si une seule catégorie est filtrée
Vérifier si le résultat d'une expression est vide	ISBLANK(<Expression>)	ISBLANK(SUM(Ventes[Remise])) pour vérifier si le total des remises est vide
Vérifier si un filtre est appliqué sur une colonne	ISFILTERED(<ColumnName>)	ISFILTERED(Clients[Region]) pour vérifier si un filtre est appliqué sur 'Region'
Vérifier si un croisement de filtres est appliqué	ISCROSSFILTERED(<TableName>, <ColumnName>)	ISCROSSFILTERED(Ventes, Ventes[Produit]) pour vérifier si un croisement de filtres est appliqué sur 'Produit'

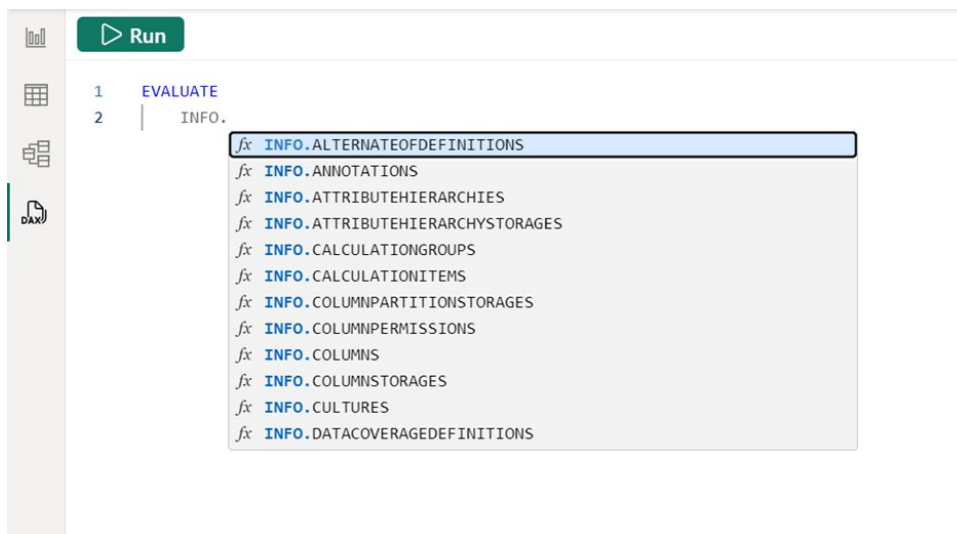
POWER BI - BUSINESS INTELLIGENCE

ANTI-SECHE DAX

Fonction de Fenêtrage en DAX

Problème Traité	Expression DAX	Exemple d'Utilisation
Récupérer une ligne à une position absolue	INDEX(<Position> [, <Arguments> ...])	INDEX(1, Sales) pour obtenir la première ligne de la table Ventes
Déplacer une ligne dans une partition spécifiée	OFFSET(<Delta> [, <Arguments> ...])	OFFSET(1, Sales) pour déplacer d'une ligne dans la table Ventes
Récupérer une plage de lignes dans une partition	WINDOW(<From>, <To> [, <Arguments> ...])	WINDOW(1, 10, Sales) pour obtenir une plage de dix lignes à partir de la première ligne de Ventes
Partitionner les données pour le fenêtrage	PARTITIONBY(<ColumnName> [, <ColumnName>])	PARTITIONBY(Sales[Region]) pour partitionner les données par région dans la table Ventes
Ordonner les données dans chaque partition	ORDERBY(<Expression> [, <Order> ...])	ORDERBY(Sales[Date], ASC, Sales[Amount], DESC) pour trier par date croissante et montant décroissant

Fonctions INFO en DAX



Les fonctions INFO DAX offrent un accès direct à des informations détaillées sur les modèles de données Power BI.

Elles permettent une exploration approfondie et facilitent la documentation des modèles, en donnant des informations sur les tables, les colonnes et les mesures sans avoir à recourir à une syntaxe de requête distincte. Ces fonctions ne sont pas destinées à être utilisées dans des tables calculées ou des mesures mais sont précieuses pour interroger et comprendre les métadonnées d'un modèle dans Power BI.

Pour plus d'informations, veuillez consulter l'article complet sur le [blog](#) de Power BI.