



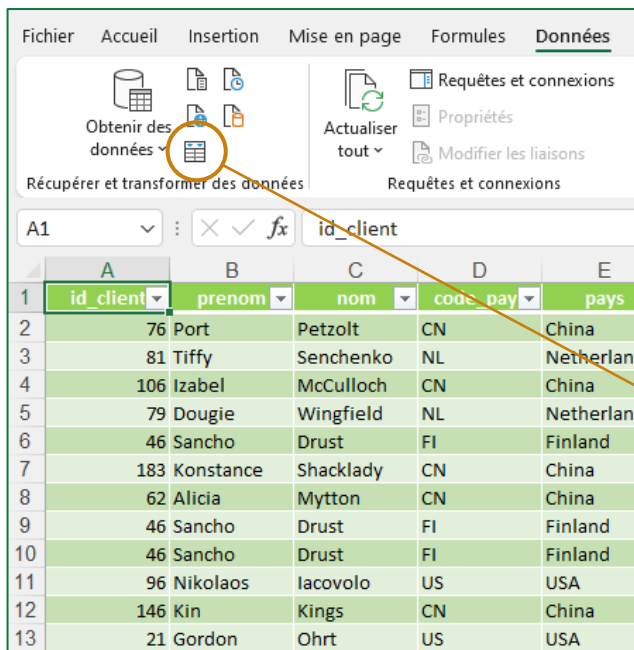
L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

Obtenir et Transformer des Données avec Power Query

Finissez-en avec les copier/coller et les RECHERCHEV ! Lorsque vous **Obtenez & transformez** des données dans **Excel** ou **Power BI** avec l'éditeur de requêtes **Power Query**, vous pouvez rechercher des sources de données, créer des connexions permanentes, puis mettre en forme ces données (par exemple, supprimer une colonne, modifier un type de données ou fusionner des tableaux) en fonction de vos besoins. Lorsque vous aurez mis en forme vos données, vous pourrez partager vos conclusions ou utiliser votre requête pour créer des **tableaux croisés dynamiques** dans Excel ou des **cartes** dans Power BI pour préparer vos rapports instantanés.

Si nous prenons les étapes dans l'ordre, la séquence est la suivante :



	A	B	C	D	E
1	id_client	prenom	nom	code_pay	pays
2	76	Port	Petzolt	CN	China
3	81	Tiffany	Senchenko	NL	Netherlan
4	106	Izabel	McCulloch	CN	China
5	79	Dougie	Wingfield	NL	Netherlan
6	46	Sancho	Drust	FI	Finland
7	183	Konstance	Shacklady	CN	China
8	62	Alicia	Mytton	CN	China
9	46	Sancho	Drust	FI	Finland
10	46	Sancho	Drust	FI	Finland
11	96	Nikolaos	Iacovolo	US	USA
12	146	Kin	Kings	CN	China
13	21	Gordon	Ohrt	US	USA

- **Connecter** : créez des connexions entre des données stockées dans le cloud, dans un service ou en local.
- **Transformer** : mettez en forme les données en fonction de vos besoins ; la source d'origine reste intacte.
- **Combiner** : créez un modèle de données à partir de plusieurs sources de données et obtenez un affichage unique des données avec **Power Pivot**.
- **Gérer** : une fois que vous avez terminé d'effectuer votre requête, vous pouvez l'enregistrer, la copier ou l'utiliser pour créer vos rapports.

Dès lors que vous vous connectez à des données, transformez-les ou combinez-les avec d'autres sources de données, **l'éditeur de requête Power Query** enregistre chaque étape et vous permet de la modifier si vous le souhaitez. L'éditeur de requête vous permet également d'annuler, de refaire, de changer l'ordre ou de modifier n'importe quelle étape... tout pour vous permettre de modifier l'affichage des données connectées comme vous le souhaitez.

Avec **Power Query**, vous pouvez créer des requêtes simples ou complexes selon vos besoins.

À mesure que vous ajoutez des étapes à une requête, l'éditeur de requête travaille en coulisses pour créer un ensemble d'instructions discrètes qui permettent d'exécuter vos commandes. Ces instructions sont créées dans le langage M, différent du langage VBA natif d'Excel.

Vous pouvez commencer une nouvelle requête à partir de l'onglet **Données > Obtenir des données >** dans **Excel** ou dans l'onglet **Accueil** dans **Power BI** puis sélectionnez une source de données externe ou bien cliquez sur le bouton **Nouvelle requête**, pour créer une requête directement à partir d'un tableau de votre classeur. Aller plus loin avec tableau-excel.com et en particulier avec l'article [Prise en main de Power Query](#).



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

Modélisez vos Données et Calculez avec Power Pivot

Ne multipliez plus les fonctions **RECHERCHEV()** ou **RECHERCHEX()** qui alourdissent vos classeurs et sollicitent le plus le processeur. Pour insérer une colonne d'une table dans une autre table, le modèle de données **Power Pivot** intégré directement à **Power BI** et en tant que complément dans **Excel** rend les fonctions de RECHERCHE obsolètes. Vous pourrez ainsi créer directement et sans formule une relation entre deux tables de données basée sur des valeurs correspondantes dans chaque table.

Power Pivot est un complément Excel disponible depuis la version 2010 et intégré à Power BI depuis sa sortie en 2015, qui permet d'effectuer une analyse de données puissante et de créer des modèles de données sophistiqués. **Power Pivot** vous permet de modifier de grandes quantités de données à partir de différentes sources, d'effectuer une analyse des informations rapidement et de partager ces analyses en toute simplicité.

Principales fonctionnalités de Power Pivot :

- Importer des millions de lignes de données de plusieurs sources : avec Power Pivot pour Excel, vous pouvez importer des millions de lignes de données de plusieurs sources de données dans un seul classeur Excel, créer des relations entre des données hétérogènes, créer des colonnes calculées et des mesures à l'aide de formules, générer des tableaux croisés dynamiques et des graphiques croisés dynamiques, puis analyser les données afin de pouvoir prendre des décisions pertinentes, sans besoin d'une assistance informatique
- Profitez de calculs et d'analyses rapides : tirez le meilleur parti des processeurs multicœurs et des gigaoctets de mémoire pour un traitement rapide des calculs grâce à des algorithmes de compression efficaces permettant de charger les plus grands jeux de données en mémoire.
- Prise en charge presque illimitée des sources de données : importez et combinez des données sources à partir de n'importe quel emplacement pour une analyse de données massive sur le bureau, notamment des bases de données relationnelles, des sources multidimensionnelles, des services Cloud, des flux de données, des fichiers Excel, des fichiers texte et des données à partir du Web.
- Fonctions DAX (Data Analysis Expressions) : DAX est un langage de formule qui étend les fonctionnalités de manipulation des données d'Excel afin d'offrir un regroupement, un calcul et une analyse plus sophistiqués et complexes. La syntaxe des formules DAX est très similaire à celle des formules Excel. Aller plus loin avec Microsoft : **Toutes les formules DAX**.

Aller plus loin avec tableau-excel.com et en particulier l'article **Power Pivot, Relations entre Tables**.



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

AGREGATION	FONCTION DAX	EQUIVALENT EXCEL	SYNTAXE	DESCRIPTION
	SUM	SOMME	=SUM(<colonne>)	<colonne> contient la colonne des valeurs à additionner. L'exemple suivant additionne tous les nombres contenus dans la colonne Amt de la table Sales : =SUM(Sales[Amt])
	SUMX	SOMME.SI SOMME.SI.ENS SOMMEPROD	=SUMX(<table>, <colonne>)	La fonction SUMX prend comme premier argument une table ou une expression filtrée qui retourne une partie de table. Le deuxième argument est une colonne qui contient les nombres dont vous voulez calculer la somme. L'exemple suivant commence par filtrer la table InternetSales sur l'expression ShippingTerritoryID=5 , puis retourne la somme de toutes les valeurs de la colonne Freight . : =SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])
	COUNT	NB	=COUNT(<colonne>)	Compte le nombre de cellules d'une colonne qui contiennent des nombres uniquement. L'argument <colonne> désigne la colonne des valeurs à compter.
	COUNTA	NBVAL	=COUNTA(<colonne>)	Compte le nombre de cellules non vides d'une colonne. L'argument <colonne> désigne la colonne des valeurs à compter.
	COUNTBLANK	NB.VIDE	=COUNTBLANK(<colonne>)	Compte le nombre de cellules vides d'une colonne. L'argument <colonne> désigne la colonne des valeurs à compter.
	COUNTROW	NB.VAL	=COUNTROWS([<table>])	Compte le nombre de lignes dans la table spécifiée ou dans une table définie par une expression.
	COUNTX	NB.SI.ENS	=COUNTX(<table>,<expression>)	La fonction COUNTX utilise 2 arguments. Le premier argument doit toujours être une table ou toute expression qui retourne une table. Le deuxième argument est la colonne ou l'expression recherchée par la fonction COUNTX. La fonction COUNTX compte uniquement les valeurs, les dates et les chaînes. Si la fonction ne trouve aucune ligne à compter, elle retourne une valeur vide. La formule suivante retourne le nombre total de lignes de la table Product contenant un prix catalogue. = COUNTX(Product,[ListPrice])
	DISTINCTCOUNT DISTINCT	NBVAL UNIQUE	=DISTINCTCOUNT(<column>) =DISTINCT(<column>)	DISTINCTCOUNT compte le nombre de valeurs distinctes dans une colonne. DISTINCT renvoie les valeurs uniques d'une colonne.
	MIN, MAX	MIN, MAX	=FONCTION((<column>) =FONCTION((<expr.1>, <expr.2>)	Retourne la plus petite ou la plus grande valeur d'une colonne ou entre deux expressions scalaires.
	MINA, MAXA	MINA, MAXA	=FONCTION(<column>)	Retourne la plus petite ou la plus grande valeur d'une colonne. L'argument <colonne> désigne la colonne des valeurs à évaluer.
	MINX, MAXX	MIN/MAX.SI.ENS	=FONCTION(<table>, < expr.>)	Retourne la plus petite ou la plus grande valeur qui résulte de l'évaluation d'une expression pour chaque ligne d'une table selon le même principe que les fonctions SUMX ou COUNTX .



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

FILTRAGE

FONCTION DAX	EQUIVALENT EXCEL	SYNTAXE	DESCRIPTION
ALL	X	=ALL([<table> <colonne>[, <colonne>[, <colonne>[,...]]]])	<p>La fonction ALL retourne toutes les lignes d'une table, ou toutes les valeurs d'une colonne, en ignorant les filtres qui ont été éventuellement appliqués. Cette fonction est utile pour effacer les filtres et créer des calculs sur toutes les lignes d'une table. L'argument <table> indique la table dont vous voulez effacer les filtres et les arguments <colonne>, la ou les colonnes spécifiques de la table dont vous voulez effacer les filtres.</p> <p>La fonction sans argument ALL() supprime tous les filtres partout.</p> <p>La fonction ALL(Table) supprime tous les filtres de la table spécifiée.</p> <p>La fonction ALL(Colonne) supprime tous les filtres de la seule colonne spécifiée.</p>
ALLEXCEPT	X	=ALLEXCEPT([<table> <colonne>[, <colonne>[, <colonne>[,...]]]])	Supprime tous les filtres de contexte de la table, à l'exception des filtres appliqués aux colonnes spécifiées. C'est une méthode simple à utiliser si vous souhaitez supprimer les filtres sur beaucoup de colonnes d'une table, mais pas sur toutes.
CALCULATE	SOMMEPROD BDSOMME	=CALCULATE(<expression>[, <filtre1> [, <filtre2> [, ...]]])	<p>La fonction CALCULATE évalue une expression dans un contexte de filtre modifié. Le premier argument <expression> est l'expression à évaluer qui peut être une somme ou toute autre opération d'agrégation. Les arguments suivants sont des expressions booléennes ou de table qui définissent des filtres ou des fonctions de modification de filtre. Par exemple, la mesure suivante pour la table Sales produit un résultat sous forme de chiffre d'affaires, mais uniquement pour les produits bleus :</p> <p>=CALCULATE(SUM(Sales[Sales Amount]), 'Product'[Color] = "Blue")</p>
FILTER	FILTRE	= FILTER(<table>,<filtre>)	<p>Retourne une table contenant uniquement les lignes filtrées.</p> <p>Le premier argument <table> représente la table ou l'expression de table à filtrer. Le deuxième argument <filtre> est une expression booléenne à évaluer pour chaque ligne de la table. Par exemple, [Amount] > 0 ou [Region] = "France". Vous pouvez utiliser FILTER pour réduire le nombre de lignes dans la table avec laquelle vous travaillez et utiliser uniquement des données spécifiques dans les calculs.</p> <p>La fonction FILTER n'étant pas indépendante, elle doit être utilisée en tant que fonction incorporée dans d'autres fonctions nécessitant une table comme argument.</p>
LOOKUPVALUE	RECHERCHEV	=LOOKUPVALUE(<result_columnName>, <search_columnName>, <search_value> [, <search2_columnName>, <search2_value>]... [, <alternateResult>])	<p><result_columnName> Nom d'une colonne existante qui contient la valeur que vous voulez retourner. Il ne peut pas s'agir d'une expression.</p> <p><search_columnName> Nom d'une colonne existante. Elle peut se trouver dans la même table que <result_columnName> ou dans une table associée. Il ne peut pas s'agir d'une expression.</p> <p><search_value> Valeur à rechercher dans <search_columnName>.</p> <p><alternateResult> (Facultatif)</p>



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

DATE

FONCTION DAX	EQUIVALENT EXCEL	SYNTAXE	DESCRIPTION
CALENDAR	X	=CALENDAR(<start_date>, <end_date>)	<p>Retourne une table avec une seule colonne nommée « Date » qui contient un ensemble de dates contigües. La plage de dates est la période allant de la date de début spécifiée à la date de fin spécifiée, incluses.</p> <p>La formule suivante retourne une table avec les dates comprises entre le 1er janvier 2021 et le 31 décembre 2022 :</p> <p>= CALENDAR (DATE (2021, 1, 1), DATE (2022, 12, 31))</p> <p>Pour un modèle de données qui comprend les données de ventes réelles et les prévisions de ventes, l'expression suivante retourne la table de dates qui couvre la plage de dates de ces deux tables :</p> <p>= CALENDAR (MINX (Sales, [Date]), MAXX (Forecast, [Date]))</p>
CALENDARAUTO	X	=CALENDARAUTO([fiscal_year_end_m onth])	<p>Retourne une table avec une seule colonne nommée « Date » qui contient un ensemble de dates contigües. La plage de dates est calculée automatiquement en fonction des données du modèle existant.</p> <p>Dans cet exemple, les informations MinDate et MaxDate dans le modèle de données sont le 1er juillet 2010 et le 30 juin 2011.</p> <p>CALENDARAUTO() retourne toutes les dates comprises entre le 1er janvier 2010 et le 31 décembre 2011 et CALENDARAUTO(3) retourne toutes les dates comprises entre le 1er mars 2010 et le 31 mars 2012.</p>
DATE	DATE	=DATE(<year>, <month>, <day>)	Retourne la date spécifiée au format datetime.
DATEDIFF	DATEDIF	= DATEDIFF(<start_date>, <end_date>, <interval>)	Retourne le nombre de limites d'intervalle franchies entre deux dates. La valeur de l'argument <interval> peut-être : SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER ou YEAR.
TODAY NOW	AUJOURDHUI MAINTENANT	=TODAY() =NOW()	<p>Retourne la date actuelle du système.</p> <p>Retourne la date et l'heure actuelles du système.</p>
DAY MONTH QUARTER YEAR	JOUR MOIS X ANNEE	=DAY(<date>) =MONTH(<date>) =QUARTER(<date>) =YEAR(<date>)	Retourne le jour, le mois, le trimestre ou l'année d'une date sous forme d'un entier entre compris entre 1 et 31 pour le jour, ou entre 1 et 12 pour le mois, ou entre 1 et 4 pour le trimestre, ou entre 1900-9999 pour l'année.

TOUTES LES FONCTIONS D'INTELLIGENCE TEMPORELLE (TIME INTELLIGENCE)



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

LOGIQUE

FONCTION DAX	EQUIVALENT EXCEL	SYNTAXE	DESCRIPTION
IF	SI	=IF(<logical_test>, <value_if_true>[, <value_if_false>])	<logical_test> Valeur ou expression dont le résultat de l'évaluation peut être TRUE (vrai) ou FALSE (faux). <value_if_true> Valeur retournée si le test logique est TRUE (vrai). <value_if_false> (Facultatif) Valeur retournée si le test logique est FALSE (faux). S'il n'est pas spécifié, la valeur BLANK est retournée, c'est à dire un vide. La logique de la fonction est la même que dans Excel.
AND	ET	=AND(<logical1>,<logical2>)	Vérifie si les deux arguments ont la valeur TRUE, puis retourne TRUE si tel est le cas. Sinon, retourne FALSE. Retourne true ou false en fonction de la combinaison des valeurs testées. La logique de la fonction est la même que dans Excel.
OR	OU	=OR(<logical1>,<logical2>)	Vérifie si l'un des arguments a la valeur TRUE pour retourner TRUE. La fonction retourne FALSE si les deux arguments ont la valeur FALSE. Valeur booléenne. La valeur est TRUE si l'un des deux arguments a la valeur TRUE ; la valeur est FALSE si les deux arguments ont la valeur FALSE. La logique de la fonction est la même que dans Excel.
TRUE	VRAI	=TRUE()	Retourne la valeur logique TRUE. La logique de la fonction est la même que dans Excel.
FALSE	FAUX	=FALSE()	Retourne la valeur logique FALSE. La logique de la fonction est la même que dans Excel.
NOT	NON	=NOT(<logical>)	Change FALSE en TRUE, ou TRUE en FALSE. <logical> Valeur ou expression qui peut prendre la valeur TRUE ou FALSE. La logique de la fonction est la même que dans Excel.
IFERROR	SIERREUR	=IFERROR(<value>, <value_if_error>)	Évalue une expression et retourne une valeur spécifiée si l'expression retourne une erreur. Sinon, retourne la valeur de l'expression elle-même. Les arguments (<value> et <value_if_error> représente une Valeur ou expression quelconque. La logique de la fonction est la même que dans Excel.



L'ESSENTIEL EXCEL/POWER BI¹ & DAX²

¹Business Intelligence & ²Data Analysis Expression

TEXTE

FONCTION DAX	EQUIVALENT EXCEL	SYNTAXE	DESCRIPTION
CONCATENATE	CONCAT	=CONCATENATE(<text1>, <text2>)	La fonction CONCATENATE concatène 2 chaînes de texte. Les éléments concaténés peuvent être du texte, des nombres, des valeurs booléennes représentées sous forme de texte ou une combinaison de ces éléments. Vous pouvez également utiliser une référence de colonne si la colonne contient les valeurs appropriées. Dans DAX, la fonction CONCATENATE accepte 2 arguments uniquement, alors qu'elle en accepte jusqu'à 255 dans Excel. Si vous devez concaténer plusieurs colonnes, vous pouvez créer une série de calculs ou, mieux, utiliser l'opérateur de concaténation (&) pour les joindre toutes dans une expression plus simple. Si vous souhaitez utiliser des chaînes de texte directement (plutôt qu'une référence de colonne), vous devez placer chaque chaîne entre guillemets doubles.
FIND	TROUVE	=FIND(<find_text>, <within_text>[, <start_num>][, <NotFoundValue>])	Retourne la position de départ d'une chaîne de texte à l'intérieur d'une autre chaîne de texte. La fonction FIND respecte la casse. La logique de la fonction est la même que dans Excel.
SEARCH	CHERCHE	=SEARCH(<find_text>, <within_text>[, <start_num>][, <NotFoundValue>])	Retourne le numéro du caractère où apparaît pour la première fois un caractère ou une chaîne de texte spécifique selon une lecture de gauche à droite. La recherche ne respecte pas la casse mais respecte les accents. La logique de la fonction est la même que dans Excel.
FORMAT	TEXTE	=FORMAT(<value>, <format_string>)	Convertit une valeur en texte au format spécifié. La logique de la fonction est la même que dans Excel.
LEFT MID RIGHT	GAUCHE STXT DROITE	=LEFT(<text>, <num_chars>) =MID(<text>, <start_num>, <num_chars>) =RIGHT(<text>, <num_chars>)	Retourne une chaîne de caractères à partir de la gauche, du milieu ou de la droite d'une chaîne de texte, en fonction d'une position de départ et d'une longueur. La logique de ces fonctions est la même que dans Excel.
LOWER UPPER	MINUSCULE MAJUSCULE	=LOWER(<text>) =UPPER(<text>)	Retourne la chaîne de texte <text> en minuscule ou en majuscule. La logique de la fonction est la même que dans Excel.
VALUE	CNUM	= VALUE(<text>)	Convertit en nombre une chaîne de texte qui représente un nombre. La logique de la fonction est la même que dans Excel.

TOUTES LES FONCTIONS DAX