

# Riferimento a Data Analysis Expressions (DAX)

Data Analysis Expressions (DAX) è una libreria di funzioni e operatori che è possibile combinare per creare formule ed espressioni nei modelli di dati per Power BI, Analysis Services e Power Pivot in Excel.

## Funzioni

### RIFERIMENTO

[Funzioni DAX](#)

[Istruzioni DAX](#)

### NOVITÀ

[Nuove funzioni DAX](#)

## Informazioni su DAX

### PANORAMICA

[Panoramica di DAX](#)

### VIDEO

[Video](#)

### FORMAZIONE

[Usare DAX in Power BI Desktop](#)

## Informazioni generali

### CONCETTO

[Glossario di DAX](#)

[Operatori DAX](#)

[Query DAX](#)

[Sintassi DAX](#)

# Panoramica di DAX

Articolo • 20/10/2023

DAX (Data Analysis Expressions) è un linguaggio di espressioni delle formule usato in Analysis Services, Power BI e Power Pivot in Excel. Le formule DAX includono funzioni, operatori e valori per eseguire query e calcoli avanzati sui dati presenti in tabelle e colonne correlate nei modelli di dati tabulari.

Questo articolo offre solo un'introduzione di base ai concetti più importanti del linguaggio DAX, descrivendone le caratteristiche generali applicabili a tutti i prodotti da cui viene usato. Alcune funzionalità potrebbero tuttavia non essere valide per determinati prodotti o casi d'uso. Per un'implementazione di DAX specifica, vedere la documentazione del prodotto.

## Calcoli

Le formule DAX vengono usate in misure, colonne calcolate, tabelle calcolate e sicurezza a livello di riga.

## Misure

Le misure sono formule di calcolo dinamiche i cui risultati cambiano a seconda del contesto. Le misure vengono usate per la creazione di report che supportano la combinazione e il filtraggio dei dati dei modelli usando più attributi, ad esempio un report di Power BI oppure una tabella pivot o un grafico pivot di Excel. Le misure vengono create tramite la barra della formula DAX in Progettazione modelli.

Una formula in una misura può usare le funzioni di aggregazione standard create automaticamente tramite la funzionalità Somma automatica, ad esempio COUNT o SUM, oppure può essere definita dall'utente tramite la barra della formula DAX. Le misure con nome possono essere passate come argomento ad altre misure.

Quando si definisce una formula per una misura nella barra della formula, tramite una descrizione comando viene visualizzata un'anteprima dei possibili risultati per il totale nel contesto corrente. Generalmente i risultati non vengono restituiti immediatamente in nessuna posizione. Il motivo per cui non è possibile visualizzare i risultati (filtrati) del calcolo immediatamente è dovuto al fatto che il risultato di una misura non può essere determinato senza contesto. La valutazione di una misura richiede la presenza di un'applicazione client di creazione di report in grado di fornire il contesto necessario per recuperare i dati relativi a ogni cella e valutare quindi l'espressione per ogni cella. Il

client può essere una tabella pivot o un grafico pivot di Excel, un report di Power BI o un'espressione di tabella in una query DAX in SQL Server Management Studio (SSMS).

Indipendentemente dal client, viene eseguita una query separata per ogni cella nei risultati. In altre parole, ogni combinazione di intestazioni di riga e colonna in una tabella pivot oppure ogni selezione di filtri dei dati e filtri in un report di Power BI genera un subset di dati diverso in base al quale viene calcolata la misura. Si consideri, ad esempio, questa formula di misura molto semplice:

DAX

```
Total Sales = SUM([Sales Amount])
```

Quando un utente inserisce la misura TotalSales in un report e quindi inserisce la colonna Product Category da una tabella Product nei filtri, la somma di Sales Amount viene calcolata e visualizzata per ogni categoria di prodotto.

A differenza delle colonne calcolate, la sintassi per una misura include il nome della misura che precede la formula. Nell'esempio sopra riportato, il nome **Total Sales** viene visualizzato prima della formula. Dopo aver creato una misura, il nome e la relativa definizione vengono visualizzati nell'elenco dei campi dell'applicazione client per la creazione di report e, a seconda delle prospettive e dei ruoli, è disponibile per tutti gli utenti del modello.

Per altre informazioni, vedere:

[Misure in Power BI Desktop](#)

[Misure in Analysis Services](#)

[Misure in Power Pivot](#) ↗

## Colonne calcolate

Una colonna calcolata è una colonna che viene aggiunta a una tabella esistente in Progettazione modelli e per la quale viene creata una formula DAX che ne definisce i valori. Quando una colonna calcolata contiene una formula DAX valida, i valori vengono calcolati per ogni riga non appena si immette la formula. I valori vengono quindi archiviati nel modello di dati in memoria. Ad esempio, in una tabella Date, quando la formula viene immessa nella barra della formula:

DAX

```
= [Calendar Year] & " Q" & [Calendar Quarter]
```

Viene calcolato un valore per ogni riga della tabella prendendo i valori dalla colonna Calendar Year (nella stessa tabella Date), aggiungendo uno spazio e la lettera maiuscola Q, quindi aggiungendo i valori dalla colonna Calendar Quarter (nella stessa tabella Date). Il risultato per ogni riga della colonna calcolata viene calcolato immediatamente e visualizzato, ad esempio, come **2017 Q1**. I valori delle colonne vengono ricalcolati solo se la tabella, o una tabella correlata, viene elaborata (aggiornata) oppure se il modello viene scaricato dalla memoria e quindi ricaricato, ad esempio quando si chiude e si riapre un file di Power BI Desktop.

Per altre informazioni, vedere:

[Colonne calcolate in Power BI Desktop](#)

[Colonne calcolate in Analysis Services](#)

[Colonne calcolate in Power Pivot](#) ↗

## Tabelle calcolate

Una tabella calcolata è un oggetto calcolato, basato su un'espressione di formula e derivato da tutte le altre tabelle nello stesso modello o da alcune di esse. Invece di eseguire query e caricare i valori nelle colonne della nuova tabella da un'origine dati, una formula DAX definisce i valori della tabella.

Le tabelle calcolate possono essere utili in una dimensione con ruoli multipli. Un esempio è offerto dalla tabella Date, come OrderDate, ShipDate o DueDate, a seconda della relazione di chiave esterna. Creando una tabella calcolata per DataSpedizione in modo esplicito, è possibile ottenere una tabella autonoma disponibile per le query, utilizzabile in modo completo come qualsiasi altra tabella. Le tabelle calcolate sono utili anche per la configurazione di un set di righe filtrato o di un subset o superset di colonne di altre tabelle esistenti. Questo consente di mantenere la tabella originale intatta e di creare allo stesso tempo varianti della tabella per supportare scenari specifici.

Le tabelle calcolate supportano le relazioni con altre tabelle. Le colonne nella tabella calcolata contengono i tipi di dati, la formattazione e possono appartenere a una categoria di dati. È possibile assegnare un nome alle tabelle e rivelarle o nasconderle esattamente come qualsiasi altra tabella. Le tabelle calcolate vengono ricalcolate se una delle tabelle da cui vengono estratti i dati viene aggiornata.

Per altre informazioni, vedere:

[Tabelle calcolate in Power BI Desktop](#)

[Tabelle calcolate in Analysis Services](#)

## Sicurezza a livello di riga

Con la sicurezza a livello di riga, una formula DAX deve restituire una condizione booleana TRUE/FALSE, definendo le righe che possono essere restituite dai risultati di una query in base ai membri di un ruolo specifico. Si prenda, ad esempio, la tabella Customers con la seguente formula DAX per i membri del ruolo Sales:

DAX

```
= Customers[Country] = "USA"
```

I membri del ruolo Sales saranno in grado di visualizzare solo i dati per i clienti negli Stati Uniti e le aggregazioni, come SUM, verranno restituite solo per i clienti negli Stati Uniti. La sicurezza a livello di riga non è disponibile in Power Pivot in Excel.

Quando si definisce la sicurezza a livello di riga usando una formula DAX, si crea un set di righe consentito. Questo non significa che l'accesso alle altre righe viene negato, ma semplicemente che tali righe non vengono restituite come parte del set di righe consentito. Altri ruoli possono consentire l'accesso alle righe escluse dalla formula DAX. Se un utente è membro di un altro ruolo e la sicurezza a livello di riga di tale ruolo consente l'accesso a quel particolare set di righe, l'utente può visualizzare i dati per la riga.

Le formule per la sicurezza a livello di riga si applicano alle righe specificate e a quelle correlate. Quando una tabella dispone di più relazioni, tramite i filtri viene applicata la sicurezza alla relazione che è attiva. Le formule per la sicurezza a livello di riga vengono intersecate con altre formule definite per le tabelle correlate.

Per altre informazioni, vedere:

[Sicurezza a livello di riga con Power BI](#)

[Ruoli in Analysis Services](#)

## Query

Le query DAX possono essere create ed eseguite in SQL Server Management Studio (SSMS) e in strumenti open source come DAX Studio ([daxstudio.org](http://daxstudio.org)). Diversamente dalle formule di calcolo DAX, che possono essere create solo nei modelli di dati tabulari, è possibile eseguire query DAX anche su modelli multidimensionali di Analysis Services. Le query DAX sono spesso più facili da scrivere e più efficienti rispetto alle query MDX (Multidimensional Data Expressions).

Una query DAX è costituita da un'istruzione, simile a un'istruzione SELECT in T-SQL. Il tipo di query DAX più semplice è un'istruzione *evaluate*. Ad esempio,

## DAX

```
EVALUATE  
    ( FILTER ( 'DimProduct', [SafetyStockLevel] < 200 ) )  
ORDER BY [EnglishProductName] ASC
```

Restituisce nei risultati una tabella che elenca solo i prodotti con un valore di SafetyStockLevel inferiore a 200, in ordine crescente in base a EnglishProductName.

È possibile creare misure come parte della query. In tal caso, le misure sono disponibili solo per la durata della query. Per altre informazioni, vedere [Query DAX](#).

## Formule

Le formule DAX sono fondamentali per la creazione di calcoli in misure e colonne calcolate e per la protezione dei dati usando la sicurezza a livello di riga. Per creare formule per misure e colonne calcolate, usare la barra della formula nella parte superiore della finestra di progettazione del modello o l'editor DAX. Per creare formule per la sicurezza a livello di riga, usare la finestra di dialogo Gestione ruoli o Gestisci ruoli. Lo scopo delle informazioni fornite in questa sezione è quello di facilitare la comprensione delle nozioni fondamentali relative alle formule DAX.

## Nozioni fondamentali sulle formule

Le formule DAX possono essere molto semplici o piuttosto complesse. Nella tabella seguente sono riportati alcuni esempi di formule semplici che potrebbero essere utilizzate in una colonna calcolata.

Formula	Definizione
= TODAY()	Inserisce la data odierna in ogni riga di una colonna calcolata.
= 3	Inserisce il valore 3 in ogni riga di una colonna calcolata.
= [Column1] + [Column2]	Consente di sommare i valori nella stessa riga di [Column1] e [Column2] e di inserire i risultati nella colonna calcolata della stessa riga.

Per le formule create, sia semplici che complesse, è possibile utilizzare i passaggi seguenti durante la compilazione di una formula:

1. Ogni formula deve iniziare con un segno di uguale (=).
2. È possibile digitare o selezionare il nome di una funzione oppure digitare un'espressione.

3. Quando si inizia a digitare le prime lettere della funzione o del nome desiderato, la funzionalità Completamento automatico visualizza un elenco di funzioni, tabelle e colonne disponibili. Premere TAB per aggiungere alla formula un elemento dell'elenco Completamento automatico.

È anche possibile fare clic sul pulsante **Fx** per visualizzare un elenco di funzioni disponibili. Per selezionare una funzione dall'elenco a discesa, utilizzare i tasti di direzione per evidenziare l'elemento, quindi fare clic su **OK** per aggiungere la funzione alla formula.

4. Fornire gli argomenti per la funzione selezionandoli da un elenco a discesa in cui sono incluse le possibili tabelle e colonne oppure digitando i valori.
5. Verificare la presenza di errori di sintassi: assicurarsi che tutte le parentesi siano chiuse e che i riferimenti a colonne, tabelle e valori siano corretti.
6. Premere INVIO per accettare la formula.

#### ① Nota

In una colonna calcolata, non appena si immette la formula e questa viene convalidata, vengono inseriti automaticamente i valori. In una misura, la relativa definizione viene salvata con la tabella premendo INVIO. Se una formula non è valida, viene visualizzato un errore.

In questo esempio verrà esaminata una formula in una misura denominata **Days in Current Quarter**:

DAX

```
Days in Current Quarter = COUNTROWS( DATESBETWEEN( 'Date'[Date],  
STARTOFQUARTER( LASTDATE('Date'[Date])), ENDOFQUARTER('Date'[Date])))
```

Questa misura viene utilizzata per creare un rapporto di confronto tra un periodo incompleto e il periodo precedente. È necessario che nella formula la proporzione del periodo trascorso venga presa in considerazione e confrontata con la stessa proporzione del periodo precedente. In questo caso, [Days Current Quarter to Date]/[Days in Current Quarter] fornisce la proporzione trascorsa nel periodo corrente.

Questa formula contiene i seguenti elementi:

<b>Elemento della formula</b>	<b>Descrizione</b>
<code>Days in Current Quarter</code>	Nome della misura.
<code>=</code>	Il segno di uguale (=) inizia la formula.
<code>COUNTROWS</code>	<code>COUNTROWS</code> conta il numero di righe nella tabella Date.
<code>()</code>	Le parentesi di apertura e chiusura specificano gli argomenti.
<code>DATESBETWEEN</code>	La funzione DATESBETWEEN restituisce le date tra l'ultima data per ogni valore nella colonna Date della tabella Date.
<code>'Date'</code>	Specifica la tabella Date. Le tabelle sono racchiuse tra virgolette singole.
<code>[Date]</code>	Specifica la colonna Date nella tabella Date. Le colonne sono racchiuse tra parentesi.
<code>,</code>	
<code>STARTOFQUARTER</code>	La funzione STARTOFQUARTER restituisce la data dell'inizio del trimestre.
<code>LASTDATE</code>	La funzione LASTDATE restituisce l'ultima data del trimestre.
<code>'Date'</code>	Specifica la tabella Date.
<code>[Date]</code>	Specifica la colonna Date nella tabella Date.
<code>,</code>	
<code>ENDOFQUARTER</code>	Funzione ENDOFQUARTER
<code>'Date'</code>	Specifica la tabella Date.
<code>[Date]</code>	Specifica la colonna Date nella tabella Date.

## Utilizzo di Completamento automatico formule

che consente di immettere una sintassi della formula valida visualizzando le opzioni per ogni elemento della formula.

- È possibile utilizzare Completamento automatico formule in una formula esistente con funzioni nidificate. Il testo immediatamente prima del punto di inserimento viene utilizzato per visualizzare i valori nell'elenco a discesa mentre tutto il testo dopo tale punto rimane invariato.

- In Completamento automatico non viene aggiunta la parentesi di chiusura delle funzioni né vengono inserite automaticamente le parentesi corrispondenti. È necessario assicurarsi che ogni funzione sia sintatticamente corretta, altrimenti non sarà possibile salvare o utilizzare la formula.

## Uso di più funzioni in una formula

È possibile nidificare funzioni, pertanto è possibile utilizzare i risultati di una funzione come argomento di un'altra funzione. È possibile annidare fino a 64 livelli di funzioni nelle colonne calcolate. La nidificazione può tuttavia rendere più complessa la creazione di formule o la risoluzione dei problemi relativi alle formule. Numerose funzioni sono progettate per essere utilizzate esclusivamente come funzioni nidificate. Tramite queste funzioni viene restituita una tabella, che non può essere salvata direttamente come risultato, ma deve essere fornita come input per una funzione di tabella. Ad esempio, per le funzioni SUMX, AVERAGEX e MINX è richiesta una tabella come primo argomento.

## Funzioni

Una funzione è una formula denominata all'interno di un'espressione. La maggior parte delle funzioni ha come input argomenti obbligatori e facoltativi, noti anche come parametri. Quando viene eseguita la funzione, viene restituito un valore. DAX include funzioni che consentono di eseguire calcoli con date e ore, creare valori condizionali, utilizzare stringhe ed eseguire ricerche basate su relazioni, oltre a offrire la possibilità di eseguire iterazioni su una tabella per l'esecuzione di calcoli ricorsivi. Se si ha dimestichezza con le formule di Excel, molte di queste funzioni appariranno molto simili, tuttavia, le formule DAX sono diverse nelle seguenti modalità importanti:

- Una funzione DAX fa sempre riferimento a una colonna completa o una tabella. Se si desidera utilizzare solo particolari valori di una tabella o colonna, è possibile aggiungere filtri alla formula.
- Se è necessario personalizzare i calcoli per ogni singola riga, in DAX sono disponibili funzioni che consentono di utilizzare il valore della riga corrente o un valore correlato come un tipo di parametro, per eseguire i calcoli che variano in base al contesto. Per comprendere la logica alla base di queste funzioni, vedere la sezione [Contesto](#) in questo articolo.
- In DAX sono incluse molte funzioni mediante le quali viene restituita una tabella, anziché un valore. La tabella non viene visualizzata in uno strumento client di creazione report, ma viene utilizzata per fornire input ad altre funzioni. Ad

esempio, è possibile recuperare una tabella e contare i valori distinti in essa contenuti o calcolare somme dinamiche nelle tabelle o colonne filtrate.

- Nelle funzioni DAX sono incluse numerose funzioni di *Business Intelligence per le gerarchie temporali*. Queste funzioni consentono di definire o selezionare intervalli di date e di eseguire calcoli dinamici in base a tali date o intervalli. Ad esempio, è possibile confrontare somme in periodi paralleli.

## Funzione di aggregazione

Le funzioni di aggregazione calcolano un valore (scalare) come il numero, la somma, la media, il minimo o il massimo per tutte le righe di una colonna o di una tabella, come definito dall'espressione. Per altre informazioni, vedere [Funzioni di aggregazione](#).

## Funzioni di data e ora

Le funzioni di data e ora in DAX sono molto simili alle funzioni di data e ora di Microsoft Excel. Le funzioni DAX, tuttavia, sono basate su un tipo di dati **datetime** a partire dal 1° marzo 1900. Per altre informazioni, vedere [Funzioni di data e ora](#).

## Funzioni di filtro

Le funzioni di filtro in DAX restituiscono specifici tipi di dati, cercano valori nelle tabelle correlate e filtrano in base a valori correlati. Le funzioni di ricerca utilizzano tabelle e relazioni, come un database. Le funzioni di filtro consentono di modificare il contesto dei dati per creare calcoli dinamici. Per altre informazioni, vedere [Funzioni di filtro](#).

## Funzioni finanziarie

Le funzioni finanziarie in DAX vengono usate in formule che eseguono calcoli finanziari, ad esempio il valore attuale netto o il tasso di rendimento. Queste funzioni sono simili alle funzioni finanziarie usate in Microsoft Excel. Per altre informazioni, vedere [Funzioni finanziarie](#).

## Funzioni informative

Una funzione informativa analizza la cella o la riga fornita come argomento e indica se il valore corrisponde al tipo previsto. La funzione ISERROR, ad esempio, restituisce TRUE se il valore a cui si fa riferimento contiene un errore. Per altre informazioni, vedere [Funzioni informative](#).

## Funzioni logiche

Le funzioni logiche eseguono operazioni su un'espressione per restituire informazioni sui valori nell'espressione. La funzione TRUE, ad esempio, consente di sapere se un'espressione che si sta valutando restituirà un valore TRUE. Per altre informazioni, vedere [Funzioni logiche](#).

## Funzioni matematiche e trigonometriche

Le funzioni matematiche in DAX sono molto simili alle funzioni matematiche e trigonometriche di Excel. Esistono tuttavia alcune piccole differenze nei tipi di dati numerici utilizzati dalle funzioni DAX. Per altre informazioni, vedere [Funzioni matematiche e trigonometriche](#).

## Altre funzioni

Queste funzioni eseguono azioni particolari che non possono essere definite in base a nessuna delle categorie a cui appartiene la maggior parte delle altre funzioni. Per altre informazioni, vedere [Altre funzioni](#).

## Funzioni di relazione

Le funzioni di relazione in DAX consentono di restituire valori da un'altra tabella correlata, specificare una particolare relazione da usare in un'espressione e specificare la direzione del filtro incrociato. Per altre informazioni, vedere [Funzioni di relazione](#).

## Funzioni statistiche

Le funzioni statistiche calcolano i valori correlati a distribuzioni statistiche e probabilità, ad esempio deviazione standard e numero di permutazioni. Per altre informazioni, vedere [Funzioni statistiche](#).

## Funzioni di testo

Le funzioni di testo in DAX sono molto simili alle funzioni corrispondenti in Excel. È possibile restituire parte di una stringa, cercare testo all'interno di una stringa o concatenare valori stringa. In DAX sono inoltre disponibili funzioni per il controllo dei formati per date, ore e numeri. Per altre informazioni, vedere [Funzioni di testo](#).

## Funzioni di Business Intelligence per le gerarchie temporali

Le funzioni di Business Intelligence per le gerarchie temporali disponibili in DAX consentono di creare calcoli in cui vengono utilizzate informazioni predefinite su calendari e date. Tramite gli intervalli di ore e date in combinazione con aggregazioni o calcoli è possibile compilare confronti significativi tra periodi di tempo paragonabili relativamente a vendite, scorte e così via. Per altre informazioni, vedere [Funzioni di Business Intelligence per le gerarchie temporali \(DAX\)](#).

## Funzioni di manipolazione delle tabelle

Queste funzioni restituiscono una tabella o modificano le tabelle esistenti. Usando ADDCOLUMNS, ad esempio, è possibile aggiungere colonne calcolate a una tabella specificata oppure è possibile restituire una tabella di riepilogo per un set di gruppi con la funzione SUMMARIZECOLUMNNS. Per altre informazioni, vedere [Funzioni di manipolazione delle tabelle](#).

## Variabili

È possibile creare variabili all'interno di un'espressione usando [VAR](#). VAR tecnicamente non è una funzione, ma una parola chiave che consente di archiviare il risultato di un'espressione come variabile denominata, che può quindi essere passata come argomento ad altre espressioni di misura. Ad esempio:

```
DAX

VAR
    TotalQty = SUM ( Sales[Quantity] )

Return

    IF (
        TotalQty > 1000,
        TotalQty * 0.95,
        TotalQty * 1.25
    )
```

In questo esempio, è possibile passare TotalQty come variabile denominata ad altre espressioni. Le variabili possono essere costituite da qualsiasi tipo di dati scalare, incluse tabelle. L'uso di variabili nelle formule DAX può essere incredibilmente potente.

# Tipo di dati

In un modello è possibile importare dati provenienti da numerose origini dati diverse in grado di supportare tipi di dati differenti. Quando si importano dati in un modello, i dati vengono convertiti in uno dei tipi di dati dei modelli tabulari. Quando i dati del modello vengono usati in un calcolo, i dati vengono quindi convertiti in un tipo di dati DAX per la durata e l'output del calcolo. Quando si crea una formula di DAX, i termini utilizzati nella formula determinano automaticamente il tipo di dati del valore restituito.

In DAX sono supportati i tipi di dati seguenti:

<b>Tipo di dati nel modello</b>	<b>Tipi di dati in DAX</b>	<b>Descrizione</b>
Numero intero	Valore intero a 64 bit (otto byte) <sup>1, 2</sup>	Numeri senza cifre decimali. I numeri interi possono essere positivi o negativi ma devono essere numeri interi compresi tra -9.223.372.036.854.775.808 (-2^63) e 9.223.372.036.854.775.807 (2^63-1).
Numero decimale	Numero reale a 64 bit (otto byte) <sup>1, 2</sup>	I numeri reali sono numeri che possono avere cifre decimali e coprono un ampio intervallo di valori:  Valori negativi compresi tra -1,79E +308 e -2,23E -308  Zero  Valori positivi compresi tra 2,23E -308 e 1,79E + 308  Tuttavia, il numero di cifre significative è limitato a 17 cifre decimali.
Boolean	Boolean	Valore True o False.
Testo	Stringa	Stringa di dati di tipo carattere Unicode. Può trattarsi di stringhe, numeri o date rappresentati in un formato di testo.
Data	Data/ora	Date e ore in una rappresentazione di data e ora valida.  Le date valide sono tutte le date successive al 1 marzo del 1900.
Valuta	Valuta	Il tipo di dati currency consente valori compresi tra -922.337.203.685.477,5808 e 922.337.203.685.477,5807 con quattro cifre decimali di precisione fissa.
N/D	Blank	Un tipo di dati blank in DAX rappresenta e sostituisce i valori Null di SQL. È possibile creare un tipo di dati blank utilizzando la

<b>Tipo di dati nel modello</b>	<b>Tipi di dati in DAX</b>	<b>Descrizione</b>
		funzione BLANK, nonché verificare la presenza di tipi di dati blank utilizzando la funzione logica ISBLANK.

I modelli di dati tabulari includono anche il tipo di dati *Table* come input o output per molte funzioni DAX. Ad esempio, la funzione FILTER consente di utilizzare una tabella come input e di restituire un'altra tabella in cui sono contenute solo le righe che soddisfano le condizioni di filtro. Combinando le funzioni delle tabelle con le funzioni di aggregazione, è possibile eseguire calcoli complessi in set di dati definiti in modo dinamico.

Anche se i tipi di dati vengono in genere impostati automaticamente, è importante capire i tipi di dati e il modo si applicano, in particolare, alle formule DAX. Gli errori in formule o i risultati imprevisti, ad esempio, sono spesso causati dall'utilizzo di un particolare operatore che non può essere utilizzato con un tipo di dati specificato in un argomento. La formula `= 1 & 2` restituisce, ad esempio, come risultato la stringa 12. La formula `= "1" + "2"` restituisce tuttavia come risultato il valore intero 3.

## Contesto

Il *contesto* è un concetto importante da comprendere quando si creano formule DAX. Rappresenta l'elemento che consente di eseguire analisi dinamiche, dal momento che i risultati di una formula vengono modificati per riflettere la riga o la selezione della cella corrente, nonché anche eventuali dati correlati. La comprensione del contesto e l'utilizzo efficace di quest'ultimo sono fondamentali per la realizzazioni di analisi dinamiche ad elevate prestazioni e per la risoluzione dei problemi riscontrati nelle formule.

Le formule nei modelli tabulari possono essere valutate in un contesto diverso, in base ad altri elementi di progettazione quali:

- Filtri applicati in una tabella pivot o in un report
- Filtri definiti all'interno di una formula
- Relazioni specificate tramite funzioni speciali all'interno di una formula

Esistono diversi tipi di contesto: *contesto di riga*, *contesto di query* e *contesto di filtro*.

## Contesto di riga

Il *contesto di riga* può essere considerato come "la riga corrente". Se si crea una formula in una colonna calcolata, nel contesto di riga per tale formula sono inclusi i valori di

tutte le colonne presenti nella riga corrente. Se la tabella è correlata a un'altra tabella, il contenuto include anche tutti i valori dell'altra tabella che sono correlati alla riga corrente.

Si supponga ad esempio di creare una colonna calcolata, `= [Freight] + [Tax]`, in cui vengono sommati i valori di due colonne, Freight e Tax, della stessa tabella. Tramite questa formula si ottengono automaticamente solo i valori dalla riga corrente delle colonne specificate.

Il contesto di riga segue inoltre qualsiasi relazione definita tra le tabelle, incluse le relazioni definite all'interno di una colonna calcolata tramite formule DAX, per determinare quali righe nelle tabelle correlate sono associate alla riga corrente.

Nella formula seguente viene ad esempio utilizzata la funzione RELATED per recuperare un valore relativo all'imposta da una tabella correlata, in base all'area in cui è stato eseguito l'ordine. Il valore dell'imposta viene determinato utilizzando il valore per regione nella tabella corrente, effettuando la ricerca della regione nella tabella correlata e quindi ottenendo l'aliquota di imposta per tale regione dalla tabella correlata.

#### DAX

```
= [Freight] + RELATED('Region'[TaxRate])
```

Questa formula ottiene l'aliquota di imposta per la regione corrente dalla tabella Region e la somma al valore della colonna Freight. Nelle formule DAX non è necessario conoscere o specificare la relazione specifica che connette le tabelle.

## Contesto di più righe

In DAX sono inoltre incluse funzioni che iterano i calcoli in una tabella. Queste funzioni possono presentare più righe correnti, ognuna con un proprio contesto di riga. In pratica, queste funzioni consentono di creare formule mediante le quali vengono eseguite operazioni in modo ricorsivo su un ciclo interno ed esterno.

Si supponga ad esempio che in un modello sia contenuta una tabella **Products** e una tabella **Sales**. Potrebbe essere necessario scorrere l'intera tabella delle vendite, piena di transazioni che riguardano più prodotti, e individuare la quantità più grande ordinata per ogni prodotto in una transazione qualunque.

Con DAX è possibile compilare una sola formula mediante la quale viene restituito il valore corretto e i risultati vengono aggiornati automaticamente tutte le volte che un utente aggiunge dati alle tabelle.

## DAX

```
= MAXX(FILTER(Sales,[ProdKey] = EARLIER([ProdKey])),Sales[OrderQty])
```

Per un esempio dettagliato di questa formula, vedere [Funzione EARLIER](#).

Per riepilogare, la funzione EARLIER consente di archiviare il contesto di riga dall'operazione che ha preceduto l'operazione corrente. La funzione archivia sempre in memoria due set di contesto: uno rappresenta la riga corrente del ciclo interno della formula e l'altro rappresenta la riga corrente del ciclo esterno della formula. DAX utilizza automaticamente i valori tra due cicli in modo che sia possibile creare aggregazioni complesse.

## Contesto di query

Il *contesto di query* fa riferimento al subset di dati recuperato in modo implicito per una formula. Quando un utente inserisce una misura o un campo in un report, ad esempio, il motore esamina le intestazioni di riga e di colonna, i filtri dei dati e i filtri del report per determinare il contesto. Le query necessarie vengono quindi eseguite sui dati del modello per ottenere il subset corretto di dati, effettuare i calcoli definiti dalla formula e quindi popolare i valori nel report.

Dal momento che il contesto cambia a seconda della posizione della formula, anche i risultati della formula possono cambiare. Si supponga ad esempio di creare una formula in cui vengono sommati i valori della colonna **Profit** della tabella **Sales** : =  
`SUM('Sales'[Profit])`. Se si utilizza tale formula in una colonna calcolata all'interno della tabella **Sales**, i risultati saranno uguali per l'intera tabella, in quanto il contesto di query per la formula è sempre l'intero set di dati della tabella **Sales**. I risultati indicheranno i profitti per tutte le regioni, tutti i prodotti, tutti gli anni e così via.

In genere, non è tuttavia necessario visualizzare lo stesso risultato centinaia di volte, poiché è più utile ottenere i profitti per un anno, un paese, un prodotto specifico o una combinazione di tali elementi, per pervenire quindi a un totale complessivo.

In un report è possibile modificare il contesto filtrando, aggiungendo o rimuovendo campi e usando i filtri dei dati. Per ogni modifica, cambia il contesto di query in cui la misura viene valutata. Pertanto, la stessa formula utilizzata in una misura viene valutata in un *contesto di query* diverso per ogni cella.

## Contesto di filtro

Il *contesto di filtro* è il set di valori consentito in ogni colonna o nei valori recuperati da una tabella correlata. I filtri possono essere applicati alla colonna nella finestra di progettazione o nel livello di presentazione (report e tabelle pivot). Possono essere definiti inoltre in modo esplicito dalle espressioni di filtro all'interno della formula.

Il contesto di filtro viene aggiunto quando si specificano vincoli del filtro sul set di valori consentito in una colonna o una tabella utilizzando gli argomenti di una formula. Tale contesto viene applicato su altri contesti, ad esempio il contesto di riga o il contesto di query.

Nei modelli tabulari sono disponibili molti modi per creare il contesto di filtro.

All'interno del contesto dei client che possono usare il modello, ad esempio report di Power BI, gli utenti possono creare filtri rapidamente aggiungendo filtri dei dati o filtri dei report sulle intestazioni di riga e di colonna. È possibile specificare anche espressioni di filtro direttamente all'interno della formula, per specificare i valori correlati, per filtrare tabelle utilizzate come input o per ottenere dinamicamente il contesto per i valori utilizzati nei calcoli. È inoltre possibile deselezionare completamente o in modo selettivo i filtri in colonne specifiche. Questa operazione risulta molto utile quando si creano formule che consentono di calcolare totali complessivi.

Per altre informazioni su come creare filtri nelle formule, vedere [Funzione FILTER \(DAX\)](#).

Per un esempio della modalità di deselezione dei filtri per creare totali complessivi, vedere [Funzione ALL \(DAX\)](#).

Per esempi su come deselezionare e applicare in modo selettivo i filtri nelle formule, vedere [ALLEXCEPT](#).

## Determinazione del contesto nelle formule

Una volta creata una formula DAX, viene innanzitutto testata la validità della relativa sintassi e viene quindi verificato che i nomi delle colonne e delle tabelle incluse nella formula siano presenti nel contesto corrente. Se non è possibile trovare una colonna o una tabella specificata dalla formula, viene restituito un errore.

Il contesto durante la convalida, e le operazioni di ricalcolo, viene determinato come descritto nelle sezioni precedenti utilizzando le tabelle disponibili nel modello, eventuali relazioni tra le tabelle ed eventuali filtri applicati.

Se ad esempio sono appena stati importati dati in una nuova tabella che non sono correlati a nessun'altra e a cui non è stato applicato alcun filtro, il *contesto corrente* è tutto il set di colonne della tabella. Se la tabella è collegata tramite relazioni ad altre tabelle, nel contesto corrente sono incluse le tabelle correlate. Se si aggiunge una

colonna della tabella a un report che dispone di filtri dei dati e forse di alcuni filtri report, il contesto per la formula è il subset di dati in ogni cella del report.

Quello di contesto è un concetto articolato e complesso, che può rendere difficile risolvere i problemi relativi alle formule. È consigliabile iniziare con formule e relazioni semplici per verificare il funzionamento del contesto. Nella sezione seguente vengono forniti alcuni esempi del modo in cui le formule utilizzano tipi diversi di contesto per restituire risultati in modo dinamico.

## Operatori

Nel linguaggio DAX vengono utilizzati quattro tipi diversi di operatori di calcolo nelle formule:

- Operatori di confronto per confrontare valori e restituire un valore logico TRUE\FALSE.
- Operatori aritmetici per eseguire calcoli aritmetici che restituiscono valori numerici.
- Operatori di concatenazione di testo per unire in join due o più stringhe di testo.
- Operatori logici per combinare due o più espressioni e restituire un singolo risultato.

Per informazioni dettagliate sugli operatori usati nelle formule DAX, vedere [Operatori DAX](#).

## Utilizzo di tabelle e colonne

Le tabelle nei modelli di dati tabulari sono simili a quelle di Excel, ma differiscono nell'utilizzo dei dati e delle formule:

- Nelle formule vengono utilizzate solo tabelle e colonne, non celle singole, riferimenti a intervalli o matrici.
- Nelle formule possono essere utilizzate relazioni per ottenere i valori dalle tabelle correlate. I valori recuperati sono sempre correlati al valore della riga corrente.
- Non è possibile avere dati irregolari o non allineati come in un foglio di lavoro di Excel. Ogni riga di una tabella deve contenere lo stesso numero di colonne. Tuttavia è possibile disporre di valori vuoti in alcune colonne. Le tabelle di dati di Excel e quelle dei modelli tabulari non sono intercambiabili.
- Poiché per ogni colonna viene impostato un tipo di dati, ogni valore nella colonna deve essere dello stesso tipo.

## Riferimento a tabelle e colonne nelle formule

È possibile fare riferimento a qualsiasi tabella e colonna tramite il relativo nome. Ad esempio, nella formula seguente viene illustrato come fare riferimento alle colonne da due tabelle utilizzando il nome *completo* :

DAX

```
= SUM('New Sales'[Amount]) + SUM('Past Sales'[Amount])
```

Durante la valutazione di una formula, in Progettazione modelli viene verificata prima la sintassi generale, quindi vengono controllati i nomi delle colonne e delle tabelle forniti in base alle possibili colonne e tabelle del contesto corrente. Se il nome è ambiguo o non è possibile trovare la colonna o tabella, si verificherà un errore nella formula e nelle celle in cui si è verificato l'errore verrà visualizzata una stringa #ERRORE anziché il valore di dati. Per altre informazioni sui requisiti di denominazione per tabelle, colonne e altri oggetti, vedere la sezione "Requisiti per la denominazione" in [Sintassi DAX](#).

## Relazioni tra tabelle

Creando relazioni tra le tabelle, è possibile usare i valori correlati in altre tabelle nei calcoli. È ad esempio possibile usare una colonna calcolata per determinare tutti i record di spedizione correlati al rivenditore corrente e quindi sommare i costi di spedizione per ognuno. In molti casi, tuttavia, una relazione potrebbe non essere necessaria. È possibile usare la funzione [LOOKUPVALUE](#) in una formula per restituire il valore in *result\_columnName* per la riga che soddisfa i criteri specificati negli argomenti *search\_column* e *search\_value*.

Molte funzioni DAX richiedono l'esistenza di una relazione tra le tabelle, o tra più tabelle, per consentire l'individuazione delle colonne cui è stato fatto riferimento e restituire risultati appropriati. Tramite altre funzioni verrà tentata l'identificazione della relazione, tuttavia per ottenere i migliori risultati è consigliabile creare sempre una relazione dove possibile. I modelli di dati tabulari supportano più relazioni tra tabelle. Per evitare confusione o risultati errati, viene designata come relazione attiva una sola relazione alla volta, ma è possibile modificare tale relazione, in base alle esigenze, per attraversare connessioni diverse nei dati dei calcoli. Per specificare una o più relazioni da usare in un calcolo specifico è disponibile la funzione [USERELATIONSHIP](#).

Quando si usano le relazioni è importante rispettare le seguenti regole di progettazione delle formule:

- Quando le tabelle sono connesse tramite una relazione, è necessario assicurarsi che le due colonne usate come chiavi dispongano di valori corrispondenti. L'integrità referenziale non viene tuttavia applicata, pertanto è possibile che in una colonna chiave siano presenti valori non corrispondenti, ma che si possa comunque creare una relazione. In tal caso, è necessario tenere presente che i valori vuoti o non corrispondenti potrebbero influire sui risultati delle formule.
- Quando si collegano tabelle nel modello tramite relazioni, viene ampliato l'ambito o *contestone*l quale vengono valutate le formule. Le modifiche al contesto che derivano dall'aggiunta di nuove tabelle, di nuove relazioni o da cambiamenti della relazione attiva possono causare modiche ai risultati difficili da prevedere. Per altre informazioni, vedere la sezione [Contesto](#) in questo articolo.

## Elaborazione e aggiornamento

L'*elaborazione* e il *ricalcolo* sono due operazioni distinte ma correlate. È necessario comprendere in modo approfondito questi concetti ai fini della progettazione di un modello contenente formule complesse, grandi quantità di dati o dati ottenuti da origini dati esterne.

L'*elaborazione (aggiornamento)* è il processo di aggiornamento dei dati in un modello con nuovi dati provenienti da un'origine dati esterna.

Il *ricalcolo* è il processo di aggiornamento dei risultati delle formule in modo che riflettano qualsiasi modifica alle formule stesse e le modifiche nei dati sottostanti. Il ricalcolo può avere effetto sulle prestazioni nei modi seguenti:

- I valori in una colonna calcolata vengono calcolati e archiviati nel modello. Per aggiornare i valori nella colonna calcolata, è necessario elaborare il modello utilizzando uno di tre comandi di elaborazione: . Elaborazione completa, Elaborazione dati o Elabora ricalcolo. È necessario sempre ricalcolare il risultato della formula per la colonna intera, ogni volta che la formula viene modificata.
- I valori calcolati dalle misure vengono valutati dinamicamente ogni volta che un utente aggiunge la misura a una tabella pivot o apre un report. Quando l'utente modifica il contesto, i valori restituiti dalla misura cambiano. I risultati della misura riflettono sempre gli ultimi dati nella cache in memoria.

L'elaborazione e il ricalcolo non hanno effetto sulle formule per la sicurezza a livello di riga, a meno che il risultato di un ricalcolo non restituisca un valore diverso, rendendo in tal modo possibile o impedendo l'esecuzione di query sulla riga da parte di membri del ruolo.

# Aggiornamenti

In DAX vengono continuamente apportati miglioramenti. [Funzioni nuove e aggiornate](#) vengono rilasciate con il successivo aggiornamento disponibile, che in genere è a cadenza mensile. I servizi vengono aggiornati per primi, seguiti dalle applicazioni installate come Power BI Desktop, Excel, SQL Server Management Studio (SSMS) e l'estensione dei progetti di Analysis Services per Visual Studio (SSDT). SQL Server Analysis Services viene aggiornato con l'aggiornamento cumulativo successivo. Le nuove funzioni vengono prima annunciate e descritte nella documentazione di riferimento delle funzioni DAX in concomitanza con gli aggiornamenti di Power BI Desktop.

Non tutte le funzioni sono supportate nelle versioni precedenti di SQL Server Analysis Services ed Excel.

## Risoluzione dei problemi

Se si ottiene un errore quando si definisce una formula, è possibile che la formula contenga un *errore sintattico*, un *errore semantico* o un *errore di calcolo*.

Gli errori sintattici sono i più facili da risolvere. In genere sono dovuti a una parentesi o una virgola mancante.

L'altro tipo di errore si verifica quando la sintassi è corretta, ma il valore o la colonna a cui si fa riferimento non è appropriato nel contesto della formula. Tali errori semanticici e di calcolo potrebbero essere causati da uno qualsiasi dei problemi seguenti:

- La formula fa riferimento a una colonna, una tabella o una funzione non esistente.
- La formula sembra essere corretta, ma quando il motore dati recupera i dati, viene rilevato un tipo non corrispondente e quindi generato un errore.
- La formula passa a una funzione un numero o un tipo di argomento errato.
- La formula fa riferimento a una colonna diversa che contiene un errore e pertanto i valori non sono validi.
- La formula fa riferimento a una colonna che non è stata elaborata, pertanto dispone di metadati ma non dati effettivi da utilizzare per i calcoli.

Nei primi quattro casi, tramite DAX viene contrassegnata l'intera colonna in cui è contenuta la formula non valida. Nell'ultimo caso, tramite DAX la colonna che si trova in uno stato non elaborato viene visualizzata in grigio.

## App e strumenti

## Power BI Desktop



Power BI Desktop [🔗](#) è un'applicazione gratuita per la creazione di report e la modellazione dei dati. La funzionalità per la progettazione dei modelli include un editor DAX che consente di creare formule di calcolo DAX.

## Power Pivot in Excel



La funzionalità per la progettazione dei modelli di Power Pivot in Excel [🔗](#) include un editor DAX per la creazione di formule di calcolo DAX.

## Visual Studio



Visual Studio con l'estensione dei [progetti di Analysis Services](#) [🔗](#) (VSIX) consente di creare progetti di modelli di Analysis Services. Progettazione modelli tabulari, installato con l'estensione dei progetti, include un editor DAX.

## SQL Server Management Studio



[SQL Server Management Studio](#) (SSMS) è uno strumento essenziale per l'utilizzo di Analysis Services. Include un editor di query DAX per l'esecuzione di query su modelli tabulari e multidimensionali.

## DAX Studio



[DAX Studio](#) è uno strumento client open source per la creazione e l'esecuzione di query DAX su modelli di Analysis Services, Power BI Desktop e Power Pivot in Excel.

## Tabular Editor



[Tabular Editor](#) è uno strumento open source che offre una visualizzazione gerarchica intuitiva di ogni oggetto nei metadati del modello tabulare. Tabular Editor include un editor DAX con evidenziazione della sintassi, che consente di modificare in modo semplice misure, colonne calcolate ed espressioni di tabella calcolata.

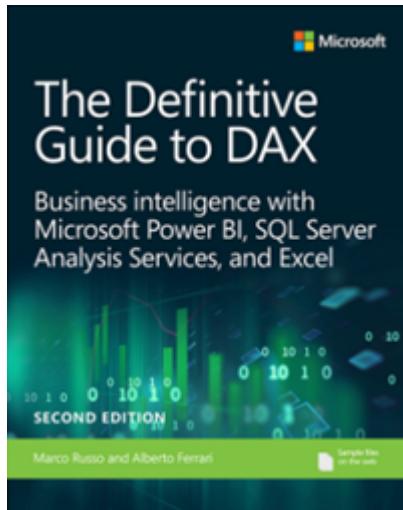
## Risorse di formazione

Durante la fase di apprendimento del linguaggio DAX, è preferibile eseguire l'applicazione che si userà per creare i modelli di dati. Per Analysis Services, Power BI Desktop e Power Pivot in Excel sono disponibili articoli ed esercitazioni che includono lezioni sulla creazione di misure, colonne calcolate e filtri di riga tramite DAX. Altre risorse:

### [Video](#)

Percorso di apprendimento: [Usare DAX in Power BI Desktop](#).

[The Definitive Guide to DAX](#) di Alberto Ferrari e Marco Russo (Microsoft Press). Nella seconda edizione, questa guida completa fornisce nozioni di base relativamente alle tecniche innovative ad alte prestazioni per autori di modelli di dati e professionisti di Business Intelligence che si vogliono avvicinare a questo linguaggio.



## Community

DAX ha una vivace community sempre disposta a condividere le proprie competenze. La [Community di Power BI](#) Microsoft presenta un forum di discussione speciale solo per DAX, [suggerimenti e comandi DAX](#).

---

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì No

# Video

Articolo • 20/10/2023

Indipendentemente dallo strumento usato, Power BI Desktop, Power Pivot in Excel o Analysis Services, la conoscenza del linguaggio DAX (Data Analysis Expressions) è di fondamentale importanza per creare modelli di dati efficaci. Di seguito sono riportati alcuni video utili per iniziare a usare questo potente linguaggio di espressioni.

## DAX 101

In questo video DAX 101 il Microsoft Partner Alberto Ferrari presenta i concetti fondamentali del linguaggio DAX. Con esempi pratici e chiari, viene illustrato l'uso di misure, colonne calcolate ed espressioni di modellazione dei dati di base con DAX.

[https://www.youtube-nocookie.com/embed/kIQAZLr5vxA ↗](https://www.youtube-nocookie.com/embed/kIQAZLr5vxA)

## Concetti DAX avanzati

In questo video relativo ai concetti avanzati di DAX il Microsoft Partner Alberto Ferrari descrive gli aspetti teorici di DAX, il contesto di filtro e di riga e altri concetti fondamentali di questo linguaggio.

[https://www.youtube-nocookie.com/embed/6ncHnWMEdic ↗](https://www.youtube-nocookie.com/embed/6ncHnWMEdic)

## DAX pratico per Power BI

In questo video, Phil Seemark, Microsoft Principal Program Manager, illustra alcune procedure consigliate per gli usi pratici di DAX in Power BI.

[https://www.youtube-nocookie.com/embed/1fGfqzS37qs ↗](https://www.youtube-nocookie.com/embed/1fGfqzS37qs)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Modello DAX di esempio

Articolo • 20/10/2023

Il modello di esempio **Adventure Works DW 2020** di Power BI Desktop è progettato per supportare l'apprendimento DAX. Il modello si basa sul [data warehouse di esempio Adventure Works](#) per AdventureWorksDW2017, tuttavia i dati sono stati modificati in modo da essere adeguati agli obiettivi del modello di esempio.

Il modello campione non contiene formule DAX. Tuttavia, supporta centinaia, o persino migliaia di formule di calcoli potenziali e query. Alcuni esempi di funzioni, come CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF e LOOKUPVALUE possono essere aggiunti al modello campione senza modifiche. Stiamo lavorando per includere più esempi in altri articoli di riferimento delle funzioni che funzionano con il modello campione.

## Scenario



Adventure Works è un'azienda che produce biciclette e le vende con i relativi accessori ai mercati globali. L'azienda ha il proprio data warehouse archiviato in un database SQL di Azure.

## Struttura del modello

Il modello contiene sette tabelle:

Tabella	Descrizione
Cliente	Describe i clienti e la relativa posizione geografica. I clienti acquistano i prodotti online (vendite Internet).
Date	Esistono tre relazioni tra le tabelle Date e Sales: data dell'ordine, data di spedizione e data di scadenza. La relazione "Order date" è attiva. Per le vendite nei report dell'azienda si usa un anno fiscale che inizia il 1° luglio di ogni anno. La tabella è contrassegnata come tabella data usando la colonna Date.

Tabella	Descrizione
Prodotto	Archivia solo i prodotti finiti.
Reseller	Describe i rivenditori e la relativa posizione geografica. I rivenditori vendono i prodotti ai clienti.
Sales	Archivia le righe a livello di voce dell'ordine di vendita. Tutti i valori finanziari sono in dollari statunitensi (USD). La data meno recente dell'ordine è il 1 luglio 2017 e la data più recente è il 15 giugno 2020.
Sales Order	Describe i numeri di ordine e riga dell'ordine di vendita nonché il canale di vendita, che è Reseller o Internet. Questa tabella ha una relazione uno-a-uno con la tabella Sales.
Sales Territory	I territori di vendita sono organizzati in gruppi (America del Nord, Europa e Pacifico), paesi e aree geografiche. Solo gli Stati Uniti vendono i prodotti a livello di area.

## Scaricare l'esempio

Scaricare il file del modello di esempio di Power BI Desktop [qui](#).

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#)
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?

 Sì
 No

# Informazioni sulle funzioni ORDERBY, PARTITIONBY e MATCHBY

Articolo • 05/10/2023

[ORDERBY](#), [PARTITIONBY](#) e [MATCHBY](#) in DAX sono funzioni speciali che possono essere usate solo insieme alle funzioni finestra DAX: [INDEX](#), [OFFSET](#), [WINDOW](#), [RANK](#), [ROWNUMBER](#).

Comprendere ORDERBY, PARTITIONBY e MATCHBY è fondamentale per usare correttamente le funzioni finestra. Gli esempi forniti qui usano OFFSET, ma sono applicabili in modo analogo alle altre funzioni finestra.

## Scenario

Iniziamo con un esempio che non usa affatto le funzioni finestra. Di seguito è riportata una tabella che restituisce le vendite totali, per colore e per anno di calendario. Esistono vari modi per definire questa tabella, ma poiché l'obiettivo è comprendere cosa accade in DAX, si userà una tabella calcolata. Ecco l'espressione di tabella:

```
DAX

BasicTable =
    SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
    0 )
    )
```

Questa espressione di tabella calcolata usa [SUMMARIZECOLUMNS](#) per calcolare la somma della colonna SalesAmount nella tabella FactInternetSales, in base alla colonna Color della tabella DimProduct e in base alla colonna CalendarYear della tabella DimDate. Ecco il risultato:

Colore	CalendarYear	CurrentYearSales
"Black"	2017	393885
"Black"	2018	1818835
"Black"	2019	3981638

Colore	CalendarYear	CurrentYearSales
"Black"	2020	2644054
"Blue"	2019	994448
"Blue"	2020	1284648
"Multi"	2019	48622
"Multi"	2020	57849
"NA"	2019	207822
"NA"	2020	227295
"Red"	2017	2961198
"Red"	2018	3686935
"Red"	2019	900175
"Red"	2020	176022
"Silver"	2017	326399
"Silver"	2018	750026
"Silver"	2019	2165176
"Silver"	2020	1871788
"White"	2019	2517
"White"	2020	2589
"Yellow"	2018	163071
"Yellow"	2019	2072083
"Yellow"	2020	2621602

Supponiamo di voler rispondere alla domanda aziendale di calcolare la differenza nelle vendite, anno per anno, per ogni colore. In effetti è necessario un modo per trovare le vendite per lo stesso colore nell'anno precedente e sottrarre il valore dalle vendite nell'anno corrente, nel contesto. Ad esempio, per la combinazione [Red, 2019] stiamo cercando le vendite per [Red, 2018]. Una volta ottenuto questo valore, è possibile sottrarlo dalle vendite correnti e restituire il valore richiesto.

## Utilizzo di OFFSET

OFFSET è perfetto per i tipici calcoli *confronta con i precedenti* necessari per rispondere alla domanda aziendale descritta in precedenza, in quanto ci consente di eseguire un movimento relativo. Il primo tentativo potrebbe essere:

```
DAX

1stAttempt =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
    0 )
)
RETURN
ADDCOLUMNS (
    vRelation,
    "PreviousColorSales",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation
        ),
        [CurrentYearSales]
    )
)
```

In questa espressione accade molto. È stato usato **ADDCOLUMNS** per espandere la tabella precedente con una colonna denominata PreviousColorSales. Il contenuto di tale colonna viene impostato su CurrentYearSales, ovvero  $\text{SUM}(\text{FactInternetSales}[\text{SalesAmount}])$ , per il colore precedente (recuperato tramite **OFFSET**).

Il risultato è:

Colore	CalendarYear	CurrentYearSales	PreviousColorSales
"Black"	2017	393885	
"Black"	2018	1818835	393885
"Black"	2019	3981638	1818835
"Black"	2020	2644054	3981638
"Blue"	2019	994448	2644054
"Blue"	2020	1284648	994448
"Multi"	2019	48622	1284648

Colore	CalendarYear	CurrentYearSales	PreviousColorSales
"Multi"	2020	57849	48622
"NA"	2019	207822	57849
"NA"	2020	227295	207822
"Red"	2017	2961198	227295
"Red"	2018	3686935	2961198
"Red"	2019	900175	3686935
"Red"	2020	176022	900175
"Silver"	2017	326399	176022
"Silver"	2018	750026	326399
"Silver"	2019	2165176	750026
"Silver"	2020	1871788	2165176
"White"	2019	2517	1871788
"White"	2020	2589	2517
"Yellow"	2018	163071	2589
"Yellow"	2019	2072083	163071
"Yellow"	2020	2621602	2072083

Siamo un passo più vicini al nostro obiettivo, ma se guardiamo attentamente non corrisponde esattamente a ciò che cerchiamo. Ad esempio, per [Silver, 2017] PreviousColorSales è impostato su [Red, 2020].

## Aggiunta di ORDERBY

La definizione precedente equivale a:

DAX

```
1stAttemptWithORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
        0 )
    )
```

```

RETURN
ADDCOLUMNS (
    vRelation,
    "PreviousColorSales",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation,
            ORDERBY ([Color], ASC, [CalendarYear], ASC,
[CurrentYearSales], ASC)
        ),
        [CurrentYearSales]
    )
)

```

In questo caso, la chiamata a OFFSET usa ORDERBY per ordinare la tabella in base a Color e CalendarYear in ordine crescente, il che determina cosa viene considerato la riga precedente restituita.

Il motivo per cui questi due risultati sono equivalenti è che ORDERBY contiene automaticamente tutte le colonne della relazione che non sono in PARTITIONBY. Poiché PARTITIONBY non è stato specificato, ORDERBY è impostato su Color, CalendarYear e CurrentYearSales. Tuttavia, poiché le coppie Color e CalendarYear nella relazione sono univoche, l'aggiunta di CurrentYearSales non modifica il risultato. Infatti, anche specificando solo Color in ORDERBY, i risultati sarebbero gli stessi poiché CalendarYear verrebbe aggiunto automaticamente. Ciò è dovuto al fatto che la funzione aggiungerà il numero di colonne necessarie a ORDERBY per garantire che ogni riga possa essere identificata in modo univoco dalle colonne ORDERBY e PARTITIONBY:

#### DAX

```

1stAttemptWithORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
    )
RETURN
ADDCOLUMNS(
    vRelation,
    "PreviousColorSales",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation,
            ORDERBY ([Color])
        ),
        [CurrentYearSales]
)

```

```
)  
)
```

## Aggiunta di PARTITIONBY

Ora, per ottenere *quasi* il risultato desiderato possiamo utilizzare PARTITIONBY, come illustrato nell'espressione di tabella calcolata seguente:

DAX

```
UsingPARTITIONBY =  
    VAR vRelation = SUMMARIZECOLUMNS (  
        DimProduct[Color],  
        DimDate[CalendarYear],  
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),  
0 )  
    )  
    RETURN  
    ADDCOLUMNS (  
        vRelation,  
        "PreviousColorSales",  
        SELECTCOLUMNS (  
            OFFSET (  
                -1,  
                vRelation,  
                ORDERBY ([CalendarYear]),  
                PARTITIONBY ([Color])  
            ),  
            [CurrentYearSales]  
        )  
    )
```

Si noti che specificare ORDERBY è facoltativo, perché ORDERBY contiene automaticamente tutte le colonne della relazione non specificate in PARTITIONBY. L'espressione seguente restituisce quindi gli stessi risultati, perché ORDERBY è impostato automaticamente su CalendarYear e CurrentYearSales:

DAX

```
UsingPARTITIONBYWithoutORDERBY =  
    VAR vRelation = SUMMARIZECOLUMNS (  
        DimProduct[Color],  
        DimDate[CalendarYear],  
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),  
0 )  
    )  
    RETURN
```

```

ADDCOLUMNS (
    vRelation,
    "PreviousColorSales",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation,
            PARTITIONBY ([Color])
        ),
        [CurrentYearSales]
    )
)

```

### ① Nota

Mentre ORDERBY è impostato automaticamente su CalendarYear e CurrentYearSales, non esiste alcuna garanzia in merito all'ordine in cui verranno sommati. Se CurrentYearSales viene aggiunto prima di CalendarYear, l'ordine risultante non è in linea con quello previsto. **Specificare ORDERBY e PARTITIONBY in modo esplicito per evitare confusione e risultati imprevisti.**

Entrambe le espressioni restituiscono il risultato cercato:

Colore	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"Black"	2017	393885	
"Black"	2018	1818835	393885
"Black"	2019	3981638	1818835
"Black"	2020	2644054	3981638
"Blue"	2019	994448	
"Blue"	2020	1284648	994448
"Multi"	2019	48622	
"Multi"	2020	57849	48622
"NA"	2019	207822	
"NA"	2020	227295	207822
"Red"	2017	2961198	

Colore	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"Red"	2018	3686935	2961198
"Red"	2019	900175	3686935
"Red"	2020	176022	900175
"Silver"	2017	326399	
"Silver"	2018	750026	326399
"Silver"	2019	2165176	750026
"Silver"	2020	1871788	2165176
"White"	2019	2517	
"White"	2020	2589	2517
"Yellow"	2018	163071	
"Yellow"	2019	2072083	163071
"Yellow"	2020	2621602	2072083

Come illustrato in questa tabella, la colonna PreviousYearSalesForSameColor mostra le vendite per l'anno precedente per lo stesso colore. Per [Red, 2020] restituisce le vendite per [Red, 2019] e così via. Se non è presente un anno precedente, ad esempio nel caso di [Red, 2017], non viene restituito alcun valore.

È possibile considerare PARTITIONBY come un modo per dividere la tabella in parti in cui eseguire il calcolo OFFSET. Nell'esempio precedente la tabella è suddivisa in tante parti quanti sono i colori, una per ogni colore. Quindi, all'interno di ogni parte, viene calcolato il valore OFFSET, ordinato in base a CalendarYear.

Visivamente, ciò che accade è questo:

Color	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"Black"	2017	[ 393885 ]	
"Black"	2018	[ 1818835 ]	393885
"Black"	2019	[ 3981638 ]	1818835
"Black"	2020	2644054	3981638
"Blue"	2019	[ 994448 ]	
"Blue"	2020	1284648	994448
"Multi"	2019	[ 48622 ]	
"Multi"	2020	57849	48622
"NA"	2019	[ 207822 ]	
"NA"	2020	227295	207822
"Red"	2017	[ 2961198 ]	
"Red"	2018	[ 3686935 ]	2961198
"Red"	2019	[ 900175 ]	3686935
"Red"	2020	176022	900175
"Silver"	2017	[ 326399 ]	
"Silver"	2018	[ 750026 ]	326399
"Silver"	2019	[ 2165176 ]	750026
"Silver"	2020	1871788	2165176
"White"	2019	[ 2517 ]	
"White"	2020	2589	2517
"Yellow"	2018	[ 163071 ]	
"Yellow"	2019	[ 2072083 ]	163071
"Yellow"	2020	2621602	2072083

Prima di tutto, la chiamata a PARTITIONBY risulta nella suddivisione della tabella in parti, una per ogni colore. Questo è rappresentato dalle caselle blu chiaro nell'immagine della tabella. Quindi, ORDERBY fa in modo che ogni parte sia ordinata in base a CalendarYear (rappresentato dalle frecce arancioni). Infine, all'interno di ogni parte ordinata, OFFSET trova la riga precedente e restituisce tale valore nella colonna PreviousYearSalesForSameColor. Poiché per ogni prima riga di ogni parte non esiste alcuna riga precedente nella stessa parte, il risultato della riga per la colonna PreviousYearSalesForSameColor è vuoto.

Per ottenere il risultato finale, è sufficiente sottrarre il valore CurrentYearSales dalle vendite dell'anno precedente per lo stesso colore, restituito dalla chiamata a OFFSET. Poiché non ci interessa visualizzare le vendite dell'anno precedente per lo stesso colore, ma solo le vendite dell'anno corrente e la differenza di anno in anno. Ecco l'espressione di tabella calcolata finale:

DAX

```

FinalResult =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
        0 )

```

```

)
RETURN
ADDCOLUMNS (
    vRelation,
    "YoYSalesForSameColor",
    [CurrentYearSales] -
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation,
            ORDERBY ([CalendarYear]),
            PARTITIONBY ([Color])
        ),
        [CurrentYearSales]
    )
)

```

Ed ecco il risultato dell'espressione:

Colore	CalendarYear	CurrentYearSales	YoYSalesForSameColor
"Black"	2017	393885	393885
"Black"	2018	1818835	1424950
"Black"	2019	3981638	2162803
"Black"	2020	2644054	-1337584
"Blue"	2019	994448	994448
"Blue"	2020	1284648	290200
"Multi"	2019	48622	48622
"Multi"	2020	57849	9227
"NA"	2019	207822	207822
"NA"	2020	227295	19473
"Red"	2017	2961198	2961198
"Red"	2018	3686935	725737
"Red"	2019	900175	-2786760
"Red"	2020	176022	-724153
"Silver"	2017	326399	326399
"Silver"	2018	750026	423627

Colore	CalendarYear	CurrentYearSales	YoYSalesForSameColor
"Silver"	2019	2165176	1415150
"Silver"	2020	1871788	-293388
"White"	2019	2517	2517
"White"	2020	2589	72
"Yellow"	2018	163071	163071
"Yellow"	2019	2072083	1909012
"Yellow"	2020	2621602	549519

## Uso di MATCHBY

Probabilmente si è notato che MATCHBY non è stato specificato. In questo caso, non è necessario. Le colonne in ORDERBY e PARTITIONBY (per quanto specificato negli esempi precedenti) sono sufficienti per identificare in modo univoco ogni riga. Poiché non è stato specificato MATCHBY, le colonne specificate in ORDERBY e PARTITIONBY vengono usate per identificare in modo univoco ogni riga in modo che sia possibile confrontarle per consentire a OFFSET di fornire un risultato significativo. Se le colonne in ORDERBY e PARTITIONBY non possono identificare in modo univoco ogni riga, è possibile aggiungere altre colonne alla clausola ORDERBY se tali colonne extra consentono di identificare in modo univoco ogni riga. Se questo non è possibile, viene restituito un errore. In questo ultimo caso, specificare MATCHBY può consentire di risolvere l'errore.

Se si specifica MATCHBY, le colonne in MATCHBY e PARTITIONBY vengono utilizzate per identificare in modo univoco ogni riga. Se questo non è possibile, viene restituito un errore. Anche se MATCHBY non è obbligatorio, è consigliabile specificare in modo esplicito MATCHBY per evitare confusione.

Continuando dagli esempi precedenti, ecco l'ultima espressione:

```
DAX

FinalResult =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
        0 )
    )
    RETURN
    ADDCOLUMNS (
```

```

    vRelation,
    "YoYSalesForSameColor",
    [CurrentYearSales] -
    SELECTCOLUMNS (
        OFFSET (
            -1,
            vRelation,
            ORDERBY ([CalendarYear]),
            PARTITIONBY ([Color])
        ),
        [CurrentYearSales]
    )
)

```

Se si vuole indicare esplicitamente come identificare in modo univoco le righe, è possibile specificare MATCHBY come illustrato nell'espressione equivalente seguente:

DAX

```

FinalResultWithExplicitMATCHBYOnColorAndCalendarYear =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
    )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color]),
                MATCHBY ([Color], [CalendarYear])
            ),
            [CurrentYearSales]
        )
    )
)

```

Poiché è specificato MATCHBY, entrambe le colonne specificate in MATCHBY e in PARTITIONBY vengono usate per identificare in modo univoco le righe. Poiché Color è specificato sia in MATCHBY che in PARTITIONBY, l'espressione seguente equivale all'espressione precedente:

## DAX

```
FinalResultWithExplicitMATCHBYOnCalendarYear =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ),
0 )
    )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color]),
                MATCHBY ([CalendarYear])
            ),
            [CurrentYearSales]
        )
    )
)
```

Poiché specificare MATCHBY non è necessario negli esempi esaminati finora, si esaminerà un esempio leggermente diverso che richiede MATCHBY. In questo caso, è presente un elenco di righe di ordini. Ogni riga rappresenta una riga di ordine per un ordine. Un ordine può avere più righe di ordine e la riga ordine 1 compare in molti ordini. Inoltre, per ogni riga di ordine abbiamo un valore ProductKey e un valore SalesAmount. Un esempio delle colonne pertinenti nella tabella può essere simile al seguente:

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount
SO51900	1	528	4.99
SO51948	1	528	5,99
SO52043	1	528	4.99
SO52045	1	528	4.99
SO52094	1	528	4.99
SO52175	1	528	4.99
SO52190	1	528	4.99

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount
SO52232	1	528	4.99
SO52234	1	528	4.99
SO52234	2	529	3,99

Si noti che SalesOrderNumber e SalesOrderLineNumber sono entrambi necessari per identificare in modo univoco le righe.

Per ogni ordine, si vuole restituire l'importo delle vendite precedenti dello stesso prodotto (rappresentato da ProductKey) ordinato in base a SalesAmount in ordine decrescente. L'espressione seguente non funzionerà perché sono presenti potenzialmente più righe in vRelation nel momento in cui viene passato a OFFSET:

DAX

```
ThisExpressionFailsBecauseMATCHBYIsMissing =
ADDCOLUMNS (
    FactInternetSales,
    "Previous Sales Amount",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            FactInternetSales,
            ORDERBY ( FactInternetSales[SalesAmount], DESC ),
            PARTITIONBY ( FactInternetSales[ProductKey] )
        ),
        FactInternetSales[SalesAmount]
    )
)
```

Questa espressione restituisce un errore: "Il parametro Relation di OFFSET può contenere righe duplicate. Questo non è consentito".

Per far funzionare questa espressione, è necessario specificare MATCHBY e includere tutte le colonne che definiscono in modo univoco una riga. Qui MATCHBY è necessario, perché la relazione FactInternetSales non contiene chiavi esplicite o colonne univoche. Tuttavia, le colonne SalesOrderNumber e SalesOrderLineNumber insieme formano una *chiave composita*, in cui la loro esistenza contemporanea è unica nella relazione e pertanto può identificare in modo univoco ogni riga. Semplicemente specificare SalesOrderNumber o SalesOrderLineNumber non è sufficiente, perché entrambe le colonne contengono valori ripetuti. L'espressione seguente risolve il problema:

DAX

```

ThisExpressionWorksBecauseOfMATCHBY =
    ADDCOLUMNS (
        FactInternetSales,
        "Previous Sales Amount",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                FactInternetSales,
                ORDERBY ( FactInternetSales[SalesAmount], DESC ),
                PARTITIONBY ( FactInternetSales[ProductKey] ),
                MATCHBY ( FactInternetSales[SalesOrderNumber],
                    FactInternetSales[SalesOrderLineNumber] )
            ),
            FactInternetSales[SalesAmount]
        )
    )
)

```

E questa espressione restituisce effettivamente i risultati desiderati:

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount	Previous Sales Amount
SO51900	1	528	5,99	
SO51948	1	528	4.99	5,99
SO52043	1	528	4.99	4.99
SO52045	1	528	4.99	4.99
SO52094	1	528	4.99	4.99
SO52175	1	528	4.99	4.99
SO52190	1	528	4.99	4.99
SO52232	1	528	4.99	4.99
SO52234	1	528	4.99	4.99
SO52234	2	529	3,99	

## Vedi anche

[ORDERBY](#)

[PARTITIONBY](#)

[MATCHBY](#)

[INDEX](#)

OFFSET

WINDOW

RANK

ROWNUMBER

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Uso appropriato delle funzioni di errore

Articolo • 05/10/2023

Se un'espressione DAX può generare un errore in fase di valutazione, un modeler di dati può considerare l'uso di due utili funzioni DAX.

- La funzione **ISERROR** accetta una sola espressione e restituisce TRUE se tale espressione genera un errore.
- La funzione **IFERROR** accetta due espressioni. Se la prima espressione genera un errore, viene restituito il valore per la seconda espressione. Si tratta in realtà di un'implementazione più ottimizzata dell'annidamento della funzione ISERROR all'interno di una funzione **IF**.

Sebbene queste funzioni possano essere utili e possano contribuire a scrivere espressioni facilmente comprensibili, è anche possibile che riducano significativamente le prestazioni dei calcoli. Ciò può avvenire perché tali funzioni aumentano il numero delle analisi che il motore di archiviazione deve eseguire.

La maggior parte degli errori in fase di valutazione è dovuta a valori vuoti o zero imprevisti oppure a una conversione del tipo di dati non valida.

## Consigli

È preferibile evitare di usare le funzioni ISERROR e IFERROR. In alternativa, applicare strategie difensive durante lo sviluppo del modello e la scrittura di espressioni. Possibili strategie:

- **Verifica del caricamento dei dati qualitativi nel modello:** usare le trasformazioni di Power Query per rimuovere o sostituire valori non validi o mancanti e impostare i tipi di dati corretti. È anche possibile usare una trasformazione Power Query per filtrare le righe quando si verificano errori, ad esempio in caso di conversione di dati non valida.

La qualità dei dati può essere controllata anche disabilitando la proprietà **Is Nullable** della colonna del modello che non aggiornerà i dati se si dovesse rilevare dati vuoti. Se si verifica questo errore, i dati caricati in seguito a un aggiornamento riuscito rimarranno nelle tabelle.

- **Uso della funzione IF:** l'espressione di test logico della funzione IF può determinare se si verificherebbe un risultato di errore. Si noti che, analogamente alle funzioni ISERROR e IFERROR, questa funzione può richiedere analisi aggiuntive del motore

di archiviazione, ma è probabile che le prestazioni siano migliori rispetto alle altre funzioni, in quanto non si devono generare errori.

- **Uso delle funzioni a tolleranza di errore:** alcune funzioni DAX testeranno e compenseranno le condizioni di errore. Queste funzioni consentono di immettere un risultato alternativo che verrebbe invece restituito. La funzione [DIVIDE](#) è uno di questi esempi. Per altre indicazioni su questa funzione, vedere l'articolo [Confronto tra funzione DIVIDE e operatore di divisione \(/\)](#).

## Esempio

Nell'espressione di misura seguente viene verificato se si genererà un errore. In questa istanza viene restituito un valore vuoto. Si verifica quando non si specifica la funzione IF con un'espressione value-if-false.

DAX

```
Profit Margin  
= IF(ISERROR([Profit] / [Sales]))
```

La versione successiva dell'espressione di misura è stata migliorata usando la funzione IFERROR al posto delle funzioni IF e ISERROR.

DAX

```
Profit Margin  
= IFERROR([Profit] / [Sales], BLANK())
```

Questa versione finale dell'espressione di misura ottiene lo stesso risultato, ma in modo più efficiente ed elegante.

DAX

```
Profit Margin  
= DIVIDE([Profit], [Sales])
```

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#)
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# Evitare di convertire risultati BLANK in valori

Articolo • 05/10/2023

Durante la scrittura di espressioni di misura, gli autori di modelli di dati potrebbero riscontrare casi in cui non è possibile restituire un valore significativo. In questi casi si potrebbe essere tentati di restituire un valore, ad esempio zero, in alternativa. È consigliabile determinare attentamente se questa progettazione è efficiente e pratica.

Si consideri la seguente definizione di misura che converte in modo esplicito i risultati BLANK in zero.

DAX

```
Sales (No Blank) =  
IF(  
    ISBLANK([Sales]),  
    0,  
    [Sales]  
)
```

Si consideri un'altra definizione di misura che converte anch'essa i risultati BLANK in zero.

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales], 0)
```

La funzione **DIVIDE** divide la misura **Profit** per la misura **Sales**. Se il risultato è zero o BLANK, viene restituito il terzo argomento, ovvero il risultato alternativo (facoltativo). In questo esempio, poiché viene passato zero come risultato alternativo, è garantito che la misura restituisca sempre un valore.

Queste progettazioni delle misure sono inefficienti e portano a progettazioni di report di scarsa qualità.

Quando vengono aggiunte a un oggetto visivo del report, Power BI tenta di recuperare tutti i raggruppamenti all'interno del contesto di filtro. La valutazione e il recupero di risultati di query di grandi dimensioni spesso causano rallentamenti del rendering del report. Ogni misura di esempio converte di fatto un calcolo sparso in uno denso, forzando Power BI a usare più memoria del necessario.

Inoltre, troppi raggruppamenti creano spesso confusione per gli utenti del report.

Di seguito viene illustrato cosa accade quando si aggiunge la misura **Profit Margin** a un oggetto visivo tabella, raggruppando i dati in base al cliente.

Customer	Sales	Profit Margin
AW00011000		0.00%
AW00011001		0.00%
AW00011002		0.00%
AW00011003		0.00%
AW00011004		0.00%
AW00011005		0.00%
AW00011006		0.00%
AW00011007		0.00%
AW00011008		0.00%
AW00011009		0.00%

L'oggetto visivo tabella visualizza un numero eccessivo di righe. Ci sono infatti 18.484 clienti nel modello, quindi la tabella tenta di visualizzarli tutti. Si noti che per i clienti visualizzati non risultano vendite. Vengono però visualizzati perché la misura **Profit Margin** restituisce sempre un valore.

#### ⚠️ Nota

Quando sono presenti troppi punti dati da visualizzare in un oggetto visivo, Power BI potrebbe usare strategie di riduzione dei dati per rimuovere o riepilogare risultati di query molto estesi. Per altre informazioni, vedere [Limiti dei dati per gli oggetti visivi e strategie in base al tipo di oggetto visivo](#).

Di seguito viene illustrato cosa accade migliorando la definizione della misura **Profit Margin**. Ora restituisce un valore solo quando la misura **Sales** non è BLANK (o zero).

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales])
```

L'oggetto visivo tabella visualizza ora solo i clienti con vendite nel contesto di filtro corrente. La misura migliorata produce un'esperienza più efficiente e pratica per gli utenti del report.

Customer	Sales	Profit Margin
AW00011034	1,264.51	6.72%
AW00011131	427.92	19.59%
AW00024100	1,329.08	21.07%
AW00028194	248.19	11.37%
<b>Total</b>	<b>3,269.70</b>	<b>17.57%</b>

### 💡 Suggerimento

Quando necessario, è possibile configurare un oggetto visivo per visualizzare tutti i raggruppamenti (che restituiscono valori o BLANK) nel contesto del filtro abilitando l'opzione **Mostra elementi senza dati**.

## Recommendation

È consigliabile che le misure restituiscano BLANK quando non è possibile restituire un valore significativo.

Questo approccio di progettazione è efficiente, consentendo a Power BI di eseguire il rendering dei report più velocemente. La restituzione di BLANK è consigliata anche perché per impostazione predefinita gli oggetti visivi del report eliminano i raggruppamenti quando i riepiloghi sono BLANK.

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#)
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

[Sì](#)   [No](#)

# Evitare l'uso di FILTER come argomento di filtro

Articolo • 05/10/2023

Chi si occupa della modellazione dei dati si trova spesso a scrivere espressioni DAX che devono essere valutate in un contesto di filtro modificato. È ad esempio possibile scrivere una definizione di misura per calcolare le vendite per i prodotti con margine elevato. Questo calcolo verrà descritto più avanti nell'articolo.

## ⚠ Nota

Questo articolo riguarda in particolare i calcoli dei modelli che applicano filtri alle tabelle di importazione.

Le funzioni DAX **CALCULATE** e **CALCULATETABLE** sono importanti e utili. Consentono di scrivere calcoli per la rimozione o l'aggiunta di filtri o per la modifica dei percorsi delle relazioni. A tale scopo, vengono passati argomenti di filtro che sono espressioni booleane, espressioni di tabella o funzioni di filtro speciali. In questo articolo verranno illustrate solo le espressioni booleane e di tabella.

Si consideri la definizione di misura seguente, che consente di calcolare le vendite dei prodotti rossi usando un'espressione di tabella. L'istruzione sostituisce i filtri applicati alla tabella **Product**.

DAX

```
Red Sales =  
CALCULATE(  
    [Sales],  
    FILTER('Product', 'Product'[Color] = "Red")  
)
```

La funzione **CALCULATE** accetta un'espressione di tabella restituita dalla funzione DAX **FILTER** che valuta l'espressione filtro per ogni riga della tabella **Product**. Ottiene il risultato corretto, ovvero le vendite per i prodotti rossi. È tuttavia possibile ottenere lo stesso risultato in modo molto più efficiente usando un'espressione booleana.

Ecco una definizione di misura migliorata che usa un'espressione booleana invece dell'espressione di tabella. La funzione DAX **KEEPFILTERS** garantisce che tutti i filtri esistenti applicati alla colonna **Color** vengano mantenuti, non sovrascritti.

## DAX

```
Red Sales =  
CALCULATE(  
    [Sales],  
    KEEPFILTERS('Product'[Color] = "Red")  
)
```

È consigliabile passare argomenti di filtro come espressioni booleane, quando possibile. Ciò perché le tabelle dei modelli di importazione sono archivi di colonne in memoria. Sono ottimizzate in modo esplicito per filtrare efficacemente le colonne in questo modo.

Alle espressioni booleane usate come argomenti di filtro si applicano tuttavia alcune restrizioni, ovvero:

- Impossibile fare riferimento a colonne da più tabelle
- Non possono fare riferimento a una misura
- Non possono usare funzioni CALCULATE annidate
- Non possono usare funzioni che analizzano o restituiscono una tabella

Ciò significa che per i requisiti di filtro più complessi è necessario usare le espressioni di tabella.

Si consideri ora una definizione di misura diversa. Il requisito è quello di calcolare le vendite, ma solo per i mesi che hanno ottenuto un profitto.

## DAX

```
Sales for Profitable Months =  
CALCULATE(  
    [Sales],  
    FILTER(  
        VALUES('Date'[Month]),  
        [Profit] > 0  
)  
)
```

In questo esempio è necessario usare la funzione FILTER. Ciò perché è necessario valutare la misura **Profit** per eliminare i mesi in cui non è stato ottenuto un profitto. Non è possibile usare una misura in un'espressione booleana quando viene usata come argomento di filtro.

## Consigli

Per ottenere prestazioni ottimali, è consigliabile usare espressioni booleane come argomenti di filtro, quando possibile.

La funzione FILTER deve quindi essere usata solo quando è necessario. È possibile usarla per eseguire confronti tra colonne con filtro complesso. Questi confronti tra colonne possono includere:

- Misure
- Altre colonne
- Uso della funzione DAX [OR](#) o dell'operatore logico OR (||)

## Vedi anche

- [Funzioni di filtro \(DAX\)](#)
  - Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
  - Domande? [Contattare la community di Power BI](#)
  - inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# Riferimenti a colonne e misure

Articolo • 05/10/2023

Gli autori di modelli di dati potrebbero avere l'esigenza di scrivere espressioni DAX che facciano riferimento a colonne e misure dei modelli. Le colonne e le misure sono sempre associate alle tabelle del modello, ma queste associazioni sono diverse, pertanto sono disponibili raccomandazioni diverse su come farvi riferimento nelle espressioni.

## Colonne

Una colonna è un oggetto a livello di tabella e i nomi di colonna devono essere univoci all'interno di una tabella. È quindi possibile che lo stesso nome di colonna venga usato più volte nel modello, purché appartengano a tabelle diverse. Esiste un'altra regola: un nome di colonna non può avere lo stesso nome del nome di una misura o di una gerarchia presente nella stessa tabella.

In generale, DAX non impone l'uso di un riferimento *completo* a una colonna. Un riferimento completo significa che il nome della tabella precede il nome della colonna.

Di seguito è riportato un esempio di una definizione di colonna calcolata che usa solo riferimenti a nomi di colonna. Le colonne **Sales** e **Cost** appartengono entrambe a una tabella denominata **Orders**.

DAX

```
Profit = [Sales] - [Cost]
```

La stessa definizione può essere riscritta con riferimenti a colonne completi.

DAX

```
Profit = Orders[Sales] - Orders[Cost]
```

In alcuni casi, tuttavia, sarà necessario usare riferimenti a colonne completi quando Power BI rileva ambiguità. Quando si immette una formula, viene generato un avviso con un messaggio di errore ondulato in rosso. Inoltre, alcune funzioni DAX come [LOOKUPVALUE](#) richiedono l'uso di colonne complete.

È consigliabile qualificare sempre completamente i riferimenti alle colonne. I motivi sono riportati nella sezione [Raccomandazioni](#).

# Misure

Una misura è un oggetto a livello di modello. Per questo motivo, i nomi delle misure devono essere univoci all'interno del modello. Tuttavia, nel riquadro **Campi** gli autori del report vedranno ogni misura associata a una singola tabella del modello. Questa associazione è impostata per motivi estetici ed è possibile configurarla impostando la proprietà **Tabella home** per la misura. Per altre informazioni, vedere [Misure in Power BI Desktop - Organizzazione delle misure](#).

È possibile usare una misura completa nelle espressioni. Anche DAX IntelliSense offre il suggerimento. Tuttavia, non è necessario e non è una procedura consigliata. Se si modifica la tabella home per una misura, qualsiasi espressione che usi un riferimento di misura completo verrà interrotta. Sarà quindi necessario modificare ogni formula interrotta per rimuovere (o aggiornare) il riferimento alla misura.

È consigliabile non qualificare mai i riferimenti alle misure. I motivi sono riportati nella sezione [Raccomandazioni](#).

## Consigli

Queste raccomandazioni sono semplici e facili da ricordare:

- Usare sempre riferimenti a colonne completi
- Non usare mai riferimenti a misure completi

Ecco perché:

- **Voce formula:** le espressioni verranno accettate, perché non saranno presenti riferimenti ambigui da risolvere. Inoltre, verranno soddisfatti i requisiti per quelle funzioni DAX che richiedono riferimenti a colonne completi.
- **Robustezza:** le espressioni continueranno a funzionare, anche quando si modifica una proprietà della tabella home della misura.
- **Leggibilità:** le espressioni saranno rapide e facili da comprendere—si determinerà rapidamente che si tratta di una colonna o di una misura, in base al fatto che sia completa o meno.

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#)
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# Confronto tra funzione DIVIDE e operatore di divisione (/)

Articolo • 05/10/2023

Quando si scrive un'espressione DAX per dividere un numeratore per un denominatore, un modeler di dati può scegliere di usare la funzione **DIVIDE** o l'operatore di divisione (/ - barra).

Quando si usa la funzione DIVIDE, è necessario passare le espressioni del numeratore e del denominatore. Facoltativamente, è possibile passare un valore che rappresenta un *risultato alternativo*.

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

La funzione DIVIDE è stata progettata per gestire automaticamente i casi di divisione per zero. Se non viene passato un risultato alternativo e il denominatore è zero o BLANK, la funzione restituisce BLANK. Quando viene passato un risultato alternativo, questo viene restituito al posto di BLANK.

La funzione DIVIDE è comoda perché consente di evitare di dover prima testare il valore del denominatore nell'espressione. La funzione è inoltre ottimizzata meglio per il test del valore del denominatore rispetto alla funzione **IF**. Il miglioramento delle prestazioni è significativo perché il controllo della divisione per zero è dispendioso. L'uso di DIVIDE consente anche di ottenere un'espressione più concisa ed elegante.

## Esempio

L'espressione di misura seguente genera una divisione sicura, ma comporta l'uso di quattro funzioni DAX.

DAX

```
Profit Margin =  
IF(  
    OR(  
        ISBLANK([Sales]),  
        [Sales] == 0  
    ),  
    BLANK(),
```

```
[Profit] / [Sales]
```

```
)
```

Questa espressione di misura ottiene lo stesso risultato, ma in modo più efficiente ed elegante.

DAX

```
Profit Margin =  
DIVIDE([Profit], [Sales])
```

## Consigli

Si consiglia di usare la funzione DIVIDE ogni volta che il denominatore è un'espressione che *potrebbe* restituire zero o BLANK.

Se il denominatore è un valore costante, è consigliabile usare l'operatore di divisione. In questo caso, la riuscita della divisione è garantita e l'espressione offrirà prestazioni migliori perché vengono evitate verifiche non necessarie.

Valutare attentamente se la funzione DIVIDE deve restituire un valore alternativo. Per le misure, la progettazione consigliata prevede in genere la restituzione di BLANK. La restituzione di BLANK è consigliata perché per impostazione predefinita gli oggetti visivi del report eliminano i raggruppamenti quando i riepiloghi sono BLANK. Questo consente all'oggetto visivo di concentrare l'attenzione sui gruppi in cui sono presenti dati. Quando necessario, in POWER BI, è possibile configurare l'oggetto visivo per visualizzare tutti i gruppi (che restituiscono valori o BLANK) nel contesto di filtro abilitando l'opzione [Mostra elementi senza dati](#).

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#)
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Usare SELECTEDVALUE invece di VALUES

Articolo • 05/10/2023

Gli autori di modelli di dati potrebbero avere a volte l'esigenza di scrivere un'espressione DAX che verifica se una colonna è filtrata in base a un valore specifico.

Nelle versioni precedenti di DAX questa esigenza poteva essere soddisfatta in modo sicuro usando un modello che coinvolge tre funzioni DAX. [IF](#), [HASONEVALUE](#) e [VALUES](#). Nella definizione di misura seguente viene presentato un esempio. Calcola l'importo dell'imposta di vendita, ma solo per le vendite effettuate ai clienti australiani.

DAX

```
Australian Sales Tax =  
IF(  
    HASONEVALUE(Customer[Country-Region]),  
    IF(  
        VALUES(Customer[Country-Region]) = "Australia",  
        [Sales] * 0.10  
    )  
)
```

In questo esempio la funzione HASONEVALUE restituisce TRUE solo quando un singolo valore della colonna **Country-Region** è visibile nel contesto di filtro corrente. Quando è TRUE, la funzione VALUES viene confrontata con il testo letterale "Australia". Quando la funzione VALUES restituisce TRUE, la misura **Sales** viene moltiplicata per 0,10 (che rappresenta il 10%). Se la funzione HASONEVALUE restituisce FALSE, poiché più di un valore filtra la colonna, la prima funzione IF restituisce BLANK.

L'uso di HASONEVALUE è una tecnica difensiva. È necessario perché è possibile che più valori filtrino la colonna **Country-Region**. In questo caso, la funzione VALUES restituisce una tabella di più righe. Il confronto tra una tabella di più righe e un valore scalare causa un errore.

## Recommendation

Si consiglia di usare la funzione [SELECTEDVALUE](#). Ottiene lo stesso risultato del modello descritto in questo articolo, ma in modo più efficiente ed elegante.

La definizione della misura di esempio viene ora riscritta usando la funzione SELECTEDVALUE.

DAX

```
Australian Sales Tax =  
IF(  
    SELECTEDVALUE(Customer[Country-Region]) = "Australia",  
    [Sales] * 0.10  
)
```

### 💡 Suggerimento

È possibile passare un valore di *risultato alternativo* nella funzione SELECTEDVALUE. Il valore del risultato alternativo viene restituito quando alla colonna non è applicato alcun filtro o sono applicati più filtri.

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
- Domande? [Contattare la community di Power BI](#) ↗
- inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#) ↗

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# Usare COUNTROWS invece di COUNT

Articolo • 05/10/2023

Gli autori di modelli di dati potrebbero avere l'esigenza di scrivere un'espressione DAX che conteggi le righe di una tabella. La tabella può essere una tabella del modello o un'espressione che restituisce una tabella.

Questa esigenza può essere soddisfatta in due modi. Per conteggiare i valori delle colonne, è possibile usare la funzione **COUNT** oppure è possibile usare la funzione **COUNTROWS** per conteggiare le righe della tabella. Entrambe le funzioni otterranno lo stesso risultato, a condizione che la colonna conteggiata non includa valori BLANK.

Nella definizione di misura seguente viene presentato un esempio. Calcola il numero di valori della colonna **OrderDate**.

DAX

```
Sales Orders =  
COUNT(Sales[OrderDate])
```

Poiché la granularità della tabella **Sales** prevede una riga per ogni ordine di vendita e la colonna **OrderDate** non contiene valori BLANK, la misura restituirà un risultato corretto.

Tuttavia, la definizione della misura seguente è una soluzione migliore.

DAX

```
Sales Orders =  
COUNTROWS(Sales)
```

Esistono tre motivi per cui la seconda definizione della misura è migliore:

- Si tratta di una soluzione più efficiente, che consente quindi di ottenere prestazioni migliori.
- Non considera i valori BLANK contenuti in alcuna colonna della tabella.
- L'intenzione della formula è più chiara, al punto di essere autodescrittiva.

## Recommendation

Quando si intende conteggiare le righe di una tabella, è consigliabile usare sempre la funzione COUNTROWS.

## Vedi anche

- Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
  - Domande? [Contattare la community di Power BI](#)
  - inviare suggerimenti, [Contribuire con idee per migliorare Power BI](#)
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# Usare le variabili per migliorare le formule DAX

Articolo • 05/10/2023

Per gli autori di modelli di dati, la scrittura e il debug di alcuni calcoli DAX possono risultare difficoltosi. È normale che i requisiti di calcolo complessi includano spesso la scrittura di espressioni composte o complesse. Le espressioni composte possono comportare l'uso di molte funzioni annidate ed eventualmente il riutilizzo della logica dell'espressione.

L'uso delle variabili nelle formule DAX consente di scrivere calcoli più complessi ed efficienti. Le variabili possono migliorare le prestazioni, l'affidabilità, la leggibilità e ridurre la complessità.

In questo articolo verranno illustrati i primi tre vantaggi usando una misura di esempio per l'incremento delle vendite rispetto all'anno precedente (YoY, Year-over-Year). La formula per calcolare l'incremento delle vendite rispetto all'anno precedente è: vendite del periodo meno vendite dello stesso periodo dell'ultimo anno, diviso per le vendite dello stesso periodo dell'ultimo anno.

Si inizierà con la definizione della misura seguente.

```
DAX  
  
Sales YoY Growth % =  
DIVIDE(  
    ([Sales] - CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12,  
MONTH))),  
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))  
)
```

La misura produce il risultato corretto, ma è possibile migliorarla, come si vedrà ora.

## Migliorare le prestazioni

Si noti che la formula ripete l'espressione che calcola "lo stesso periodo dell'ultimo anno". Questa formula non è efficiente perché Power BI deve valutare la stessa espressione due volte. La definizione della misura può essere resa più efficiente usando una variabile, [VAR](#).

La definizione della misura seguente rappresenta un miglioramento. Usa un'espressione per assegnare il risultato dello "stesso periodo dell'ultimo anno" a una variabile

denominata **SalesPriorYear**. La variabile viene quindi usata due volte nell'espressione RETURN.

DAX

```
Sales YoY Growth % =  
VAR SalesPriorYear =  
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))  
RETURN  
    DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
```

La misura continua a produrre il risultato corretto in circa metà del tempo della query.

## Migliorare la leggibilità

Nella definizione della misura precedente è possibile notare come la scelta del nome della variabile renda più facilmente comprensibile l'espressione RETURN. L'espressione è breve e auto-descrittiva.

## Semplificare il debug

Le variabili consentono anche di eseguire il debug di una formula. Per testare un'espressione assegnata a una variabile, si riscrive temporaneamente l'espressione RETURN in modo che restituisca la variabile.

La definizione della misura seguente restituisce solo la variabile **SalesPriorYear**. Si noti come imposta come commento l'espressione RETURN prevista. Questa tecnica consente di ripristinarla facilmente al termine del debug.

DAX

```
Sales YoY Growth % =  
VAR SalesPriorYear =  
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))  
RETURN  
    --DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)  
SalesPriorYear
```

## Riduci la complessità

Nelle versioni precedenti di DAX le variabili non erano ancora supportate. Erano necessarie espressioni complesse, che introducevano nuovi contesti di filtro, per usare le funzioni DAX **EARLIER** o **EARLIEST** per poter fare riferimento a contesti di filtro esterni.

Queste funzioni erano purtroppo difficili da comprendere e da usare per gli autori di modelli di dati.

Le variabili vengono sempre valutate al di fuori dei filtri applicati dall'espressione RETURN. Per questo motivo, quando si usa una variabile in un contesto di filtro modificato, si ottiene lo stesso risultato ottenuto con la funzione EARLIERST. È quindi possibile evitare l'uso delle funzioni EARLIER o EARLIERST. Ciò significa che è ora possibile scrivere formule meno complesse e più facili da comprendere.

Si consideri la definizione di colonna calcolata seguente aggiunta alla tabella **Subcategory**. Valuta una classificazione per ogni sottocategoria di prodotto in base ai valori della colonna **Subcategory Sales**.

#### DAX

```
Subcategory Sales Rank =  
COUNTRWS(  
    FILTER(  
        Subcategory,  
        EARLIER(Subcategory[Subcategory Sales]) < Subcategory[Subcategory  
Sales]  
    )  
) + 1
```

La funzione EARLIER viene usata per fare riferimento al valore della colonna **Subcategory Sales** nel *contesto della riga corrente*.

Per migliorare la definizione della colonna calcolata, è possibile usare una variabile invece della funzione EARLIER. La variabile **CurrentSubcategorySales** archivia il valore della colonna **Subcategory Sales** nel *contesto della riga corrente* e l'espressione RETURN lo usa in un contesto di filtro modificato.

#### DAX

```
Subcategory Sales Rank =  
VAR CurrentSubcategorySales = Subcategory[Subcategory Sales]  
RETURN  
    COUNTRWS(  
        FILTER(  
            Subcategory,  
            CurrentSubcategorySales < Subcategory[Subcategory Sales]  
        )  
) + 1
```

## Vedi anche

- Articolo di DAX su [VAR](#)
  - Percorso di apprendimento: [Usare DAX in Power BI Desktop](#)
  - Domande? [Contattare la community di Power BI](#)
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# Informazioni di riferimento sulle funzioni DAX

Articolo • 20/10/2023

Questo articolo contiene informazioni dettagliate, inclusi parametri, sintassi, valori restituiti ed esempi, per ognuna delle oltre 250 funzioni usate nelle formule DAX (Data Analysis Expression).

## Importante

Non tutte le funzioni DAX sono supportate o incluse nelle versioni precedenti di Power BI Desktop, Analysis Services e Power Pivot in Excel.

## In questa sezione

[Nuove funzioni DAX](#): funzioni nuove o funzioni esistenti che sono state aggiornate in modo significativo.

[Funzioni di aggregazione](#): queste funzioni calcolano un valore (scalare), ad esempio count, sum, average, minimum o maximum per tutte le righe di una colonna o di una tabella, come definito dall'espressione.

[Funzioni di data e ora](#) - Queste funzioni in DAX sono simili alle funzioni di data e ora in Microsoft Excel. Le funzioni DAX sono tuttavia basate sui tipi di dati datetime utilizzati da Microsoft SQL Server.

[Funzioni di filtro](#) - Queste funzioni consentono di restituire tipi di dati specifici, cercare valori in tabelle correlate e applicare filtri in base ai valori correlati. Le funzioni di ricerca si basano su tabelle e relazioni tra tabelle. Le funzioni di filtro consentono di modificare il contesto dei dati per creare calcoli dinamici.

[Funzioni finanziarie](#) - Queste funzioni vengono usate in formule che eseguono calcoli finanziari, ad esempio il valore attuale netto e il tasso di rendimento.

[Funzioni informative](#) - Queste funzioni esaminano una tabella o una colonna fornita come argomento a un'altra funzione e indicano se il valore corrisponde al tipo previsto. La funzione ISERROR, ad esempio, restituisce TRUE se il valore a cui si fa riferimento contiene un errore.

**Funzioni logiche** - Queste funzioni restituiscono informazioni sui valori in un'espressione. La funzione TRUE, ad esempio, consente di sapere se un'espressione che si sta valutando restituirà un valore TRUE.

**Funzioni matematiche e trigonometriche** - Le funzioni matematiche in DAX sono simili alle funzioni matematiche e trigonometriche di Excel. Ci sono tuttavia alcune piccole differenze per quanto riguarda i tipi di dati numerici usati dalle funzioni DAX.

**Altre funzioni** - Queste funzioni eseguono azioni uniche che non possono essere definite in base a nessuna delle categorie a cui appartiene la maggior parte delle altre funzioni.

**Funzioni padre e figlio:** queste funzioni consentono agli utenti di gestire i dati presentati come gerarchia padre/figlio nei modelli di dati.

**Funzioni di relazione** - Queste funzioni sono destinate alla gestione e all'uso delle relazioni tra le tabelle. È ad esempio possibile specificare una determinata relazione da usare in un calcolo.

**Funzioni statistiche:** calcolano i valori correlati a distribuzioni statistiche e probabilità, ad esempio deviazione standard e numero di permutazioni.

**Funzioni di modifica delle tabelle** - Queste funzioni restituiscono una tabella o modificano le tabelle esistenti.

**Funzioni di testo** - Con queste funzioni è possibile restituire parte di una stringa, cercare testo all'interno di una stringa o concatenare valori stringa. Sono inoltre disponibili funzioni per il controllo dei formati per date, ore e numeri.

**Funzioni di Business Intelligence per le gerarchie temporali** - Queste funzioni consentono di creare calcoli basati su informazioni predefinite relative a calendari e date. Tramite gli intervalli di ore e date in combinazione con aggregazioni o calcoli è possibile compilare confronti significativi tra periodi di tempo paragonabili relativamente a vendite, scorte e così via.

## Vedi anche

[Informazioni di riferimento sulla sintassi DAX](#)

[Informazioni di riferimento sugli operatori DAX](#)

[Convenzioni di denominazione dei parametri DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Nuove funzioni DAX

Articolo • 20/10/2023

DAX viene costantemente migliorato con nuove funzioni e funzionalità per supportare le nuove caratteristiche. Le funzioni nuove e aggiornate sono incluse negli aggiornamenti di servizi, applicazioni e strumenti, che nella maggior parte dei casi sono mensili.

Mentre le funzioni e le funzionalità vengono aggiornate continuamente, nella documentazione sono descritti solo gli aggiornamenti che hanno una modifica visibile e funzionale esposta agli utenti. Le nuove funzioni e gli aggiornamenti alle funzioni esistenti dell'anno precedente sono illustrati di seguito.

## ⓘ Importante

Non tutte le funzioni sono supportate in tutte le versioni di Power BI Desktop, Analysis Services e Power Pivot in Excel. Le funzioni nuove e aggiornate vengono in genere introdotte in Power BI Desktop e successivamente in Analysis Services, Power Pivot in Excel e negli strumenti.

## Nuove funzioni

Funzione	Mese	Descrizione
<a href="#">MATCHBY</a>	Maggio 2023	Definire le colonne utilizzate per trovare le corrispondenze con i dati e identificare la riga corrente in un'espressione di funzione finestra.
<a href="#">RANK</a>	Aprile 2023	Restituisce la classificazione per il contesto corrente all'interno della partizione specificata, ordinata in base all'ordine specificato.
<a href="#">ROWNUMBER</a>	Aprile 2023	Restituisce la classificazione per il contesto corrente all'interno della partizione specificata, ordinata in base all'ordine specificato.
<a href="#">LINEST</a>	Febbraio 2023	Usa il metodo dei minimi quadrati (Least Squares) per calcolare la linea retta che meglio corrisponde ai dati specificati.
<a href="#">LINESTX</a>	Febbraio 2023	Usa il metodo dei minimi quadrati (Least Squares) per calcolare la linea retta che meglio corrisponde ai dati specificati. Il risultato dei dati dalle espressioni valutate per ogni riga di una tabella.

<b>Funzione</b>	<b>Mese</b>	<b>Descrizione</b>
INDEX	Dicembre 2022	Restituisce una riga in una posizione assoluta, specificata dal parametro di posizione, all'interno della partizione specificata, ordinata in base all'ordine specificato o all'asse specificato.
OFFSET	Dicembre 2022	Restituisce una singola riga posizionata prima o dopo la <i>riga corrente</i> all'interno della stessa tabella da un offset specificato.
ORDERBY	Dicembre 2022	Definisce le colonne che determinano l'ordinamento all'interno di ognuna delle partizioni di una funzione finestra.
PARTITIONBY	Dicembre 2022	Definisce le colonne utilizzate per partizionare il parametro di <relazione> di una funzione finestra.
WINDOW	Dicembre 2022	Restituisce più righe posizioionate all'interno dell'intervallo specificato.
EVALUATEANDLOG	Novembre 2022	Restituisce il valore del primo argomento e lo registra in un evento del profiler di valutazione DAX.
TOCSV	Novembre 2022	Restituisce una tabella come stringa in formato CSV. Questa funzione si applica solo a Power BI Desktop.
TOJSON	Novembre 2022	Restituisce una tabella come stringa in formato JSON. Questa funzione si applica solo a Power BI Desktop.
NETWORKDAYS	Luglio 2022	Restituisce il numero di giorni lavorativi interi tra due date.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# Funzione di aggregazione

Articolo • 20/10/2023

Le funzioni di aggregazione calcolano un valore (scalare) come il numero, la somma, la media, il minimo o il massimo per tutte le righe di una colonna o di una tabella, come definito dall'espressione.

## In questa categoria

Funzione	Descrizione
<a href="#">APPROXIMATEDISTINCTCOUNT</a>	Restituisce un conteggio <i>stimato</i> dei valori univoci in una colonna.
<a href="#">AVERAGE</a>	Restituisce la media (aritmetica) di tutti i numeri presenti in una colonna.
<a href="#">AVERAGEA</a>	Restituisce la media (media aritmetica) di tutti i valori di una colonna.
<a href="#">AVERAGEX</a>	Calcola la media (aritmetica) di un set di espressioni valutate su una tabella.
<a href="#">COUNT</a>	Conta il numero di righe in una colonna specifica contenenti valori non vuoti.
<a href="#">COUNTA</a>	Conta il numero di righe in una colonna specifica contenenti valori non vuoti.
<a href="#">COUNTAX</a>	Conta i risultati non vuoti quando viene valutato il risultato di un'espressione su una tabella.
<a href="#">COUNTBLANK</a>	Conta il numero di celle vuote presenti in una colonna.
<a href="#">COUNTROWS</a>	Conta il numero di righe nella tabella specificata o in una tabella definita da un'espressione.
<a href="#">COUNTX</a>	Conta il numero di righe che contengono un numero o un'espressione che restituisce un numero quando si valuta un'espressione in una tabella.
<a href="#">DISTINCTCOUNT</a>	Conta il numero di valori distinti di una colonna.
<a href="#">DISTINCTCOUNTNOBLANK</a>	Conta il numero di valori distinti di una colonna.
<a href="#">MAX</a>	Restituisce il valore numerico più grande in una colonna o tra due espressioni scalari.

Funzione	Descrizione
MAXA	Restituisce il valore più grande in una colonna.
MAXX	Valuta un'espressione per ogni riga di una tabella e restituisce il valore numerico più grande.
MIN	Restituisce il valore numerico più piccolo in una colonna o tra due espressioni scalari.
MINA	Restituisce il valore più piccolo in una colonna, inclusi i valori logici e i numeri rappresentati come testo.
MINX	Restituisce il valore numerico più piccolo risultante dalla valutazione di un'espressione per ogni riga di una tabella.
PRODUCT	Restituisce il prodotto dei numeri in una colonna.
PRODUCTX	Restituisce il prodotto di un'espressione valutata per ogni riga di una tabella.
SUM	Somma tutti i numeri in una colonna.
SUMX	Restituisce la somma di un'espressione valutata per ogni riga di una tabella.

## Commenti e suggerimenti

Questa pagina è stata utile?



# APPROXIMATEDISTINCTCOUNT

Articolo • 20/10/2023

Restituisce un conteggio *stimato* dei valori univoci in una colonna. Questa funzione richiama un'operazione di aggregazione corrispondente nell'origine dati, ottimizzata per le prestazioni delle query, ma con una precisione leggermente ridotta. Questa funzione può essere usata con le origini dati seguenti: Azure SQL, pool SQL dedicato di Azure Synapse, BigQuery, Databricks e Snowflake. Questa funzione richiede la modalità DirectQuery. La modalità di importazione e la modalità di archiviazione doppia non sono supportate.

## Sintassi

DAX

```
APPROXIMATEDISTINCTCOUNT(<columnName>)
```

## Parametri

Termine	Descrizione
colonna	Colonna contenente i valori da contare. Non può essere un'espressione.

## Valore restituito

Numero approssimativo di valori distinti in *column*.

## Osservazioni:

L'unico argomento per questa funzione è una colonna. È possibile usare colonne contenenti qualsiasi tipo di dati. Quando la funzione non trova righe da contare, restituisce un valore BLANK. In caso contrario, restituisce il conteggio dei valori distinti.

## Commenti e suggerimenti

Questa pagina è stata utile?

Si

No

# AVERAGE

Articolo • 20/10/2023

Restituisce la media (aritmetica) di tutti i numeri presenti in una colonna.

## Sintassi

DAX

`AVERAGE(<column>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i numeri dei quali calcolare la media.

## Valore restituito

Restituisce un numero decimale che rappresenta la media aritmetica dei numeri nella colonna.

## Osservazioni:

- Questa funzione accetta la colonna specificata come argomento e trova la media dei valori in questa colonna. Se si vuole trovare la media di un'espressione che restituisce un set di numeri, usare invece la funzione AVERAGEX.
- I valori non numerici nella colonna vengono gestiti come indicato di seguito:
  - Se la colonna contiene testo, non è possibile eseguire alcuna aggregazione e le funzioni restituiscono valori vuoti.
  - Se la colonna contiene valori logici o celle vuote, tali valori vengono ignorati.
  - Le celle con valore zero sono incluse.
- Quando si calcola la media di celle, è necessario tenere presente la differenza tra una cella vuota e una cella che contiene il valore 0 (zero). Quando una cella contiene 0, viene aggiunta alla somma di numeri e la riga viene conteggiata tra il numero di righe usato come divisore. Quando invece una cella contiene un valore vuoto, la riga non viene conteggiata.

- Ogni volta che non vi sono righe da aggregare, la funzione restituisce un valore vuoto. Tuttavia, se sono presenti righe, ma nessuna soddisfa i criteri specificati, la funzione restituisce 0. Anche Excel restituisce zero se non viene trovata alcuna riga che soddisfa le condizioni.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente restituisce la media dei valori della colonna ExtendedSalesAmount nella tabella InternetSales.

DAX

```
= AVERAGE(InternetSales[ExtendedSalesAmount])
```

## Funzioni correlate

La funzione AVERAGEX può accettare come argomento un'espressione valutata per ogni riga di una tabella. In questo modo è possibile eseguire calcoli e quindi ottenere la media dei valori calcolati.

La funzione AVERAGEA accetta una colonna come argomento, ma per gli altri aspetti è analoga alla funzione corrispondente di Excel. Usando la funzione AVERAGEa è possibile calcolare una media in una colonna che contiene valori vuoti.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# AVERAGEA

Articolo • 20/10/2023

Restituisce la media (media aritmetica) di tutti i valori di una colonna. Gestisce testo e valori non numerici.

## Sintassi

DAX

`AVERAGEA(<column>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori per i quali calcolare la media.

## Valore restituito

Numero decimale.

## Osservazioni:

- La funzione AVERAGEA accetta una colonna e calcola la media dei numeri in essa contenuti, ma gestisce anche i tipi di dati non numerici in base alle regole seguenti:
  - I valori che restituiscono TRUE vengono conteggiati come 1.
  - I valori che restituiscono FALSE vengono conteggiati come 0 (zero).
  - I valori che contengono testo non numerico vengono conteggiati come 0 (zero).
  - Il testo vuoto ("") viene conteggiato come 0 (zero).
- Se non si vogliono includere valori logici e rappresentazioni testuali di numeri in un riferimento come parte del calcolo, usare la funzione AVERAGE.
- Ogni volta che non vi sono righe da aggregare, la funzione restituisce un valore vuoto. Tuttavia, se sono presenti righe, ma nessuna soddisfa i criteri specificati, la funzione restituisce 0. Anche Microsoft Excel restituisce zero se non viene trovata alcuna riga che soddisfa le condizioni.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio che segue restituisce la media delle celle non vuote nella colonna a cui si fa riferimento, in base alla tabella seguente. Usando la funzione AVERAGE la media sarebbe 21/2, mentre con la funzione AVERAGEA il risultato è 22/5.

ID transazione	Importo	Risultato
0000123	1	Conteggiato come 1
0000124	20	Conteggiato come 20
0000125	n/d	Conteggiato come 0
0000126		Conteggiato come 0
0000126	TRUE	Conteggiato come 1

DAX

```
= AVERAGEA([Amount])
```

## Vedi anche

[funzione AVERAGE](#)

[Funzione AVERAGEX](#)

[Funzioni statistiche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**    **No**

# AVERAGEX

Articolo • 20/10/2023

Calcola la media (aritmetica) di un set di espressioni valutate su una tabella.

## Sintassi

DAX

```
AVERAGEX(<table>,<expression>)
```

## Parametri

Termine	Definizione
table	Nome di una tabella o espressione che specifica la tabella in cui può essere eseguita l'aggregazione.
expression	Espressione con un risultato scalare, che verrà valutata per ogni riga della tabella nel primo argomento.

## Valore restituito

Numero decimale.

## Osservazioni:

- La funzione AVERAGEX consente di valutare le espressioni per ogni riga di una tabella, quindi di accettare il set risultante di valori e calcolarne la media aritmetica. La funzione accetta pertanto una tabella come primo argomento e un'espressione come secondo argomento.
- Per tutti gli altri aspetti, AVERAGEX segue le stesse regole di AVERAGE. Non è possibile includere celle non numeriche o Null. Entrambi gli argomenti, table ed expression, sono obbligatori.
- Quando non vi sono righe da aggregare, la funzione restituisce un valore vuoto. Quando sono presenti righe, ma nessuna soddisfa i criteri specificati, la funzione restituisce 0.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene calcolato il costo medio di spedizione e imposte per ogni ordine nella tabella InternetSales, sommando innanzitutto i valori di Freight e TaxAmt in ogni riga e, successivamente, calcolando la media di queste somme.

DAX

```
= AVERAGEX(InternetSales, InternetSales[Freight]+ InternetSales[TaxAmt])
```

Se si usano più operazioni nell'espressione usata come secondo argomento, è necessario usare le parentesi per controllare l'ordine dei calcoli. Per altre informazioni, vedere i [Riferimenti per la sintassi DAX](#).

## Vedi anche

[funzione AVERAGE](#)

[funzione AVERAGEA](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNT

Articolo • 20/10/2023

Conta il numero di righe nella colonna specifica che contengono valori non vuoti.

## Sintassi

DAX

```
COUNT(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori da contare.

## Valore restituito

Numero intero.

## Osservazioni:

- L'unico argomento consentito per questa funzione è una colonna. La funzione COUNT conta le righe che contengono i tipi di valori seguenti:
  - Numeri
  - Date
  - Stringhe
- Quando la funzione non trova righe da contare, restituisce un valore vuoto.
- I valori vuoti vengono ignorati. I valori TRUE/FALSE non sono supportati.
- Per valutare una colonna di valori TRUE/FALSE, usare la funzione COUNTA.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- Per le procedure consigliate quando si usa COUNT, vedere [Usare COUNTROWS invece di COUNT](#).

# Esempio

L'esempio seguente mostra come contare il numero di valori nella colonna ShipDate.

DAX

```
= COUNT([ShipDate])
```

Per contare testo o valori logici, usare la funzione COUNTA o COUNTAX.

## Vedi anche

[Funzione COUNTA](#)

[Funzione COUNTAX](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNTA

Articolo • 20/10/2023

Conta il numero di righe in una colonna specifica contenenti valori non vuoti.

## Sintassi

DAX

```
COUNTA(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori da contare.

## Valore restituito

Numero intero.

## Osservazioni:

- Quando non trova righe da contare, la funzione restituisce un valore vuoto.
- A differenza di [COUNT](#), COUNTA supporta il tipo di dati booleani.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce tutte le righe nella tabella `Reseller` in cui è presente qualsiasi tipo di valore nella colonna per l'archiviazione dei numeri di telefono.

DAX

```
= COUNTA(Reseller[Phone])
```

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTAX](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNTAX

Articolo • 20/10/2023

La funzione COUNTAX conta i risultati non vuoti quando viene valutato il risultato di un'espressione su una tabella. Il funzionamento è quindi uguale a quello della funzione COUNTA, ma questa funzione viene usata per scorrere le righe di una tabella e contare quelle in cui le espressioni specificate restituiscono un risultato non vuoto.

## Sintassi

DAX

```
COUNTAX(<table>,<expression>)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.

## Valore restituito

Numero intero.

## Osservazioni:

- Analogamente alla funzione COUNTA, la funzione COUNTAX conta le celle contenenti qualsiasi tipo di informazioni, incluse altre espressioni. Ad esempio, se la colonna contiene un'espressione che restituisce una stringa vuota, la funzione COUNTAX tratta tale risultato come valore non vuoto. In genere la funzione COUNTAX non conta le celle vuote, ma in questo caso la cella contiene una formula, quindi viene contata.
- Ogni volta che la funzione non trova righe da aggregare, restituisce un valore vuoto.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene contato il numero di righe non vuote nella colonna Phone, usando la tabella risultante dall'applicazione del filtro [Status] = **Active** alla tabella Reseller.

DAX

```
= COUNTA(FILTER('Reseller',[Status]="Active"),[Phone])
```

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTA](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNTBLANK

Articolo • 20/10/2023

Conta il numero di celle vuote presenti in una colonna.

## Sintassi

DAX

`COUNTBLANK(<column>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente le celle vuote da contare.

## Valore restituito

Numero intero. Se non viene trovata alcuna riga che soddisfa la condizione, vengono restituiti valori vuoti.

## Osservazioni:

- L'unico argomento consentito per questa funzione è una colonna. È possibile usare colonne contenenti qualsiasi tipo di dati, ma vengono contate solo le celle vuote. Le celle con valore zero (0) non vengono contate, perché lo zero è considerato un valore numerico e non un valore vuoto.
- Ogni volta che non vi sono righe da aggregare, la funzione restituisce un valore vuoto. Tuttavia, se sono presenti righe, ma nessuna soddisfa i criteri specificati, la funzione restituisce 0. Anche Microsoft Excel restituisce zero se non viene trovata alcuna riga che soddisfa le condizioni.
- In altre parole, se la funzione COUNTBLANK non trova celle vuote, il risultato sarà zero, ma se non sono presenti righe da controllare, il risultato sarà un valore vuoto.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

L'esempio seguente illustra come contare il numero di righe nella tabella Reseller in cui sono presenti valori vuoti per BankName.

DAX

```
= COUNTBLANK(Reseller[BankName])
```

Per contare testo o valori logici, usare la funzione COUNTA o COUNTAX.

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTA](#)

[Funzione COUNTAX](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNTROWS

Articolo • 20/10/2023

La funzione COUNTROWS conta il numero di righe nella tabella specificata o in una tabella definita da un'espressione.

## Sintassi

DAX

```
COUNTROWS([<table>])
```

## Parametri

Termine	Definizione
table	(Facoltativo) Nome della tabella che contiene le righe da contare, oppure un'espressione che restituisce una tabella. Se non specificato, il valore predefinito è la tabella home dell'espressione corrente.

## Valore restituito

Numero intero.

## Osservazioni:

- Questa funzione può essere usata per contare il numero di righe in una tabella di base, ma più spesso viene usata per contare il numero di righe risultante dall'applicazione di un filtro a una tabella o dall'applicazione del contesto a una tabella.
- Ogni volta che non vi sono righe da aggregare, la funzione restituisce un valore vuoto. Tuttavia, se sono presenti righe, ma nessuna soddisfa i criteri specificati, la funzione restituisce 0. Anche Microsoft Excel restituisce zero se non viene trovata alcuna riga che soddisfa le condizioni.
- Per ulteriori informazioni sulle procedure consigliate nell'uso di COUNT e COUNTROWS, vedere [Usare COUNTROWS invece di COUNT in DAX](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

L'esempio seguente mostra come contare il numero di righe nella tabella Orders. Il risultato previsto è 52761.

```
DAX
= COUNTROWS('Orders')
```

## Esempio 2

L'esempio seguente dimostra come usare COUNTROWS con un contesto di riga. In questo scenario vi sono due set di dati correlati dal numero d'ordine. La tabella Reseller contiene una riga per ogni rivenditore, la tabella ResellerSales contiene più righe per ogni ordine e ogni riga contiene un ordine per un rivenditore specifico. Le tabelle sono connesse da una relazione nella colonna ResellerKey.

La formula ottiene il valore di ResellerKey, quindi conta il numero di righe nella tabella correlata che hanno lo stesso valore relativo all'ID del rivenditore. Il risultato viene restituito nella colonna **CalculatedColumn1**.

```
DAX
= COUNTROWS(RELATEDTABLE(ResellerSales))
```

La tabella seguente mostra una parte dei risultati previsti:

ResellerKey	CalculatedColumn1
1	73
2	70
3	394

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTA](#)

[Funzione COUNTAX](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUNTX

Articolo • 20/10/2023

Conta il numero di righe che contengono un valore non vuoto o un'espressione che restituisce un valore non vuoto quando si valuta un'espressione in una tabella.

## Sintassi

DAX

`COUNTX(<table>, <expression>)`

## Parametri

Termine	Definizione
table	Tabella contenente le righe da contare.
expression	Espressione che restituisce il set di valori che contiene i valori da contare.

## Valore restituito

Valore intero.

## Osservazioni:

- La funzione COUNTX accetta due argomenti. Il primo argomento deve essere sempre una tabella o qualsiasi espressione che restituisca una tabella. Il secondo argomento è la colonna o l'espressione in cui COUNTX esegue la ricerca.
- La funzione COUNTX conta solo valori, date o stringhe. Se la funzione non trova righe da contare, restituisce un valore vuoto.
- Se si vogliono contare valori logici, usare la funzione COUNTAX.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

La formula seguente formula restituisce il conteggio di tutte le righe della tabella Product che contengono un prezzo di listino.

DAX

```
= COUNTX(Product,[ListPrice])
```

## Esempio 2

La formula seguente illustra come passare una tabella filtrata a COUNTX come primo argomento. La formula usa un'espressione di filtro per ottenere solo le righe della tabella Product che soddisfano la condizione, ProductSubCategory = "Caps", quindi conta le righe nella tabella risultante che includono un prezzo di listino. L'espressione FILTER si applica alla tabella Product, ma usa un valore cercato nella tabella correlata, ProductSubCategory.

DAX

```
=  
COUNTX(FILTER(Product,RELATED(ProductSubcategory[EnglishProductSubcategoryNa  
me])="Caps"), Product[ListPrice])
```

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTA](#)

[Funzione COUNTAX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# DISTINCTCOUNT

Articolo • 20/10/2023

Conta il numero di valori distinti di una colonna.

## Sintassi

DAX

```
DISTINCTCOUNT(<column>)
```

## Parametri

Termine	Descrizione
colonna	Colonna contenente i valori da contare

## Valore restituito

Numero di valori distinti in *column*.

## Osservazioni:

- L'unico argomento consentito per questa funzione è una colonna. È possibile usare colonne contenenti qualsiasi tipo di dati. Quando la funzione non trova righe da contare, restituisce un valore BLANK. In caso contrario, restituisce il conteggio dei valori distinti.
- La funzione DISTINCTCOUNT conta il valore BLANK. Per ignorare il valore BLANK, usare la funzione [DISTINCTCOUNTNOBLANK](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente mostra come contare il numero di ordini di vendita distinti nella colonna ResellerSales\_USD[SalesOrderNumber].

DAX

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

L'uso della misura sopra indicata in una tabella con l'anno di calendario visualizzato di lato e la categoria di prodotto in alto restituisce i risultati seguenti:

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Componenti</b>	-	<b>Grand Total</b>
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521
2008	293	724	561	601		894
					1	1
<b>Grand Total</b>	<b>1315</b>	<b>3153</b>	<b>2410</b>	<b>2646</b>	<b>1</b>	<b>3797</b>

Nell'esempio precedente è possibile notare che i numeri delle righe Grand Total non vengono sommati in quanto nello stesso ordine potrebbero essere contenute voci provenienti da categorie di prodotto diverse.

## Vedi anche

[Funzione COUNT](#)

[Funzione COUNTA](#)

[Funzione COUNTAX](#)

[Funzione COUNTX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**    **No**

# DISTINCTCOUNTNOBLANK

Articolo • 20/10/2023

Conta il numero di valori distinti di una colonna.

## Sintassi

DAX

```
DISTINCTCOUNTNOBLANK (<column>)
```

## Parametri

Termine	Descrizione
colonna	Colonna contenente i valori da contare

## Valore restituito

Numero di valori distinti in *column*.

## Osservazioni:

- A differenza della funzione [DISTINCTCOUNT](#), DISTINCTCOUNTNOBLANK non include il valore BLANK.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente mostra come contare il numero di ordini di vendita distinti nella colonna ResellerSales\_USD[SalesOrderNumber].

DAX

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

## Query DAX

DAX

```
EVALUATE  
ROW(  
    "DistinctCountNoBlank", DISTINCTCOUNTNOBLANK(DimProduct[EndDate]),  
    "DistinctCount", DISTINCTCOUNT(DimProduct[EndDate])  
)
```

[DistinctCountNoBlank]	[DistinctCount]
2	3

## Vedi anche

[DISTINCTCOUNT](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# MAX

Articolo • 20/10/2023

Restituisce il valore più grande in una colonna o tra due espressioni scalari.

## Sintassi

DAX

```
MAX(<column>)
```

DAX

```
MAX(<expression1>, <expression2>)
```

## Parametri

Termine	Definizione
colonna	Colonna in cui trovare il valore più grande.
expression	Qualsiasi espressione DAX che restituisce un singolo valore.

## Valore restituito

Mostra il valore più alto.

## Osservazioni:

- Nel confronto di due espressioni un valore vuoto viene considerato uguale a 0. Questo significa che Max(1,Blank() ) restituisce 1 e Max( -1, Blank() ) restituisce 0. Se entrambi gli argomenti sono vuoti, MAX restituisce un valore vuoto. Se una delle due espressioni restituisce un valore non consentito, MAX restituisce un errore.
- I valori TRUE/FALSE non sono supportati. Per valutare una colonna di valori TRUE/FALSE, usare la funzione MAXA.

## Esempio 1

Nell'esempio seguente viene restituito il valore più grande trovato nella colonna ExtendedAmount della tabella InternetSales.

DAX

```
= MAX(InternetSales[ExtendedAmount])
```

## Esempio 2

Nell'esempio seguente viene restituito il valore più grande del risultato di due espressioni.

DAX

```
= Max([TotalSales], [TotalPurchases])
```

## Vedi anche

[Funzione MAXA](#)

[Funzione MAXX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MAXA

Articolo • 20/10/2023

Restituisce il valore più grande in una colonna.

## Sintassi

DAX
<code>MAXA(&lt;column&gt;)</code>

## Parametri

Termine	Definizione
colonna	Colonna in cui trovare il valore più grande.

## Valore restituito

Mostra il valore più alto.

## Osservazioni:

- La funzione MAXA accetta come argomento una colonna e cerca il valore più grande tra i tipi di valore seguenti:
  - Numeri
  - Date
- Valori logici, come TRUE e FALSE. Le righe che restituiscono TRUE contano come 1, quelle che restituiscono FALSE come 0 (zero).
- Le celle vuote vengono ignorate. Se la colonna non contiene valori utilizzabili, MAXA restituisce 0 (zero).
- Per confrontare valori di testo, usare la funzione MAX.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Nell'esempio seguente viene restituito il valore più grande di una colonna calcolata, denominata **ResellerMargin**, che calcola la differenza tra il prezzo di listino e il prezzo del rivenditore.

DAX

```
= MAXA([ResellerMargin])
```

## Esempio 2

Nell'esempio seguente viene restituito il valore più grande di una colonna contenente date e ore. Questa formula pertanto ottiene la data della transazione più recente.

DAX

```
= MAXA([TransactionDate])
```

## Vedi anche

[Funzione MAX](#)

[Funzione MAXX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MAXX

Articolo • 20/10/2023

Restituisce il valore più grande risultante dalla valutazione di un'espressione per ogni riga di una tabella.

## Sintassi

DAX

```
MAXX(<table>,<expression>,[<variant>])
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.
variante	(Facoltativo) Se TRUE e se sono presenti tipi di valori varianti o misti, viene restituito il valore più alto basato su ORDER BY DESC.

## Valore restituito

Il valore massimo.

## Osservazioni:

- L'argomento **table** della funzione MAXX può essere un nome di tabella o un'espressione che restituisce una tabella. Il secondo argomento indica l'espressione da valutare per ogni riga della tabella.
- Tra i valori da valutare, vengono contati solo i seguenti:
  - Numeri
  - Testi
  - Date
- I valori vuoti vengono ignorati. I valori TRUE/FALSE non sono supportati.

- Se l'espressione ha tipi varianti o valori misti, ad esempio testo e numeri, per impostazione predefinita MAXX considera solo i numeri. Se <variant> = TRUE, viene restituito il valore massimo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

La formula seguente usa un'espressione come secondo argomento per calcolare la quantità totale di imposte e spese di spedizione per ogni ordine nella tabella InternetSales. Il risultato previsto è 375.7184.

DAX

```
= MAXX(InternetSales, InternetSales[TaxAmt]+ InternetSales[Freight])
```

## Esempio 2

La formula seguente filtra innanzitutto la tabella InternetSales, tramite un'espressione FILTER, per restituire un subset di ordini per un'area di vendita specifica, definita come [SalesTerritory] = 5. La funzione MAXX quindi valuta l'espressione usata come secondo argomento per ogni riga della tabella filtrata e restituisce l'importo più elevato relativo a imposte e spese di spedizione solo per tali ordini. Il risultato previsto è 250.3724.

DAX

```
= MAXX(FILTER(InternetSales,[SalesTerritoryCode]="5"),  
InternetSales[TaxAmt]+ InternetSales[Freight])
```

## Vedi anche

[Funzione MAX](#)

[Funzione MAXA](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# MIN

Articolo • 20/10/2023

Restituisce il valore più piccolo in una colonna o tra due espressioni scalari.

## Sintassi

DAX

```
MIN(<column>)
```

DAX

```
MIN(<expression1>, <expression2>)
```

## Parametri

Termine	Definizione
colonna	Colonna in cui si vuole trovare il valore più piccolo.
expression	qualsiasi espressione DAX che restituisce un singolo valore.

## Valore restituito

Il valore più piccolo.

## Osservazioni:

- La funzione MIN accetta una colonna o due espressioni come argomento e restituisce il valore più piccolo. Vengono contati i tipi di valori seguenti nelle colonne:
  - Numeri
  - Testi
  - Date
  - Valori vuoti
- Nel confronto di espressioni un valore vuoto viene considerato uguale a 0. Questo significa che  $\text{Min}(1, \text{Blank}())$  restituisce 0 e  $\text{Min}(-1, \text{Blank}())$  restituisce -1. Se

entrambi gli argomenti sono vuoti, MIN restituisce un valore vuoto. Se una delle due espressioni restituisce un valore non consentito, MIN restituisce un errore.

- I valori TRUE/FALSE non sono supportati. Per valutare una colonna di valori TRUE/FALSE, usare la funzione MINA.

## Esempio 1

Nell'esempio seguente viene restituito il valore più piccolo della colonna calcolata ResellerMargin.

DAX

```
= MIN([ResellerMargin])
```

## Esempio 2

Nell'esempio seguente viene restituito il valore più piccolo di una colonna TransactionDate contenente date e ore. Questa formula restituisce pertanto la data che ricorre per prima.

DAX

```
= MIN([TransactionDate])
```

## Esempio 3

Nell'esempio seguente viene restituito il valore più piccolo del risultato di due espressioni scalari.

DAX

```
= Min([TotalSales], [TotalPurchases])
```

## Vedi anche

[Funzione MINA](#)

[Funzione MINX](#)

[Funzioni statistiche](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# MINA

Articolo • 20/10/2023

Restituisce il valore più piccolo in una colonna.

## Sintassi

DAX

```
MINA(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna per cui trovare il valore minimo.

## Valore restituito

Il valore più piccolo.

## Osservazioni:

- La funzione MINA accetta come argomento una colonna contenente numeri e determina il valore più piccolo come indicato di seguito:
  - Se la colonna non contiene valori, MINA restituisce 0 (zero).
  - Le righe nella colonna che restituiscono valori logici, come TRUE e FALSE, vengono trattate come 1 se TRUE e come 0 (zero) se FALSE.
  - Le celle vuote vengono ignorate.
- Per confrontare valori di testo, usare la funzione MIN.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

L'espressione seguente restituisce le spese di spedizione minime della tabella InternetSales.

DAX

```
= MINA(InternetSales[Freight])
```

## Esempio 2

L'espressione seguente restituisce il valore minimo della colonna PostalCode. Poiché il tipo di dati della colonna è testo, la funzione non trova alcun valore e la formula restituisce zero (0).

DAX

```
= MINA([PostalCode])
```

## Vedi anche

[Funzione MIN](#)

[Funzione MINX](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MINX

Articolo • 20/10/2023

Restituisce il valore più basso risultante dalla valutazione di un'espressione per ogni riga di una tabella.

## Sintassi

DAX

```
MINX(<table>, < expression>,[<variant>])
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.
variante	(Facoltativo) Se TRUE e se sono presenti tipi varianti o valori misti, viene restituito il valore più basso basato su ORDER BY ASC.

## Valore restituito

Il valore minimo.

## Osservazioni:

- La funzione MINX accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento contiene l'espressione valutata per ogni riga della tabella.
- I valori vuoti vengono ignorati. I valori TRUE/FALSE non sono supportati.
- Se l'espressione ha tipi varianti o valori misti, ad esempio testo e numeri, per impostazione predefinita MINX considera solo i numeri. Se `<variant> = TRUE`, viene restituito il valore minimo.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Nell'esempio seguente viene filtrata la tabella InternetSales e vengono restituite solo le righe per un territorio di vendita specifico. La formula trova quindi il valore minimo nella colonna Freight.

DAX

```
= MINX( FILTER(InternetSales, [SalesTerritoryKey] = 5),[Freight])
```

## Esempio 2

Nell'esempio seguente viene usata la stessa tabella filtrata dell'esempio precedente, ma invece di cercare soltanto i valori nella colonna per ogni riga della tabella filtrata, la funzione calcola la somma di due colonne, Freight e TaxAmt, e restituisce il valore più basso risultante dal calcolo.

DAX

```
= MINX( FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5),  
InternetSales[Freight] + InternetSales[TaxAmt])
```

Nel primo esempio i nomi delle colonne sono non qualificati. Nel secondo esempio i nomi delle colonne sono completi.

## Vedi anche

[Funzione MIN](#)

[Funzione MINA](#)

[Funzioni statistiche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# PRODUCT

Articolo • 20/10/2023

Restituisce il prodotto dei numeri in una colonna.

## Sintassi

DAX

```
PRODUCT(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna contenente i numeri per i quali deve essere calcolato il prodotto.

## Valore restituito

Numero decimale.

## Osservazioni:

- Per restituire il prodotto di un'espressione valutata per ogni riga di una tabella, usare la [funzione PRODUCTX](#).
- Vengono conteggiati solo i numeri nella colonna. I valori vuoti, logici e di testo vengono ignorati. Ad esempio,

`PRODUCT( Table[Column] )` è pari a `PRODUCTX( Table, Table[Column] )`.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente calcola il prodotto della colonna AdjustedRates in una tabella Annuity:

DAX

```
= PRODUCT( Annuity[AdjustedRates] )
```

## Vedi anche

[PRODUCTX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PRODUCTX

Articolo • 20/10/2023

Restituisce il prodotto di un'espressione valutata per ogni riga di una tabella.

## Sintassi

DAX

```
PRODUCTX(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.

## Valore restituito

Numero decimale.

## Osservazioni:

- Per restituire il prodotto dei numeri in una colonna, usare [PRODUCT](#).
- La funzione PRODUCTX accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento è una colonna che contiene i numeri per i quali si vuole calcolare il prodotto o un'espressione che restituisce una colonna.
- Vengono conteggiati solo i numeri nella colonna. I valori vuoti, logici e di testo vengono ignorati.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente calcola il valore futuro di un investimento:

DAX

```
= [PresentValue] * PRODUCTX( AnnuityPeriods, 1+[FixedInterestRate] )
```

## Vedi anche

[PRODUCT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# SUM

Articolo • 20/10/2023

Somma tutti i numeri in una colonna.

## Sintassi

DAX

```
SUM(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna contenente i numeri da sommare.

## Valore restituito

Numero decimale.

## Osservazioni:

Per filtrare i valori da sommare, è possibile usare la funzione SUMX e specificare un'espressione da sommare.

## Esempio

L'esempio seguente aggiunge tutti i numeri contenuti nella colonna Amt della tabella Sales.

DAX

```
= SUM(Sales[Amt])
```

## Vedi anche

## Commenti e suggerimenti

Questa pagina è stata utile?



# SUMX

Articolo • 20/10/2023

Restituisce la somma di un'espressione valutata per ogni riga di una tabella.

## Sintassi

DAX

```
SUMX(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.

## Valore restituito

Numero decimale.

## Osservazioni:

- La funzione SUMX accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento è una colonna che contiene i numeri che si vogliono sommare o un'espressione che restituisce una colonna.
- Vengono conteggiati solo i numeri nella colonna. I valori vuoti, logici e di testo vengono ignorati.
- Per esempi più complessi di SUMX nelle formule, vedere [ALL](#) e [CALCULATETABLE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente prima filtra la tabella InternetSales in base all'espressione 'InternetSales[SalesTerritoryID] = 5` e quindi restituisce la somma di tutti i valori nella colonna Freight. In altre parole, l'espressione restituisce la somma degli addebiti per la spedizione solo per l'area di vendita specificata.

DAX

```
= SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])
```

Se non è necessario filtrare la colonna, usare la funzione SUM. La funzione SUM è simile alla funzione di Excel con lo stesso nome, ma accetta una colonna come riferimento.

## Vedi anche

[SUM](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni di data e ora

Articolo • 20/10/2023

Queste funzioni consentono di creare calcoli basati su data e ora. Molte delle funzioni in DAX sono simili alle funzioni di data e ora di Excel. Le funzioni DAX, tuttavia, usano un tipo di dati **datetime** e possono accettare valori da una colonna come argomento.

## In questa categoria

Funzione	Descrizione
<a href="#">CALENDAR</a>	Restituisce una tabella con una sola colonna denominata "Date" che contiene un set di date contiguo.
<a href="#">CALENDARAUTO</a>	Restituisce una tabella con una sola colonna denominata "Date" che contiene un set di date contiguo.
<a href="#">DATE</a>	Restituisce la data specificata in formato datetime.
<a href="#">DATEDIFF</a>	Restituisce il numero di limiti di intervallo tra due date.
<a href="#">DATEVALUE</a>	Converte una data dal formato di testo al formato datetime.
<a href="#">DAY</a>	Restituisce il giorno del mese, con un numero compreso tra 1 e 31.
<a href="#">EDATE</a>	Restituisce la data corrispondente al numero di mesi indicato prima o dopo la data di inizio.
<a href="#">EOMONTH</a>	Restituisce la data in formato datetime corrispondente all'ultimo giorno del mese prima o dopo un numero di mesi specificato.
<a href="#">HOUR</a>	Restituisce l'ora come numero compreso tra 0 (12.00 A.M.) e 23 (11.00 P.M.).
<a href="#">MINUTE</a>	Restituisce i minuti come numero compreso tra 0 e 59, in base a un valore di data e ora specificato.
<a href="#">MONTH</a>	Restituisce il mese come un numero compreso tra 1 (gennaio) e 12 (dicembre).
<a href="#">NETWORKDAYS</a>	Restituisce il numero di giorni lavorativi interi tra due date.
<a href="#">NOW</a>	Restituisce la data e l'ora correnti in formato datetime.
<a href="#">QUARTER</a>	Restituisce il trimestre come numero compreso tra 1 e 4.
<a href="#">SECOND</a>	Restituisce i secondi di un valore di ora, come numero compreso tra 0 e 59.
<a href="#">ORA</a>	Converte le ore, i minuti e i secondi specificati come numeri in un'ora in formato datetime.

Funzione	Descrizione
TIMEVALUE	Converte un'ora dal formato di testo al formato datetime.
TODAY	Restituisce la data corrente.
UTCNOW	Restituisce la data e l'ora UTC correnti
UTCTODAY	Restituisce la data UTC corrente.
WEEKDAY	Restituisce un numero compreso tra 1 e 7 che identifica il giorno della settimana di una data.
WEEKNUM	Restituisce il numero della settimana per la data e l'anno specificati in base al valore di return_type.
YEAR	Restituisce l'anno di una data come numero intero di quattro cifre nell'intervallo da 1900 a 9999.
YEARFRAC	Calcola la frazione dell'anno rappresentata dal numero di giorni interi tra due date.

## Commenti e suggerimenti

Questa pagina è stata utile?

# CALENDAR

Articolo • 20/10/2023

Restituisce una tabella con una sola colonna denominata "Date" che contiene un set di date contiguo. L'intervallo di date è compreso tra la data di inizio specificata e la data di fine specificata, incluse le due date.

## Sintassi

DAX
<code>CALENDAR(&lt;start_date&gt;, &lt;end_date&gt;)</code>

## Parametri

Termine	Definizione
start_date	qualsiasi espressione DAX che restituisce un valore DateTime.
end_date	qualsiasi espressione DAX che restituisce un valore DateTime.

## Valore restituito

Restituisce una tabella con una sola colonna denominata "Date" contenente un set di date contiguo. L'intervallo di date è compreso tra la data di inizio specificata e la data di fine specificata, incluse le due date.

## Osservazioni:

- Se start\_date è maggiore di end\_date, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

La formula seguente restituisce una tabella con le date comprese tra il 1° gennaio 2015 e il 31 dicembre 2021.

DAX

```
= CALENDAR (DATE (2015, 1, 1), DATE (2021, 12, 31))
```

Per un modello di dati che include i dati di vendita effettivi e le previsioni di vendita future, l'espressione seguente restituisce una tabella di date che copre l'intervallo di date nelle tabelle Vendita e Previsioni.

DAX

```
= CALENDAR (MINX (Sales, [Date]), MAXX (Forecast, [Date]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# CALENDARAUTO

Articolo • 20/10/2023

Restituisce una tabella con una sola colonna denominata "Date" che contiene un set di date contiguo. L'intervallo di date viene calcolato automaticamente in base ai dati nel modello.

## Sintassi

DAX
<code>CALENDARAUTO([fiscal_year_end_month])</code>

## Parametri

Termine	Definizione
fiscal_year_end_month	Qualsiasi espressione DAX che restituisce un numero intero compreso tra 1 e 12. Se omesso, per impostazione predefinita viene usato il valore specificato nel modello di tabella di calendario per l'utente corrente, se presente. In caso contrario, il valore predefinito è 12.

## Valore restituito

Restituisce una tabella con una sola colonna denominata "Date" che contiene un set di date contiguo. L'intervallo di date viene calcolato automaticamente in base ai dati nel modello.

## Osservazioni:

- L'intervallo di date viene calcolato come segue:
  - La data meno recente nel modello che non si trova in una colonna calcolata o in una tabella calcolata viene usata come data minima.
  - La data più recente nel modello che non si trova in una colonna calcolata o in una tabella calcolata viene usata come data massima.
  - L'intervallo di date restituito è costituito dalle date comprese tra l'inizio dell'anno fiscale associato alla data minima e la fine dell'anno fiscale associato alla data massima.

- Se il modello non contiene valori DateTime che non sono presenti nelle colonne calcolate o nelle tabelle calcolate, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

In questo esempio, le date minima e massima nel modello di dati sono il 1° luglio 2010 e il 30 giugno 2011.

`CALENDARAUTO()` restituirà tutte le date comprese tra il 1° gennaio 2010 e il 31 dicembre 2011.

`CALENDARAUTO(3)` restituirà tutte le date tra il 1° aprile 2010 e il 31 marzo 2012.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATE

Articolo • 20/10/2023

Restituisce la data specificata in formato **datetime**.

## Sintassi

DAX

```
DATE(<year>, <month>, <day>)
```

## Parametri

Termine	Definizione
year	<p>Numero che rappresenta l'anno.</p> <p>Il valore dell'argomento <b>year</b> può includere un numero di cifre compreso tra uno e quattro. L'argomento <b>year</b> viene interpretato in base al sistema di data usato dal computer.</p> <p>Sono supportate le date a partire dal 1° marzo 1900.</p> <p>Se si immette un numero con cifre decimali, il numero viene arrotondato.</p> <p>Per i valori maggiori di 9999 o minori di zero (valori negativi), la funzione restituisce un errore <b>#VALUE!</b>.</p> <p>Se il valore di <b>year</b> è compreso tra 0 e 1899, il valore viene sommato a 1900 per generare il valore finale. Vedere gli esempi seguenti. <b>Nota:</b> È consigliabile usare quattro cifre per l'argomento <b>anno</b> ogni volta che è possibile, evitare risultati indesiderati. Se ad esempio si usa 07, come valore dell'anno viene restituito 1907.</p>
mese	<p>Numero che rappresenta il mese o il risultato di un calcolo eseguito in base alle regole seguenti:</p> <p>I valori interi negativi non sono supportati. I valori validi sono 1-12.</p> <p>Se <b>month</b> è un numero compreso tra 1 e 12, rappresenta un mese dell'anno. 1 rappresenta gennaio, 2 rappresenta febbraio e così via fino a 12, che rappresenta dicembre.</p> <p>Se si immette un numero intero maggiore di 12, viene eseguito il calcolo seguente: la data viene calcolata sommando il valore di <b>month</b> al valore di <b>year</b>. Nel caso di DATE</p>

Termine	Definizione
	<p>(2008, 18, 1), ad esempio, la funzione restituisce un valore datetime equivalente al 1° giugno 2009, perché vengono sommati 18 mesi all'inizio del 2008, ottenendo così un valore pari al mese di giugno 2009. Vedere gli esempi seguenti.</p>
Giorno	<p>Numero che rappresenta il giorno o il risultato di un calcolo eseguito in base alle regole seguenti:</p> <p>I valori interi negativi non sono supportati. I valori validi sono 1-31.</p> <p>Se <b>day</b> è un numero compreso tra 1 e l'ultimo giorno del mese specificato, rappresenta un giorno del mese.</p> <p>Se si immette un numero intero maggiore dell'ultimo giorno del mese, viene eseguito il calcolo seguente: la data viene calcolata sommando il valore di <b>day</b> al valore di <b>month</b>. Nella formula <code>DATE( 2008, 3, 32 )</code>, ad esempio, la funzione DATE restituisce un valore <b>datetime</b> equivalente al 1° aprile 2008, perché vengono sommati 32 giorni all'inizio di marzo, ottenendo così un valore pari al 1° aprile.</p> <p>Se il valore di <b>day</b> contiene cifre decimali, viene arrotondato al valore intero più vicino.</p>

## Valore restituito

Restituisce la data specificata (**datetime**).

## Osservazioni:

- La funzione DATE accetta i numeri interi immessi come argomenti e genera la data corrispondente. La funzione DATE è particolarmente utile nelle situazioni in cui l'anno, il mese e il giorno sono forniti da formule. I dati sottostanti potrebbero ad esempio contenere date in un formato non riconosciuto come data, ad esempio AAAAMMGG. È possibile usare la funzione DATE insieme ad altre funzioni per convertire le date in un numero che può essere riconosciuto come data.
- A differenza di Microsoft Excel, in cui le date vengono archiviate come numero di serie, le funzioni di data DAX restituiscono sempre un tipo di dati **datetime**. È tuttavia possibile usare la formattazione per visualizzare le date come numeri di serie, se lo si desidera.
- È anche possibile specificare data e datetime come valore letterale nel formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Se specificato come valore letterale, l'uso della funzione DATE nell'espressione non è necessario. Per ulteriori informazioni, vedere [Sintassi DAX | Data e ora](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

### Data semplice

La formula seguente restituisce la data dell'8 luglio 2009:

DAX

```
= DATE(2009,7,8)
```

### Anni prima del 1899

Se il valore immesso per l'argomento **year** è compreso tra 0 (zero) e 1899 (incluso), tale valore viene sommato a 1900 per calcolare l'anno. La formula seguente restituisce il 2 gennaio 1908: (1900+08).

DAX

```
= DATE(08,1,2)
```

### Anni dopo il 1899

Se **year** è compreso tra 1900 e 9999 (incluso), tale valore viene usato come anno. La formula seguente restituisce la data del 2 gennaio 2008:

DAX

```
= DATE(2008,1,2)
```

## Mesi

Se il valore di **month** è maggiore di 12, **month** somma tale numero al primo mese dell'anno specificato. La formula seguente restituisce la data del 2 febbraio 2009:

DAX

```
= DATE(2008,14,2)
```

# giorni

Se il valore di **day** è maggiore del numero di giorni nel mese specificato, **day** somma tale numero di giorni al primo giorno del mese. La formula seguente restituisce la data del 4 febbraio 2008:

DAX

```
= DATE(2008,1,35)
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione DAY](#)

[Funzione TODAY](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATEDIFF

Articolo • 20/10/2023

Restituisce il numero di limiti di intervallo tra due date.

## Sintassi

DAX

```
DATEDIFF(<Date1>, <Date2>, <Interval>)
```

## Parametri

Termine	Definizione
Date1	Valore datetime scalare.
Data2	Valore datetime scalare.
Intervallo	Intervallo da usare per il confronto delle date. Il valore può essere uno dei seguenti: <ul style="list-style-type: none"><li>- SECONDI</li><li>- MINUTI</li><li>- ORE</li><li>- GIORNO</li><li>- SETTIMANA</li><li>- MESE</li><li>- TRIMESTRE</li><li>- ANNO</li></ul>

## Valore restituito

Numero di limiti di intervallo tra due date.

## Osservazioni:

Se Date2 è maggiore di Date1, viene restituito un risultato positivo. Se Date1 è maggiore di Date2, viene restituito un risultato negativo.

## Esempio

Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).

La query DAX seguente:

```
DAX

EVALUATE
VAR StartDate = DATE ( 2019, 07, 01 )
VAR EndDate = DATE ( 2021, 12, 31 )
RETURN
{
    ( "Year",      DATEDIFF ( StartDate, EndDate, YEAR ) ),
    ( "Quarter",   DATEDIFF ( StartDate, EndDate, QUARTER ) ),
    ( "Month",     DATEDIFF ( StartDate, EndDate, MONTH ) ),
    ( "Week",      DATEDIFF ( StartDate, EndDate, WEEK ) ),
    ( "Day",       DATEDIFF ( StartDate, EndDate, DAY ) )
}
```

Restituisce quanto segue:

Value1	Value2
Year	2
Trimestrale	9
Mese	29
Settimanale	130
Giornaliera	914

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# DATEVALUE

Articolo • 20/10/2023

Converte una data dal formato di testo al formato datetime.

## Sintassi

DAX

```
DATEVALUE(date_text)
```

## Parametri

Termine	Definizione
date_text	Testo che rappresenta una data.

## Valore proprietà/valore restituito

Data in formato **datetime**.

## Osservazioni:

- Durante la conversione, DATEVALUE usa le impostazioni locali e quelle relative a data e ora del modello per determinare un valore di data. Se le impostazioni relative a data e ora del modello rappresentano le date nel formato mese/giorno/anno, la stringa "1/8/2009" viene convertita in un valore **datetime** equivalente a 8 gennaio 2009. Tuttavia, se le impostazioni relative a data e ora del modello rappresentano le date nel formato giorno/mese/anno, la stessa stringa viene convertita in un valore **datetime** equivalente a 1 agosto 2009.
- Se la conversione che usa le impostazioni locali e quelle relative a data e ora del modello ha esito negativo, DATEVALUE tenterà di usare altri formati di data. In questo caso, è possibile che alcune righe siano convertite in un formato e altre righe in un formato diverso. Ad esempio, la data "5/4/2018" potrebbe essere convertita in 4 maggio 2018 e "20/4/2018" in 20 aprile.
- Se la parte relativa all'anno dell'argomento **date\_text** viene omessa, la funzione DATEVALUE usa l'anno corrente dal clock integrato del computer. Le informazioni

relative all'ora nell'argomento **date\_text** vengono ignorate.

- Le impostazioni locali e quelle relative a data e ora del modello sono determinate inizialmente dall'applicazione e dal computer al momento della creazione del modello.
- È anche possibile specificare data e datetime come valore letterale nel formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Se specificato come valore letterale, l'utilizzo della funzione DATEVALUE nell'espressione non è necessario. Per ulteriori informazioni, vedere [Sintassi DAX | Data e ora](#).

## Esempio

L'esempio seguente restituisce un valore **datetime** diverso a seconda delle impostazioni locali del modello e delle impostazioni relative alla modalità di presentazione di date e ore.

- Nelle impostazioni di data/ora in cui il giorno precede il mese, l'esempio restituisce un valore **datetime** corrispondente all'8 gennaio 2009.
- Nelle impostazioni di data/ora in cui il mese precede il giorno, l'esempio restituisce un valore **datetime** corrispondente al 1° agosto 2009.

DAX

```
= DATEVALUE("8/1/2009")
```

## Vedi anche

[Funzioni di data e ora](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# DAY

Articolo • 20/10/2023

Restituisce il giorno del mese, con un numero compreso tra 1 e 31.

## Sintassi

DAX

```
DAY(<date>)
```

## Parametri

Termine	Definizione
data	Data in formato <b>datetime</b> o rappresentazione di testo di una data.

## Valore restituito

Numero intero che indica il giorno del mese.

## Osservazioni:

- La funzione DAY accetta come argomento la data del giorno che si sta cercando di trovare. È possibile fornire date alla funzione usando un'altra funzione di data o un'espressione che restituisce una data oppure digitando una data in formato **datetime**. È anche possibile digitare una data in uno dei formati stringa validi per le date.
- I valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori gregoriani indipendentemente dal formato di visualizzazione del valore di data fornito. Ad esempio, se il formato di visualizzazione della data fornita è Hijri, i valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori associati alla data gregoriana equivalente.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, la funzione DAY usa le impostazioni locali e quelle relative a data/ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Se le impostazioni correnti relative a data/ora rappresentano le date nel formato

mese/giorno/anno, la stringa "1/8/2009" viene interpretata come valore **datetime** equivalente a 8 gennaio 2009 e la funzione restituisce 8. Tuttavia, se le impostazioni correnti relative a data/ora rappresentano le date nel formato giorno/mese/anno, la stessa stringa viene interpretata come valore **datetime** equivalente a 1 agosto 2009 e la funzione restituisce 1.

## Esempio: Recupero del giorno da una colonna relativa alla data

La formula seguente restituisce il giorno dalla data nella colonna [Birthdate].

DAX

```
= DAY([Birthdate])
```

## Esempio: recupero del giorno da una data in formato stringa

Le formule seguenti restituiscono il giorno 4, usando le date fornite come stringhe in un formato di testo valido.

DAX

```
= DAY("3-4-1007")  
= DAY("March 4 2007")
```

## Esempio: uso di un valore di giorno come condizione

L'espressione seguente restituisce il giorno in cui è stato inserito ogni ordine di vendita e contrassegna la riga come un elemento di vendita promozionale se l'ordine è stato effettuato il giorno 10 del mese.

DAX

```
= IF( DAY([SalesDate])=10, "promotion", "")
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione TODAY](#)

[Funzione DATA](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EDATE

Articolo • 20/10/2023

Restituisce la data corrispondente al numero di mesi indicato prima o dopo la data di inizio. Usare EDATE per calcolare le date di maturità o di scadenza che cadono lo stesso giorno del mese della data di emissione.

## Sintassi

DAX

```
EDATE(<start_date>, <months>)
```

## Parametri

Termine	Definizione
start_date	Data in formato <b>datetime</b> o <b>text</b> che rappresenta la data di inizio.
months	Numero intero che rappresenta il numero di mesi prima o dopo <b>start_date</b> .

## Valore restituito

Data (**datetime**).

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri di serie, per le date in DAX viene usato un formato **datetime**. Le date archiviate in altri formati vengono convertite in modo implicito.
- Se **start\_date** non è una data valida, EDATE restituisce un errore. Verificare che il riferimento di colonna o il valore di data specificato come primo argomento sia una data.
- Se **months** non è un numero intero, la parte decimale viene troncata.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, la funzione EDATE usa le impostazioni locali e quelle relative a data/ora del computer client per comprendere il valore di testo in modo da poter eseguire la

conversione. Se le impostazioni correnti relative a data/ora rappresentano una data nel formato mese/giorno/anno, la stringa "1/8/2009" viene interpretata come valore datetime equivalente a 8 gennaio 2009. Tuttavia, se le impostazioni correnti relative a data/ora rappresentano una data nel formato giorno/mese/anno, la stessa stringa viene interpretata come valore datetime equivalente a 1 agosto 2009.

- Se la data richiesta è successiva all'ultimo giorno del mese corrispondente, viene restituito l'ultimo giorno del mese. Ad esempio, le funzioni seguenti: EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) restituiscono il 28 febbraio 2009; che corrisponde a un mese dopo la data di inizio.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce la data tre mesi dopo la data dell'ordine, archiviata nella colonna [TransactionDate].

DAX

```
= EDATE([TransactionDate],3)
```

## Vedi anche

[funzione EOMONTH](#)

[Funzioni di data e ora](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EOMONTH

Articolo • 20/10/2023

Restituisce la data in formato **datetime** corrispondente all'ultimo giorno del mese prima o dopo un numero di mesi specificato. Usare EOMONTH per calcolare le date di maturità o di scadenza che cadono l'ultimo giorno del mese.

## Sintassi

DAX

```
EOMONTH(<start_date>, <months>)
```

## Parametri

Termine	Definizione
start_date	Data di inizio in formato <b>datetime</b> o in una rappresentazione di testo valida di una data.
months	Numero di mesi prima o dopo <b>start_date</b> . <b>Nota:</b> se si immette un numero diverso da un intero, il numero viene arrotondato per eccesso o per difetto all'intero più vicino.

## Valore restituito

Data (**datetime**).

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri di serie, per le date in DAX viene usato un formato **datetime**. La funzione EOMONTH può accettare date in altri formati, con le restrizioni seguenti:
- Se **start\_date** non è una data valida, EOMONTH restituisce un errore.
- Se **start\_date** è un valore numerico che non si trova in un formato **datetime**, EOMONTH convertirà il numero in una data. Per evitare risultati imprevisti, convertire il numero in un formato **datetime** prima di usare la funzione EOMONTH.

- Se **start\_date** più **months** genera una data non valida, EOMONTH restituisce un errore. Le date precedenti il 1° marzo 1900 e successive al 31 dicembre 9999 non sono valide.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, la funzione EDATE usa le impostazioni locali e quelle relative a data/ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Se le impostazioni correnti relative a data/ora rappresentano una data nel formato mese/giorno/anno, la stringa "1/8/2009" viene interpretata come valore datetime equivalente a 8 gennaio 2009. Tuttavia, se le impostazioni correnti relative a data/ora rappresentano una data nel formato giorno/mese/anno, la stessa stringa viene interpretata come valore datetime equivalente a 1 agosto 2009.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'espressione seguente restituisce la data del 31 maggio 2008, perché l'argomento **months** è arrotondato a 2.

DAX

```
= EOMONTH("March 3, 2008",1.5)
```

## Vedi anche

[Funzione EDATE](#)

[Funzioni di data e ora](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# HOUR

Articolo • 20/10/2023

Restituisce l'ora come numero compreso tra 0 (12.00 A.M.) e 23 (11.00 P.M.).

## Sintassi

DAX

```
HOUR(<datetime>)
```

## Parametri

Termine	Definizione
data/ora	Valore <b>datetime</b> , come 16:48:00 o 4:48 PM.

## Valore restituito

Numero intero compreso tra 0 e 23.

## Osservazioni:

- La funzione HOUR accetta come argomento l'orario che contiene l'ora che si vuole trovare. È possibile fornire l'orario usando una funzione di data/ora o un'espressione che restituisce un valore **datetime** oppure digitando direttamente il valore in uno dei formati di ora accettati. Gli orari possono anche essere immessi come qualsiasi rappresentazione di testo accettata per l'orario.
- Quando l'argomento **datetime** è una rappresentazione testuale della data e ora, la funzione usa le impostazioni locali e quelle relative a data e ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Nella maggior parte dei paesi vengono usati i due punti (:) come separatore dell'ora e qualsiasi testo di input in cui viene usato questo separatore verrà analizzato correttamente. Verificare le impostazioni locali per comprendere i risultati.

## Esempio 1

L'esempio seguente restituisce l'ora dalla colonna **TransactionTime** di una tabella denominata **Orders**.

DAX

```
= HOUR('Orders'[TransactionTime])
```

## Esempio 2

L'esempio seguente restituisce 15, ovvero l'ora corrispondente a 3 PM in un formato a 24 ore. Il valore di testo viene analizzato automaticamente e convertito in un valore di data/ora.

DAX

```
= HOUR("March 3, 2008 3:00 PM")
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione MINUTE](#)

[Funzione YEAR](#)

[Funzione SECOND](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MINUTE

Articolo • 20/10/2023

Restituisce i minuti come numero compreso tra 0 e 59, in base a un valore di data e ora specificato.

## Sintassi

DAX

`MINUTE(<datetime>)`

## Parametri

Termine	Definizione
data/ora	Valore <b>datetime</b> o testo in un formato ora accettato, come 16:48:00 o 4:48 PM.

## Valore restituito

Numero intero compreso tra 0 e 59.

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date e le ore vengono archiviate in formato di numeri di serie, DAX usa un tipo di dati **datetime** per le date e ore. È possibile fornire il valore di **datetime** alla funzione MINUTE facendo riferimento a una colonna in cui sono archiviate date e ore, tramite una funzione di data/ora o tramite un'espressione che restituisce data e ora.
- Quando l'argomento **datetime** è una rappresentazione testuale della data e ora, la funzione usa le impostazioni locali e quelle relative a data e ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Nella maggior parte dei paesi vengono usati i due punti (:) come separatore dell'ora e qualsiasi testo di input in cui viene usato questo separatore verrà analizzato correttamente. Verificare le impostazioni locali per comprendere i risultati.

## Esempio 1

Nell'esempio seguente vengono restituiti i minuti dal valore archiviato nella colonna **TransactionTime** della tabella **Orders**.

DAX

```
= MINUTE(Orders[TransactionTime])
```

## Esempio 2

Nell'esempio seguente viene restituito 45, cioè il numero di minuti dell'ora, 1:45 PM.

DAX

```
= MINUTE("March 23, 2008 1:45 PM")
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione HOUR](#)

[Funzione YEAR](#)

[Funzione SECOND](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MONTH

Articolo • 20/10/2023

Restituisce il mese come un numero compreso tra 1 (gennaio) e 12 (dicembre).

## Sintassi

DAX

**MONTH(<datetime>)**

## Parametri

Termine	Definizione
data	Data in formato <b>datetime</b> o testo.

## Valore restituito

Numero intero compreso tra 1 e 12.

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri seriali, in DAX per le date viene usato un formato **datetime**. È possibile immettere la data usata come argomento della funzione MONTH digitando un formato **datetime** accettato, fornendo un riferimento a una colonna che contiene date o usando un'espressione che restituisce una data.
- I valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori gregoriani indipendentemente dal formato di visualizzazione del valore di data fornito. Ad esempio, se il formato di visualizzazione della data fornita è Hijri, i valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori associati alla data gregoriana equivalente.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, la funzione usa le impostazioni locali e quelle relative a data e ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Se le impostazioni correnti relative alla data e ora rappresentano una data nel

formato mese/giorno/anno, la stringa 1/8/2009 viene interpretata come valore datetime equivalente a 8 gennaio 2009 e la funzione restituisce un risultato pari a 1. Se invece le impostazioni correnti relative alla data e ora rappresentano una data nel formato giorno/mese/anno, la stessa stringa viene interpretata come valore datetime equivalente a 1 agosto 2009 e la funzione restituisce un risultato pari a 8.

- Se la rappresentazione di testo della data non può essere convertita correttamente in un valore datetime, la funzione restituisce un errore.

## Esempio 1

L'espressione seguente restituisce 3, cioè il numero intero che corrisponde a marzo, il mese dell'argomento **date**.

DAX

```
= MONTH("March 3, 2008 3:45 PM")
```

## Esempio 2

L'espressione seguente restituisce il mese dalla data nella colonna **TransactionDate** della tabella **Orders**.

DAX

```
= MONTH(Orders[TransactionDate])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione HOUR](#)

[Funzione MINUTE](#)

[Funzione YEAR](#)

[Funzione SECOND](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NETWORKDAYS

Articolo • 20/10/2023

Restituisce il numero di giorni lavorativi interi tra due date (inclusi). I parametri specificano quali e quanti giorni sono compresi nel fine settimana. I giorni del fine settimana e i giorni specificati come festivi non vengono considerati giorni lavorativi.

## Sintassi

DAX

```
NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])
```

## Parametri

Termine	Definizione
start_date	Una data che rappresenta la data di inizio. Date per cui calcolare la differenza. La start_date può essere precedente, uguale o successiva all'end_date.
end_date	Una data che rappresenta la data di fine. Date per cui calcolare la differenza. La start_date può essere precedente, uguale o successiva all'end_date.
weekend	Indica i giorni della settimana che sono giorni festivi e non sono inclusi nel numero di giorni lavorativi interi tra start_date ed end_date. Il fine settimana è un numero di fine settimana che specifica quando cadono i fine settimana. I valori dei numeri del fine settimana indicano i giorni del fine settimana seguenti: 1 o omesso: sabato, domenica 2: domenica, lunedì 3: lunedì, martedì 4: martedì, mercoledì 5: mercoledì, giovedì 6: giovedì, venerdì 7: venerdì, sabato 11: solo domenica 12: solo lunedì 13: solo martedì 14: solo mercoledì solo 15: solo giovedì 16: solo venerdì 17: solo sabato
festività	Tabella a colonna di una o più date che devono essere escluse dal calendario dei giorni lavorativi.

# Valore restituito

Numero intero di giorni lavorativi interi.

## Osservazioni:

- Questa funzione DAX è simile alle funzioni NETWORKDAYS.INTL e NETWORKDAYS di Excel.
- Se start\_date ed end\_date entrambi sono BLANK, anche il valore di output è BLANK.
- Se start\_date o end\_date è BLANK, start\_date o end\_date BLANK verrà considerato come Date(1899, 12, 30).
- Le date devono essere specificate utilizzando la [funzione DATE](#) o come risultato di un'altra espressione. Ad esempio, `DATE ( 2022, 5, 30 )` specifica il 30 maggio 2022. Le date possono anche essere specificate come valore [letterale](#) nel formato `(dt"2022-05-30")`. Non specificare date come testo.

## Esempio

L'espressione seguente:

```
DAX  
  
= NETWORKDAYS (  
    DATE ( 2022, 5, 28 ),  
    DATE ( 2022, 5, 30 ),  
    1,  
    {  
        DATE ( 2022, 5, 30 )  
    }  
)
```

Restituisce:

[Valore]
0

In questo esempio viene restituito 0 perché la data di inizio è un sabato e la data di fine è un lunedì. Il parametro weekend specifica che il fine settimana è sabato e domenica,

quindi non si tratta di giorni lavorativi. Il parametro festività contrassegna il 30 (la data di fine) come festività, quindi non rimangono giorni lavorativi.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NOW

Articolo • 20/10/2023

Restituisce la data e l'ora correnti in formato **datetime**.

La funzione NOW è utile quando è necessario visualizzare la data e l'ora correnti in un foglio di lavoro o calcolare un valore basato sulla data e sull'ora correnti e fare in modo che tale valore venga aggiornato ogni volta che si apre il foglio di lavoro.

## Sintassi

DAX

```
NOW()
```

## Valore restituito

Data (**datetime**).

## Osservazioni:

- Il risultato della funzione NOW cambia solo quando viene aggiornata la colonna che contiene la formula. Tale risultato non viene aggiornato continuamente.
- Nel servizio Power BI il risultato della funzione NOW è sempre nel fuso orario UTC.
- La funzione TODAY restituisce la stessa data ma non è precisa per quanto riguarda l'ora. L'ora restituita è sempre 12:00:00 e solo la data viene aggiornata.

## Esempio

L'esempio seguente restituisce la data e l'ora correnti più 3,5 giorni:

DAX

```
= NOW()+3.5
```

## Vedi anche

[Funzione UTCNOW](#)

[Funzione TODAY](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# QUARTER

Articolo • 20/10/2023

Restituisce il trimestre come numero compreso tra 1 (gennaio-marzo) e 4 (ottobre-dicembre).

## Sintassi

DAX

```
QUARTER(<date>)
```

## Parametri

Termine	Definizione
data	Data.

## Valore restituito

Numero intero compreso tra 1 e 4.

## Osservazioni:

Se il valore di input è vuoto, anche il valore di output è vuoto.

## Esempio 1

La query DAX seguente:

DAX

```
EVALUATE { QUARTER(DATE(2019, 2, 1)), QUARTER(DATE(2018, 12, 31)) }
```

Restituisce:

[Value]
1

**[Value]**

## Esempio 2

La query DAX seguente:

```
DAX

EVALUATE
ADDCOLUMNS(
    FILTER(
        VALUES(
            FactInternetSales[OrderDate]),
        [OrderDate] >= DATE(2008, 3, 31) && [OrderDate] <= DATE(2008, 4,
1)
    ),
    "Quarter", QUARTER([OrderDate])
)
```

Restituisce:

FactInternetSales[OrderDate]	[Quarter]
3/31/2008	1
4/1/2008	2

## Vedi anche

[YEAR](#)

[MONTH](#)

[DAY](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)     [No](#)

# SECOND

Articolo • 20/10/2023

Restituisce i secondi di un valore di ora, come numero compreso tra 0 e 59.

## Sintassi

DAX

```
SECOND(<time>)
```

## Parametri

Termine	Definizione
Ora	Ora in formato <b>datetime</b> , ad esempio 16:48:23 o 4:48:47 PM.

## Valore restituito

Numero intero compreso tra 0 e 59.

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date e le ore vengono archiviate come numeri seriali, in DAX per le date e le ore viene usato un formato **datetime**. Se i dati di origine non sono in questo formato, DAX converte in modo implicito i dati. È possibile usare la formattazione per visualizzare le date e le ore come numero seriale, se necessario.
- Il valore di data/ora specificato come argomento della funzione SECOND può essere immesso come stringa di testo tra virgolette (ad esempio, "6:45 PM"). È anche possibile specificare un valore di ora come risultato di un'altra espressione o come riferimento a una colonna che contiene ore.
- Se si specifica un valore numerico di un altro tipo di dati, ad esempio 13.60, il valore viene interpretato come numero seriale ed è rappresentato come tipo di dati **datetime** prima di estrarre il valore per i secondi. Per comprendere più facilmente i risultati, potrebbe essere necessario rappresentare tali numeri come date prima di usarli nella funzione SECOND. Se ad esempio si usa SECOND con una

colonna che contiene un valore numerico, come 25.56, la formula restituisce 24 perché, se formattato come data, il valore 25.56 equivale a January 25, 1900, 1:26:24 PM.

- Quando l'argomento **time** è una rappresentazione testuale di una data e ora, la funzione usa le impostazioni locali e quelle relative a data e ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Nella maggior parte dei paesi vengono usati i due punti (:) come separatore dell'ora e qualsiasi testo di input in cui viene usato questo separatore verrà analizzato correttamente. Verificare le impostazioni locali per comprendere i risultati.

## Esempio 1

La formula seguente restituisce il numero di secondi nell'ora contenuta nella colonna **TransactionTime** di una tabella denominata **Orders**.

DAX

```
= SECOND('Orders'[TransactionTime])
```

## Esempio 2

La formula seguente restituisce 3, che è il numero di secondi nella data e ora rappresentata dal valore **March 3, 2008 12:00:03**.

DAX

```
= SECOND("March 3, 2008 12:00:03")
```

## Vedi anche

[Funzioni di data e ora](#)

[HOUR](#)

[MINUTE](#)

[YEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ORA

Articolo • 20/10/2023

Converte le ore, i minuti e i secondi specificati come numeri in un'ora in formato **datetime**.

## Sintassi

DAX

```
TIME(hour, minute, second)
```

## Parametri

Termine	Definizione
ora	<b>Modalità di importazione:</b> Numero compreso tra 0 e 32767 che rappresenta l'ora. Qualsiasi valore maggiore di 23 verrà diviso per 24 e il resto verrà considerato come valore dell'ora, rappresentato come frazione di un giorno. Ad esempio, TIME(27,0,0) = TIME(3,0,0) = 3:00:00 AM  <b>Modalità DirectQuery:</b> numero compreso tra 0 e 23 che rappresenta l'ora.
minuto	<b>Modalità di importazione:</b> Numero compreso tra 0 e 32767 che rappresenta i minuti. Qualsiasi valore maggiore di 59 verrà convertito in ore e minuti. Qualsiasi valore maggiore di 1440 (24 ore) non modifica la parte della data, ma verrà divisa per 1440 e il resto verrà considerato come valore del minuto, rappresentato come frazione di un giorno. Ad esempio, TIME(0,2190,0) = TIME(0,750,0) = TIME(12,30,0) = 12:30:00 PM  <b>Modalità DirectQuery:</b> numero compreso tra 0 e 59 che rappresenta il minuto.
second	<b>Modalità di importazione:</b> Numero compreso tra 0 e 32767 che rappresenta i secondi. Qualsiasi valore maggiore di 59 verrà convertito in ore, minuti e secondi. Ad esempio, TIME(0,0,2000) = TIME(0,33,20) = 12:33:20 AM  <b>Modalità DirectQuery:</b> numero compreso tra 0 e 59 che rappresenta il secondo.

## Valore restituito

Ora (**datetime**) compresa tra le 00:00:00 (12:00:00) e le 23:59:59 (11:59:59 PM).

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date e le ore vengono archiviate come numeri seriali, in DAX viene usato un formato **datetime** per l'uso di valori di data e ora. I numeri in altri formati vengono convertiti in modo implicito quando si usa un valore di data/ora in una funzione DAX. Se è necessario usare numeri seriali, è possibile usare la formattazione per modificare la modalità di visualizzazione dei numeri.
- I valori di ora sono una parte di un valore di data e nel sistema di numerazione seriale sono rappresentati da un numero decimale. Il valore **datetime** 12:00 PM equivale quindi a 0.5, perché corrisponde alla metà di un giorno.
- È possibile fornire gli argomenti alla funzione TIME come valori digitati direttamente, come risultato di un'altra espressione o tramite un riferimento a una colonna che contiene un valore numerico.
- È anche possibile specificare data e datetime come valore letterale nel formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Se specificato come valore letterale, l'uso della funzione TIME nell'espressione non è necessario. Per ulteriori informazioni, vedere [Sintassi DAX | Data e ora](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Gli esempi seguenti restituiscono entrambi l'ora 3:00 AM:

DAX

```
= TIME(27,0,0)
```

DAX

```
= TIME(3,0,0)
```

## Esempio 2

Gli esempi seguenti restituiscono entrambi l'ora 12:30 PM:

DAX

```
= TIME(0,750,0)
```

DAX

```
= TIME(12,30,0)
```

## Esempio 3

L'esempio seguente crea un'ora in base ai valori delle colonne `intHours`, `intMinutes`, `intSeconds`:

DAX

```
= TIME([intHours],[intMinutes],[intSeconds])
```

## Vedi anche

[DATE](#)

[Funzioni di data e ora](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# TIMEVALUE

Articolo • 20/10/2023

Converte un'ora dal formato di testo al formato datetime.

## Sintassi

DAX

```
TIMEVALUE(time_text)
```

## Parametri

Termine	Definizione
time_text	Stringa di testo che rappresenta una determinata ora del giorno. Le informazioni sulla data incluse nell'argomento <b>time_text</b> vengono ignorate.

## Valore restituito

Data (datetime).

## Osservazioni:

- I valori di ora sono una parte di un valore di data e sono rappresentati da un numero decimale. Ad esempio, 12:00 PM viene rappresentato come 0.5 perché è la metà di un giorno.
- Quando l'argomento **time\_text** è una rappresentazione testuale della data e dell'ora, la funzione usa le impostazioni locali e quelle relative a data e ora del modello per comprendere il valore di testo e poter quindi eseguire la conversione. Nella maggior parte dei paesi vengono usati i due punti (:) come separatore dell'ora e qualsiasi testo di input in cui viene usato questo separatore verrà analizzato correttamente. Verificare le impostazioni locali per comprendere i risultati.
- È anche possibile specificare data e datetime come valore letterale nel formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Se specificato come valore letterale, l'utilizzo della funzione TIMEVALUE

nell'espressione non è necessario. Per ulteriori informazioni, vedere [Sintassi DAX | Data e ora](#).

## Esempio

DAX

```
= TIMEVALUE("20:45:30")
```

## Vedi anche

[Funzioni di data e ora](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# TODAY

Articolo • 20/10/2023

Restituisce la data corrente.

## Sintassi

DAX

`TODAY()`

## Valore restituito

Data (datetime).

## Osservazioni:

- La funzione TODAY è utile quando è necessario visualizzare la data corrente in un foglio di lavoro, indipendentemente da quando si apre la cartella di lavoro. È utile anche per il calcolo degli intervalli.
- Se la funzione TODAY non aggiorna la data quando previsto, potrebbe essere necessario modificare le impostazioni che controllano quando la colonna o la cartella di lavoro viene aggiornata.
- La funzione NOW è simile, ma restituisce l'ora esatta, mentre TODAY restituisce il valore orario 12:00:00 PM per tutte le date.

## Esempio

Se si è certi che qualcuno è nato nel 1963, è possibile usare la formula seguente per trovare l'età della persona corrispondente al compleanno di quest'anno:

DAX

`= YEAR(TODAY())-1963`

Questa formula usa la funzione TODAY come argomento per la funzione YEAR per ottenere l'anno corrente e quindi sottrae 1963, restituendo l'età della persona.

## Vedi anche

[Funzioni di data e ora](#)  
[NOW](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# UTCNOW

Articolo • 20/10/2023

Restituisce la data e l'ora UTC correnti.

## Sintassi

DAX

```
UTCNOW()
```

## Valore restituito

Un elemento (datetime).

## Osservazioni:

Il risultato della funzione UTCNOW cambia solo quando viene aggiornata la formula. Tale risultato non viene aggiornato continuamente.

## Esempio

Il codice seguente:

DAX

```
EVALUATE { FORMAT(UTCNOW(), "General Date") }
```

Restituisce:

[Value]

2/2/2018 4:48:08 AM

## Vedi anche

[Funzione NOW](#)

[Funzione UTCTODAY](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# UTCTODAY

Articolo • 20/10/2023

Restituisce la data UTC corrente.

## Sintassi

DAX

```
UTCTODAY()
```

## Valore restituito

Data.

## Osservazioni:

- UTCTODAY restituisce il valore 12:00:00 PM per tutte le date.
- La funzione UTCNOW è simile, ma restituisce la data e l'ora esatte.

## Esempio

Il codice seguente:

DAX

```
EVALUATE { FORMAT(UTCTODAY(), "General Date") }
```

Restituisce:

[Value]

02/02/2018

## Vedi anche

[Funzione NOW](#)

[Funzione UTCNOW](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# WEEKDAY

Articolo • 20/10/2023

Restituisce un numero compreso tra 1 e 7 che identifica il giorno della settimana di una data. Per impostazione predefinita, il giorno è compreso tra 1 (domenica) e 7 (sabato).

## Sintassi

DAX

```
WEEKDAY(<date>, <return_type>)
```

## Parametri

Termine	Definizione
data	Data in formato <b>datetime</b> .  Le date devono essere immesse con la funzione DATE, usando espressioni che hanno come risultato una data, o come risultato di altre formule.
return_type	Numero che determina il valore restituito:  Tipo restituito: <b>1</b> , la settimana inizia la domenica (1) e termina il sabato (7), con numerazione da 1 a 7.  Tipo restituito: <b>2</b> , la settimana inizia il lunedì (1) e termina la domenica (7).  Tipo restituito: <b>3</b> , la settimana inizia il lunedì (0) e termina la domenica (6), con numerazione da 1 a 7.

## Valore restituito

Numero intero compreso tra 1 e 7.

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri seriali, in DAX viene usato un formato **datetime** per l'uso di date e ore. Se è necessario visualizzare le date come numeri seriali, è possibile usare le opzioni di formattazione in Excel.

- È anche possibile digitare date in una rappresentazione testuale accettata di una data, ma per evitare risultati imprevisti, è preferibile convertire prima la data testuale nel formato **datetime**.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, nella funzione vengono usate le impostazioni locali e quelle relative a data/ora del computer client per comprendere il valore di testo in modo da poter eseguire la conversione. Se le impostazioni correnti relative a data/ora rappresentano le date nel formato mese/giorno/anno, la stringa "1/8/2009" viene interpretata come valore **datetime** equivalente a 8 gennaio 2009. Tuttavia, se le impostazioni correnti relative a data/ora rappresentano le date nel formato giorno/mese/anno, la stessa stringa viene interpretata come valore **datetime** equivalente a 1 agosto 2009.

## Esempio

Nell'esempio seguente viene ottenuta la data dalla colonna [HireDate], viene aggiunto il valore 1 e viene visualizzato il giorno della settimana corrispondente a tale data. Poiché l'argomento **return\_type** è stato omesso, viene usato il formato predefinito, in cui 1 è domenica e 7 è sabato. Se il risultato è 4, il giorno sarà mercoledì.

DAX

```
= WEEKDAY([HireDate]+1)
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione WEEKNUM](#)

[Funzione YEARFRAC](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# WEEKNUM

Articolo • 20/10/2023

Restituisce il numero della settimana per la data specificata in base al valore di `return_type`. Il numero della settimana indica il punto in cui la settimana cade numericamente nell'arco di un anno.

Per questa funzione vengono usati due *sistemi*:

- **Sistema 1:** la settimana che contiene il 1° gennaio è la prima settimana dell'anno ed è numerata settimana 1.
- **Sistema 2:** la settimana contenente il primo giovedì dell'anno è la prima settimana dell'anno e viene numerata come settimana 1. Questo sistema è la metodologia specificata in ISO 8601, comunemente nota come sistema di numerazione delle settimane europeo.

## Sintassi

DAX

```
WEEKNUM(<date>[ , <return_type>])
```

## Parametri

Termine	Definizione
data	Data in formato <code>datetime</code> .
return_type	(Facoltativo) Numero che determina il giorno di inizio della settimana. Il valore predefinito è 1. Vedere la sezione Osservazioni.

## Valore restituito

Numero intero.

## Osservazioni:

- Per impostazione predefinita, la funzione WEEKNUM usa una convenzione di calendario in cui la settimana che contiene il 1° gennaio viene considerata la prima

settimana dell'anno. Tuttavia, lo standard del calendario ISO 8601, ampiamente usato in Europa, definisce la prima settimana come quella con il maggior numero di giorni (quattro o più) che rientrano nel nuovo anno. Questo significa che, se **return\_type** è qualsiasi valore valido diverso da 21, per gli anni in cui sono presenti tre giorni o meno nella prima settimana di gennaio, la funzione WEEKNUM restituisce numeri di settimana diversi dalla definizione ISO 8601.

- Per **return\_type**, ad eccezione di 21, i valori validi seguenti potrebbero non essere supportati da alcune origini dati DirectQuery:

<b>return_type</b>	<b>Inizio della settimana</b>	<b>Sistema</b>
1 o omesso	Domenica	1
2	lunedì	1
11	lunedì	1
12	Martedì	1
13	mercoledì	1
14	giovedì	1
15	Venerdì	1
16	Sabato	1
17	Domenica	1
21	lunedì	2

## Esempio 1

L'esempio seguente viene restituito il numero della settimana del 14 febbraio 2010. Questo calcolo presuppone che le settimane inizino di lunedì.

DAX

```
= WEEKNUM("Feb 14, 2010", 2)
```

## Esempio 2

Nell'esempio seguente viene restituito il numero della settimana della data archiviata nella colonna **HireDate** dalla tabella **Employees**. Questo calcolo presuppone che le

settimane inizino di domenica.

DAX

```
= WEEKNUM('Employees'[HireDate])
```

## Vedi anche

[Funzione YEARFRAC](#)

[Funzione WEEKDAY](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# YEAR

Articolo • 20/10/2023

Restituisce l'anno di una data come numero intero di quattro cifre nell'intervallo da 1900 a 9999.

## Sintassi

DAX

`YEAR(<date>)`

## Parametri

Termine	Definizione
data	Data in formato <b>datetime</b> o testo contenente l'anno che si vuole trovare.

## Valore restituito

Numero intero compreso nell'intervallo da 1900 a 9999.

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri seriali, in DAX viene usato un tipo di dati **datetime** per l'uso di date e ore.
- Le date devono essere immesse usando la funzione DATE o come risultati di altre formule o funzioni. È anche possibile immettere le date con rappresentazioni testuali accettate per una data, ad esempio 3 marzo 2007 o 3-mar-2003.
- I valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori gregoriani indipendentemente dal formato di visualizzazione del valore di data fornito. Ad esempio, se il formato di visualizzazione della data fornita usa il calendario Hijri, i valori restituiti dalle funzioni YEAR, MONTH e DAY saranno valori associati alla data gregoriana equivalente.
- Quando l'argomento relativo alla data è una rappresentazione testuale della data, la funzione usa le impostazioni locali e quelle relative a data e ora del computer

client per comprendere il valore di testo in modo da poter eseguire la conversione. Potrebbero verificarsi errori se il formato delle stringhe non è compatibile con le impostazioni locali correnti. Ad esempio, se le impostazioni locali definiscono il formato mese/giorno/anno per le date e la data viene specificata come giorno/mese/anno, la data 25/1/2009 non verrà interpretata come il 25 gennaio 2009 ma come data non valida.

## Esempio

L'esempio seguente restituisce 2007.

DAX

```
= YEAR("March 2007")
```

## Esempio - Data come risultato dell'espressione

### Descrizione

Nell'esempio seguente viene restituito l'anno per la data odierna.

DAX

```
= YEAR(TODAY())
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzione HOUR](#)

[Funzione MINUTE](#)

[Funzione YEAR](#)

[Funzione SECOND](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# YEARFRAC

Articolo • 20/10/2023

Calcola la frazione dell'anno rappresentata dal numero di giorni interi tra due date. Usare la funzione per foglio di lavoro YEARFRAC per identificare la percentuale di vantaggi o obblighi dell'intero anno da assegnare a un termine specifico.

## Sintassi

DAX

```
YEARFRAC(<start_date>, <end_date>, <basis>)
```

## Parametri

Termine	Definizione
start_date	Data di inizio in formato <b>datetime</b> .
end_date	Data di fine in formato <b>datetime</b> .
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Tutti gli argomenti vengono troncati in numeri interi.  Basis - Descrizione  0 - US (NASD) 30/360 (valore predefinito)  1 - Effettivo/effettivo  2 - Effettivo/360  3 - Effettivo/365  4 - Europeo 30/360

## Valore restituito

Numero decimale. Il tipo di dati interno è un numero a virgola mobile a precisione doppia IEEE a 64 bit (8 byte).

## Osservazioni:

- A differenza di Microsoft Excel, in cui le date vengono archiviate come numeri seriali, in DAX viene usato un formato **datetime** per l'uso di date e ore. Se è necessario visualizzare le date come numeri seriali, è possibile usare le opzioni di formattazione in Excel.
- Se **start\_date** o **end\_date** non sono date valide, YEARFRAC restituisce un errore.
- Se **basis**< 0 o **basis**> 4, YEARFRAC restituisce un errore.

## Esempio 1

Nell'esempio seguente viene restituita la frazione di un anno rappresentata dalla differenza tra le date nelle due colonne, `TransactionDate` e `ShippingDate`:

DAX

```
= YEARFRAC(Orders[TransactionDate],Orders[ShippingDate])
```

## Esempio 2

Nell'esempio seguente viene restituita la frazione di un anno rappresentata dalla differenza tra le date 1 gennaio e 1 marzo:

DAX

```
= YEARFRAC("Jan 1 2007","Mar 1 2007")
```

Usare gli anni a quattro cifre, quando possibile, per evitare di ottenere risultati imprevisti. Quando l'anno viene troncato, si presuppone l'anno corrente. Quando la data viene omessa, viene presupposta la prima data del mese.

Anche il secondo argomento, **basis**, è stato omesso. La frazione dell'anno viene pertanto calcolata in base allo standard US (NASD) 30/360.

## Vedi anche

[Funzioni di data e ora](#)

[Funzione WEEKNUM](#)

[Funzione YEARFRAC](#)

[Funzione WEEKDAY](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni di filtro

Articolo • 20/10/2023

Le funzioni di filtro e per i valori in DAX sono tra le più complesse ed efficaci e sono molto diverse dalle funzioni di Excel. Le funzioni di ricerca utilizzano tabelle e relazioni, come un database. Le funzioni di filtro consentono di modificare il contesto dei dati per creare calcoli dinamici.

## In questa categoria

Funzione	Descrizione
ALL	Restituisce tutte le righe di una tabella o tutti i valori di una colonna ignorando i filtri eventualmente applicati.
ALLCROSSFILTERED	Cancella tutti i filtri applicati a una tabella.
ALLEXCEPT	Rimuove tutti i filtri di contesto nella tabella ad eccezione dei filtri che sono stati applicati alle colonne specificate.
ALLNOBLANKROW	Restituisce dalla tabella padre di una relazione tutte le righe ad eccezione della riga vuota o tutti i valori distinti di una colonna ad eccezione della riga vuota e ignora gli eventuali filtri di contesto esistenti.
ALLSELECTED	Rimuove i filtri di contesto dalle colonne e le righe nella query corrente, mantenendo tutti gli altri filtri di contesto o filtri espliciti.
CALCULATE	Valuta un'espressione in un contesto di filtro modificato.
CALCULATETABLE	Valuta un'espressione di tabella in un contesto di filtro modificato.
EARLIER	Restituisce il valore corrente della colonna specificata in un passaggio di valutazione esterno della colonna indicata.
EARLIEST	Restituisce il valore corrente della colonna specificata in un passaggio di valutazione esterno della colonna specificata.
FILTER	Restituisce una tabella che rappresenta un subset di un'altra tabella o espressione.
INDEX	Restituisce una riga in una posizione assoluta, specificata dal parametro di posizione, all'interno della partizione specificata, ordinata in base all'ordine specificato o all'asse specificato.
KEEPFILTERS	Modifica la modalità di applicazione dei filtri durante la valutazione di una funzione CALCULATE o CALCULATETABLE.

Funzione	Descrizione
LOOKUPVALUE	Restituisce il valore per la riga che soddisfa tutti i criteri specificati dalle condizioni di ricerca. La funzione può applicare una o più condizioni di ricerca.
MATCHBY	Nelle funzioni finestra definisce le colonne utilizzate per determinare come trovare le corrispondenze con i dati e identificare la <i>riga corrente</i> .
OFFSET	Restituisce una singola riga posizionata prima o dopo la <i>riga corrente</i> all'interno della stessa tabella da un offset specificato.
ORDERBY	Definisce le colonne che determinano l'ordinamento all'interno di ognuna delle partizioni di una funzione WINDOW.
PARTITIONBY	Definisce le colonne utilizzate per partizionare il parametro di <relazione> di una funzione WINDOW.
RANK	Restituisce la classificazione di una riga all'interno dell'intervallo specificato.
REMOVEFILTERS	Cancella i filtri dalle tabelle o dalle colonne specificate.
ROWNUMBER	Restituisce la classificazione univoca di una riga entro l'intervallo specificato.
SELECTEDVALUE	Restituisce il valore se il contesto per columnName è stato filtrato in base a un solo valore distinto. In caso contrario, restituisce alternateResult.
WINDOW	Restituisce più righe posizioionate all'interno dell'intervallo specificato.

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# ALL

Articolo • 20/10/2023

Restituisce tutte le righe di una tabella o tutti i valori di una colonna ignorando i filtri eventualmente applicati. Questa funzione è utile per cancellare i filtri e creare calcoli in tutte le righe di una tabella.

## Sintassi

DAX

```
ALL( <table> | <column>[, <column>[, <column>[,...]]] )
```

## Parametri

Termine	Definizione
table	Tabella in cui cancellare i filtri.
colonna	Colonna in cui cancellare i filtri.

L'argomento della funzione ALL deve essere un riferimento a una tabella di base o un riferimento a una colonna di base. Non è possibile usare espressioni di tabella o espressioni di colonna con la funzione ALL.

## Valore restituito

Tabella o colonna con i filtri rimossi.

## Osservazioni:

- Questa funzione non viene usata da sola, ma serve come funzione intermedia che è possibile usare per modificare il set di risultati su cui viene eseguito qualche altro calcolo.
- Il comportamento normale per le espressioni DAX che contengono la funzione ALL() prevede che tutti i filtri applicati vengano ignorati. Questo tuttavia non accade in alcuni scenari grazie ad *Auto Exist*, una tecnologia DAX che ottimizza l'applicazione dei filtri per ridurre le operazioni di elaborazione necessarie per

determinate query DAX. Auto Exist e ALL(), ad esempio, danno risultati imprevisti quando si applicano filtri a due o più colonne della stessa tabella (come quando si usano i filtri dei dati) e in questa stessa tabella è presente una misura che usa ALL(). In questo caso, Auto Exist *unirà* i diversi filtri in uno solo e lo applicherà solo alle combinazioni esistenti di valori. A causa di questa unione, la misura verrà calcolata sulle combinazioni di valori esistenti e il risultato sarà basato sui valori filtrati invece che su tutti i valori come previsto. Per altre informazioni su Auto Exist e su come influisce sui calcoli, vedere l'articolo [Understanding DAX Auto-Exist](#) (Informazioni sulla tecnologia Auto Exist di DAX) del Microsoft MVP Alberto Ferrari su [sql.bi.com](#).

- La tabella seguente descrive come è possibile usare le funzioni ALL e ALLEXCEPT in scenari diversi.

<b>Funzione e utilizzo</b>	<b>Descrizione</b>
ALL()	Rimuove tutti i filtri ovunque. Si può usare ALL() solo per cancellare i filtri, ma non per restituire una tabella.
ALL(Table)	Rimuove tutti i filtri dalla tabella specificata. ALL(Table) restituisce in pratica tutti i valori di una tabella, rimuovendo qualsiasi filtro applicato nel contesto. Questa funzione è utile quando si usano numerosi livelli di raggruppamento e si vuole creare un calcolo che produca un rapporto di un valore aggregato con il valore totale. Questo scenario è illustrato nel primo esempio.
ALL (Column[, Column[, ...]])	Rimuove tutti i filtri dalle colonne specificate nella tabella. Tutti gli altri filtri in altre colonne nella tabella vengono mantenuti. Tutti gli argomenti della colonna devono provenire dalla stessa tabella. La variante ALL(Column) è utile quando si vogliono rimuovere i filtri di contesto per una o più colonne specifiche e mantenere tutti gli altri filtri di contesto. Questo scenario è illustrato nel secondo e nel terzo esempio.
ALLEXCEPT(Table, Column1 [,Column2...])	Rimuove tutti i filtri di contesto nella tabella ad eccezione dei filtri applicati alle colonne specificate. Si tratta di un metodo veloce e utile per le situazioni in cui si vogliono rimuovere i filtri da molte colonne di una tabella, ma non da tutte.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Calcolare il rapporto tra vendite di una categoria e vendite totali

Si supponga di voler calcolare l'importo delle vendite per la cella corrente nella tabella pivot, diviso per le vendite totali per tutti i rivenditori. Per assicurarsi che il denominatore sia lo stesso indipendentemente dai filtri o dai raggruppamenti applicati dall'utente alla tabella pivot, è possibile definire una formula in cui viene usata la funzione ALL per calcolare il totale complessivo corretto.

La tabella seguente mostra i risultati quando viene creata una nuova misura, **All Reseller Sales Ratio**, usando la formula mostrata nella sezione relativa al codice. Per vedere come funziona, aggiungere il campo CalendarYear all'area **Etichette di riga** della tabella pivot e aggiungere il campo ProductCategoryName all'area **Etichette di colonna**. Trascinare quindi la misura **All Reseller Sales Ratio** nell'area dei valori della tabella pivot. Per visualizzare i risultati sotto forma di percentuali, usare le funzionalità di formattazione di Excel per applicare una formattazione di numero percentuale alle celle che contengono la misura.

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Componenti</b>	<b>Grand Total</b>
2005	0.02%	9.10%	0,04%	0,75%	9.91%
2006	0.11%	24.71%	0.60%	4.48%	29.90%
2007	0.36%	31.71%	1,07%	6.79%	39.93%
2008	0.20%	16.95%	0,48%	2.63%	20.26%
Grand Total	0.70%	82.47%	2.18%	14,65%	100,00%

## Formula

DAX

```
= SUMX(ResellerSales_USD,
ResellerSales_USD[SalesAmount_USD])/SUMX(ALL(ResellerSales_USD),
ResellerSales_USD[SalesAmount_USD])
```

La formula si crea come segue:

1. Il numeratore, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, è la somma dei valori in `ResellerSales_USD[SalesAmount_USD]` per la cella corrente nella tabella pivot, con i filtri di contesto applicati a `CalendarYear` e `ProductCategoryName`.
2. Per il denominatore si inizia specificando una tabella, `ResellerSales_USD`, e si usa la funzione `ALL` per rimuovere tutti i filtri di contesto della tabella.

3. Si usa poi la funzione SUMX per sommare i valori nella colonna ResellerSales\_USD[SalesAmount\_USD]. In altre parole, si ottiene la somma di ResellerSales\_USD[SalesAmount\_USD] per tutte le vendite dei rivenditori.

## Esempio 2

Calcolare il rapporto tra vendite prodotto e vendite totali dell'anno corrente

Si supponga di voler creare una tabella in cui mettere a confronto la percentuale di vendite negli anni per ogni categoria di prodotto, ProductCategoryName. Per ottenere la percentuale per ogni anno per ogni valore di ProductCategoryName, è necessario dividere la somma delle vendite per l'anno specifico e la categoria di prodotto per la somma delle vendite della stessa categoria di prodotto nel corso di tutti gli anni. In altre parole, si vuole mantenere il filtro applicato a ProductCategoryName, ma rimuovere il filtro applicato all'anno quando si calcola il denominatore della percentuale.

La tabella seguente mostra i risultati quando viene creata una nuova misura, **Reseller Sales Year**, usando la formula mostrata nella sezione del codice. Per vedere come funziona, aggiungere il campo CalendarYear all'area **Etichette di riga** di una tabella pivot, poi aggiungere il campo ProductCategoryName all'area **Etichette di colonna**. Per visualizzare i risultati sotto forma di percentuali, usare le funzionalità di formattazione di Excel per applicare una formattazione di numero percentuale alle celle che contengono la misura **Reseller Sales Year**.

Etichette di riga	Accessori	Bikes	Clothing	Componenti	Grand Total
2005	3.48%	11.03%	1.91%	5.12%	9.91%
2006	16.21%	29.96%	27.29%	30.59%	29.90%
2007	51.62%	38.45%	48.86%	46.36%	39.93%
2008	28.69%	20.56%	21.95%	17.92%	20.26%
Grand Total	100,00%	100,00%	100,00%	100,00%	100,00%

### Formula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(  
SUM( ResellerSales_USD[SalesAmount_USD]), ALL(DateTime[CalendarYear]))
```

La formula si crea come segue:

- Il numeratore, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, è la somma dei valori in `ResellerSales_USD[SalesAmount_USD]` per la cella corrente nella tabella pivot, con i filtri di contesto applicati alle colonne `CalendarYear` e `ProductCategoryName`.
- Per il denominatore si rimuove il filtro esistente su `CalendarYear` usando la funzione `ALL(Column)`. Questa operazione consente di calcolare la somma sulle righe restanti della tabella `ResellerSales_USD`, dopo avere applicato i filtri di contesto esistenti dalle etichette di colonna. Il risultato finale è che per il denominatore la somma viene calcolata sugli elementi `ProductCategoryName` selezionati (filtro di contesto implicito) e per tutti i valori di `Year`.

## Esempio 3

Calcolare il contributo delle categorie di prodotto rispetto alle vendite totali per anno

Si supponga di voler creare una tabella che mostra la percentuale di vendite per ogni categoria di prodotto su base annua. Per ottenere la percentuale per ogni categoria di prodotto per un determinato anno, è necessario calcolare la somma delle vendite per quella specifica categoria di prodotto, `ProductCategoryName`, nell'anno n, quindi dividere il valore risultante per la somma delle vendite per l'anno n per tutte le categorie di prodotto. In altre parole, si vuole mantenere il filtro applicato all'anno, ma rimuovere il filtro applicato a `ProductCategoryName` quando si calcola il denominatore della percentuale.

La tabella seguente mostra i risultati quando viene creata una nuova misura, **Reseller Sales CategoryName**, usando la formula mostrata nella sezione del codice. Per vedere come funziona, aggiungere il campo `CalendarYear` all'area **Etichette di riga** della tabella pivot, poi aggiungere il campo `ProductCategoryName` all'area **Etichette di colonna**. Aggiungere quindi la nuova misura all'area dei **valori** della tabella pivot. Per visualizzare i risultati sotto forma di percentuali, usare le funzionalità di formattazione di Excel per applicare una formattazione di numero percentuale alle celle che contengono la nuova misura **Reseller Sales CategoryName**.

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Componenti</b>	<b>Grand Total</b>
2005	0,25%	91.76%	0.42%	7.57%	100,00%
2006	0.38%	82.64%	1.99%	14.99%	100,00%
2007	0,90%	79.42%	2.67%	17.01%	100,00%
2008	0,99%	83.69%	2.37%	12.96%	100,00%

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Componenti</b>	<b>Grand Total</b>
Grand Total	0.70%	82.47%	2.18%	14,65%	100,00%

## Formula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(
SUM( ResellerSales_USD[SalesAmount_USD]),
ALL(ProductCategory[ProductCategoryName]))
```

La formula si crea come segue:

1. Il numeratore, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, è la somma dei valori in `ResellerSales_USD[SalesAmount_USD]` per la cella corrente nella tabella pivot, con i filtri di contesto applicati ai campi `CalendarYear` e `ProductCategoryName`.
2. Per il denominatore si usa la funzione `ALL(Column)` per rimuovere il filtro in `ProductCategoryName` e si calcola la somma delle righe rimanenti nella tabella `ResellerSales_USD`, dopo avere applicato i filtri di contesto esistenti dalle etichette di riga. Il risultato finale è che, per il denominatore, la somma viene calcolata sull'anno selezionato (filtro di contesto implicito) e per tutti i valori di `ProductCategoryName`.

## Vedi anche

[Funzioni di filtro](#)

[Funzione ALL](#)

[Funzione ALLEXCEPT](#)

[Funzione FILTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ALLCROSSFILTERED

Articolo • 20/10/2023

Cancella tutti i filtri applicati a una tabella.

## Sintassi

DAX

```
ALLCROSSFILTERED(<table>)
```

## Parametri

Termine	Definizione
table	Tabella in cui cancellare i filtri.

## Valore restituito

N/D. Vedere la sezione Osservazioni.

## Osservazioni:

- Si può usare ALLCROSSFILTERED solo per cancellare i filtri, ma non per restituire una tabella.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
DEFINE  
MEASURE FactInternetSales[TotalQuantity1] =  
    CALCULATE(SUM(FactInternetSales[OrderQuantity]),  
    ALLCROSSFILTERED(FactInternetSales))  
MEASURE FactInternetSales[TotalQuantity2] =  
    CALCULATE(SUM(FactInternetSales[OrderQuantity]), ALL(FactInternetSales))  
EVALUATE
```

```

SUMMARIZECOLUMNS(DimSalesReason[SalesReasonName],
    "TotalQuantity1", [TotalQuantity1],
    "TotalQuantity2", [TotalQuantity2])
ORDER BY DimSalesReason[SalesReasonName]

```

Restituisce:

<b>DimSalesReason[SalesReasonName]</b>	<b>[TotalQuantity1]</b>	<b>[TotalQuantity2]</b>
Demo Event	60398	
Magazine Advertisement	60398	
Produttore	60398	1818
On Promotion	60398	7390
Altro	60398	3653
Prezzo	60398	47733
Qualità	60398	1551
Rivedi	60398	1640
Sponsorship	60398	
Television Advertisement	60398	730

#### ⓘ Nota

Tra le tabelle FactInternetSales e DimSalesReason esiste una relazione molti-a-molti direttamente o indirettamente.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# ALLEXCEPT

Articolo • 20/10/2023

Rimuove tutti i filtri di contesto nella tabella ad eccezione dei filtri che sono stati applicati alle colonne specificate.

## Sintassi

DAX

```
ALLEXCEPT(<table>,<column>[,<column>[,...]])
```

## Parametri

Termine	Definizione
table	Tabella da cui vengono eliminati tutti i filtri di contesto, eccetto i filtri nelle colonne specificate in argomenti successivi.
colonna	Colonna per la quale i filtri di contesto devono essere mantenuti.

Il primo argomento della funzione ALLEXCEPT deve essere un riferimento a una tabella di base. Tutti gli argomenti successivi devono essere riferimenti alle colonne di base. Non è possibile usare espressioni di tabella o espressioni di colonna con la funzione ALLEXCEPT.

## Valore restituito

Tabella da cui sono stati rimossi tutti i filtri ad eccezione di quelli nelle colonne specificate.

## Osservazioni:

- Questa funzione non viene usata da sola, ma serve come funzione intermedia che è possibile usare per modificare il set di risultati su cui viene eseguito qualche altro calcolo.
- ALL e ALLEXCEPT possono essere usati in scenari diversi:

Funzione e utilizzo	Descrizione
ALL(Table)	Rimuove tutti i filtri dalla tabella specificata. ALL(Table) restituisce in pratica tutti i valori di una tabella, rimuovendo qualsiasi filtro applicato nel contesto. Questa funzione è utile quando si usano numerosi livelli di raggruppamento e si vuole creare un calcolo che produca un rapporto di un valore aggregato con il valore totale.
ALL (Column[, Column[, ...]])	Rimuove tutti i filtri dalle colonne specificate nella tabella. Tutti gli altri filtri in altre colonne nella tabella vengono mantenuti. Tutti gli argomenti della colonna devono provenire dalla stessa tabella. La variante ALL(Column) è utile quando si vogliono rimuovere i filtri di contesto per una o più colonne specifiche e mantenere tutti gli altri filtri di contesto.
ALLEXCEPT(Table, Column1 [,Column2]...)	Rimuove tutti i filtri di contesto nella tabella ad eccezione dei filtri applicati alle colonne specificate. Si tratta di un metodo veloce e utile per le situazioni in cui si vogliono rimuovere i filtri da molte colonne di una tabella, ma non da tutte.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nella formula viene eseguita la somma di SalesAmount\_USD e viene usata la funzione ALLEXCEPT per rimuovere qualsiasi filtro di contesto nella tabella DateTime ad eccezione del caso in cui il filtro è applicato alla colonna CalendarYear.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), ALLEXCEPT(DateTime,
DateTime[CalendarYear]))
```

Poiché la formula usa la funzione ALLEXCEPT, ogni volta che qualsiasi colonna della tabella DateTime, ad eccezione di CalendarYear, viene usata per sezionare una visualizzazione, la formula rimuove qualsiasi filtro dei dati, fornendo un valore uguale alla somma di SalesAmount\_USD. Tuttavia, se la colonna CalendarYear viene usata per sezionare la visualizzazione, i risultati sono diversi. Poiché CalendarYear è specificato come argomento di ALLEXCEPT, quando i dati vengono sezionati in base all'anno, viene applicato un filtro agli anni a livello di riga.

## Vedi anche

[Funzioni di filtro](#)

[Funzione ALL](#)

[Funzione FILTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ALLNOBLANKROW

Articolo • 20/10/2023

Restituisce dalla tabella padre di una relazione tutte le righe ad eccezione della riga vuota o tutti i valori distinti di una colonna ad eccezione della riga vuota e ignora gli eventuali filtri di contesto esistenti.

## Sintassi

DAX

```
ALLNOBLANKROW( {<table> | <column>[ , <column>[ , <column>[,...]]]} )
```

## Parametri

Termine	Definizione
table	Tabella da cui vengono eliminati tutti i filtri di contesto.
colonna	Colonna da cui vengono eliminati tutti i filtri di contesto.

È necessario passare un solo parametro; il parametro è una tabella o una colonna.

## Valore restituito

Una tabella, quando il parametro passato è una tabella o una colonna di valori, quando il parametro passato è una colonna.

## Osservazioni:

- La funzione ALLNOBLANKROW filtra solo la riga vuota che verrà mostrata da una tabella padre di una relazione in presenza di una o più righe della tabella figlio contenenti valori non corrispondenti alla colonna padre. Per una spiegazione completa, vedere l'esempio seguente.
- La tabella seguente riepiloga le variazioni della funzione ALL fornite in DAX con le relative differenze:

Funzione e utilizzo	Descrizione
ALL(Column)	Rimuove tutti i filtri dalle colonne specificate nella tabella. Tutti gli altri filtri nella tabella, applicati alle altre colonne, vengono mantenuti.
ALL(Table)	Rimuove tutti i filtri dalla tabella specificata.
ALLEXCEPT(Table,Col1,Col2...)	Esegue l'override di tutti i filtri di contesto nella tabella, ad eccezione di quelli applicati alle colonne specificate.
ALLNOBLANK(table column)	Restituisce dalla tabella padre di una relazione tutte le righe ad eccezione della riga vuota o tutti i valori distinti di una colonna ad eccezione della riga vuota e ignora gli eventuali filtri di contesto esistenti

Per una descrizione generale del funzionamento di ALL, insieme a esempi dettagliati in cui vengono usate le funzioni ALL(Table) e ALL(Column), vedere [Funzione ALL](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nei dati di esempio, la tabella ResellerSales\_USD contiene una sola riga senza valori e non può pertanto essere correlata alle tabelle padre nelle relazioni presenti nella cartella di lavoro. Si userà questa tabella in una tabella pivot per seguire il comportamento della riga vuota e la modalità di gestione dei conteggi sui dati non correlati.

Passaggio 1: verificare i dati non correlati

Aprire la **finestra di PowerPivot**, quindi selezionare la tabella ResellerSales\_USD. Nella colonna ProductKey filtrare i valori vuoti. Rimarrà una sola riga. In tale riga tutti i valori di colonna dovrebbero essere vuoti, ad eccezione di SalesOrderLineNumber.

Passaggio 2: Creare una tabella pivot

Creare una nuova tabella pivot, quindi trascinare la colonna datetime.[Calendar Year] sul riquadro delle etichette di riga. Nella tabella seguente vengono illustrati i risultati previsti:

Etichette di riga
2005

<b>Etichette di riga</b>
2006
2007
2008
Grand Total

Si noti l'etichetta vuota tra **2008** e **Totale complessivo**. Questa etichetta vuota rappresenta il membro sconosciuto, ovvero un gruppo speciale creato per gestire qualsiasi valore della tabella figlio che non ha un valore corrispondente nella tabella padre e che in questo esempio è indicato dalla colonna datetime.[Calendar Year].

La presenza di questa etichetta vuota nella tabella pivot indica che in alcune delle tabelle correlate alla colonna datetime.[Calendar Year] sono presenti valori vuoti o non corrispondenti. La tabella padre è quella che mostra l'etichetta vuota, ma le righe che non corrispondono si trovano in una o più delle tabelle figlio.

Le righe che vengono aggiunte a questo gruppo di etichette vuote sono valori che non corrispondono a valori della tabella padre, ad esempio una data che non esiste nella tabella datetime, oppure sono valori Null che indicano l'assenza di valori per la data. In questo esempio è stato inserito un valore vuoto in tutte le colonne della tabella delle vendite figlio. La presenza di un numero maggiore di valori nella tabella padre rispetto alle tabelle figlio non rappresenta un problema.

### Passaggio 3: Contare le righe usando ALL e ALLNOBLANK

Aggiungere le due misure seguenti alla tabella datetime per contare le righe della tabella: **Countrows ALLNOBLANK of datetime**, **Countrows ALL of datetime**. Le formule che è possibile usare per definire queste misure sono:

```
DAX

// Countrows ALLNOBLANK of datetime
= COUNTROWS(ALLNOBLANKROW('DateTime'))

// Countrows ALL of datetime
= COUNTROWS(ALL('DateTime'))

// Countrows ALLNOBLANKROW of ResellerSales_USD
= COUNTROWS(ALLNOBLANKROW('ResellerSales_USD'))

// Countrows ALL of ResellerSales_USD
= COUNTROWS(ALL('ResellerSales_USD'))
```

In una tabella pivot vuota aggiungere la colonna datetime.[Calendar Year] alle etichette di riga, quindi aggiungere le misure appena create. I risultati saranno simili alla tabella seguente:

<b>Etichette di riga</b>	<b>Countrows ALLNOBLANK of datetime</b>	<b>Countrows ALL of datetime</b>
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
Grand Total	1280	1281

Nei risultati viene mostrata una differenza di una riga nel conteggio delle righe della tabella. Tuttavia, se si apre la **finestra di PowerPivot** e si seleziona la tabella datetime, non è possibile trovare righe vuote in quanto la riga vuota speciale menzionata corrisponde al membro sconosciuto.

#### Passaggio 4: Verificare l'accuratezza del conteggio

Per avere la certezza che ALLNOBLANKROW non conti le righe realmente vuote e gestisca solo la riga vuota speciale nella tabella padre, aggiungere le due misure seguenti alla tabella ResellerSales\_USD: **Countrows ALLNOBLANKROW of ResellerSales\_USD, Countrows ALL of ResellerSales\_USD**.

Creare una nuova tabella pivot e quindi trascinare la colonna datetime.[Calendar Year] sul riquadro delle etichette di riga. A questo punto aggiungere le misure appena create. I risultati saranno simili ai seguenti:

<b>Etichette di riga</b>	<b>Countrows ALLNOBLANKROW of ResellerSales_USD</b>	<b>Countrows ALL of ResellerSales_USD</b>
2005	60856	60856
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856
Grand Total	60856	60856

Ora le due misure hanno gli stessi risultati. Il motivo è che la funzione ALLNOBLANKROW non conta le righe realmente vuote di una tabella, ma gestisce solo la riga vuota che rappresenta un caso speciale generato in una tabella padre, quando una o più delle tabelle figlio della relazione contengono valori non corrispondenti o valori vuoti.

## Vedi anche

[Funzioni di filtro](#)

[Funzione ALL](#)

[Funzione FILTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ALLSELECTED

Articolo • 20/10/2023

Rimuove i filtri di contesto dalle colonne e le righe nella query corrente, mantenendo tutti gli altri filtri di contesto o filtri esplicativi.

La funzione ALLSELECTED ottiene il contesto che rappresenta tutte le righe e le colonne nella query, mantenendo filtri e contesti esplicativi diversi dai filtri di riga e colonna. Questa funzione si può usare per ottenere i totali visualizzati nelle query.

## Sintassi

DAX

```
ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[,...]]]])
```

## Parametri

Termine	Definizione
tableName	Nome di una tabella esistente con la sintassi DAX standard. Questo parametro non può essere un'espressione. Il parametro è facoltativo.
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione. Il parametro è facoltativo.

## Valore restituito

Contesto della query senza filtri di colonna e riga.

## Osservazioni:

- Se c'è un solo argomento, sarà il parametro *tableName* o *columnName*. Se ci sono più argomenti, devono essere colonne della stessa tabella.
- Questa funzione è diversa da ALL() perché mantiene tutti i filtri impostati in modo esplicito all'interno della query, oltre a tutti i filtri di contesto diversi da quelli di riga e colonna.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente mostra come generare livelli differenti di totali visualizzati in un report tabella mediante espressioni DAX. Nel report due (2) i filtri precedenti sono stati applicati ai dati Reseller Sales, uno su Sales Territory Group = *Europe* e l'altro su Promotion Type = *Volume Discount*. Una volta applicati i filtri, è possibile calcolare i totali visualizzati per l'intero report, per All Years o per All Product Categories. Inoltre, per scopi illustrativi, si ottiene anche il totale complessivo per All Reseller Sales, rimuovendo tutti i filtri nel report. La valutazione dell'espressione DAX seguente comporta la generazione di una tabella con tutte le informazioni necessarie per compilare una tabella con i totali visualizzati.

DAX

```

define
measure 'Reseller Sales'[Reseller Sales Amount]=sum('Reseller Sales'[Sales
Amount])
measure 'Reseller Sales'[Reseller Grand Total]=calculate(sum('Reseller
Sales'[Sales Amount]), ALL('Reseller Sales'))
measure 'Reseller Sales'[Reseller Visual Total]=calculate(sum('Reseller
Sales'[Sales Amount]), ALLSELECTED())
measure 'Reseller Sales'[Reseller Visual Total for All of Calendar
Year]=calculate(sum('Reseller Sales'[Sales Amount]),
ALLSELECTED('Date'[Calendar Year]))
measure 'Reseller Sales'[Reseller Visual Total for All of Product Category
Name]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Product
Category'[Product Category Name]))
evaluate
CalculateTable(
    //CT table expression
    summarize(
//summarize table expression
crossjoin(distinct('Product Category'[Product Category Name]),
distinct('Date'[Calendar Year]))
//First Group by expression
, 'Product Category'[Product Category Name]
//Second Group by expression
, 'Date'[Calendar Year]
//Summary expressions
, "Reseller Sales Amount", [Reseller Sales Amount]
, "Reseller Grand Total", [Reseller Grand Total]
, "Reseller Visual Total", [Reseller Visual Total]
, "Reseller Visual Total for All of Calendar Year", [Reseller Visual Total
for All of Calendar Year]
, "Reseller Visual Total for All of Product Category Name", [Reseller Visual
Total for All of Product Category Name]
```

```

)
//CT filters
, 'Sales Territory'[Sales Territory Group]="Europe", 'Promotion'[Promotion
Type]="Volume Discount"
)
order by [Product Category Name], [Calendar Year]

```

Dopo aver eseguito l'espressione sopra indicata in SQL Server Management Studio su AdventureWorks DW Tabular Model, si ottengono i risultati seguenti:

[Product Category Name]	[Calendar Year]	[Reseller Sales Amount]	[Reseller Grand Total]	[Reseller Visual Total]	[Reseller Visual Total for All of Calendar Year]	[Reseller Visual Total for All of Product Category Name]
Accessori	2000		80450596.9823	877006.7987	38786.018	
Accessori	2001		80450596.9823	877006.7987	38786.018	
Accessori	2002	625.7933	80450596.9823	877006.7987	38786.018	91495.3104
Accessori	2003	26037.3132	80450596.9823	877006.7987	38786.018	572927.0136
Accessori	2004	12122.9115	80450596.9823	877006.7987	38786.018	212584.4747
Accessori	2005		80450596.9823	877006.7987	38786.018	
Accessori	2006		80450596.9823	877006.7987	38786.018	
Bikes	2000		80450596.9823	877006.7987	689287.7939	
Bikes	2001		80450596.9823	877006.7987	689287.7939	
Bikes	2002	73778.938	80450596.9823	877006.7987	689287.7939	91495.3104
Bikes	2003	439771.4136	80450596.9823	877006.7987	689287.7939	572927.0136
Bikes	2004	175737.4423	80450596.9823	877006.7987	689287.7939	212584.4747
Bikes	2005		80450596.9823	877006.7987	689287.7939	
Bikes	2006		80450596.9823	877006.7987	689287.7939	
Clothing	2000		80450596.9823	877006.7987	95090.7757	
Clothing	2001		80450596.9823	877006.7987	95090.7757	
Clothing	2002	12132.4334	80450596.9823	877006.7987	95090.7757	91495.3104
Clothing	2003	58234.2214	80450596.9823	877006.7987	95090.7757	572927.0136

[Product Category Name]	[Calendar Year]	[Reseller Sales Amount]	[Reseller Grand Total]	[Reseller Visual Total]	[Reseller Visual Total for All of Calendar Year]	[Reseller Visual Total for All of Product Category Name]
Clothing	2004	24724.1209	80450596.9823	877006.7987	95090.7757	212584.4747
Clothing	2005		80450596.9823	877006.7987	95090.7757	
Clothing	2006		80450596.9823	877006.7987	95090.7757	
Componenti	2000		80450596.9823	877006.7987	53842.2111	
Componenti	2001		80450596.9823	877006.7987	53842.2111	
Componenti	2002	4958.1457	80450596.9823	877006.7987	53842.2111	91495.3104
Componenti	2003	48884.0654	80450596.9823	877006.7987	53842.2111	572927.0136
Componenti	2004		80450596.9823	877006.7987	53842.2111	212584.4747
Componenti	2005		80450596.9823	877006.7987	53842.2111	
Componenti	2006		80450596.9823	877006.7987	53842.2111	

Le colonne del report sono:

#### Reseller Sales Amount

Valore effettivo di Reseller Sales per l'anno e la categoria di prodotti. Questo valore compare in una cella al centro del report, in corrispondenza dell'intersezione tra anno e categoria.

#### Reseller Visual Total for All of Calendar Year

Valore totale per una categoria di prodotti in tutti gli anni. Questo valore viene visualizzato alla fine di una colonna o riga per una categoria di prodotti specificata e in tutti gli anni nel report.

#### Reseller Visual Total for All of Product Category Name

Valore totale per un anno in tutte le categorie di prodotti. Questo valore viene visualizzato alla fine di una colonna o riga per un anno specificato e in tutte le categorie di prodotti nel report.

#### Reseller Visual Total

Valore totale per tutti gli anni e per tutte le categorie di prodotti. Questo valore viene visualizzato di solito nell'angolo inferiore più a destra della tabella.

## Reseller Grand Total

Si tratta del totale complessivo per tutte le vendite del rivenditore, prima che sia stato applicato qualsiasi filtro. È opportuno notare la differenza con [Reseller Visual Total]. Tenere presente che questo report include due (2) filtri, uno su Product Category Group e l'altro su Promotion Type.

### ⓘ Nota

se l'espressione contiene filtri esplicativi, questi filtri vengono applicati anche all'espressione.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# CALCULATE

Articolo • 20/10/2023

Valuta un'espressione in un contesto di filtro modificato.

## ⚠ Nota

È disponibile anche la funzione **CALCULATETABLE**, che ha esattamente la stessa funzionalità, ad eccezione del fatto che modifica il **contesto di filtro** applicato a un'espressione che restituisce un *oggetto tabella*.

## Sintassi

DAX

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])
```

## Parametri

Termine	Definizione
expression	Espressione da valutare.
filter1, filter2,...	(Facoltativo) Espressioni booleane o di tabella che definiscono i filtri o le funzioni modificatore di filtro.

L'espressione usata come primo parametro corrisponde essenzialmente a una misura.

I filtri possono essere:

- Espressioni filtro booleane
- Espressioni filtro tabella
- Funzioni di modifica del filtro

Quando sono presenti più filtri, possono essere valutati usando [l'operatore logico AND \(&&\)](#) ovvero tutte le condizioni devono essere TRUE o dall'operatore logico OR (||), il che significa che entrambe le condizioni possono essere vere.

## Espressioni filtro booleane

Un filtro di espressione booleana è un'espressione che restituisce TRUE o FALSE. Queste espressioni devono rispettare diverse regole:

- Possono fare riferimento a colonne da una singola tabella.
- Non possono fare riferimento a misure.
- Non possono usare una funzione CALCULATE annidata.

A partire dalla versione di settembre 2021 di Power BI Desktop, si applica anche quanto segue:

- Non possono usare funzioni che analizzano o restituiscono una tabella, a meno che siano passate come argomenti alle funzioni di aggregazione.
- Possono contenere una funzione di aggregazione che restituisce un valore scalare.

Ad esempio, usare

DAX

```
Total sales on the last selected date =  
CALCULATE (  
    SUM ( Sales[Sales Amount] ),  
    'Sales'[OrderDateKey] = MAX ( 'Sales'[OrderDateKey] )  
)
```

## Espressione filtro tabella

Un filtro di espressione tabella applica un oggetto tabella come filtro. Può trattarsi di un riferimento a una tabella del modello, ma è più probabile che si tratti di una funzione che restituisce un oggetto tabella. È possibile usare la funzione FILTER per applicare condizioni di filtro complesse, incluse quelle che non è possibile definire tramite un'espressione filtro booleana.

## Funzioni di modificatore di filtro

Le funzioni di modificatore di filtro consentono di eseguire altre operazioni oltre alla semplice aggiunta di filtri. Forniscono un maggiore controllo quando si modifica il contesto di filtro.

Funzione	Scopo
REMOVEFILTERS	Rimuovere tutti i filtri oppure i filtri da una o più colonne di una tabella o da tutte le colonne di una singola tabella.
ALL <sup>1</sup> , ALLEXCEPT, ALLNOBLANKROW	Rimuovere i filtri da una o più colonne o da tutte le colonne di una singola tabella.

Funzione	Scopo
KEEPFILTERS	Aggiungere un filtro senza rimuovere i filtri esistenti nelle stesse colonne.
USERELATIONSHIP	Creare una relazione inattiva tra colonne correlate. In questo caso, la relazione attiva diventa automaticamente inattiva.
CROSSFILTER	Modificare la direzione del filtro (da entrambe le direzioni a una singola direzione o viceversa) oppure disabilitare una relazione.

<sup>1</sup> La funzione ALL e le relative varianti si comportano sia come modificatori di filtro che come funzioni che restituiscono oggetti tabella. Se la funzione REMOVEFILTERS è supportata dallo strumento in uso, è preferibile usare questa funzione per rimuovere i filtri.

## Valore restituito

Valore che rappresenta il risultato dell'espressione.

## Osservazioni:

- Quando vengono fornite espressioni filtro, la funzione CALCULATE modifica il contesto di filtro per valutare l'espressione. Per ogni espressione filtro, ci sono due possibili risultati standard quando non si esegue il wrapping dell'espressione filtro nella funzione KEEPFILTERS:
  - Se le colonne (o le tabelle) non sono presenti nel contesto di filtro, i nuovi filtri verranno aggiunti al contesto di filtro per valutare l'espressione.
  - Se le colonne (o le tabelle) sono già nel contesto di filtro, i filtri esistenti verranno sovrascritti dai nuovi filtri per valutare l'espressione CALCULATE.
- La funzione CALCULATE usata *senza filtri* soddisfa un requisito specifico. Converte il contesto di riga in contesto di filtro. Ciò è necessario quando un'espressione (non una misura del modello) che riepiloga i dati del modello deve essere valutata nel contesto di riga. Questo scenario può verificarsi in una formula di colonna calcolata o quando viene valutata un'espressione in una funzione iteratore. Si noti che quando una misura del modello viene usata nel contesto di riga, la transizione del contesto è automatica.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempi

La definizione di misura della tabella **Sales** seguente produce un risultato relativo ai ricavi, ma solo per i prodotti di colore blu.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
Blue Revenue =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    'Product'[Color] = "Blue"  
)
```

Categoria	Sales Amount	Blue Revenue
Accessori	\$1,272,057.89	\$165,406.62
Bikes	\$94,620,526.21	\$8,374,313.88
Clothing	\$2,117,613.45	\$259,488.37
Componenti	\$11,799,076.66	\$803,642.10
Totali	\$109,809,274.20	\$9,602,850.97

La funzione CALCULATE valuta la somma della colonna **Sales Amount** della tabella **Sales** in un contesto di filtro modificato. Viene aggiunto un nuovo filtro alla colonna **Color** della tabella **Product** oppure il filtro sovrascrive i filtri già applicati alla colonna.

La definizione di misura della tabella **Sales** seguente produce un rapporto tra vendite e vendite per tutti i canali di vendita.

Canale	Sales Amount	% ricavi canale sul totale
Internet	\$29,358,677.22	26.74%
Reseller	\$80,450,596.98	73.26%
Totali	\$109,809,274.20	100.00%

DAX

```
Revenue % Total Channel =  
DIVIDE(  
    SUM(Sales[Sales Amount]),  
    CALCULATE(  
        SUM(Sales[Sales Amount]),  
        REMOVEFILTERS('Sales Order'[Channel])  
    )  
)
```

La funzione **DIVIDE** divide un'espressione che somma il valore della colonna **Sales Amount** della tabella **Sales** (nel contesto di filtro) per la stessa espressione in un contesto di filtro modificato. La funzione **CALCULATE** modifica il contesto di filtro usando la funzione **REMOVEFILTERS**, che è una funzione di modificatore di filtro. Rimuove i filtri dalla colonna **Channel** della tabella **Sales Order**.

La colonna calcolata della tabella **Customer** seguente classifica i clienti in base a una classe di fedeltà. Si tratta di uno scenario molto semplice: quando i ricavi prodotti dal cliente sono inferiori a 2500 dollari, vengono classificati come *Low*; in caso contrario, sono classificati come *High*.

DAX

```
Customer Segment =  
IF(  
    CALCULATE(SUM(Sales[Sales Amount]), ALLEXCEPT(Customer,  
Customer[CustomerKey])) < 2500,  
    "Low",  
    "High"  
)
```

In questo esempio, il contesto di riga viene convertito nel contesto di filtro. In questo caso si parla di *transizione del contesto*. La funzione **ALLEXCEPT** rimuove i filtri da tutte le colonne della tabella **Customer** ad eccezione della colonna **CustomerKey**.

## Vedi anche

[Contesto di filtro](#)

[Contesto di riga](#)

[Funzione CALCULATETABLE](#)

[Funzioni di filtro](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# CALCULATETABLE

Articolo • 20/10/2023

Valuta un'espressione di tabella in un contesto di filtro modificato.

## ⚠️ Nota

È disponibile anche la funzione **CALCULATE** che ha esattamente la stessa funzionalità, ad eccezione del fatto che modifica il **contesto di filtro** applicato a un'espressione che restituisce un *valore scalare*.

## Sintassi

DAX

```
CALCULATETABLE(<expression>[, <filter1> [, <filter2> [, ...]]])
```

## Parametri

Termino	Definizione
expression	Espressione di tabella da valutare.
filter1, filter2,...	(Facoltativo) Espressioni booleane o di tabella che definiscono i filtri o le funzioni modificatore di filtro.

L'espressione usata come primo parametro deve essere una tabella modello o una funzione che restituisce una tabella.

I filtri possono essere:

- Espressioni filtro booleane
- Espressioni filtro tabella
- Funzioni di modifica del filtro

Quando sono presenti più filtri, vengono valutati usando l'[operatore logico AND](#). Ciò significa che tutte le condizioni devono essere VERE contemporaneamente.

## Espressioni filtro booleane

Un filtro di espressione booleana è un'espressione che restituisce TRUE o FALSE. Queste espressioni devono rispettare diverse regole:

- Possono fare riferimento solo a una singola colonna.
- Non possono fare riferimento a misure.
- Non possono usare una funzione CALCULATE annidata.

A partire dalla versione di settembre 2021 di Power BI Desktop, si applica anche quanto segue:

- Non possono usare funzioni che analizzano o restituiscono una tabella, a meno che siano passate come argomenti alle funzioni di aggregazione.
- *Possono* contenere una funzione di aggregazione che restituisce un valore scalare.

## Espessione filtro tabella

Un filtro di espressione tabella applica un oggetto tabella come filtro. Può trattarsi di un riferimento a una tabella del modello, ma è più probabile che si tratti di una funzione che restituisce un oggetto tabella. È possibile usare la funzione **FILTER** per applicare condizioni di filtro complesse, incluse quelle che non è possibile definire tramite un'espressione filtro booleana.

## Funzioni di modificatore di filtro

Le funzioni di modificatore di filtro consentono di eseguire altre operazioni oltre alla semplice aggiunta di filtri. Forniscono un maggiore controllo quando si modifica il contesto di filtro.

Funzione	Scopo
REMOVEFILTERS	Rimuovere tutti i filtri oppure i filtri da una o più colonne di una tabella o da tutte le colonne di una singola tabella.
ALL <sup>1</sup> , ALLEXCEPT, ALLNOBLANKROW	Rimuovere i filtri da una o più colonne o da tutte le colonne di una singola tabella.
KEEPFILTERS	Aggiungere un filtro senza rimuovere i filtri esistenti nelle stesse colonne.
USERELATIONSHIP	Creare una relazione inattiva tra colonne correlate. In questo caso, la relazione attiva diventa automaticamente inattiva.
CROSSFILTER	Modificare la direzione del filtro (da entrambe le direzioni a una singola direzione o viceversa) oppure disabilitare una relazione.

<sup>1</sup> La funzione ALL e le relative varianti si comportano sia come modificatori di filtro che come funzioni che restituiscono oggetti tabella. Se la funzione REMOVEFILTERS è supportata dallo strumento in uso, è preferibile usare questa funzione per rimuovere i filtri.

## Valore restituito

Tabella di valori.

## Osservazioni:

- Quando vengono fornite espressioni filtro, la funzione CALCULATETABLE modifica il contesto di filtro per valutare l'espressione. Per ogni espressione filtro, ci sono due possibili risultati standard quando non si esegue il wrapping dell'espressione filtro nella funzione KEEPFILTERS:
  - Se le colonne (o le tabelle) non sono presenti nel contesto di filtro, i nuovi filtri verranno aggiunti al contesto di filtro per valutare l'espressione.
  - Se le colonne (o le tabelle) sono già presenti nel contesto di filtro, i filtri esistenti verranno sovrascritti dai nuovi filtri per valutare l'espressione CALCULATETABLE.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente, la funzione CALCULATETABLE viene usata per ottenere la somma delle vendite Internet per il 2006. Questo valore viene poi usato per calcolare la percentuale delle vendite Internet rispetto a tutte le vendite per l'anno 2006.

La formula seguente:

```
DAX  
= SUMX(  
    CALCULATETABLE(  
        'InternetSales_USD',  
        'DateTime'[CalendarYear] = 2006  
    ),  
    [SalesAmount_USD]  
)
```

Ha come risultato la tabella seguente:

<b>Etichette di riga</b>	<b>Internet SalesAmount_USD</b>	<b>CalculateTable 2006 Internet Sales</b>	<b>Internet Sales to 2006 ratio</b>
2005	\$2,627,031.40	\$5,681,440.58	0,46
2006	\$5,681,440.58	\$5,681,440.58	1.00
2007	\$8,705,066.67	\$5,681,440.58	1.53
2008	\$9,041,288.80	\$5,681,440.58	1,59
Grand Total	\$26,054,827.45	\$5,681,440.58	4.59

## Vedi anche

- [Contesto di filtro](#)
- [Funzione CALCULATE \(DAX\)](#)
- [Funzioni di filtro \(DAX\)](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# EARLIER

Articolo • 20/10/2023

Restituisce il valore corrente della colonna specificata in un passaggio di valutazione esterno della colonna indicata.

EARLIER è utile per i calcoli annidati in cui si vuole usare un determinato valore come input e generare calcoli in base a tale input. In Microsoft Excel è possibile eseguire tali calcoli solo nel contesto della riga corrente. In DAX è invece possibile archiviare il valore dell'input e quindi eseguire il calcolo usando i dati dell'intera tabella.

EARLIER viene usato principalmente nel contesto delle colonne calcolate.

## Sintassi

DAX

`EARLIER(<column>, <number>)`

## Parametri

Termine	Definizione
colonna	Colonna o espressione che viene risolta in una colonna.
num	(Facoltativo) Numero positivo per il passaggio di valutazione esterno.  Il passaggio al livello di valutazione successivo è rappresentato da 1. Il passaggio a due livelli è rappresentato da 2 e così via.  Quando viene omesso, il valore predefinito è 1.

## Valore restituito

Il valore corrente della riga, da **column**, in base al valore di **number** definito per i passaggi di valutazione esterni.

## Eccezioni

Descrizione degli errori

## Osservazioni:

- **EARLIER** ha esito positivo se è presente un contesto di riga prima dell'inizio dell'analisi della tabella. In caso contrario, restituisce un errore.
- Le prestazioni di **EARLIER** potrebbero risultare rallentate in quanto, teoricamente, la funzione potrebbe dovere eseguire un numero di operazioni simile al numero complessivo di righe (nella colonna) moltiplicato per lo stesso numero (a seconda della sintassi dell'espressione). Se, ad esempio, sono presenti 10 righe nella colonna, potrebbero essere necessarie circa 100 operazioni. Se sono presenti 100 righe, potrebbero essere eseguite quasi 10.000 operazioni.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

! **Nota**

In pratica, il motore di analisi in memoria VertiPaq esegue ottimizzazioni per ridurre il numero effettivo di calcoli, ma è necessario prestare attenzione quando si creano formule che coinvolgono la ricorsione.

## Esempio

Per illustrare l'uso di **EARLIER**, è necessario creare uno scenario che calcola un valore di classificazione e quindi usa tale valore in altri calcoli.

L'esempio seguente si basa su questa semplice tabella, **ProductSubcategory**, che mostra le vendite totali per ogni ProductSubcategory.

Di seguito è riportata la tabella finale, inclusa la colonna di classificazione.

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
18	Bib-Shorts	\$156,167.88	18
26	Bike Racks	\$220,720.70	14
27	Bike Stands	\$35,628.69	30
28	Bottles and Cages	\$59,342.43	24
5	Bottom Brackets	\$48,643.47	27
6	Brakes	\$62,113.16	23
19	Cappelli	\$47,934.54	28
7	Modelli a catena	\$8,847.08	35
29	Cleaners	\$16,882.62	32

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
8	Cranksets	\$191,522.09	15
9	Derailleurs	\$64,965.33	22
30	Fenders	\$41,974.10	29
10	Forks	\$74,727.66	21
20	Gloves	\$228,353.58	12
4	Manubrio	\$163,257.06	17
11	Headsets	\$57,659.99	25
31	Helmets	\$451,192.31	9
32	Hydration Packs	\$96,893.78	20
21	Jerseys	\$699,429.78	7
33	Luci		36
34	Locks	\$15,059.47	33
1	Mountain Bikes	\$34,305,864.29	2
12	Mountain Frames	\$4,511,170.68	4
35	Panniers		36
13	Pedals	\$140,422.20	19
36	Pumps	\$12,695.18	34
2	Road Bikes	\$40,551,696.34	1
14	Road Frames	\$3,636,398.71	5
15	Saddles	\$52,526.47	26
22	Pantaloncini	\$385,707.80	10
23	Calzini	\$28,337.85	31
24	Tights	\$189,179.37	16
37	Tires and Tubes	\$224,832.81	13
3	Touring Bikes	\$13,334,864.18	3
16	Touring Frames	\$1,545,344.02	6
25	Vests	\$240,990.04	11
17	Ruote	\$648,240.04	8

# Creazione di un valore di classificazione

Per ottenere un valore di classificazione per un determinato valore in una riga è possibile contare il numero di righe, nella stessa tabella, che hanno un valore più grande (o più piccolo) rispetto a quello che viene confrontato. Questa tecnica restituisce un valore blank o zero per il valore massimo nella tabella, mentre i valori uguali avranno lo stesso valore di classificazione e il valore successivo (dopo i valori uguali) avrà un valore di classificazione non consecutivo. Vedere l'esempio seguente.

Viene creata una nuova colonna calcolata, **SubCategorySalesRanking**, usando la formula seguente.

DAX

```
= COUNTROWS(FILTER(ProductSubcategory,  
EARLIER(ProductSubcategory[TotalSubcategorySales])  
<ProductSubcategory[TotalSubcategorySales]))+1
```

Nei passaggi seguenti viene descritto il metodo di calcolo in modo più dettagliato.

1. La funzione **EARLIER** ottiene il valore di *TotalSubcategorySales* per la riga corrente nella tabella. In questo caso, poiché il processo viene avviato, si tratta della prima riga della tabella
2. **EARLIER([TotalSubcategorySales])** restituisce \$156,167.88, ovvero la riga corrente del ciclo esterno.
3. La funzione **FILTER** restituisce ora una tabella in cui tutte le righe hanno un valore di *TotalSubcategorySales* maggiore di \$156.167,88, ovvero il valore corrente per **EARLIER**.
4. La funzione **COUNTROWS** conta le righe della tabella filtrata e assegna tale valore alla nuova colonna calcolata nella riga corrente più 1. L'aggiunta di 1 è necessaria per evitare che il valore classificato per primo diventi un valore Blank.
5. La formula della colonna calcolata passa alla riga successiva e ripete i passaggi da 1 a 4. Questi passaggi vengono ripetuti finché non viene raggiunta la fine della tabella.

Il funzione **EARLIER** otterrà sempre il valore della colonna prima dell'operazione della tabella corrente. Se è necessario ottenere un valore dal ciclo precedente, impostare il secondo argomento su 2.

## Vedi anche

[Funzione EARLIEST](#)

[Funzioni di filtro](#)

# Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# EARLiest

Articolo • 20/10/2023

Restituisce il valore corrente della colonna specificata in un passaggio di valutazione esterno della colonna specificata.

## Sintassi

DAX

```
EARLiest(<column>)
```

## Parametri

Termine	Definizione
colonna	Un riferimento a una colonna.

## Valore restituito

Colonna con i filtri rimossi.

## Osservazioni:

- La funzione EARLiest è simile a EARLier, ma consente di specificare un livello aggiuntivo di ricorsione.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Questo scenario non è supportato nei dati di esempio correnti.

DAX

```
= EARLiest(<column>)
```

## Vedi anche

[funzione EARLIER](#)

[Funzioni di filtro](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FILTER

Articolo • 20/10/2023

Restituisce una tabella che rappresenta un subset di un'altra tabella o espressione.

## Sintassi

DAX

```
FILTER(<table>,<filter>)
```

## Parametri

Termine	Definizione
table	Tabella a cui applicare filtri. La tabella può essere anche un'espressione tramite cui viene restituita una tabella.
filter	Espressione booleana da valutare per ogni riga della tabella. Ad esempio, <code>[Amount] &gt; 0</code> o <code>[Region] = "France"</code>

## Valore restituito

Tabella che contiene solo le righe filtrate.

## Osservazioni:

- Tramite FILTER è possibile ridurre il numero di righe usate nella tabella e usare solo dati specifici nei calcoli. La funzione FILTER non viene usata in modo indipendente, ma come funzione incorporata in altre funzioni che richiedono una tabella come argomento.
- Per le procedure consigliate quando si usa FILTER, vedere [Evitare di usare FILTER come argomento di filtro](#).

Usare COUNTROWS invece di COUNT in DAX

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

L'esempio seguente crea un report delle vendite Internet fuori dagli Stati Uniti usando una misura che esclude le vendite effettuate negli Stati Uniti e quindi sezionando in base alle categorie di prodotto e all'anno del calendario. Per creare questa misura occorre filtrare la tabella Internet Sales USD tramite Sales Territory e quindi usare la tabella filtrata in una funzione SUMX.

In questo esempio l'espressione:

DAX

```
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])  
<>"United States")
```

Restituisce una tabella che rappresenta un subset delle vendite Internet meno tutte le righe che rappresentano le vendite effettuate negli Stati Uniti. La funzione RELATED collega la chiave Territory della tabella Internet Sales a SalesTerritoryCountry della tabella SalesTerritory.

La tabella seguente è un modello di prova per la misura NON USA Internet Sales, la cui formula è fornita nella sezione di codice seguente. La tabella confronta tutte le vendite Internet con le vendite Internet non realizzate negli Stati Uniti per mostrare che l'espressione di filtro funziona tramite l'esclusione delle vendite degli Stati Uniti dal calcolo.

Per ricreare questa tabella, aggiungere il campo SalesTerritoryCountry all'area **Etichette di riga** di un report o di una tabella pivot.

## Tabella 1. Confronto delle vendite totali degli Stati Uniti rispetto alle altre aree geografiche

Etichette di riga	Internet Sales	Non USA Internet Sales
Australia	\$4,999,021.84	\$4,999,021.84
Canada	\$1,343,109.10	\$1,343,109.10
Francia	\$2,490,944.57	\$2,490,944.57
Germania	\$2,775,195.60	\$2,775,195.60
Regno Unito	\$5,057,076.55	\$5,057,076.55
Stati Uniti	\$9,389,479.79	

<b>Etichette di riga</b>	<b>Internet Sales</b>	<b>Non USA Internet Sales</b>
Grand Total	\$26,054,827.45	\$16,665,347.67

La tabella del report finale mostra i risultati quando si crea una tabella pivot usando la misura NON USA Internet Sales. Aggiungere il campo CalendarYear all'area Etichette di riga della tabella pivot, quindi aggiungere il campo ProductCategoryName all'area Etichette di colonna.

## Tabella 2. Confronto delle vendite non effettuate negli Stati Uniti in base alla categoria di prodotto

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Grand Total</b>
2005		\$1,526,481.95		\$1,526,481.95
2006		\$3,554,744.04		\$3,554,744.04
2007	\$156,480.18	\$5,640,106.05	\$70,142.77	\$5,866,729.00
2008	\$228,159.45	\$5,386,558.19	\$102,675.04	\$5,717,392.68
Grand Total	\$384,639.63	\$16,107,890.23	\$172,817.81	\$16,665,347.67

DAX

```
SUMX(FILTER('InternetSales_USD',
RELATED('SalesTerritory'[SalesTerritoryCountry]))<>"United States")
,'InternetSales_USD'[SalesAmount_USD])
```

## Vedi anche

[Funzioni di filtro](#)

[Funzione ALL](#)

[Funzione ALLEXCEPT](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# INDEX

Articolo • 20/10/2023

Restituisce una riga in una posizione assoluta, specificata dal parametro di posizione, all'interno della partizione specificata, ordinata in base all'ordine specificato. Se non è possibile dedurre la partizione corrente in una singola partizione, è possibile che vengano restituite più righe.

## Sintassi

DAX

```
INDEX(<position>[, <relation>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>])
```

## Parametri

Termine	Definizione
località	La posizione assoluta (in base 1) da cui ottenere i dati: <ul style="list-style-type: none"><li>- &lt;posizione&gt; è positiva: 1 è la prima riga, 2 è la seconda riga e così via.</li><li>- &lt;posizione&gt; è negativa: -1 è l'ultima riga, -2 è la seconda ultima riga e così via.</li></ul> Quando la <posizione> non rientra nel limite o zero o BLANK(), INDEX restituirà una tabella vuota. Può essere qualsiasi espressione DAX che restituisce un valore scalare.
relazione	(Facoltativo) Espressione di tabella da cui viene restituito l'output. Se specificato, tutte le colonne in <partitionBy> devono provenire da essa o da una tabella correlata. Se omesso: <ul style="list-style-type: none"><li>- &lt;orderBy&gt; deve essere specificato in modo esplicito.</li><li>- Tutte le espressioni &lt;orderBy&gt; e &lt;partitionBy&gt; devono essere nomi di colonna completi e provenire da una singola tabella.</li><li>- L'impostazione predefinita è ALLSELECTED() di tutte le colonne in &lt;orderBy&gt; e &lt;partitionBy&gt;.</li></ul>
orderBy	(Facoltativo) Clausola ORDERBY() contenente le colonne che definiscono la modalità di ordinamento di ogni partizione. Se omesso: <ul style="list-style-type: none"><li>- &lt;la relazione&gt; deve essere specificata in modo esplicito.</li><li>- Per impostazione predefinita, l'ordinamento viene eseguito in base a ogni colonna in &lt;relazione&gt; non già specificata in &lt;partitionBy&gt;.</li></ul>
valori blank	(Facoltativo) Enumerazione che definisce come gestire i valori (blank) durante l'ordinamento. Questo parametro è riservato per usi futuri.

Termine	Definizione
	Attualmente, l'unico valore supportato è DEFAULT, in cui il comportamento per i valori numerici è vuoto viene ordinato tra zero e valori negativi. Il comportamento per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote.
partitionBy	(Facoltativo) Clausola PARTITIONBY() contenente le colonne che definiscono la modalità <di partizionamento della relazione>. Se omesso, <la relazione> viene considerata come una singola partizione.
matchBy	(Facoltativo) Clausola MATCHBY() contenente le colonne che definiscono come trovare le corrispondenze con i dati e identificare la riga corrente.

## Valore restituito

Riga in una posizione assoluta.

## Osservazioni:

Ogni colonna <partitionBy> e <matchBy> deve avere un valore esterno corrispondente per definire la "partizione corrente" su cui operare, con il comportamento seguente:

- Se è presente esattamente una colonna esterna corrispondente, viene usato il relativo valore.
- Se non è presente alcuna colonna esterna corrispondente:
  - INDEX determinerà innanzitutto tutte le colonne <partitionBy>, <matchBy> che non dispongono di colonne esterne corrispondenti.
  - Per ogni combinazione di valori esistenti per queste colonne nel contesto padre INDEX, INDEX viene valutata e viene restituita una riga.
  - L'output finale di INDEX è un'unione di queste righe.
- Se sono presenti più colonne esterne corrispondenti, viene restituito un errore.

Se <matchBy> è presente, INDEX tenterà di usare le colonne <matchBy> e <partitionBy> per identificare la riga.

Se <matchBy> non è presente e le colonne specificate all'interno di <orderBy> e <partitionBy> non possono identificare in modo univoco ogni riga in <relazione>:

- INDEX tenterà di trovare il minor numero di colonne aggiuntive necessarie per identificare in modo univoco ogni riga.
- Se è possibile trovare tali colonne, INDEX accorderà automaticamente queste nuove colonne a <orderBy> e ogni partizione viene ordinata usando questo nuovo set di colonne OrderBy.
- Se non è possibile trovare tali colonne, viene restituito un errore.

Viene restituita una tabella vuota se:

- Il valore esterno corrispondente di una colonna PartitionBy non esiste all'interno della <relazione>.
- Il valore di <posizione> fa riferimento a una posizione che non esiste all'interno della partizione.

Se INDEX viene utilizzato all'interno di una colonna calcolata definita nella stessa tabella della <relazione> e <orderBy> viene omesso, viene restituito un errore.

## Esempio 1

La query DAX seguente:

```
DAX  
EVALUATE INDEX(1, ALL(DimDate[CalendarYear]))
```

Restituisce la tabella seguente:

DimDate[CalendarYear]
2005

## Esempio 2

La query DAX seguente:

```
DAX  
EVALUATE  
SUMMARIZECOLUMNS (  
    FactInternetSales[ProductKey],  
    DimDate[MonthNumberOfYear],  
    FILTER (  
        VALUES(FactInternetSales[ProductKey]),  
        [ProductKey] < 222  
    ),  
    "CurrentSales", SUM(FactInternetSales[SalesAmount]),  
    "LastMonthSales",  
    CALCULATE (  
        SUM(FactInternetSales[SalesAmount]),  
        INDEX(-1, ORDERBY(DimDate[MonthNumberOfYear]))  
    )  
)  
ORDER BY [ProductKey], [MonthNumberOfYear]
```

Restituisce la tabella seguente:

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
214	1	5423.45	8047.7
214	2	4968.58	8047.7
214	3	5598.4	8047.7
214	4	5073.55	8047.7
214	5	5248.5	8047.7
214	6	7487.86	8047.7
214	7	7382.89	8047.7
214	8	6543.13	8047.7
214	9	6788.06	8047.7
214	10	6858.04	8047.7
214	11	8607.54	8047.7
214	12	8047.7	8047.7
217	1	5353.47	7767.78
217	2	4268.78	7767.78
217	3	5773.35	7767.78
217	4	5738.36	7767.78
217	5	6158.24	7767.78
217	6	6998	7767.78
217	7	5563.41	7767.78
217	8	5913.31	7767.78
217	9	5913.31	7767.78
217	10	6823.05	7767.78
217	11	6683.09	7767.78
217	12	7767.78	7767.78

Vedi anche

OFFSET

ORDERBY

PARTITIONBY

WINDOW

RANK

ROWNUMBER

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# KEEPFILTERS

Articolo • 20/10/2023

Modifica la modalità di applicazione dei filtri durante la valutazione di una funzione CALCULATE o CALCULATETABLE.

## Sintassi

DAX

```
KEEPFILTERS(<expression>)
```

## Parametri

Termine	Definizione
expression	qualsiasi espressione.

## Valore restituito

Tabella di valori.

## Osservazioni:

- La funzione KEEPFILTERS si usa nel contesto delle funzioni CALCULATE e CALCULATETABLE per eseguire l'override del comportamento standard di queste funzioni.
- Per impostazione predefinita, gli argomenti di filtro nelle funzioni come CALCULATE sono usati come contesto per la valutazione dell'espressione e, in quanto tali, gli argomenti di filtro per CALCULATE sostituiscono tutti i filtri esistenti sulle stesse colonne. Il nuovo contesto applicato dall'argomento di filtro per CALCULATE influenza solo sui filtri esistenti sulle colonne indicate come parte dell'argomento di filtro. I filtri su colonne diverse da quelle indicate negli argomenti di CALCULATE o di altre funzioni correlate rimangono validi e non vengono modificati.
- La funzione KEEPFILTERS consente di modificare questo comportamento. Quando si usa KEEPFILTERS, qualsiasi filtro esistente nel contesto corrente viene

confrontato con le colonne negli argomenti di filtro e l'intersezione di tali argomenti viene usata come contesto per valutare l'espressione. Il risultato finale su ogni colonna è che vengono applicati sia gli argomenti di filtro usati nella funzione CALCULATE sia i filtri negli argomenti della funzione KEEPFILTERS. In altre parole, mentre i filtri di CALCULATE sostituiscono il contesto corrente, KEEPFILTERS aggiunge filtri al contesto corrente.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra alcuni scenari comuni che dimostrano l'utilizzo della funzione KEEPFILTERS come parte di una formula CALCULATE o CALCULATETABLE.

Le prime tre espressioni ottengono dati semplici da usare per i confronti:

- Vendite Internet per lo stato di Washington.
- Vendite Internet per gli stati di Washington e Oregon (combinati).
- Vendite Internet per lo stato di Washington e la provincia della British Columbia (entrambe le aree combinate).

La quarta espressione calcola le vendite Internet per gli stati di Washington e Oregon, mentre viene applicato il filtro per Washington e British Columbia.

L'espressione successiva calcola le vendite Internet per gli stati di Washington e Oregon, ma usa KEEPFILTERS; il filtro per Washington e British Columbia fa parte del contesto precedente.

DAX

```
EVALUATE ROW(
    "$$ in WA"
    , CALCULATE('Internet Sales'[Internet Total Sales]
        , 'Geography'[State Province Code]="WA"
    )
    , "$$ in WA and OR"
    , CALCULATE('Internet Sales'[Internet Total Sales]
        , 'Geography'[State Province Code]="WA"
        || 'Geography'[State Province Code]="OR"
    )
    , "$$ in WA and BC"
    , CALCULATE('Internet Sales'[Internet Total Sales]
        , 'Geography'[State Province Code]="WA"
        || 'Geography'[State Province Code]="BC"
    )
)
```

```

        )
    , " $$ in WA and OR ??"
    , CALCULATE(
        CALCULATE('Internet Sales'[Internet Total Sales]
            , 'Geography'[State Province Code]="WA"
            || 'Geography'[State Province Code]="OR"
        )
    , 'Geography'[State Province Code]="WA"
    || 'Geography'[State Province Code]="BC"
)
, " $$ in WA !!"
, CALCULATE(
    CALCULATE('Internet Sales'[Internet Total Sales]
        , KEEPFILTERS('Geography'[State Province Code]="WA"
        || 'Geography'[State Province Code]="OR"
    )
)
, 'Geography'[State Province Code]="WA"
|| 'Geography'[State Province Code]="BC"
)
)
)
)

```

Quando questa espressione viene valutata rispetto al database di esempio AdventureWorks DW, si ottengono i risultati seguenti.

Colonna	Valore
[\$\$ in WA]	\$ 2.467.248,34
[\$\$ in WA and OR]	\$ 3.638.239,88
[\$\$ in WA and BC]	\$ 4.422.588,44
[\$\$ in WA and OR ??]	\$ 3.638.239,88
[\$\$ in WA !!!]	\$ 2.467.248,34

### ⓘ Nota

I risultati riportati sopra sono stati formattati come tabella, anziché in una singola riga, a scopo didattico.

Esaminare prima di tutto l'espressione **[\$\$ in WA and OR ??]**. Ci si potrebbe chiedere come sia possibile che questa formula restituisca il valore per le vendite negli stati di Washington e Oregon, dato che l'espressione CALCULATE esterna include un filtro per Washington e British Columbia. La risposta è che il comportamento predefinito di CALCULATE esegue l'override dei filtri esterni in 'Geography'[State Province Code] e sostituisce gli argomenti di filtro, perché i filtri si applicano alla stessa colonna.

Esaminare quindi l'espressione `[$$ in WA !!]`. Anche in questo caso ci si potrebbe chiedere come sia possibile che questa formula restituisca il valore solo per le vendite nello stato di Washington, dato che l'argomento di filtro include Oregon e l'espressione CALCULATE esterna include un filtro per Washington e British Columbia. La risposta è che KEEPFILTERS modifica il comportamento predefinito di CALCULATE e aggiunge un ulteriore filtro. Poiché viene usata l'intersezione dei filtri, ora il filtro esterno `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC"` viene aggiunto all'argomento di filtro `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR"`. Dal momento che entrambi i filtri si applicano alla stessa colonna, il filtro risultante `'Geography'[State Province Code]="WA"` è il filtro che viene applicato quando viene valutata l'espressione.

## Vedi anche

[Funzioni di filtro](#)

[Funzione CALCULATE](#)

[Funzione CALCULATETABLE](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# LOOKUPVALUE

Articolo • 20/10/2023

Restituisce il valore per la riga che soddisfa tutti i criteri specificati da una o più condizioni di ricerca.

## Sintassi

DAX

```
LOOKUPVALUE (
    <result_columnName>,
    <search_columnName>,
    <search_value>
    [, <search2_columnName>, <search2_value>]...
    [, <alternateResult>]
)
```

## Parametri

Termine	Definizione
result_columnName	Nome di una colonna esistente contenente i valori da restituire. Non può essere un'espressione.
search_columnName	Nome della colonna esistente. Può trovarsi nella stessa tabella di result_columnName o in una tabella correlata. Non può essere un'espressione. È possibile specificare più coppie di search_columnName e search_value.
search_value	Valore da cercare in search_columnName. È possibile specificare più coppie di search_columnName e search_value.
alternateResult	(Facoltativo) Valore restituito quando il contesto per result_columnName è stato filtrato fino a zero o a più di un valore distinto. Se non specificato, la funzione restituisce BLANK quando result_columnName viene filtrato fino a valori zero o un errore quando è presente più di un valore distinto nel contesto per result_columnName.

## Valore restituito

Valore del parametro **result\_columnName** in corrispondenza della riga in cui per tutte le coppie di **search\_columnName** e **search\_value** esiste una corrispondenza esatta.

Se non è presente alcuna corrispondenza che soddisfa tutti i valori di ricerca, viene restituito BLANK o **alternateResult**, se specificato. In altre parole, la funzione non restituisce un valore di ricerca se vengono soddisfatti solo alcuni dei criteri.

Se più righe corrispondono ai valori di ricerca e i valori in **resultColumnName** per queste righe sono identici, viene restituito questo valore. Se invece **resultColumnName** restituisce valori diversi, viene restituito un errore o **alternateResult**, se specificato.

## Osservazioni:

- Se esiste una relazione tra la tabella che contiene la colonna dei risultati e le tabelle che contengono le colonne di ricerca, nella maggior parte dei casi l'uso della funzione **RELATED** al posto di LOOKUPVALUE è più efficiente e offre prestazioni migliori.
- È possibile specificare più coppie di **searchColumnName** e **search\_value**.
- I parametri **search\_value** e **alternateResult** vengono valutati prima che la funzione esegua l'iterazione delle righe della tabella di ricerca.
- Evitare di usare funzioni ISERROR o IFERROR per acquisire un errore restituito da LOOKUPVALUE. Se alcuni input per la funzione generano un errore quando non è possibile determinare un singolo valore di output, specificare un parametro **alternateResult** rappresenta il modo più veloce e affidabile per gestire l'errore.
- Il parametro **alternateResult** restituisce un errore se specificato in una colonna calcolata di Power Pivot.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

In questo esempio, LOOKUPVALUE viene usato per cercare il tasso medio (Average Rate) per la valuta utilizzata per pagare l'ordine nel giorno in cui è stato effettuato l'ordine:

DAX

```
Exchange Rate =  
LOOKUPVALUE (  
    'Currency Rate'[Average Rate],  
    'Currency Rate'[CurrencyKey], [CurrencyKey],  
    'Currency Rate'[DateKey], [OrderDateKey]
```

)

Per trovare il tasso medio per la data e la valuta corrette sono necessari sia la data dell'ordine che la valuta. OrderDateKey e CurrencyKey sono le chiavi usate per cercare il tasso medio nella tabella Currency Rate.

È possibile utilizzare il tasso di cambio (Exchange Rate) per calcolare l'importo delle vendite nella valuta locale con:

DAX

```
Sales Amount Local Currency = [Sales Amount] * [Exchange Rate]
```

## Esempio 2

In questo esempio, nella colonna calcolata seguente definita nella tabella **Sales** viene usata la funzione LOOKUPVALUE per restituire i valori del canale dalla tabella **Sales Order**.

DAX

```
CHANNEL =  
LOOKUPVALUE (  
    'Sales Order'[Channel],  
    'Sales Order'[SalesOrderLineKey],  
    [SalesOrderLineKey]  
)
```

Tuttavia, poiché in questo caso esiste una relazione tra le tabelle **Sales Order** e **Sales**, è più efficiente usare la funzione **RELATED**.

DAX

```
CHANNEL = RELATED('Sales Order'[Channel])
```

## Vedi anche

[Funzione RELATED \(DAX\)](#)

[Funzioni informative](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# MATCHBY

Articolo • 20/10/2023

Nelle funzioni finestra definisce le colonne utilizzate per determinare come trovare le corrispondenze con i dati e identificare la riga corrente.

## Sintassi

DAX

```
MATCHBY ( [<matchBy_columnName>[, matchBy_columnName [, ...]]] )
```

## Parametri

Termine	Definizione
matchBy_columnName	(Facoltativo) Nome di una colonna esistente da utilizzare per identificare la riga corrente nella <relazione> della funzione finestra. RELATED() può essere usato anche per fare riferimento a una colonna in una tabella correlata alla <relazione>.

## Valore restituito

Questa funzione non restituisce un valore.

## Osservazioni:

Questa funzione può essere usata solo all'interno di un'espressione di funzione window.

## Esempio

Vedere [OFFSET](#).

## Vedi anche

[Informazioni sulle funzioni ORDERBY, PARTITIONBY e MATCHBY INDEX](#)  
[OFFSET](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# OFFSET

Articolo • 20/10/2023

Restituisce una singola riga posizionata prima o dopo la *riga corrente* all'interno della stessa tabella da un offset specificato. Se non è possibile dedurre la riga corrente in una singola riga, è possibile che vengano restituite più righe.

## Sintassi

DAX

```
OFFSET ( <delta>[, <relation>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>] )
```

## Parametri

Termino	Definizione
delta	Numero di righe prima (valore negativo) o dopo (valore positivo) la riga corrente da cui ottenere i dati. Può essere qualsiasi espressione DAX che restituisce un valore scalare.
relazione	(Facoltativo) Espressione di tabella da cui viene restituita la riga di output. Se specificato, tutte le colonne in <partitionBy> devono provenire da essa o da una tabella correlata. Se omesso: <ul style="list-style-type: none"><li>- &lt;orderBy&gt; deve essere specificato in modo esplicito.</li><li>- Tutte le colonne &lt;orderBy&gt; e &lt;partitionBy&gt; devono essere complete e provenire da una singola tabella.</li><li>- L'impostazione predefinita è ALLSELECTED() di tutte le colonne in &lt;orderBy&gt; e &lt;partitionBy&gt;.</li></ul>
orderBy	(Facoltativo) Clausola ORDERBY() contenente le colonne che definiscono la modalità di ordinamento di ogni partizione. Se omesso: <ul style="list-style-type: none"><li>- &lt;la relazione&gt; deve essere specificata in modo esplicito.</li><li>- Per impostazione predefinita, l'ordinamento viene eseguito in base a ogni colonna in &lt;relazione&gt; non già specificata in &lt;partitionBy&gt;.</li></ul>
valori blank	(Facoltativo) Enumerazione che definisce come gestire i valori (blank) durante l'ordinamento. Questo parametro è riservato per usi futuri. Attualmente, l'unico valore supportato è DEFAULT, in cui il comportamento per i valori numerici è vuoto viene ordinato tra zero e valori negativi. Il comportamento

Termine	Definizione
	per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote.
partitionBy	(Facoltativo) Clausola PARTITIONBY() contenente le colonne che definiscono la modalità <di partizionamento della relazione>. Se omesso, <la relazione> viene considerata come una singola partizione.
matchBy	(Facoltativo) Clausola MATCHBY() contenente le colonne che definiscono come trovare le corrispondenze con i dati e identificare la riga corrente.

## Valore restituito

Una o più righe dalla <relazione>.

## Osservazioni:

Ad eccezione delle colonne aggiunte dalle funzioni di tabella DAX, ogni colonna in <relazione>, quando <matchBy> non è presente, o ogni colonna in <matchBy> e <partitionBy>, quando <matchBy> è presente, deve avere un valore esterno corrispondente per definire la riga corrente su cui operare, con il comportamento seguente:

- Se è presente esattamente una colonna esterna corrispondente, viene usato il relativo valore.
- Se non è presente alcuna colonna esterna corrispondente, eseguire le operazioni seguenti:
  - OFFSET determinerà innanzitutto tutte le colonne che non dispongono di una colonna esterna corrispondente.
  - Per ogni combinazione di valori esistenti per queste colonne nel contesto padre di OFFSET, OFFSET viene valutato e viene restituita una riga.
  - L'output finale di OFFSET è un'unione di queste righe.
- Se sono presenti più colonne esterne corrispondenti, viene restituito un errore.

Se tutte le colonne della <relazione> sono state aggiunte dalle funzioni di tabella DAX, viene restituito un errore.

Se <matchBy> è presente, OFFSET tenterà di usare le colonne <matchBy> e <partitionBy> per identificare la riga.

Se <matchBy> non è presente e le colonne specificate all'interno di <orderBy> e <partitionBy> non possono identificare in modo univoco ogni riga in <relazione>, allora:

- OFFSET tenterà di trovare il minor numero di colonne aggiuntive necessarie per identificare in modo univoco ogni riga.
- Se è possibile trovare tali colonne, OFFSET aggiungerà automaticamente queste nuove colonne a <orderBy> e ogni partizione viene ordinata usando questo nuovo set di colonne OrderBy.
- Se non è possibile trovare tali colonne, viene restituito un errore.

Viene restituita una tabella vuota se:

- Il valore esterno corrispondente di una colonna OrderBy o PartitionBy non esiste all'interno della <relazione>.
- Il valore <delta> causa uno spostamento in una riga che non esiste all'interno della partizione.

Se OFFSET viene utilizzato all'interno di una colonna calcolata definita nella stessa tabella della <relazione>, e <orderBy> viene omesso, allora viene restituito un errore.

## Esempio 1

La query DAX seguente:

```
DAX

DEFINE
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProductCategory[EnglishProductName],
        DimDate[CalendarYear],
        "CurrentYearSales", SUM(FactInternetSales[SalesAmount])
    )
EVALUATE
ADDCOLUMNS (
    vRelation,
    "PreviousYearSales",
    SELECTCOLUMNS(
        OFFSET (
            -1,
            vRelation,
            ORDERBY([CalendarYear]),
            PARTITIONBY([EnglishProductName])
        ),
        [CurrentYearSales]
    )
)
```

Restituisce una tabella che riepiloga le vendite totali per ogni categoria di prodotto e anno di calendario, nonché le vendite totali per tale categoria nell'anno precedente.

## Esempio 2

La query DAX seguente:

```
DAX

DEFINE
MEASURE DimProduct[CurrentYearSales] = SUM(FactInternetSales[SalesAmount])
MEASURE DimProduct[PreviousYearSales] =
CALCULATE(SUM(FactInternetSales[SalesAmount]), OFFSET(-1, ,
ORDERBY(DimDate[CalendarYear])))
EVALUATE
SUMMARIZECOLUMNS (
    DimDate[CalendarYear],
    "CurrentYearSales", DimProduct[CurrentYearSales],
    "PreviousYearSales", DimProduct[PreviousYearSales]
)
```

Usa OFFSET() in una misura per restituire una tabella che riepiloga le vendite totali per ogni anno di calendario e le vendite totali per l'anno precedente.

## Esempio 3

La query DAX seguente:

```
DAX

EVALUATE
ADDCOLUMNS (
    FactInternetSales,
    "Previous Sales Amount",
    SELECTCOLUMNS (
        OFFSET (
            -1,
            FactInternetSales,
            ORDERBY ( FactInternetSales[SalesAmount], DESC ),
            PARTITIONBY ( FactInternetSales[ProductKey] ),
            MATCHBY( FactInternetSales[SalesOrderNumber],
FactInternetSales[SalesOrderLineNumber] )
        ),
        FactInternetSales[SalesAmount]
    )
)
```

Restituisce la tabella FactInternetSales con l'aggiunta di una colonna che indica, per ogni vendita, l'importo della vendita precedente, dello stesso prodotto, in ordine decrescente dell'importo delle vendite, con la vendita corrente identificata dal relativo

SalesOrderNumber e SalesOrderLineNumber. Senza MATCHBY, la query restituirà un errore perché non sono presenti colonne chiave nella tabella FactInternetSales.

## Vedi anche

[INDEX](#)  
[ORDERBY](#)  
[PARTITIONBY](#)  
[MATCHBY](#)  
[WINDOW](#)  
[RANK](#)  
[ROWNUMBER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ORDERBY

Articolo • 20/10/2023

Definisce le espressioni che determinano l'ordinamento all'interno di ognuna delle partizioni di una funzione di finestra.

## Sintassi

DAX

```
ORDERBY ( [<orderBy_expression>[, <order>[, <orderBy_expression>[, <order>]] ...]] )
```

## Parametri

Termine	Definizione
orderBy_expression	(Facoltativo) Qualsiasi espressione scalare che verrà usata per ordinare i dati all'interno di ognuna delle partizioni di una funzione finestra.
order	(Facoltativo) Valore in due parti del formato "<OrderDirection> [<BlankHandling>]".  <OrderDirection> specifica come ordinare valori <orderBy_expression>, ad esempio crescente o decrescente. I valori validi includono: <b>DESC</b> . Valore alternativo: 0(zero)/ <b>FALSE</b> . Ordinamenti in senso decrescente dei valori di <orderBy_expression>. <b>ASC</b> . Valore alternativo: 1/ <b>TRUE</b> . Ordinamenti in senso decrescente dei valori di <orderBy_expression>. Questo è il valore predefinito se <order> viene omesso.  La parte <BlankHandling> è facoltativa. Specifica la modalità di ordinamento dei valori vuoti. I valori validi includono: <b>BLANKS DEFAULT</b> . Si tratta del valore predefinito. Il comportamento per i valori numerici è costituito da valori vuoti ordinati tra zero e valori negativi. Il comportamento per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote. <b>BLANKS FIRST</b> . I valori blank vengono sempre ordinati all'inizio, indipendentemente dall'ordinamento crescente o decrescente. <b>BLANKS LAST</b> . I valori blank vengono sempre ordinati alla fine, indipendentemente dall'ordinamento crescente o decrescente.

## Valore restituito

Questa funzione non restituisce un valore.

## Osservazioni:

Questa funzione può essere usata solo all'interno di un'espressione di funzione window.

## Esempio

Vedere [OFFSET](#).

## Vedi anche

[Informazioni sulle funzioni ORDERBY, PARTITIONBY e MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[PARTITIONBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PARTITIONBY

Articolo • 20/10/2023

Definisce le colonne utilizzate per partizionare il parametro di <relazione> di una funzione finestra.

## Sintassi

DAX

```
PARTITIONBY ( [<partitionBy_columnName>[ , partitionBy_columnName [ , ...]]] )
```

## Parametri

Termine	Definizione
partitionBy_columnName	(Facoltativo) Nome di una colonna esistente da utilizzare per partizionare la <relazione> della funzione finestra. RELATED() può essere usato anche per fare riferimento a una colonna in una tabella correlata alla <relazione>.

## Valore restituito

Questa funzione non restituisce un valore.

## Osservazioni:

Questa funzione può essere usata solo all'interno di un'espressione di funzione window.

## Esempio

Vedere [OFFSET](#).

## Vedi anche

[Informazioni sulle funzioni ORDERBY, PARTITIONBY e MATCHBY](#)  
[INDEX](#)  
[OFFSET](#)

[ORDERBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# RANK

Articolo • 20/10/2023

Restituisce la classificazione per il contesto corrente all'interno della partizione specificata, ordinata in base all'ordine specificato. Se non è possibile trovare una corrispondenza, la classificazione è vuota.

## Sintassi

DAX

```
RANK ( [<ties>][, <relation>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>] )
```

## Parametri

Termino	Definizione
tà	(Facoltativo) Definisce come gestire la classificazione quando due o più righe sono associate. Se specificato, il valore supportato è DENSE o SKIP. Se omesso: <ul style="list-style-type: none"><li>- Da Impostazione predefinita a IGNORA</li></ul>
relazione	(Facoltativo) Espressione di tabella da cui viene restituita la riga di output. Se specificato, tutte le colonne in <orderBy> e <partitionBy> devono provenire da essa. Se omesso: <ul style="list-style-type: none"><li>- &lt;orderBy&gt; deve essere specificato in modo esplicito.</li><li>- Tutte le colonne &lt;orderBy&gt; e &lt;partitionBy&gt; devono essere complete e provenire da una singola tabella.</li><li>- L'impostazione predefinita è ALLSELECTED() di tutte le colonne in &lt;orderBy&gt; e &lt;partitionBy&gt;.</li></ul>
orderBy	(Facoltativo) Clausola ORDERBY() contenente le colonne che definiscono la modalità di ordinamento di ogni partizione. Se omesso: <ul style="list-style-type: none"><li>- la &lt;relazione&gt; deve essere specificata in modo esplicito.</li><li>- Per impostazione predefinita, l'ordinamento viene eseguito in base a ogni colonna in &lt;relazione&gt; non già specificata in &lt;partitionBy&gt;.</li></ul>
valori blank	(Facoltativo) Enumerazione che definisce come gestire i valori (blank) durante l'ordinamento. I valori supportati sono:

Termino	Definizione
	<ul style="list-style-type: none"> <li>• L'IMPOSTAZIONE PREDEFINITA (valore predefinito), laddove il comportamento per i valori numerici è vuoto, i valori vengono ordinati tra zero e valori negativi. Il comportamento per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote.</li> <li>• PRIMI, i valori blank vengono sempre ordinati all'inizio, indipendentemente dall'ordinamento crescente o decrescente.</li> <li>• ULTIMI, i valori blank vengono sempre ordinati alla fine, indipendentemente dall'ordinamento crescente o decrescente.</li> </ul> <p>Si noti che quando vengono specificati parametri &lt;blank&gt; e blank nella funzione ORDERBY() in singole espressioni, i &lt;blank&gt; per ogni espressione orderBy hanno la priorità per l'espressione orderBy pertinente e le espressioni orderBy senza &lt;blank&gt; specificate rispettano il parametro &lt;blank&gt; nella funzione Window principale.</p>
partitionBy	(Facoltativo) Clausola PARTITIONBY() contenente le colonne che definiscono la modalità <di partizionamento della relazione>. Se omesso, <la relazione> viene considerata come una singola partizione.
matchBy	(Facoltativo) Clausola MATCHBY() contenente le colonne che definiscono come trovare le corrispondenze con i dati e identificare la riga corrente.

## Valore restituito

Numero di classificazione per il contesto corrente.

## Osservazioni:

- Ogni colonna <orderBy>, <partitionBy> e <matchBy> deve avere un valore esterno corrispondente per definire la riga corrente su cui operare, con il comportamento seguente:
  - Se è presente esattamente una colonna esterna corrispondente, viene usato il relativo valore.
  - Se non è presente alcuna colonna esterna corrispondente, eseguire le operazioni seguenti:
    - RANK determinerà innanzitutto tutte le colonne <orderBy>, <partitionBy> e <matchBy> che non dispongono di colonne esterne corrispondenti.
    - Per ogni combinazione di valori esistenti per queste colonne nel contesto padre RANK, RANK viene valutata e viene restituita una riga.
    - L'output finale di RANK è un numero di classificazione.

- Se <matchBy> è presente, RANK tenterà di usare le colonne in <matchBy> e <partitionBy> per identificare la riga corrente.
- Se le colonne specificate all'interno <di orderBy> e <partitionBy> non possono identificare in modo univoco ogni riga in <relazione>, è possibile che due o più righe abbiano la stessa classificazione e che la classificazione venga determinata dal parametro ties.
- RANK restituisce un valore blank per le righe totali. È consigliabile testare accuratamente l'espressione.
- RANK non viene confrontato con RANKX come SUM viene confrontato a SUMX.

## Esempio

La query DAX seguente:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "Rank",
    RANK(
        DENSE,
        'DimGeography'[StateProvinceName], desc,
        'DimGeography'[City], asc),
    LAST,
    PARTITIONBY(
        'DimGeography'[EnglishCountryRegionName]))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]
asc
```

Restituisce una tabella che classifica ogni area geografica con la stessa proprietà EnglishCountryRegionName, in base a StateProvinceName e City. I valori di colonna <orderBy> blank vengono ordinati alla fine.

## Vedi anche

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# REMOVEFILTERS

Articolo • 20/10/2023

Cancellare i filtri dalle tabelle o dalle colonne specificate.

## Sintassi

DAX

```
REMOVEFILTERS([<table> | <column>[, <column>[, <column>[,...]]]])
```

## Parametri

Termine	Definizione
table	Tabella in cui cancellare i filtri.
colonna	Colonna in cui cancellare i filtri.

## Valore restituito

N/D. Vedere la sezione Osservazioni.

## Osservazioni:

- Si può usare REMOVEFILTERS solo per cancellare i filtri, ma non per restituire una tabella.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Query DAX

DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS()))
EVALUATE
SUMMARIZECOLUMNS(
ROLLUPADDISSTOTAL(DimProductCategory[EnglishProductName],
"IsGrandTotal"),
```

```

        "TotalSales", [TotalSales],
        "%Sales", [%Sales]
    )
ORDER BY
    [IsGrandTotal] DESC, [TotalSales] DESC

```

Resi

<b>DimProductCategory[EnglishProductCategoryName]</b>	<b>[IsGrandTotal]</b>	<b>[TotalSales]</b>	<b>[%Sales]</b>
Riga1	Vero	29358677.2207	1
Bikes	False	28318144.6507	0.964557920570538
Accessori	False	700759.96	0.023868921434441
Clothing	False	339772.61	0.0115731579950215

## Esempio 2

Query DAX

```

DAX

DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategory
Name])))
EVALUATE
SUMMARIZECOLUMNS(
    DimProductCategory[EnglishProductCategoryName],
    DimProductSubcategory[EnglishProductSubcategoryName],
    "TotalSales", [TotalSales],
    "%Sales", [%Sales]
)
ORDER BY
    DimProductCategory[EnglishProductCategoryName] ASC,
    DimProductSubcategory[EnglishProductSubcategoryName] ASC

```

Resi

<b>DimProductCategory [EnglishProductCategoryName]</b>	<b>DimProductSubcategory [EnglishProductSubcategoryName]</b>	<b>[TotalSales]</b>	<b>[%Sales]</b>
Accessori	Bike Racks	39360	0.05616759
Accessori	Bike Stands	39591	0.05649723
Accessori	Bottles and Cages	56798.19	0.08105228
Accessori	Cleaners	7218.6	0.0103011

<b>DimProductCategory</b> [EnglishProductCategoryName]	<b>DimProductSubcategory</b> [EnglishProductSubcategoryName]	<b>[TotalSales]</b>	<b>[%Sales]</b>
Accessori	Fenders	46619.58	0.06652717
Accessori	Helmets	225335.6	0.3215589
Accessori	Hydration Packs	40307.67	0.05751994
Accessori	Tires and Tubes	245529.32	0.35037578
Bikes	Mountain Bikes	9952759.564	0.35146228
Bikes	Road Bikes	14520584.04	0.51276608
Bikes	Touring Bikes	3844801.05	0.13577164
Clothing	Cappelli	19688.1	0.05794493
Clothing	Gloves	35020.7	0.10307099
Clothing	Jerseys	172950.68	0.5090189
Clothing	Pantalocini	71319.81	0.20990453
Clothing	Calzini	5106.32	0.01502864
Clothing	Vests	35687	0.10503201

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ROWNUMBER

Articolo • 20/10/2023

Restituisce la classificazione per il contesto corrente all'interno della partizione specificata, ordinata in base all'ordine specificato. Se non è possibile trovare una corrispondenza, la classificazione è vuota.

## Sintassi

DAX

```
ROWNUMBER ( [<relation>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>] )
```

## Parametri

Termino	Definizione
relazione	(Facoltativo) Espressione di tabella da cui viene restituita la riga di output. Se specificato, tutte le colonne in <orderBy> e <partitionBy> devono provenire da essa. Se omesso: <ul style="list-style-type: none"><li>- &lt;orderBy&gt; deve essere specificato in modo esplicito.</li><li>- Tutte le colonne &lt;orderBy&gt; e &lt;partitionBy&gt; devono essere complete e provenire da una singola tabella.</li><li>- L'impostazione predefinita è ALLSELECTED() di tutte le colonne in &lt;orderBy&gt; e &lt;partitionBy&gt;.</li></ul>
orderBy	(Facoltativo) Clausola ORDERBY() contenente le colonne che definiscono la modalità di ordinamento di ogni partizione. Se omesso: <ul style="list-style-type: none"><li>- la &lt;relazione&gt; deve essere specificata in modo esplicito.</li><li>- Per impostazione predefinita, l'ordinamento viene eseguito in base a ogni colonna in &lt;relazione&gt; non già specificata in &lt;partitionBy&gt;.</li></ul>
valori blank	(Facoltativo) Enumerazione che definisce come gestire i valori (blank) durante l'ordinamento. I valori supportati sono: <ul style="list-style-type: none"><li>• L'IMPOSTAZIONE PREDEFINITA (valore predefinito), laddove il comportamento per i valori numerici è vuoto, i valori vengono ordinati tra zero e valori negativi. Il comportamento per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote.</li></ul>

Termine	Definizione
	<ul style="list-style-type: none"> <li>• PRIMI, i valori blank vengono sempre ordinati all'inizio, indipendentemente dall'ordinamento crescente o decrescente.</li> <li>• ULTIMI, i valori blank vengono sempre ordinati alla fine, indipendentemente dall'ordinamento crescente o decrescente.</li> </ul> <p>Si noti che quando vengono specificati parametri &lt;blank&gt; e blank nella funzione ORDERBY() in singole espressioni, i &lt;blank&gt; per ogni espressione orderBy hanno la priorità per l'espressione orderBy pertinente e le espressioni orderBy senza &lt;blank&gt; specificate rispettano il parametro &lt;blank&gt; nella funzione Window principale.</p>
partitionBy	(Facoltativo) Clausola PARTITIONBY() contenente le colonne che definiscono la modalità <di partizionamento della relazione>. Se omesso, <la relazione> viene considerata come una singola partizione.
matchBy	(Facoltativo) Clausola MATCHBY() contenente le colonne che definiscono come trovare le corrispondenze con i dati e identificare la riga corrente.

## Valore restituito

Numero di riga per il contesto corrente.

## Osservazioni:

Ogni colonna <orderBy>, <partitionBy> e <matchBy> deve avere un valore esterno corrispondente per definire la riga corrente su cui operare, con il comportamento seguente:

- Se è presente esattamente una colonna esterna corrispondente, viene usato il relativo valore.
- Se non è presente alcuna colonna esterna corrispondente, eseguire le operazioni seguenti:
  - ROWNUMBER determinerà innanzitutto tutte le colonne <orderBy>, <partitionBy> e <matchBy> che non dispongono di colonne esterne corrispondenti.
  - Per ogni combinazione di valori esistenti per queste colonne nel contesto padre ROWNUMBER, ROWNUMBER viene valutata e viene restituita una riga.
  - L'output finale di ROWNUMBER è un'unione di queste righe.
- Se sono presenti più colonne esterne corrispondenti, viene restituito un errore.

Se è presente <matchBy>, ROWNUMBER tenterà di usare le colonne in <matchBy> e <partitionBy> per identificare la riga corrente.

Se le colonne specificate all'interno di <orderBy> e <partitionBy> non possono identificare in modo univoco ogni riga in <relazione>, allora:

- ROWNUMBER tenterà di trovare il minor numero di colonne aggiuntive necessarie per identificare in modo univoco ogni riga.
- Se tali colonne sono disponibili, ROWNUMBER eseguirà le seguenti operazioni
  - Provare a trovare il minor numero di colonne aggiuntive necessarie per identificare in modo univoco ogni riga.
  - Accodare automaticamente queste nuove colonne alla clausola <orderBy>.
  - Ordinare ogni partizione usando questo nuovo set di colonne orderBy.
- Se non è possibile trovare tali colonne e la funzione rileva un legame in fase di esecuzione, viene restituito un errore.

## Esempio

La query DAX seguente:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "UniqueRank",
    ROWNUMBER(
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName]))
    ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]
    asc
```

Restituisce una tabella che classifica ogni area geografica con la stessa proprietà EnglishCountryRegionName, in base a StateProvinceName e City.

## Vedi anche

[INDEX](#)

[ORDERBY](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# SELECTEDVALUE

Articolo • 20/10/2023

Restituisce il valore se il contesto per columnName è stato filtrato in base a un solo valore distinto. In caso contrario, restituisce alternateResult.

## Sintassi

DAX

```
SELECTEDVALUE(<columnName>[ , <alternateResult>])
```

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.
alternateResult	(facoltativo) Valore restituito quando il contesto per columnName è stato filtrato fino a zero o a più di un valore distinto. Se non viene fornito, il valore predefinito è BLANK().

## Valore restituito

Il valore quando il contesto per columnName è stato filtrato in base a un solo valore distinto. In caso contrario, alternateResult.

## Osservazioni:

- Un'espressione equivalente di `SELECTEDVALUE(<columnName>, <alternateResult>)` è `IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>)`.
- Per ulteriori informazioni sulle procedure consigliate quando si usa SELECTEDVALUE, vedere [Usare SELECTEDVALUE invece di VALUES in DAX](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La query DAX seguente:

```
DAX

DEFINE
    MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No
Single Selection")
EVALUATE
    SUMMARIZECOLUMNS
        (ROLLUPADDISSTOTAL(DimProduct[Color], "Is Total"),
        "Selected Color", [Selected Color])ORDER BY [Is Total] ASC,
        [Color] ASC
```

Restituisce quanto segue:

DimProduct[Color]	[Is Total]	[Selected Color]
Nero	FALSE	Nero
Blu	FALSE	Blu
Grigio	FALSE	Grigio
Multi	FALSE	Multi
N/D	FALSE	N/D
Rosso	FALSE	Rosso
Medio	FALSE	Medio
Silver/Black	FALSE	Silver/Black
Bianco	FALSE	Bianco
Giallo	FALSE	Giallo
	TRUE	Nessuna selezione singola

## Commenti e suggerimenti

Questa pagina è stata utile?

👍 Sì👎 No

# WINDOW

Articolo • 20/10/2023

Restituisce più righe posizionate all'interno dell'intervallo specificato.

## Sintassi

DAX

```
WINDOW ( from[, from_type], to[, to_type][, <relation>][, <orderBy>][,
<blanks>][, <partitionBy>][, <matchBy>] )
```

## Parametri

Termino	Definizione
da	Indica dove viene avviata la finestra. Può essere qualsiasi espressione DAX che restituisce un valore scalare. Il comportamento dipende dal parametro <from_type> : - Se <from_type> è REL, il numero di righe da riportare indietro (valore negativo) o in avanti (valore positivo) dalla riga corrente per ottenere la prima riga nella finestra. - Se <from_type> è ABS e <from> è positivo, allora è la posizione dell'inizio della finestra dall'inizio della partizione. L'indicizzazione è basata su 1. Ad esempio, 1 indica che la finestra inizia dall'inizio della partizione. Se <from> è negativo, è la posizione dell'inizio della finestra dalla fine della partizione. -1 indica l'ultima riga nella partizione.
from_type	Modifica il comportamento del parametro <from>. I valori possibili sono ABS (assoluto) e REL (relativo). Il valore predefinito è REL.
a	Uguale a <from>, ma indica la fine della finestra. L'ultima riga è inclusa nella finestra.
to_type	Uguale a <from_type>, ma modifica il comportamento di <to>.
relazione	(Facoltativo) Espressione di tabella da cui viene restituita la riga di output. Se specificato, tutte le colonne in <partitionBy> devono provenire da essa o da una tabella correlata. Se omesso: - <orderBy> deve essere specificato in modo esplicito. - Tutte le colonne <orderBy> e <partitionBy> devono essere complete e provenire da una singola tabella. - L'impostazione predefinita è ALLSELECTED() di tutte le colonne in <orderBy> e <partitionBy>.

Termino	Definizione
orderBy	(Facoltativo) Clausola ORDERBY() contenente le colonne che definiscono la modalità di ordinamento di ogni partizione. Se omesso: <ul style="list-style-type: none"> <li>- &lt;la relazione&gt; deve essere specificata in modo esplicito.</li> <li>- Per impostazione predefinita, l'ordinamento viene eseguito in base a ogni colonna in &lt;relazione&gt; non già specificata in &lt;partitionBy&gt;.</li> </ul>
valori blank	(Facoltativo) Enumerazione che definisce come gestire i valori (blank) durante l'ordinamento. Questo parametro è riservato per usi futuri. Attualmente, l'unico valore supportato è DEFAULT, in cui il comportamento per i valori numerici è vuoto viene ordinato tra zero e valori negativi. Il comportamento per le stringhe è costituito dall'ordinamento dei valori blank prima di tutte le stringhe, incluse le stringhe vuote.
partitionBy	(Facoltativo) Clausola PARTITIONBY() contenente le colonne che definiscono la modalità di partizionamento della <relazione>. Se omesso, la <relazione> viene considerata come una singola partizione.
matchBy	(Facoltativo) Clausola MATCHBY() contenente le colonne che definiscono come trovare le corrispondenze con i dati e identificare la riga corrente.

## Valore restituito

Tutte le righe della finestra.

## Osservazioni:

Ad eccezione delle colonne aggiunte dalle funzioni di tabella DAX, ogni colonna in <relazione>, quando <matchBy> non è presente o ogni colonna in <matchBy> e <partitionBy>, quando <matchBy> è presente, deve avere un valore esterno corrispondente per definire la riga corrente su cui operare. Se <from\_type> e <to\_type> entrambi hanno valore ABS, il codice seguente si applica solo alle colonne <partitionBy>:

- Se è presente esattamente una colonna esterna corrispondente, viene usato il relativo valore.
- Se non è presente alcuna colonna esterna corrispondente:
  - WINDOW determinerà innanzitutto tutte le colonne che non dispongono di una colonna esterna corrispondente.
  - Per ogni combinazione di valori esistenti per queste colonne nel contesto padre di WINDOW, viene valutato WINDOW e vengono restituite le righe

corrispondenti.

- L'output finale WINDOW è un'unione di queste righe.
- Se sono presenti più colonne esterne corrispondenti, viene restituito un errore.

Se tutte le colonne della <relazione> sono state aggiunte dalle funzioni di tabella DAX, viene restituito un errore.

Se <matchBy> è presente, WINDOW tenterà di usare le colonne <matchBy> e <partitionBy> per identificare la riga.

Se <matchBy> non è presente e le colonne specificate all'interno di <orderBy> e <partitionBy> non possono identificare in modo univoco ogni riga in <relazione>, allora:

- WINDOW tenterà di trovare il minor numero di colonne aggiuntive necessarie per identificare in modo univoco ogni riga.
- Se è possibile trovare tali colonne, WINDOW aggiungerà automaticamente queste nuove colonne a <orderBy> e ogni partizione viene ordinata usando questo nuovo set di colonne orderBy.
- Se non è possibile trovare tali colonne, viene restituito un errore.

Viene restituita una tabella vuota se:

- Il valore esterno corrispondente di una colonna <orderBy> o <partitionBy> non esiste all'interno della <relazione>.
- L'intera finestra si trova all'esterno della partizione oppure l'inizio della finestra si trova dopo la fine.

Se WINDOW viene utilizzato all'interno di una colonna calcolata definita nella stessa tabella della <relazione> e <orderBy> viene omesso, viene restituito un errore.

Se l'inizio della finestra viene visualizzato prima della prima riga, viene impostato sulla prima riga. Analogamente, se la fine della finestra si trova dopo l'ultima riga della partizione, viene impostata sull'ultima riga.

## Esempio 1

La misura seguente:

DAX

```
3-day Average Price =  
AVERAGEX(  
    WINDOW(  
        -2,REL,0,REL,  
        SUMMARIZE(ALLSELECTED('Sales'), 'Date'[Date], 'Product'[Product]),
```

```

        ORDERBY( 'Date'[Date]),
        KEEP,
        PARTITIONBY( 'Product'[Product])
    ),
    CALCULATE(AVERAGE(Sales[Unit Price]))
)

```

Restituisce la media di 3 giorni dei prezzi unità per ogni prodotto. Si noti che la finestra di 3 giorni è costituita da tre giorni in cui il prodotto ha vendite, non necessariamente tre giorni di calendario consecutivi.

## Esempio 2

La misura seguente:

DAX

```

RunningSum =
SUMX (
    WINDOW (
        1, ABS, 0, REL,
        ALLSELECTED (
            'Date'[Fiscal Year],
            'Date'[Month Number Of Year]
        ),
        PARTITIONBY ( 'Date'[Fiscal Year] )
    ),
    [Total Sales]
)

```

Restituisce la somma in esecuzione per Total Sales by Month Number Of Year, riavviando per ogni anno fiscale:

<b>Year</b>	<b>Mese numero dell'anno</b>	<b>Sales Amount</b>	<b>RunningSum</b>
FY2018	1	\$ 1.327.675	\$ 1.327.675
FY2018	2	\$ 3.936.463	\$ 5.264.138
FY2018	3	\$ 700.873	\$ 5.965.011
FY2018	4	\$ 1.519,.275	\$ 7.484.286
FY2018	5	\$ 2.960.378	\$ 10.444.664
FY2018	6	\$ 1.487.671	\$ 11.932.336

<b>Year</b>	<b>Mese numero dell'anno</b>	<b>Sales Amount</b>	<b>RunningSum</b>
FY2018	7	\$ 1.423.357	\$ 13.355.693
FY2018	8	\$ 2.057.902	\$ 15.413.595
FY2018	9	\$ 2.523.948	\$ 17.937.543
FY2018	10	\$ 561.681	\$ 18.499.224
FY2018	11	\$ 4.764.920	\$ 23.264.145
FY2018	12	\$ 596.747	\$ 23.860.891
AF2019	1	\$ 1.847.692	\$ 1.847.692
AF2019	2	\$ 2.829.362	\$ 4.677.054
AF2019	3	\$ 2.092.434	\$ 6.769.488
AF2019	4	\$ 2.405.971	\$ 9.175.459
AF2019	5	\$ 3.459.444	\$ 12.634.903
AF2019	6	\$ 2.850.649	\$ 15.485.552
AF2019	7	\$ 2.939.691	\$ 18.425.243
AF2019	8	\$ 3.964.801	\$ 22.390.045
AF2019	9	\$ 3.287.606	\$ 25.677.650
AF2019	10	\$ 2.157.287	\$ 27.834.938
AF2019	11	\$ 3.611.092	\$ 31.446.030
AF2019	12	\$ 2.624.078	\$ 34.070.109
FY2020	1	\$ 3.235.187	\$ 3.235.187
FY2020	2	\$ 4.070.046	\$ 7.305.233
FY2020	3	\$ 4.429.833	\$ 11.735.066
FY2020	4	\$ 4.002.614	\$ 15.737.680
FY2020	5	\$ 5.265.797	\$ 21.003.477
FY2020	6	\$ 3.465.241	\$ 24.468.717
FY2020	7	\$ 3.513.064	\$ 27.981.781
FY2020	8	\$ 5.247.165	\$ 33.228.947

Year	Mese numero dell'anno	Sales Amount	RunningSum
FY2020	9	\$ 5.104.088	\$ 38.333.035
FY2020	10	\$ 3.542.150	\$ 41.875.184
FY2020	11	\$ 5.151.897	\$ 47.027.081
FY2020	12	\$ 4.851.194	\$ 51.878.275

## Vedi anche

[INDEX](#)

[OFFSET](#)

[ORDERBY](#)

[PARTITIONBY](#)

[RANK](#)

[ROWNUMBER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni finanziarie

Articolo • 20/10/2023

Le funzioni finanziarie in DAX vengono usate in formule che eseguono calcoli finanziari, ad esempio del valore attuale netto o del tasso di rendimento. Queste funzioni sono simili alle funzioni finanziarie usate in Microsoft Excel.

## In questa categoria

Funzione	Descrizione
ACCRINT	Restituisce l'interesse maturato per un titolo per cui viene pagato un interesse periodico.
ACCRINTM	Restituisce l'interesse maturato per un titolo i cui interessi vengono pagati alla scadenza.
AMORDEGRC	Restituisce l'ammortamento per ogni periodo contabile. È simile ad AMORLINC, ad eccezione del fatto che viene applicato un coefficiente di ammortamento a seconda della durata dei cespiti.
AMORLINC	Restituisce l'ammortamento per ogni periodo contabile.
COUPDAYBS	Restituisce il numero di giorni dall'inizio del periodo di una cedola fino alla data di regolamento.
COUPDAYS	Restituisce il numero di giorni nel periodo della cedola che contiene la data di regolamento.
COUPDAYSNC	Restituisce il numero di giorni dalla data di regolamento alla data della cedola successiva.
COUPNCD	Restituisce la data della cedola successiva alla data di regolamento.
COUPNUM	Restituisce il numero di cedole pagabili tra la data di regolamento e la data di scadenza, arrotondato alla cedola intera più vicina.
COUPPCD	Restituisce la data della cedola precedente prima della data di regolamento.
CUMIPMT	Restituisce l'interesse cumulativo pagato su un prestito tra start_period ed end_period.
CUMPRINC	Restituisce il capitale cumulativo pagato su un prestito tra start_period ed end_period.
DB	Restituisce l'ammortamento di un cespote per un periodo specificato usando il metodo di ammortamento a quote proporzionali.

<b>Funzione</b>	<b>Descrizione</b>
DDB	Restituisce l'ammortamento di un cespote per un periodo specificato usando il metodo di ammortamento a quote dimezzate o un altro metodo specificato.
DISC	Restituisce il tasso di sconto per un titolo.
DOLLARDE	Converte un prezzo in dollari espresso come parte intera e parte frazionaria, ad esempio 1,02, in un prezzo in dollari espresso come numero decimale.
DOLLARFR	Converte un prezzo in dollari espresso come numero decimale in un prezzo in dollari espresso come parte intera e parte frazionaria, ad esempio 1,02.
DURATION	Restituisce la durata Macauley per un valore nominale presunto di \$ 100.
EFFECT	Restituisce il tasso di interesse annuo effettivo in base al tasso di interesse nominale annuo e al numero di periodi di capitalizzazione per anno.
FV	Calcola il valore futuro di un investimento in base a un tasso di interesse costante.
INTRATE	Restituisce il tasso di interesse per un titolo interamente investito.
IPMT	Restituisce il pagamento degli interessi per un determinato periodo relativi a un investimento sulla base di pagamenti costanti periodici e di un tasso di interesse costante.
ISPMT	Calcola l'interesse pagato (o ricevuto) per il periodo specificato di un prestito o di un investimento con pagamenti di capitale di importo fisso.
MDURATION	Restituisce la durata Macauley modificata per un titolo con un valore nominale presunto di \$ 100.
NOMINAL	Restituisce il tasso di interesse annuo nominale in base al tasso di interesse effettivo e al numero di periodi di capitalizzazione per anno.
NPER	Restituisce il numero di periodi per un investimento sulla base di pagamenti costanti periodici e di un tasso di interesse costante.
ODDFPRICE	Restituisce il prezzo per un valore nominale di \$100 di un titolo con un primo periodo irregolare (breve o lungo).
ODDFYIELD	Restituisce il rendimento di un titolo il cui primo periodo di durata è irregolare (breve o lungo).
ODDLPRICE	Restituisce il prezzo per un valore nominale di \$ 100 di un titolo il cui ultimo periodo di durata è irregolare (lungo o breve).
ODDLYIELD	Restituisce il rendimento di un titolo il cui ultimo periodo di durata è irregolare (breve o lungo).

<b>Funzione</b>	<b>Descrizione</b>
PDURATION	Restituisce il numero di periodi necessari affinché un investimento raggiunga un valore specificato.
PMT	Calcola il pagamento per un prestito sulla base di pagamenti costanti e di un tasso di interesse costante.
PPMT	Restituisce il pagamento sul capitale di un investimento per un determinato periodo sulla base di pagamenti costanti periodici e di un tasso di interesse costante.
PRICE	Restituisce il prezzo per un valore nominale di \$100 di un titolo che paga interessi periodici.
PRICEDISC	Restituisce il prezzo per un valore nominale di \$100 di un titolo scontato.
PRICEMAT	Restituisce il prezzo per un valore nominale di \$ 100 per un titolo i cui interessi vengono pagati alla scadenza.
PV	Calcola il valore attuale di un investimento sulla base di un tasso di interesse costante.
RATE	Restituisce il tasso di interesse per periodo di un'annualità.
RECEIVED	Restituisce l'importo ricevuto alla scadenza di un titolo interamente investito.
RRI	Restituisce un tasso di interesse equivalente per l'incremento di un investimento.
SLN	Restituisce l'ammortamento lineare di un cespote per un periodo.
SYD	Restituisce l'ammortamento anticipato a quote decrescenti di un cespote per un periodo specificato.
TBILLEQ	Restituisce il rendimento equivalente a un'obbligazione per un buono del tesoro.
TBILLPRICE	Restituisce il prezzo per un valore nominale di \$ 100 di un buono del tesoro.
TBILLYIELD	Restituisce il rendimento per un buono del tesoro.
VDB	Restituisce l'ammortamento di un cespote per un periodo specificato, inclusi periodi parziali, usando il metodo di ammortamento a quote dimezzate o un altro metodo specificato dall'utente.
XIRR	Restituisce il tasso interno di rendimento di una pianificazione di flussi di cassa non necessariamente periodica.
XNPV	Restituisce il valore attuale per una pianificazione di flussi di cassa non necessariamente periodica.
YIELD	Restituisce il rendimento per un titolo per cui viene pagato un interesse periodico.

Funzione	Descrizione
<a href="#">YIELDDISC</a>	Restituisce il rendimento annuale per un titolo scontato.
<a href="#">YIELDMAT</a>	Restituisce il rendimento annuo per un titolo i cui interessi vengono pagati alla scadenza.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ACCRINT

Articolo • 20/10/2023

Restituisce l'interesse maturato per un titolo per cui viene pagato un interesse periodico.

## Sintassi

DAX

```
ACCRINT(<issue>, <first_interest>, <settlement>, <rate>, <par>, <frequency>
[, <basis>[, <calc_method>]]])
```

## Parametri

Termine	Definizione
problema	Data di emissione del titolo.
first_interest	Data del primo interesse del titolo.
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
rate	Tasso di interesse annuo della cedola del titolo.
par	Valore nominale del titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.
calc_method	(Facoltativo) Valore logico che specifica il modo in cui viene calcolato l'interesse accumulato totale quando la data di settlement è successiva alla data di first_interest. Se calc_method viene omesso, si presuppone il valore TRUE. - Se calc_method restituisce TRUE o viene omesso, ACCRINT restituisce l'interesse totale accumulato da issue a settlement. - Se calc_method restituisce FALSE, ACCRINT restituisce l'interesse totale accumulato da first_interest a settlement.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Interesse accumulato.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La funzione ACCRINT viene calcolata in questo modo:

$$\text{ACCRINT} = \text{par} \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{\text{NC}} \frac{A_i}{N\text{L}_i}$$

dove:

- $A_i$  = numero di giorni maturati per il periodo di quasi cedola  $i^{th}$  all'interno del periodo di durata irregolare.
- $\text{NC}$  = numero di periodi di quasi cedola che rientrano nel periodo di durata irregolare. Se questo numero contiene una frazione, aumentarla al numero intero successivo.
- $N\text{L}_i$  = durata normale in giorni del periodo di quasi cedola all'interno del periodo di durata irregolare.
- issue, first\_interest e settlement vengono troncati ai numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il valore di issue, first\_interest o settlement non è una data valida.
  - issue  $\geq$  settlement.

- rate  $\leq 0$ .
- par  $\leq 0$ .
- Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
- base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
1-marzo-2007	Data di emissione
31-agosto-2008	Data del primo interesse
1-maggio-2008	Data di regolamento
10%	Tasso della cedola
1000	Valore nominale
2	La frequenza è semestrale (vedere sopra)
0	Base 30/360 (vedere sopra)

## Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0)
}
```

Restituisce l'interesse accumulato da issue a settlement per un titolo con i termini specificati sopra.

[Valore]
116,944444444444

## Esempio 2

La query DAX seguente:

DAX

```
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0,
    FALSE)
}
```

Restituisce l'interesse accumulato da first\_interest a settlement per un titolo con i termini specificati sopra.

[Valore]

66,9444444444445

## Commenti e suggerimenti

Questa pagina è stata utile?



# ACCRINTM

Articolo • 20/10/2023

Restituisce l'interesse maturato per un titolo i cui interessi vengono pagati alla scadenza.

## Sintassi

DAX

```
ACCRINTM(<issue>, <maturity>, <rate>, <par>[, <basis>])
```

## Parametri

Termine	Definizione
issue	Data di emissione del titolo.
maturity	Data di scadenza del titolo.
rate	Tasso di interesse annuo della cedola del titolo.
par	Valore nominale del titolo.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Interesse accumulato.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La funzione ACCRINTM viene calcolata come segue:

$$\text{ACCRINTM} = \text{par} \times \text{rate} \times \frac{\text{A}}{\text{D}}$$

dove:

- A = numero di giorni maturati conteggiati su base mensile. Per il calcolo degli interessi degli elementi in scadenza viene usato il numero di giorni dalla data di emissione alla data di scadenza.
- D = base annuale.
- I parametri issue e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro issue o maturity non rappresenta una data valida.
  - issue  $\geq$  maturity.
  - rate  $\leq 0$ .
  - par  $\leq 0$ .
  - base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
1-aprile-2008	Data di emissione
15-giugno-2008	Data di scadenza
10%	Percentuale cedola
1000	Valore nominale
3	Base giorni effettivi/365 (vedere sopra)

La query DAX seguente:

```
DAX
```

```
EVALUATE
{
    ACCRINTM(DATE(2008,4,1), DATE(2008,6,15), 0.1, 1000, 3)
```

Restituisce l'interesse accumulato per un titolo con i termini specificati in precedenza.

[Valore]
20,5479452054795

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# AMORDEGRC

Articolo • 20/10/2023

Restituisce l'ammortamento per ogni periodo contabile. Questa funzione viene fornita per il sistema contabile francese. Se un cespote viene acquistato a metà del periodo contabile, viene tenuto conto dell'ammortamento ripartito proporzionalmente. Questa funzione è simile ad AMORLINC, ad eccezione del fatto che nel calcolo viene applicato un coefficiente di ammortamento a seconda della durata dei cespiti.

## Sintassi

DAX

```
AMORDEGRC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

## Parametri

Termine	Definizione
cost	Costo del cespote.
date_purchased	Data di acquisto del cespote.
first_period	Data della fine del primo periodo.
salvage	Valore di realizzo al termine della durata del cespote.
period	Periodo.
rate	Tasso di ammortamento.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Sistema di date
0 o omesso	360 giorni (metodo NASD)
1	Valore effettivo

Basis	Sistema di date
3	365 giorni in un anno
4	360 giorni in un anno (metodo europeo)

## Valore restituito

Ammortamento per ogni periodo contabile.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- Questa funzione restituirà l'ammortamento fino all'ultimo periodo della durata dei cespiti o fino a quando il valore di ammortamento accumulato non è maggiore del costo dei cespiti meno il valore di realizzo.
- I coefficienti di ammortamento sono i seguenti:

Durata dei cespiti (1/tasso)	Coefficiente di ammortamento
Tra 3 e 4 anni	1.5
Tra 5 e 6 anni	2
Più di 6 anni	2.5

- Il tasso di ammortamento aumenterà fino al 50% per il periodo precedente l'ultimo periodo e fino al 100% per l'ultimo periodo.
- I parametri period e basis verranno arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - costo < 0.
  - Il valore di first\_period o date\_purchased non è una data valida.
  - date\_purchased > first\_period.
  - valore residuo < 0 o valore residuo > cost.
  - periodo < 0.
  - rate ≤ 0.
  - La durata dei cespiti è compresa tra 0 (zero) e 1, 1 e 2, 2 e 3 o 4 e 5.

- Il valore di basis è qualsiasi numero diverso da 0, 1, 3 o 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
2400	Costi
19-agosto-2008	Data di acquisto
31-dicembre-2008	Fine del primo periodo
300	Valore di realizzo
1	Period
15%	Tasso di ammortamento
1	Base giorni effettivi (vedere sopra)

La query DAX seguente:

```
DAX
EVALUATE
{
    AMORDEGRC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Restituisce l'ammortamento del primo periodo, in base ai termini specificati sopra.

[Valore]
776

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# AMORLINC

Articolo • 20/10/2023

Restituisce l'ammortamento per ogni periodo contabile. Questa funzione viene fornita per il sistema contabile francese. Se un cespote viene acquistato a metà del periodo contabile, viene tenuto conto dell'ammortamento ripartito proporzionalmente.

## Sintassi

DAX

```
AMORLINC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

## Parametri

Termino	Definizione
cost	Costo del cespote.
date_purchased	Data di acquisto del cespote.
first_period	Data della fine del primo periodo.
salvage	Valore di realizzo al termine della durata del cespote.
period	Periodo.
rate	Tasso di ammortamento.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Sistema di date
0 o omesso	360 giorni (metodo NASD)
1	Valore effettivo
3	365 giorni in un anno
4	360 giorni in un anno (metodo europeo)

# Valore restituito

Ammortamento per ogni periodo contabile.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- I parametri period e basis verranno arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - costo < 0.
  - Il valore di first\_period o date\_purchased non è una data valida.
  - date\_purchased > first\_period.
  - valore residuo < 0 o valore residuo > cost.
  - periodo < 0.
  - rate ≤ 0.
  - Il valore di basis è qualsiasi numero diverso da 0, 1, 3 o 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
2400	Costi
19-agosto-2008	Data di acquisto
31-dicembre-2008	Fine del primo periodo
300	Valore di realizzo
1	Period
15%	Tasso di ammortamento
1	Base giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Restituisce l'ammortamento del primo periodo, in base ai termini specificati sopra.

[Valore]

360

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# COUPDAYBS

Articolo • 20/10/2023

Restituisce il numero di giorni dall'inizio del periodo di una cedola fino alla data di regolamento.

## Sintassi

DAX

```
COUPDAYBS(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Numero di giorni dall'inizio di un periodo di cedola fino alla data di regolamento.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-11	Data di regolamento
15-nov-11	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPDAYBS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Restituisce il numero di giorni dall'inizio del periodo di cedola fino alla data di regolamento, per un'obbligazione con i termini specificati in precedenza.

[Valore]

71

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUPDAYS

Articolo • 20/10/2023

Restituisce il numero di giorni nel periodo della cedola che contiene la data di regolamento.

## Sintassi

DAX

```
COUPDAYS(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Numero di giorni nel periodo della cedola che contiene la data di regolamento.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1 gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza è il 1° gennaio 2038, 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-11	Data di regolamento
15-nov-11	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPDAYS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Restituisce il numero di giorni nel periodo della cedola che contiene la data di regolamento, per un'obbligazione con i termini specificati in precedenza.

[Valore]

181

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUPDAYSNC

Articolo • 20/10/2023

Restituisce il numero di giorni dalla data di regolamento alla data della cedola successiva.

## Sintassi

DAX

```
COUPDAYSNC(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Numero di giorni dalla data di regolamento alla data della cedola successiva.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-11	Data di regolamento
15-nov-11	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPDAYSNC(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Restituisce il numero di giorni dalla data di regolamento alla data della cedola successiva, per un'obbligazione con i termini specificati in precedenza.

[Valore]

110

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUPNCD

Articolo • 20/10/2023

Restituisce la data della cedola successiva alla data di regolamento.

## Sintassi

DAX

```
COUPNCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Data della cedola successiva alla data di regolamento.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1 gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza è il 1° gennaio 2038, 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-11	Data di regolamento
15-nov-11	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPNCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Restituisce la data della cella successiva dopo la data di regolamento, per un'obbligazione con i termini specificati in precedenza.

[Valore]
15/5/2011 12:00:00

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUPNUM

Articolo • 20/10/2023

Restituisce il numero di cedole pagabili tra la data di regolamento e la data di scadenza, arrotondato alla cedola intera più vicina.

## Sintassi

DAX

```
COUPNUM(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Numero di cedole pagabili tra la data di regolamento e la data di scadenza.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-07	Data di regolamento
15-nov-08	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPNUM(DATE(2007,1,25), DATE(2008,11,15), 2, 1)
}
```

Restituisce il numero di pagamenti di cedole per un'obbligazione con i termini specificati in precedenza.

[Valore]

4

## Commenti e suggerimenti

Questa pagina è stata utile?



# COUPPCD

Articolo • 20/10/2023

Restituisce la data della cedola precedente prima della data di regolamento.

## Sintassi

DAX

```
COUPPCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Data della cedola precedente prima della data di regolamento.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
25-gen-11	Data di regolamento
15-nov-11	Data di scadenza
2	Cedola semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    COUPPCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Restituisce la data della cedola precedente prima della data di regolamento per un'obbligazione usando le condizioni sopra specificate.

[Valore]
15/11/2010 00:00:00

## Commenti e suggerimenti

Questa pagina è stata utile?



# CUMIPMT

Articolo • 20/10/2023

Restituisce l'interesse cumulativo pagato su un prestito tra start\_period ed end\_period.

## Sintassi

DAX

```
CUMIPMT(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

## Parametri

Termine	Definizione
rate	Tasso di interesse.
nper	Numero totale dei periodi di pagamento.
pv	Valore attuale.
start_period	Primo periodo nel calcolo. Deve essere compreso tra 1 ed end_period (inclusi).
end_period	Ultimo periodo nel calcolo. Deve essere compreso tra start_period e nper (inclusi).
Tipo	Tempistica del pagamento. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Tipo	Temporizzazione
0 (zero)	Pagamento alla fine del periodo
1	Pagamento all'inizio del periodo

## Valore restituito

Interesse cumulativo pagato nel periodo specificato.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 10%, usare 0,1/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,1 per rate e 4 per nper.
- I parametri start\_period, end\_period e type vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - rate  $\leq$  0.
  - nper  $<$  1.
  - pv  $\leq$  0.
  - start\_period  $<$  1 o start\_period  $>$  end\_period.
  - end\_period  $<$  start\_period or end\_period  $>$  nper.
  - Il valore di type è qualsiasi numero diverso da 0 o 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
%9	Tasso di interesse annuale
30	Anni del prestito
125000	Valore corrente

## Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Restituisce l'interesse totale pagato nel secondo anno dei pagamenti, ovvero i periodi da 13 a 24, presupponendo che i pagamenti vengano effettuati all'inizio di ogni mese.

[Valore]

-11052,3395838718

## Esempio 2

La query DAX seguente:

DAX

```
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Restituisce l'interesse pagato in un singolo pagamento nel primo mese, supponendo che il pagamento venga effettuato alla fine del mese.

[Valore]

-937,5

## Commenti e suggerimenti

Questa pagina è stata utile?



# CUMPRINC

Articolo • 20/10/2023

Restituisce il capitale cumulativo pagato su un prestito tra start\_period ed end\_period.

## Sintassi

DAX

```
CUMPRINC(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

## Parametri

Termine	Definizione
rate	Tasso di interesse.
nper	Numero totale dei periodi di pagamento.
pv	Valore attuale.
start_period	Primo periodo nel calcolo. Deve essere compreso tra 1 ed end_period (inclusi).
end_period	Ultimo periodo nel calcolo. Deve essere compreso tra start_period e nper (inclusi).
Tipo	Tempistica del pagamento. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Tipo	Temporizzazione
0 (zero)	Pagamento alla fine del periodo
1	Pagamento all'inizio del periodo

## Valore restituito

Capitale cumulativo pagato nel periodo specificato.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 10%, usare 0,1/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,1 per rate e 4 per nper.
- I parametri start\_period, end\_period e type vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - rate  $\leq$  0.
  - nper  $<$  1.
  - pv  $\leq$  0.
  - start\_period  $<$  1 o start\_period  $>$  end\_period.
  - end\_period  $<$  start\_period or end\_period  $>$  nper.
  - Il valore di type è qualsiasi numero diverso da 0 o 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
%9	Tasso di interesse annuale
30	Termine in anni
125000	Valore corrente

## Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Restituisce il capitale totale pagato nel secondo anno dei pagamenti, ovvero i periodi da 13 a 24, presupponendo che i pagamenti vengano effettuati all'inizio di ogni mese.

[Valore]

-927,153472378062

## Esempio 2

La query DAX seguente:

DAX

```
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Restituisce il capitale pagato in un singolo pagamento nel primo mese, supponendo che il pagamento venga effettuato alla fine del mese.

[Valore]

-68,2782711809784

## Commenti e suggerimenti

Questa pagina è stata utile?



# Database del sito (DB)

Articolo • 20/10/2023

Restituisce l'ammortamento di un cespote per un periodo specificato usando il metodo di ammortamento a quote proporzionali.

## Sintassi

DAX

```
DB(<cost>, <salvage>, <life>, <period>[, <month>])
```

## Parametri

Termine	Definizione
cost	Costo iniziale del bene.
salvage	Valore al termine dell'ammortamento (talvolta chiamato valore residuo del cespote). Questo valore può essere 0.
dei dati	Numero di periodi di ammortamento del cespote (talvolta chiamato vita utile del cespote).
period	Periodo per il quale si vuole calcolare l'ammortamento. Period deve usare le stesse unità di life. Deve essere compreso tra 1 e life (inclusi).
mese	(Facoltativo) Numero di mesi nel primo anno. Se month viene omesso, si presuppone il valore 12.

## Valore restituito

Ammortamento per il periodo specificato.

## Osservazioni:

- Il metodo di ammortamento a quote proporzionali calcola l'ammortamento con quote proporzionali. DB usa la formula seguente per calcolare l'ammortamento per un periodo:

$$(\text{costo} - \text{ammortamento totale per i periodi precedenti}) \times \text{tasso}$$

dove:

$$\text{tasso} = 1 - \left( \left( \frac{\text{valore residuo}}{\text{costo}} \right)^{\left( \frac{1}{\text{life}} \right)} \right), \text{arrotondato a tre posizioni decimali}$$

- L'ammortamento per il primo e l'ultimo periodo è un caso speciale.

- Per il primo periodo, DB usa questa formula:

$$\frac{\text{costo} \times \text{tasso} \times \text{mese}}{12}$$

- Per l'ultimo periodo, DB usa questa formula:

$$\frac{(\text{costo} - \text{ammortamento totale per i periodi precedenti}) \times \text{tasso} \times (12 - \text{mese})}{12}$$

- I parametri period e month vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - cost < 0.
  - salvage < 0.
  - vita < 1.
  - periodo < 1 o periodo > vita.
  - mese < 1 o mese > 12.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

### Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    DB(1000000, 0, 6, 1, 2)
}
```

Restituisce l'ammortamento di un cespote negli ultimi due mesi del primo anno, presupponendo che valga \$0 dopo 6 anni.

[Valore]
166666.666666667

### Esempio 2

Di seguito viene calcolato l'ammortamento totale di tutti i cespiti in anni diversi della loro durata. In questo caso il primo anno include solo 7 mesi di ammortamento e l'ultimo anno include solo 5 mesi.

## DAX

```
DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])+1
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
ADDCOLUMNS (
    DepreciationPeriods,
    "Current Period Total Depreciation",
    SUMX (
        FILTER (
            Asset,
            [Value] <= [LifetimeYears]+1
        ),
        DB([InitialCost], [SalvageValue], [LifetimeYears], [Value], 7)
    )
)
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# DDB

Articolo • 20/10/2023

Restituisce l'ammortamento di un cespote per un periodo specificato usando il metodo di ammortamento a quote dimezzate o un altro metodo specificato.

## Sintassi

DAX

```
DDB(<cost>, <salvage>, <life>, <period>[, <factor>])
```

## Parametri

Termine	Definizione
cost	Costo iniziale del bene.
salvage	Valore al termine dell'ammortamento (talvolta chiamato valore residuo del cespote). Questo valore può essere 0.
dei dati	Numero di periodi di ammortamento del cespote (talvolta chiamato vita utile del cespote).
period	Periodo per il quale si vuole calcolare l'ammortamento. Period deve usare le stesse unità di life. Deve essere compreso tra 1 e life (inclusi).
fattore	(Facoltativo) Quota di ammortamento. Se factor viene omesso, si presuppone il valore 2 (metodo di ammortamento a quote dimezzate).

## Valore restituito

Ammortamento per il periodo specificato.

## Osservazioni:

- Il metodo di ammortamento a quote dimezzate calcola l'ammortamento con quote accelerate. L'ammortamento è più alto nel primo periodo e diminuisce nei periodi successivi. DDB usa la formula seguente per calcolare l'ammortamento per un periodo:

$$\text{Min}((\text{cost} - \text{ammortamento totale dai periodi precedenti}) \times (\frac{\text{factor}}{\text{life}}), (\text{cost} - \text{salvage} - \text{ammortamento totale dai periodi precedenti}))$$

- Modificare factor se non si vuole usare il metodo di ammortamento a quote dimezzate.
- Usare la funzione VDB se si vuole passare al metodo di ammortamento lineare quando l'ammortamento è maggiore del calcolo dell'ammortamento decrescente.
- Il parametro period viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - cost < 0.
  - salvage < 0.
  - vita < 1.
  - periodo < 1 o periodo > vita.
  - factor ≤ 0.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

### Esempio 1

La query DAX seguente:

DAX

```
EVALUATE
{
    DDB(1000000, 0, 10, 5, 1.5)
}
```

Restituisce l'ammortamento di un cespito nell'anno 5<sup>o</sup>, presumendo che il valore sia \$0 dopo 10 anni. Questo calcolo usa un fattore di 1,5.

[Valore]
78300,9375

## Esempio 2

Di seguito viene calcolato l'ammortamento totale di tutti i cespi in anni diversi della loro durata. Questo calcolo usa il fattore predefinito 2 (metodo di ammortamento a quote dimezzate).

```
DAX

DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
    ADDCOLUMNS (
        DepreciationPeriods,
        "Current Period Total Depreciation",
        SUMX (
            FILTER (
                Asset,
                [Value] <= [LifetimeYears]
            ),
            DDB([InitialCost], [SalvageValue], [LifetimeYears], [Value])
        )
    )
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Si  No



# DISC

Articolo • 20/10/2023

Restituisce il tasso di sconto per un titolo.

## Sintassi

DAX

```
DISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
pr	Prezzo del titolo per \$100 di valore nominale.
redemption	Valore di riscatto del titolo per il valore nominale \$100.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Tasso di sconto.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2018 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2018, la data di regolamento sarà il 1° luglio 2018 e la data di scadenza sarà il 1° gennaio 2048, 30 anni dopo il 1° gennaio 2018, la data di emissione.
- La funzione DISC viene calcolata come segue:

$$\text{DISC} = \frac{\text{redemption} - \text{par}}{\text{redemption}} \times \frac{B}{DSM}$$

dove:

- B = numero di giorni in un anno, a seconda della base per l'anno.
- DSM = numero di giorni tra regolamento e scadenza.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - settlement  $\geq$  maturity.
  - pr  $\leq$  0.
  - redemption  $\leq$  0.
  - base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
01/07/2018	Data di regolamento
01/01/2048	Data di scadenza
97,975	Prezzo
100	Valore di riscatto
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

```
DAX
EVALUATE
{
    DISC(DATE(2018,7,1), DATE(2048,1,1), 97.975, 100, 1)
}
```

Restituisce il tasso di sconto per un'obbligazione con i termini specificati in precedenza.

[Valore]
0,000686384169121348

## Commenti e suggerimenti

Questa pagina è stata utile?

**Sì**    **No**

# DOLLARDE

Articolo • 20/10/2023

Converte un prezzo in dollari espresso come numero decimale in un prezzo in dollari espresso come parte intera e parte frazionaria, ad esempio 1,02. I numeri in dollari frazionari vengono talvolta usati per i prezzi dei titoli.

## Sintassi

DAX
<code>DOLLARDE(&lt;fractional_dollar&gt;, &lt;fraction&gt;)</code>

## Parametri

Termine	Definizione
fractional_dollar	Numero espresso come parte intera e parte frazionaria, separate da un simbolo decimale.
frazione	Intero che costituisce il denominatore della frazione.

## Valore restituito

Valore decimale di *fractional\_dollar*.

## Osservazioni:

- La parte frazionaria del valore è divisa per un valore integer specificato. Ad esempio, se si vuole esprimere il prezzo con una precisione di 1/16 di dollaro, dividere la parte frazionaria per 16. In questo caso, 1,02 rappresenta \$1,125 (\$1 + 2/16 = \$1,125).
- Il parametro fraction viene arrotondato al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - fraction < 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La query DAX seguente:

```
DAX  
EVALUATE  
{  
    DOLLARDE(1.02, 16)  
}
```

Restituisce 1,125, il prezzo decimale del prezzo frazionario originale (1,02) letto come 1 e 2/16. Poiché il valore frazionario è 16, il prezzo ha una precisione di 1/16 di dollaro.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DOLLARFR

Articolo • 20/10/2023

Converte un prezzo in dollari espresso come numero decimale in un prezzo in dollari espresso come parte intera e parte frazionaria, ad esempio 1,02. I numeri in dollari frazionari vengono talvolta usati per i prezzi dei titoli.

## Sintassi

DAX

```
DOLLARFR(<decimal_dollar>, <fraction>)
```

## Parametri

Termine	Definizione
decimal_dollar	Numero decimale.
fraction	Intero che costituisce il denominatore della frazione.

## Valore restituito

Valore frazionario di *decimal\_dollar*, espresso come parte integer e parte frazionaria.

## Osservazioni:

- Il parametro *fraction* viene arrotondato al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - *fraction* < 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE
{
    DOLLARFR(1.125, 16)
}
```

Restituisce 1,02, letto come 1 e 2/16, ovvero il prezzo frazionario corrispondente del prezzo decimale originale 1,125. Poiché il valore frazionario è 16, il prezzo ha una precisione di 1/16 di dollaro.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DURATION

Articolo • 20/10/2023

Restituisce la durata Macauley per un valore nominale presunto di \$100. La durata viene definita come la media ponderata del valore attuale dei flussi di cassa e viene usata come misura della risposta del prezzo di un'obbligazione alle variazioni del rendimento.

## Sintassi

DAX

```
DURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
coupon	Tasso di interesse annuo della cedola del titolo.
yld	Rendimento annuale del titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

# Valore restituito

Durata Macauley.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza è il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - $\text{coupon} < 0$ .
  - $\text{yld} < 0$
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
01/07/2018	Data di regolamento
01/01/2048	Data di scadenza
8,0%	Percentuale cedola

Dati	Descrizione
9,0%	Percentuale rendimento
2	La frequenza è semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

```
DAX
EVALUATE
{
    DURATION(DATE(2018,7,1), DATE(2048,1,1), 0.08, 0.09, 2, 1)
}
```

Restituisce la durata Macauley per un'obbligazione con i termini specificati in precedenza.

[Valore]
10,9191452815919

## Commenti e suggerimenti

Questa pagina è stata utile?



# EFFECT

Articolo • 20/10/2023

Restituisce il tasso di interesse annuo effettivo in base al tasso di interesse nominale annuo e al numero di periodi di capitalizzazione per anno.

## Sintassi

DAX

```
EFFECT(<nominal_rate>, <nper>)
```

## Parametri

Termine	Definizione
nominal_rate	Tasso di interesse nominale.
nper	Numero dei periodi di capitalizzazione per anno.

## Valore restituito

Tasso di interesse annuale effettivo.

## Osservazioni:

- La funzione EFFECT viene calcolata come segue:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal\_rate}}{\text{nper}}\right)^{\text{nper}} - 1$$

- Il parametro nper viene arrotondato al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - nominal\_rate ≤ 0.
  - nper < 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Dati	Descrizione
5,25%	Tasso di interesse nominale
4	Numero dei periodi di capitalizzazione per anno

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    EFFECT(0.0525, 4)  
}
```

Restituisce il tasso di interesse effettivo in base ai termini specificati in precedenza.

[Valore]
0,0535426673707584

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FV

Articolo • 20/10/2023

Calcola il valore futuro di un investimento in base a un tasso di interesse costante. È possibile usare FV con pagamenti periodici, costanti e/o un singolo pagamento forfettario.

## Sintassi

DAX
<code>FV(&lt;rate&gt;, &lt;nper&gt;, &lt;pmt&gt;[, &lt;pv&gt;[, &lt;type&gt;]])</code>

## Parametri

Termine	Definizione
rate	Tasso di interesse per periodo.
nper	Numero totale dei periodi di pagamento in un'annualità.
pmt	Pagamento effettuato in ogni periodo. Non può cambiare per la durata dell'annualità. In genere, pmt contiene il capitale e gli interessi, ma non altre tariffe o imposte.
pv	(Facoltativo) Valore attuale o importo forfettario pari al valore attuale di una serie di pagamenti futuri. Se pv viene omesso, si presuppone il valore BLANK.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

**Nota:** Per una descrizione più completa degli argomenti della funzione FV e per altre informazioni sulle funzioni di annualità, vedere la funzione PV.

## Valore restituito

Il valore futuro di un investimento.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Per tutti gli argomenti, i contanti pagati, ad esempio i depositi sui conti di risparmio, sono rappresentati da numeri negativi, mentre i contanti ricevuti, ad esempio gli assegni dei dividendi, sono rappresentati da numeri positivi.
- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - nper < 1
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
%6	Tasso di interesse annuale
10	Numero di pagamenti
200-	Importo del pagamento
-500	Valore corrente
1	Il pagamento è dovuto all'inizio del periodo (0 indica che i pagamenti sono dovuti alla fine del periodo)

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    FV(0.06/12, 10, -200, -500, 1)  
}
```

Restituisce il valore futuro di un investimento in base ai termini specificati in precedenza.

[Valore]

2581,40337406012

## Commenti e suggerimenti

Questa pagina è stata utile?



# INTRATE

Articolo • 20/10/2023

Restituisce il tasso di interesse per un titolo interamente investito.

## Sintassi

DAX

```
INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
investment	Importo investito nel titolo.
redemption	Importo da ricevere alla scadenza.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Tasso di interesse.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza è il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La funzione INTRATE viene calcolata come segue:

$$\text{INTRATE} = \frac{\text{redemption} - \text{investment}}{\text{investment}} \times \frac{B}{\text{DIM}}$$

dove:

- B = numero di giorni in un anno, a seconda della base per l'anno.
- DIM = numero di giorni dal regolamento alla scadenza.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - settlement  $\geq$  maturity.
  - investment  $\leq 0$ .
  - redemption  $\leq 0$ .
  - base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
15/2/2008	Data di regolamento
15/5/2008	Data di scadenza
\$1.000.000	Investimento
\$1.014.420	Valore di riscatto
2	Base giorni effettivi/360

La query DAX seguente:

```
DAX
EVALUATE
{
    INTRATE(DATE(2008,2,15), DATE(2008,5,15), 1000000, 1014420, 2)
}
```

Restituisce il tasso di sconto per un'obbligazione in base ai termini specificati in precedenza.

[Valore]
0,05768

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# IPMT

Articolo • 20/10/2023

Restituisce il pagamento degli interessi per un determinato periodo relativi a un investimento sulla base di pagamenti costanti periodici e di un tasso di interesse costante.

## Sintassi

DAX

```
IPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per periodo.
per	Periodo per cui si vuole calcolare l'interesse. Deve essere compreso tra 1 e nper (inclusi).
nper	Numero totale dei periodi di pagamento in un'annualità.
pv	Valore attuale o somma forfettaria pari al valore attuale di una serie di pagamenti futuri.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone il valore BLANK.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

## Valore restituito

Pagamento degli interessi per il periodo specificato.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Per tutti gli argomenti, i contanti pagati, ad esempio i depositi sui conti di risparmio, sono rappresentati da numeri negativi, mentre i contanti ricevuti, ad esempio gli assegni dei dividendi, sono rappresentati da numeri positivi.
- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - per < 1 o per > nper
  - nper < 1
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
10,00%	Interesse annuale
3	Anni di prestito
\$8.000	Valore attuale del prestito

## Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    IPMT(0.1/12, 1, 3*12, 8000)
}
```

Restituisce l'interesse mensile dovuto il primo mese per un prestito con i termini specificati in precedenza.

[Valore]
-66,6666666666667

## Esempio 2

La query DAX seguente:

DAX
EVALUATE
{
IPMT(0.1, 3, 3, 8000)
}

Restituisce l'interesse annuale dovuto nell'ultimo anno per un prestito con i termini specificati in precedenza, nel caso di pagamenti effettuati annualmente.

[Valore]
-292,447129909366

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No



# ISPMT

Articolo • 20/10/2023

Calcola l'interesse pagato (o ricevuto) per il periodo specificato di un prestito o di un investimento con pagamenti di capitale di importo fisso.

## Sintassi

DAX

```
ISPMT(<rate>, <per>, <nper>, <pv>)
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per l'investimento.
per	Periodo per cui si vuole calcolare l'interesse. Deve essere compreso tra 0 e nper-1, inclusi.
nper	Numero totale dei periodi di pagamento per l'investimento.
pv	Valore attuale dell'investimento. Per un prestito, pv è l'importo del prestito.

## Valore restituito

Interesse pagato (o ricevuto) per il periodo specificato.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuo pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Per tutti gli argomenti, i contanti pagati, ad esempio i depositi sui conti di risparmio o altri prelievi, sono rappresentati da numeri negativi, mentre i contanti

ricevuti, ad esempio gli assegni dei dividendi e altri depositi, sono rappresentati da numeri positivi.

- ISPMT conta ogni periodo iniziando da zero e non da uno.
- La maggior parte dei prestiti usa un piano di rimborso con pagamenti periodici di importo fisso. La funzione IPMT restituisce il pagamento degli interessi per un determinato periodo per questo tipo di prestito.
- Alcuni prestiti usano un piano di rimborso con pagamenti del capitale di importo fisso. La funzione ISPMT restituisce il pagamento degli interessi per un determinato periodo per questo tipo di prestito.
- Viene restituito un errore nei casi seguenti:
  - nper = 0.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
\$4.000	Valore corrente
4	Numero di periodi
10%	Tariffa

Per indicare quando usare ISPMT, la tabella di ammortamento seguente usa un piano di rimborso del capitale di importo fisso con i termini specificati sopra. L'addebito degli interessi per ogni periodo è uguale al tasso per il saldo non pagato per il periodo precedente. Il pagamento di ogni periodo è uguale al capitale di importo fisso più l'interesse per il periodo.

Period	Pagamento del capitale	Pagamento degli interessi	Pagamento totale	Stato patrimoniale
				4,000,00
1	1,000,00	400,00	1.400,00	3,000,00
2	1,000,00	300,00	1.300,00	2,000,00
3	1,000,00	200,00	1.200,00	1,000,00

<b>Period</b>	<b>Pagamento del capitale</b>	<b>Pagamento degli interessi</b>	<b>Pagamento totale</b>	<b>Stato patrimoniale</b>
4	1.000,00	100,00	1.100,00	0,00

La query DAX seguente:

DAX
<pre>DEFINE     VAR NumPaymentPeriods = 4     VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1) EVALUATE     ADDCOLUMNS (         PaymentPeriods,         "Interest Payment",         ISPMT(0.1, [Value], NumPaymentPeriods, 4000)     )</pre>

Restituisce l'interesse pagato durante ogni periodo, usando il piano di rimborso del capitale di importo fisso e i termini specificati sopra. I valori sono negativi per indicare che si tratta di un interesse pagato, non ricevuto.

[Valore]	[Interest Payment]
0	-400
1	-300
2	200-
3	-100

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# MDURATION

Articolo • 20/10/2023

Restituisce la durata Macauley modificata per un titolo con un valore nominale presunto di \$100.

## Sintassi

DAX

```
MDURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
coupon	Tasso di interesse annuo della cedola del titolo.
yld	Rendimento annuale del titolo.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

# Valore restituito

Durata Macauley modificata.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza è il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La durata modificata viene definita nel modo seguente:

$$\text{MDURATION} = \frac{\text{DURATION}}{1 + \left( \frac{\text{Rendimento mercato}}{\text{Pagamenti cedole all'anno}} \right)}$$

- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - $\text{coupon} < 0$ .
  - $\text{yld} < 0$
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
1/1/2008	Data di regolamento
1/1/2016	Data di scadenza
%8	Percentuale cedola
%9	Percentuale rendimento
2	La frequenza è semestrale (vedere sopra)
1	Base giorni effettivi/giorni effettivi (vedere sopra)

La query DAX seguente:

```
DAX
EVALUATE
{
    MDURATION(DATE(2008,1,1), DATE(2016,1,1), 0.08, 0.09, 2, 1)
}
```

Restituisce la durata Macauley modificata di un'obbligazione in base ai termini specificati in precedenza.

[Valore]

5,73566981391884

## Commenti e suggerimenti

Questa pagina è stata utile?



# NOMINAL

Articolo • 20/10/2023

Restituisce il tasso di interesse annuo nominale in base al tasso di interesse effettivo e al numero di periodi di capitalizzazione per anno.

## Sintassi

DAX

```
NOMINAL(<effect_rate>, <nper>)
```

## Parametri

Termine	Definizione
effect_rate	Tasso di interesse effettivo.
nper	Numero dei periodi di capitalizzazione per anno.

## Valore restituito

Tasso di interesse annuale nominale.

## Osservazioni:

- La relazione tra NOMINAL ed EFFECT è illustrata nell'equazione seguente:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal\_rate}}{\text{nper}}\right)^{\text{nper}} - 1$$

- Il parametro nper viene arrotondato al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - effect\_rate ≤ 0.
  - nper < 1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Dati	Descrizione
5,3543%	Tasso di interesse effettivo
4	Numero dei periodi di capitalizzazione per anno

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    NOMINAL(0.053543, 4)  
}
```

Restituisce il tasso di interesse nominale in base ai termini specificati in precedenza.

[Valore]
0,052500319868356

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NPER

Articolo • 20/10/2023

Restituisce il numero di periodi per un investimento sulla base di pagamenti costanti periodici e di un tasso di interesse costante.

## Sintassi

DAX

```
NPER(<rate>, <pmt>, <pv>[, <fv>[, <type>]])
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per periodo.
pmt	Pagamento effettuato in ogni periodo. Non può cambiare per la durata dell'annualità. In genere, pmt contiene il capitale e gli interessi, ma non altre tariffe o imposte.
pv	Valore attuale o somma forfettaria pari al valore attuale di una serie di pagamenti futuri.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone il valore BLANK.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

## Valore restituito

Numero di periodi per un investimento.

## Osservazioni:

- Il parametro type viene arrotondato all'intero più vicino.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
12%	Tasso di interesse annuale
-100	Pagamento effettuato ogni periodo
-1000	Valore corrente
10000	Valore futuro
1	Il pagamento è dovuto all'inizio del periodo (vedere sopra)

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    NPER(0.12/12, -100, -1000, 10000, 1)  
}
```

Restituisce il numero di periodi per l'investimento in base ai termini specificati in precedenza.

[Valore]

59,6738656742946

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# ODDFPRICE

Articolo • 20/10/2023

Restituisce il prezzo per un valore nominale di \$100 di un titolo il cui primo periodo di durata è irregolare (lungo o breve).

## Sintassi

DAX

```
ODDFPRICE(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
problema	Data di emissione del titolo.
first_coupon	Data della prima cedola del titolo.
rate	Tasso di interesse del titolo.
yld	Rendimento annuale del titolo.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Prezzo per il valore nominale \$100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La funzione ODDFPRICE viene calcolata come segue:

Prima cedola con periodo irregolare breve:

$$\text{ODDFPRICE} = \left[ \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N-1+\frac{\text{DSC}}{\text{E}})}} \right] + \left[ \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{\text{E}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\frac{\text{DSC}}{\text{E}})}} \right] + \left[ \sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k-1+\frac{\text{DSC}}{\text{E}})}} \right] - \left[ 100 \times \frac{\text{rate}}{\text{frequency}} \right] >$$

dove:

- A = numero di giorni dall'inizio del periodo di cedola alla data di regolamento (giorni maturati).
- DSC = numero di giorni dal regolamento alla data della cedola successiva.
- DFC = numero di giorni dall'inizio della prima cedola irregolare alla data della prima cedola.
- E = numero di giorni nel periodo di cedola.
- N = numero di cedole pagabili tra la data di regolamento e la data di riscatto. Se questo numero contiene una frazione, viene aumentata al numero intero successivo.

**Prima cedola con periodo irregolare lungo:**

$$\text{ODDFPRICE} = \left[ \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N+N_q+\frac{\text{DSC}}{\text{E}})}} \right] + \left[ \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \left[ \sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i} \right]}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N_q+\frac{\text{DSC}}{\text{E}})}} \right] + \left[ \sum_{k=1}^{\text{N}} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k-N_q+\frac{\text{DSC}}{\text{E}})}} \right] - \left[ 100 \times \frac{\text{rate}}{\text{frequency}} \right] >$$

dove:

- $A_i$  = numero di giorni dall'inizio del periodo di quasi cedola  $i^n$ , ovvero l'ultimo, all'interno del periodo di durata irregolare.
- $DC_i$  = numero di giorni dalla data di inizio (o di emissione) alla prima quasi cedola ( $i = 1$ ) oppure numero di giorni nel periodo di quasi cedola ( $i = 2, \dots, i = \text{NC}$ ).
- DSC = numero di giorni dal regolamento alla data della cedola successiva.
- E = numero di giorni nel periodo della cedola.
- N = numero di cedole pagabili tra la data della prima cedola reale e la data di riscatto. Se questo numero contiene una frazione, viene aumentata al numero intero successivo.
- NC = numero di periodi di quasi cedola che rientrano nel periodo di durata irregolare. Se questo numero contiene una frazione, viene aumentata al numero intero successivo.
- $\text{NL}_i$  = durata normale in giorni dell'intero periodo di quasi cedola  $i^n$ , ovvero l'ultimo, all'interno del periodo di durata irregolare.
- $N_q$  = numero di periodi di quasi cedola compresi tra la data di regolamento e la prima cedola.

- I parametri settlement, maturity, issue e first\_coupon vengono troncati a numeri interi.
- I parametri basis e frequency vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity, issue o first\_coupon non rappresenta una data valida.
  - La condizione maturity > first\_coupon > settlement > issue non è soddisfatta.
  - rate < 0.
  - yld < 0.
  - redemption ≤ 0.
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione dell'argomento
11/11/2008	Data di regolamento
1/3/2021	Data di scadenza
15/10/2008	Data di emissione
1/3/2009	Data prima cedola
7,85%	Percentuale cedola
6,25%	Percentuale rendimento
\$100,00	Valore di riscatto
2	La frequenza è semestrale
1	Base giorni effettivi/giorni effettivi

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    ODDFPRICE(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15), DATE(2009,3,1), 0.0785, 0.0625, 100.00, 2, 1)  
}
```

Restituisce il prezzo per un valore nominale di \$100 di un titolo il cui primo periodo di durata è irregolare (lungo o breve), in base ai termini specificati in precedenza.

[Valore]
113,597717474079

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# ODDFYIELD

Articolo • 20/10/2023

Restituisce il rendimento di un titolo il cui primo periodo di durata è irregolare (breve o lungo).

## Sintassi

DAX

```
ODDFYIELD(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
problematico	Data di emissione del titolo.
first_coupon	Data della prima cedola del titolo.
rate	Tasso di interesse del titolo.
pr	Prezzo del titolo.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360

Basis	Base per conteggio dei giorni
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Rendimento del titolo.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La funzione ODDFYIELD viene calcolata usando un metodo iterativo. Usa il metodo Newton in base alla formula usata per la funzione ODDFPRICE. Il rendimento viene modificato tramite 100 iterazioni fino a quando il prezzo stimato con il rendimento specificato non è prossimo al prezzo. Vedere ODDFPRICE per la formula usata da ODDFYIELD.
- I parametri settlement, maturity, issue e first\_coupon vengono troncati a numeri interi.
- I parametri basis e frequency vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity, issue o first\_coupon non rappresenta una data valida.

- La condizione maturity > first\_coupon > settlement > issue non è soddisfatta.
- rate < 0.
- pr ≤ 0.
- redemption ≤ 0.
- Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
- base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione dell'argomento
11 novembre 2008	Data di regolamento
1 marzo 2021	Data di scadenza
15 ottobre 2008	Data di emissione
1 marzo 2009	Data prima cedola
5,75%	Percentuale cedola
84,50	Prezzo
100	Valore di riscatto
2	La frequenza è semestrale
0	Base 30/360

La query DAX seguente:

```
DAX
EVALUATE
{
    ODDFYIELD(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15),
    DATE(2009,3,1), 0.0575, 84.50, 100, 2, 0)
}
```

Restituisce il rendimento di un titolo il cui primo periodo di durata è irregolare (breve o lungo), in base ai termini specificati in precedenza.

[Valore]

0,0772455415972989

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ODDLPRICE

Articolo • 20/10/2023

Restituisce il prezzo per un valore nominale di \$100 di un titolo il cui ultimo periodo di cedola è irregolare (lungo o breve).

## Sintassi

DAX

```
ODDLPRICE(<settlement>, <maturity>, <last_interest>, <rate>, <yld>,
<redemption>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
last_interest	Data dell'ultima cedola del titolo.
rate	Tasso di interesse del titolo.
yld	Rendimento annuale del titolo.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo

Basis	Base per conteggio dei giorni
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Prezzo per il valore nominale \$100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement, maturity e last\_interest vengono troncati a numeri interi.
- I parametri basis e frequency vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity o last\_interest non rappresenta una data valida.
  - La condizione di data maturity > settlement > last\_interest non è soddisfatta.
  - rate < 0.
  - yld < 0.
  - redemption ≤ 0.
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La query DAX seguente:

Dati	Descrizione dell'argomento
7 febbraio 2008	Data di regolamento
15 giugno 2008	Data di scadenza
15 ottobre 2007	Data dell'ultimo interesse
3,75%	Percentuale cedola
4,05%	Percentuale rendimento
\$100	Valore di riscatto
2	La frequenza è semestrale
0	Base 30/360

DAX

```
EVALUATE
{
    ODDLPRICE(DATE(2008,2,7), DATE(2008,6,15), DATE(2007,10,15), 0.0375,
    0.0405, 100, 2, 0)
}
```

Restituisce il prezzo per un valore nominale di \$100 di un titolo il cui ultimo periodo di cedola è irregolare (lungo o breve), in base ai termini specificati in precedenza.

[Valore]

99,8782860147213

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì    No

# ODDLYIELD

Articolo • 20/10/2023

Restituisce il rendimento di un titolo il cui ultimo periodo di durata è irregolare (breve o lungo).

## Sintassi

DAX

```
ODDLYIELD(<settlement>, <maturity>, <last_interest>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
last_interest	Data dell'ultima cedola del titolo.
rate	Tasso di interesse del titolo.
pr	Prezzo del titolo.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Rendimento del titolo.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il

1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.

- La funzione ODDLYIELD viene calcolata come segue:

$$\text{ODDLYIELD} = \left[ \frac{\left( \text{redemption} + \left( \left( \sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right) \right) - \left( \text{par} + \left( \left( \sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right) \right)}{\text{par} + \left( \left( \sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right)} \right] \times \left[ \frac{\text{frequency}}{\left( \sum_{i=1}^{\text{NC}} \frac{\text{DSC}_i}{\text{NL}_i} \right)} \right]$$

dove:

- $\text{A}_i$  = numero di giorni accumulati per il  $i^{th}$ , o l'ultimo periodo di quasi cedola entro un periodo dispari contando in avanti dall'ultima data di interesse prima del riscatto.
- $\text{DC}_i$  = numero di giorni conteggiati nel periodo di quasi cedola  $i^n$ , ovvero l'ultimo, come delimitato dalla lunghezza dell'effettivo periodo della cedola.
- NC = numero di periodi di quasi cedola che rientrano nel periodo di durata irregolare. Se questo numero contiene una frazione, verrà aumentato al numero intero successivo.
- $\text{NL}_i$  = durata normale in giorni del periodo di quasi cedola  $i^n$ , o l'ultimo, all'interno del periodo di durata irregolare della cedola.
- I parametri settlement, maturity e last\_interest vengono troncati a numeri interi.
- I parametri basis e frequency vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity o last\_interest non rappresenta una data valida.
  - La condizione di data maturity > settlement > last\_interest non è soddisfatta.
  - rate < 0.
  - pr ≤ 0.
  - redemption ≤ 0.
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

Dati	Descrizione dell'argomento
20/4/2008	Data di regolamento
15/6/2008	Data di scadenza
24/12/2007	Data dell'ultimo interesse
3,75%	Percentuale cedola
\$99.875	Prezzo
\$100	Valore di riscatto
2	La frequenza è semestrale
0	Base 30/360

DAX

```
EVALUATE
{
```

```
ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375, 99.875, 100, 2, 0)
}
```

Restituisce il rendimento di un titolo il cui ultimo periodo di durata è irregolare (breve o lungo), in base ai termini specificati in precedenza.

[Valore]

0,0451922356291692

## Commenti e suggerimenti

Questa pagina è stata utile?

 Si

 No

# PDURATION

Articolo • 20/10/2023

Restituisce il numero di periodi necessari affinché un investimento raggiunga un valore specificato.

## Sintassi

DAX

```
PDURATION(<rate>, <pv>, <fv>)
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per periodo.
pv	Valore attuale dell'investimento.
fv	Valore futuro desiderato dell'investimento.

## Valore restituito

Numero di periodi.

## Osservazioni:

- PDURATION usa l'equazione seguente:

$$\text{PDURATION} = \frac{\log(\text{fv}) - \log(\text{pv})}{\log(1 + \text{rate})}$$

- Viene restituito un errore nei casi seguenti:
  - $\text{rate} \leq 0$ .
  - $\text{pv} \leq 0$ .
  - $\text{fv} \leq 0$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    PDURATION(0.025, 2000, 2200)  
}
```

Restituisce il numero di anni necessari affinché un investimento pari a \$2000, che guadagna il 2,5% annualmente, raggiungerà \$2200.

[Valore]

3,85986616262266

## Esempio 2

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    PDURATION(0.025/12, 1000, 1200)  
}
```

Restituisce il numero di mesi necessari affinché un investimento pari a \$1000, che guadagna il 2,5% annualmente, raggiunga \$1200.

[Valore]

87,6054764193714

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# PMT

Articolo • 20/10/2023

Calcola il pagamento per un prestito sulla base di pagamenti costanti e di un tasso di interesse costante.

## Sintassi

DAX

```
PMT(<rate>, <nper>, <pv>[, <fv>[, <type>]])
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per il prestito.
nper	Numero totale dei pagamenti per il prestito.
pv	Valore attuale o importo totale pari al valore attuale di una serie di pagamenti futuri. Noto anche come capitale.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone il valore BLANK.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

**Nota:** Per una descrizione più completa degli argomenti in PMT, vedere la funzione PV.

## Valore restituito

Importo di un singolo pagamento del prestito.

## Osservazioni:

- Il valore del pagamento restituito da PMT include il capitale e gli interessi, ma non include tasse, riserve o altri costi a volte associati ai prestiti.
- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuo pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - nper < 1

**Suggerimento:** per trovare l'importo totale pagato per la durata del prestito, moltiplicare il valore PMT restituito per nper.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

### Esempio 1

Dati	Descrizione
%8	Tasso di interesse annuale
10	Numero di mesi di pagamenti
\$ 10.000	Importo del prestito

La query DAX seguente:

```
DAX
EVALUATE
{
    PMT(0.08/12, 10, 10000, 0, 1)
}
```

Restituisce l'importo del pagamento mensile, versato all'inizio del mese, per un prestito con le condizioni specificate in precedenza.

[Valore]
-1030,16432717797

**Nota:** 1030,16432717797 è il pagamento per periodo. Di conseguenza, l'importo totale pagato per la durata del prestito è di circa  $1030,16 * 10 = \$ 10.301,60$ . In altri termini, vengono pagati approssimativamente \$ 301,60 di interessi.

## Esempio 2

Dati	Descrizione
%6	Tasso di interesse annuale
18	Numero di anni di pagamenti
\$ 50.000	Importo del prestito

La query DAX seguente:

DAX
<pre>EVALUATE {     PMT(0.06/12, 18*12, 0, 50000) }</pre>

[Valore]
-129,081160867991

Restituisce l'importo da risparmiare ogni mese per ottenere \$ 50.000 alla fine dei 18 anni, in base ai termini specificati in precedenza.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PPMT

Articolo • 20/10/2023

Restituisce il pagamento sul capitale di un investimento per un determinato periodo sulla base di pagamenti costanti periodici e di un tasso di interesse costante.

## Sintassi

DAX

```
PPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per il prestito.
per	Specifica il periodo. Deve essere compreso tra 1 e nper (inclusi).
nper	Numero totale dei periodi di pagamento in un'annualità.
pv	Valore attuale o importo complessivo pari al valore corrente di una serie di pagamenti futuri.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone il valore BLANK.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

**Nota:** Per una descrizione più completa degli argomenti in PPMT, vedere la funzione PV.

## Valore restituito

Pagamento sul capitale per un periodo specificato.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuo pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - per < 1 o per > nper
  - nper < 1
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

Dati	Descrizione dell'argomento
10%	Tasso di interesse annuale
2	Numero di anni per il prestito
\$2.000,00	Importo del prestito

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    PPMT(0.1/12, 1, 2*12, 2000.00)  
}
```

Restituisce il pagamento sul capitale effettuato il primo mese per un prestito con i termini specificati in precedenza.

[Valore]
-75,6231860083663

## Esempio 2

Dati	Descrizione dell'argomento
%8	Tasso di interesse annuale
10	Numero di anni per il prestito
\$200.000,00	Importo del prestito

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    PPMT(0.08, 10, 10, 200000.00)  
}
```

Restituisce il pagamento sul capitale effettuato il decimo anno per un prestito con i termini specificati in precedenza.

[Valore]

-27598,0534624214

## Commenti e suggerimenti

Questa pagina è stata utile?



# PREZZO

Articolo • 20/10/2023

Restituisce il prezzo per un valore nominale di \$ 100 per un titolo i cui interessi vengono pagati su base periodica.

## Sintassi

DAX

```
PRICE(<settlement>, <maturity>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

## Parametri

Termino	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
rate	Tasso di interesse annuo della cedola del titolo.
yld	Rendimento annuale del titolo.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Prezzo per il valore nominale \$100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli.  
In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri basis e frequency vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - $\text{rate} < 0$ .
  - $\text{yld} < 0$ .
  - $\text{redemption} \leq 0$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

**Importante:**

- Quando  $N > 1$  ( $N$  è il numero di cedole pagabili tra la data di liquidazione e la data di riscatto), **PRICE** viene calcolato come segue:
$$\text{PRICE} = \left[ \frac{\text{redemption}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{(N-1+\frac{\text{DSC}}{\text{E}})}} \right] + \left[ \sum_{k=1}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{(k-1+\frac{\text{DSC}}{\text{E}})}} \right] - \left[ 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}} \right]$$
- Quando  $N = 1$  ( $N$  è il numero di cedole pagabili tra la data di liquidazione e la data di riscatto), **PRICE** viene calcolato come segue:

$$\text{DSR} = \text{E} - \text{A}$$

$$T1 = 100 \times \frac{\text{rate}}{\text{frequency}} + \text{redemption}$$

$$T2 = \frac{\text{yld}}{\text{frequency}} \times \frac{\text{DSR}}{\text{E}} + 1$$

$$T3 = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}$$

$$\text{PRICE} = \frac{T1}{T2} - T3$$

dove:

- $\text{DSC}$  = numero di giorni dal regolamento alla data della cedola successiva.
- $\text{E}$  = numero di giorni nel periodo della cedola in cui cade la data di regolamento.

- A = numero di giorni dall'inizio del periodo della cedola alla data di regolamento.

## Esempio

Dati	Descrizione dell'argomento
15/2/2008	Data di regolamento
11/15/2017	Data di scadenza
5,75%	Percentuale della cedola semestrale
6,50%	Percentuale rendimento
\$100	Valore di riscatto
2	La frequenza è semestrale
0	Base 30/360

La query DAX seguente:

```
DAX
EVALUATE
{
    PRICE(DATE(2008,2,15), DATE(2017,11,15), 0.0575, 0.065, 100, 2, 0)
}
```

Restituisce il prezzo per un'obbligazione in base ai termini specificati in precedenza.

[Valore]
94,6343616213221

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Si  No

# PRICEDISC

Articolo • 20/10/2023

Restituisce il prezzo per il valore nominale \$100 di un titolo scontato.

## Sintassi

DAX

```
PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
discount	Tasso di sconto del titolo.
redemption	Valore di riscatto del titolo per il valore nominale \$100.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Prezzo per il valore nominale \$100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2018 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2018, la data di regolamento sarà il 1° luglio 2018 e la data di scadenza sarà il 1° gennaio 2048, 30 anni dopo il 1° gennaio 2018, la data di emissione.
- La funzione PRICEDISC viene calcolata come segue:

$$\text{PRICEDISC} = \text{redemption} - \text{discount} \times \text{redemption} \times \frac{\text{DSM}}{\text{B}}$$

dove:

- B = numero di giorni nell'anno, a seconda della base per l'anno.
- DSM = numero di giorni dal regolamento alla scadenza.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - settlement  $\geq$  maturity.
  - discount  $\leq$  0.
  - redemption  $\leq$  0.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione dell'argomento
16/2/2008	Data di regolamento
1/3/2008	Data di scadenza
5,25%	Tasso di sconto percentuale
\$100	Valore di riscatto
2	Base giorni effettivi/360

La query DAX seguente:

```
DAX
EVALUATE
{
    PRICEDISC(DATE(2008,2,16), DATE(2008,3,1), 0.0525, 100, 2)
}
```

Restituisce il prezzo delle obbligazioni per il valore nominale \$100, per un'obbligazione con i termini specificati in precedenza.

[Valore]
99,795833333333

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì    No

# PRICEMAT

Articolo • 20/10/2023

Restituisce il prezzo per un valore nominale di \$ 100 per un titolo i cui interessi vengono pagati alla scadenza.

## Sintassi

DAX

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
problematico	Data di emissione del titolo.
rate	Tasso di interesse del titolo alla data di emissione.
yld	Rendimento annuale del titolo.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

# Valore restituito

Prezzo per il valore nominale \$100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La funzione PRICEMAT viene calcolata come segue:

$$\text{PRICEMAT} = \frac{100 + \left( \frac{\text{DIM}}{\text{B}} \times \text{rate} \times 100 \right)}{1 + \left( \frac{\text{DSM}}{\text{B}} \times \text{yld} \right)} - \left( \frac{\text{A}}{\text{B}} \times \text{rate} \times 100 \right)$$

dove:

- B = numero di giorni nell'anno, a seconda della base per l'anno.
- DSM = numero di giorni dal regolamento alla scadenza.
- DIM = numero di giorni dall'emissione alla scadenza.
- A = numero di giorni dall'emissione al regolamento.
- I parametri settlement, maturity e issue vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity o issue non rappresenta una data valida.
  - La condizione di data maturity > settlement > issue non è soddisfatta.
  - rate < 0.
  - yld < 0.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La query DAX seguente:

Dati	Descrizione
15/2/2008	Data di regolamento
13/4/2008	Data di scadenza
11/11/2007	Data di emissione
6,10%	Percentuale della cedola semestrale
6,10%	Percentuale rendimento
0	Base 30/360

DAX

```
EVALUATE
{
    PRICEMAT(DATE(2008,2,15), DATE(2008,4,13), DATE(2007,11,11), 0.061, 0.061,
    0)
}
```

Restituisce il prezzo per valore nominale di \$ 100 di un titolo con i termini specificati in precedenza.

[Valore]

99,9844988755569

## Commenti e suggerimenti

Questa pagina è stata utile?



# PV

Articolo • 20/10/2023

Calcola il valore attuale di un investimento sulla base di un tasso di interesse costante. È possibile usare PV con pagamenti costanti periodici, ad esempio un mutuo o un'altra forma di prestito, o con un valore futuro che rappresenta l'obiettivo dell'investimento.

## Sintassi

DAX

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

## Parametri

Termine	Definizione
rate	Tasso di interesse per periodo. Ad esempio, se si ottiene un prestito automobilistico con un tasso di interesse annuale pari al 10% e si effettuano pagamenti mensili, il tasso di interesse mensile è 0,1/12 o 0,0083. Il valore di rate da inserire nella formula sarebbe 0,1/12 o 0,0083.
nper	Numero totale dei periodi di pagamento in un'annualità. Ad esempio, se si ottiene un prestito automobilistico della durata di quattro anni e si effettuano pagamenti mensili, il prestito ha 4*12 (o 48) periodi. Il valore di nper da inserire nella formula sarebbe 48.
pmt	Pagamento effettuato in ogni periodo, che non può cambiare per la durata dell'annualità. In genere, pmt include il capitale e gli interessi, ma non altre tariffe o imposte. Ad esempio, i pagamenti mensili per un prestito di \$ 10.000 della durata di quattro anni al 12% equivalgono a \$ 263,33. Il valore di pmt da inserire nella formula sarebbe -263,33.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone il valore BLANK. Se, ad esempio, si vogliono risparmiare \$ 50.000 in 18 anni per finanziare un progetto speciale, il valore futuro è \$ 50.000. Si potrebbe quindi fare una previsione prudenziale su un tasso di interesse e determinare quanto occorre risparmiare ogni mese.
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo
1	All'inizio del periodo

## Valore restituito

Valore attuale di un prestito o un investimento.

### Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di rate e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 12%, usare 0,12/12 per rate e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per rate e 4 per nper.
- Le funzioni seguenti si applicano alle annualità:
  - CUMIPMT
  - CUMPRINC
  - FV
  - IPMT
  - PMT
  - PPMT
  - PV
  - RATE
  - XIRR
  - XNPV
- Un'annualità è una serie di pagamenti costanti effettuati in un periodo continuato. Ad esempio, i prestiti per l'acquisto di un'automobile e i mutui sono annualità. Per altre informazioni, vedere la descrizione di ciascuna funzione per il calcolo delle annualità.
- Nelle funzioni per il calcolo delle annualità, le somme versate, quali i depositi effettuati su un conto, sono rappresentate da un numero negativo, mentre le somme incassate, quali i dividendi, sono rappresentate da un numero positivo. Ad esempio, un deposito bancario di \$ 1.000 sarà rappresentato dall'argomento -1000 per chi lo effettua e dall'argomento 1000 per la banca.
- Ogni argomento finanziario viene calcolato in base agli altri.

- Se rate non è 0:

$$pv \times (1 + \text{rate})^{\text{nper}} + \text{pmt}(1 + \text{rate} \times \text{type}) \times \left( \frac{(1 + \text{rate})^{\text{nper}} - 1}{\text{rate}} \right) + \text{fv} = 0$$

- Se rate è 0:

$$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$$

- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - nper < 1 o vuoto
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
\$ 500,00	Somma di denaro pagata per l'annualità di un'assicurazione alla fine di ogni mese.
%8	Tasso di interesse guadagnato sulla somma pagata.
20	Anni per cui dovranno essere effettuati i pagamenti.

La query DAX seguente:

```
DAX
EVALUATE
{
    PV(0.08/12, 12*20, 500.00, 0, 0)
```

Restituisce il valore attuale di un'annualità in base ai termini specificati in precedenza.

[Valore]
-59777,1458511878

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# RATE

Articolo • 20/10/2023

Restituisce il tasso di interesse per periodo di un'annualità. La funzione RATE viene calcolata per iterazione e può avere zero o più soluzioni. Se i risultati successivi di RATE non convergono a un valore entro 0,0000001 dopo 20 iterazioni, viene restituito un errore.

## Sintassi

DAX

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]]])
```

## Parametri

Termine	Definizione
nper	Numero totale dei periodi di pagamento in un'annualità.
pmt	Pagamento effettuato in ogni periodo e che non può cambiare per la durata dell'annualità. In genere, pmt include il capitale e gli interessi, ma non altre tariffe o imposte.
pv	Valore attuale o importo complessivo pari al valore corrente di una serie di pagamenti futuri.
fv	(Facoltativo) Valore futuro o saldo di cassa che si vuole ottenere dopo l'ultimo pagamento. Se fv viene omesso, si presuppone che sia 0 (il valore futuro di un prestito, ad esempio, è 0).
Tipo	(Facoltativo) Numero 0 o 1 che indica quando sono dovuti i pagamenti. Se type viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.
guess	(Facoltativo) Previsione del tasso di interesse futuro. - Se omesso, si presuppone che sia pari al 10%. - Se i risultati di RATE non convergono, provare a usare dei valori differenti per guess. In genere, i risultati di RATE convergono se guess è compreso tra 0 e 1.

Il parametro **type** accetta i valori seguenti:

Impostare type su	Se i pagamenti sono dovuti
0 o omesso	Alla fine del periodo

Impostare type su	Se i pagamenti sono dovuti
1	All'inizio del periodo

## Valore restituito

Tasso di interesse per periodo.

## Osservazioni:

- Assicurarsi di usare con coerenza le unità di misura per specificare i valori di guess e nper. Se si effettuano pagamenti mensili per un prestito di quattro anni con un tasso di interesse annuale pari al 12%, usare 0,12/12 per guess e 4\*12 per nper. Se si effettuano pagamenti annuali per lo stesso prestito, usare 0,12 per guess e 4 per nper.
- Il parametro type viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - nper ≤ 0.
  - I risultati di RATE non convergono entro 0,0000001 dopo 20 iterazioni
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
4	Anni del prestito
200-	Pagamento mensile
8000	Importo del prestito

## Esempio 1

La query DAX seguente:

DAX

```
EVALUATE
{
    RATE(4*12, -200, 8000)
}
```

Restituisce il tasso mensile del prestito in base ai termini specificati in precedenza.

[Valore]

0,00770147248820137

## Esempio 2

La query DAX seguente:

```
DAX
EVALUATE
{
    RATE(4*12, -200, 8000) * 12
}
```

Restituisce il tasso annuale del prestito in base ai termini specificati in precedenza.

[Valore]

0,0924176698584164

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATA RICEZIONE

Articolo • 20/10/2023

Restituisce l'importo ricevuto alla scadenza di un titolo interamente investito.

## Sintassi

DAX

```
RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
investment	Importo investito nel titolo.
discount	Tasso di sconto del titolo.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Importo ricevuto alla scadenza.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- La funzione RECEIVED viene calcolata come segue:

$$\text{RECEIVED} = \frac{\text{investment}}{1 - (\text{discount} \times \frac{\text{DIM}}{\text{B}})}$$

dove:

- B = numero di giorni in un anno, a seconda della base per l'anno.
- DIM = numero di giorni dall'emissione alla scadenza.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - settlement  $\geq$  maturity.
  - investment  $\leq$  0.
  - discount  $\leq$  0.
  - base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

Dati	Descrizione
15-feb-08	Data di regolamento (emissione)
15-mag-08	Data di scadenza
\$ 1.000.000,00	Investimento
5,75%	Tasso di sconto percentuale
2	Base giorni effettivi/360

DAX

```
EVALUATE
{
    RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

Restituisce l'importo totale da ricevere alla scadenza, per un'obbligazione con i termini specificati in precedenza.

[Valore]

1014584,6544071

## Commenti e suggerimenti

Questa pagina è stata utile?



# RRI

Articolo • 20/10/2023

Restituisce un tasso di interesse equivalente per l'incremento di un investimento.

## Sintassi

DAX

```
RRI(<nper>, <pv>, <fv>)
```

## Parametri

Termine	Definizione
nper	Numero di periodi per l'investimento.
pv	Valore attuale dell'investimento.
fv	Valore futuro dell'investimento.

## Valore restituito

Tasso di interesse equivalente.

## Osservazioni:

- RRI restituisce il tasso di interesse dato nper (numero di periodi), pv (valore attuale) e fv (valore futuro), calcolato usando l'equazione seguente:

$$\left( \frac{fv}{pv} \right)^{\left( \frac{1}{nper} \right)} - 1$$

- Viene restituito un errore nei casi seguenti:
  - nper ≤ 0.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Dati	Descrizione
\$10.000	Valore corrente
\$21.000	Valore futuro
4	Anni investiti

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    RRI(4*12, 10000, 21000)  
}
```

Restituisce un tasso di interesse equivalente per l'incremento di un investimento con i termini specificati in precedenza.

[Valore]
0,0155771057566627

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì No

# SLN

Articolo • 20/10/2023

Restituisce l'ammortamento lineare di un cespote per un periodo.

## Sintassi

DAX

```
SLN(<cost>, <salvage>, <life>)
```

## Parametri

Termine	Definizione
cost	Costo iniziale del bene.
salvage	Valore al termine dell'ammortamento (talvolta chiamato valore residuo del cespote).
dei dati	Numero di periodi di ammortamento del cespote (talvolta chiamato vita utile del cespote).

## Valore restituito

Ammortamento lineare per un periodo.

## Osservazioni:

- Viene restituito un errore nei casi seguenti:  
 $life = 0$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
\$30.000	Costi

Dati	Descrizione
\$7.500	Valore di realizzo
10	Anni di vita utile

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    SLN(30000, 7500, 10)  
}
```

Restituisce la quota di ammortamento annuale in base ai termini specificati in precedenza.

[Valore]
2250

## Commenti e suggerimenti

Questa pagina è stata utile?



# SYD

Articolo • 20/10/2023

Restituisce l'ammortamento anticipato a quote decrescenti di un cespite per un periodo specificato.

## Sintassi

DAX

```
SYD(<cost>, <salvage>, <life>, <per>)
```

## Parametri

Termine	Definizione
cost	Costo iniziale del bene.
salvage	Valore al termine dell'ammortamento (talvolta chiamato valore residuo del cespite).
dei dati	Numero di periodi di ammortamento del cespite (talvolta chiamato vita utile del cespite).
per	Periodo. Deve usare le stesse unità di life. Deve essere compreso tra 1 e life (inclusi).

## Valore restituito

Ammortamento anticipato a quote decrescenti per il periodo specificato.

## Osservazioni:

- La funzione SYD viene calcolata come segue:

$$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life}) \times (\text{life} + 1)}$$

- Viene restituito un errore nei casi seguenti:
  - vita < 1.
  - per < 1 o per > life.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
\$30.000,00	Costo iniziale
\$7.500,00	Valore di realizzo
10	Durata in anni

### Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 1)
```

Restituisce la quota di ammortamento anticipato a quote decrescenti di un cespote per il primo anno, in base ai termini specificati in precedenza.

[Valore]
4090,90909090909

### Esempio 2

La query DAX seguente:

```
DAX
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 10)
```

Restituisce la quota di ammortamento anticipato a quote decrescenti di un cespote per il decimo anno (finale), in base ai termini specificati in precedenza.

[Valore]

409,090909090909

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# TBILLEQ

Articolo • 20/10/2023

Restituisce il rendimento equivalente a un'obbligazione per un buono del tesoro.

## Sintassi

DAX

```
TBILLEQ(<settlement>, <maturity>, <discount>)
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del buono del tesoro. La data di regolamento del titolo è la data successiva alla data di emissione in cui il buono del tesoro viene ceduto all'acquirente.
maturity	Data di scadenza del buono del tesoro. La data di scadenza è la data in cui scade il buono del tesoro.
discount	Tasso di sconto del buono del tesoro.

## Valore restituito

Rendimento equivalente a un'obbligazione per un buono del tesoro.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La funzione TBILLEQ viene calcolata come segue:

$$\text{TBILLEQ} = \frac{365 \times \text{discount}}{360 - (\text{discount} \times \text{DSM})}$$

dove:

- DSM indica il numero di giorni tra il regolamento e la scadenza calcolati in base all'anno di 360 giorni.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - Il parametro settlement  $\geq$  maturity o maturity è successivo a settlement per più di un anno.
  - discount  $\leq 0$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
3/31/2008	Data di regolamento
1/6/2008	Data di scadenza
9,14%	Tasso di sconto percentuale

La query DAX seguente:

```
DAX
EVALUATE
{
    TBILLEQ(DATE(2008,3,31), DATE(2008,6,1), 0.0914)
}
```

Restituisce il rendimento equivalente a un'obbligazione per un buono del tesoro in base ai termini specificati in precedenza.

[Valore]
0,094151493565943

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# TBILLPRICE

Articolo • 20/10/2023

Restituisce il prezzo per un valore nominale di \$ 100 per un buono del tesoro.

## Sintassi

DAX

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del buono del tesoro. La data di regolamento del titolo è la data successiva alla data di emissione in cui il buono del tesoro viene ceduto all'acquirente.
maturity	Data di scadenza del buono del tesoro. La data di scadenza è la data in cui scade il buono del tesoro.
discount	Tasso di sconto del buono del tesoro.

## Valore restituito

Prezzo del buono del tesoro con valore nominale di \$ 100.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La funzione TBILLPRICE viene calcolata come segue:

$$\text{TBILLPRICE} = 100 \times \left(1 - \frac{\text{discount} \times \text{DSM}}{360}\right)$$

dove:

- DSM = numero di giorni dal regolamento alla scadenza, esclusa un'eventuale data di scadenza successiva alla data di regolamento per più di un anno di calendario.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - Il parametro settlement  $\geq$  maturity o maturity è successivo a settlement per più di un anno.
  - discount  $\leq 0$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
3/31/2008	Data di regolamento
1/6/2008	Data di scadenza
9,0%	Tasso di sconto percentuale

La query DAX seguente:

```
DAX
EVALUATE
{
    TBILLPRICE(DATE(2008,3,31), DATE(2008,6,1), 0.09)
}
```

Restituisce il prezzo del buono del tesoro per valore nominale di \$ 100 in base ai termini specificati in precedenza.

[Valore]
98,45

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# TBILLYIELD

Articolo • 20/10/2023

Restituisce il rendimento per un buono del tesoro.

## Sintassi

DAX

```
TBILLYIELD(<settlement>, <maturity>, <pr>)
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del buono del tesoro. La data di regolamento del titolo è la data successiva alla data di emissione in cui il buono del tesoro viene ceduto all'acquirente.
maturity	Data di scadenza del buono del tesoro. La data di scadenza è la data in cui scade il buono del tesoro.
pr	Prezzo del buono del tesoro per \$100 di valore nominale.

## Valore restituito

Rendimento del buono del tesoro.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La funzione TBILLYIELD viene calcolata come segue:

$$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

dove:

- DSM = numero di giorni dal regolamento alla scadenza, esclusa un'eventuale data di scadenza successiva alla data di regolamento per più di un anno di calendario.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - Il parametro settlement  $\geq$  maturity o maturity è successivo a settlement per più di un anno.
  - $pr \leq 0$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

Dati	Descrizione
3/31/2008	Data di regolamento
1/6/2008	Data di scadenza
\$98,45	Prezzo per \$100 di valore nominale

DAX

```
EVALUATE
{
    TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45)
}
```

Restituisce il rendimento di un buono del tesoro in base ai termini specificati in precedenza.

[Valore]
0,0914169629253426

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# VDB

Articolo • 20/10/2023

Restituisce l'ammortamento di un cespote per un periodo specificato, inclusi periodi parziali, usando il metodo di ammortamento a quote dimezzate o un altro metodo specificato dall'utente. VDB significa residuo variabile decrescente.

## Sintassi

DAX

```
VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[, <no_switch>]])
```

## Parametri

Termino	Definizione
cost	Costo iniziale del bene.
salvage	Valore al termine dell'ammortamento (talvolta chiamato valore residuo del cespote). Questo valore può essere 0.
dei dati	Numero di periodi di ammortamento del cespote (talvolta chiamato vita utile del cespote).
start_period	Periodo iniziale per il quale si vuole calcolare l'ammortamento. Start_period deve usare le stesse unità di life. Deve essere compreso tra 1 e life (inclusi).
end_period	Periodo finale per il quale si vuole calcolare l'ammortamento. End_period deve usare le stesse unità di life. Deve essere compreso tra start_period e life (inclusi).
factor	(Facoltativo) Quota di ammortamento. Se factor viene omesso, si presuppone il valore 2 (metodo di ammortamento a quote dimezzate). Modificare factor se non si vuole usare il metodo di ammortamento a quote dimezzate. Per una descrizione del metodo di ammortamento a quote dimezzate, vedere DDB.
no_switch	(Facoltativo) Valore logico che specifica se passare all'ammortamento lineare quando l'ammortamento è maggiore del calcolo della quota decrescente. Se type viene omesso, si presuppone il valore FALSE. - Se no-switch restituisce TRUE, VDB non passa all'ammortamento lineare, neanche quando l'ammortamento è maggiore del calcolo della quota decrescente. - Se no-switch restituisce FALSE o viene omesso, VDB passa all'ammortamento lineare quando l'ammortamento è maggiore del calcolo della quota decrescente.

# Valore restituito

Ammortamento per il periodo specificato.

## Osservazioni:

- Viene restituito un errore nei casi seguenti:
  - cost < 0.
  - salvage < 0.
  - life < 1.
  - start\_period < 1 o start\_period > end\_period.
  - end\_period < start\_period o end\_period > life.
  - factor < 0.
  - no\_switch non restituisce TRUE o FALSE.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

Dati	Descrizione
2400	Costo iniziale
300	Valore di realizzo
10	Durata in anni

## Esempio 1

La query DAX seguente:

```
DAX
EVALUATE
{
    VDB(2400, 300, 10*365, 0, 1)
}
```

Restituisce l'ammortamento del primo giorno di una cedola usando un fattore di 2.

[Valore]

1,31506849315068

## Esempio 2

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    VDB(2400, 300, 10*12, 6, 18, 3)  
}
```

Restituisce l'ammortamento di un cespite tra il mese 6<sup>o</sup> e il mese 18<sup>o</sup>. Questo calcolo usa un fattore di 3.

[Valore]

540,185558199698

## Esempio 3

La query DAX seguente:

```
DAX  
  
EVALUATE  
{  
    VDB(2400, 300, 10, 0, 0.875, 1.5)  
}
```

Restituisce l'ammortamento di un cespite nel primo anno fiscale di proprietà, presumendo che le leggi fiscali limitino l'ammortamento al 150% della quota decrescente. Il cespite viene acquistato a metà del primo trimestre dell'anno fiscale.

[Valore]

315

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# XIRR

Articolo • 20/10/2023

Restituisce il tasso interno di rendimento di una pianificazione di flussi di cassa non necessariamente periodica.

## Sintassi

DAX

```
XIRR(<table>, <values>, <dates>, [, <guess>[, <alternateResult>]]])
```

## Parametri

Termine	Definizione
table	Tabella per la quale è necessario calcolare le espressioni di valori e date.
valori	Espressione che restituisce il valore del flusso di cassa per ogni riga della tabella.
date	Espressione che restituisce la data del flusso di cassa per ogni riga della tabella.
guess	(Facoltativo) Ipotesi iniziale del tasso di rendimento interno. Se omesso, viene usata l'ipotesi predefinita 0,1.
alternateResult	(Facoltativo) Valore restituito al posto di un errore quando non è possibile determinare una soluzione.

## Valore restituito

Tasso di rendimento interno per gli input specificati. Se il calcolo non restituisce un risultato valido, viene restituito un errore o un valore specificato come alternateResult.

## Osservazioni:

- Il valore viene calcolato come tasso che soddisfa la funzione seguente:

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Dove:

- $P_j$  is the  $j^{th}$  payment
  - $d_j$  is the  $j^{th}$  payment date
  - $d_1$  is the first payment date
- La serie di valori del flusso di cassa deve contenere almeno un numero positivo e un numero negativo.
  - Evitare di usare funzioni ISERROR o IFERROR per acquisire un errore restituito da XIRR. Se alcuni input per la funzione possono generare un errore di soluzione, fornire un parametro alternateResult è il modo più affidabile e con prestazioni più elevate per gestire l'errore.
  - Per altre informazioni sull'uso del parametro alternateResult, vedere questo [video](#).
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente calcola il tasso di rendimento interno della tabella CashFlows:

DAX

```
= XIRR( CashFlows, [Payment], [Date] )
```

Data	Pagamento
1/1/2014	10000-
3/1/2014	2750
30/10/2014	4250
15/2/2015	3250
01/04/2015	2750

Tasso di rendimento = 37,49%

---

## Commenti e suggerimenti

Questo articolo è stato utile?

 Sì

 No

# XNPV

Articolo • 20/10/2023

Restituisce il valore attuale per una pianificazione di flussi di cassa non necessariamente periodica.

## Sintassi

DAX

```
XNPV(<table>, <values>, <dates>, <rate>)
```

## Parametri

Termine	Definizione
table	Tabella per la quale è necessario calcolare le espressioni di valori e date.
valori	Espressione che restituisce il valore del flusso di cassa per ogni riga della tabella.
date	Espressione che restituisce la data del flusso di cassa per ogni riga della tabella.
rate	Tasso di sconto da applicare al flusso di cassa per ogni riga della tabella.

## Valore restituito

Valore attuale netto.

## Osservazioni:

- Il valore viene calcolato come la somma seguente:

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Dove:

- $P_j$  is the  $j^{th}$  payment
- $d_j$  is the  $j^{th}$  payment date
- $d_1$  is the first payment date

- La serie di valori del flusso di cassa deve contenere almeno un numero positivo e un numero negativo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene calcolato il valore attuale della tabella CashFlows:

```
DAX
= XNPV( CashFlows, [Payment], [Date], 0.09 )
```

Data	Pagamento
1/1/2014	10000-
3/1/2014	2750
30/10/2014	4250
15/2/2015	3250
01/04/2015	2750

Valore attuale = 2086,65

## Commenti e suggerimenti

Questo articolo è stato utile?

Sì
 No

# YIELD

Articolo • 20/10/2023

Restituisce il rendimento per un titolo per cui viene pagato un interesse periodico. Usare YIELD per calcolare il rendimento dell'obbligazione.

## Sintassi

DAX

```
YIELD(<settlement>, <maturity>, <rate>, <pr>, <redemption>, <frequency>[,  
<basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
rate	Tasso di interesse annuo della cedola del titolo.
pr	Prezzo del titolo per \$100 di valore nominale.
redemption	Valore di riscatto del titolo per \$100 di valore nominale.
frequenza	Numero di pagamenti della cedola all'anno. Per i pagamenti annuali frequency = 1; per i pagamenti semestrali frequency = 2; per i pagamenti trimestrali frequency = 4.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360

Basis	Base per conteggio dei giorni
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Rendimento del titolo.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- Se al riscatto manca un periodo di cedola o meno, YIELD viene calcolato come segue:

$$YIELD = \frac{\left( \frac{\text{redemption}}{100} + \frac{\text{rate}}{\text{frequency}} \right) - \left( \frac{\text{par}}{100} + \left( \frac{A}{E} \times \frac{\text{rate}}{\text{frequency}} \right) \right)}{\frac{\text{par}}{100} + \left( \frac{A}{E} \times \frac{\text{rate}}{\text{frequency}} \right)} \times \frac{\text{frequency} \times E}{DSR}$$

dove:

- A = numero di giorni dall'inizio del periodo di cedola alla data di regolamento (giorni maturati).
- DSR = numero di giorni dalla data di regolamento alla data di riscatto.
- E = numero di giorni nel periodo di cedola.
- Se al riscatto manca più di un periodo di cedola, YIELD viene calcolato tramite cento iterazioni. Per la risoluzione viene usato il metodo Newton, in base alla formula usata per la funzione PRICE. Il rendimento viene modificato fino a quando il prezzo stimato dato il rendimento è prossimo al prezzo.

- I parametri settlement e maturity vengono troncati a numeri interi.
- I parametri frequency e basis vengono arrotondati al numero intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - settlement  $\geq$  maturity.
  - rate  $< 0$ .
  - pr  $\leq 0$ .
  - redemption  $\leq 0$ .
  - Il valore di frequency è qualsiasi numero diverso da 1, 2 o 4.
  - base  $< 0$  o base  $> 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
15-feb-08	Data di regolamento
15-nov-16	Data di scadenza
5,75%	Percentuale cedola
95,04287	Prezzo
\$100	Valore di riscatto
2	La frequenza è semestrale (vedere sopra)
0	Base 30/360 (vedere sopra)

La query DAX seguente:

```
DAX
EVALUATE
{
    YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

Restituisce il rendimento di un'obbligazione con i termini specificati in precedenza.

[Valore]

0,0650000068807314

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# YIELDDISC

Articolo • 20/10/2023

Restituisce il rendimento annuale per un titolo scontato.

## Sintassi

DAX

```
YIELDDISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
pr	Prezzo del titolo per \$100 di valore nominale.
redemption	Valore di riscatto del titolo per il valore nominale \$100.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

## Valore restituito

Rendimento annuale.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement e maturity vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement o maturity non rappresenta una data valida.
  - $\text{settlement} \geq \text{maturity}$ .
  - $\text{pr} \leq 0$ .
  - $\text{redemption} \leq 0$ .
  - $\text{base} < 0$  o  $\text{base} > 4$ .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	-
16-feb-08	Data di regolamento
1-mar-08	Data di scadenza
99,795	Prezzo
\$100	Valore di riscatto
2	Base giorni effettivi/360

La query DAX seguente:

```
DAX
```

```
EVALUATE
{
    YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2)
}
```

Restituisce il rendimento annuale del titolo, in base ai termini specificati sopra.

[Valore]
0,0528225719868583

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# YIELDMAT

Articolo • 20/10/2023

Restituisce il rendimento annuo per un titolo i cui interessi vengono pagati alla scadenza.

## Sintassi

DAX

```
YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

## Parametri

Termine	Definizione
settlement	Data di regolamento del titolo. La data di regolamento del titolo è la data successiva alla data di emissione in cui il titolo viene ceduto all'acquirente.
maturity	Data di scadenza del titolo. La data di scadenza è la data in cui scade il titolo.
problematico	Data di emissione del titolo.
rate	Tasso di interesse del titolo alla data di emissione.
pr	Prezzo del titolo per \$100 di valore nominale.
basis	(Facoltativo) Tipo di base da usare per il conteggio dei giorni. Se basis viene omesso, si presuppone il valore 0. I valori accettati sono elencati dopo questa tabella.

Il parametro **basis** accetta i valori seguenti:

Basis	Base per conteggio dei giorni
0 o omesso	US (NASD) 30/360
1	Effettivo/effettivo
2	Effettivo/360
3	Effettivo/365
4	Europeo 30/360

# Valore restituito

Rendimento annuale.

## Osservazioni:

- Le date vengono archiviate come numeri di serie sequenziali in modo da consentirne l'uso nei calcoli. In DAX il 30 dicembre 1899 è il giorno 0, mentre il 1° gennaio 2008 è il giorno 39448 perché cade 39.448 giorni dopo il 30 dicembre 1899.
- La data di regolamento è la data in cui un acquirente acquista una cedola, ad esempio un'obbligazione. La data di scadenza è la data in cui scade una cedola. Si supponga, ad esempio, che un'obbligazione trentennale venga emessa il 1° gennaio 2008 e che venga acquistata da un acquirente sei mesi dopo. La data di emissione sarà il 1° gennaio 2008, la data di regolamento sarà il 1° luglio 2008 e la data di scadenza sarà il 1° gennaio 2038, ovvero 30 anni dopo il 1° gennaio 2008, la data di emissione.
- I parametri settlement, maturity e issue vengono troncati a numeri interi.
- Il parametro basis viene arrotondato all'intero più vicino.
- Viene restituito un errore nei casi seguenti:
  - Il parametro settlement, maturity o issue non rappresenta una data valida.
  - La condizione di data maturity > settlement > issue non è soddisfatta.
  - rate < 0.
  - pr ≤ 0.
  - base < 0 o base > 4.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Dati	Descrizione
15-mar-08	Data di regolamento
3-nov-08	Data di scadenza
8-nov-07	Data di emissione

Dati	Descrizione
6,25%	Percentuale della cedola semestrale
100,0123	Prezzo
0	Base 30/360 (vedere sopra)

La query DAX seguente:

DAX

```
EVALUATE
{
    YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625,
    100.0123, 0)
}
```

Restituisce il rendimento di un titolo in base ai termini specificati in precedenza.

[Valore]

0,0609543336915387

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni informative

Articolo • 20/10/2023

Le funzioni informative DAX esaminano la cella o la riga specificata come argomento e indicano se il valore corrisponde al tipo previsto. La funzione ISERROR, ad esempio, restituisce TRUE se il valore a cui si fa riferimento contiene un errore.

## In questa categoria

Funzione	Descrizione
COLUMNSTATISTICS	Restituisce una tabella di statistiche relative a ogni colonna di ogni tabella del modello.
CONTAINS	Restituisce True se i valori per tutte le colonne a cui si fa riferimento esistono, o sono contenuti, nelle colonne; in caso contrario, restituisce False.
CONTAINSROW	Restituisce TRUE se una riga di valori esiste o è contenuta in una tabella; in caso contrario, restituisce FALSE.
CONTAINSSTRING	Restituisce TRUE o FALSE per indicare se una stringa contiene un'altra stringa.
CONTAINSSTRINGEXACT	Restituisce TRUE o FALSE per indicare se una stringa contiene un'altra stringa.
CUSTOMDATA	Restituisce il contenuto della proprietà CustomData nella stringa di connessione.
HASONEFILTER	Restituisce TRUE se il numero di valori filtrati direttamente in <i>columnName</i> è uno; in caso contrario, restituisce FALSE.
HASONEVALUE	Restituisce TRUE se il contesto per <i>columnName</i> è stato filtrato in base a un solo valore distinto. In caso contrario, restituisce FALSE.
ISAFTER	Funzione booleana che emula il comportamento di una clausola Start At e restituisce true per una riga che soddisfa tutti i parametri della condizione.
ISBLANK	Controlla se un valore è vuoto e restituisce TRUE o FALSE.
ISCROSSFILTERED	Restituisce TRUE se <i>columnName</i> o un'altra colonna nella stessa tabella o in una tabella correlata viene filtrata.
ISEMPTY	Verifica se una tabella è vuota.

Funzione	Descrizione
<b>ISERROR</b>	Controlla se un valore è un errore e restituisce TRUE o FALSE.
<b>ISEVEN</b>	Restituisce TRUE se il numero è pari, FALSE se il numero è dispari.
<b>ISFILTERED</b>	Restituisce TRUE quando <i>columnName</i> viene filtrata direttamente.
<b>ISINSCOPE</b>	Restituisce true quando la colonna specificata rappresenta il livello in una gerarchia di livelli.
<b>ISLOGICAL</b>	Controlla se un valore è un valore logico (TRUE o FALSE) e restituisce TRUE o FALSE.
<b>ISNOTEXT</b>	Controlla se un valore non è costituito da testo (le celle vuote non sono testo) e restituisce TRUE o FALSE.
<b>ISNUMBER</b>	Controlla se un valore è un numero e restituisce TRUE o FALSE.
<b>ISODD</b>	Restituisce TRUE se il numero è dispari, FALSE se il numero è pari.
<b>ISONORAFTER</b>	Funzione booleana che emula il comportamento di una clausola Start At e restituisce true per una riga che soddisfa tutti i parametri della condizione.
<b>ISSELECTEDMEASURE</b>	Usata dalle espressioni per gli elementi di calcolo per determinare se la misura nel contesto è una di quelle specificate in un elenco di misure.
<b>ISSUBTOTAL</b>	Crea un'altra colonna in un'espressione SUMMARIZE, che restituisce True se la riga contiene i valori dei subtotali per la colonna specificata come argomento. In caso contrario, restituisce False.
<b>ISTEXT</b>	Controlla se un valore è di tipo testo e restituisce TRUE o FALSE.
<b>NONVISUAL</b>	Contrassegna un filtro di valore in un'espressione SUMMARIZECOLUMNS come non visivo.
<b>SELECTEDMEASURE</b>	Usata dalle espressioni per gli elementi di calcolo per fare riferimento alla misura nel contesto.
<b>SELECTEDMEASUREFORMATSTRING</b>	Usata dalle espressioni per gli elementi di calcolo per recuperare la stringa di formato della misura nel contesto.

Funzione	Descrizione
SELECTEDMEASURENAME	Usata dalle espressioni per gli elementi di calcolo per determinare la misura nel contesto in base al nome.
USERCULTURE	Restituisce le impostazioni locali per l'utente corrente.
USERNAME	Restituisce il nome di dominio e il nome utente in base alle credenziali fornite al sistema al momento della connessione.
USEROBJECTID	Restituisce l'ID oggetto o il SID dell'utente corrente.
USERPRINCIPALNAME	Restituisce il nome dell'entità utente.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COLUMNSTATISTICS

Articolo • 20/10/2023

Restituisce una tabella di statistiche relative a ogni colonna di ogni tabella del modello.

## Sintassi

DAX

```
COLUMNSTATISTICS ()
```

## Parametri

Questa funzione non accetta parametri.

## Valore restituito

Tabella delle statistiche. Ogni riga di questa tabella rappresenta una colonna diversa nel modello. Le colonne di tabella includono:

- **Nome tabella:** tabella della colonna corrente.
- **Nome colonna:** nome della colonna corrente.
- **Min:** valore minimo trovato all'interno della colonna corrente.
- **Max:** valore massimo trovato all'interno della colonna corrente.
- **Cardinalità:** numero di valori distinti trovati all'interno della colonna corrente.
- **Lunghezza massima:** lunghezza della stringa più lunga trovata all'interno della colonna corrente (applicabile solo per le colonne stringa).

## Osservazioni:

- Le colonne con stato di errore e colonne delle tabelle calcolate con ambito query non vengono visualizzate nella tabella dei risultati.
- Se viene applicato un filtro dal contesto di filtro a COLUMNSTATISTICS(), viene restituito un errore.

## Esempio

Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).

La query DAX seguente:

```
DAX

DEFINE
    TABLE FilteredProduct =
        FILTER (
            Product,
            [Color] == "Blue"
        )
    COLUMN Customer[Location] = [State-Province] & " " & [Country-Region]

EVALUATE
COLUMNSTATISTICS ()
```

Restituisce una tabella con statistiche relative a tutte le colonne di tutte le tabelle del modello. La tabella include anche le statistiche per la colonna calcolata con ambito di query, Customer[Location]. Tuttavia, la tabella non include le colonne della tabella calcolata con ambito di query, FilteredProduct.

Table Name	Column Name	Min	Max	Cardinality	Max Length
Customer	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			18485	
Customer	CustomerKey	-1	29483	18485	
Customer	Customer ID	[Not Applicable]	AW00029483	18485	16
Customer	Customer	[Not Applicable]	Zoe Watson	18401	26
Customer	City	[Not Applicable]	York	270	21
Customer	State-Province	[Not Applicable]	Yveline	54	19
Customer	Country-Region	[Not Applicable]	United States	7	16
Customer	Postal Code	[Not Applicable]	YO15	324	16
Customer	Location	[Not Applicable] [Not Applicable]	Yveline France	54	33
Date	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			1461	
Date	DateKey	20170701	20210630	1461	
Date	Date	7/1/2017 12:00:00 AM	6/30/2021 12:00:00 AM	1461	
Date	Fiscal Year	FY2018	FY2021	4	6
Date	Fiscal Quarter	FY2018 Q1	FY2021 Q4	16	9
Date	Month	2017 Aug	2021 May	48	8
Date	MonthKey	201707	202106	48	
Date	Full Date	2017 Aug, 01	2021 May, 31	1461	12
Sales Territory	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			11	
Sales Territory	SalesTerritoryKey	1	11	11	
Sales Territory	Region	Australia	United Kingdom	11	

## Vedi anche

[Contesto di filtro](#)

[Funzione CALCULATETABLE](#)

# Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# CONTAINS

Articolo • 20/10/2023

Restituisce True se i valori per tutte le colonne a cui si fa riferimento esistono, o sono contenuti, nelle colonne; in caso contrario, restituisce False.

## Sintassi

DAX

```
CONTAINS(<table>, <columnName>, <value>[ , <columnName>, <value>]...)
```

## Parametri

Termine	Definizione
table	qualsiasi espressione DAX che restituisce una tabella di dati.
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.
value	qualsiasi espressione DAX che restituisce un singolo valore scalare, che deve essere cercato in <i>columnName</i> . L'espressione deve essere valutata esattamente una volta e prima di essere passata all'elenco di argomenti.

## Valore restituito

Valore **TRUE** se ogni argomento *value* specificato può essere trovato, o è contenuto, nel *columnName* corrispondente. In caso contrario, la funzione restituisce **FALSE**.

## Osservazioni:

- Gli argomenti *columnName* e *value* devono essere in coppia; in caso contrario, viene restituito un errore.
- *columnName* deve appartenere alla *tabella* specificata o a una tabella correlata a *table*.
- Se *columnName* fa riferimento a una colonna in una tabella correlata, è necessario usare il nome completo. In caso contrario, viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente crea una misura che indica se sono state realizzate vendite Internet del prodotto 214 al cliente 11185.

DAX

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzione CONTAINSROW

Articolo • 20/10/2023

Restituisce VERO se esiste almeno una riga in cui in tutte le colonne hanno valori specificati.

## Sintassi

DAX

```
CONTAINSROW(<Table>, <Value> [, <Value> [, ...] ] )
```

## Parametri

Termine	Definizione
Tabella	Una tabella da testare.
Valore	qualsiasi espressione DAX valida che restituisce un valore scalare.

## Valore restituito

TRUE o FALSE.

## Osservazioni:

- Tranne che per la sintassi, l'operatore IN e la funzione CONTAINSROW sono equivalenti dal punto di vista funzionale.

DAX

```
<scalarExpr> IN <tableExpr>
( <scalarExpr1>, <scalarExpr2>, ... ) IN <tableExpr>
```

- Il numero di scalarExprN deve corrispondere al numero di colonne in tableExpr.
- NOT IN non è un operatore in DAX. Per eseguire la negazione logica dell'operatore IN, posizionare NOT davanti all'intera espressione. Ad esempio, NOT [Color] IN { "Red", "Yellow", "Blue" }.

- A differenza dell'operatore `=`, l'operatore `IN` e la funzione `CONTAINSROW` eseguono un confronto rigoroso. Ad esempio, il valore `BLANK` non corrisponde a `0`.

## Esempi

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

### Esempio 1

Le query DAX seguenti:

```
DAX

EVALUATE
FILTER (
    ALL ( Product[Color] ),
    ( [Color] )
        IN {
            "Red",
            "Yellow",
            "Blue"
        }
)
ORDER BY [Color]
```

e

```
DAX

EVALUATE
FILTER (
    ALL ( Product[Color] ),
    CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
        [Color]
    )
)
ORDER BY [Color]
```

Restituiscono la tabella seguente con una singola colonna:

### [Colore]

Blu

Rosso

Giallo

## Esempio 2

Le query DAX equivalenti seguenti:

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    NOT [Color]
        IN {
            "Red",
            "Yellow",
            "Blue"
        }
)
ORDER BY [Color]
```

e

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    NOT CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
        [Color]
    )
)
ORDER BY [Color]
```

Restituiscono la tabella seguente con una singola colonna:

### [Colore]

Nero

## [Colore]

Multi

N/D

Medio

Argento\Nero

Bianco

## Vedi anche

IN

[Query DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CONTAINSSTRING

Articolo • 20/10/2023

Restituisce TRUE o FALSE per indicare se una stringa contiene un'altra stringa.

## Sintassi

DAX

```
CONTAINSSTRING(<within_text>, <find_text>)
```

## Parametri

Termine	Definizione
within_text	Testo in cui cercare find_text.
find_text	Testo da trovare.

## Valore restituito

TRUE se find\_text è una sottostringa di within\_text; in caso contrario, FALSE.

## Osservazioni:

- CONTAINSSTRING non fa distinzione tra maiuscole e minuscole.
- È possibile usare i caratteri jolly ? e \*. Usare ~ come carattere di escape per i caratteri jolly.

## Esempio

Query DAX

DAX

```
EVALUATE  
ROW(  
    "Case 1", CONTAINSSTRING("abcd", "bc"),  
    "Case 2", CONTAINSSTRING("abcd", "BC"),
```

```
        "Case 3", CONTAINSSTRING("abcd", "a*d"),
        "Case 4", CONTAINSSTRING("abcd", "ef")
    )
```

Resi

[Case 1]	[Case 2]	[Case 3]	[Case 4]
TRUE	TRUE	TRUE	FALSE

## Commenti e suggerimenti

Questa pagina è stata utile?



# CONTAINSSTRINGEXACT

Articolo • 20/10/2023

Restituisce TRUE o FALSE per indicare se una stringa contiene un'altra stringa.

## Sintassi

DAX

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

## Parametri

Termine	Definizione
within_text	Testo in cui cercare find_text.
find_text	Testo da trovare.

## Valore restituito

TRUE se find\_text è una sottostringa di within\_text; in caso contrario, FALSE.

## Osservazioni:

CONTAINSSTRINGEXACT fa distinzione tra maiuscole e minuscole.

## Esempio

Query DAX

DAX

```
EVALUATE  
ROW(  
    "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),  
    "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),  
    "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),  
    "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
```

)

Resi

[Case 1]	[Case 2]	[Case 3]	[Case 4]
TRUE	FALSE	FALSO	FALSE

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CUSTOMDATA

Articolo • 20/10/2023

Restituisce il contenuto della proprietà **CustomData** nella stringa di connessione.

## Sintassi

DAX

```
CUSTOMDATA()
```

## Valore restituito

Contenuto della proprietà **CustomData** nella stringa di connessione.

Vuoto, se la proprietà **CustomData** non è stata definita al momento della connessione.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula DAX seguente verifica se la proprietà CustomData è stata impostata su "OK".

DAX

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No  
custom data in connection string property or unexpected value")
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# HASONEFILTER

Articolo • 20/10/2023

Restituisce **TRUE** se il numero di valori filtrati direttamente in *columnName* è uno; in caso contrario, restituisce **FALSE**.

## Sintassi

DAX

```
HASONEFILTER(<columnName>)
```

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.

## Valore restituito

**TRUE** se il numero di valori filtrati direttamente in *columnName* è uno; in caso contrario, **FALSE**.

## Osservazioni:

- Questa funzione è simile a HASONEVALUE() con la differenza che HASONEVALUE() opera con filtri incrociati mentre HASONEFILTER() opera con un filtro diretto.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente mostra come usare HASONEFILTER() per restituire il filtro per ResellerSales\_USD[ProductKey]) in presenza di un solo filtro o per restituire BLANK in presenza di nessuno o più filtri su ResellerSales\_USD[ProductKey]).

DAX

```
=  
IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[Pro  
ductKey]),BLANK())
```

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# HASONEVALUE

Articolo • 20/10/2023

Restituisce **TRUE** se il contesto per *columnName* è stato filtrato in base a un solo valore distinto. In caso contrario, restituisce **FALSE**.

## Sintassi

HTML

```
HASONEVALUE(<columnName>)
```

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.

## Valore restituito

Restituisce **TRUE** se il contesto per *columnName* è stato filtrato in base a un solo valore distinto. In caso contrario, restituisce **FALSE**.

## Osservazioni:

- Un'espressione equivalente di HASONEVALUE() è `COUNTROWS(VALUES(<columnName>)) = 1`.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di misura seguente verifica se il contesto viene sezionato in base a un valore per stimare una percentuale rispetto a uno scenario predefinito. In questo caso si vogliono confrontare i risultati di Reseller Sales con le vendite del 2007, quindi occorre

sapere se il contesto è filtrato in base a singoli anni. Inoltre, se il confronto non è significativo, deve essere restituito un valore BLANK.

DAX

```
=  
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])  
]/CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=  
2007),BLANK())
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISAFTER

Articolo • 20/10/2023

Funzione booleana che emula il comportamento di una clausola 'Start At' e restituisce true per una riga che soddisfa tutti i parametri della condizione.

In base all'ordinamento, il primo parametro viene confrontato con il secondo parametro. Se l'ordinamento è crescente, il confronto da eseguire è il primo parametro maggiore del secondo parametro. Se l'ordinamento è decrescente, il confronto da eseguire è il secondo parametro minore del primo parametro.

## Sintassi

DAX

```
ISAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

## Parametri

Termine	Definizione
scalar_expression	Qualsiasi espressione che restituisce un valore scalare come un riferimento di colonna o un valore integer o stringa. In genere il primo parametro è un riferimento di colonna e il secondo parametro è un valore scalare.
ordinamento	(Facoltativo) Ordinamento della colonna. Può essere crescente (ASC) o decrescente (DEC). Per impostazione predefinita, l'ordinamento è crescente.

## Valore restituito

True o false.

## Osservazioni:

Questa funzione è simile a [ISONORAFTER](#). La differenza è che ISAFTER restituisce vero per i valori ordinati rigorosamente *dopo* i valori del filtro, mentre ISONORAFTER restituisce vero per i valori ordinati *su* o *dopo* i valori del filtro.

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nome tabella: 'Info'

Paese/area geografica	Provincia	Count	Totali
IND	5K	20	800
IND	MH	25	1000
IND	WB	10	900
USA	CA	5	500
USA	WA	10	900

L'espressione seguente:

```
DAX  
  
FILTER (  
    Info,  
    ISAFTER (  
        Info[Country], "IND", ASC,  
        Info[State], "MH", ASC )  
)
```

Restituisce:

Country	Provincia	Count	Totali
IND	WB	10	900
USA	CA	5	500
USA	WA	10	900

## Vedi anche

[ISONORAFTER](#)

# Commenti e suggerimenti

Questa pagina è stata utile?



# ISBLANK

Articolo • 20/10/2023

Controlla se un valore è vuoto e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISBLANK(<value>)
```

## Parametri

Termine	Definizione
value	Valore o espressione da testare.

## Valore restituito

Valore booleano TRUE se il valore è vuoto; in caso contrario, FALSE.

## Osservazioni:

Per ulteriori informazioni sulle procedure consigliate quando si lavora con BLANKS, vedere [Evitare la conversione di BLANKS in valori in DAX](#).

## Esempio

Questa formula calcola la percentuale di aumento o diminuzione delle vendite rispetto all'anno precedente. Nell'esempio viene usata la funzione IF per controllare il valore delle vendite dell'anno precedente in modo da evitare un errore di divisione per zero.

DAX

```
//Sales to Previous Year Ratio  
  
= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])  
    , BLANK()  
    , ( 'CalculatedMeasures'[Total Sales]-
```

```
'CalculatedMeasures'[PreviousYearTotalSales] )  
/ 'CalculatedMeasures'[PreviousYearTotalSales])
```

Risultato:

<b>Etichette di riga</b>	<b>Totale vendite</b>	<b>Total Sales Previous Year</b>	<b>Sales to Previous Year Ratio</b>
2005	\$10,209,985.08		
2006	\$28,553,348.43	\$10,209,985.08	179.66%
2007	\$39,248,847.52	\$28,553,348.43	37.46%
2008	\$24,542,444.68	\$39,248,847.52	-37.47%
Grand Total	\$102,554,625.71		

## Vedi anche

[Funzioni informative](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISCROSSFILTERED

Articolo • 20/10/2023

Restituisce TRUE quando alla tabella o colonna specificata sono applicati filtri incrociati.

## Sintassi

DAX

```
ISCROSSFILTERED(<TableNameOrColumnName>)
```

## Parametri

Termine	Definizione
TableNameOrColumnName	Nome di una tabella o di una colonna esistente. Non può essere un'espressione.

## Valore restituito

TRUE quando *ColumnName* o una colonna di *TableName* viene filtrata in modo incrociato. In caso contrario restituisce FALSE.

## Osservazioni:

- Una colonna o una tabella viene detta filtrata in modo incrociato quando un filtro viene applicato a *ColumnName*, a qualsiasi colonna di *TableName* o a qualsiasi colonna di una tabella correlata.
- Una colonna o una tabella viene detta filtrata direttamente quando un filtro viene applicato a *ColumnName* o a qualsiasi colonna di *TableName*. Pertanto, la funzione **ISFILTERED** restituisce TRUE anche quando *ColumnName* o qualsiasi colonna di *TableName* viene filtrata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[Funzione ISFILTERED](#)

[Funzione FILTERS](#)

[Funzione HASONEFILTER](#)

[Funzione HASONEVALUE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISEMPTY

Articolo • 20/10/2023

Verifica se una tabella è vuota.

## Sintassi

DAX

```
ISEMPTY(<table_expression>)
```

## Parametri

Termine	Definizione
table_expression	Riferimento a una tabella o espressione DAX che restituisce una tabella.

## Valore restituito

True se la tabella è vuota (non contiene righe); in caso contrario, false.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Per la tabella seguente denominata 'Info':

Paese/area geografica	Provincia	Contea	Totali
IND	5K	20	800
IND	MH	25	1000
IND	WB	10	900
USA	CA	5	500

Paese/area geografica	Provincia	Contea	Totali
USA	WA	10	900

DAX

```
EVALUATE  
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info,  
[County]>25))))
```

Valore restituito: FALSE

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISERROR

Articolo • 20/10/2023

Controlla se un valore è un errore e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISERROR(<value>)
```

## Parametri

Termine	Definizione
value	Valore da testare.

## Valore restituito

Valore booleano TRUE se il valore è un errore; in caso contrario, FALSE.

## Osservazioni:

- Per le procedure consigliate quando si usa ISERROR, vedere [Uso appropriato delle funzioni di errore](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene calcolato il rapporto delle vendite Internet totali per ottenere la somma delle vendite dei rivenditori. La funzione ISERROR viene usata per verificare se sono presenti errori, ad esempio una divisione per zero. Se è presente un errore, viene restituito un valore vuoto. In caso contrario, viene restituito il rapporto.

DAX

```
= IF( ISERROR(
    SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
    )
, BLANK()
, SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
)
```

## Vedi anche

[Funzioni informative](#)

[Funzione IFERROR](#)

[Funzione IF](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISEVEN

Articolo • 20/10/2023

Restituisce TRUE se il numero è pari, FALSE se il numero è dispari.

## Sintassi

DAX

`ISEVEN(number)`

## Parametri

Termine	Definizione
number	Valore su cui eseguire il test. Se number non è un numero intero, la parte decimale viene troncata.

## Valore restituito

Restituisce TRUE se il numero è pari, FALSE se il numero è dispari.

## Osservazioni:

- Se number non è un valore numerico, ISEVEN restituisce il valore di errore #VALUE!.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# ISFILTERED

Articolo • 20/10/2023

Restituisce TRUE quando la tabella o la colonna specificata viene filtrata direttamente.

## Sintassi

DAX

```
ISFILTERED(<TableNameOrColumnName>)
```

## Parametri

Termine	Definizione
TableNameOrColumnName	Nome di una tabella o di una colonna esistente. Non può essere un'espressione.

## Valore restituito

TRUE quando *ColumnName* o una colonna di *TableName* viene filtrata direttamente. In caso contrario restituisce FALSE.

## Osservazioni:

- Una colonna o una tabella viene detta filtrata direttamente quando un filtro viene applicato a *ColumnName* o a qualsiasi colonna di *TableName*.
- Una colonna o una tabella viene detta filtro incrociato quando un filtro viene applicato a *ColumnName*, a qualsiasi colonna di *TableName* o a qualsiasi colonna di una tabella correlata. Pertanto, la funzione [ISCROSSFILTERED](#) restituisce TRUE anche quando *ColumnName*, qualsiasi colonna di *TableName* o una colonna di una tabella correlata viene filtrata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[Funzione ISCROSSFILTERED](#)

[Funzione FILTERS](#)

[Funzione HASONEFILTER](#)

[Funzione HASONEVALUE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISINSCOPE

Articolo • 20/10/2023

Restituisce true quando la colonna specificata rappresenta il livello in una gerarchia di livelli.

## Sintassi

DAX
ISINSCOPE(<columnName>)

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.

## Valore restituito

TRUE quando la colonna specificata rappresenta il livello in una gerarchia di livelli.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX
DEFINE MEASURE FactInternetSales[% of Parent] = SWITCH (TRUE(), ISINSCOPE(DimProduct[Subcategory]), DIVIDE( SUM(FactInternetSales[Sales Amount]), CALCULATE( SUM(FactInternetSales[Sales Amount]), ALLSELECTED(DimProduct[Subcategory])) ), ISINSCOPE(DimProduct[Category]), DIVIDE(

```

        SUM(FactInternetSales[Sales Amount]),
CALCULATE(
    SUM(FactInternetSales[Sales Amount]),
    ALLSELECTED(DimProduct[Category]))
),
1
) * 100
EVALUATE
SUMMARIZECOLUMNS
(
    ROLLUPADDISSUBTOTAL
(
    DimProduct[Category], "Category Subtotal",
    DimProduct[Subcategory], "Subcategory Subtotal"
),
TREATAS(
    {"Bike Racks", "Bike Stands", "Mountain Bikes", "Road Bikes", "Touring
Bikes"},
    DimProduct[Subcategory]),
    "Sales", SUM(FactInternetSales[Sales Amount]),
    "% of Parent", [% of Parent]
)
ORDER BY
    [Category Subtotal] DESC, [Category],
    [Subcategory Subtotal] DESC, [Subcategory]

```

Restituisce:

DimProduct[Category]	DimProduct[SubCategory]	[Category Subtotal]	[Subcategory Subtotal]	[Sales]	[% of Parent]
		TRUE	TRUE	28,397,095.65	100.00
Accessori		FALSE	TRUE	78,951.00	0,28
Accessori	Bike Racks	FALSE	FALSE	39,360.00	49.85
Accessori	Bike Stands	FALSE	FALSE	39,591.00	50,15
Bikes		FALSE	TRUE	28,318,144.65	99.72
Bikes	Mountain Bikes	FALSE	FALSE	9,952,759.56	35.15
Bikes	Road Bikes	FALSE	FALSE	14,520,584.04	51.28
Bikes	Touring Bikes	FALSE	FALSE	3,844,801.05	13.58

## Vedi anche

[Funzione SUMMARIZECOLUMNS](#)

[Funzione CALCULATE](#)

# Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ISLOGICAL

Articolo • 20/10/2023

Controlla se un valore è un valore logico (TRUE o FALSE) e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISLOGICAL(<value>)
```

## Parametri

Termine	Definizione
value	Valore da testare.

## Valore restituito

TRUE se si tratta di un valore logico. FALSE se si tratta di qualunque valore diverso da TRUE o FALSE.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

I tre esempi seguenti mostrano il comportamento di ISLOGICAL.

DAX

```
//RETURNS: Is Boolean type or Logical  
= IF(ISLOGICAL(true), "Is Boolean type or Logical", "Is different type")  
  
//RETURNS: Is Boolean type or Logical  
= IF(ISLOGICAL(false), "Is Boolean type or Logical", "Is different type")
```

```
//RETURNS: Is different type  
= IF(ISLOGICAL(25), "Is Boolean type or Logical", "Is different type")
```

## Vedi anche

[Funzioni informative](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISNOTEXT

Articolo • 20/10/2023

Controlla se un valore non è costituito da testo (le celle vuote non sono testo) e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISNOTTEXT(<value>)
```

## Parametri

Termine	Definizione
value	Valore da controllare.

## Valore restituito

TRUE se il valore è vuoto o non di tipo testo, FALSE se il valore è di tipo testo.

## Osservazioni:

- Una stringa vuota viene considerata come testo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Gli esempi seguenti illustrano il comportamento della funzione ISNOTTEXT.

DAX

```
//RETURNS: Is Non-Text  
= IF(ISNOTTEXT(1), "Is Non-Text", "Is Text")  
  
//RETURNS: Is Non-Text  
= IF(ISNOTTEXT(BLANK()), "Is Non-Text", "Is Text")
```

```
//RETURNS: Is Text  
= IF(ISNONTEXT(""), "Is Non-Text", "Is Text")
```

## Vedi anche

[Funzioni informative](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISNUMBER

Articolo • 20/10/2023

Controlla se un valore è un numero e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISNUMBER(<value>)
```

## Parametri

Termine	Definizione
value	Valore da testare.

## Valore restituito

TRUE se il valore è numerico; in caso contrario FALSE.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

I tre esempi seguenti mostrano il comportamento di ISNUMBER.

DAX

```
//RETURNS: Is number  
= IF(ISNUMBER(0), "Is number", "Is Not number")  
  
//RETURNS: Is number  
= IF(ISNUMBER(3.1E-1),"Is number", "Is Not number")  
  
//RETURNS: Is Not number  
= IF(ISNUMBER("123"), "Is number", "Is Not number")
```

## Vedi anche

[Funzioni informative](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISODD

Articolo • 20/10/2023

Restituisce TRUE se il numero è dispari, FALSE se il numero è pari.

## Sintassi

DAX

`ISODD(number)`

## Parametri

Termine	Definizione
number	Valore su cui eseguire il test. Se number non è un numero intero, la parte decimale viene troncata.

## Valore restituito

Restituisce TRUE se il numero è dispari, FALSE se il numero è pari.

## Osservazioni:

- Se number non è un valore numerico, ISODD restituisce il valore di errore #VALUE! .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**    **No**

# ISONORAFTER

Articolo • 20/10/2023

Funzione booleana che emula il comportamento di una clausola Start At e restituisce true per una riga che soddisfa tutti i parametri della condizione.

In base all'ordinamento, il primo parametro viene confrontato con il secondo parametro. Se l'ordinamento è crescente, il confronto da eseguire è il primo parametro maggiore del secondo parametro. Se l'ordinamento è decrescente, il confronto da eseguire è il secondo parametro minore del primo parametro.

## Sintassi

DAX

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

## Parametri

Termine	Definizione
scalar_expression	Qualsiasi espressione che restituisce un valore scalare come un riferimento di colonna o un valore integer o stringa. In genere il primo parametro è un riferimento di colonna e il secondo parametro è un valore scalare.
ordinamento	(Facoltativo) Ordinamento della colonna. Può essere crescente (ASC) o decrescente (DEC). Per impostazione predefinita, l'ordinamento è crescente.

## Valore restituito

True o false.

## Osservazioni:

Questa funzione è simile a [ISAFTER](#). La differenza è che ISONORAFTER restituisce vero per i valori ordinati *su o dopo* i valori del filtro, mentre ISAFTER restituisce vero per i valori ordinati rigorosamente *dopo* i valori del filtro.

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Per la tabella seguente denominata, Info:

Paese/area geografica	Provincia	Count	Totali
IND	5K	20	800
IND	MH	25	1000
IND	WB	10	900
USA	CA	5	500
USA	WA	10	900

L'espressione seguente:

DAX
<pre>FILTER (     Info,     ISONORAFTER (         Info[Country], "IND", ASC,         Info[State], "MH", ASC ) )</pre>

Restituisce:

Paese/area geografica	Provincia	Count	Totali
IND	MH	25	1000
IND	WB	10	900
USA	CA	5	500
USA	WA	10	900

## Vedi anche

[ISAFTER](#)

# Commenti e suggerimenti

Questa pagina è stata utile?



# ISSELECTEDMEASURE

Articolo • 20/10/2023

Usata dalle espressioni per gli elementi di calcolo per determinare se la misura nel contesto è una di quelle specificate in un elenco di misure.

## Sintassi

DAX

```
ISSELECTEDMEASURE( M1, M2, ... )
```

## Parametri

Termine	Definizione
M1, M2, ...	Elenco di misure.

## Valore restituito

Valore booleano che indica se la misura attualmente nel contesto corrisponde a una di quelle specificate nell'elenco di parametri.

## Osservazioni:

- È possibile farvi riferimento nell'espressione solo per un elemento di calcolo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'espressione di elemento di calcolo seguente controlla se la misura corrente è una di quelle specificate nell'elenco di parametri. Se le misure vengono rinominate, la correzione della formula rifletterà le modifiche al nome nell'espressione.

DAX

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

## Vedi anche

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISSUBTOTAL

Articolo • 20/10/2023

Crea un'altra colonna in un'espressione [SUMMARIZE](#), che restituisce True se la riga contiene i valori dei subtotali per la colonna specificata come argomento. In caso contrario, restituisce False.

## Sintassi

DAX

```
ISSUBTOTAL(<columnName>)
```

Con [SUMMARIZE](#)

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[,  
ROLLUP(<groupBy_columnName>[,< groupBy_columnName>...])][, <name>,  
{<expression>|ISSUBTOTAL(<columnName>)}]...)
```

## Parametri

Termine	Definizione
columnName	Nome di qualsiasi colonna nella tabella della funzione SUMMARIZE o di qualsiasi colonna in una tabella correlata alla tabella.

## Valore restituito

Un valore True se la riga contiene un valore di totale per la colonna specificata come argomento. In caso contrario, restituisce False.

## Osservazioni:

- Questa funzione può essere usata solo nella parte dell'espressione di una funzione [SUMMARIZE](#).
- Questa funzione deve essere preceduta dal nome della colonna booleana.

# Esempio

Vedere [SUMMARIZE](#).

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISTEXT

Articolo • 20/10/2023

Controlla se un valore è di tipo testo e restituisce TRUE o FALSE.

## Sintassi

DAX

```
ISTEXT(<value>)
```

## Parametri

Termine	Definizione
value	Valore da controllare.

## Valore restituito

TRUE se il valore è di tipo testo. In caso contrario FALSE.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Gli esempi seguenti illustrano il comportamento della funzione ISTEEXT.

DAX

```
//RETURNS: Is Text  
= IF(ISTEXT("text"), "Is Text", "Is Non-Text")  
  
//RETURNS: Is Text  
= IF(ISTEXT(""), "Is Text", "Is Non-Text")  
  
//RETURNS: Is Non-Text  
= IF(ISTEXT(1), "Is Text", "Is Non-Text")
```

```
//RETURNS: Is Non-Text  
= IF(ISTEXT(BLANK()), "Is Text", "Is Non-Text")
```

## Vedi anche

[Funzioni informative](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NONVISUAL

Articolo • 20/10/2023

Contrassegna un filtro di valore in un'espressione [SUMMARIZECOLUMNS](#) come non visivo. Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZECOLUMNS](#).

## Sintassi

DAX
<code>NONVISUAL(&lt;expression&gt;)</code>

## Parametri

Termine	Definizione
expression	qualsiasi espressione DAX che restituisce un singolo valore (non una tabella).

## Valore restituito

Tabella di valori.

## Osservazioni:

- Indica che un filtro di valore in [SUMMARIZECOLUMNS](#) non influisce sui valori di misura, ma si applica solo alle colonne group-by.
- Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZECOLUMNS](#). Viene usata come argomento filterTable della funzione [SUMMARIZECOLUMNS](#) o come argomento groupLevelFilter della funzione [ROLLUPADDISSTOTAL](#) o [ROLLUPISSSTOTAL](#).

## Esempio

Vedere [SUMMARIZECOLUMNS](#).

# Commenti e suggerimenti

Questa pagina è stata utile?



# SELECTEDMEASURE

Articolo • 20/10/2023

Usato dalle espressioni per gli elementi di calcolo o le stringhe di formato dinamico per fare riferimento alla misura nel contesto.

## Sintassi

DAX

```
SELECTEDMEASURE()
```

## Parametri

None

## Valore restituito

Riferimento alla misura attualmente nel contesto quando viene valutata l'elemento di calcolo o la stringa di formato.

## Osservazioni:

- È possibile farvi riferimento solo nell'espressione per un elemento di calcolo o una stringa di formato.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'espressione di elemento di calcolo seguente calcola il valore da inizio anno qualunque sia la misura nel contesto.

DAX

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

L'espressione seguente può essere usata per regolare dinamicamente la stringa di formato di una misura in base al fatto che un valore sia costituito da centinaia, migliaia o milioni.

```
DAX

SWITCH(
    TRUE(),
    SELECTEDMEASURE() < 1000, "#,##0", //Values less than 1000 have
    no text after them
    SELECTEDMEASURE() < 1000000, "#,##0,.0 K", //Values between 1000 and
    1000000 are formatted as #.## K
    "#,##0,.0 M" //Values greater than 1000000
    are formatted as #.## M
)
```

## Vedi anche

[SELECTEDMEASURENAME](#)

[ISSELECTEDMEASURE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# SELECTEDMEASUREFORMATSTRING

Articolo • 20/10/2023

Usata dalle espressioni per gli elementi di calcolo per recuperare la stringa di formato della misura nel contesto.

## Sintassi

DAX

```
SELECTEDMEASUREFORMATSTRING()
```

## Parametri

None

## Valore restituito

Stringa che contiene la stringa di formato della misura attualmente nel contesto quando viene valutato l'elemento di calcolo.

## Osservazioni:

- È possibile fare riferimento a questa funzione solo nelle espressioni per gli elementi di calcolo nei gruppi di calcolo. È stata progettata per essere usata dalla proprietà **Format String Expression** degli elementi di calcolo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'espressione seguente viene valutata dalla proprietà Format String Expression per un elemento di calcolo. Se nel contesto di filtro è presente una singola valuta, la stringa di formato viene recuperata dalla colonna DimCurrency[FormatString]. In caso contrario, viene usata la stringa di formato della misura nel contesto.

DAX

```
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

## Vedi anche

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SELECTEDMEASURENAME

Articolo • 20/10/2023

Usata dalle espressioni per gli elementi di calcolo per determinare la misura nel contesto in base al nome.

## Sintassi

DAX

```
SELECTEDMEASURENAME()
```

## Parametri

None

## Valore restituito

Valore di stringa che contiene il nome della misura attualmente nel contesto quando viene valutato l'elemento di calcolo.

## Osservazioni:

- È possibile farvi riferimento nell'espressione solo per un elemento di calcolo.
- Questa funzione viene spesso usata a scopo di debug durante la creazione di gruppi di calcolo.

## Esempio

L'espressione di elemento di calcolo seguente controlla se la misura corrente è Expense Ratio e applica in modo condizionale la logica di calcolo. Poiché il controllo si basa su un confronto tra stringhe, non è soggetto alla correzione della formula e non trarrà vantaggio dalla ridefinizione degli oggetti riflessa automaticamente. Per un confronto analogo che trarrebbe vantaggio dalla correzione della formula, vedere invece la funzione ISSELECTEDMEASURE.

DAX

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

## Vedi anche

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# USERCULTURE

Articolo • 20/10/2023

Restituisce le impostazioni locali (codice paese di lingua) per l'utente corrente, determinate dal sistema operativo, dalle impostazioni del browser o dal servizio Power BI.

**Nota:** questa funzione è attualmente supportata in Power BI Premium per capacità, Power BI Premium per utente e solo Power BI Embedded.

## Sintassi

DAX

`USERCULTURE()`

## Parametri

Questa espressione non ha parametri.

## Valore restituito

Impostazioni locali come stringa.

## Osservazioni:

- Nel servizio Power BI le impostazioni locali sono determinate da **Impostazioni>lingua>Impostazioni lingua**. Il valore predefinito è determinato dall'impostazione della lingua del browser dell'utente.
- Se utilizzata nelle espressioni di tabella calcolata ed espressioni di colonna calcolate, il risultato può variare a seconda che la tabella sia in modalità DirectQuery o Import. Quando si usa la modalità DirectQuery, il risultato viene determinato dalla lingua (impostazioni locali) specificate in Impostazioni lingua nel servizio Power BI. Il valore predefinito in Impostazioni lingua specifica le impostazioni locali è determinato dall'impostazione della lingua del browser dell'utente, il che significa che la stessa tabella o colonna calcolata può restituire risultati diversi a seconda delle impostazioni della lingua del browser per ogni utente. Quando si usa la modalità importazione, il risultato viene determinato in

modo statico durante l'aggiornamento e non varia in fase di query. Per gli aggiornamenti gestiti, ad esempio pianificati o interattivi, le impostazioni locali non sono basate sull'impostazione della lingua del browser dell'utente, ma usano invece impostazioni locali invarianti. Le impostazioni locali invarianti, tuttavia, possono essere sottoposte a override tramite l'endpoint XMLA per specificare impostazioni locali personalizzate.

- In combinazione con la funzionalità Parametri campo in Power BI, è possibile usare `USERCULTURE` per tradurre in modo affidabile i titoli e le didascalie dinamiche di visualizzazione quando vengono usati nelle espressioni di misura e sicurezza a livello di riga (RLS) all'interno dello stesso modello. Tuttavia, le espressioni contenenti `USERCULTURE` chiamate dall'esterno del modello, ad esempio query e misure del report con connessione dinamica, non si deve fare affidamento su titoli e didascalie tradotti correttamente.
- `USERCULTURE` restituisce le impostazioni locali utente corrette quando vengono usate nelle espressioni di oggetto chiamate dall'interno del modello, ad esempio misure, sicurezza a livello di riga e elementi di calcolo. Tuttavia, potrebbe non restituire le impostazioni locali dell'utente corrette quando vengono usate nelle espressioni dall'esterno del modello, ad esempio query e misure del report con connessione dinamica.
- Nei report di Live Connect, `USERCULTURE` potrebbe non restituire le impostazioni locali utente corrette quando viene chiamato da un'espressione di misura del report.

## Esempio

Per l'espressione seguente,

DAX

```
FORMAT(TODAY(), "dddd", USERCULTURE())
```

A seconda dell'impostazione della lingua per l'utente corrente, `USERCULTURE` restituisce il giorno corrente, ad esempio:

impostazioni locali	Giorno feriale formattato
de-DE	Dienstag
en-US	Tuesday

impostazioni locali	Giorno feriale formattato
es-ES_tradnl	martes
eu-ES	asteartea
it-IT	martedì
nl-NL	dinsdag
pl-PL	wtorek
ro-RO	marți
ru-RU	вторник
uk-UA	вівторок

## Vedi anche

[titoli basati su espressioni in Power BI](#)

[USERNAME](#)

[USERPRINCIPALNAME](#)

[USEROBJECTID](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# USERNAME

Articolo • 20/10/2023

Restituisce il nome di dominio e il nome utente in base alle credenziali fornite al sistema al momento della connessione.

## Sintassi

DAX

```
USERNAME()
```

## Parametri

Questa espressione non ha parametri.

## Valore restituito

Nome utente in base alle credenziali fornite al sistema al momento della connessione

## Esempio

La formula seguente verifica se il nome di accesso dell'utente fa parte di UsersTable.

DAX

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# USEROBJECTID

Articolo • 26/10/2023

Restituisce l'ID oggetto dell'utente corrente in base a Microsoft Entra ID o all'identificatore di sicurezza (SID).

## Sintassi

DAX

```
USEROBJECTID()
```

## Parametri

Questa espressione non ha parametri.

## Valore restituito

ID oggetto dell'utente corrente ricavato da Microsoft Entra ID per i modelli Power BI o Azure Analysis Services o da SID per i modelli SQL Server Analysis Services.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# USERPRINCIPALNAME

Articolo • 20/10/2023

Restituisce il nome dell'entità utente.

## Sintassi

DAX

```
USERPRINCIPALNAME()
```

## Parametri

Questa espressione non ha parametri.

## Valore restituito

Il nome dell'entità utente in fase di connessione.

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# Funzioni logiche

Articolo • 20/10/2023

Le funzioni logiche eseguono operazioni su un'espressione per restituire informazioni sui valori o sui set nell'espressione. È ad esempio possibile usare la funzione IF per controllare il risultato di un'espressione e creare risultati condizionali.

## In questa categoria

Funzione	Descrizione
AND	Controlla se entrambi gli argomenti sono TRUE e restituisce TRUE se entrambi gli argomenti sono TRUE.
BITAND	Restituisce un "AND" bit per bit di due numeri.
BITLSHIFT	Restituisce un numero scorso a sinistra dal numero di bit specificato.
BITOR	Restituisce un "OR" bit per bit di due numeri.
BITRSHIFT	Restituisce un numero scorso a destra dal numero di bit specificato.
BITXOR	Restituisce un "XOR" bit per bit di due numeri.
COALESCE	Restituisce la prima espressione che non restituisce BLANK.
FALSE	Restituisce il valore logico FALSE.
IF	Controlla una condizione e restituisce un valore quando TRUE, in caso contrario restituisce un secondo valore.
IF.EAGER	Controlla una condizione e restituisce un valore quando TRUE, in caso contrario restituisce un secondo valore. Usa un piano di esecuzione <i>eager</i> che esegue sempre le espressioni di ramo indipendentemente dall'espressione della condizione.
IFERROR	Valuta un'espressione e restituisce un valore specificato se l'espressione restituisce un errore.
NOT	Cambia FALSE in TRUE o TRUE in FALSE.
OPPURE	Per restituire TRUE, verifica se uno degli argomenti è TRUE.
SWITCH	Valuta un'espressione sulla base di un elenco di valori e restituisce una tra più espressioni di risultato possibili.
TRUE	Restituisce il valore logico TRUE.

# Commenti e suggerimenti

Questa pagina è stata utile?



# E

Articolo • 20/10/2023

Controlla se entrambi gli argomenti sono TRUE e restituisce TRUE se entrambi gli argomenti sono TRUE. In caso contrario, restituisce false.

## Sintassi

DAX

```
AND(<logical1>,<logical2>)
```

## Parametri

Termine	Definizione
logical_1, logical_2	Valori logici da testare.

## Valore restituito

Restituisce True o False a seconda della combinazione di valori testati.

## Osservazioni:

La funzione AND in DAX accetta solo due (2) argomenti. Se è necessario eseguire un'operazione AND su più espressioni, è possibile creare una serie di calcoli, o meglio, usare l'operatore AND (**&&**) per unirle tutte in un'espressione più semplice.

## Esempio 1

La formula seguente illustra la sintassi della funzione AND.

DAX

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

Poiché entrambi le condizioni passate come argomenti alla funzione AND sono vere, la formula restituisce "All true".

## Esempio 2

Nell'esempio seguente viene usata la funzione AND con formule annidate per confrontare contemporaneamente due set di calcoli. Per ogni categoria di prodotto la formula determina se le vendite dell'anno corrente e quelle dell'anno precedente tramite il canale Internet sono maggiori rispetto al canale Reseller per gli stessi periodi. Se entrambe le condizioni sono true, per ogni categoria viene restituito il valore "Internet Hit".

DAX

```
= IF( AND( SUM('InternetSales_USD'[SalesAmount_USD])  
        >SUM('ResellerSales_USD'[SalesAmount_USD])  
        , CALCULATE(SUM('InternetSales_USD'[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey] ))  
        >CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey] ))  
        )  
        , "Internet Hit"  
        , ""  
    )
```

Resi

<b>Etichette di riga</b>	<b>2005</b>	<b>2006</b>	<b>2007</b>	<b>2008</b>	-	<b>Grand Total</b>
Bib-Shorts						
Bike Racks						
Bike Stands				Internet Hit		
Bottles and Cages				Internet Hit		
Bottom Brackets						
Brakes						
Cappelli						
Modelli a catena						
Cleaners						
Cranksets						
Derailleurs						
Fenders				Internet Hit		

<b>Etichette di riga</b>	<b>2005</b>	<b>2006</b>	<b>2007</b>	<b>2008</b>	<b>-</b>	<b>Grand Total</b>
Forks						
Gloves						
Manubrio						
Headsets						
Helmets						
Hydration Packs						
Jerseys						
Luci						
Locks						
Mountain Bikes						
Mountain Frames						
Panniers						
Pedals						
Pumps						
Road Bikes						
Road Frames						
Saddles						
Pantaloncini						
Calzini						
Tights						
Tires and Tubes				Internet Hit		
Touring Bikes						
Touring Frames						
Vests						
Ruote						
Grand Total						

## Vedi anche

[Funzioni logiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# BITAND

Articolo • 20/10/2023

Restituisce un AND bit per bit di due numeri.

## Sintassi

DAX

```
BITAND(<number>, <number>)
```

## Parametri

Termine	Definizione
Numero	qualsiasi espressione scalare che restituisce il numero. Se non è un numero intero, viene troncato.

## Valore restituito

Un AND bit per bit di due numeri.

## Osservazioni:

- Questa funzione supporta numeri sia positivi che negativi.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { BITAND(13, 11) }
```

Restituisce 9.

## Vedi anche

[BITLSHIFT](#)

[BITRSHIFT](#)

[BITOR](#)

[BITXOR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# BITLSHIFT

Articolo • 20/10/2023

Restituisce un numero scorso a sinistra dal numero di bit specificato.

## Sintassi

DAX

```
BITLSHIFT(<Number>, <Shift_Amount>)
```

## Parametri

Termine	Definizione
Numero	qualsiasi espressione DAX che restituisce un numero intero.
Shift_Amount	qualsiasi espressione DAX che restituisce un numero intero.

## Valore restituito

Valore intero.

## Osservazioni:

- Assicurarsi di comprendere la natura delle operazioni di bitshift e dell'overflow/underflow di interi prima di usare le funzioni DAX bitshift.
- Se Shift\_Amount è negativo, si sposterà nella direzione opposta.
- Se il valore assoluto di Shift\_Amount è maggiore di 64, non verrà generato alcun errore, ma si verificherà un overflow/underflow.
- Non c'è alcun limite al Numero, ma il risultato può verificarsi un overflow/underflow.

## Esempi

### Esempio 1

La query DAX seguente:

```
DAX
```

```
EVALUATE  
{ BITLSHIFT(2, 3) }
```

Restituisce 16.

## Esempio 2

La query DAX seguente:

```
DAX
```

```
EVALUATE  
{ BITLSHIFT(128, -1) }
```

Restituisce 64.

## Esempio 3

La query DAX seguente:

```
DAX
```

```
Define  
    Measure Sales[LeftShift] = BITLSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
    Sales[Amount],  
    "LEFTSHIFT",  
    [LeftShift]  
)
```

Sposta a sinistra ogni importo delle vendite con 3 bit e restituisce l'importo delle vendite spostato in bit.

## Vedi anche

[BITRSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# BITOR

Articolo • 20/10/2023

Restituisce un OR bit per bit di due numeri.

## Sintassi

DAX

```
BITOR(<number>, <number>)
```

## Parametri

Termine	Definizione
Numero	qualsiasi espressione scalare che restituisce il numero. Se non è un numero intero, viene troncato.

## Valore restituito

Un OR bit per bit di due numeri.

## Osservazioni:

- Questa funzione supporta numeri sia positivi che negativi.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
{ BITOR(9, 10) }
```

Restituisce 11.

## Vedi anche

[BITAND](#)

[BITXOR](#)

[BITLSHIFT](#)

[BITRSHIFT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# BITRSHIFT

Articolo • 20/10/2023

Restituisce un numero scorso a destra dal numero di bit specificato.

## Sintassi

DAX

```
BITRSHIFT(<Number>, <Shift_Amount>)
```

## Parametri

Termine	Definizione
Numero	qualsiasi espressione DAX che restituisce un numero intero.
Shift_Amount	qualsiasi espressione DAX che restituisce un numero intero.

## Valore restituito

Valore intero.

## Osservazioni:

- Assicurarsi di comprendere la natura delle operazioni di bitshift e dell'overflow/underflow di interi prima di usare le funzioni DAX bitshift.
- Se Shift\_Amount è negativo, si sposterà nella direzione opposta.
- Se il valore assoluto di Shift\_Amount è maggiore di 64, non verrà generato alcun errore, ma si verificherà un overflow/underflow.
- Non c'è alcun limite al Numero, ma il risultato può verificarsi un overflow/underflow.

## Esempi

### Esempio 1

La query DAX seguente:

```
DAX
```

```
EVALUATE  
{ BITRSHIFT(16, 3) }
```

Restituisce 2.

## Esempio 2

La query DAX seguente:

```
DAX
```

```
EVALUATE  
{ BITRSHIFT(1024, -3) }
```

Restituisce 8192.

## Esempio 3

La query DAX seguente:

```
DAX
```

```
Define  
    Measure Sales[RightShift] = BITRSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
    Sales[Amount],  
    "RIGHTSHIFT",  
    [RightShift]  
)
```

Sposta a destra ogni importo delle vendite con 3 bit e restituisce l'importo delle vendite spostato in bit.

## Vedi anche

[BITLSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# BITXOR

Articolo • 20/10/2023

Restituisce un XOR bit per bit di due numeri.

## Sintassi

DAX

```
BITXOR(<number>, <number>)
```

## Parametri

Termine	Definizione
Numero	qualsiasi espressione scalare che restituisce il numero. Se non è un numero intero, viene troncato.

## Valore restituito

Un XOR bit per bit di due numeri.

## Osservazioni:

- Questa funzione supporta numeri sia positivi che negativi.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { BITXOR(9, 10) }
```

Restituisce 3.

## Vedi anche

[BITOR](#)

[BITAND](#)

[BITLSHIFT](#)

[BITRSHIFT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COALESCE

Articolo • 20/10/2023

Restituisce la prima espressione che non restituisce BLANK. Se tutte le espressioni restituiscono BLANK, viene restituito BLANK.

## Sintassi

DAX

```
COALESCE(<expression>, <expression>[, <expression>]...)
```

## Parametri

Termine	Definizione
expression	Espressione DAX che restituisce un'espressione scalare.

## Valore restituito

Valore scalare proveniente da una delle espressioni oppure BLANK se tutte le espressioni restituiscono BLANK.

## Osservazioni:

Le espressioni di input possono essere di tipi di dati diversi.

## Esempio 1

La query DAX seguente:

DAX

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Restituisce 10, ovvero la prima espressione che non restituisce BLANK.

## Esempio 2

L'espressione DAX seguente:

DAX

```
= COALESCE(SUM(FactInternetSales[SalesAmount]), 0)
```

Restituisce la somma di tutti i valori nella colonna SalesAmount della tabella FactInternetSales oppure 0. Il risultato può essere usato per convertire i valori BLANK delle vendite totali in 0.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FALSE

Articolo • 20/10/2023

Restituisce il valore logico FALSE.

## Sintassi

DAX

```
FALSE()
```

## Valore restituito

Sempre FALSE.

## Osservazioni:

La parola FALSE viene interpretata anche come valore logico FALSE.

## Esempio

La formula restituisce il valore logico FALSE quando il valore nella colonna 'InternetSales\_USD'[SalesAmount\_USD] è minore o uguale a 200000.

DAX

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

La tabella seguente mostra i risultati dell'utilizzo della formula di esempio con 'ProductCategory' [ProductName] in Etichette di riga e 'DateTime'[CalendarYear] in Etichette di colonna.

Etichette di riga	2005	2006	2007	2008	-	Grand Total
Accessori	FALSE	FALSE	TRUE	TRUE	FALSO	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSO	TRUE
Clothing	FALSE	FALSO	FALSO	FALSO	FALSE	TRUE

<b>Etichette di riga</b>	<b>2005</b>	<b>2006</b>	<b>2007</b>	<b>2008</b>	-	<b>Grand Total</b>
Componenti	FALSE	FALSO	FALSO	FALSO	FALSO	FALSO
	FALSO	FALSO	FALSO	FALSO	FALSO	FALSE
Grand Total	TRUE	TRUE	TRUE	TRUE	FALSO	TRUE

## Vedi anche

[Funzione TRUE](#)

[Funzione NOT](#)

[Funzione IF](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# IF

Articolo • 20/10/2023

Controlla una condizione e restituisce un valore quando è TRUE. In caso contrario, restituisce un secondo valore.

## Sintassi

DAX

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

## Parametri

Termine	Definizione
logical_test	qualsiasi valore o espressione che può dare come risultato TRUE o FALSE.
value_if_true	Valore restituito se il test logico dà come risultato TRUE.
value_if_false	(Facoltativo) Valore restituito se il test logico dà come risultato FALSE. Se omesso, viene restituito BLANK.

## Valore restituito

value\_if\_true, value\_if\_false o BLANK.

## Osservazioni:

- La funzione IF può restituire un tipo di dati Variant se value\_if\_true e value\_if\_false sono tipi di dati diversi, ma la funzione tenta di restituire un unico tipo di dati se value\_if\_true e value\_if\_false sono entrambi tipi di dati numerici. Nel secondo caso la funzione IF convertirà in modo implicito i tipi di dati in modo da usare entrambi i valori.

Ad esempio, la formula `IF(<condition>, TRUE(), 0)` restituisce TRUE o 0, ma la formula `IF(<condition>, 1.0, 0)` restituisce solo valori decimali anche se value\_if\_false è un tipo di dati numero intero. Per altre informazioni sulla conversione implicita dei tipi di dati, vedere [Tipi di dati](#).

- Per eseguire le espressioni di ramo indipendentemente dall'espressione della condizione, usare [IF.EAGER](#).

## Esempi

Le definizioni delle colonne calcolate della tabella **Product** seguenti usano la funzione IF in modi diversi per classificare ogni prodotto in base al prezzo di listino.

Nel primo esempio viene verificato se il valore della colonna **List Price** è inferiore a 500. Quando questa condizione è true, viene restituito il valore **Low**. Poiché non esiste alcun valore **value\_if\_false**, viene restituito BLANK.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low"
)
```

Nel secondo esempio viene usato lo stesso test, ma questa volta include un valore **value\_if\_false**. Quindi, la formula classifica ogni prodotto come **Low** o **High**.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    "High"
)
```

Nel terzo esempio viene usato lo stesso test, ma questa volta con una funzione IF annidata per eseguire un test aggiuntivo. La formula classifica quindi ogni prodotto come **Low**, **Medium** o **High**.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
```

```
"Low",
IF(
    'Product'[List Price] < 1500,
    "Medium",
    "High"
)
)
```

## 💡 Suggerimento

Quando è necessario annidare più funzioni IF, la funzione **SWITCH** potrebbe essere un'opzione migliore. Questa funzione offre un modo più elegante per scrivere un'espressione che restituisce più di due valori possibili.

## Vedi anche

[Funzione IF.EAGER](#)

[Funzione SWITCH \(DAX\)](#)

[Funzioni logiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# IF.EAGER

Articolo • 20/10/2023

Controlla una condizione e restituisce un valore quando TRUE, in caso contrario restituisce un secondo valore. Usa un piano di esecuzione *eager* che esegue sempre le espressioni di ramo indipendentemente dall'espressione della condizione.

## Sintassi

DAX

```
IF.EAGER(<logical_test>, <value_if_true>[, <value_if_false>])
```

## Parametri

Termine	Definizione
logical_test	Qualsiasi valore o espressione che può dare come risultato TRUE o FALSE.
value_if_true	Valore restituito se il test logico dà come risultato TRUE.
value_if_false	(Facoltativo) Valore restituito se il test logico dà come risultato FALSE. Se omesso, viene restituito BLANK.

## Valore restituito

value\_if\_true, value\_if\_false o BLANK.

## Osservazioni:

- La funzione IF.EAGER può restituire un tipo di dati variant se value\_if\_true e value\_if\_false sono di tipi di dati diversi, ma la funzione tenta di restituire un singolo tipo di dati se sia value\_if\_true che value\_if\_false sono di tipi di dati numerici. Nel secondo caso la funzione IF.EAGER convertirà in modo implicito i tipi di dati in modo da usare entrambi i valori.

Ad esempio, la formula `IF.EAGER(<condition>, TRUE(), 0)` restituisce TRUE o 0, ma la formula `IF.EAGER(<condition>, 1.0, 0)` restituisce solo valori decimali anche se

**value\_if\_false** è un tipo di dati numero intero. Per altre informazioni sulla conversione implicita dei tipi di dati, vedere [Tipi di dati](#).

- IF.EAGER ha lo stesso comportamento funzionale della funzione IF, ma le prestazioni possono variare a causa delle differenze nei piani di esecuzione. `IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>)` ha lo stesso piano di esecuzione dell'espressione DAX seguente:

DAX

```
VAR _value_if_true = <value_if_true>
VAR _value_if_false = <value_if_false>
RETURN
IF (<logical_test>, _value_if_true, _value_if_false)
```

Nota: le due espressioni di ramo vengono valutate indipendentemente dall'espressione della condizione.

## Esempi

Vedere [Esempi IF](#).

## Vedi anche

[Funzione IF](#)

[Funzioni logiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# IFERROR

Articolo • 20/10/2023

Valuta un'espressione e restituisce un valore specificato se l'espressione restituisce un errore. In caso contrario, restituisce il valore dell'espressione stessa.

## Sintassi

DAX

```
IFERROR(value, value_if_error)
```

## Parametri

Termine	Definizione
value	qualsiasi valore o espressione.
value_if_error	qualsiasi valore o espressione.

## Valore restituito

Valore scalare dello stesso tipo di **value**.

## Osservazioni:

- È possibile usare la funzione IFERROR per individuare e gestire gli errori in un'espressione.
- Se **value** o **value\_if\_error** è una cella vuota, IFERROR la considera come un valore stringa vuoto ("").
- La funzione IFERROR è basata sulla funzione IF e usa gli stessi messaggi di errore, ma ha meno argomenti. La relazione tra la funzione IFERROR e la funzione IF è la seguente:

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

I valori restituiti per A e B devono essere dello stesso tipo di dati, pertanto la colonna o l'espressione usata per **value** e il valore restituito per **value\_if\_error**

devono avere lo stesso tipo di dati.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- Per le procedure consigliate quando si usa ISERROR, vedere [Uso appropriato delle funzioni di errore](#).

## Esempio

L'esempio seguente restituisce 9999 se l'espressione 25/0 restituisce un errore. Se l'espressione restituisce un valore diverso da un errore, questo valore viene passato all'espressione chiamante.

DAX

```
= IFERROR(25/0,9999)
```

## Vedi anche

[Funzioni logiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# NOT

Articolo • 20/10/2023

Cambia FALSE in TRUE o TRUE in FALSE.

## Sintassi

DAX

```
NOT(<logical>)
```

## Parametri

Termine	Definizione
logico	Valore o espressione che può dare come risultato TRUE o FALSE.

## Valore restituito

TRUE o FALSE.

## Esempio

Nell'esempio seguente vengono recuperati i valori della colonna calcolata creata per illustrare la funzione IF. Per tale esempio, alla colonna calcolata è stato assegnato il nome predefinito, **Calculated Column1**, e la colonna contiene la formula seguente: =  
IF([Orders]<300,"true","false")

La formula controlla il valore della colonna [Orders] e restituisce "true" se il numero di ordini è minore di 300.

Creare ora una nuova colonna calcolata, **Calculated Column2** e digitare la formula seguente.

DAX

```
= NOT([CalculatedColumn1])
```

Per ogni riga in **Calculated Column1** i valori "true" e "false" vengono interpretati come i valori logici TRUE o FALSE e la funzione NOT restituisce l'opposto logico di tale valore.

## Vedi anche

[Funzione TRUE](#)

[Funzione FALSE](#)

[Funzione IF](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# OPPURE

Articolo • 20/10/2023

Per restituire TRUE, verifica se uno degli argomenti è TRUE. Se entrambi gli argomenti sono FALSE, la funzione restituisce FALSE.

## Sintassi

DAX

```
OR(<logical1>,<logical2>)
```

## Parametri

Termine	Definizione
logical_1, logical_2	Valori logici da testare.

## Valore restituito

Valore booleano. Il valore è TRUE se uno dei due argomenti è TRUE; è FALSE se entrambi gli argomenti sono FALSE.

## Osservazioni:

- La funzione OR in DAX accetta solo due (2) argomenti. Se è necessario eseguire un'operazione OR su più espressioni, è possibile creare una serie di calcoli o, meglio, usare l'operatore OR (||) per unirli tutti in un'espressione più semplice.
- La funzione valuta gli argomenti fino al primo argomento TRUE, quindi restituisce TRUE.

## Esempio

L'esempio seguente illustra come usare la funzione OR per ottenere gli agenti di vendita che appartengono al Circle of Excellence. Il riconoscimento Circle of Excellence viene attribuito agli agenti che hanno realizzato più di un milione di dollari di vendite nel settore delle Touring Bikes o vendite superiori a due milioni e mezzo di dollari nel 2007.

DAX

```
IF( OR( CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),  
'ProductSubcategory'[ProductSubcategoryName] = "Touring Bikes") > 1000000  
        , CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),  
'DateTime'[CalendarYear] = 2007) > 2500000  
        )  
    , "Circle of Excellence"  
    , ""  
)
```

Resi

<b>Etichette di riga</b>	<b>2005</b>	<b>2006</b>	<b>2007</b>	<b>2008</b>	-	<b>Grand Total</b>
Reiter, Tsvi Michael					-	
Saraiva, José Edvaldo	Circle of Excellence					
Tsoflias, Lynn N					-	
Valdez, Rachel B					-	
Vargas, Garrett R					-	
Varkey Chudukatil, Ranjit R						Circle of Excellence
Grand Total	Circle of Excellence					

## Vedi anche

[Funzioni logiche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# SWITCH

Articolo • 20/10/2023

Valuta un'espressione sulla base di un elenco di valori e restituisce una tra più espressioni di risultato possibili. Questa funzione può essere usata per evitare di avere più istruzioni [IF](#) annidate.

## Sintassi

DAX

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

## Parametri

Termine	Definizione
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).
value	Valore costante da confrontare con i risultati di <b>expression</b> .
result	Qualsiasi espressione scalare da valutare se i risultati di <b>expression</b> corrispondono all'elemento <b>value</b> corrispondente.
else	Qualsiasi espressione scalare da valutare se i risultati di <b>expression</b> non corrispondono a nessuno degli argomenti di <b>value</b> .

## Valore restituito

Se esiste una corrispondenza con un argomento **value**, viene restituito un valore scalare dal **result** corrispondente. Se non esiste una corrispondenza con un **value**, viene restituito un valore **else**. Se nessuno degli argomenti **value** corrisponde ed **else** non è specificato, viene restituito BLANK.

## Osservazioni:

- L'argomento **expression** da valutare può essere un valore costante o un'espressione. Un uso comune di questa funzione consiste nell'impostare il primo parametro su TRUE. Vedere gli esempi seguenti.

- Tutte le espressioni **result** l'espressione **else** devono essere dello stesso tipo di dati.
- L'ordine delle condizioni è importante. Non appena un **value** corrisponde, viene restituito il **result** corrispondente e gli altri argomenti **value** successivi non vengono valutati. Assicurarsi di specificare gli argomenti **value** più restrittivi da valutare prima dei **value** meno restrittivi. Vedere gli esempi seguenti.

## Esempi

Un uso comune di SWITCH consiste nel confrontare **expression** con argomenti **value** costanti. L'esempio seguente crea una colonna calcolata di nomi di mesi:

DAX

```
= SWITCH (
    [Month Number Of Year],
    1, "January",
    2, "February",
    3, "March",
    4, "April",
    5, "May",
    6, "June",
    7, "July",
    8, "August",
    9, "September",
    10, "October",
    11, "November",
    12, "December",
    "Unknown month number"
)
```

Un altro uso comune di SWITCH è sostituire più istruzioni IF annidate. Questa operazione viene eseguita impostando l'espressione su TRUE, come illustrato nell'esempio seguente, che confronta Reorder Point and Safety Stock Level nei prodotti per identificare i potenziali rischi di esaurimento delle scorte:

DAX

```
= SWITCH (
    TRUE,
    [Reorder Point] > [Safety Stock Level], "Good: Safety stock level exceeded",
    [Reorder Point] = [Safety Stock Level], "Minimal: Safety stock level met",
    [Reorder Point] < [Safety Stock Level], "At risk: Safety stock level not met",
    ISBLANK ( [Reorder Point] ), "Incomplete: Reorder point not set",
```

```
ISBLANK ( [Safety Stock Level] ), "Incomplete: Safety stock level  
not set",  
        "Unknown"  
    )
```

L'ordine degli argomenti **value** è importante. Nell'esempio seguente il secondo **result** non viene mai restituito perché il primo valore è meno restrittivo del secondo. Il risultato di questo esempio è sempre "A" o "C", ma mai "B".

DAX

```
= SWITCH (  
    TRUE,  
    Product[Standard Cost] < 100, "A",  
    Product[Standard Cost] < 10, "B",  
    "C"  
)
```

L'istruzione seguente restituisce un errore perché i tipi di dati nell'argomento **result** sono diversi. Tenere presente che i tipi di dati in tutti gli argomenti **result** ed **else** devono essere gli stessi.

DAX

```
= SWITCH (  
    [Class],  
    "L", "Large",  
    "H", 0.1  
)
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# TRUE

Articolo • 20/10/2023

Restituisce il valore logico TRUE.

## Sintassi

DAX

```
TRUE()
```

## Valore restituito

Sempre TRUE.

## Osservazioni:

La parola TRUE viene interpretata anche come valore logico TRUE.

## Esempio

La formula restituisce il valore logico TRUE quando il valore nella colonna 'InternetSales\_USD'[SalesAmount\_USD] è maggiore di 200000.

DAX

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

La tabella seguente mostra i risultati quando la formula di esempio è usata in un report con 'ProductCategory'[ProductName] in Etichette di riga e 'DateTime'[CalendarYear] in Etichette di colonna.

Etichette di riga	2005	2006	2007	2008	-	Grand Total
Accessori	FALSE	FALSE	TRUE	TRUE	FALSO	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSO	TRUE
Clothing	FALSE	FALSO	FALSO	FALSO	FALSE	TRUE

<b>Etichette di riga</b>	<b>2005</b>	<b>2006</b>	<b>2007</b>	<b>2008</b>	-	<b>Grand Total</b>
Componenti	FALSE	FALSO	FALSO	FALSO	FALSO	FALSO
	FALSO	FALSO	FALSO	FALSO	FALSO	FALSE
Grand Total	TRUE	TRUE	TRUE	TRUE	FALSO	TRUE

## Vedi anche

[FALSE](#)

[NOT](#)

[IF](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# Funzioni matematiche e trigonometriche

Articolo • 20/10/2023

Le funzioni matematiche in Data Analysis Expressions (DAX) sono molto simili alle funzioni matematiche e trigonometriche di Excel. Questa sezione elenca le funzioni matematiche fornite da DAX.

## In questa categoria

Funzione	Descrizione
ABS	Restituisce il valore assoluto di un numero.
ACOS	Restituisce l'arcocoseno, o coseno inverso, di un numero.
ACOSH	Restituisce l'inversa del coseno iperbolico di un numero.
ACOT	Restituisce l'arcocotangente, o cotangente inversa, di un numero.
ACOTH	Restituisce la cotangente iperbolica inversa di un numero.
ASIN	Restituisce l'arcoseno, o seno inverso, di un numero.
ASINH	Restituisce l'inversa del seno iperbolico di un numero.
ATAN	Restituisce l'arcotangente, o tangente inversa, di un numero.
ATANH	Restituisce la tangente iperbolica inversa di un numero.
CEILING	Arrotonda un numero per eccesso all'intero più vicino o al multiplo più vicino.
CONVERT	Consente di convertire un'espressione da un tipo di dati a un altro.
COS	Restituisce il coseno dell'angolo specificato.
COSH	Restituisce il coseno iperbolico di un numero.
COT	Restituisce la cotangente di un angolo, espresso in radianti.
COTH	Restituisce la cotangente iperbolica di un angolo iperbolico.
CURRENCY	Valuta l'argomento e restituisce il risultato come tipo di dati valuta.
DEGREES	Converte i radianti in gradi.
DIVIDE	Esegue una divisione e restituisce il risultato alternativo o BLANK() della

<b>Funzione</b>	<b>Descrizione</b>
	divisione per 0.
<b>EVEN</b>	Restituisce il numero arrotondato per eccesso al numero intero pari più vicino.
<b>EXP</b>	Restituisce il numero e elevato alla potenza di un dato numero.
<b>FACT</b>	Restituisce il fattoriale di un numero, uguale alla serie $1*2*3*...*$ , che termina con il numero specificato.
<b>FLOOR</b>	Arrotonda un numero per difetto, verso lo zero, al multiplo significativo più vicino.
<b>GCD</b>	Restituisce il massimo comun divisore di due o più numeri interi.
<b>INT</b>	Arrotonda un numero per difetto all'intero più vicino.
<b>ISO.CEILING</b>	Arrotonda un numero per eccesso all'intero più vicino o al multiplo più vicino.
<b>LCM</b>	Restituisce il minimo comune multiplo dei numeri interi.
<b>LN</b>	Restituisce il logaritmo naturale di un numero.
<b>LOG</b>	Restituisce il logaritmo di un numero nella base specificata.
<b>LOG10</b>	Restituisce il logaritmo in base 10 di un numero.
<b>MOD</b>	Restituisce il resto dopo la divisione di un numero per un divisore. Il risultato ha sempre lo stesso segno del divisore.
<b>MROUND</b>	Restituisce un numero arrotondato al multiplo desiderato.
<b>ODD</b>	Restituisce il numero arrotondato per eccesso al numero intero dispari più vicino.
<b>PI</b>	Restituisce il valore di pi greco, 3,14159265358979, arrotondato a 15 cifre.
<b>POWER</b>	Restituisce il risultato di un numero elevato a potenza.
<b>QUOTIENT</b>	Esegue una divisione e restituisce solo la parte intera del risultato della divisione.
<b>RADIANS</b>	Converte i gradi in radianti.
<b>RAND</b>	Restituisce un numero casuale uniformemente distribuito e superiore o uguale a 0 e minore di 1.
<b>RANDBETWEEN</b>	Restituisce un numero casuale nell'intervallo tra due numeri specificati.
<b>ROUND</b>	Arrotonda un numero al numero di cifre specificato.
<b>ROUNDDOWN</b>	Arrotonda per difetto il valore assoluto di un numero.

Funzione	Descrizione
ROUNDUP	Arrotonda per eccesso il valore assoluto di un numero.
SIGN	Determina il segno di un numero, del risultato di un calcolo o di un valore in una colonna.
SIN	Restituisce il seno dell'angolo specificato.
SINH	Restituisce il seno iperbolico di un numero.
SQRT	Restituisce la radice quadrata di un numero.
SQRTPI	Restituisce la radice quadrata di (numero * pi).
TAN	Restituisce la tangente dell'angolo specificato.
TANH	Restituisce la tangente iperbolica di un numero.
TRUNC	Elimina la parte decimale o frazionaria di un numero.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# abs

Articolo • 20/10/2023

Restituisce il valore assoluto di un numero.

## Sintassi

DAX

```
ABS(<number>)
```

## Parametri

Termine	Definizione
number	Numero di cui calcolare il valore assoluto.

## Valore restituito

Numero decimale.

## Osservazioni:

Il valore assoluto di un numero è un numero reale, intero o decimale, senza segno. È possibile usare la funzione ABS per assicurarsi che le espressioni restituiscano solo numeri non negativi quando sono annidate in funzioni che richiedono un numero positivo.

## Esempio

L'esempio seguente restituisce il valore assoluto della differenza tra il prezzo di listino e il prezzo del rivenditore, che è possibile usare in una nuova colonna calcolata, DealerMarkup.

DAX

```
= ABS([DealerPrice]-[ListPrice])
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)  
[funzione SIGN](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ACOS

Articolo • 20/10/2023

Restituisce l'arcocoseno, o coseno inverso, di un numero. L'arcocoseno è l'angolo il cui coseno è *number*. Il valore dell'angolo restituito è in radianti in un intervallo compreso tra 0 (zero) e pi greco.

## Sintassi

DAX

`ACOS(number)`

## Parametri

Termine	Definizione
Numero	Coseno dell'angolo desiderato che deve essere compreso tra -1 e 1.

## Valore restituito

Restituisce l'arcocoseno, o coseno inverso, di un numero.

## Osservazioni:

Se si vuole convertire il risultato in gradi, moltiplicarlo per  $180/\text{PI}()$  o usare la funzione DEGREES.

## Esempio

Formula	Description	Result
= ACOS(-0.5)	Arcocoseno di -0.5 in radianti, $2\pi/3$ .	2.094395102
= ACOS(-0.5)*180/PI()	Arcocoseno di -0.5 in gradi.	120

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ACOSH

Articolo • 20/10/2023

Restituisce l'inversa del coseno iperbolico di un numero. Il numero deve essere maggiore o uguale a 1. L'inversa del coseno iperbolico è il valore il cui coseno iperbolico è *number*. ACOSH(COSH(number)) equivale quindi a *number*.

## Sintassi

DAX

`ACOSH(number)`

## Parametri

Termine	Definizione
number	qualsiasi numero reale maggiore o uguale a 1.

## Valore restituito

Restituisce l'inversa del coseno iperbolico di un numero.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= ACOSH(1)	Inversa del coseno iperbolico di 1.	0
= ACOSH(10)	Inversa del coseno iperbolico di 10.	2.993228

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ACOT

Articolo • 20/10/2023

Restituisce il valore principale dell'arcotangente, o funzione inversa della cotangente, di un numero.

## Sintassi

DAX

ACOT(number)

## Parametri

Termine	Definizione
Numero	Il coseno dell'angolo che interessa. Deve essere un numero reale.

## Valore restituito

Un singolo valore decimale.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ACOTH

Articolo • 20/10/2023

Restituisce la cotangente iperbolica inversa di un numero.

## Sintassi

DAX

`ACOTH(number)`

## Parametri

Termine	Definizione
Numero	Il valore assoluto del numero deve essere maggiore di 1.

## Valore restituito

Un singolo valore decimale.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# ASIN

Articolo • 20/10/2023

Restituisce l'arcoseno, o seno inverso, di un numero. L'arcoseno è l'angolo il cui seno è *number*. L'angolo risultante viene espresso in radianti con un valore compreso tra -pi greco/2 e pi greco/2.

## Sintassi

DAX

**ASIN(number)**

## Parametri

Termine	Definizione
number	Seno dell'angolo desiderato che deve essere compreso tra -1 e 1.

## Valore restituito

Restituisce l'arcoseno, o seno inverso, di un numero.

## Osservazioni:

Per esprimere l'arcoseno in gradi, moltiplicare il risultato per 180/PI( ) o usare la funzione DEGREES.

## Esempio

Formula	Description	Result
= ASIN(-0.5)	Arcoseno di 0.5 in radianti, -pi greco/6	-0.523598776
= ASIN(-0.5)*180/PI()	Arcoseno di -0.5 in gradi	-30
= DEGREES(ASIN(-0.5))	Arcoseno di -0.5 in gradi	-30

# Commenti e suggerimenti

Questa pagina è stata utile?



# ASINH

Articolo • 20/10/2023

Restituisce l'inversa del seno iperbolico di un numero. La funzione inversa del seno iperbolico è il valore il cui seno iperbolico è *number*, quindi ASINH(SINH(*number*)) equivale *number*.

## Sintassi

DAX

`ASINH(number)`

## Parametri

Termine	Definizione
number	qualsiasi numero reale.

## Valore restituito

Restituisce l'inversa del seno iperbolico di un numero.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= ASINH(-2.5)	Inversa del seno iperbolico di -2.5	-1.647231146
= ASINH(10)	Inversa del seno iperbolico di 10	2.99822295

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ATAN

Articolo • 20/10/2023

Restituisce l'arcotangente, o tangente inversa, di un numero. L'arcotangente è l'angolo la cui tangente è *number*. L'angolo risultante viene espresso in radianti con un valore compreso tra -pi greco/2 e pi greco/2.

## Sintassi

DAX

`ATAN(number)`

## Parametri

Termine	Definizione
number	Tangente dell'angolo desiderato.

## Valore restituito

Restituisce la tangente iperbolica inversa di un numero.

## Osservazioni:

Per esprimere l'arcotangente in gradi, moltiplicare il risultato per 180/PI() o usare la funzione DEGREES.

## Esempio

Formula	Description	Result
= ATAN(1)	Arcotangente di 1 in radianti, pi greco/4	0.785398163
= ATAN(1)*180/PI()	Arcotangente di 1 in gradi	45

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ATANH

Articolo • 20/10/2023

Restituisce la tangente iperbolica inversa di un numero. Il numero deve essere compreso tra -1 e 1 (esclusi -1 e 1). L'inversa della tangente iperbolica è il valore la cui tangente iperbolica è *numero*, quindi ATANH(TANH(*numero*)) è uguale al *numero*.

## Sintassi

DAX

`ATANH(number)`

## Parametri

Termine	Definizione
number	qualsiasi numero reale compreso tra 1 e -1.

## Valore restituito

Restituisce la tangente iperbolica inversa di un numero.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= ATANH(0.76159416)	Inversa della tangente iperbolica di 0.76159416	1.00000001
= ATANH(-0.1)		-0.100335348

## Vedi anche

## Commenti e suggerimenti

Questa pagina è stata utile?



# CEILING

Articolo • 20/10/2023

Arrotonda un numero per eccesso all'intero più vicino o al multiplo più vicino.

## Sintassi

DAX

`CEILING(<number>, <significance>)`

## Parametri

Termine	Definizione
number	Numero da arrotondare o riferimento a una colonna contenente numeri.
significance	Multiplo a cui si vuole eseguire l'arrotondamento. Ad esempio per arrotondare al numero intero più vicino, digitare 1.

## Valore restituito

Numero arrotondato come specificato.

## Osservazioni:

- In DAX sono disponibili due funzioni CEILING, con le differenze seguenti:
  - La funzione CEILING emula il comportamento della funzione ARROTONDA.ECESSO in Excel.
  - La funzione ISO.CEILING segue il comportamento definito da ISO per la determinazione del valore di arrotondamento per eccesso.
- Le due funzioni restituiscono lo stesso valore per i numeri positivi, ma valori diversi per i numeri negativi. Se si usa un multiplo significativo positivo, sia la funzione CEILING sia la funzione ISO.CEILING arrotondano i numeri negativi per eccesso (verso l'infinito positivo). Se si usa un multiplo significativo negativo, la funzione CEILING arrotonda i numeri negativi per difetto (verso l'infinito negativo), mentre la funzione ISO.CEILING arrotonda i numeri negativi per eccesso (verso l'infinito positivo).

- Il tipo restituito è generalmente dello stesso tipo dell'argomento significativo, con le eccezioni seguenti:
  - Se il tipo di argomento del numero è currency, il tipo restituito è currency.
  - Se il tipo di argomento di significance è Boolean, il tipo restituito è integer.
  - Se il tipo di argomento di significance è non numerico, il tipo restituito è real.

## Esempio 1

La formula seguente restituisce 4.45. Questo potrebbe essere utile se si vuole evitare di usare unità più piccole nella determinazione dei prezzi. Se il prezzo di un prodotto esistente è fissato a \$4.42, si può usare la funzione CEILING per arrotondare per eccesso all'unità cinque centesimi più vicina.

DAX

```
= CEILING(4.42,0.05)
```

## Esempio 2

La formula seguente restituisce risultati simili all'esempio precedente, ma usa i valori numerici archiviati nella colonna **ProductPrice**.

DAX

```
= CEILING([ProductPrice],0.05)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione FLOOR](#)

[Funzione ISO.CEILING](#)

[Funzione ROUNDUP](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# CONVERT

Articolo • 20/10/2023

Consente di convertire un'espressione da un tipo di dati a un altro.

## Sintassi

DAX

```
CONVERT(<Expression>, <Datatype>)
```

## Parametri

Termine	Definizione
Expression	Qualsiasi espressione valida.
Datatype	Enumerazione che include: INTEGER(Numero intero), DOUBLE(Numero decimale), STRING(Testo), BOOLEAN(Vero/Falso), CURRENCY(Numero decimale fisso), DATETIME(Data, Ora, ecc.).

## Valore restituito

Restituisce il valore di <Espressione>, convertito in <Tipo di dato>.

## Osservazioni:

- Quando non è possibile convertire un valore nel tipo di dati specificato, la funzione restituisce un errore.
- Le colonne calcolate DAX devono essere di un solo tipo di dati. Poiché le funzioni MEDIAN e MEDIANX su una colonna Integer restituiscono tipi di dati misti, Integer o Double, l'espressione di colonna calcolata seguente restituirà come risultato un errore: `MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned])`.
- Per evitare tipi di dati misti, modificare l'espressione in modo che restituisca sempre il tipo di dati Double, ad esempio:  
`MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned], DOUBLE))`.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Query DAX

```
DAX  
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Resi

[Value]
2

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COS

Articolo • 20/10/2023

Restituisce il coseno dell'angolo specificato.

## Sintassi

DAX

`COS(number)`

## Parametri

Termine	Definizione
number	Required. Angolo in radianti per cui calcolare il coseno.

## Valore restituito

Restituisce il coseno dell'angolo specificato.

## Osservazioni:

Se l'angolo è espresso in gradi, moltiplicarlo per PI()/180 oppure usare la funzione RADIANS per convertire l'angolo in radianti.

## Esempio

Formula	Description	Result
= COS(1.047)	Coseno di 1.047 radianti	0.5001711
= COS(60*PI()/180)	Coseno di 60 gradi	0.5
= COS(RADIANS(60))	Coseno di 60 gradi	0.5

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# COSH

Articolo • 20/10/2023

Restituisce il coseno iperbolico di un numero.

## Sintassi

DAX

`COSH(number)`

## Parametri

Termine	Definizione
number	Required. Qualsiasi numero reale per il quale si vuole calcolare il coseno iperbolico.

## Valore restituito

Coseno iperbolico di un numero.

## Osservazioni:

- La formula per il coseno iperbolico è:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= COSH(4)	Coseno iperbolico di 4	27.308233
= COSH(EXP(1))	Coseno iperbolico della base del logaritmo naturale.	7.6101251

# Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# COT

Articolo • 20/10/2023

Restituisce la cotangente di un angolo, espresso in radianti.

## Sintassi

DAX

```
COT (<number>)
```

## Parametri

Termine	Definizione
number	Angolo in radianti per cui calcolare la cotangente.

## Valore restituito

Cotangente dell'angolo specificato.

## Osservazioni:

- Il valore assoluto del numero deve essere minore di  $2^{27}$  e non può essere 0.
- Se il numero non rientra nei vincoli, viene restituito un errore.
- Se il numero non è un valore numerico, viene restituito un errore.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { COT(30) }
```

Resi

[Value]

-0.156119952161659

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COTH

Articolo • 20/10/2023

Restituisce la cotangente iperbolica di un angolo iperbolico.

## Sintassi

DAX

```
COTH (<number>)
```

## Parametri

Termine	Definizione
number	Angolo iperbolico in radianti per cui calcolare la cotangente iperbolica.

## Valore restituito

Cotangente iperbolica dell'angolo specificato.

## Osservazioni:

- La cotangente iperbolica è analoga alla cotangente ordinaria (circolare).
- Il valore assoluto di number deve essere minore di  $2^{27}$  e non può essere 0.
- Se il numero non rientra nei vincoli, viene restituito un errore
- Se il numero non è un valore numerico, viene restituito un errore.
- Usare l'equazione seguente:

$$\text{COTH}(N) = \frac{1}{\text{TANH}(N)} = \frac{\text{COSH}(N)}{\text{SINH}(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { COTH(2) }
```

Resi

[Value]

1.03731472072755

## Commenti e suggerimenti

Questa pagina è stata utile?



# VALUTA

Articolo • 20/10/2023

Valuta l'argomento e restituisce il risultato come tipo di dati valuta.

## Sintassi

DAX

`CURRENCY(<value>)`

## Parametri

Termine	Definizione
value	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata esattamente una volta prima di tutte le altre operazioni.

## Valore restituito

Valore dell'espressione valutato e restituito come tipo di dati valuta.

## Osservazioni:

- La funzione CURRENCY arrotonda il 5° decimale significativo, in valore, per restituire la 4a cifra decimale. L'arrotondamento si verifica se il 5° decimale significativo è uguale o maggiore di 5. Ad esempio, se il valore è 3,666666666666, la conversione in valuta restituisce \$3,6667. Tuttavia, se il valore è 3,0123456789, la conversione in valuta restituisce \$3,0123.
- Se il tipo di dati dell'espressione è `TrueFalse`, `CURRENCY(<TrueFalse>)` restituirà `1.0000` per i valori `True` e `0.0000` per i valori `False`.
- Se il tipo di dati dell'espressione è `Text`, `CURRENCY(<Text>)` tenterà di convertire il testo in un numero. Se la conversione ha esito positivo, il numero verrà convertito in valuta, in caso contrario viene restituito un errore.
- Se il tipo di dati dell'espressione è `DateTime`, `CURRENCY(<DateTime>)` convertirà il valore `datetime` in un numero e quel numero in valuta. I valori `DateTime` hanno

una parte intera che rappresenta il numero di giorni tra la data specificata e il 1 marzo 1900 e una frazione che rappresenta la frazione di un giorno (dove 0.5 equivale a 12 ore o mezzogiorno). Se il valore dell'espressione non è un valore DateTime appropriato, viene restituito un errore.

## Esempio

Convertire il numero 1234.56 nel tipo di dati valuta.

DAX

```
= CURRENCY(1234.56)
```

Restituisce il valore 1234.56000.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DEGREES

Articolo • 20/10/2023

Converte i radienti in gradi.

## Sintassi

DAX
<code>DEGREES(angle)</code>

## Parametri

Termine	Definizione
angle	Required. Angolo in radienti da convertire.

## Esempio

Formula	Description	Result
= DEGREES(PI())	Gradi di radienti pi greco	180

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# Divisione

Articolo • 20/10/2023

Esegue una divisione e restituisce il risultato alternativo o BLANK() della divisione per 0.

## Sintassi

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

## Parametri

Termine	Definizione
numerator	Dividendo o numero da dividere.
denominator	Divisore o numero per cui dividere.
alternateresult	(Facoltativo) Il valore restituito se la divisione per zero genera un errore. Se non viene fornito, il valore predefinito è BLANK().

## Valore restituito

- Numero decimale.

## Osservazioni:

- Il risultato alternativo nelle divisione per 0 deve essere una costante.
- Per le procedure consigliate quando si usa DIVIDE, vedere [funzione DIVIDE vs. operatore di divisione \(/\) in DAX](#).

## Esempio

L'esempio seguente restituisce 2.5.

DAX

```
= DIVIDE(5,2)
```

## Esempio 1

L'esempio seguente restituisce BLANK.

```
DAX
```

```
= DIVIDE(5,0)
```

## Esempio 2

L'esempio seguente restituisce 1.

```
DAX
```

```
= DIVIDE(5,0,1)
```

## Vedi anche

[Funzione QUOTIENT](#)

[Funzioni matematiche e trigonometriche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EVEN

Articolo • 20/10/2023

Restituisce il numero arrotondato per eccesso al numero intero pari più vicino. È possibile usare questa funzione per elaborare gli elementi a coppie. Ad esempio, una cassa da imballaggio può contenere file di uno o due elementi. La cassa è piena quando il numero di elementi, arrotondato al numero pari più vicino, corrisponde alla capacità della cassa.

## Sintassi

DAX
<code>EVEN(number)</code>

## Parametri

Termine	Definizione
number	Il valore da arrotondare.

## Valore restituito

Restituisce il numero arrotondato per eccesso al numero intero pari più vicino.

## Osservazioni:

- Se number non è un valore numerico, EVEN restituisce il valore di errore #VALUE! .
- Indipendentemente dal segno del numero, un valore viene arrotondato per eccesso quando il suo valore si allontana da zero. Se number è un numero intero pari, non viene eseguito alcun arrotondamento.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= EVEN(1.5)	Arrotonda 1.5 al numero intero pari più vicino	2
= EVEN(3)	Arrotonda 3 al numero intero pari più vicino	4
= EVEN(2)	Arrotonda 2 al numero intero pari più vicino	2
= EVEN(-1)	Arrotonda -1 al numero intero pari più vicino	-2

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# EXP

Articolo • 20/10/2023

Restituisce il numero e elevato alla potenza di un dato numero. La costante e è uguale a 2.71828182845904, la base del logaritmo naturale.

## Sintassi

DAX

`EXP(<number>)`

## Parametri

Termine	Definizione
number	Esponente applicato alla base e. La costante e è uguale a 2.71828182845904, la base del logaritmo naturale.

## Valore restituito

Numero decimale.

## Eccezioni

## Osservazioni:

- EXP è l'inverso di LN, ovvero il logaritmo naturale del numero specificato.
- Per calcolare la potenza di basi diverse da e, usare l'operatore di elevamento a potenza (^). Per altre informazioni, vedere [Informazioni di riferimento sulla sintassi DAX](#).

## Esempio

La formula seguente calcola un valore ed eleva tale valore alla potenza del numero contenuto nella colonna, [Power].

DAX

= EXP([Power])

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione LN](#)

[Funzione EXP](#)

[Funzione LOG](#)

[Funzione LOG](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



Sì



No

# FACT

Articolo • 20/10/2023

Restituisce il fattoriale di un numero, uguale alla serie  $1*2*3*...*$ , che termina con il numero dato.

## Sintassi

DAX

```
FACT(<number>)
```

## Parametri

Termine	Definizione
number	Numero non negativo per il quale si vuole calcolare il fattoriale.

## Valore restituito

Numero decimale.

## Osservazioni:

- Se il numero non è un numero intero, viene troncato e viene restituito un errore.  
Se il risultato è troppo grande, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente restituisce il fattoriale per la serie di numeri interi nella colonna `[Values]`.

DAX

```
= FACT([Values])
```

Nella tabella seguente vengono illustrati i risultati previsti:

Valori	Risultati
0	1
1	1
2	2
3	6
4	24
5	120
170	7.257415615308E+306

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione TRUNC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FLOOR

Articolo • 20/10/2023

Arrotonda un numero per difetto, verso lo zero, al multiplo significativo più vicino.

## Sintassi

DAX

```
FLOOR(<number>, <significance>)
```

## Parametri

Termine	Definizione
number	Valore numerico da arrotondare.
significance	Multiplo a cui arrotondare il numero. Gli argomenti <b>number</b> e <b>significance</b> devono essere entrambi positivi o negativi.

## Valore restituito

Numero decimale.

## Osservazioni:

- Se uno degli argomenti è di tipo non numerico, FLOOR restituisce il valore di errore \*\*#VALUE!\*\*.
- Se number e significance hanno segni diversi, FLOOR restituisce il valore di errore \*\*#NUM!\*\*.
- Indipendentemente dal segno del numero, un valore viene arrotondato per difetto quando il suo valore si allontana da zero. Se il numero è un multiplo significativo esatto, non viene eseguito alcun arrotondamento.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula seguente accetta i valori della colonna [Total Product Cost] della tabella InternetSales e arrotonda per difetto al multiplo più vicino di 0.1.

DAX

```
= FLOOR(InternetSales[Total Product Cost],.1)
```

La tabella seguente mostra i risultati previsti per alcuni valori di esempio:

Valori	Risultato previsto
10.8423	10.8
8.0373	8
2.9733	2,9

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# GCD

Articolo • 20/10/2023

Restituisce il massimo comun divisore di due o più numeri interi. Il massimo comun divisore è il più grande numero intero che divide perfettamente sia number1 che number2, senza resto.

## Sintassi

DAX

`GCD(number1, [number2], ...)`

## Parametri

Termine	Definizione
number1,	Number1 è obbligatorio, i numeri successivi sono facoltativi. Valori da 1 a 255.
number2, ...	Se un qualsiasi valore non è un numero intero, la parte decimale viene troncata.

## Valore restituito

Massimo comun divisore di due o più numeri interi.

## Osservazioni:

- Se uno degli argomenti non è numerico, GCD restituisce il valore di errore #VALUE! .
- Se uno degli argomenti è minore di zero, GCD restituisce il valore di errore #NUM! .
- Qualunque valore è perfettamente divisibile per uno.
- Un numero primo è perfettamente divisibile solo per se stesso e per uno.
- Se un parametro di GCD è  $>=2^{53}$ , GCD restituisce il valore di errore #NUM! .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Formula	Description	Result
= GCD(5, 2)	Massimo comun divisore di 5 e 2.	1
= GCD(24, 36)	Massimo comun divisore di 24 e 36.	12
= GCD(7, 1)	Massimo comun divisore di 7 e 1.	1

## Commenti e suggerimenti

Questa pagina è stata utile?



# INT

Articolo • 20/10/2023

Arrotonda un numero per difetto all'intero più vicino.

## Sintassi

DAX

```
INT(<number>)
```

## Parametri

Termine	Definizione
number	Numero da arrotondare per difetto a un intero.

## Valore restituito

Numero intero.

## Osservazioni:

Le funzioni TRUNC e INT sono simili per il fatto che entrambe restituiscono numeri interi. TRUNC rimuove la parte frazionaria del numero. INT arrotonda per difetto al valore intero più vicino in base al valore della parte frazionaria del numero. INT e TRUNC differiscono solo in caso di utilizzo di numeri negativi: `TRUNC(-4.3)` restituisce -4, mentre `INT(-4.3)` restituisce -5 perché -5 è il numero inferiore.

## Esempio

L'espressione seguente arrotonda il valore a 1. Se si usa la funzione ROUND, il risultato sarà 2.

DAX

```
= INT(1.5)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione ROUND](#)

[Funzione ROUNDUP](#)

[Funzione ROUNDDOWN](#)

[Funzione MROUND](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ISO.CEILING

Articolo • 20/10/2023

Arrotonda un numero per eccesso all'intero più vicino o al multiplo più vicino.

## Sintassi

DAX

```
ISO.CEILING(<number>[, <significance>])
```

## Parametri

Termino	Definizione
number	Numero da arrotondare o riferimento a una colonna contenente numeri.
significance	(facoltativo) Multiplo significativo a cui si vuole eseguire l'arrotondamento. Ad esempio per arrotondare al numero intero più vicino, digitare 1. Se l'unità significativa non viene specificata, il numero viene arrotondato per eccesso al numero intero più vicino.

## Valore restituito

Numero dello stesso tipo dell'argomento *number*, arrotondato come specificato.

## Osservazioni:

In DAX sono disponibili due funzioni CEILING, con le differenze seguenti:

- La funzione CEILING emula il comportamento della funzione ARROTONDA.ECESSO in Excel.
- La funzione ISO.CEILING segue il comportamento definito da ISO per la determinazione del valore di arrotondamento per eccesso.

Le due funzioni restituiscono lo stesso valore per i numeri positivi, ma valori diversi per i numeri negativi. Se si usa un multiplo significativo positivo, sia la funzione CEILING sia la funzione ISO.CEILING arrotondano i numeri negativi per eccesso (verso l'infinito positivo). Se si usa un multiplo significativo negativo, la funzione CEILING arrotonda i

numeri negativi per difetto (verso l'infinito negativo), mentre la funzione ISO.CEILING arrotonda i numeri negativi per eccesso (verso l'infinito positivo).

Il tipo restituito è generalmente lo stesso tipo significativo usato come argomento, con le eccezioni seguenti:

- Se il primo argomento è di tipo currency il risultato sarà il tipo currency.
- Se l'argomento facoltativo non è incluso il risultato è di tipo integer.
- Se il tipo di argomento significativo è booleano, il tipo restituito è integer.
- Se il tipo di argomento significativo è di tipo non numerico, il tipo restituito è real.

## Esempio: numeri positivi

La formula seguente restituisce 4.45. Questo potrebbe essere utile se si vuole evitare di usare unità più piccole nella determinazione dei prezzi. Se il prezzo di un prodotto esistente è fissato a \$4.42, si può usare la funzione ISO.CEILING per arrotondare per eccesso all'unità cinque centesimi più vicina.

DAX

```
= ISO.CEILING(4.42,0.05)
```

## Esempio: numeri negativi

La formula seguente restituisce il valore di arrotondamento per eccesso ISO -4.40.

DAX

```
= ISO.CEILING(-4.42,0.05)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione FLOOR](#)

[Funzione CEILING](#)

[Funzione ROUNDUP](#)

# Commenti e suggerimenti

Questa pagina è stata utile?



# LCM

Articolo • 20/10/2023

Restituisce il minimo comune multiplo dei numeri interi. Il minimo comune multiplo è il numero intero positivo più piccolo, multiplo di tutti gli argomenti number1, number2 e così via, rappresentati da numeri interi. Usare LCM per addizionare frazioni con denominatori diversi.

## Sintassi

DAX

`LCM(number1, [number2], ...)`

## Parametri

Termine	Definizione
number1,	Number1 è obbligatorio, i numeri successivi sono facoltativi. Da 1 a 255 valori di
number2,...	cui si vuole calcolare il minimo comune multiplo. Se il valore non è un numero intero, la parte decimale viene troncata.

## Valore restituito

Restituisce il minimo comune multiplo dei numeri interi.

## Osservazioni:

- Se uno degli argomenti non è numerico, LCM restituisce il valore di errore #VALUE! .
- Se uno degli argomenti è minore di zero, LCM restituisce il valore di errore #NUM! .
- Se  $LCM(a,b) >= 2^{53}$ , LCM restituisce il valore di errore #NUM! .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Formula	Description	Result
= LCM(5, 2)	Minimo comune multiplo di 5 e 2.	10
= LCM(24, 36)	Minimo comune multiplo di 24 e 36.	72

## Commenti e suggerimenti

Questa pagina è stata utile?



# LN

Articolo • 20/10/2023

Restituisce il logaritmo naturale di un numero. I logaritmi naturali si basano sulla costante e (2.71828182845904).

## Sintassi

DAX

```
LN(<number>)
```

## Parametri

Termine	Definizione
number	Numero positivo di cui calcolare il logaritmo naturale.

## Valore restituito

Numero decimale.

## Osservazioni:

LN è l'inverso della funzione EXP.

## Esempio

L'esempio seguente restituisce il logaritmo naturale del numero nella colonna [Values].

DAX

```
= LN([Values])
```

## Vedi anche

## Commenti e suggerimenti

Questa pagina è stata utile?



# LOG

Articolo • 20/10/2023

Restituisce il logaritmo di un numero nella base specificata.

## Sintassi

DAX

```
LOG(<number>,<base>)
```

## Parametri

Termine	Definizione
number	Numero positivo di cui calcolare il logaritmo.
base	Base del logaritmo. Se omessa, viene usata la base 10.

## Valore restituito

Numero decimale.

## Osservazioni:

Se il valore è troppo grande per essere visualizzato, è possibile che venga generato un errore.

La funzione LOG10 è simile, ma restituisce sempre il logaritmo comune, cioè quello in base 10.

## Esempio

Le formule seguenti restituiscono lo stesso risultato, 2.

DAX

```
= LOG(100,10)  
= LOG(100)
```

```
= LOG10(100)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione EXP](#)

[Funzione LOG](#)

[Funzione LOG](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# LOG10

Articolo • 20/10/2023

Restituisce il logaritmo in base 10 di un numero.

## Sintassi

DAX

```
LOG10(<number>)
```

## Parametri

Termine	Definizione
number	Numero positivo di cui calcolare il logaritmo in base 10.

## Valore restituito

Numero decimale.

## Osservazioni:

La funzione LOG consente di modificare la base del logaritmo, anziché usare la base 10.

## Esempio

Le formule seguenti restituiscono lo stesso risultato, 2:

DAX

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione EXP](#)

[Funzione LOG](#)

[Funzione LOG](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**

 **No**

# MOD

Articolo • 20/10/2023

Restituisce il resto dopo la divisione di un numero per un divisore. Il risultato ha sempre lo stesso segno del divisore.

## Sintassi

DAX

```
MOD(<number>, <divisor>)
```

## Parametri

Termine	Definizione
number	Numero di cui calcolare il resto dopo la divisione.
divisor	Numero per cui dividere il dividendo.

## Valore restituito

Numero intero.

## Osservazioni:

- Se il divisore è 0 (zero), MOD restituisce un errore. Non è possibile dividere per 0.
- La funzione MOD può essere espressa in termini di funzione INT:  $MOD(n, d) = n - d * INT(n/d)$

## Esempio 1

La formula seguente restituisce 1, il resto di 3 diviso per 2.

DAX

```
= MOD(3,2)
```

## Esempio 2

La formula seguente restituisce -1, il resto di 3 diviso per 2. Il segno corrisponde sempre a quello del divisore.

DAX

```
= MOD(-3, -2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione ROUND](#)

[Funzione ROUNDUP](#)

[Funzione ROUNDDOWN](#)

[Funzione MROUND](#)

[Funzione INT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MROUND

Articolo • 20/10/2023

Restituisce un numero arrotondato al multiplo desiderato.

## Sintassi

DAX

```
MROUND(<number>, <multiple>)
```

## Parametri

Termine	Definizione
number	Numero da arrotondare.
multipli	Multiplo significativo a cui si vuole eseguire l'arrotondamento del numero.

## Valore restituito

Numero decimale.

## Osservazioni:

MROUND arrotonda il valore assoluto del numero per eccesso se il resto della divisione di **number** per il valore **multiple** specificato è maggiore o uguale alla metà del valore di **multiple**.

## Esempio: posizioni decimali

L'espressione seguente arrotonda 1.3 al multiplo più vicino di 0.2. Il risultato previsto è 1.4.

DAX

```
= MROUND(1.3,0.2)
```

## Esempio: numeri negativi

L'espressione seguente arrotonda -10 al multiplo più vicino di -3. Il risultato previsto è -9.

DAX

```
= MROUND(-10, -3)
```

## Esempio: errore

L'espressione seguente restituisce un errore perché i numeri hanno segni diversi.

DAX

```
= MROUND(5, -2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzione ROUND](#)

[Funzione ROUNDUP](#)

[Funzione ROUNDDOWN](#)

[Funzione MROUND](#)

[Funzione INT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# ODD

Articolo • 20/10/2023

Restituisce il numero arrotondato per eccesso al numero intero dispari più vicino.

## Sintassi

DAX
<code>ODD(number)</code>

## Parametri

Termine	Definizione
number	Required. Il valore da arrotondare.

## Valore restituito

Restituisce il numero arrotondato per eccesso al numero intero dispari più vicino.

## Osservazioni:

- Se number non è un valore numerico, ODD restituisce il valore di errore #VALUE! .
- Indipendentemente dal segno del numero, un valore viene arrotondato per eccesso quando il suo valore si allontana da zero. Se number è un numero intero dispari, non viene eseguito alcun arrotondamento.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= ODD(1.5)	Arrotonda 1.5 per eccesso all'intero dispari più vicino.	3
= ODD(3)	Arrotonda 3 per eccesso all'intero dispari più vicino.	3

Formula	Description	Result
= ODD(2)	Arrotonda 2 per eccesso all'intero dispari più vicino.	3
= ODD(-1)	Arrotonda -1 per eccesso all'intero dispari più vicino.	-1
= ODD(-2)	Arrotonda il valore assoluto di -2 per eccesso all'intero dispari più vicino.	-3

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# PI

Articolo • 20/10/2023

Restituisce il valore di pi greco, 3,14159265358979, arrotondato a 15 cifre.

## Sintassi

DAX

```
PI()
```

## Valore restituito

Numero decimale con il valore di pi greco, 3,14159265358979, arrotondato a 15 cifre.

## Osservazioni:

Pi greco è una costante matematica. In DAX pi greco è rappresentato come numero reale arrotondato a 15 cifre, analogamente a quanto avviene in Excel.

## Esempio

La formula seguente calcola l'area di un cerchio in base al raggio indicato nella colonna `[Radius]`.

DAX

```
= PI()*([Radius]*2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

## Commenti e suggerimenti

Questo articolo è stato utile?

 Sì

 No

# POWER

Articolo • 20/10/2023

Restituisce il risultato di un numero elevato a potenza.

## Sintassi

DAX

```
POWER(<number>, <power>)
```

## Parametri

Termine	Definizione
number	Numero di base, che può essere qualsiasi numero reale.
power	L'esponente a cui elevare il numero base.

## Valore restituito

Numero decimale.

## Esempio

L'esempio seguente restituisce 25.

DAX

```
= POWER(5,2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# QUOTIENT

Articolo • 20/10/2023

Esegue una divisione e restituisce solo la parte intera del risultato della divisione. Usare questa funzione quando si vuole eliminare il resto della divisione.

## Sintassi

DAX

`QUOTIENT(<numerator>, <denominator>)`

## Parametri

Termine	Definizione
numerator	Dividendo o numero da dividere.
denominator	Divisore o numero per cui dividere.

## Valore restituito

Numero intero.

## Osservazioni:

- Se l'argomento non è di tipo numerico, QUOTIENT restituisce il valore di errore **#VALUE!**.
- È possibile usare un riferimento a una colonna invece di un valore letterale per gli argomenti. Se tuttavia la colonna a cui si fa riferimento contiene uno 0 (zero), viene restituito un errore per l'intera colonna di valori.

## Esempio

Le formule seguenti restituiscono lo stesso risultato, 2.

DAX

```
= QUOTIENT(5,2)
```

DAX

```
= QUOTIENT(10/2,2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# RADIANS

Articolo • 20/10/2023

Converte i gradi in radianti.

## Sintassi

DAX
<code>RADIANS(angle)</code>

## Parametri

Termine	Definizione
angle	Required. Angolo in gradi da convertire.

## Esempio

Formula	Description	Result
= RADIANS(270)	270 gradi come radianti (4.712389 o $3\pi/2$ radianti)	4.712389

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# RAND

Articolo • 20/10/2023

Restituisce un numero casuale uniformemente distribuito e superiore o uguale a 0 e minore di 1. Il numero restituito cambia ogni volta che la cella contenente questa funzione viene ricalcolata.

## Sintassi

DAX

`RAND()`

## Valore restituito

Numero decimale.

## Osservazioni:

- Il ricalcolo dipende da vari fattori, tra cui l'impostazione del modello sulla modalità di ricalcolo **Manuale** o **Automatico** e l'aggiornamento dei dati.
- RAND e altre funzioni volatili che non hanno valori fissi non vengono sempre ricalcolate. L'esecuzione di una query o di un filtro, ad esempio, in genere non comporta una nuova valutazione di tali funzioni. I risultati di queste funzioni vengono tuttavia ricalcolati quando viene ricalcolata l'intera colonna. Queste situazioni includono l'aggiornamento da un'origine dati esterna o la modifica manuale dei dati che causa una nuova valutazione delle formule contenenti queste funzioni.
- La funzione RAND viene sempre ricalcolata se usata nella definizione di una misura.
- La funzione RAND non può restituire un risultato uguale a zero per impedire errori quale la divisione per zero.

## Esempi

Per generare un numero reale casuale tra due numeri diversi, usare:

DAX

```
= RAND()*(b-a)+a
```

Per generare un numero casuale maggiore di 0 e minore di 1:

DAX

```
= RAND()
```

Per generare un numero casuale maggiore di 0 e minore di 100

DAX

```
= RAND()*100
```

Per generare un numero intero casuale maggiore di 0 e minore di 100

DAX

```
INT(RAND()*100)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# RANDBETWEEN

Articolo • 20/10/2023

Restituisce un numero casuale nell'intervallo tra due numeri specificati.

## Sintassi

DAX

```
RANDBETWEEN(<bottom>,<top>)
```

## Parametri

Termine	Definizione
Ultimo	Intero più piccolo restituito dalla funzione.
In alto	Intero più grande restituito dalla funzione.

## Valore restituito

Numero intero.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente restituisce un numero casuale compreso tra 1 e 10.

DAX

```
= RANDBETWEEN(1,10)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ROUND

Articolo • 20/10/2023

Arrotonda un numero al numero di cifre specificato.

## Sintassi

DAX

```
ROUND(<number>, <num_digits>)
```

## Parametri

Termine	Definizione
number	Numero da arrotondare.
num_digits	Numero di cifre a cui arrotondare. Un valore negativo arrotonda le cifre a sinistra del separatore decimale. Il valore zero viene arrotondato all'intero più vicino.

## Valore restituito

Numero decimale.

## Osservazioni:

- Se **num\_digits** è maggiore di 0 (zero), il numero viene arrotondato al numero specificato di posizioni decimali.
- Se **num\_digits** è pari a 0, il numero viene arrotondato all'intero più vicino.
- Se **num\_digits** è minore di 0, il numero viene arrotondato a sinistra del separatore decimale.
- Funzioni correlate
  - Per eseguire sempre l'arrotondamento per eccesso (lontano da zero), usare la funzione ROUNDUP.
  - Per eseguire sempre l'arrotondamento per difetto (verso lo zero), usare la funzione ROUNDDOWN.

- Per arrotondare un numero a un multiplo specifico (ad esempio, per eseguire l'arrotondamento al multiplo di 0.5 più vicino), usare la funzione MROUND.
- Usare le funzioni TRUNC e INT per ottenere la parte intera del numero.

## Esempio 1

La formula seguente arrotonda per eccesso 2.15, a una posizione decimale. Il risultato previsto è 2.2.

DAX

```
= ROUND(2.15,1)
```

## Esempio 2

La formula seguente arrotonda 21.5 a una posizione decimale a sinistra del separatore decimale. Il risultato previsto è 20.

DAX

```
= ROUND(21.5,-1)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

[TRUNC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ROUNDDOWN

Articolo • 20/10/2023

Arrotonda per difetto il valore assoluto di un numero.

## Sintassi

DAX

```
ROUNDDOWN(<number>, <num_digits>)
```

## Parametri

Termine	Definizione
number	Numero reale da arrotondare per difetto.
num_digits	Numero di cifre a cui arrotondare. Un valore negativo arrotonda a sinistra del separatore decimale, mentre zero arrotonda all'intero più vicino.

## Valore restituito

Numero decimale.

## Osservazioni:

- Se **num\_digits** è maggiore di 0 (zero), il valore in **number** viene arrotondato per difetto al numero specificato di posizioni decimali.
- Se **num\_digits** è pari a 0, il valore in **number** viene arrotondato per difetto all'intero più vicino.
- Se **num\_digits** è minore di 0, il valore in **number** viene arrotondato per difetto a sinistra del separatore decimale.
- ROUNDDOWN si comporta come ROUND, ma arrotonda un numero sempre per difetto. Anche la funzione INT arrotonda per difetto, ma con INT il risultato è sempre un numero intero, mentre con ROUNDDOWN è possibile controllare la precisione del risultato.

## Esempio 1

L'esempio seguente arrotonda per difetto 3.14159 a tre posizioni decimali. Il risultato previsto è 3.141.

DAX

```
= ROUNDOWN(3.14159,3)
```

## Esempio 2

L'esempio seguente arrotonda per difetto il valore 31415.92654 a 2 posizioni decimali a sinistra del separatore decimale. Il risultato previsto è 31400.

DAX

```
= ROUNDOWN(31415.92654, -2)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ROUNDUP

Articolo • 20/10/2023

Arrotonda per eccesso il valore assoluto di un numero.

## Sintassi

DAX

```
ROUNDUP(<number>, <num_digits>)
```

## Parametri

Termine	Definizione
number	Numero reale da arrotondare per eccesso.
num_digits	Numero di cifre a cui arrotondare. Un valore negativo per <b>num_digits</b> esegue l'arrotondamento a sinistra del separatore decimale. Se <b>num_digits</b> è pari a zero o viene omesso, <b>number</b> viene arrotondato all'intero più vicino.

## Valore restituito

Numero decimale.

## Osservazioni:

- Se **num\_digits** è maggiore di 0 (zero), il numero viene arrotondato per eccesso al numero specificato di posizioni decimali.
- Se **num\_digits** è pari a 0, il numero viene arrotondato per eccesso all'intero più vicino.
- Se **num\_digits** è minore di 0, il numero viene arrotondato per eccesso a sinistra del separatore decimale.
- ROUNDUP si comporta come ROUND, ma arrotonda un numero sempre per eccesso.

# Esempio

La formula seguente arrotonda pi greco a quattro posizioni decimali. Il risultato previsto è 3.1416.

```
DAX
```

```
= ROUNDUP(PI(),4)
```

## Esempio: decimali come secondo argomento

La formula seguente arrotonda 1.3 al multiplo più vicino di 0.2. Il risultato previsto è 2.

```
DAX
```

```
= ROUNDUP(1.3,0.2)
```

## Esempio: numero negativo come secondo argomento

La formula seguente arrotonda il valore nella colonna **FreightCost** con i risultati previsti indicati nella tabella seguente:

```
DAX
```

```
= ROUNDUP([Values],-1)
```

Quando **num\_digits** è minore di zero, il numero di posizioni a sinistra del separatore decimale aumenta del valore specificato.

FreightCost	Risultato previsto
13.25	20
2.45	10
25.56	30
1.34	10
345.01	350

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SIGN

Articolo • 20/10/2023

Determina il segno di un numero, del risultato di un calcolo o di un valore in una colonna. La funzione restituisce 1 se il numero è positivo, 0 (zero) se il numero è zero o -1 se il numero è negativo.

## Sintassi

DAX

```
SIGN(<number>)
```

## Parametri

Termine	Definizione
number	qualsiasi numero reale, una colonna contenente numeri o un'espressione che restituisce un numero.

## Valore restituito

Numero intero. I possibili valori restituiti sono 1, 0 e -1.

Valore restituito	Descrizione
1	Il numero è positivo
0	Il numero è zero
-1	Il numero è negativo

## Esempio

La formula seguente restituisce il segno del risultato dell'espressione che calcola il prezzo di vendita meno il costo.

DAX

```
= SIGN( ([Sale Price] - [Cost]) )
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SIN

Articolo • 20/10/2023

Restituisce il seno dell'angolo specificato.

## Sintassi

DAX

`SIN(number)`

## Parametri

Termine	Definizione
number	Required. Angolo in radianti per cui calcolare il seno.

## Valore restituito

Restituisce il seno dell'angolo specificato.

## Osservazioni:

Se un argomento è in gradi, moltiplicarlo per PI()/180 o usare la funzione RADIANS per convertirlo in radianti.

## Esempio

Formula	Description	Result
= SIN(PI())	Seno di pi greco in radianti (circa 0).	0.0
= SIN(PI()/2)	Seno di pi greco/2 in radianti.	1.0
= SIN(30*PI()/180)	Seno di 30 gradi.	0.5
= SIN(RADIANS(30))	Seno di 30 gradi.	0.5

# Commenti e suggerimenti

Questa pagina è stata utile?



# SINH

Articolo • 20/10/2023

Restituisce il seno iperbolico di un numero.

## Sintassi

DAX

```
SINH(number)
```

## Parametri

Termine	Definizione
number	Required. Qualsiasi numero reale.

## Valore restituito

Restituisce il seno iperbolico di un numero.

## Osservazioni:

- La formula per il seno iperbolico è:

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Probabilità di ottenere un risultato inferiore a 1.03 secondi.

DAX

```
= 2.868*SINH(0.0342\*1.03)
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# SQRT

Articolo • 20/10/2023

Restituisce la radice quadrata di un numero.

## Sintassi

DAX

```
SQRT(<number>)
```

## Parametri

Termine	Definizione
number	Il numero di cui si vuole ottenere la radice quadrata, una colonna contenente numeri o un'espressione che restituisce un numero.

## Valore restituito

Numero decimale.

## Osservazioni:

Se il numero è negativo, la funzione SQRT restituisce un errore.

## Esempio

La formula seguente:

DAX

```
= SQRT(25)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# SQRTPI

Articolo • 20/10/2023

Restituisce la radice quadrata di (numero \* pi).

## Sintassi

DAX
<code>SQRTPI(number)</code>

## Parametri

Termine	Definizione
number	Required. Numero per il quale viene moltiplicato pi.

## Valore restituito

Restituisce la radice quadrata di (numero \* pi).

## Esempio

Formula	Description	Result
= SQRTPI(1)	Radice quadrata di pi greco.	1.772454
= SQRTPI(2)	Radice quadrata di 2 * pi greco.	2.506628

## Commenti e suggerimenti

Questo articolo è stato utile?

 Sì  No

# TAN

Articolo • 20/10/2023

Restituisce la tangente dell'angolo specificato.

## Sintassi

DAX

`TAN(number)`

## Parametri

Termine	Definizione
number	Required. Angolo in radianti per cui calcolare la tangente.

## Valore restituito

Restituisce la tangente dell'angolo specificato.

## Osservazioni:

Se l'argomento è in gradi, moltiplicarlo per PI()/180 o usare la funzione RADIANS per convertirlo in radianti.

## Esempio

Formula	Description	Result
= TAN(0.785)	Tangente di 0.785 in radianti (0.99920)	0.99920
= TAN(45*PI()/180)	Tangente di 45 in gradi (1)	1
= TAN(RADIANS(45))	Tangente di 45 in gradi (1)	1

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# TANH

Articolo • 20/10/2023

Restituisce la tangente iperbolica di un numero.

## Sintassi

DAX

`TANH(number)`

## Parametri

Termine	Definizione
number	Required. Qualsiasi numero reale.

## Valore restituito

Restituisce la tangente iperbolica di un numero.

## Osservazioni:

- La formula per la tangente iperbolica è:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= TANH(-2)	Tangente iperbolica di -2 (-0.96403)	-0.964028
= TANH(0)	Tangente iperbolica di 0 (0)	0
= TANH(0.5)	Tangente iperbolica di -0.5 (-0.462117)	0.462117

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# TRUNC

Articolo • 20/10/2023

Elimina la parte decimale o frazionaria di un numero.

## Sintassi

DAX

```
TRUNC(<number>,<num_digits>)
```

## Parametri

Termine	Definizione
number	Numero di cui eliminare la parte decimale.
num_digits	Numero che specifica la precisione del troncamento; 0 (zero) se omesso

## Valore restituito

Numero intero.

## Osservazioni:

Le funzioni TRUNC e INT sono simili per il fatto che entrambe restituiscono numeri interi. TRUNC rimuove la parte frazionaria del numero. INT arrotonda per difetto al valore intero più vicino in base al valore della parte frazionaria del numero. INT e TRUNC differiscono solo in caso di utilizzo di numeri negativi: `TRUNC(-4.3)` restituisce -4, mentre `INT(-4.3)` restituisce -5 perché -5 è il numero più piccolo.

## Esempio 1

La formula seguente restituisce 3, la parte intera di pi greco.

DAX

```
= TRUNC(PI())
```

## Esempio 2

La formula seguente restituisce -8, la parte intera di -8.9.

DAX

```
= TRUNC(-8.9)
```

## Vedi anche

[Funzioni matematiche e trigonometriche](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Altre funzioni

Articolo • 20/10/2023

Queste funzioni eseguono azioni uniche che non possono essere definite da alcuna categoria.

## In questa categoria

Funzione	Descrizione
BLANK	Restituisce un valore vuoto.
ERROR	Genera un errore con un messaggio di errore.
EVALUATEANDLOG	Restituisce il valore del primo argomento e lo registra in un evento del profiler di valutazione DAX.
TOCSV	Restituisce una tabella come stringa in formato CSV.
TOJSON	Restituisce una tabella come stringa in formato JSON.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# BLANK

Articolo • 20/10/2023

Restituisce un valore vuoto.

## Sintassi

DAX

```
BLANK()
```

## Valore restituito

Valore vuoto.

## Osservazioni:

- I valori vuoti non sono equivalenti ai valori Null. DAX usa i valori vuoti sia per i valori Null dei database che per le celle vuote in Excel.
- Alcune funzioni DAX trattano le celle vuote in modo diverso rispetto a Microsoft Excel. I valori vuoti e le stringhe vuote ("") non sempre sono equivalenti, ma alcune operazioni potrebbero trattarli come tali.

## Esempio

L'esempio seguente illustra come usare i valori vuoti nelle formule. La formula calcola il rapporto di vendite tra i rivenditori e i canali Internet. Tuttavia, prima di tentare di calcolare il rapporto, è necessario verificare la presenza di valori zero nel denominatore. Se il denominatore è pari a zero, verrà restituito un valore vuoto. In caso contrario verrà calcolato il rapporto.

DAX

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,  
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_USD]) )
```

La tabella mostra i risultati previsti quando questa formula viene usata per creare una visualizzazione in tabella.

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Grand Total</b>
2005		2.65		2.89
2006		3.33		4,03
2007	1.04	2.92	6,63	3.51
2008	0,41	1.53	2.00	1,71
Grand Total	0.83	2.51	5,45	2.94

Nell'origine dati originale, la colonna valutata dalla funzione BLANK potrebbe includere testo, stringhe vuote o valori Null. Se l'origine dati originale è un database di SQL Server, i valori Null e le stringhe vuote sono tipi di dati diversi. Per questa operazione viene tuttavia eseguito un cast di tipo implicito e in DAX questi tipi di dati vengono trattati nello stesso modo.

## Vedi anche

[Funzioni di testo](#)

[Funzione ISBLANK](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ERROR

Articolo • 20/10/2023

Genera un errore con un messaggio di errore.

## Sintassi

DAX

```
ERROR(<text>)
```

## Parametri

Termine	Definizione
testo	Stringa di testo contenente un messaggio di errore.

## Valore restituito

None

## Osservazioni:

- La funzione ERROR può essere inserita in un'espressione DAX ovunque sia previsto un valore scalare.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

La query DAX seguente:

DAX

```
DEFINE  
MEASURE DimProduct[Measure] =  
    IF(  
        SELECTEDVALUE(DimProduct[Color]) = "Red",  
        ERROR("red color encountered"),
```

```

    SELECTEDVALUE(DimProduct[Color])
)
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]

```

Ha esito negativo e genera un messaggio di errore per segnalare che è stato rilevato il colore rosso.

## Esempio 2

La query DAX seguente:

```

DAX

DEFINE
MEASURE DimProduct[Measure] =
IF(
    SELECTEDVALUE(DimProduct[Color]) = "Magenta",
    ERROR("magenta color encountered"),
    SELECTEDVALUE(DimProduct[Color])
)
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]

```

Restituisce la tabella seguente:

DimProduct[Color]	[Measure]
Nero	Nero
Blu	Blu
Grigio	Grigio
Multi	Multi
N/D	N/D
Rosso	Rosso
Medio	Medio
Argento\Nero	Argento\Nero
Bianco	Bianco
Giallo	Giallo

Poiché Magenta non è uno dei colori del prodotto, la funzione ERROR non viene eseguita.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EVALUATEANDLOG

Articolo • 20/10/2023

Restituisce il valore del primo argomento e lo registra in un evento del profiler di valutazione DAX. Questa funzione è completamente funzionale solo in Power BI Desktop. Funge da semplice funzione pass-through in altri ambienti.

## Sintassi

DAX
<pre>EVALUATEANDLOG(&lt;Value&gt;, [Label], [MaxRows])</pre>

## Parametri

Termine	Definizione
Valore	qualsiasi espressione scalare o espressione di tabella da valutare e registrare.
Etichetta	(Facoltativo) Stringa costante inclusa sia nel testo json che nella colonna Etichetta dell'evento DAX Evaluation Log che può essere usata per identificare facilmente un'istanza della chiamata di funzione.
MaxRows	(Facoltativo) Numero massimo di righe nel testo json dell'evento DAX Evaluation Log quando il primo argomento è un'espressione di tabella. Default is 10.

## Valore restituito

Valore del primo argomento.

La struttura JSON registrata in un evento del profiler di valutazione DAX include:

- "expression" è la versione di testo del primo argomento.
- "label" è il parametro Label quando specificato nell'espressione.
- "input" è un elenco di colonne nel contesto di valutazione che influisce sui valori del primo argomento.
- "outputs" è un elenco di una singola colonna [Value] quando il primo argomento è un'espressione scalare e un elenco di colonne di output quando il primo argomento è un'espressione di tabella.
- "data" è un elenco di valori di input e valori di output quando il primo argomento è un'espressione scalare e un elenco di valori di input e le righe di output

corrispondenti quando il primo argomento è un'espressione di tabella.

- "rowCount" è il numero di righe quando il primo argomento è un'espressione di tabella. Anche se il numero di righe nell'output json viene troncato dal parametro MaxRows, rowCount è il numero reale di righe senza troncamento.

## Osservazioni:

- Gli eventi di traccia possono essere acquisiti usando [SQL Server Profiler](#) e lo strumento di [output di debug DAX](#) open source.
- Questa funzione può essere usata con quasi qualsiasi sottoespressione in un'espressione DAX e l'intera espressione sarà comunque valida.
- Quando il primo argomento viene valutato più volte in una singola query, la funzione genera un singolo evento DAX Evaluation Log che contiene sia i valori di input che i valori di output corrispondenti.
- Quando si specifica il parametro label, il relativo valore viene restituito sia nell'output json che nella colonna Label dell'evento DAX Evaluation Log.
- Se il primo argomento è un'espressione di tabella, nell'evento DAX Evaluation Log vengono visualizzate solo le prime righe MaxRows.
- In alcuni casi, questa funzione non viene eseguita a causa di ottimizzazioni.
- Se l'evento DAX Evaluation Log è maggiore di un milione di caratteri, viene troncato per mantenere la struttura JSON corretta.

## Esempio 1

La query DAX seguente:

DAX

```
evaluate
SUMMARIZE(
    EVALUATEANDLOG(FILTER(Sales, [ProductKey] = 528)),
    Sales[SalesTerritoryKey],
    "sum",
    sum(Sales[Sales Amount])
)
```

Restituisce l'evento DAX Evaluation Log seguente:

JSON

```
{  
    "expression": "FILTER(Sales, [ProductKey] = 528)",  
    "inputs": [],  
    "outputs": ["'Sales'[SalesOrderLineKey]", "'Sales'[ResellerKey]",  
    "'Sales'[CustomerKey]", "'Sales'[ProductKey]", "'Sales'[OrderDateKey]",  
    "'Sales'[DueDateKey]", "'Sales'[ShipDateKey]", "'Sales'[SalesTerritoryKey]",  
    "'Sales'[Order Quantity]", "'Sales'[Unit Price]", "'Sales'[Extended  
    Amount]", "'Sales'[Product Standard Cost]", "'Sales'[Total Product Cost]",  
    "'Sales'[Sales Amount]", "'Sales'[Unit Price Discount Pct]"],  
    "data": [  
        {  
            "input": [],  
            "rowCount": 3095,  
            "output": [  
                [52174001, -1, 23785, 528, 20190707, 20190717, 20190714, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52173001, -1, 26278, 528, 20190707, 20190717, 20190714, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52082001, -1, 23831, 528, 20190705, 20190715, 20190712, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52054002, -1, 11207, 528, 20190704, 20190714, 20190711, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [52036001, -1, 25337, 528, 20190704, 20190714, 20190711, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51939002, -1, 23670, 528, 20190702, 20190712, 20190709, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51911002, -1, 11746, 528, 20190701, 20190711, 20190708, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51379003, -1, 13745, 528, 20190612, 20190622, 20190619, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51264002, -1, 11282, 528, 20190605, 20190615, 20190612, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],  
                [51184003, -1, 11263, 528, 20190531, 20190610, 20190607, 1,  
                1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0]  
            ]  
        }  
    ]  
}
```

## Esempio 2

La query DAX seguente con un argomento scalare e attributi variabili:

DAX

```
evaluate  
SELECTCOLUMNS(  
    TOPN(5, Customer),  
    [Customer],
```

```
"Customer",
EVALUATEANDLOG([Customer] & ", " & [Country-Region], "customerLog")
)
```

Restituisce l'evento DAX Evaluation Log seguente:

JSON

```
{
    "expression": "[Customer] & \", \" & [Country-Region]",
    "label": "customerLog",
    "inputs": ["'Customer'[Customer]", "'Customer'[Country-Region]"],
    "data": [
        {
            "input": ["Russell Xie", "United States"],
            "output": "Russell Xie, United States"
        },
        {
            "input": ["Savannah Baker", "United States"],
            "output": "Savannah Baker, United States"
        },
        {
            "input": ["Maurice Tang", "United States"],
            "output": "Maurice Tang, United States"
        },
        {
            "input": ["Emily Wood", "United States"],
            "output": "Emily Wood, United States"
        },
        {
            "input": ["Meghan Hernandez", "United States"],
            "output": "Meghan Hernandez, United States"
        }
    ]
}
```

## Vedi anche

[TOCSV](#)

[TOJSON](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# TOCSV

Articolo • 20/10/2023

Restituisce una tabella come stringa in formato CSV.

## Sintassi

DAX

```
TOCSV(<Table>, [MaxRows], [Delimiter], [IncludeHeaders])
```

## Parametri

Termine	Definizione
Tabella	La tabella da convertire in CSV.
MaxRows	(Facoltativo) Numero massimo di righe da convertire. Il valore predefinito è 10 righe.
Delimitatore	(Facoltativo) Un delimitatore di colonna. Il valore predefinito è la virgola ",".
IncludeHeaders	(Facoltativo) Specifica un'intestazione con nome di colonna come prima riga. L'impostazione predefinita è True.

## Valore restituito

Una stringa con rappresentazione CSV della tabella.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
{TOCSV(DimSalesTerritory)}
```

Restituisce:

```
'DimSalesTerritory'[SalesTerritoryKey], 'DimSalesTerritory'[SalesTerritoryAlternateKey], 'DimSalesTerritory'[SalesTerritoryRegion], 'DimSalesTerritory'[SalesTerritoryCountry], 'DimSalesTerritory'[SalesTerritoryGroup]
1,1,Northwest,United States,North America
2,2,Northeast,United States,North America
3,3,Central,United States,North America
4,4,Southwest,United States,North America
5,5,Southeast,United States,North America
6,6,Canada,Canada,North America
7,7,France,France,Europe
8,8,Germany,Germany,Europe
9,9,Australia,Australia,Pacific
10,10,United Kingdom,United Kingdom,Europe
```

## Vedi anche

[TOJSON](#)

[EVALUATEANDLOG](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# TOJSON

Articolo • 20/10/2023

Restituisce una tabella come stringa usando il formato JSON.

## Sintassi

DAX

```
TOJSON(<Table>, [MaxRows])
```

## Parametri

Termine	Definizione
Tabella	Tabella da convertire in JSON.
MaxRows	(Facoltativo) Numero massimo di righe da convertire. Il valore predefinito è 10 righe.

## Valore restituito

Stringa con rappresentazione JSON della tabella. La rappresentazione contiene i nomi di colonna come "header", il conteggio delle righe come "rowCount" e i valori come "data".

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
{TOJSON(DimSalesTerritory)}
```

Restituisce:

JSON

```
{  
"header": ["'DimSalesTerritory'[SalesTerritoryKey]",
```

```
"'DimSalesTerritory'[SalesTerritoryAlternateKey]",  
"'DimSalesTerritory'[SalesTerritoryRegion]",  
"'DimSalesTerritory'[SalesTerritoryCountry]",  
"'DimSalesTerritory'[SalesTerritoryGroup]" ],<br>  
"rowCount": 11,  
"data": [  
[1, 1, "Northwest", "United States", "North America"],  
[2, 2, "Northeast", "United States", "North America"],  
[3, 3, "Central", "United States", "North America"],  
[4, 4, "Southwest", "United States", "North America"],  
[5, 5, "Southeast", "United States", "North America"],  
[6, 6, "Canada", "Canada", "North America"],  
[7, 7, "France", "France", "Europe"],  
[8, 8, "Germany", "Germany", "Europe"],  
[9, 9, "Australia", "Australia", "Pacific"],  
[10, 10, "United Kingdom", "United Kingdom", "Europe"]  
]  
}
```

## Vedi anche

[TOCSV](#)

[EVALUATEANDLOG](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni padre e figlio

Articolo • 20/10/2023

Queste funzioni gestiscono i dati presentati come gerarchie padre/figlio. Per altre informazioni, vedere [Informazioni sulle funzioni per le gerarchie padre-figlio in DAX](#).

## In questa categoria

Funzione	Descrizione
PATH	Restituisce una stringa di testo delimitata con gli identificatori di tutti gli elementi padre dell'identificatore corrente.
PATHCONTAINS	Restituisce TRUE se l'elemento <i>item</i> specificato è presente nel <i>path</i> specificato.
PATHITEM	Restituisce l'elemento in corrispondenza del parametro <i>position</i> specificato da una stringa risultante dalla valutazione di una funzione PATH.
PATHITEMREVERSE	Restituisce l'elemento in corrispondenza del parametro <i>position</i> specificato da una stringa risultante dalla valutazione di una funzione PATH.
PATHLENGTH	Restituisce il numero di elementi padre relativi all'elemento specificato in un determinato risultato PATH, incluso il risultato stesso.

## Commenti e suggerimenti

Questa pagina è stata utile?



# Informazioni sulle funzioni per le gerarchie padre-figlio in DAX

Articolo • 20/10/2023

In DAX sono disponibili cinque funzioni che consentono agli utenti di gestire i dati presentati come gerarchia padre-figlio nei modelli. Grazie a queste funzioni un utente può ottenere l'intera derivazione di elementi padre di una riga, ovvero determinare quanti livelli ha la derivazione del primo elemento padre, qual è l'elemento padre n livelli sopra la riga corrente, qual è il discendente n dall'inizio della gerarchia di righe corrente e se un determinato elemento padre è un elemento padre nella gerarchia di righe corrente.

## Funzioni padre-figlio in DAX

Nella tabella seguente è contenuta una gerarchia padre-figlio nelle colonne **EmployeeKey** e **ParentEmployeeKey** utilizzata in tutti gli esempi di funzioni.

EmployeeKey	ParentEmployeeKey
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

Nella tabella precedente è possibile notare che per il dipendente 112 non sono definiti elementi padre, il dipendente 14 ha il dipendente 112 come responsabile (ParentEmployeeKey), il dipendente 3 ha il dipendente 14 come responsabile e i dipendenti 11, 13 e 162 hanno il dipendente 3 come responsabile. In base a ciò, è possibile stabilire che il dipendente 112 non ha responsabili a cui rispondere e che è il

responsabile principale di tutti i dipendenti indicati qui. Inoltre, il dipendente 3 risponde al dipendente 14 e i dipendenti 11, 13, 162 rispondono al dipendente 3.

La tabella seguente illustra le funzioni disponibili, una breve descrizione della funzione e un esempio della funzione sugli stessi dati mostrati in precedenza.

**Funzione PATH** - Restituisce un testo delimitato con gli identificatori di tutti gli elementi padre della riga corrente, a partire dal meno recente o da quello superiore fino a quello corrente.

EmployeeKey	ParentEmployeeKey	Percorso
112		112
14	112	112 14
3	14	112 14 3
11	3	112 14 3 11
13	3	112 14 3 13
162	3	112 14 3 162
117	162	112 14 3 162 117
221	162	112 14 3 162 221
81	162	112 14 3 162 81

**Funzione PATHLENGTH** - Restituisce il numero di livelli in un determinato PATH(), a partire dal livello corrente fino al livello padre meno recente o a quello superiore. Nell'esempio seguente la colonna PathLength è definita come '`= PATHLENGTH([Path])`'. L'esempio include tutti i dati dell'esempio Path() che consentono di comprendere il funzionamento di questa funzione.

EmployeeKey	ParentEmployeeKey	Percorso	PathLength
112		112	1
14	112	112 14	2
3	14	112 14 3	3
11	3	112 14 3 11	4
13	3	112 14 3 13	4
162	3	112 14 3 162	4

<b>EmployeeKey</b>	<b>ParentEmployeeKey</b>	<b>Percorso</b>	<b>PathLength</b>
117	162	112 14 3 162 117	5
221	162	112 14 3 162 221	5
81	162	112 14 3 162 81	5

**Funzione PATHITEM** - Restituisce l'elemento nella posizione specificata da un risultato di tipo PATH(), contando da sinistra a destra. Nell'esempio seguente la colonna PathItem, la quarta da sinistra, è definita come '`= PATHITEM([Path], 4)`'. Questo esempio restituisce l'elemento EmployeeKey nella quarta posizione da sinistra della stringa Path, usando gli stessi dati di esempio dell'esempio Path().

<b>EmployeeKey</b>	<b>ParentEmployeeKey</b>	<b>Percorso</b>	<b>PathItem, quarta da sinistra</b>
112		112	
14	112	112 14	
3	14	112 14 3	
11	3	112 14 3 11	11
13	3	112 14 3 13	13
162	3	112 14 3 162	162
117	162	112 14 3 162 117	162
221	162	112 14 3 162 221	162
81	162	112 14 3 162 81	162

**Funzione PATHITEMREVERSE** - Restituisce l'elemento in *position* dal risultato di una funzione di tipo PATH(), contando da destra a sinistra.

Nell'esempio seguente la colonna PathItemReverse, la terza da destra, è definita come '`= PATHITEMREVERSE([Path], 3)`'. Questo esempio restituisce l'elemento EmployeeKey nella terza posizione da destra nella stringa Path, usando gli stessi dati di esempio dell'esempio Path().

<b>EmployeeKey</b>	<b>ParentEmployeeKey</b>	<b>Percorso</b>	<b>PathItemReverse, terza da destra</b>
112		112	
14	112	112 14	
3	14	112 14 3	112

<b>EmployeeKey</b>	<b>ParentEmployeeKey</b>	<b>Percorso</b>	<b>PathItemReverse, terza da destra</b>
11	3	112 14 3 11	14
13	3	112 14 3 13	14
162	3	112 14 3 162	14
117	162	112 14 3 162 117	3
221	162	112 14 3 162 221	3
81	162	112 14 3 162 81	3

**Funzione PATHCONTAINS** - Restituisce **TRUE** se l'elemento *item* specificato è presente nel *path* specificato. Nell'esempio seguente la colonna PathContains, con il dipendente 162, è definita come '`= PATHCONTAINS([Path], "162")`'. Questo esempio restituisce **TRUE** se il percorso specificato contiene il dipendente 162. Questo esempio usa i risultati dell'esempio Path() precedente.

<b>EmployeeKey</b>	<b>ParentEmployeeKey</b>	<b>Percorso</b>	<b>PathContains, dipendente 162</b>
112		112	FALSE
14	112	112 14	FALSE
3	14	112 14 3	FALSE
11	3	112 14 3 11	FALSE
13	3	112 14 3 13	FALSE
162	3	112 14 3 162	TRUE
117	162	112 14 3 162 117	TRUE

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# PATH

Articolo • 20/10/2023

Restituisce una stringa di testo delimitata con gli identificatori di tutti gli elementi padre dell'identificatore corrente, a partire dal meno recente fino a quello corrente.

## Sintassi

DAX

```
PATH(<ID_columnName>, <parent_columnName>)
```

## Parametri

Termine	Definizione
ID_columnName	Nome di una colonna esistente che contiene l'identificatore univoco delle righe nella tabella. Non può essere un'espressione. Il tipo di dati del valore in <i>ID_columnName</i> deve essere testo o un numero intero e deve essere lo stesso tipo di dati della colonna a cui si fa riferimento in <i>parent_columnName</i> .
parent_columnName	Nome di una colonna esistente che contiene l'identificatore univoco dell'elemento padre della riga corrente. Non può essere un'espressione. Il tipo di dati del valore in <i>parent_columnName</i> deve essere testo o un numero intero e deve essere lo stesso tipo di dati del valore in <i>parent_columnName</i> .

## Valore restituito

Stringa di testo delimitata che contiene gli identificatori di tutti gli elementi padre dell'identificatore corrente.

## Osservazioni:

- Questa funzione viene usata in tabelle che hanno una sorta di gerarchia interna per restituire gli elementi correlati al valore della riga corrente. Ad esempio, in una tabella Dipendenti che contiene i dipendenti, i responsabili dei dipendenti e i responsabili dei responsabili, è possibile restituire il percorso che connette un dipendente al suo responsabile.

- Il percorso non è vincolato a un singolo livello di relazioni padre-figlio; può restituire righe correlate che rappresentano diversi livelli a partire dalla riga iniziale specificata.
  - Il delimitatore usato per separare i predecessori è la barra verticale, "|".
  - I valori in *ID\_columnName* e *parent\_columnName* devono essere dello stesso tipo di dati, ossia testo o un numero intero.
  - I valori in *parent\_columnName* devono essere presenti in *ID\_columnName*. Questo significa che non è possibile cercare un elemento padre se non è presente alcun valore a livello di elemento figlio.
  - Se *parent\_columnName* è vuoto, PATH() restituisce il valore *ID\_columnName*. In altre parole, se si cerca il responsabile di un dipendente, ma la colonna *parent\_columnName* non contiene dati, la funzione PATH restituisce solo l'ID del dipendente.
  - Se *ID\_columnName* ha dei duplicati e il valore di *parent\_columnName* è lo stesso per tali duplicati, PATH() restituisce il valore *parent\_columnName* comune. Se invece il valore di *parent\_columnName* è diverso per tali duplicati, PATH() restituisce un errore. In altre parole, se sono presenti due voci per lo stesso ID dipendente con lo stesso ID responsabile, la funzione PATH restituisce l'ID di tale responsabile. Se invece sono presenti due ID dipendente identici con ID responsabile diversi, la funzione PATH restituisce un errore.
  - Se *ID\_columnName* è vuoto, PATH() restituisce un valore vuoto.
  - Se *ID\_columnName* contiene una barra verticale "|", PATH () restituisce un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene creata una colonna calcolata che elenca tutti i responsabili di ogni dipendente.

DAX

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# PATHCONTAINS

Articolo • 20/10/2023

Restituisce **TRUE** se l'elemento *item* specificato è presente nel *path* specificato.

## Sintassi

DAX

```
PATHCONTAINS(<path>, <item>)
```

## Parametri

Termino	Definizione
path	Stringa creata come risultato della valutazione di una funzione PATH.
oggetto	Espressione di testo da cercare nel risultato del percorso.

## Valore restituito

Valore **TRUE** se *item* esiste in *path*; in caso contrario, **FALSE**.

## Osservazioni:

- Se *item* è un numero intero, viene convertito in testo e quindi la funzione viene valutata. Se la conversione non riesce, la funzione restituisce un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente viene creata una colonna calcolata che accetta un ID responsabile e controlla un set di dipendenti. Se l'ID responsabile è nell'elenco di responsabili restituiti dalla funzione PATH, la funzione PATHCONTAINS restituisce True; in caso contrario, restituisce False.

DAX

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
"23")
```

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PATHITEM

Articolo • 20/10/2023

Restituisce l'elemento in corrispondenza del parametro *position* specificato da una stringa risultante dalla valutazione di una funzione PATH. Le posizioni vengono conteggiate da sinistra a destra.

## Sintassi

DAX

```
PATHITEM(<path>, <position>[, <type>])
```

## Parametri

Termine	Definizione
path	Stringa di testo nel formato di risultati di una funzione PATH.
località	Espressione integer con la posizione dell'elemento da restituire.
Tipo	(Facoltativo) Enumerazione che definisce il tipo di dati del risultato:

## Enumerazione di type

Enumerazione	Enumerazione alternativa	Descrizione
TEXT	0	Risultati restituiti con il testo del tipo di dati (impostazione predefinita).
INTEGER	1	Risultati restituiti come Integer.

## Valore restituito

Identificatore restituito dalla funzione PATH in corrispondenza della posizione specificata nell'elenco di identificatori. Gli elementi restituiti dalla funzione PATH sono ordinati dal più distante al corrente.

## Osservazioni:

- Questa funzione può essere usata per restituire un livello specifico da una gerarchia restituita da una funzione PATH. Ad esempio, è possibile restituire solo i responsabili di tutti i dipendenti.
- Se si specifica un numero per il parametro *position* minore di uno (1) o maggiore del numero di elementi nel parametro *path*, la funzione PATHITEM restituisce BLANK.
- Se *type* non è un elemento di enumerazione valido, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce il responsabile di terzo livello del dipendente corrente; accetta gli ID del dipendente e del responsabile come input di una funzione PATH che restituisce una stringa con la gerarchia di elementi padre del dipendente corrente. Dalla stringa in questione PATHITEM restituisce la terza voce come numero intero.

DAX

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# PATHITEMREVERSE

Articolo • 20/10/2023

Restituisce l'elemento in corrispondenza del parametro *position* specificato da una stringa risultante dalla valutazione di una funzione PATH. Le posizioni vengono conteggiate partendo da destra verso sinistra.

## Sintassi

DAX

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

## Parametri

Termine	Definizione
path	Stringa di testo risultante dalla valutazione di una funzione PATH.
località	Espressione integer con la posizione dell'elemento da restituire. La posizione viene conteggiata partendo da destra verso sinistra.
Tipo	(Facoltativo) Enumerazione che definisce il tipo di dati del risultato:

## Enumerazione di type

Enumerazione	Enumerazione alternativa	Descrizione
TEXT	0	Risultati restituiti con il testo del tipo di dati (impostazione predefinita).
INTEGER	1	Risultati restituiti come Integer.

## Valore restituito

Posizione n crescente nel percorso specificato, partendo dalla posizione corrente fino all'ultima.

## Osservazioni:

- Questa funzione può essere usata per ottenere un elemento singolo da una gerarchia risultante da una funzione PATH.
- La funzione inverte l'ordine standard della gerarchia, in modo che gli elementi più vicini siano elencati per primi. Ad esempio, se la funzione PATH restituisce un elenco di superiori di un dipendente in una gerarchia, la funzione PATHITEMREVERSE restituisce il superiore diretto del dipendente in posizione 2 perché la posizione 1 contiene l'ID del dipendente.
- Se il numero specificato per *posizione* è minore di uno (1) o maggiore del numero di elementi nel *percorso*, la funzione PATHITEM restituisce BLANK.
- Se *type* non è un elemento di enumerazione valido, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente accetta una colonna ID dipendente come input di una funzione PATH e inverte l'elenco di elementi padre del padre restituiti. La posizione specificata è 3 e il tipo restituito è 1, pertanto la funzione PATHITEMREVERSE restituisce un numero intero che rappresenta il responsabile di due livelli superiore al dipendente.

DAX

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
3, 1)
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# PATHLENGTH

Articolo • 20/10/2023

Restituisce il numero di elementi padre relativi all'elemento specificato in un determinato risultato PATH, incluso il risultato stesso.

## Sintassi

DAX

```
PATHLENGTH(<path>)
```

## Parametri

Termine	Definizione
path	Espressione di testo risultante dalla valutazione di una funzione PATH.

## Valore restituito

Numero di elementi padre dell'elemento specificato in un determinato risultato PATH, incluso l'elemento specificato.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente un ID dipendente viene accettato come input di una funzione PATH e viene restituito un elenco di responsabili sopra tale dipendente nella gerarchia. La funzione PATHLENGTH accetta il risultato e conteggia i livelli diversi di dipendenti e responsabili, incluso il dipendente con cui si è iniziato.

DAX

```
= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni di relazione

Articolo • 20/10/2023

Le funzioni in questa categoria sono destinate alla gestione e all'uso delle relazioni tra le tabelle.

## In questa categoria

Funzione	Descrizione
CROSSFILTER	Specifica la direzione tra filtri da usare in un calcolo per una relazione esistente tra due colonne.
RELATED	Restituisce un valore correlato da un'altra tabella.
RELATEDTABLE	Valuta un'espressione di tabella in un contesto modificato dai filtri specificati.
USERELATIONSHIP	Specifica la relazione da usare in un calcolo specifico come relazione esistente tra columnName1 e columnName2.

## Commenti e suggerimenti

Questa pagina è stata utile?



# CROSSFILTER

Articolo • 20/10/2023

Specifica la direzione tra filtri da usare in un calcolo per una relazione esistente tra due colonne.

## Sintassi

DAX

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

## Parametri

Termine	Definizione
columnName1	Nome completo di una colonna esistente in sintassi DAX standard, che generalmente rappresenta il lato "molti" della relazione da usare. Se gli argomenti vengono dati in ordine inverso, prima dell'uso la funzione li scambia. Questo argomento non può essere un'espressione.
columnName2	Nome completo di una colonna esistente in sintassi DAX standard, che generalmente rappresenta il lato "uno" o "ricerca" della relazione da usare. Se gli argomenti vengono dati in ordine inverso, prima dell'uso la funzione li scambia. Questo argomento non può essere un'espressione.
Direction	<p>Direzione del filtro incrociato da usare. I possibili valori sono i seguenti:</p> <p><b>Nessuna:</b> nessun filtro incrociato si verifica lungo questa relazione.</p> <p><b>Entrambi:</b> i filtri su entrambi i lati filtrano l'altro lato.</p> <p><b>OneWay:</b> filtri su un lato o sul lato di ricerca di una relazione filtrano l'altro lato. Questa opzione non può essere usata con una relazione uno-a-uno. Non usare questa opzione in una relazione molti-a-molti perché non è chiaro quale lato è il lato di ricerca; usare invece OneWay_LeftFiltersRight o OneWay_RightFiltersLeft.</p> <p><b>OneWay_LeftFiltersRight:</b> i filtri sul lato di &lt;columnName1&gt; filtrano il lato di &lt;columnName2&gt;. Questa opzione non può essere usata con una relazione uno-a-uno o molti-a-uno.</p> <p><b>OneWay_RightFiltersLeft:</b> i filtri sul lato di &lt;columnName2&gt; filtrano il lato di &lt;columnName1&gt;. Questa opzione non può essere usata con una relazione uno-a-uno o molti-a-uno.</p>

# Valore restituito

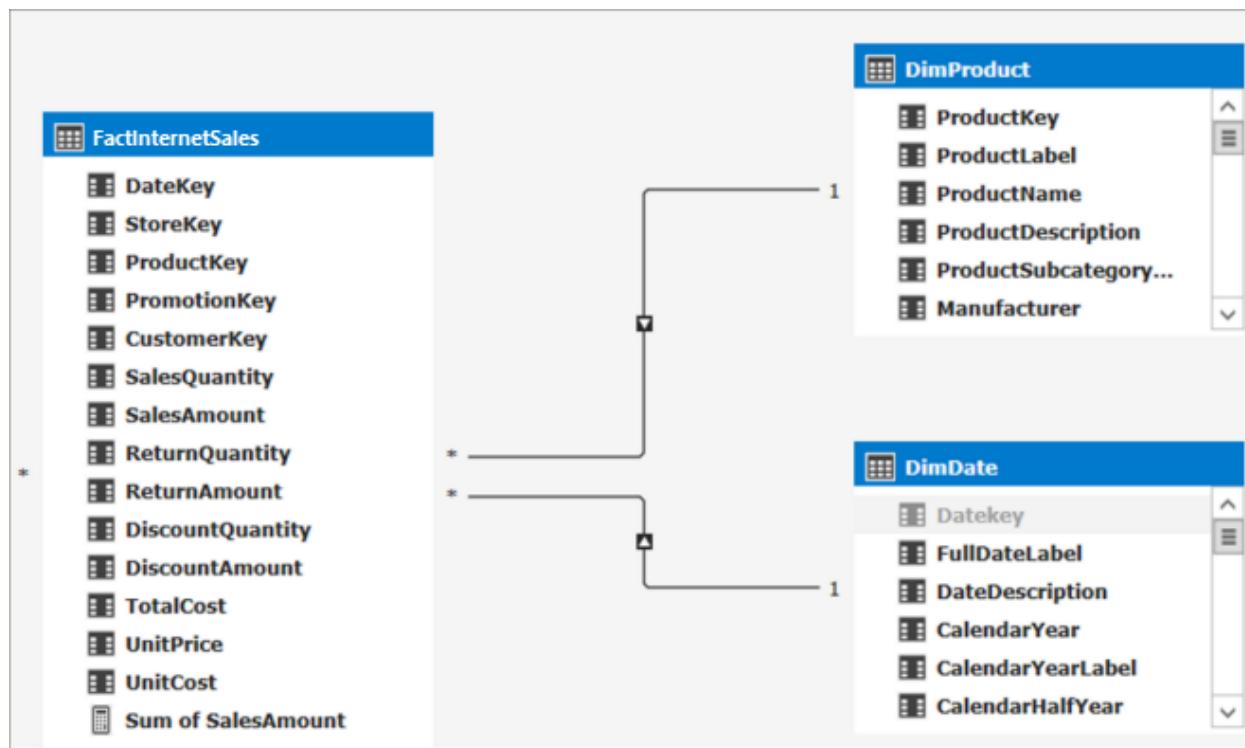
La funzione non restituisce alcun valore. La funzione imposta solo la direzione del filtro incrociato per la relazione indicata, per la durata della query.

## Osservazioni:

- Nel caso di una relazione 1:1, non c'è differenza tra le direzioni "one" e "both".
- CROSSFILTER può essere usato solo nelle funzioni che accettano un filtro come argomento, ad esempio: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD e TOTALYTD.
- CROSSFILTER usa le relazioni esistenti nel modello, identificandole in base alle colonne punto finale.
- In CROSSFILTER, l'impostazione del filtro incrociato di una relazione non è importante. In altre parole, il fatto che la relazione sia impostata per l'applicazione del filtro in una o entrambe le direzioni nel modello non influisce sull'uso della funzione. CROSSFILTER eseguirà l'override di qualsiasi impostazione di filtro incrociato esistente.
- Se alcune delle colonne indicate come argomento non fanno parte di una relazione o se gli argomenti appartengono a relazioni diverse, viene restituito un errore.
- Se le espressioni CALCULATE sono annidate e più espressioni CALCULATE contengono una funzione CROSSFILTER, la funzione CROSSFILTER più interna sarà quella prevalente in caso di conflitto o ambiguità.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nel diagramma del modello seguente, Sia DimProduct che DimDate hanno una singola relazione di direzione con FactInternetSales.



Per impostazione predefinita, non è possibile ottenere il numero di prodotti venduti per anno:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey
2005	\$3,266,373.66	606
2006	\$6,530,343.53	606
2007	\$9,791,060.30	606
2008	\$9,770,899.74	606
2009		606
2010		606
<b>Grand Total</b>	<b>\$29,358,677.22</b>	<b>606</b>

Esistono due modi per ottenere il numero di prodotti in base all'anno:

- Attivare l'uso di filtri incrociati bidirezionali per la relazione. Questo modificherà il funzionamento dei filtri per tutti i dati tra le due tabelle.
- Usare la funzione CROSSFILTER per modificare il funzionamento delle relazioni per solo questa misura.

Quando si usa DAX, è possibile usare la funzione CROSSFILTER per modificare il comportamento della direzione del filtro incrociato tra due colonne definite da una relazione. In questo caso, l'espressione DAX ha un aspetto simile al seguente:

```
DAX
BiDi:= CALCULATE([Distinct Count of ProductKey],
```

```
CROSSFILTER(FactInternetSales[ProductKey], DimProduct[ProductKey] , Both))
```

Usando la funzione CROSSFILTER nell'espressione di misura si ottengono i risultati previsti:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey	BiDi
2005	\$3,266,373.66	606	25
2006	\$6,530,343.53	606	56
2007	\$9,791,060.30	606	133
2008	\$9,770,899.74	606	102
2009		606	
2010		606	
<b>Grand Total</b>	<b>\$29,358,677.22</b>	<b>606</b>	<b>606</b>

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì No

# RELATED

Articolo • 20/10/2023

Restituisce un valore correlato da un'altra tabella.

## Sintassi

DAX
<code>RELATED(&lt;column&gt;)</code>

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori che si vogliono recuperare.

## Valore restituito

Singolo valore correlato alla riga corrente.

## Osservazioni:

- Per la funzione RELATED è necessario che esista una relazione tra la tabella corrente e la tabella con le informazioni correlate. Si specifica la colonna che contiene i dati desiderati e la funzione segue una relazione molti-a-uno esistente per recuperare il valore dalla colonna specificata nella tabella correlata. Se non esiste una relazione, è necessario crearne una.
- Quando la funzione RELATED esegue una ricerca, esamina tutti i valori nella tabella specificata indipendentemente da eventuali filtri applicati.
- Per la funzione RELATED è necessario un contesto di riga. Può quindi essere usata solo nell'espressione di colonna calcolata, in cui il contesto della riga corrente non è ambiguo, o come funzione annidata in un'espressione che usa una funzione di scansione di tabella. Una funzione di scansione di tabella, ad esempio SUMX, ottiene il valore della riga corrente e quindi cerca in un'altra tabella le istanze di tale valore.

- Non è possibile usare la funzione RELATED per recuperare una colonna in una relazione limitata.

## Esempio

Nell'esempio seguente viene creata la misura Non USA Internet Sales per generare un report sulle vendite che esclude le vendite negli Stati Uniti. Per creare la misura, è necessario filtrare la tabella InternetSales\_USD per escludere tutte le vendite che appartengono agli Stati Uniti nella tabella SalesTerritory. Gli Stati Uniti, come paese, vengono visualizzati 5 volte nella tabella SalesTerritory, ovvero una volta per ognuna delle regioni seguenti: nord-ovest, nord-est, centrale, sud-ovest e sud-est.

Il primo approccio per filtrare le vendite Internet, per poter creare la misura, potrebbe consistere nell'aggiungere un'espressione di filtro come la seguente:

DAX

```
FILTER('InternetSales_USD'
, 'InternetSales_USD'[SalesTerritoryKey]<>1 &&
'InternetSales_USD'[SalesTerritoryKey]<>2 &&
'InternetSales_USD'[SalesTerritoryKey]<>3 &&
'InternetSales_USD'[SalesTerritoryKey]<>4 &&
'InternetSales_USD'[SalesTerritoryKey]<>5)
```

Questo approccio è tuttavia controllato, soggetto a errori di digitazione e potrebbe non funzionare se una delle aree esistenti venisse suddivisa in futuro.

Un approccio migliore consiste nell'usare la relazione esistente tra InternetSales\_USD e SalesTerritory e nell'indicare in modo esplicito che il paese deve essere diverso da United States. A tale scopo, creare un'espressione di filtro simile alla seguente:

DAX

```
FILTER( 'InternetSales_USD',
RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

Questa espressione usa la funzione RELATED per cercare il valore del paese nella tabella SalesTerritory, a partire dal valore della colonna chiave SalesTerritoryKey nella tabella InternetSales\_USD. Il risultato della ricerca viene usato dalla funzione di filtro per determinare se la riga InternetSales\_USD sia filtrata o meno.

ⓘ Nota

Se l'esempio non funziona, potrebbe essere necessario creare una relazione tra le tabelle.

DAX

```
= SUMX(FILTER('InternetSales_USD'  
    , RELATED('SalesTerritory'[SalesTerritoryCountry])  
    <>"United States"  
)  
, 'InternetSales_USD'[SalesAmount_USD])
```

La tabella seguente illustra solo i totali per ogni area, per dimostrare che l'espressione di filtro nella misura, Non USA Internet Sales, funziona come previsto.

Row Labels	Internet Sales	Non USA Internet Sales
Australia	\$4,999,021.84	\$4,999,021.84
Canada	\$1,343,109.10	\$1,343,109.10
Francia	\$2,490,944.57	\$2,490,944.57
Germania	\$2,775,195.60	\$2,775,195.60
Regno Unito	\$5,057,076.55	\$5,057,076.55
Stati Uniti	\$9,389,479.79	
Grand Total	\$26,054,827.45	\$16,665,347.67

Di seguito viene illustrato ciò che è possibile ottenere se è stata usata questa misura in un oggetto visivo della tabella del report:

Row Labels	Accessories	Bikes	Clothing	Grand Total
2005		\$1,526,481.95		\$1,526,481.95
2006		\$3,554,744.04		\$3,554,744.04
2007	\$156,480.18	\$5,640,106.05	\$70,142.77	\$5,866,729.00
2008	\$228,159.45	\$5,386,558.19	\$102,675.04	\$5,717,392.68
Grand Total	\$384,639.63	\$16,107,890.23	\$172,817.81	\$16,665,347.67

**Vedi anche**

RELATEDTABLE

Funzioni di filtro

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# RELATEDTABLE

Articolo • 20/10/2023

Valuta un'espressione di tabella in un contesto modificato dai filtri specificati.

## Sintassi

DAX

```
RELATEDTABLE(<tableName>)
```

## Parametri

Termine	Definizione
tableName	Nome di una tabella esistente con la sintassi DAX standard. Non può essere un'espressione.

## Valore restituito

Tabella di valori.

## Osservazioni:

- La funzione RELATEDTABLE modifica il contesto in cui vengono filtrati i dati e valuta l'espressione nel nuovo contesto specificato.
- Questa funzione è un collegamento per la funzione CALCULATETABLE senza espressioni logiche.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente usa la funzione RELATEDTABLE per creare una colonna calcolata con le vendite Internet nella tabella Categoria di prodotto:

DAX

```
= SUMX( RELATEDTABLE('InternetSales_USD')
    , [SalesAmount_USD])
```

La tabella seguente mostra i risultati:

Product Category Key	Product Category AlternateKey	Product Category Name	Internet Sales
1	1	Bikes	\$28,318,144.65
2	2	Componenti	
3	3	Clothing	\$339,772.61
4	4	Accessori	\$700,759.96

## Vedi anche

[CALCULATETABLE](#)

[Funzioni di filtro](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# USERELATIONSHIP

Articolo • 20/10/2023

Specifica la relazione da usare in un calcolo specifico come relazione esistente tra columnName1 e columnName2.

## Sintassi

DAX

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

## Parametri

Termine	Definizione
columnName1	Nome completo di una colonna esistente in sintassi DAX standard, che generalmente rappresenta il lato "molti" della relazione da usare. Se gli argomenti vengono dati in ordine inverso, prima dell'uso la funzione li scambia. Questo argomento non può essere un'espressione.
columnName2	Nome completo di una colonna esistente in sintassi DAX standard, che generalmente rappresenta il lato "uno" o "ricerca" della relazione da usare. Se gli argomenti vengono dati in ordine inverso, prima dell'uso la funzione li scambia. Questo argomento non può essere un'espressione.

## Valore restituito

La funzione non restituisce alcun valore. La funzione abilita solo la relazione indicata per la durata del calcolo.

## Osservazioni:

- CROSSFILTER può essere usato solo nelle funzioni che accettano un filtro come argomento, ad esempio: funzioni CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD and TOTALYTD.

- Non è possibile usare USERELATIONSHIP quando viene definita la sicurezza a livello di riga per la tabella in cui è inclusa la misura.  
`CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))`, ad esempio, restituirà un errore se è definita la sicurezza a livello di riga per DimCustomer.
- USERELATIONSHIP usa le relazioni esistenti nel modello, identificandole in base alle colonne punto finale.
- In USERELATIONSHIP lo stato di una relazione non è importante: il fatto che la relazione sia attiva o meno non influisce sull'utilizzo della funzione. Anche se è inattiva, la relazione viene usata ed esegue l'override di eventuali altre relazioni attive che potrebbero essere presenti nel modello, ma non sono indicate negli argomenti della funzione.
- Se alcune delle colonne indicate come argomento non fanno parte di una relazione o se gli argomenti appartengono a relazioni diverse, viene restituito un errore.
- Se sono necessarie più relazioni per creare un join tra la tabella A e la tabella B in un calcolo, ogni relazione deve essere indicata in una funzione USERELATIONSHIP diversa.
- Se le espressioni CALCULATE sono annidate e più espressioni CALCULATE contengono una funzione USERELATIONSHIP, la funzione USERELATIONSHIP più interna sarà quella prevalente in caso di conflitto o ambiguità.
- È possibile annidare fino a 10 funzioni USERELATIONSHIP. L'espressione potrebbe tuttavia avere un livello più profondo di annidamento. L'espressione di esempio seguente, ad esempio, è annidata a 3 livelli di profondità, di cui solo 2 per USEREALTIONSHIP: `=CALCULATE(CALCULATE( CALCULATE( <anyExpression>, USERELATIONSHIP( t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP( t1[colA], t2[colA]))`.
- Per le relazioni da 1 a 1, USERELATIONSHIP attiverà la relazione solo in una direzione. In particolare, i filtri potranno passare solo dalla tabella *columnName2* alla tabella *columnName1*. Se si desidera un filtro incrociato bidirezionale, è possibile utilizzare due USERELATIONSHIPS con direzione opposta nello stesso calcolo. Ad esempio, `CALCULATE(..., USERELATIONSHIP(T1[K], T2[K]), USERELATIONSHIP(T2[K], T1[K]))`.

## Esempio

L'esempio seguente illustra come eseguire l'override della relazione predefinita attiva tra le tabelle InternetSales e DateTime. La relazione predefinita esiste tra la colonna OrderDate della tabella InternetSales e la colonna Date della tabella DateTime.

Per calcolare la somma delle vendite Internet e consentire il sezionamento in base a ShippingDate invece che al tradizionale valore OrderDate, creare la misura [InternetSales by ShippingDate] usando l'espressione seguente:

DAX

```
= CALCULATE(SUM(InternetSales[SalesAmount]),  
USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Le relazioni tra InternetSales[ShipmentDate] e DateTime[Date] devono esistere e non devono essere la relazione attiva. Inoltre, la relazione tra InternetSales[OrderDate] e DateTime[Date] deve esistere e deve essere la relazione attiva.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni statistiche

Articolo • 20/10/2023

Le funzioni statistiche calcolano i valori correlati a distribuzioni statistiche e probabilità, ad esempio deviazione standard e numero di permutazioni.

## In questa categoria

Funzione	Descrizione
BETA.DIST	Restituisce la distribuzione beta.
BETA.INV	Restituisce l'inversa della funzione di densità della probabilità cumulativa beta (BETA.DIST).
CHISQ.DIST	Restituisce la distribuzione chi quadrato.
CHISQ.DIST.RT	Restituisce la probabilità a una coda destra della distribuzione chi quadrato.
CHISQ.INV	Restituisce l'inversa della probabilità a una coda sinistra della distribuzione chi quadrato.
CHISQ.INV.RT	Restituisce l'inversa della probabilità a una coda destra della distribuzione chi quadrato.
COMBIN	Restituisce il numero di combinazioni per un numero specificato di elementi.
COMBINA	Restituisce il numero di combinazioni (con ripetizioni) per un numero specificato di elementi.
CONFIDENCE.NORM	L'intervallo di confidenza è un intervallo di valori.
CONFIDENCE.T	Restituisce l'intervallo di confidenza per una media di popolazione usando una distribuzione t di Student.
EXPON.DIST	Restituisce la distribuzione esponenziale.
GEOMEAN	Restituisce la media geometrica dei numeri in una colonna.
GEOMEANX	Restituisce la media geometrica di un'espressione valutata per ogni riga di una tabella.
LINEST	Usa il metodo dei minimi quadrati per calcolare la linea retta che meglio corrisponde ai dati specificati.
LINESTX	Usa il metodo dei minimi quadrati per calcolare la linea retta che meglio corrisponde ai dati specificati. Il risultato dei dati dalle espressioni valutate

<b>Funzione</b>	<b>Descrizione</b>
	per ogni riga di una tabella.
MEDIAN	Restituisce la mediana dei numeri in una colonna.
MEDIANX	Restituisce il numero mediano di un'espressione valutata per ogni riga di una tabella.
NORM.DIST	Restituisce la distribuzione normale per la media e la deviazione standard specificate.
NORM.INV	Inversa della distribuzione normale cumulativa per la media e la deviazione standard specificate.
NORM.S.DIST	Restituisce la distribuzione normale standard (ha media pari a zero e deviazione standard pari a 1).
NORM.S.INV	Restituisce l'inversa della distribuzione cumulativa normale standard.
PERCENTILE.EXC	Restituisce il k-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi esclusi.
PERCENTILE.INC	Restituisce il k-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi inclusi.
PERCENTILEX.EXC	Restituisce il numero percentile di un'espressione valutata per ogni riga di una tabella.
PERCENTILEX.INC	Restituisce il numero percentile di un'espressione valutata per ogni riga di una tabella.
PERMUT	Restituisce il numero delle permutazioni per un determinato numero di oggetti che è possibile selezionare da oggetti numerici.
POISSON.DIST	Restituisce la distribuzione di probabilità di Poisson.
RANK.EQ	Restituisce il rango di un numero in un elenco di numeri.
RANKX	Restituisce il rango di un numero in un elenco di numeri per ogni riga nell'argomento <i>table</i> .
SAMPLE	Restituisce un esempio di N righe dalla tabella specificata.
STDEV.P	Restituisce la deviazione standard dell'intera popolazione.
STDEV.S	Restituisce la deviazione standard di una popolazione di esempio.
STDEVX.P	Restituisce la deviazione standard dell'intera popolazione.
STDEVX.S	Restituisce la deviazione standard di una popolazione di esempio.
T.DIST	Restituisce la distribuzione t di Student a una coda sinistra.

Funzione	Descrizione
T.DIST.2T	Restituisce la distribuzione t di Student a due code.
T.DIST.RT	Restituisce la distribuzione t di Student a una coda destra.
T.INV	Restituisce l'inversa della distribuzione t di Student a una coda sinistra.
T.INV.2t	Restituisce l'inversa della distribuzione t di Student a due code.
VAR.P	Restituisce la varianza dell'intera popolazione.
VAR.S	Restituisce la varianza di una popolazione di esempio.
VARX.P	Restituisce la varianza dell'intera popolazione.
VARX.S	Restituisce la varianza di una popolazione di esempio.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# BETA.DIST

Articolo • 20/10/2023

Restituisce la distribuzione beta. La distribuzione beta viene in genere usata per studiare le variazioni nella percentuale di un oggetto tra campioni, ad esempio la frazione della giornata che le persone trascorrono guardando la televisione.

## Sintassi

DAX

```
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

## Parametri

Termine	Definizione
x	Valore compreso tra A e B in cui calcolare la funzione
Alfa	Parametro della distribuzione.
Beta	Parametro della distribuzione.
A	(Facoltativo). Limite inferiore per l'intervallo di x.
B	(Facoltativo). Limite superiore dell'intervallo di x.

## Valore restituito

Restituisce la distribuzione beta.

## Osservazioni:

- Se uno degli argomenti non è numerico, BETA.DIST restituisce il valore di errore #VALUE! .
- Se uno degli argomenti non è un numero intero, viene arrotondato.
- Se  $\alpha \leq 0$  o  $\beta \leq 0$ , BETA.DIST restituisce il valore di errore #NUM! .
- Se  $x < A$ ,  $x > B$  o  $A = B$ , BETA.DIST restituisce il valore di errore #NUM! .

- Se non vengono specificati valori per A e B, la funzione BETA.DIST usa la distribuzione cumulativa beta standard in modo che venga soddisfatta la condizione A = 0 e B = 1.
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# BETA.INV

Articolo • 20/10/2023

Restituisce l'inversa della funzione di densità della probabilità cumulativa beta (BETA.DIST).

Se probability = BETA.INV(x;...TRUE), si avrà BETA.INV(probability;...) = x. Dati un tempo di completamento e una variabilità previsti, la distribuzione beta può essere usata nella pianificazione di progetti per calcolare i tempi di completamento probabili.

## Sintassi

DAX

**BETA.INV(probability, alpha, beta, [A], [B])**

## Parametri

Termine	Definizione
Probability	Probabilità associata alla distribuzione beta.
Alfa	Parametro della distribuzione.
Beta	Parametro della distribuzione.
A	(Facoltativo). Limite inferiore per l'intervallo di x.
B	(Facoltativo). Limite superiore dell'intervallo di x.

## Valore restituito

Restituisce l'inversa della funzione di densità della probabilità cumulativa beta (BETA.DIST).

## Osservazioni:

- Se uno degli argomenti non è numerico, BETA.INV restituisce il valore di errore #VALUE! .
- Se uno degli argomenti non è un numero intero, viene arrotondato.

- Se alpha ≤ 0 o beta ≤ 0, BETA.INV restituisce il valore di errore #NUM! .
  - Se probability ≤ 0 o probability > 1, BETA.INV restituisce il valore di errore #NUM! .
  - Se non vengono specificati valori per A e B, la funzione BETA.INV usa la distribuzione cumulativa beta standard in modo che venga soddisfatta la condizione A = 0 e B = 1.
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# CHISQ.DIST

Articolo • 20/10/2023

Restituisce la distribuzione chi quadrato.

La distribuzione chi quadrato viene in genere usata per studiare le variazioni nella percentuale di un oggetto tra campioni, ad esempio il numero di ore che si trascorrono quotidianamente davanti alla televisione.

## Sintassi

DAX

```
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

## Parametri

Termine	Definizione
x	Valore per cui calcolare la distribuzione.
Deg_freedom	Numero di gradi di libertà.
cumulative	Valore logico che determina la forma della funzione. Se cumulative è TRUE, CHISQ.DIST restituisce la funzione di distribuzione cumulativa. Se è FALSE, restituisce la funzione di densità di probabilità.

## Valore restituito

Distribuzione chi quadrato.

## Osservazioni:

- Se x o deg\_freedom non è un valore numerico, viene restituito un errore.
- Se deg\_freedom non è un numero intero, viene arrotondato.
- Se x < 0, viene restituito un errore.
- Se deg\_freedom < 1 o deg\_freedom > 10^10, viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Resi

[Value]

0.632120558828558

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CHISQ.DIST.RT

Articolo • 20/10/2023

Restituisce la probabilità a una coda destra della distribuzione chi quadrato.

La distribuzione chi quadrato è associata a un test del chi quadrato. Usare il test del chi quadrato per confrontare i valori osservati e previsti. Ad esempio, un esperimento genetico potrebbe ipotizzare che la prossima generazione di piante presenterà un determinato set di colori. Confrontando i risultati osservati con quelli previsti, è possibile stabilire se l'ipotesi formulata in origine è valida.

## Sintassi

DAX

```
CHISQ.DIST.RT(<x>, <deg_freedom>)
```

## Parametri

Termine	Definizione
x	Valore per cui calcolare la distribuzione.
Deg_freedom	Numero di gradi di libertà.

## Valore restituito

Probabilità a una coda destra della distribuzione chi quadrato.

## Osservazioni:

- Se x o deg\_freedom non è un valore numerico, viene restituito un errore.
- Se deg\_freedom non è un numero intero, viene arrotondato.
- Se x < 0, viene restituito un errore.
- Se deg\_freedom < 1 o deg\_freedom > 10^10, viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Resi

[Value]

0.84914503608461

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CHISQ.INV

Articolo • 20/10/2023

Restituisce l'inversa della probabilità a una coda sinistra della distribuzione chi quadrato.

La distribuzione chi quadrato viene in genere usata per studiare le variazioni nella percentuale di un oggetto tra campioni, ad esempio il numero di ore che si trascorrono quotidianamente davanti alla televisione.

## Sintassi

DAX

`CHISQ.INV(probability,deg_freedom)`

## Parametri

Termine	Definizione
Probability	Probabilità associata alla distribuzione chi quadrato.
Deg_freedom	Numero di gradi di libertà.

## Valore restituito

Restituisce l'inversa della probabilità a una coda sinistra della distribuzione chi quadrato.

## Osservazioni:

- Se l'argomento non è numerico, CHISQ.INV restituisce il valore di errore #VALUE! .
- Se probabilità < 0 o probabilità > 1, CHISQ.INV. restituisce il valore di errore #NUM! .
- Se deg\_freedom non è un numero intero, viene arrotondato.
- Se deg\_freedom < 0 o deg\_freedom > 10^10, CHISQ.INV restituisce il valore di errore #NUM! .

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= CHISQ.INV(0.93,1)	Inversa della probabilità a una coda sinistra della distribuzione chi quadrato per 0.93, usando 1 grado di libertà.	5.318520074
= CHISQ.INV(0.6,2)	Inversa della probabilità a una coda sinistra della distribuzione chi quadrato per 0.6, usando 2 gradi di libertà.	1.832581464

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# CHISQ.INV.RT

Articolo • 20/10/2023

Restituisce l'inversa della probabilità a una coda destra della distribuzione chi quadrato.

Se probability = CHISQ.DIST.RT(x,...), allora CHISQ.INV.RT(probability,...) = x. Usare questa funzione per confrontare i risultati osservati con quelli previsti e stabilire se l'ipotesi formulata in origine è valida.

## Sintassi

DAX

**CHISQ.INV.RT**(probability,deg\_freedom)

## Parametri

Termine	Definizione
Probability	Probabilità associata alla distribuzione chi quadrato.
Deg_freedom	Numero di gradi di libertà.

## Valore restituito

Restituisce l'inversa della probabilità a una coda destra della distribuzione chi quadrato.

## Osservazioni:

- Se uno degli argomenti non è un valore numerico, CHISQ.INV.RT restituisce il valore di errore #VALUE! .
- Se probability < 0 o probability > 1, CHISQ.INV.RT restituisce il valore di errore #NUM! .
- Se deg\_freedom non è un numero intero, viene arrotondato.
- Se deg\_freedom < 1, CHISQ.INV.RT restituisce il valore di errore #NUM! .

- Dato un valore probability, CHISQ.INV.RT ricerca tale valore x in base al presupposto che CHISQ.DIST.RT(x, deg\_freedom) = probability. Quindi, la precisione di CHISQ.INV.RT dipende dalla precisione di CHISQ.DIST.RT. CHISQ.INV.RT usa una tecnica di ricerca iterativa. Se la ricerca non converge dopo 64 iterazioni, la funzione restituirà come risultato il valore di errore #N/A.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# COMBIN

Articolo • 20/10/2023

Restituisce il numero di combinazioni per un numero specificato di elementi. Usare COMBIN per calcolare tutti i possibili gruppi che si possono formare con un determinato numero di elementi.

## Sintassi

DAX

```
COMBIN(number, number_chosen)
```

## Parametri

Termine	Definizione
number	Numero di elementi.
number_chosen	Numero di elementi in ogni combinazione.

## Valore restituito

Restituisce il numero di combinazioni per un numero specificato di elementi.

## Osservazioni:

- Gli argomenti numerici vengono troncati in numeri interi.
- Se uno degli argomenti non è un valore numerico, COMBIN restituisce il valore di errore #VALUE! .
- Se  $\text{number} < 0$ ,  $\text{number\_chosen} < 0$  oppure  $\text{number} < \text{number\_chosen}$ , COMBIN restituisce il valore di errore #NUM! .
- Una combinazione è un set o un subset di elementi, indipendentemente dal loro ordine interno. Le combinazioni si distinguono dalle permutazioni per il rilievo che assume l'ordine interno in queste ultime.

- Considerando number =  $n$  e number\_chosen =  $k$ , il numero delle possibili combinazioni sarà:

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

Dove

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= COMBIN(8,2)	Possibili gruppi di due persone che possono essere formati con 8 candidati.	28

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì
 No

# COMBINA

Articolo • 20/10/2023

Restituisce il numero di combinazioni (con ripetizioni) per un numero specificato di elementi.

## Sintassi

DAX

`COMBINA(number, number_chosen)`

## Parametri

Termine	Definizione
number	Deve essere maggiore o uguale a 0 e maggiore o uguale a number_chosen. I valori non interi vengono troncati.
number_chosen	Deve essere maggiore o uguale a 0. I valori non interi vengono troncati.

## Valore restituito

Restituisce il numero di combinazioni (con ripetizioni) per un numero specificato di elementi.

## Osservazioni:

- Se il valore di uno dei due argomenti non rispetta i vincoli, COMBINA restituisce il valore di errore #NUM! .
- Se uno dei due argomenti è un valore non numerico, COMBINA restituisce il valore di errore #VALUE! .
- Viene usata l'equazione seguente, dove  $N$  è Number e  $M$  è Number\_chosen:

$$\binom{N + M - 1}{N - 1}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= COMBINA(4,3)	Restituisce il numero delle combinazioni (con ripetizioni) per 4 e 3.	20
= COMBINA(10,3)	Restituisce il numero delle combinazioni (con ripetizioni) per 10 e 3.	220

## Commenti e suggerimenti

Questo articolo è stato utile?



# CONFIDENCE.NORM

Articolo • 20/10/2023

L'intervallo di confidenza è un intervallo di valori. La media campione,  $x$ , è al centro dell'intervallo, e l'intervallo è  $x \pm \text{CONFIDENCE.NORM}$ . Ad esempio, se  $x$  è la media campione dei tempi di recapito per i prodotti ordinati tramite posta,  $x \pm \text{CONFIDENCE.NORM}$  è un intervallo di medie della popolazione. Per qualsiasi media della popolazione  $\mu_0$  compresa in questo intervallo, la probabilità di ottenere una media campione che si discosti più da  $\mu_0$  che da  $x$  è maggiore di alpha. Per qualsiasi media della popolazione  $\mu_0$  non compresa in questo intervallo, la probabilità di ottenere una media campione che si discosti più da  $\mu_0$  che da  $x$  è minore di alpha. In altre parole, si supponga di usare  $x$ , standard\_dev e size per creare un test a due code al livello di significatività alpha dell'ipotesi secondo cui la media della popolazione è  $\mu_0$ . Tale ipotesi non verrà quindi rifiutata se  $\mu_0$  è compreso nell'intervallo di confidenza, mentre verrà rifiutata se  $\mu_0$  non è compreso nell'intervallo di confidenza. L'intervallo di confidenza non consente di dedurre che esiste una probabilità  $1 - \alpha$  che il pacchetto successivo richiederà un tempo di recapito compreso nell'intervallo di confidenza.

## Sintassi

DAX

```
CONFIDENCE.NORM(alpha,standard_dev,size)
```

## Parametri

Termine	Definizione
alfa	Livello di significatività usato per calcolare il livello di confidenza. Il livello di confidenza è uguale a $100*(1 - \alpha)\%$ . In altre parole, un valore alpha di 0.05 indica un livello di confidenza del 95%.
standard_dev	Deviazione standard della popolazione per l'intervallo di dati. Si presuppone che sia nota.
standard_dev,size	Dimensioni del campione.

## Valore restituito

Intervallo di valori

## Osservazioni:

- Se uno degli argomenti non è numerico, CONFIDENCE.NORM restituisce il valore di errore #VALUE! .
- Se alpha ≤ 0 o alpha ≥ 1, CONFIDENCE.NORM restituisce il valore di errore #NUM! .
- Se standard\_dev ≤ 0, CONFIDENCE.NORM restituisce il valore di errore #NUM! .
- Se size non è un numero intero, il valore viene arrotondato.
- Se size < 1, CONFIDENCE.NORM restituisce il valore di errore #NUM! .
- Se si suppone che alpha sia uguale a 0.05, sarà necessario calcolare l'area sottostante la curva normale standard che è uguale a (1 - alpha) o al 95%. Questo valore è ± 1.96. L'intervallo di confidenza è pertanto:

$$\bar{x} \pm 1.96 \left( \frac{\sigma}{\sqrt{n}} \right)$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# CONFIDENCE.T

Articolo • 20/10/2023

Restituisce l'intervallo di confidenza per una media di popolazione usando una distribuzione t di Student.

## Sintassi

DAX

```
CONFIDENCE.T(alpha,standard_dev,size)
```

## Parametri

Termine	Definizione
alfa	Livello di significatività usato per calcolare il livello di confidenza. Il livello di confidenza è uguale a $100*(1 - \text{alpha})\%$ . In altre parole, un valore alpha di 0.05 indica un livello di confidenza del 95%.
standard_dev	Deviazione standard della popolazione per l'intervallo di dati. Si presuppone che sia nota.
size	Dimensioni del campione.

## Valore restituito

Restituisce l'intervallo di confidenza per una media di popolazione usando una distribuzione t di Student.

## Osservazioni:

- Se uno degli argomenti non è numerico, CONFIDENCE.T restituisce il valore di errore #VALUE! .
- Se  $\text{alpha} \leq 0$  o  $\text{alpha} \geq 1$ , CONFIDENCE.T restituisce il valore di errore #NUM! .
- Se  $\text{standard\_dev} \leq 0$ , CONFIDENCE.T restituisce il valore di errore #NUM! .
- Se size non è un numero intero, il valore viene arrotondato.

- Se size è uguale a 1, CONFIDENCE.T restituisce il valore di errore #DIV/0! .
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Formula	Description	Result
= CONFIDENCE.T(0.05,1,50)	Intervallo di confidenza per una media di popolazione sulla base di una dimensione del campione di 50, con un livello di significatività del 5% e una deviazione standard di 1. Viene usata una distribuzione t di Student.	0.284196855

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EXPON.DIST

Articolo • 20/10/2023

Restituisce la distribuzione esponenziale. Usare la funzione EXPON.DIST per calcolare il tempo che intercorre tra gli eventi, ad esempio il tempo impiegato da uno sportello automatico per fornire la somma in contanti richiesta. È ad esempio possibile usare questa funzione per determinare la probabilità che questa operazione richieda al massimo un minuto.

## Sintassi

DAX
<code>EXPON.DIST(x,lambda,cumulative)</code>

## Parametri

Termine	Definizione
x	Required. Valore della funzione.
funzione lambda	Required. Valore del parametro.
cumulative	Required. Valore logico che indica la forma della funzione esponenziale da fornire. Se cumulative è TRUE, EXPON.DIST restituisce la funzione di distribuzione cumulativa. Se è FALSE, restituisce la funzione di densità di probabilità.

## Valore restituito

Restituisce la distribuzione esponenziale.

## Osservazioni:

- Se x o lambda non è un valore numerico, EXPON.DIST restituisce il valore di errore #VALUE! .
- Se x o lambda non è un numero intero, il valore viene arrotondato.
- Se x < 0, EXPON.DIST restituisce il valore di errore #NUM! .

- Se lambda ≤ 0, EXPON.DIST restituisce il valore di errore #NUM! .
- L'equazione per la funzione di densità di probabilità è:

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

- L'equazione per la funzione di distribuzione cumulativa è la seguente:

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- 

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# GEOMEAN

Articolo • 20/10/2023

Restituisce la media geometrica dei numeri in una colonna.

Per restituire la media geometrica di un'espressione valutata per ogni riga di una tabella, usare la [funzione GEOMEANX](#).

## Sintassi

DAX

`GEOMEAN(<column>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i numeri per i quali deve essere calcolata la media geometrica.

## Valore restituito

Numero decimale.

## Osservazioni:

- Vengono conteggiati solo i numeri nella colonna. I valori vuoti, logici e di testo vengono ignorati.
- `GEOMEAN( Table[Column] )` è equivalente a `GEOMEANX( Table, Table[Column] )`.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente calcola la media geometrica della colonna Return nella tabella Investment:

DAX

```
= GEOMEAN( Investment[Return] )
```

## Vedi anche

[Funzione GEOMEANX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# GEOMEANX

Articolo • 20/10/2023

Restituisce la media geometrica di un'espressione valutata per ogni riga di una tabella.

Per restituire la media geometrica dei numeri in una colonna, usare la [funzione GEOMEAN](#).

## Sintassi

DAX

```
GEOMEANX(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.

## Valore restituito

Numero decimale.

## Osservazioni:

- La funzione GEOMEANX accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento è una colonna che contiene i numeri per i quali si vuole calcolare la media geometrica o un'espressione che restituisce una colonna.
- Vengono conteggiati solo i numeri nella colonna. I valori vuoti, logici e di testo vengono ignorati.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

L'esempio seguente calcola la media geometrica della colonna ReturnPct nella tabella Investments:

DAX

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

## Vedi anche

[Funzione GEOMEAN](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# LINEST

Articolo • 20/10/2023

Usa il metodo Least Squares per calcolare una linea retta più adatta ai dati specificati, quindi restituisce una tabella che descrive la linea. L'equazione per la linea è nel formato:  $y = \text{Slope1} \cdot x_1 + \text{Slope2} \cdot x_2 + \dots + \text{Intercetta}$ .

## Sintassi

DAX

```
LINEST ( <columnY>, <columnX>[ , ...][ , <const> ] )
```

## Parametri

Termino	Definizione
columnY	Colonna dei valori y noti. Deve avere un tipo scalare.
columnX	Colonne dei valori x noti. Deve avere un tipo scalare. Deve esserne fornito almeno uno.
const	(Facoltativo) Valore VERO/FALSO costante che specifica se forzare la costante <b>Intercetta</b> su uguale a 0. Se VERO o omesso, il valore <b>Intercetta</b> viene calcolato normalmente; Se FALSO, il valore <b>Intercetta</b> è impostato su zero.

## Valore restituito

Tabella a riga singola che descrive la riga, oltre a statistiche aggiuntive. Di seguito sono riportate le colonne disponibili:

- **Slope1, Slope2, ..., SlopeN**: coefficienti corrispondenti a ogni valore x;
- **Intercetta**: valore intercetta;
- **StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN**: i valori di errore standard per i coefficienti **Slope1, Slope2, ..., SlopeN**;
- **StandardErrorIntercept**: valore di errore standard per la costante **Intercetta**;
- **CoefficientOfDetermination**: coefficiente di determinazione ( $r^2$ ). Confronta i valori y stimati e effettivi e gli intervalli in valore compreso tra 0 e 1: maggiore è il valore, maggiore è la correlazione nel campione;
- **StandardError**: errore standard per la stima y;
- **FStatistic**: statistica F o valore osservato da F. Usare la statistica F per determinare se la relazione osservata tra le variabili dipendenti e indipendenti si verifica per caso;
- **GradiOfFreedom**: gradi di libertà. Usare questo valore per trovare i valori critici F in una tabella statistica e determinare un livello di confidenza per il modello;
- **RegressionSumOfSquares**: la somma di regressione dei quadrati;
- **ResidualSumOfSquares**: somma residua di quadrati.

# Osservazioni:

<columnY> e <columnX> devono appartenere alla stessa tabella.

## Esempio 1

La query DAX seguente:

```
DAX  
  
EVALUATE LINEST(  
    'FactInternetSales'[SalesAmount],  
    'FactInternetSales'[TotalProductCost]  
)
```

Restituisce una tabella a riga singola con dieci colonne:

Slope1	Intercetta	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
1.67703250456677	6.34550460373026	0.000448675725548806	0.279131821917317	0.995695557281456
StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
60.9171030357485	13970688.6139993	60396	51843736761.658	224123120.339218

- **Slope1** e **Intercetta**: coefficienti del modello lineare calcolato;
- **StandardErrorSlope1** e **StandardErrorIntercept**: valori di errore standard per i coefficienti precedenti;
- **CoefficientOfDetermination**, **StandardError**, **FStatistic**, **DegreesOfFreedom**,  
**RegressionSumOfSquares** e **ResidualSumOfSquares**: statistiche di regressione sul modello.

Per una determinata vendita internet, questo modello stima l'importo della vendita in base alla formula seguente:

```
SalesAmount = Slope1 * TotalProductCost + Intercept
```

## Esempio 2

La query DAX seguente:

```
DAX  
  
EVALUATE LINEST(  
    'DimCustomer'[TotalSalesAmount],  
    'DimCustomer'[YearlyIncome],  
    'DimCustomer'[TotalChildren],  
    'DimCustomer'[BirthDate]  
)
```

Restituisce una tabella a riga singola con quattordici colonne:

- Slope1
- Slope2
- Slope3
- Intercetta
- StandardErrorSlope1
- StandardErrorSlope2
- StandardErrorSlope3
- StandardErrorIntercept
- CoefficientOfDetermination
- StandardError
- FStatistic
- DegreesOfFreedom
- RegressionSumOfSquares
- ResidualSumOfSquares

Per un determinato cliente, questo modello stima le vendite totali in base alla formula seguente (la data di nascita viene convertita automaticamente in un numero):

```
TotalSalesAmount = Slope1 * YearlyIncome + Slope2 * TotalChildren + Slope3 * BirthDate +
Intercept
```

## Vedi anche

[LINESTX](#)

[Funzioni statistiche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)

 [No](#)

# LINESTX

Articolo • 20/10/2023

Usa il metodo dei minimi quadrati (Least Squares) per calcolare una linea retta più adatta ai dati specificati, quindi restituisce una tabella che descrive la linea. Il risultato dei dati dalle espressioni valutate per ogni riga di una tabella. L'equazione per la linea è nel formato:  $y = \text{Slope1} * x_1 + \text{Slope2} * x_2 + \dots + \text{Intercetta}$ .

## Sintassi

DAX

```
LINESTX ( <table>, <expressionY>, <expressionX>[ , ...][ , <const>] )
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per cui verranno valutate le espressioni.
expressionY	Espressione da valutare per ogni riga della tabella, per ottenere i valori y noti. Deve avere un tipo scalare.
expressionX	Espressioni da valutare per ogni riga della tabella, per ottenere i valori x noti. Deve avere un tipo scalare. Deve esserne fornito almeno uno.
const	(Facoltativo) Valore VERO/FALSO costante che specifica se forzare la costante <b>Intercetta</b> su uguale a 0. Se VERO o omesso, il valore <b>Intercetta</b> viene calcolato normalmente; Se FALSO, il valore <b>Intercetta</b> è impostato su zero.

## Valore restituito

Tabella a riga singola che descrive la riga, oltre a statistiche aggiuntive. Di seguito sono riportate le colonne disponibili:

- **Slope1, Slope2, ..., SlopeN**: coefficienti corrispondenti a ogni valore x;
- **Intercetta**: valore intercetta;
- **StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN**: i valori di errore standard per i coefficienti **Slope1, Slope2, ..., SlopeN**;
- **StandardErrorIntercept**: valore di errore standard per la costante **Intercetta**;
- **CoefficientOfDetermination**: coefficiente di determinazione ( $r^2$ ). Confronta i valori y stimati e effettivi e gli intervalli in valore compreso tra 0 e 1: maggiore è il valore, maggiore è la correlazione nel campione;
- **StandardError**: errore standard per la stima y;

- **FStatistic**: statistica F o valore osservato da F. Usare la statistica F per determinare se la relazione osservata tra le variabili dipendenti e indipendenti si verifica per caso;
- **GradiOfFreedom**: gradi di libertà. Usare questo valore per trovare i valori critici F in una tabella statistica e determinare un livello di confidenza per il modello;
- **RegressionSumOfSquares**: la somma di regressione dei quadrati;
- **ResidualSumOfSquares**: somma residua dei quadrati.

## Esempio 1

La query DAX seguente:

```
DAX

DEFINE VAR TotalSalesByRegion = SUMMARIZECOLUMNS(
    'Sales Territory'[Sales Territory Key],
    'Sales Territory'[Population],
    "Total Sales", SUM(Sales[Sales Amount]))
)
EVALUATE LINESTX(
    'TotalSalesByRegion',
    [Total Sales],
    [Population]
)
```

Restituisce una tabella a riga singola con dieci colonne:

Slope1	Intercetta	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
6.42271517588	-410592.76216	0.24959467764561	307826.343996223	0.973535860750193

StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
630758.1747292	662.165707642	18	263446517001130	7161405749781.07

- **Slope1 e Intercetta**: coefficienti del modello lineare calcolato;
- **StandardErrorSlope1 e StandardErrorIntercept**: valori di errore standard per i coefficienti precedenti;
- **CoefficientOfDetermination, StandardError, FStatistic,DegreesOfFreedom, RegressionSumOfSquares e ResidualSumOfSquares**: statistiche di regressione sul modello.

Questo modello stima le vendite totali per un determinato territorio di vendita in base alla formula seguente:

```
Total Sales = Slope1 * Population + Intercept
```

## Esempio 2

La query DAX seguente:

```
DAX

DEFINE VAR TotalSalesByCustomer = SUMMARIZECOLUMNS(
    'Customer'[Customer ID],
    'Customer'[Age],
    'Customer'[NumOfChildren],
    "Total Sales", SUM(Sales[Sales Amount])
)
EVALUATE LINESTX(
    'TotalSalesByCustomer',
    [Total Sales],
    [Age],
    [NumOfChildren]
)
```

Restituisce una tabella a riga singola con dodici colonne:

Slope1	Slope2	Intercetta	StandardErrorSlope1
69.0435458093763	33.005949841721	-871.118539339539	0.872588875481658
StandardErrorSlope2	StandardErrorIntercept	CoefficientOfDetermination	StandardError
6.21158863903435	26.726292527427	0.984892920482022	68.5715034014342
FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
3161.91535144391	97	29734974.9782379	456098.954637092

Questo modello stima le vendite totali per un determinato cliente in base alla formula seguente:

```
Total Sales = Slope1 * Age + Slope2 * NumOfChildren + Intercept
```

## Vedi anche

[LINEST](#)

[Funzioni statistiche](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

Si

No

# MEDIAN

Articolo • 20/10/2023

Restituisce la mediana dei numeri in una colonna.

Per restituire la mediana di un'espressione valutata per ogni riga di una tabella, usare la [funzione MEDIANX](#).

## Sintassi

DAX

`MEDIAN(<column>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i numeri per i quali deve essere calcolata la mediana.

## Valore restituito

Numero decimale.

## Osservazioni:

- Vengono conteggiati solo i numeri nella colonna. I valori vuoti vengono ignorati. I valori logici, le date e il testo non sono supportati.
- `MEDIAN( Table[Column] )` è equivalente a `MEDIANX( Table, Table[Column] )`.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente calcola la mediana di una colonna denominata Age in una tabella denominata Customers:

DAX

```
= MEDIAN( Customers[Age] )
```

## Vedi anche

[Funzione MEDIANX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MEDIANX

Articolo • 20/10/2023

Restituisce il numero mediano di un'espressione valutata per ogni riga di una tabella.

Per restituire la mediana dei numeri di una colonna, usare la [funzione MEDIAN](#).

## Sintassi

DAX

```
MEDIANX(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.

## Valore restituito

Numero decimale.

## Osservazioni:

- La funzione MEDIANX accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento è una colonna che contiene i numeri per i quali si vuole calcolare la mediana o un'espressione che restituisce una colonna.
- Vengono conteggiati solo i numeri nella colonna.
- I valori logici e il testo vengono ignorati.
- MEDIANX non ignora i valori vuoti, a differenza di MEDIAN.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

L'esempio seguente calcola l'età media dei clienti che vivono negli Stati Uniti.

DAX

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ),  
Customers[Age] )
```

## Vedi anche

[Funzione MEDIAN](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NORM.DIST

Articolo • 20/10/2023

Restituisce la distribuzione normale per la media e la deviazione standard specificate.

## Sintassi

DAX

```
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

## Parametri

Termine	Definizione
X	Valore per il quale si vuole calcolare la distribuzione.
Valore medio	Media aritmetica della distribuzione.
Standard_dev	Deviazione standard della distribuzione.
Cumulative*	Valore logico che determina la forma della funzione. Se cumulative è TRUE, NORM.DIST restituisce la funzione di distribuzione cumulativa; se è FALSE, restituisce la funzione densità di probabilità.

## Valore restituito

Distribuzione normale per la media e la deviazione standard specificate.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Resi

[Value]

0.908788780274132

## Vedi anche

[Funzione NORM.S.DIST](#)

[Funzione NORM.INV](#)

[NORM.S.INV](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NORM.INV

Articolo • 20/10/2023

Inversa della distribuzione normale cumulativa per la media e la deviazione standard specificate.

## Sintassi

DAX

```
NORM.INV(Probability, Mean, Standard_dev)
```

## Parametri

Termine	Definizione
Probability	Probabilità corrispondente alla distribuzione normale.
Valore medio	Media aritmetica della distribuzione.
Standard_dev	Deviazione standard della distribuzione.

## Valore restituito

Restituisce l'inversa della distribuzione normale cumulativa per la media e la deviazione standard specificate.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

Resi

**[Value]**

42.00000200956628780274132

## Vedi anche

[NORM.S.INV](#)

[Funzione NORM.S.DIST](#)

[Funzione NORM.DIST](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NORM.S.DIST

Articolo • 20/10/2023

Restituisce la distribuzione normale standard (ha media pari a zero e deviazione standard pari a 1).

## Sintassi

DAX

```
NORM.S.DIST(Z, Cumulative)
```

## Parametri

Termine	Definizione
Z	Valore per il quale si vuole calcolare la distribuzione.
Cumulativo	Valore logico che determina la forma della funzione. Se cumulative è TRUE, NORM.S.DIST restituisce la funzione di distribuzione cumulativa; se è FALSE, restituisce la funzione densità di probabilità.

## Valore restituito

Distribuzione normale standard (ha media pari a zero e deviazione standard pari a 1).

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Resi

**[Value]**

0.908788725604095

## Vedi anche

[Funzione NORM.INV](#)

[Funzione NORM.DIST](#)

[NORM.S.INV](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



Sì



No

# NORM.S.INV

Articolo • 20/10/2023

Restituisce l'inversa della distribuzione cumulativa normale standard. La distribuzione ha una media pari a zero e una deviazione standard pari a 1.

## Sintassi

DAX

```
NORM.S.INV(Probability)
```

## Parametri

Termine	Definizione
Probability	Probabilità corrispondente alla distribuzione normale.

## Valore restituito

Inversa della distribuzione cumulativa normale standard. La distribuzione ha una media pari a zero e una deviazione standard pari a 1.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { NORM.S.INV(0.908789) }
```

Resi

**[Value]**

1.3333467304411

## Vedi anche

[NORM.INV](#)

[Funzione NORM.S.DIST](#)

[Funzione NORM.DIST](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PERCENTILE.EXC

Articolo • 20/10/2023

Restituisce il k-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi esclusi.

Per restituire il numero percentile di un'espressione valutata per ogni riga di una tabella, usare la [funzione PERCENTILEX.EXC](#).

## Sintassi

DAX

`PERCENTILE.EXC(<column>, <k>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori che definiscono la condizione relativa.
k	Valore percentile compreso nell'intervallo 0..1 esclusi.

## Valore restituito

K-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi esclusi.

## Osservazioni:

- Se la colonna è vuota, viene restituito BLANK().
- Se k è uguale a zero o è vuoto, la classificazione percentile  $1/(n+1)$  restituisce il valore più piccolo. Se è zero, non è compreso nell'intervallo e viene restituito un errore.
- Se k non è un valore numerico o non compreso nell'intervallo tra 0 e 1, viene restituito un errore.

- Se k non è un multiplo di  $1/(n + 1)$ , PERCENTILE.EXC esegue l'interpolazione per determinare il valore in corrispondenza del k-esimo percentile.
- PERCENTILE.EXC esegue l'interpolazione quando il valore del percentile specificato è compreso tra due valori nella matrice. Se la funzione non è in grado di eseguire l'interpolazione per il k percentile specificato, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[PERCENTILEX.EXC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PERCENTILE.INC

Articolo • 20/10/2023

Restituisce il k-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi inclusi.

Per restituire il numero percentile di un'espressione valutata per ogni riga in una tabella, usare [PERCENTILEX.INC](#).

## Sintassi

DAX

`PERCENTILE.INC(<column>, <k>)`

## Parametri

Termine	Definizione
colonna	Colonna contenente i valori che definiscono la condizione relativa.
k	Valore percentile compreso nell'intervallo 0..1 inclusi.

## Valore restituito

K-esimo dato percentile dei valori in un intervallo, dove k è nell'intervallo 0..1, estremi inclusi.

## Osservazioni:

- Se la colonna è vuota, viene restituito BLANK().
- Se k è uguale a zero o è vuoto, la classificazione percentile  $1/(n+1)$  restituisce il valore più piccolo. Se è zero, non è compreso nell'intervallo e viene restituito un errore.
- Se k non è un valore numerico o non compreso nell'intervallo tra 0 e 1, viene restituito un errore.

- Se  $k$  non è un multiplo di  $1/(n + 1)$ , PERCENTILE.INC esegue l'interpolazione per determinare il valore in corrispondenza del  $k$ -esimo percentile.
- PERCENTILE.INC esegue l'interpolazione quando il valore del percentile specificato è compreso tra due valori nella matrice. Se la funzione non è in grado di eseguire l'interpolazione per il  $k$  percentile specificato, viene restituito un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[PERCENTILEX.INC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PERCENTILEX.EXC

Articolo • 20/10/2023

Restituisce il numero percentile di un'espressione valutata per ogni riga di una tabella.

Per restituire il percentile dei numeri in una colonna, usare la [funzione PERCENTILE.EXC](#).

## Sintassi

DAX

```
PERCENTILEX.EXC(<table>, <expression>, k)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.
k	Valore percentile desiderato compreso nell'intervallo da 0 a 1 esclusi.

## Valore restituito

Numero percentile di un'espressione valutata per ogni riga di una tabella.

## Osservazioni:

- Se k è uguale a zero o è vuoto, la classificazione percentile  $1/(n+1)$  restituisce il valore più piccolo. Se è zero, non è compreso nell'intervallo e viene restituito un errore.
- Se k non è un valore numerico o non compreso nell'intervallo tra 0 e 1, viene restituito un errore.
- Se k non è un multiplo di  $1/(n + 1)$ , PERCENTILEX.EXC esegue l'interpolazione per determinare il valore in corrispondenza del k-esimo percentile.
- PERCENTILEX.EXC esegue l'interpolazione quando il valore del percentile specificato è compreso tra due valori nella matrice. Se la funzione non è in grado

di eseguire l'interpolazione per il k percentile specificato, viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[PERCENTILE.EXC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PERCENTILEX.INC

Articolo • 20/10/2023

Restituisce il numero percentile di un'espressione valutata per ogni riga di una tabella.

Per restituire il percentile dei numeri in una colonna, usare [PERCENTILE.INC](#).

## Sintassi

DAX

```
PERCENTILEX.INC(<table>, <expression>;, k)
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della tabella.
k	Valore percentile desiderato compreso nell'intervallo da 0 a 1 inclusi.

## Valore restituito

Numero percentile di un'espressione valutata per ogni riga di una tabella.

## Osservazioni:

- Se k è uguale a zero o è vuoto, la classificazione percentile  $1/(n - 1)$  restituisce il valore più piccolo. Se è zero, non è compreso nell'intervallo e viene restituito un errore.
- Se k non è un valore numerico o non compreso nell'intervallo tra 0 e 1, viene restituito un errore.
- Se k non è un multiplo di  $1/(n - 1)$ , PERCENTILEX.EXC esegue l'interpolazione per determinare il valore in corrispondenza del k-esimo percentile.
- PERCENTILEX.INC esegue l'interpolazione quando il valore del percentile specificato è compreso tra due valori nella matrice. Se la funzione non è in grado

di eseguire l'interpolazione per il k percentile specificato, viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[PERCENTILE.INC](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PERMUT

Articolo • 20/10/2023

Restituisce il numero delle permutazioni per un determinato numero di oggetti che è possibile selezionare da oggetti numerici. Una permutazione è un qualsiasi set o subset di oggetti o eventi il cui ordine interno sia significativo. Le permutazioni sono diverse dalle combinazioni, il cui ordine interno non è significativo. Usare questa funzione per i calcoli delle probabilità, come quelli che si eseguono per le lotterie.

## Sintassi

DAX
<code>PERMUT(number, number_chosen)</code>

## Parametri

Termine	Definizione
number	Required. Numero intero che descrive il numero di oggetti.
number_chosen	Required. Numero intero che descrive il numero di oggetti in ogni permutazione.

## Valore restituito

Restituisce il numero delle permutazioni per un determinato numero di oggetti che è possibile selezionare da oggetti numerici.

## Osservazioni:

- Entrambi gli argomenti vengono troncati in numeri interi.
- Se number o number\_chosen è un valore non numerico, PERMUT restituisce il valore di errore #VALUE! .
- Se number  $\leq 0$  o se number\_chosen < 0, PERMUT restituisce il valore di errore #NUM! .
- Se number < number\_chosen, PERMUT restituisce il valore di errore #NUM! .

- L'equazione per il calcolo del numero di permutazioni è:

$$P_{k,n} = \frac{n!}{(n - k)!}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente include le permutazioni possibili per un gruppo di 3 oggetti, in cui ne vengono scelti 2:

DAX

```
= PERMUT(3,2)
```

Risultato:

6

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# POISSON.DIST

Articolo • 20/10/2023

Restituisce la distribuzione di probabilità di Poisson. La distribuzione di Poisson viene in genere applicata per la previsione del numero di eventi in un arco di tempo specifico, come il numero di automobili che transitano per un casello autostradale in 1 minuto.

## Sintassi

DAX
<code>POISSON.DIST(x,mean,cumulative)</code>

## Parametri

Termine	Definizione
x	Required. Numero di eventi.
mean	Required. Valore numerico previsto.
cumulative	Required. Valore logico che determina la forma della distribuzione di probabilità restituita. Se cumulative è TRUE, POISSON.DIST restituisce la probabilità cumulativa di Poisson indicante la probabilità che il numero degli eventi casuali sia compreso tra zero e x inclusi. Se è FALSE, restituisce la funzione massa di probabilità di Poisson indicante la probabilità che il numero di eventi sia esattamente uguale a x.

## Valore restituito

Restituisce la distribuzione di probabilità di Poisson.

## Osservazioni:

- Se x non è un numero intero, il valore viene arrotondato.
- Se x o mean non è un valore numerico, POISSON.DIST restituisce il valore di errore #VALUE! .
- Se x < 0, POISSON.DIST restituisce il valore di errore #NUM! .
- Se mean < 0, POISSON.DIST restituisce il valore di errore #NUM! .

- La funzione POISSON.DIST viene calcolata come segue.

- Per cumulative = FALSE:

$$\text{POISSON} = \frac{e^{-\lambda} \lambda^x}{x!}$$

- Per cumulative = TRUE:

$$\text{CUMPOISSON} = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^x}{k!}$$

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# RANK.EQ

Articolo • 20/10/2023

Restituisce il rango di un numero in un elenco di numeri.

## Sintassi

DAX

```
RANK.EQ(<value>, <columnName>[, <order>])
```

## Parametri

Termine	Definizione
value	Qalsiasi espressione DAX che restituisce un singolo valore scalare di cui è necessario trovare il rango. L'espressione deve essere valutata una sola volta, prima che la funzione venga valutata e il valore passato all'elenco di argomenti.
columnName	Nome di una colonna esistente in base alla quale verranno determinati i ranghi. Non può essere un'espressione o una colonna creata usando queste funzioni; ADDCOLUMNS, ROW o SUMMARIZE.
order	(facoltativo) Valore che specifica come classificare <i>number</i> , dal più basso al più alto o dal più alto al più basso:

## Valori di order

value	valore alternativo	Descrizione
0 (zero)	FALSE	Ranghi di <i>columnName</i> in ordine decrescente. Se <i>value</i> è uguale al numero più alto in <i>columnName</i> , RANK.EQ è pari a 1.
1	TRUE	Ranghi di <i>columnName</i> in ordine crescente. Se <i>value</i> è uguale al numero più basso in <i>columnName</i> , RANK.EQ è pari a 1.

## Valore restituito

Numero indicante il rango di *value* tra i numeri in *columnName*.

## Osservazioni:

- *columnName* non può fare riferimento ad alcuna colonna creata usando queste funzioni: ADDCOLUMNS, ROW o SUMMARIZE.I
- Se *value* non è in *columnName* o il valore è vuoto, *RANK.EQ* restituisce un valore vuoto.
- I valori duplicati di *value* ricevono lo stesso valore di rango. Il valore di rango successivo assegnato sarà il valore del rango più il numero di valori duplicati. Se ad esempio cinque (5) valori sono associati a un rango pari a 11, il valore successivo riceverà un rango pari a 16 (11 + 5).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio 1

L'esempio seguente crea una colonna calcolata che classifica i valori di SalesAmount\_USD della tabella *InternetSales\_USD*, confrontandoli con tutti i numeri nella stessa colonna.

DAX

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD],  
InternetSales_USD[SalesAmount_USD])
```

## Esempio 2

L'esempio seguente classifica un subset di valori confrontandolo con un campione specifico. Si supponga di avere una tabella di studenti locali con i risultati raggiunti da ognuno in un test nazionale specifico e di avere anche l'intero set di voti ottenuti nel test nazionale. La colonna calcolata seguente indicherà la posizione nella classifica nazionale di ognuno degli studenti locali.

DAX

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# RANKX

Articolo • 20/10/2023

Restituisce il rango di un numero in un elenco di numeri per ogni riga nell'argomento *table*.

## Sintassi

DAX

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

## Parametri

### table

Qualsiasi espressione DAX che restituisce una tabella di dati in base ai quali viene valutata l'espressione.

### expression

Qualsiasi espressione DAX che restituisce un singolo valore scalare. L'espressione viene valutata per ogni riga di *table*, per generare tutti i possibili valori per la classificazione. Per informazioni sul comportamento della funzione quando *expression* restituisce BLANK, vedere la sezione Osservazioni.

### value

(facoltativo) Qualsiasi espressione DAX che restituisce un singolo valore scalare di cui è necessario trovare il rango. Per informazioni sul comportamento della funzione quando nell'espressione non viene trovato *value*, vedere la sezione Osservazioni.

Quando il parametro *value* viene omesso, viene invece usato il valore dell'espressione nella riga corrente.

### order

(facoltativo) Valore che specifica come classificare *value*, dal più basso al più alto o dal più alto al più basso:

value	valore alternativo	Descrizione
0 (zero)	FALSE	Ranghi in ordine decrescente dei valori di <i>expression</i> . Se <i>value</i> è uguale al numero più alto in <i>expression</i> , RANKX restituisce 1.

<b>value</b>	<b>valore alternativo</b>	<b>Descrizione</b>
		Si tratta del valore predefinito quando il parametro <i>order</i> viene omesso.
1	TRUE	Ranghi di <i>expression</i> in ordine crescente. Se <i>value</i> è uguale al numero più basso in <i>expression</i> , RANKX restituisce 1.

## ties

(facoltativo) Enumerazione che definisce come determinare la classificazione in caso di valori equivalenti.

<b>enumerazione</b>	<b>Descrizione</b>
Ignora	Il valore di rango successivo, dopo un valore equivalente, è il valore di rango del valore equivalente più il numero di valori associati. Se ad esempio cinque (5) valori sono associati a un rango pari a 11, il valore successivo riceverà un rango pari a 16 (11 + 5).  Si tratta del valore predefinito quando il parametro <i>ties</i> viene omesso.
Dense	Il valore di rango successivo, dopo un valore equivalente, è il valore di rango successivo. Se ad esempio cinque (5) valori sono associati a un rango pari a 11, il valore successivo riceverà un rango pari a 12.

## Valore restituito

Numero di rango di *value* tra tutti i possibili valori di *expression* valutato per tutte le righe di numeri di *table*.

## Osservazioni:

- Se *expression* o *value* restituisce BLANK, viene considerato come 0 (zero) per tutte le espressioni che restituiscono un numero o come testo vuoto per tutte le espressioni di testo.
- Se *value* non è compreso tra tutti i possibili valori di *expression*, RANKX aggiunge temporaneamente *value* ai valori di *expression* e valuta nuovamente RANKX per determinare il rango appropriato di *value*.
- Gli argomenti facoltativi potrebbero essere ignorati inserendo una virgola vuota (,) nell'elenco di argomenti, ad esempio RANKX(Inventory, [InventoryCost],,"Dense")

- Confronti correlati all'uguaglianza (=, <>, >= e <=) tra valori con il tipo di dati *numero decimale* possono potenzialmente restituire risultati imprevisti quando si usa la funzione RANKX. I risultati non corretti possono verificarsi perché i valori con tipo di dati *numero decimale* vengono archiviati come numeri a virgola mobile IEEE Standard 754 e presentano limitazioni intrinseche nella precisione. Per evitare risultati imprevisti, modificare il tipo di dati in *numero decimale fisso* o eseguire un arrotondamento forzato usando [ROUND](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La colonna calcolata seguente della tabella Products calcola la classificazione delle vendite per ogni prodotto nel canale Internet.

DAX

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# SAMPLE

Articolo • 20/10/2023

Restituisce un esempio di N righe dalla tabella specificata.

## Sintassi

DAX

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

## Parametri

Termine	Definizione
n_value	Numero di righe da restituire. Si tratta di qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto). Se viene immesso un valore (o un'espressione) non Integer, il risultato viene trasmesso come Integer.
table	qualsiasi espressione DAX che restituisce una tabella di dati da cui estrarre le 'n' righe di esempio.
orderBy_expression	qualsiasi espressione DAX scalare in cui il valore del risultato viene valutato per ogni riga di <i>table</i> .
order	(Facoltativo) Valore che specifica come ordinare valori <i>orderBy_expression</i> , crescente o decrescente: 0 (zero), ordina in ordine decrescente di valori di <i>order_by</i> . 1 classifica in ordine crescente <i>order_by</i> .

## Valore restituito

Una tabella costituita da un campione di N righe di *table* o una tabella vuota se *n\_value* è uguale o minore di 0 (zero). Se vengono specificati argomenti di OrderBy, il campione sarà stabile e deterministico e restituirà la prima riga, l'ultima riga e le righe distribuite uniformemente tra di esse. Se non viene specificato alcun ordinamento, il campione sarà casuale, non stabile e non deterministico.

## Osservazioni:

- Se n\_value è uguale o minore di 0 (zero), SAMPLE restituisce una tabella vuota.
  - Per evitare valori duplicati nel campione, la tabella specificata come secondo argomento deve essere raggruppata in base alla colonna usata per l'ordinamento.
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# STDEV.S

Articolo • 20/10/2023

Restituisce la deviazione standard di una popolazione di esempio.

## Sintassi

DAX

`STDEV.S(<ColumnName>)`

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione.

## Valore restituito

Numero che rappresenta la deviazione standard di un campione della popolazione.

## Eccezioni

### Osservazioni:

- STDEV.S presuppone che la colonna faccia riferimento a un campione della popolazione. Se i dati rappresentano l'intera popolazione, calcolare la deviazione standard usando STDEV.P.
- STDEV.S usa la formula seguente:

$$\sqrt{\sum(x - \bar{x})^2/(n-1)}$$

dove  $\bar{x}$  è il valore medio di  $x$  per la popolazione di esempio e  $n$  è la dimensione della popolazione.

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.

- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una misura che calcola la deviazione standard della colonna, SalesAmount\_USD, quando la tabella InternetSales\_USD è un campione della popolazione.

DAX

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# STDEV.P

Articolo • 20/10/2023

Restituisce la deviazione standard dell'intera popolazione.

## Sintassi

DAX

`STDEV.P(<ColumnName>)`

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione.

## Valore restituito

Numero che rappresenta la deviazione standard dell'intera popolazione.

## Osservazioni:

- STDEV.P presuppone che la colonna si riferisca all'intera popolazione. Se i dati rappresentano un campione della popolazione, calcolare la deviazione standard usando STDEV.S.
- STDEV.P usa la formula seguente:

$$\sqrt{\frac{\sum(x - \bar{x})^2}{n}}$$

dove  $\bar{x}$  è il valore medio di  $x$  per l'intera popolazione e  $n$  è la dimensione della popolazione.

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.
- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una misura che calcola la deviazione standard della colonna, SalesAmount\_USD, quando la tabella InternetSales\_USD è l'intera popolazione.

DAX

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# STDEVX.S

Articolo • 20/10/2023

Restituisce la deviazione standard di una popolazione di esempio.

## Sintassi

DAX

```
STDEVX.S(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).

## Valore restituito

Numero con la deviazione standard di una popolazione di esempio.

## Eccezioni

## Osservazioni:

- STDEVX.S valuta *expression* per ogni riga di *table* e restituisce la deviazione standard di *expression* presumendo che *table* faccia riferimento a un campione della popolazione. Se *table* rappresenta l'intera popolazione, calcolare la deviazione standard usando STDEVX.P.
- STDEVX.S usa la formula seguente:

$$\sqrt{\sum(x - \bar{x})^2/(n-1)}$$

dove  $\bar{x}$  è il valore medio di  $x$  per l'intera popolazione e  $n$  è la dimensione della popolazione.

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.
- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una colonna calcolata che stima la deviazione standard del prezzo unitario per prodotto per un campione di popolazione, quando la formula viene usata nella tabella Product.

DAX

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# STDEVX.P

Articolo • 20/10/2023

Restituisce la deviazione standard dell'intera popolazione.

## Sintassi

DAX

```
STDEVX.P(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).

## Valore restituito

Numero che rappresenta la deviazione standard dell'intera popolazione.

## Osservazioni:

- STDEVX.P valuta *expression* per ogni riga di *table* e restituisce la deviazione standard di *expression* presumendo che *table* faccia riferimento all'intera popolazione. Se i dati in *table* rappresentano un campione della popolazione, è consigliabile calcolare la deviazione standard usando invece STDEVX.S.
- STDEVX.P usa la formula seguente:

$$\sqrt{[\sum(x - \bar{x})^2/n]}$$

dove  $\bar{x}$  è il valore medio di  $x$  per l'intera popolazione e  $n$  è la dimensione della popolazione.

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.
- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una colonna calcolata che calcola la deviazione standard del prezzo unitario per prodotto, quando la formula viene usata nella tabella *Product*.

DAX

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# T.DIST

Articolo • 20/10/2023

Restituisce la distribuzione t di Student a una coda sinistra.

## Sintassi

DAX

```
T.DIST(X,Deg_freedom,Cumulative)
```

## Parametri

Termine	Definizione
X	Valore numerico per cui calcolare la distribuzione.
Deg_freedom	Intero che indica il numero di gradi di libertà.
Cumulativo	Valore logico che determina la forma della funzione. Se cumulative è TRUE, T.DIST restituisce la funzione di distribuzione cumulativa; se è FALSE, restituisce la funzione densità di probabilità.

## Valore restituito

Distribuzione t di Student a una coda sinistra.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Restituisce:

**[Value]**

0.994695326367377

## Vedi anche

[T.DIST.2T](#)

[T.DIST.RT](#)

[T.INV](#)

[T.INV.2t](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# T.DIST.2T

Articolo • 20/10/2023

Restituisce la distribuzione t di Student a due code.

## Sintassi

DAX

```
T.DIST.2T(X,Deg_freedom)
```

## Parametri

Termine	Definizione
X	Valore numerico per cui calcolare la distribuzione.
Deg_freedom	Intero che indica il numero di gradi di libertà.

## Valore restituito

Distribuzione t di Student a due code.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Resi

[Value]
0.054644929975921

## Vedi anche

[T.DIST](#)

[T.DIST.RT](#)

[T.INV](#)

[T.INV.2t](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# T.DIST.RT

Articolo • 20/10/2023

Restituisce la distribuzione t di Student a una coda destra.

## Sintassi

DAX

```
T.DIST.RT(X,Deg_freedom)
```

## Parametri

Termine	Definizione
X	Valore numerico per cui calcolare la distribuzione.
Deg_freedom	Intero che indica il numero di gradi di libertà.

## Valore restituito

Distribuzione t di Student a una coda destra.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { T.DIST.RT(1.95999998, 60) }
```

Resi

[Value]
0.0273224649879605

## Vedi anche

[T.DIST](#)

[T.DIST.2T](#)

[T.INV](#)

[T.INV.2t](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# T.INV

Articolo • 20/10/2023

Restituisce l'inversa della distribuzione t di Student a una coda sinistra.

## Sintassi

DAX

```
T.INV(Probability,Deg_freedom)
```

## Parametri

Termine	Definizione
Probability	Probabilità associata alla distribuzione t di Student.
Deg_freedom	Numero di gradi di libertà con cui caratterizzare la distribuzione.

## Valore restituito

Inversa della distribuzione t di Student a una coda sinistra.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { T.INV(0.75, 2) }
```

Resi

[Value]
0.816496580927726

## Vedi anche

[T.INV.2T](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# T.INV.2T

Articolo • 20/10/2023

Restituisce l'inversa della distribuzione t di Student a due code.

## Sintassi

DAX

```
T.INV.2T(Probability,Deg_freedom)
```

## Parametri

Termine	Definizione
Probability	Probabilità associata alla distribuzione t di Student.
Deg_freedom	Numero di gradi di libertà con cui caratterizzare la distribuzione.

## Valore restituito

Inversa della distribuzione t di Student a due code.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Resi

[Value]
0.606533075825759

## Vedi anche

[T.INV](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# VAR.S

Articolo • 20/10/2023

Restituisce la varianza di una popolazione di esempio.

## Sintassi

DAX

**VAR.S(<columnName>)**

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione.

## Valore restituito

Numero con la varianza di un campione di popolazione.

## Osservazioni:

- VAR.S presuppone che la colonna faccia riferimento a un campione della popolazione. Se i dati rappresentano l'intera popolazione, calcolare la varianza usando VAR.P.
- VAR.S usa la formula seguente:

$$\sum(x - \bar{x})^2/(n-1)$$

dove  $\bar{x}$  è il valore medio di  $x$  per il campione della popolazione

e  $n$  è la dimensione della popolazione

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.
- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una misura che calcola la varianza della colonna SalesAmount\_USD da InternetSales\_USD per un campione della popolazione.

DAX

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# VAR.P

Articolo • 20/10/2023

Restituisce la varianza dell'intera popolazione.

## Sintassi

DAX

**VAR.P(<columnName>)**

## Parametri

Termine	Definizione
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione.

## Valore restituito

Numero con la varianza dell'intera popolazione.

## Osservazioni:

- VAR.P presuppone che la colonna faccia riferimento all'intera popolazione. Se i dati rappresentano un campione della popolazione, calcolare la varianza usando VAR.S.
- VAR.P utilizza la formula seguente:

$$\sum(x - \bar{x})^2/n$$

dove  $\bar{x}$  è il valore medio di  $x$  per l'intera popolazione

e  $n$  è la dimensione della popolazione

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.
- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una misura che stima la varianza della colonna SalesAmount\_USD dalla tabella InternetSales\_USD per l'intera popolazione.

DAX

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# VARX.S

Articolo • 20/10/2023

Restituisce la varianza di una popolazione di esempio.

## Sintassi

DAX

```
VARX.S(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Qualsiasi espressione DAX che restituisce una tabella di dati.
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).

## Valore restituito

Numero che rappresenta la varianza di un campione di popolazione.

## Osservazioni:

- VARX.S valuta *expression* per ogni riga di *table* e restituisce la varianza di *expression*, con il presupposto che *table* faccia riferimento a un campione della popolazione. Se *table* rappresenta l'intera popolazione, è consigliabile calcolare la varianza usando VARX.P.
- VAR.S usa la formula seguente:

$$\sum(x - \bar{x})^2/(n-1)$$

dove  $\bar{x}$  è il valore medio di  $x$  per il campione della popolazione

e  $n$  è la dimensione della popolazione

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.

- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una colonna calcolata che stima la varianza del prezzo unitario per prodotto per un campione di popolazione, quando la formula viene usata nella tabella Product.

DAX

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# VARX.P

Articolo • 20/10/2023

Restituisce la varianza dell'intera popolazione.

## Sintassi

DAX

```
VARX.P(<table>, <expression>)
```

## Parametri

Termine	Definizione
table	Qualsiasi espressione DAX che restituisce una tabella di dati.
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).

## Valore restituito

Numero con la varianza dell'intera popolazione.

## Osservazioni:

- VARX.P valuta <expression> per ogni riga di <table> e restituisce la varianza di <expression> presumendo che <table> faccia riferimento all'intera popolazione. Se <table> rappresenta un campione della popolazione, calcolare la varianza usando VARX.S.
- VARX.P usa la formula seguente:

$$\sum(x - \bar{x})^2/n$$

dove  $\bar{x}$  è il valore medio di  $x$  per l'intera popolazione

e  $n$  è la dimensione della popolazione

- Le righe vuote vengono escluse da *columnName* e non sono considerate nei calcoli.

- Viene restituito un errore se *columnName* contiene meno di 2 righe non vuote
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente illustra la formula per una colonna calcolata che calcola la varianza del prezzo unitario per prodotto, quando la formula viene usata nella tabella Product

DAX

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# Funzioni di manipolazione delle tabelle

Articolo • 20/10/2023

Queste funzioni restituiscono una tabella o modificano le tabelle esistenti.

## In questa categoria

Funzione	Descrizione
<a href="#">ADDCOLUMNS</a>	Aggiunge colonne calcolate alla tabella o all'espressione di tabella specificata.
<a href="#">ADDMISSINGITEMS</a>	Aggiunge combinazioni di elementi provenienti da più colonne a una tabella, se non esistono già.
<a href="#">CROSSJOIN</a>	Restituisce una tabella contenente il prodotto cartesiano di tutte le righe da tutte le tabelle negli argomenti.
<a href="#">CURRENTGROUP</a>	Restituisce un set di righe dall'argomento table di un'espressione GROUPBY.
<a href="#">DATATABLE</a>	Fornisce un meccanismo per dichiarare un set inline di valori di dati.
<a href="#">DETAILROWS</a>	Valuta un'espressione di righe di dettaglio definita per una misura e restituisce i dati.
<a href="#">DISTINCT column</a>	Restituisce una tabella a singola colonna contenente i valori distinti della colonna specificata.
<a href="#">DISTINCT table</a>	Restituisce una tabella rimuovendo le righe duplicate da un'altra tabella o espressione.
<a href="#">EXCEPT</a>	Restituisce le righe di una tabella non visualizzate in un'altra tabella.
<a href="#">FILTERS</a>	Restituisce una tabella di valori applicati direttamente come filtri a <i>columnName</i> .
<a href="#">GENERATE</a>	Restituisce una tabella con il prodotto cartesiano tra ogni riga nel parametro <i>table1</i> e la tabella risultante dalla valutazione del parametro <i>table2</i> nel contesto della riga corrente dal parametro <i>table1</i> .
<a href="#">GENERATEALL</a>	Restituisce una tabella con il prodotto cartesiano tra ogni riga nel parametro <i>table1</i> e la tabella risultante dalla valutazione del parametro <i>table2</i> nel contesto della riga corrente dal parametro <i>table1</i> .

Funzione	Descrizione
<a href="#">GENERATESERIES</a>	Restituisce una tabella a colonna singola contenente i valori di una serie aritmetica.
<a href="#">GROUPBY</a>	Simile alla funzione SUMMARIZE, GROUPBY non esegue alcuna operazione CALCULATE implicita per le colonne di estensione che aggiunge.
<a href="#">IGNORE</a>	Modifica SUMMARIZECOLUMNS omettendo espressioni specifiche dalla valutazione BLANK/NULL.
<a href="#">INTERSECT</a>	Restituisce l'intersezione di riga di due tabelle, mantenendo i duplicati.
<a href="#">NATURALINNERJOIN</a>	Esegue un inner join di una tabella con un'altra tabella.
<a href="#">NATURALLEFTOUTERJOIN</a>	Esegue un join di LeftTable con RightTable.
<a href="#">ROLLUP</a>	Modifica il comportamento di SUMMARIZE aggiungendo righe di rollup al risultato per le colonne definite dal parametro groupBy_columnName.
<a href="#">ROLLUPADDISSUBTOTAL</a>	Modifica il comportamento di SUMMARIZECOLUMNS aggiungendo righe di rollup/subtotali al risultato in base alle colonne groupBy_columnName.
<a href="#">ROLLUPISSUBTOTAL</a>	Abbinia i gruppi di rollup alla colonna aggiunta da ROLLUPADDISSUBTOTAL all'interno di un'espressione ADDMISSINGITEMS.
<a href="#">ROLLUPGROUP</a>	Modifica il comportamento di SUMMARIZE e SUMMARIZECOLUMNS aggiungendo righe di rollup al risultato per le colonne definite dal parametro groupBy_columnName.
<a href="#">ROW</a>	Restituisce una tabella con una singola riga contenente i valori risultanti dalle espressioni assegnate a ogni colonna.
<a href="#">SELECTCOLUMNS</a>	Aggiunge colonne calcolate alla tabella o all'espressione di tabella specificata.
<a href="#">SUBSTITUTEWITHINDEX</a>	Restituisce una tabella che rappresenta un semi join di sinistra delle due tabelle fornite come argomenti.
<a href="#">SUMMARIZE</a>	Restituisce una tabella di riepilogo per i totali richiesti su un set di gruppi.
<a href="#">SUMMARIZECOLUMNS</a>	Restituisce una tabella di riepilogo su un set di gruppi.
<a href="#">Costruttore di tabella</a>	Restituisce una tabella di una o più colonne.
<a href="#">TOPN</a>	Restituisce le prime N righe della tabella specificata.

Funzione	Descrizione
TREATAS	Applica il risultato di un'espressione di tabella come filtri alle colonne di una tabella non correlata.
UNION	Crea una tabella di unione (join) da una coppia di tabelle.
VALUES	Restituisce una tabella a singola colonna contenente i valori distinti della tabella o colonna specificata.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ADDCOLUMNS

Articolo • 20/10/2023

Aggiunge colonne calcolate alla tabella o all'espressione di tabella specificata.

## Sintassi

DAX

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

## Parametri

Termino	Definizione
table	qualsiasi espressione DAX che restituisce una tabella di dati.
name	Nome assegnato alla colonna, racchiuso tra virgolette doppie.
expression	qualsiasi espressione DAX che restituisce un'espressione scalare, valutata per ogni riga del parametro <i>table</i> .

## Valore restituito

Tabella con tutte le colonne originali e quelle aggiunte.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce una versione estesa della tabella Product Category in cui sono inclusi i valori delle vendite totali provenienti dal canale rivenditore e dalle vendite Internet.

DAX

```
ADDCOLUMNS(ProductCategory,
    , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

Nella tabella seguente viene mostrata un'anteprima dei dati come ricevuta da qualsiasi funzione che prevede la ricezione di una tabella:

ProductCategory[ProductCategoryName]	ProductCategory[ProductCategoryAlternateKey]	ProductCategory[ProductCategoryKey]	[Internet Sales]	[Reseller Sales]
Bikes	1	1	25107749,77	6308
Componenti	2	2		1120
Clothing	3	3	306157,5829	1669
Accessori	4	4	640920,1338	5343

## Commenti e suggerimenti

Questa pagina è stata utile?



Si



No

# ADDMISSINGITEMS

Articolo • 20/10/2023

Aggiunge righe con valori vuoti a una tabella restituita da [SUMMARIZECOLUMNS](#).

## Sintassi

DAX

```
ADDMISSINGITEMS ( [<showAll_columnName> [, <showAll_columnName> [, ... ] ] ],  
<table> [, <groupBy_columnName> [, [<filterTable>] [, <groupBy_columnName>  
[, [<filterTable>] [, ... ] ] ] ] ] )
```

## Parametri

Termine	Definizione
showAll_columnName	(Facoltativo) Colonna per cui restituire elementi senza dati per le misure usate. Se viene omesso, vengono restituite tutte le colonne.
table	Tabella <a href="#">SUMMARIZECOLUMNS</a> .
groupBy_columnName	(Facoltativo) Colonna in base alla quale eseguire il raggruppamento nell'argomento table specificato.
filterTable	(Facoltativo) Espressione di tabella che definisce le righe restituite.

## Valore restituito

Una tabella con una o più colonne.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Con [SUMMARIZECOLUMNS](#)

Una tabella restituita da [SUMMARIZECOLUMNS](#) includerà solo le righe con valori. Eseguendo il wrapping di un'espressione [SUMMARIZECOLUMNS](#) all'interno di

un'espressione ADDMISSINGITEMS, vengono restituite anche le righe che non contengono valori.

## Esempio

Senza ADDMISSINGITEMS, la query seguente:

DAX

```
SUMMARIZECOLUMNS(  
    'Sales'[CustomerId],  
    "Total Qty", SUM ( Sales[TotalQty] )  
)
```

Restituisce:

Customerid	TotalQty
A	5
B	3
C	3
E	2

Con ADDMISSINGITEMS, la query seguente:

DAX

```
EVALUATE  
ADDMISSINGITEMS (  
    'Sales'[CustomerId],  
    SUMMARIZECOLUMNS(  
        'Sales'[CustomerId],  
        "Total Qty", SUM ( Sales[TotalQty] )  
,  
        'Sales'[CustomerId]  
)
```

Restituisce:

Customerid	TotalQty
A	5
B	3

Customerid	TotalQty
C	3
D	
E	2
F	

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CROSSJOIN

Articolo • 20/10/2023

Restituisce una tabella contenente il prodotto cartesiano di tutte le righe da tutte le tabelle negli argomenti. Le colonne nella nuova tabella sono tutte le colonne in tutte le tabelle dell'argomento.

## Sintassi

DAX

```
CROSSJOIN(<table>, <table>[ , <table>]...)
```

## Parametri

Termine	Definizione
table	Qualsiasi espressione DAX che restituisce una tabella di dati

## Valore restituito

Tabella contenente il prodotto cartesiano di tutte le righe da tutte le tabelle negli argomenti.

## Osservazioni:

- I nomi di colonna derivanti dagli argomenti *table* devono essere tutti diversi in tutte le tabelle, altrimenti viene restituito un errore.
- Il numero complessivo di righe restituite da CROSSJOIN() è uguale al prodotto del numero di righe da tutte le tabelle negli argomenti; inoltre, il numero complessivo di colonne nella tabella dei risultati è la somma del numero di colonne in tutte le tabelle. Ad esempio, se **TableA** ha **rA** righe e **cA** colonne, **TableB** ha **rB** righe e **cB** colonne e **TableC** ha **rC** righe e **cC** colonne, la tabella risultante ha **rA × rb × rC** righe e **cA + cB + cC** colonne.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Nell'esempio seguente vengono illustrati i risultati dell'applicazione di CROSSJOIN() a due tabelle: **Colors** e **Stationery**.

La tabella **Colors** contiene colori e motivi:

Colore	Modello
Rosso	Horizontal Stripe
Verde	Vertical Stripe
Blu	Crosshatch

La tabella **Stationery** contiene tipi di carattere e presentazione:

Carattere	Presentazione
serif	embossed
sans-serif	engraved

L'espressione per generare il Cross Join è riportata di seguito:

DAX

```
CROSSJOIN( Colors, Stationery)
```

Quando l'espressione precedente viene usata ogni volta che è prevista un'espressione di tabella, i risultati dell'espressione saranno come segue:

Colore	Modello	Carattere	Presentazione
Rosso	Horizontal Stripe	serif	embossed
Verde	Vertical Stripe	serif	embossed
Blu	Crosshatch	serif	embossed
Rosso	Horizontal Stripe	sans-serif	engraved
Verde	Vertical Stripe	sans-serif	engraved
Blu	Crosshatch	sans-serif	engraved

# Commenti e suggerimenti

Questa pagina è stata utile?



# CURRENTGROUP

Articolo • 20/10/2023

Restituisce un set di righe dall'argomento table di un'espressione [GROUPBY](#) che appartengono alla riga corrente del risultato di [GROUPBY](#).

## Sintassi

DAX

```
CURRENTGROUP ( )
```

## Parametri

None

## Valore restituito

Righe nell'argomento table della funzione [GROUPBY](#) corrispondenti a un solo gruppo di valori degli argomenti groupBy\_columnName.

## Osservazioni:

- Questa funzione può essere usata solo all'interno di un'espressione [GROUPBY](#).
- Questa funzione non accetta argomenti ed è supportata solo come primo argomento per una delle funzioni di aggregazione seguenti: [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#), [VARX.P](#).

## Esempio

Vedere [GROUPBY](#).

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# DATABASE

Articolo • 20/10/2023

Fornisce un meccanismo per dichiarare un set inline di valori di dati.

## Sintassi

DAX

```
DATABASE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}}...)
```

## Parametri

Termine	Definizione
ColumnName	qualsiasi espressione DAX che restituisce una tabella.
Tipo di dati	enumerazione che include: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
Valore	<p>Singolo argomento che usa la sintassi di Excel per una costante di matrice unidimensionale, annidata per fornire una matrice di matrici. Questo argomento rappresenta il set di valori di dati che saranno presenti nella tabella.</p> <p>Ad esempio, { {valori in riga1}, {valori in riga2}, {valori in riga3} } e così via }</p> <p>Dove {valori in riga1} è un set delimitato da virgolette di espressioni costanti, ovvero una combinazione di costanti e di alcune funzioni di base, tra cui DATE, TIME e BLANK, nonché un operatore più tra data e ora e un operatore meno unario, per consentire di esprimere valori negativi.</p> <p>I seguenti sono tutti valori validi: 3, -5, BLANK(), "2009-04-15 02:45:21". I valori non possono fare riferimento ad alcun elemento al di fuori dell'espressione immediata e non possono fare riferimento a colonne, tabelle, relazioni o altro.</p> <p>Un valore mancante verrà gestito in modo identico a BLANK(). Ad esempio, i seguenti sono gli stessi: {1,2,BLANK(),4} {1,2,,4}</p>

## Valore restituito

Tabella che dichiara un set inline di valori.

## Osservazioni:

- Diversamente da DATATABLE, il [costruttore Table](#) consente qualsiasi espressione scalare come valori di input.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

DAX

```
= DataTable("Name", STRING,
            "Region", STRING
            ,{
                {"User1", "East"},  

                {"User2", "East"},  

                {"User3", "West"},  

                {"User4", "West"},  

                {"User4", "East"}  

            }  

        )
```

## Commenti e suggerimenti

Questa pagina è stata utile?



# DETAILROWS

Articolo • 20/10/2023

Valuta un'espressione di righe di dettaglio definita per una misura e restituisce i dati.

## Sintassi

DAX

```
DETAILROWS([Measure])
```

## Parametri

Termine	Definizione
Misura	Nome di una misura.

## Valore restituito

Tabella con i dati restituiti dall'espressione di righe di dettaglio. Se non viene definita alcuna espressione di righe di dettaglio, vengono restituiti i dati per la tabella contenente la misura.

## Commenti e suggerimenti

Questa pagina è stata utile?



# DISTINCT (column)

Articolo • 20/10/2023

Restituisce una tabella a singola colonna contenente i valori distinti della colonna specificata. In altre parole, i valori duplicati vengono rimossi e vengono restituiti solo valori univoci.

## ⓘ Nota

Questa funzione non può essere usata per restituire valori in una cella o in una colonna di un foglio di calcolo. Viene invece annidata all'interno di una formula per ottenere un elenco di valori distinti che possono essere passati a un'altra funzione e quindi conteggiati, sommati o usati per altre operazioni.

## Sintassi

DAX

```
DISTINCT(<column>)
```

## Parametri

Termine	Definizione
colonna	Colonna da cui devono essere restituiti valori univoci o espressione che restituisce una colonna.

## Valore restituito

Colonna di valori univoci.

## Osservazioni:

- I risultati di DISTINCT variano in base al contesto di filtro corrente. Se, ad esempio, si usa la formula nell'esempio seguente per creare una misura, i risultati cambiano ogni volta che la tabella viene filtrata in modo da mostrare solo un'area o un periodo di tempo particolare.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Funzioni correlate

Esiste un'altra versione della funzione DISTINCT, ovvero [DISTINCT \(table\)](#), che restituisce una tabella rimuovendo le righe duplicate da un'altra tabella o espressione.

La funzione VALUES è simile a DISTINCT. Può anche essere usata per restituire un elenco di valori univoci e, in genere, restituisce esattamente gli stessi risultati di DISTINCT. In alcuni contesti, tuttavia, VALUES restituisce un valore speciale aggiuntivo. Per altre informazioni, vedere [Funzione VALUES](#).

## Esempio

Nella formula seguente viene conteggiato il numero di clienti univoci che hanno generato ordini sul canale Internet. La tabella seguente illustra i risultati possibili quando la formula viene aggiunta a un report.

```
DAX
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

Non è possibile incollare l'elenco di valori restituiti da DISTINCT direttamente in una colonna. I risultati della funzione DISTINCT vengono invece passati a un'altra funzione che conta, filtra o aggrega i valori usando l'elenco. Per semplificare al massimo l'esempio, in questo caso la tabella dei valori distinti è stata passata alla funzione COUNTROWS.

<b>Etichette di riga</b>	<b>Accessori</b>	<b>Bikes</b>	<b>Clothing</b>	<b>Grand Total</b>
2005		1013		1013
2006		2677		2677
2007	6792	4875	2867	9309
2008	9435	5451	4196	11377
Grand Total	15114	9132	6852	18484

Si noti inoltre che i risultati non sono additivi. In altre parole, il numero totale di clienti univoci nel 2007 non è la somma dei clienti univoci di Accessori, *Biciclette* e

*Abbigliamento* per tale anno. Questo avviene perché un cliente può essere conteggiato in più gruppi.

## Vedi anche

[Funzioni di filtro](#)

[Funzione FILTER](#)

[Funzione RELATED](#)

[Funzione VALUES](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DISTINCT (table)

Articolo • 20/10/2023

Restituisce una tabella rimuovendo le righe duplicate da un'altra tabella o espressione.

## Sintassi

DAX

```
DISTINCT(<table>)
```

## Parametri

Termine	Definizione
table	Tabella da cui devono essere restituite righe univoche. La tabella può essere anche un'espressione tramite cui viene restituita una tabella.

## Valore restituito

Tabella contenente solo righe distinte.

## Funzioni correlate

Esiste un'altra versione della funzione DISTINCT, ovvero [DISTINCT \(column\)](#), che accetta un nome di colonna come parametro di input.

## Esempio

La query seguente:

DAX

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

Restituisce la tabella:

[Value1]	[Value2]
1	A
2	B

## Vedi anche

[Funzioni di filtro](#)

[DISTINCT \(colonna\)](#)

[Funzione FILTER](#)

[Funzione RELATED](#)

[Funzione VALUES](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EXCEPT

Articolo • 20/10/2023

Restituisce le righe della prima tabella nell'espressione che non compaiono nella seconda tabella.

## Sintassi

DAX

```
EXCEPT(<table_expression1>, <table_expression2>)
```

## Parametri

Termine	Definizione
Table_expression	qualsiasi espressione DAX che restituisce una tabella.

## Valore restituito

Tabella contenente le righe di una tabella meno tutte le righe di un'altra tabella.

## Osservazioni:

- Se una riga compare in entrambe le tabelle, la riga e i relativi duplicati non sono presenti nel set di risultati. Se una riga compare solo in table\_expression1, la riga e i relativi duplicati saranno presenti nel set di risultati.
- I nomi delle colonne corrispondono ai nomi delle colonne in table\_expression1.
- La tabella restituita presenta una derivazione basata sulle colonne in table\_expression1, indipendentemente dalla derivazione delle colonne nella seconda tabella. Se, ad esempio, nella prima colonna della prima table\_expression è presente una derivazione della colonna di base C1 nel modello, la funzione EXCEPT ridurrà le righe in base alla disponibilità dei valori nella prima colonna della seconda table\_expression e manterrà intatta la derivazione sulla colonna di base C1.
- Le due tabelle devono avere lo stesso numero di colonne.

- Le colonne vengono confrontate in base al posizionamento e il confronto dei dati è senza coercizione del tipo.
- Il set di righe restituito dipende dall'ordine delle due espressioni.
- La tabella restituita non include colonne di tabelle correlate a table\_expression1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

States1

<b>Provincia</b>
A
B
B
B
C
D
D

States2

<b>Provincia</b>
B
C
D
D
D
E
E
E

Except(States1, States2)

Provincia
A

Except(States2, States1)

Provincia
E
E
E

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FILTERS

Articolo • 20/10/2023

Restituisce i valori applicati direttamente come filtri a *columnName*.

## Sintassi

DAX

```
FILTERS(<columnName>)
```

## Parametri

Termine	Descrizione
columnName	Nome di una colonna esistente con la sintassi DAX standard. Non può essere un'espressione.

## Valore restituito

Valori applicati direttamente come filtri a *columnName*.

## Osservazioni:

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente mostra come determinare il numero di filtri diretti di una colonna.

DAX

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

Questo esempio consente di scoprire quanti filtri diretti su ResellerSales\_USD [ProductKey] sono stati applicati al contesto in cui viene valutata l'espressione.

# Commenti e suggerimenti

Questa pagina è stata utile?



# GENERATE

Articolo • 20/10/2023

Restituisce una tabella con il prodotto cartesiano tra ogni riga nel parametro *table1* e la tabella risultante dalla valutazione del parametro *table2* nel contesto della riga corrente dal parametro *table1*.

## Sintassi

DAX

```
GENERATE(<table1>, <table2>)
```

## Parametri

Termine	Definizione
table1	qualsiasi espressione DAX che restituisce una tabella.
table2	qualsiasi espressione DAX che restituisce una tabella.

## Valore restituito

Tabella con il prodotto cartesiano tra ogni riga nel parametro *table1* e la tabella risultante dalla valutazione del parametro *table2* nel contesto della riga corrente dal parametro *table1*.

## Osservazioni:

- Se la valutazione del parametro *table2* per la riga corrente nel parametro *table1* restituisce una tabella vuota, la tabella dei risultati non conterrà la riga corrente dal parametro *table1*. Questo comportamento è diverso da GENERATEALL(), in cui la riga corrente da *table1* sarà inclusa nei risultati e le colonne corrispondenti a *table2* conterranno valori Null per questa riga.
- Tutti i nomi di colonna di *table1* e *table2* devono essere diversi, altrimenti viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente l'utente vuole una tabella di riepilogo delle vendite per area geografica e categoria prodotto per il canale dei rivenditori, come la tabella seguente:

<b>SalesTerritory[SalesTerritoryGroup]</b>	<b>ProductCategory[ProductName]</b>	<b>[Reseller Sales]</b>
Europa	Accessori	\$ 142.227,27
Europa	Bikes	\$ 9.970.200,44
Europa	Clothing	\$ 365.847,63
Europa	Componenti	\$ 2.214.440,19
America del Nord	Accessori	\$ 379.305,15
America del Nord	Bikes	\$ 52.403.796,85
America del Nord	Clothing	\$ 1.281.193,26
America del Nord	Componenti	\$ 8.882.848,05
Pacifico	Accessori	\$ 12.769,57
Pacifico	Bikes	\$ 710.677,75
Pacifico	Clothing	\$ 22.902,38
Pacifico	Componenti	\$ 108.549,71

La formula seguente produce la tabella riportata sopra:

DAX

```
GENERATE(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
    ,SUMMARIZE(ProductCategory
    , [ProductName]
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD]))
```

)  
)

- La prima istruzione SUMMARIZE, `SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])`, produce una tabella di gruppi territorio in cui ogni riga è un gruppo territorio, come illustrato di seguito:

<b>SalesTerritory[SalesTerritoryGroup]</b>
America del Nord
Europa
Pacifico
N/D

- La seconda istruzione SUMMARIZE, `SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))`, produce una tabella di gruppi di categoria prodotto con le vendite del rivenditore per ogni gruppo, come illustrato di seguito:

<b>ProductCategory[ProductCategoryName]</b>	<b>[Reseller Sales]</b>
Bikes	\$ 63.084.675,04
Componenti	\$ 11.205.837,96
Clothing	\$ 1.669.943,27
Accessori	\$ 534.301,99

- Tuttavia, quando si prende la tabella precedente e la si valuta nel contesto di ogni riga derivante dalla tabella gruppi territorio, si ottengono risultati diversi per ogni territorio.

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**    **No**

# GENERATEALL

Articolo • 20/10/2023

Restituisce una tabella con il prodotto cartesiano tra ogni riga nel parametro *table1* e la tabella risultante dalla valutazione del parametro *table2* nel contesto della riga corrente dal parametro *table1*.

## Sintassi

DAX
<code>GENERATEALL(&lt;table1&gt;, &lt;table2&gt;)</code>

## Parametri

Termine	Definizione
table1	qualsiasi espressione DAX che restituisce una tabella.
table2	qualsiasi espressione DAX che restituisce una tabella.

## Valore restituito

Tabella con il prodotto cartesiano tra ogni riga nel parametro *table1* e la tabella risultante dalla valutazione del parametro *table2* nel contesto della riga corrente dal parametro *table1*.

## Osservazioni:

- Se la valutazione di *table2* per la riga corrente in *table1* restituisce una tabella vuota, la riga corrente da *table1* sarà inclusa nei risultati e le colonne corrispondenti a *table2* conterranno valori Null per questa riga. Questo comportamento è diverso da GENERATEALL() in cui la riga corrente da *table1* non sarà inclusa nei risultati.
- Tutti i nomi di colonna di *table1* e *table2* devono essere diversi, altrimenti viene restituito un errore.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente l'utente vuole una tabella di riepilogo delle vendite per area geografica e categoria prodotto per il canale dei rivenditori, come la tabella seguente:

<b>SalesTerritory[SalesTerritoryGroup]</b>	<b>ProductCategory[ProductName]</b>	<b>[Reseller Sales]</b>
Europa	Accessori	\$ 142.227,27
Europa	Bikes	\$ 9.970.200,44
Europa	Clothing	\$ 365.847,63
Europa	Componenti	\$ 2.214.440,19
N/D	Accessori	
N/D	Bikes	
N/D	Clothing	
N/D	Componenti	
America del Nord	Accessori	\$ 379.305,15
America del Nord	Bikes	\$ 52.403.796,85
America del Nord	Clothing	\$ 1.281.193,26
America del Nord	Componenti	\$ 8.882.848,05
Pacifico	Accessori	\$ 12.769,57
Pacifico	Bikes	\$ 710.677,75
Pacifico	Clothing	\$ 22.902,38
Pacifico	Componenti	\$ 108.549,71

La formula seguente produce la tabella riportata sopra:

## DAX

```
GENERATEALL(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
    ,SUMMARIZE(ProductCategory
    , [ProductCategoryName]
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD])
    )
)
```

1. La prima istruzione SUMMARIZE produce una tabella di gruppi territorio in cui ogni riga è un gruppo territorio, come illustrato di seguito:

SalesTerritory[SalesTerritoryGroup]
America del Nord
Europa
Pacifico
N/D

2. La seconda istruzione SUMMARIZE produce una tabella di gruppi di categoria prodotto con le vendite del rivenditore per ogni gruppo, come illustrato di seguito:

ProductCategory[ProductCategoryName]	[Reseller Sales]
Bikes	\$ 63.084.675,04
Componenti	\$ 11.205.837,96
Clothing	\$ 1.669.943,27
Accessori	\$ 534.301,99

3. Tuttavia, quando si prende la tabella precedente e la si valuta nel contesto di ogni riga derivante dalla tabella gruppi territorio, si ottengono risultati diversi per ogni territorio.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# GENERATESERIES

Articolo • 20/10/2023

Restituisce una tabella a colonna singola contenente i valori di una serie aritmetica, ossia una sequenza di valori in cui ognuno è diverso da quello precedente per una quantità costante. Il nome della colonna restituita è Value.

## Sintassi

DAX

```
GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

## Parametri

Termine	Definizione
startValue	Valore iniziale usato per generare la sequenza.
endValue	Valore finale usato per generare la sequenza.
incrementValue	(facoltativo) Valore di incremento della sequenza. Se non specificato, il valore predefinito è 1.

## Valore restituito

Tabella a colonna singola contenente i valori di una serie aritmetica. Il nome della colonna è Value.

## Osservazioni:

- Quando endValue è minore di startValue, viene restituita una tabella vuota.
- IncrementValue deve essere un valore positivo.
- La sequenza si interrompe in corrispondenza dell'ultimo valore minore o uguale a endValue.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio 1

La query DAX seguente:

DAX
EVALUATE GENERATESERIES(1, 5)

Restituisce la tabella seguente con una singola colonna:

[Value]
1
2
3
4
5

# Esempio 2

La query DAX seguente:

DAX
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)

Restituisce la tabella seguente con una singola colonna:

[Value]
1.2
1.6
2
2.4

# Esempio 3

La query DAX seguente:

DAX

```
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

Restituisce la tabella seguente con una singola colonna:

[Value]
10
10.5
11
11.5
12

## Commenti e suggerimenti

Questa pagina è stata utile?



# GROUPBY

Articolo • 20/10/2023

La funzione GROUPBY è simile alla funzione [SUMMARIZE](#). Tuttavia, GROUPBY non esegue alcuna funzione [CALCULATE](#) implicita per le colonne di estensione che aggiunge. GROUPBY consente di usare una nuova funzione, [CURRENTGROUP](#), all'interno delle funzioni di aggregazione nelle colonne di estensione che aggiunge. La funzione GROUPBY viene usata per eseguire più aggregazioni in un'unica analisi di tabella.

## Sintassi

DAX

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, ...]]] [, <name>, <expression> [, <name>, <expression> [, ...]]])
```

## Parametri

Termine	Definizione
table	qualsiasi espressione DAX che restituisce una tabella di dati.
groupBy_columnName	Nome di una colonna esistente nella tabella (o in una tabella correlata) in base alla quale devono essere raggruppati i dati. Questo parametro non può essere un'espressione.
name	Nome assegnato a una nuova colonna che viene aggiunta all'elenco delle colonne GroupBy, racchiuso tra virgolette doppie.
expression	Una delle funzioni di aggregazione X il cui primo argomento è CURRENTGROUP(). Per l'elenco completo delle funzioni di aggregazione X supportate, vedere la sezione Con CURRENTGROUP di seguito.

## Valore restituito

Una tabella con le colonne selezionate per gli argomenti groupBy\_columnName e le colonne di estensione designate dagli argomenti name.

## Osservazioni:

- La funzione GROUPBY esegue le azioni seguenti:

1. Inizia con la tabella specificata (e tutte le tabelle correlate nella direzione "a-uno").
  2. Crea un raggruppamento usando tutte le colonne GroupBy (che devono esistere nella tabella dal passaggio 1).
  3. Ogni gruppo è una riga nel risultato, ma rappresenta un set di righe nella tabella originale.
  4. Per ogni gruppo, valuta le colonne di estensione aggiunte. Diversamente dalla funzione SUMMARIZE, non viene eseguita una funzione CALCULATE implicita e il gruppo non viene inserito nel contesto di filtro.
- Ogni colonna per la quale si definisce un nome deve avere un'espressione corrispondente, altrimenti viene restituito un errore. Il primo argomento, name, definisce il nome della colonna nei risultati. Il secondo argomento, expression, definisce il calcolo eseguito per ottenere il valore di ogni riga della colonna.
  - groupBy\_columnName deve essere presente nella tabella o in una tabella correlata.
  - Ogni nome deve essere racchiuso tra virgolette doppie.
  - La funzione raggruppa un set selezionato di righe in un set di righe di riepilogo in base ai valori di una o più colonne groupBy\_columnName. Viene restituita una riga per ogni gruppo.
  - La funzione GROUPBY viene usata principalmente per eseguire aggregazioni su risultati intermedi da espressioni di tabella DAX. Per aggregazioni efficienti su tabelle fisiche nel modello, provare a usare la funzione [SUMMARIZEOLUMNS](#) o [SUMMARIZE](#).
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Con CURRENTGROUP

La funzione [CURRENTGROUP](#) può essere usata solo in un'espressione che definisce una colonna di estensione all'interno della funzione GROUPBY. In effetti, [CURRENTGROUP](#) restituisce un set di righe dall'argomento table di GROUPBY che appartengono alla riga corrente del risultato di GROUPBY. La funzione [CURRENTGROUP](#) non accetta argomenti ed è supportata solo come primo argomento per una delle funzioni di aggregazione seguenti: [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#), [VARX.P](#).

## Esempio

L'esempio seguente calcola inizialmente le vendite totali raggruppate per paese e categoria di prodotti rispetto alle tabelle fisiche usando la funzione [SUMMARIZECOLUMNNS](#). Usa quindi la funzione GROUPBY per analizzare il risultato intermedio dal primo passaggio in modo da individuare le vendite massime in ogni paese tra le categorie di prodotti.

```
DAX

DEFINE
    VAR SalesByCountryAndCategory =
        SUMMARIZECOLUMNNS(
            Geography[Country],
            Product[Category],
            "Total Sales", SUMX(Sales, Sales[Price] * Sales[Qty])
        )

EVALUATE
    GROUPBY(
        SalesByCountryAndCategory,
        Geography[Country],
        "Max Sales", MAXX(CURRENTGROUP(), [Total Sales])
    )
```

## Vedi anche

[Funzione SUMMARIZE](#)

[Funzione SUMMARIZECOLUMNNS](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# IGNORE

Articolo • 20/10/2023

Modifica il comportamento della funzione [SUMMARIZECOLUMNS](#) omettendo espressioni specifiche dalla valutazione BLANK/NULL. Le righe per le quali tutte le espressioni che non usano IGNORE restituiscono BLANK/NULL verranno escluse indipendentemente dal fatto che le espressioni che usano IGNORE restituiscano o meno BLANK/NULL. Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZECOLUMNS](#).

## Sintassi

DAX

```
IGNORE(<expression>)
```

Con [SUMMARIZECOLUMNS](#)

DAX

```
SUMMARIZECOLUMNS(<groupBy_columnName>[ , < groupBy_columnName >]...,  
[<filterTable>]...[ , <name>, IGNORE(...)]...)
```

## Parametri

Termine	Definizione
expression	qualsiasi espressione DAX che restituisce un singolo valore (non una tabella).

## Valore restituito

La funzione non restituisce un valore.

## Osservazioni:

IGNORE può essere usato solo come argomento di espressione per [SUMMARIZECOLUMNS](#).

# Esempio

Vedere [SUMMARIZECOLUMNS](#).

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# INTERSECT

Articolo • 20/10/2023

Restituisce l'intersezione di riga di due tabelle, mantenendo i duplicati.

## Sintassi

DAX

```
INTERSECT(<table_expression1>, <table_expression2>)
```

## Parametri

Termine	Definizione
Table_expression	qualsiasi espressione DAX che restituisce una tabella.

## Valore restituito

Tabella che contiene tutte le righe di table\_expression1 che sono presenti anche in table\_expression2

## Eccezioni

## Osservazioni:

- La funzione Intersect non è commutativa. In generale, Intersect(T1, T2) ha un set di risultati diverso da Intersect(T2, T1).
- Le righe duplicate vengono mantenute. Se una riga è presente in table\_expression1 e table\_expression2, tale riga e tutti i duplicati in table\_expression\_1 vengono inclusi nel set di risultati.
- I nomi delle colonne corrispondono ai nomi delle colonne in table\_expression1.
- La tabella restituita presenta una derivazione basata sulle colonne in table\_expression1, indipendentemente dalla derivazione delle colonne nella seconda tabella. Se, ad esempio, nella prima colonna della prima table\_expression

è presente una derivazione della colonna di base C1 nel modello, la funzione INTERSECT ridurrà le righe in base all'intersezione nella prima colonna della seconda table\_expression e manterrà intatta la derivazione sulla colonna di base C1.

- Le colonne vengono confrontate in base al posizionamento e il confronto dei dati è senza coercizione del tipo.
- La tabella restituita non include colonne di tabelle correlate a table\_expression1.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

States1

Provincia
A
A
B
B
C
D
D

States2

Provincia
B
C
D
D
D

Provincia
E

Intersect(States1, States2)

Provincia
B
B
B
C
D
D

Intersect(States2, States1)

Provincia
B
C
D
D
D

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# NATURALINNERJOIN

Articolo • 20/10/2023

Esegue un inner join di una tabella con un'altra tabella.

## Sintassi

DAX

```
NATURALINNERJOIN(<LeftTable>, <RightTable>)
```

## Parametri

Termine	Definizione
LeftTable	Espressione di tabella che definisce la tabella sul lato sinistro del join.
RightTable	Espressione di tabella che definisce la tabella sul lato destro del join.

## Valore restituito

Tabella che include solo le righe per le quali i valori delle colonne in comune specificati sono presenti in entrambe le tabelle. La tabella restituita includerà le colonne in comune della tabella di sinistra e altre colonne di entrambe le tabelle.

## Osservazioni:

- Le tabelle vengono unite in join e nelle colonne comuni (in base al nome) nelle due tabelle. Se le due tabelle non hanno nomi di colonna in comune, viene restituito un errore.
- L'ordinamento dei risultati non è garantito.
- Le colonne da unire in join devono avere lo stesso tipo di dati in entrambe le tabelle.
- Vengono unite in join solo le colonne della stessa tabella di origine (con la stessa derivazione). Ad esempio, Products[ProductID], WebSales[ProductID], StoreSales[ProductID] con relazioni molti-a-uno tra WebSales e StoreSales e la

tabella Products in base alla colonna ProductID, le tabelle WebSales e StoreSales vengono unite in join su [ProductID].

- Durante il join viene usata una semantica di confronto rigida. Non esiste alcuna coercizione del tipo. Ad esempio, 1 non è uguale a 1.0.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[NATURALLEFTOUTERJOIN](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NATURALLEFTOUTERJOIN

Articolo • 20/10/2023

Esegue un join della LeftTable con la RightTable utilizzando la semantica Left Outer Join.

## Sintassi

DAX

```
NATURALLEFTOUTERJOIN(<LeftTable>, <RightTable>)
```

## Parametri

Termino	Definizione
LeftTable	Espressione di tabella che definisce la tabella sul lato sinistro del join.
RightTable	Espressione di tabella che definisce la tabella sul lato destro del join.

## Valore restituito

Una tabella che include solo le righe di RightTable per le quali i valori nelle colonne comuni specificate sono presenti anche in LeftTable. La tabella restituita includerà le colonne in comune della tabella di sinistra e le altre colonne di entrambe le tabelle.

## Osservazioni:

- Le tabelle vengono unite in join e nelle colonne comuni (in base al nome) nelle due tabelle. Se le due tabelle non hanno nomi di colonna in comune, viene restituito un errore.
- L'ordinamento dei risultati non è garantito.
- Le colonne da unire in join devono avere lo stesso tipo di dati in entrambe le tabelle.
- Vengono unite in join solo le colonne della stessa tabella di origine (con la stessa derivazione). Ad esempio, Products[ProductID], WebSales[ProductID], StoreSales[ProductID] con relazioni molti-a-uno tra WebSales e StoreSales e la

tabella Products in base alla colonna ProductID, le tabelle WebSales e StoreSales vengono unite in join su [ProductID].

- Durante il join viene usata una semantica di confronto rigida. Non esiste alcuna coercizione del tipo. Ad esempio, 1 non è uguale a 1.0.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[NATURALINNERJOIN](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ROLLUP

Articolo • 20/10/2023

Modifica il comportamento della funzione [SUMMARIZE](#) aggiungendo righe di rollup al risultato per le colonne definite dal parametro `groupBy_columnName`. Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZE](#).

## Sintassi

DAX

```
ROLLUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

Con SUMMARIZE

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[ , <groupBy_columnName>]...[ ,  
ROLLUP(<groupBy_columnName>[ ,< groupBy_columnName>...]])[, <name>,  
<expression>]...)
```

## Parametri

Termine	Definizione
<code>groupBy_columnName</code>	Nome completo di una colonna esistente o funzione ROLLUPGROUP da usare per creare gruppi di riepilogo in base ai valori trovati. Questo parametro non può essere un'espressione.

## Valore restituito

Questa funzione non restituisce un valore. Specifica solo il set di colonne di cui calcolare il subtotale.

## Osservazioni:

Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZE](#).

## Esempio

Vedere [SUMMARIZE](#).

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ROLLUPADDISSUBTOTAL

Articolo • 20/10/2023

Modifica il comportamento della funzione [SUMMARIZECOLUMNS](#) aggiungendo righe di rollup/subtotali al risultato in base alle colonne groupBy\_columnName. Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZECOLUMNS](#).

## Sintassi

DAX

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, <groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, ... ] ] ] )
```

## Parametri

Termine	Definizione
grandtotalFilter	(Facoltativo) Filtro da applicare al livello di totale complessivo.
groupBy_columnName	Nome di una colonna esistente usata per creare gruppi di riepilogo in base ai valori trovati. Non può essere un'espressione.
name	Nome di una colonna ISSUBTOTAL. I valori della colonna vengono calcolati usando la funzione ISSUBTOTAL.
groupLevelFilter	(Facoltativo) Filtro da applicare al livello corrente.

## Valore restituito

La funzione non restituisce un valore.

## Osservazioni:

None

## Esempio

Vedere [SUMMARIZECOLUMNS](#).

---

# Commenti e suggerimenti

Questa pagina è stata utile?



# ROLLUPGROUP

Articolo • 20/10/2023

Modifica il comportamento delle funzioni [SUMMARIZE](#) e [SUMMARIZECOLUMNS](#) aggiungendo righe di rollup al risultato per le colonne definite dal parametro `groupBy_columnName`. Questa funzione può essere usata solo all'interno di un'espressione [SUMMARIZE](#) o [SUMMARIZECOLUMNS](#).

## Sintassi

DAX

```
ROLLUPGROUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

## Parametri

Termine	Definizione
<code>groupBy_columnName</code>	Nome completo di una colonna esistente o funzione ROLLUPGROUP da usare per creare gruppi di riepilogo in base ai valori trovati. Questo parametro non può essere un'espressione.

## Valore restituito

Questa funzione non restituisce un valore. Contrassegna un set di colonne da considerare come gruppo singolo durante il calcolo del subtotale tramite [ROLLUP](#) o [ROLLUPADDISSTOTAL](#).

## Osservazioni:

ROLLUPGROUP può essere usata solo come argomento `groupBy_columnName` per [ROLLUP](#), [ROLLUPADDISSTOTAL](#) o [ROLLUPISSSTOTAL](#).

## Esempio

Vedere [SUMMARIZE](#) e [SUMMARIZECOLUMNS](#).

# Commenti e suggerimenti

Questa pagina è stata utile?



# ROLLUPISSUBTOTAL

Articolo • 20/10/2023

Associa i gruppi di rollup alla colonna aggiunta da [ROLLUPADDISSUBTOTAL](#). Questa funzione può essere usata solo all'interno di un'espressione [ADDMISSINGITEMS](#).

## Sintassi

DAX

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>, <isSubtotal_columnName> [, [<groupLevelFilter>] [, <groupBy_columnName>, <isSubtotal_columnName> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

## Parametri

Termine	Definizione
grandTotalFilter	(Facoltativo) Filtro da applicare al livello di totale complessivo.
groupBy_columnName	Nome di una colonna esistente usata per creare gruppi di riepilogo in base ai valori trovati. Non può essere un'espressione.
isSubtotal_columnName	Nome di una colonna ISSUBTOTAL. I valori della colonna vengono calcolati usando la funzione ISSUBTOTAL.
groupLevelFilter	(Facoltativo) Filtro da applicare al livello corrente.

## Valore restituito

None

## Osservazioni:

Questa funzione può essere usata solo all'interno di un'espressione [ADDMISSINGITEMS](#).

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Funzione ROW

Articolo • 20/10/2023

Restituisce una tabella con una singola riga contenente i valori risultanti dalle espressioni assegnate a ogni colonna.

## Sintassi

DAX

```
ROW(<name>, <expression>[ [,<name>, <expression>]...])
```

## Parametri

Termine	Definizione
name	Nome assegnato alla colonna, racchiuso tra virgolette doppie.
expression	qualsiasi espressione DAX che restituisce un singolo valore scalare da popolare. <i>name</i> .

## Valore restituito

Tabella con una sola riga

## Osservazioni:

- Gli argomenti devono essere sempre coppie di *name* e *expression*.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce una tabella con una sola riga con le vendite totali per Internet e i canali dei rivenditori.

DAX

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),  
    "Resellers Total Sales (USD)",  
    SUM(ResellerSales_USD[SalesAmount_USD]))
```

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SELECTCOLUMNS

Articolo • 20/10/2023

Restituisce una tabella con le colonne selezionate dalla tabella e nuove colonne specificate dalle espressioni DAX.

## Sintassi

DAX

```
SELECTCOLUMNS(<Table>, [<Name>], <Expression>, <Name>], ...)
```

## Parametri

Termine	Definizione
Tabella	Qualsiasi espressione DAX che restituisce una tabella.
Nome	Nome assegnato alla colonna, racchiuso tra virgolette doppie.
Expression	Qualsiasi espressione che restituisce un valore scalare come un valore riferimento di colonna, intero o stringa.

## Valore restituito

Tabella con lo stesso numero di righe della tabella specificata come primo argomento. La tabella restituita ha una colonna per ogni coppia di argomenti <Name>, <Expression> arguments, e ogni espressione viene valutata nel contesto di una riga dell'argomento <Table> specificato.

## Osservazioni:

SELECTCOLUMNS ha la stessa firma di ADDCOLUMNS e ha lo stesso comportamento, ma, invece di iniziare con il parametro <Table> specificato, SELECTCOLUMNS inizia con una tabella vuota prima di aggiungere le colonne.

Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

Per la tabella seguente denominata **Customer**:

Country	Provincia	Count	Totali
IND	5K	20	800
IND	MH	25	1000
IND	WB	10	900
USA	CA	5	500
USA	WA	10	900

DAX

```
SELECTCOLUMNS(Customer, "Country, State", [Country]&", "&[State])
```

Restituisce:

Paese, Stato
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# SUBSTITUTEWITHINDEX

Articolo • 20/10/2023

Restituisce una tabella che rappresenta un semi join di sinistra delle due tabelle fornite come argomenti. Il semi join viene eseguito usando colonne comuni e viene determinato dai nomi delle colonne comuni e dal tipo di dati comune. Le colonne di cui creare un join vengono sostituite con una singola colonna nella tabella restituita che è di tipo Integer e contiene un indice. L'indice è un riferimento alla tabella di join di destra in base a un ordinamento specificato.

Le colonne presenti nella seconda tabella (di destra) fornita, che non esistono nella prima tabella (di sinistra), non sono incluse nella tabella restituita e non vengono usate per il join.

L'indice inizia da 0 (base 0) e viene incrementato di uno per ogni riga aggiuntiva nella seconda tabella di join (di destra) fornita. L'indice è basato sull'ordinamento specificato per la seconda tabella di join (di destra).

## Sintassi

DAX

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>][, <orderBy_expression>, [<order>]]...])
```

## Parametri

Termine	Definizione
table	Tabella da filtrare eseguendo un semi join di sinistra con la tabella specificata come terzo argomento (indexColumnsTable). Si tratta della tabella sul lato sinistro del semi join di sinistra, quindi la tabella restituita include le stesse colonne di questa tabella, tranne per il fatto che tutte le colonne comuni delle due tabelle verranno sostituite con una singola colonna di indice nella tabella restituita.
indexColumnName	Stringa che specifica il nome della colonna di indice che sostituisce tutte le colonne comuni delle due tabelle fornite come argomenti per questa funzione.
indexColumnsTable	Seconda tabella per il semi join di sinistra. Si tratta della tabella sul lato destro del semi join di sinistra. Solo i valori presenti in questa tabella verranno restituiti dalla funzione. Inoltre, le colonne di questa tabella

Termino	Definizione
	(basate sui nomi delle colonne) verranno sostituite con una singola colonna di indice nella tabella restituita da questa funzione.
orderBy_expression	Qualsiasi espressione DAX in cui il valore del risultato viene usato per specificare l'ordinamento desiderato della tabella indexColumnsTable per la generazione di valori di indice corretti. L'ordinamento specificato per la tabella indexColumnsTable definisce l'indice di ogni riga della tabella e tale indice viene usato nella tabella restituita per rappresentare le combinazioni di valori in indexColumnsTable, così come appaiono nella tabella fornita come primo argomento per questa funzione.
order	<p>(facoltativo) Valore che specifica come ordinare i valori di orderBy_expression, in senso crescente o decrescente:</p> <p>Valore: <b>Dec</b>. Valore alternativo: <b>0(zero)/FALSE</b>. Ordinamenti in senso decrescente dei valori di orderBy_Expression. Si tratta del valore predefinito quando il parametro order viene omesso.</p> <p>Valore: <b>ASC</b>. Valore alternativo: <b>1/TRUE</b>. Ranghi di orderBy_Expression in ordine crescente.</p>

## Valore restituito

Tabella che include solo i valori presenti nella tabella indexColumnsTable e che ha una colonna di indice invece di tutte le colonne presenti (per nome) nella tabella indexColumnsTable.

## Osservazioni:

- Questa funzione non garantisce alcun ordinamento dei risultati.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SUMMARIZE

Articolo • 20/10/2023

Restituisce una tabella di riepilogo per i totali richiesti su un set di gruppi.

## Sintassi

DAX

```
SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

## Parametri

Termino	Definizione
table	Qualsiasi espressione DAX che restituisce una tabella di dati.
groupBy_ColumnName	(Facoltativo) Nome completo di una colonna esistente da usare per creare gruppi di riepilogo in base ai valori trovati. Questo parametro non può essere un'espressione.
name	Nome assegnato a una colonna di totale o di riepilogo, racchiuso tra virgolette doppie.
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).

## Valore restituito

Una tabella con le colonne selezionate per gli argomenti *groupBy\_columnName* e le colonne riepilogative definite dagli argomenti *name*.

## Osservazioni:

- Ogni colonna per la quale si definisce un nome deve avere un'espressione corrispondente, altrimenti viene restituito un errore. Il primo argomento, *name*, definisce il nome della colonna nei risultati. Il secondo argomento, *expression*, definisce il calcolo eseguito per ottenere il valore di ogni riga della colonna.
- *groupBy\_columnName* deve essere presente in *table* o in una tabella correlata a *table*.
- Ogni nome deve essere racchiuso tra virgolette doppie.
- La funzione raggruppa un set selezionato di righe in un set di righe di riepilogo in base ai valori di una o più colonne *groupBy\_columnName*. Viene restituita una riga per ogni gruppo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'esempio seguente restituisce un riepilogo delle vendite del rivenditore raggruppate in base all'anno di calendario e al nome della categoria di prodotto. Questa tabella dei risultati consente di eseguire l'analisi sulle vendite del rivenditore per anno e per categoria di prodotto.

DAX

```
SUMMARIZE(ResellerSales_USD  
, DateTime[CalendarYear]  
, ProductCategory[ProductCategoryName]  
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
```

```

        , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
    )

```

Nella tabella seguente viene mostrata un'anteprima dei dati come ricevuta da qualsiasi funzione che prevede la ricezione di una tabella:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	Componenti	2008052.706	39.9266
2005	Componenti	574256.9865	0
2006	Componenti	3428213.05	948.7674
2007	Componenti	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Accessori	153299.924	865.5945
2005	Accessori	18594.4782	4.293
2006	Accessori	86612.7463	1061.4872
2007	Accessori	275794.8403	4756.6546

## Con ROLLUP

L'aggiunta della sintassi **ROLLUP** modifica il comportamento della funzione **SUMMARIZE** aggiungendo le righe di rollup al risultato nelle colonne `groupBy_columnName`. **ROLLUP** può essere usato solo all'interno di un'espressione **SUMMARIZE**.

### Esempio

L'esempio seguente aggiunge righe di rollup alle colonne Group-By della chiamata alla funzione **SUMMARIZE**:

DAX

```

SUMMARIZE(ResellerSales_USD
    , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductName])
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)

```

Restituisce la tabella seguente:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2008	Componenti	2008052.706	39.9266
2005	Componenti	574256.9865	0
2006	Componenti	3428213.05	948.7674
2007	Componenti	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Accessori	153299.924	865.5945
2005	Accessori	18594.4782	4.293
2006	Accessori	86612.7463	1061.4872
2007	Accessori	275794.8403	4756.6546
2008		15496115.89	41224.3038
2005		7582953.67	4326.4144
2006		22871907.85	184419.1335
2007		30543780.84	297538.0745
		76494758.25	527507.9262

## Con ROLLUPGROUP

L'aggiunta di **ROLLUPGROUP** all'interno di una sintassi **ROLLUP** può tuttavia essere usata per evitare subtotali parziali nelle righe di rollup. **ROLLUPGROUP** può essere usato solo all'interno di un'espressione **ROLLUP**, **ROLLUPADDISSTOTAL** o **ROLLUPISSSTOTAL**.

### Esempio

L'esempio seguente mostra solo il totale complessivo di tutti gli anni e le categorie senza il sottototale di ogni anno con tutte le categorie:

DAX

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductName]))
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Restituisce la tabella seguente:

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	Componenti	2008052.706	39.9266
2005	Componenti	574256.9865	0

DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2006	Componenti	3428213.05	948.7674
2007	Componenti	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Accessori	153299.924	865.5945
2005	Accessori	18594.4782	4.293
2006	Accessori	86612.7463	1061.4872
2007	Accessori	275794.8403	4756.6546
		76494758.25	527507.9262

## Con ISSUBTOTAL

Con [ISSUBTOTAL](#) è possibile creare un'altra colonna nell'espressione SUMMARIZE, che restituisce True se la riga contiene i valori dei subtotali per la colonna specificata come argomento di [ISSUBTOTAL](#). In caso contrario, restituisce False. [ISSUBTOTAL](#) può essere usato solo all'interno di un'espressione SUMMARIZE.

## Esempio

L'esempio seguente genera una colonna [ISSUBTOTAL](#) per ognuna delle colonne [ROLLUP](#) nella chiamata alla funzione SUMMARIZE specificata:

DAX
<pre>SUMMARIZE(ResellerSales_USD     , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductName])     , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])     , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])     , "Is Sub Total for DateTime[CalendarYear]", ISSUBTOTAL(DateTime[CalendarYear])     , "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductName]) )</pre>

Restituisce la tabella seguente:

[Is Sub Total for DateTime[CalendarYear]]	[Is Sub Total for ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductName]	[Sales Amount (USD)]	[Discount Amount (USD)]
FALSE	FALSO				
FALSO	FALSE	2008	Bikes	12968255.42	36167.6592
FALSE	FALSE	2005	Bikes	6958251.043	4231.1621
FALSE	FALSE	2006	Bikes	18901351.08	178175.8399
FALSE	FALSE	2007	Bikes	24256817.5	276065.992
FALSE	FALSE	2008	Componenti	2008052.706	39.9266
FALSE	FALSE	2005	Componenti	574256.9865	0
FALSE	FALSE	2006	Componenti	3428213.05	948.7674
FALSE	FALSE	2007	Componenti	5195315.216	4226.0444

[Is Sub Total for DateTimeCalendarYear]	[Is Sub Total for ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Sales Amount (USD)]	[Discount Amount (USD)]
FALSE	FALSE	2008	Clothing	366507.844	4151.1235
FALSE	FALSE	2005	Clothing	31851.1628	90.9593
FALSE	FALSE	2006	Clothing	455730.9729	4233.039
FALSE	FALSE	2007	Clothing	815853.2868	12489.3835
FALSE	FALSE	2008	Accessori	153299.924	865.5945
FALSE	FALSE	2005	Accessori	18594.4782	4.293
FALSE	FALSE	2006	Accessori	86612.7463	1061.4872
FALSE	FALSE	2007	Accessori	275794.8403	4756.6546
FALSE	TRUE				
FALSO	TRUE	2008		15496115.89	41224.3038
FALSE	TRUE	2005		7582953.67	4326.4144
FALSE	TRUE	2006		22871907.85	184419.1335
FALSE	TRUE	2007		30543780.84	297538.0745
TRUE	TRUE			76494758.25	527507.9262

## Vedi anche

[SUMMARIZECOLUMNS](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SUMMARIZECOLUMNS

Articolo • 20/10/2023

Restituisce una tabella di riepilogo su un set di gruppi.

## Sintassi

DAX

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, < groupBy_columnName >]...,  
[<filterTable>]...[, <name>, <expression>]...)
```

## Parametri

Termine	Definizione
groupBy_columnName	Riferimento di colonna completo (Table[Column]) a una tabella di base per cui i valori distinti sono inclusi nella tabella restituita. Ogni colonna groupBy_columnName viene unita con un cross join (tabelle diverse) o viene sottoposta ad Auto Exist (stessa tabella) con le colonne successive specificate.
filterTable	Espressione di tabella che viene aggiunta al contesto di filtro di tutte le colonne specificate come argomenti groupBy_columnName. I valori presenti nella tabella di filtro vengono usati per filtrare prima dell'esecuzione di un cross join o di Auto Exist.
name	Stringa che rappresenta il nome della colonna da usare per l'espressione successiva specificata.
expression	qualsiasi espressione DAX che restituisce un singolo valore (non una tabella).

## Valore restituito

Tabella che include combinazioni di valori dalle colonne specificate, in base al raggruppamento specificato. Nella tabella restituita vengono incluse solo le righe per le quali almeno una delle espressioni specificate restituisce un valore non vuoto. Se tutte le espressioni restituiscono BLANK/NULL per una riga, tale riga non verrà inclusa nella tabella restituita.

## Osservazioni:

- Questa funzione non garantisce alcun ordinamento per i risultati.
- Una colonna non può essere specificata più di una volta nel parametro groupBy\_columnName. La formula seguente, ad esempio, non è valida.

```
SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
```

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Contesto di filtro

Si consideri la query seguente:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

In questa query senza una misura le colonne groupBy non contengono colonne dell'espressione FILTER (ad esempio, della tabella Customer). Il filtro non viene applicato alle colonne groupBy. Le tabelle Sales Territory e Customer possono essere indirettamente correlate tramite la tabella dei fatti Reseller sales. Poiché non sono correlate direttamente, l'espressione filtro non prevede alcuna operazione e le colonne groupBy non sono interessate.

Tuttavia con questa query:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer'[First Name] = "Alicia")
)
```

Le colonne groupBy contengono una colonna interessata dal filtro e tale filtro viene applicato ai risultati di groupBy.

## Con IGNORE

La sintassi **IGNORE** può essere usata per modificare il comportamento della funzione **SUMMARIZECOLUMNS** omettendo espressioni specifiche dalla valutazione **BLANK/NUL**L. Le righe per le quali tutte le espressioni che non usano **IGNORE** restituiscono **BLANK/NUL**L verranno escluse, indipendentemente dal fatto che le espressioni che usano **IGNORE** restituiscano o meno **BLANK/NUL**L. La funzione **IGNORE** può essere usata solo all'interno di un'espressione **SUMMARIZECOLUMNS**.

## Esempio

DAX

```
SUMMARIZECOLUMNS(  
    Sales[CustomerId], "Total Qty",  
    IGNORE( SUM( Sales[Qty] ) ),  
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )  
)
```

Esegue il rollup della colonna **Sales[CustomerId]**, creando un subtotale per tutti i clienti nel raggruppamento specificato. Senza **IGNORE**, il risultato è:

Customerid	TotalQty	BlankIfTotalQtyIsNot3
A	5	
B	3	3
C	3	3

Con **IGNORE**,

Customerid	TotalQty	BlankIfTotalQtyIsNot3
B	3	3
C	3	3

Tutte le espressioni ignorate,

DAX

```
SUMMARIZECOLUMNS(  
    Sales[CustomerId], "Blank",  
    IGNORE( Blank() ), "BlankIfTotalQtyIsNot5",  
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )  
)
```

Anche se entrambe le espressioni restituiscono BLANK per alcune righe, vengono incluse perché non sono presenti espressioni non ignorate che restituiscono BLANK.

Customerid	TotalQty	BlankIfTotalQtyIsNot3
A	5	
B		
A		

## Con NONVISUAL

La funzione **NONVISUAL** indica che un filtro di valore nella funzione SUMMARIZECOLUMN non influisce sui valori di misura, ma si applica solo alle colonne groupBy. La funzione **NONVISUAL** può essere usata solo all'interno di un'espressione SUMMARIZECOLUMN.

## Esempio

DAX
DEFINE MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount]) EVALUATE SUMMARIZECOLUMN ( DimDate[CalendarYear], NONVISUAL(TREATAS({2007, 2008}, DimDate[CalendarYear])), "Sales", [Sales], "Visual Total Sales", CALCULATE([Sales], ALLSELECTED(DimDate[CalendarYear])) ) ORDER BY [CalendarYear]

Restituisce il risultato in cui [Visual Total Sales] è il totale per tutti gli anni:

DimDate[CalendarYear]	[Sales]	[Visual Total Sales]
2007	9,791,060.30	29,358,677.22
2008	9,770,899.74	29,358,677.22

Al contrario, la stessa query *senza* la funzione NONVISUAL:

DAX
-----

```

DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    TREATAS({2007, 2008}, DimDate[CalendarYear]),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales],
    ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]

```

Restituisce il risultato in cui [Visual Total Sales] è il totale per i due anni selezionati:

DimDate[CalendarYear]	[Sales]	[Visual Total Sales]
2007	9,791,060.30	19,561,960.04
2008	9,770,899.74	19,561,960.04

## Con ROLLUPADDISSUBTOTAL

L'aggiunta della sintassi **ROLLUPADDISSUBTOTAL** modifica il comportamento della funzione SUMMARIZECOLUMNS aggiungendo righe di rollup/subtotali al risultato in base alle colonne groupBy\_columnName. La funzione **ROLLUPADDISSUBTOTAL** può essere usata solo all'interno di un'espressione SUMMARIZECOLUMNS.

## Esempio con singolo totale

DAX

```

DEFINE
VAR vCategoryFilter =
    TREATAS({"Accessories", "Clothing"}, Product[Category])
VAR vSubcategoryFilter =
    TREATAS({"Bike Racks", "Mountain Bikes"}, Product[Subcategory])
EVALUATE
SUMMARIZECOLUMNS
(
    ROLLUPADDISSUBTOTAL
    (
        Product[Category], "IsCategorySubtotal", vCategoryFilter,
        Product[Subcategory], "IsSubcategorySubtotal", vSubcategoryFilter
    ),
    "Total Qty", SUM(Sales[Qty])
)

```

```

ORDER BY
[IsCategorySubtotal] DESC, [Category],
[IsSubcategorySubtotal] DESC, [Subcategory]

```

Restituisce la tabella seguente:

Categoria	Sottocategoria	IsCategorySubtotal	IsSubcategorySubtotal	Quantità totale
		Vero	Vero	60398
Accessori		False	Vero	36092
Accessori	Bike Racks	False	False	328
Bikes	Mountain Bikes	False	False	4970
Clothing		False	Vero	9101

## Esempio con più subtotali

DAX

```

SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSTOTAL ( Sales[CustomerId],
"IsCustomerSubtotal" ),
    ROLLUPADDISSTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM(
Sales[Qty] )
)

```

Le vendite vengono raggruppate per stato, per cliente, per data, con subtotali per 1. Vendite per stato, per data 2. Vendite per stato, per cliente 3. Il rollup viene eseguito sia per cliente che per data, ottenendo le vendite per stato.

Restituisce la tabella seguente:

IDCliente	IsCustomerSubtotal	Provincia	Quantità totale	Data	IsDateSubtotal
A	FALSE	WA	5	10/7/2014	
B	FALSE	WA	1	10/7/2014	
B	FALSE	WA	2	7/11/2014	
A	FALSE	OPPURE	2	10/7/2014	
A	FALSE	OPPURE	1	7/11/2014	
	TRUE	WA	6	10/7/2014	

IDCliente	IsCustomerSubtotal	Provincia	Quantità totale	Data	IsDateSubtotal
	TRUE	WA	2	7/11/2014	
	TRUE	OPPURE	2	10/7/2014	
	TRUE	OPPURE	1	7/11/2014	
A	FALSE	WA	5		TRUE
B	FALSE	WA	3		TRUE
A	FALSE	OPPURE	3		TRUE
	TRUE	WA	8		TRUE
	TRUE	OPPURE	3		TRUE

## Con ROLLUPGROUP

Analogamente alla funzione [SUMMARIZE](#), è possibile usare [ROLLUPGROUP](#) con [ROLLUPADDISSTOTAL](#) per specificare quali gruppi di riepilogo/granularità (subtotali) includere (riducendo il numero di righe di subtotali restituite). La funzione [ROLLUPGROUP](#) può essere usata solo all'interno di un'espressione [SUMMARIZECOLUMNS](#) o [SUMMARIZE](#).

### Esempio con più subtotali

DAX

```
SUMMARIZECOLUMNS(
    ROLLUPADDISSTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSTOTAL(ROLLUPGROUP(Regions[City], Regions[State]),
    "IsCityStateSubtotal"), "Total Qty", SUM( Sales[Qty] )
)
```

Ancora raggruppati per città e stato, ma sommati insieme quando si indica un subtotali, viene restituita la tabella seguente.

Provincia	Customerid	IsCustomerSubtotal	Quantità totale	City	IsCityStateSubtotal
WA	A	FALSE	2	Bellevue	FALSE
WA	B	FALSE	2	Bellevue	FALSE
WA	A	FALSE	3	Milano	FALSE

Provincia	Customerid	IsCustomerSubtotal	Quantità totale	City	IsCityStateSubtotal
WA	B	FALSE	1	Milano	FALSE
OPPURE	A	FALSE	3	Portland	FALSE
WA		TRUE	4	Bellevue	FALSE
WA		TRUE	4	Milano	FALSE
OPPURE		TRUE	3	Portland	FALSE
	A	FALSE	5		FALSE
	B	FALSE	3		TRUE
	A	FALSE	3		TRUE
		TRUE	11		TRUE

## Vedi anche

[SUMMARIZE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Costruttore di tabella

Articolo • 20/10/2023

Restituisce una tabella di una o più colonne.

## Sintassi

DAX

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ),  
... }
```

## Parametri

Termine	Definizione
scalarExprN	Espressione DAX che restituisce un valore scalare.

## Valore restituito

Una tabella di una o più colonne. Quando è presente una sola colonna, il nome della colonna è Value. Quando sono presenti N colonne dove  $N > 1$ , i nomi delle colonne da sinistra a destra sono Value1, Value2, ..., ValueN.

## Osservazioni:

- La prima sintassi restituisce una tabella di una singola colonna. La seconda sintassi restituisce una tabella di una o più colonne.
- Il numero di espressioni scalari deve essere lo stesso per tutte le righe.
- Quando i tipi di dati dei valori per una colonna sono diversi in righe diverse, tutti i valori vengono convertiti in un tipo di dati comune.

## Esempio 1

Le query DAX seguenti:

```
DAX
```

```
EVALUATE { 1, 2, 3 }
```

e

```
DAX
```

```
EVALUATE { (1), (2), (3) }
```

Restituiscono la tabella seguente di una singola colonna:

[Value]
1
2
3

## Esempio 2

La query DAX seguente:

```
DAX
```

```
EVALUATE
{
    (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),
    (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),
    (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")
}
```

Restituisce:

[Value1]	[Value2]	[Value3]	[Value4]
1.5	1/1/2017	199.99	A
2.5	2/1/2017	249.99	B
3.5	1/3/2017	299.99	A

## Esempio 3

La query DAX seguente:

DAX

```
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

Restituisce la tabella seguente di una singola colonna del tipo di dati stringa:

[Value]
1
1/1/2017
TRUE
A

## Commenti e suggerimenti

Questa pagina è stata utile?



# TOPN

Articolo • 20/10/2023

Restituisce le prime N righe della tabella specificata.

## Sintassi

DAX

```
TOPN(<N_Value>, <Table>, <OrderBy_Expression>, [<Order>[,<OrderBy_Expression>, [<Order>]]...])
```

## Parametri

Parametro	Definizione
N_Value	Numero di righe da restituire. Qualsiasi espressione DAX che restituisce un valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto). Per altre informazioni su quando il numero di righe restituite può essere maggiore di <i>n_value</i> , vedere la sezione Osservazioni.
Tabella	Qualsiasi espressione DAX che restituisce una tabella di dati da cui estrarre le prime 'n' righe. Per altre informazioni su quando viene restituita una tabella vuota, vedere la sezione Osservazioni.
OrderBy_Expression	(Facoltativo) Qualsiasi espressione DAX in cui il valore risultante viene usato per ordinare la tabella e viene valutato per ogni riga di <i>table</i> .
Ordine	(Facoltativo) Valore che specifica come ordinare i valori di <i>OrderBy_Expression</i> : <ul style="list-style-type: none"><li>- 0 (zero) o FALSE. Ordinamenti in senso decrescente dei valori di <i>Order</i>. Valore predefinito quando il parametro <i>Order</i> viene omesso.</li><li>- 1 o TRUE. Ranghi in ordine crescente di <i>OrderBy</i>.</li></ul>

## Valore restituito

Tabella con le prime N righe di *Table* o tabella vuota se *N\_Value* è minore o uguale a 0 (zero). Le righe non seguono un ordine particolare.

## Osservazioni:

- Se è disponibile un valore equivalente, nei valori *Order\_By*, in corrispondenza dell'ennesima riga della tabella, vengono restituite tutte le righe equivalenti. Quando dunque sono presenti valori equivalenti nell'ennesima riga, la funzione potrebbe restituire più di n righe.
- Se N\_value è uguale o minore di 0 (zero), TOPN restituisce una tabella vuota.
- TOPN non garantisce alcun ordinamento per i risultati.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di misura seguente restituisce i primi 10 prodotti venduti in base all'importo delle vendite.

```
DAX

= SUMX(
    TOPN(
        10,
        SUMMARIZE(
            InternetSales,
            InternetSales[ProductKey],
            "TotalSales", SUM(InternetSales[SalesAmount])
        ),
        [TotalSales], DESC
    ),
    [TotalSales]
)
```

---

## Commenti e suggerimenti

Questa pagina è stata utile?

Sì
 No

# TREATAS

Articolo • 20/10/2023

Applica il risultato di un'espressione di tabella come filtri alle colonne di una tabella non correlata.

## Sintassi

DAX

```
TREATAS(table_expression, <column>[, <column>[, <column>[,...]]]} )
```

## Parametri

Termine	Definizione
table_expression	Espressione tramite cui viene restituita una tabella.
colonna	Una o più colonne esistenti. Non può essere un'espressione.

## Valore restituito

Tabella che contiene tutte le righe di una o più colonne che sono presenti anche in table\_expression.

## Osservazioni:

- Il numero di colonne specificato deve corrispondere al numero di colonne nell'espressione di tabella ed essere nello stesso ordine.
- Se un valore restituito nell'espressione di tabella non è presente nella colonna, viene ignorato. Ad esempio, TREATAS ({"Red", "Green", "Yellow"}, DimProduct[Color]) imposta un filtro sulla colonna DimProduct[Color] con tre valori "Red", "Green" e "Yellow". Se "Yellow" non esiste in DimProduct[Color], i valori di filtro effettivi sono "Red" e "Green".
- Ideale per l'utilizzo quando non esiste una relazione tra le tabelle. Se sono presenti più relazioni tra le tabelle, valutare la possibilità di usare [USERELATIONSHIP](#) in alternativa.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente il modello contiene due tabelle di prodotti non correlate. Se un utente applica un filtro a DimProduct1[ProductCategory] selezionando Bikes, Seats, Tires, lo stesso filtro Bikes, Seats, Tires viene applicato a DimProduct2[ProductCategory].

DAX

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

## Vedi anche

[INTERSECT](#)

[FILTER](#)

[USERELATIONSHIP](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# UNION

Articolo • 20/10/2023

Crea una tabella di unione (join) da una coppia di tabelle.

## Sintassi

DAX

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

## Parametri

Termine	Definizione
table_expression	qualsiasi espressione DAX che restituisce una tabella.

## Valore restituito

Tabella che contiene tutte le righe di ognuna delle due espressioni di tabella.

## Osservazioni:

- Le due tabelle devono avere lo stesso numero di colonne.
- Le colonne vengono combinate in base alla posizione nelle rispettive tabelle.
- I nomi delle colonne nella tabella restituita corrispondono ai nomi delle colonne in table\_expression1.
- Le righe duplicate vengono mantenute.
- La tabella restituita presenta una derivazione laddove possibile. Se ad esempio la prima colonna di ogni table\_expression ha una derivazione della stessa colonna di base C1 nel modello, la prima colonna nel risultato di UNION avrà una derivazione di C1. Se tuttavia le colonne combinate hanno una derivazione a colonne di base diverse o se è presente una colonna di estensione, la colonna risultante in UNION non avrà alcuna derivazione.

- Quando i tipi di dati sono diversi, il tipo di dati risultante viene determinato in base alle regole per la coercizione del tipo di dati.
- La tabella restituita non conterrà colonne da tabelle correlate.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

L'espressione seguente crea un'unione combinando la tabella USAInventory e la tabella INDInventory in una singola tabella:

DAX
<code>UNION(UsaInventory, IndInventory)</code>

### USAInventory

Paese/area geografica	Provincia	Count	Totali
USA	CA	5	500
USA	WA	10	900

### INDInventory

Paese/area geografica	Provincia	Count	Totali
IND	5K	20	800
IND	MH	25	1000
IND	WB	10	900

### Tabella restituita

Paese/area geografica	Provincia	Count	Totali
USA	CA	5	500
USA	WA	10	900
IND	5K	20	800
IND	MH	25	1000

Paese/area geografica	Provincia	Count	Totali
IND	WB	10	900

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# VALUES

Articolo • 20/10/2023

Quando il parametro di input è un nome di colonna, restituisce una tabella a singola colonna contenente i valori distinti della colonna specificata. I valori duplicati vengono rimossi e vengono restituiti solo valori univoci. È possibile aggiungere un valore BLANK. Quando il parametro di input è un nome di tabella, restituisce le righe dalla tabella specificata. Le righe duplicate vengono mantenute. È possibile aggiungere una riga BLANK.

## ① Nota

Questa funzione non può essere usata per restituire valori in una cella o in una colonna di un foglio di calcolo. Viene invece usata come funzione intermedia, annidata in una formula, per ottenere un elenco di valori distinti che possono essere conteggiati o usati per filtrare o sommare altri valori.

## Sintassi

DAX

**VALUES(<TableNameOrColumnName>)**

## Parametri

Termine	Definizione
TableName o ColumnName	Colonna da cui devono essere restituiti valori univoci o tabella da cui devono essere restituite righe.

## Valore restituito

Quando il parametro di input è un nome di colonna, una tabella a colonna singola. Quando il parametro di input è un nome di tabella, viene restituita una tabella con le stesse colonne.

## Osservazioni:

- Quando si usa la funzione VALUES in un contesto filtrato, i valori univoci restituiti da VALUES sono interessati dal filtro. Se ad esempio si filtra per area e viene restituito un elenco di valori di città, l'elenco includerà solo le città nelle aree consentite dal filtro. Per restituire tutte le città, indipendentemente dai filtri esistenti, è necessario usare la funzione ALL per rimuovere i filtri dalla tabella. Il secondo esempio illustra l'uso di ALL con VALUES.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.
- Per le procedure consigliate quando si usa VALUES, vedere [Usare SELECTEDVALUE invece di VALUES](#).

## Funzioni correlate

Nella maggior parte degli scenari, quando l'argomento è un nome di colonna, i risultati della funzione VALUES sono identici a quelli della funzione **DISTINCT**. Entrambe le funzioni rimuovono i duplicati e restituiscono un elenco dei possibili valori nella colonna specificata. La funzione VALUES tuttavia può restituire anche un valore blank. Questo valore blank è utile quando si cercano valori distinti da una tabella correlata, ma un valore usato nella relazione manca in una tabella. Nella terminologia dei database questa condizione è detta violazione dell'integrità referenziale. Tali mancate corrispondenze nei dati possono verificarsi quando una tabella viene aggiornata e quella correlata no.

Quando l'argomento è un nome di tabella, il risultato della funzione VALUES restituisce tutte le righe nella tabella specificata più una riga vuota, se si verifica una violazione dell'integrità referenziale. La funzione DISTINCT rimuove le righe duplicate e restituisce righe univoche nella tabella specificata.

### ⓘ Nota

La funzione DISTINCT consente di usare come argomento un nome di colonna o un'espressione di tabella valida, ma la funzione VALUES accetta solo un nome di colonna o un nome di tabella come argomento.

La tabella seguente riepiloga la mancata corrispondenza tra i dati, che può verificarsi in due tabelle correlate quando l'integrità referenziale non viene preservata.

Tabella MyOrders	Tabella MySales
June 1	June 1 sales

Tabella MyOrders	Tabella MySales
(non sono state immesse date di ordini)	June 3 sales

Se si usa la funzione DISTINCT per restituire un elenco di date, saranno restituite solo due date. Se tuttavia si usa la funzione VALUES, la funzione restituisce le due date più un membro vuoto aggiuntivo. Inoltre, tutte le righe della tabella MySales che non hanno una data corrispondente nella tabella MyOrders verranno associate a questo membro sconosciuto.

## Esempio

La formula seguente conteggia il numero di fatture univoche (ordini di vendita) e genera i risultati seguenti se usata in un report che include i nomi delle categorie di prodotti:

DAX
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))

Resi

Etichette di riga	Conteggio fatture
Accessori	18,208
Bikes	15,205
Clothing	7,461
Grand Total	27,659

## Vedi anche

[Funzione FILTER](#)

[Funzione COUNTROWS](#)

[Funzioni di filtro](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

👍 Sì
👎 No

# Funzioni di testo

Articolo • 20/10/2023

Data Analysis Expressions (DAX) include un set di funzioni per il testo basate sulla libreria di funzioni di stringa in Excel, ma modificate per l'utilizzo di tabelle e colonne nei modelli tabulari. Questa sezione descrive le funzioni per il testo disponibili nel linguaggio DAX.

## In questa categoria

Funzione	Descrizione
<a href="#">COMBINEVALUES</a>	unisce due o più stringhe di testo in una sola.
<a href="#">CONCATENATE</a>	Unisce due stringhe di testo in una sola.
<a href="#">CONCATENATEX</a>	Concatena il risultato di un'espressione valutata per ogni riga di una tabella.
<a href="#">EXACT</a>	Confronta due stringhe di testo e restituisce TRUE se sono identiche. In caso contrario, FALSE.
<a href="#">FIND</a>	Restituisce la posizione iniziale di una stringa di testo all'interno di un'altra stringa.
<a href="#">FIXED</a>	Arrotonda un numero al numero di decimali specificato e restituisce il risultato come testo.
<a href="#">FORMAT</a>	Converte un valore in testo in base al formato specificato.
<a href="#">LEFT</a>	Restituisce il numero specificato di caratteri dall'inizio di una stringa di testo.
<a href="#">LEN</a>	Restituisce il numero di caratteri in una stringa di testo.
<a href="#">LOWER</a>	Converte tutte le lettere di una stringa di testo in caratteri minuscoli.
<a href="#">MID</a>	Restituisce una stringa di caratteri dal centro di una stringa di testo, date una posizione di inizio e una lunghezza.
<a href="#">REPLACE</a>	REPLACE sostituisce parte di una stringa di testo, in base al numero di caratteri specificati, con una stringa di testo diversa.
<a href="#">REPT</a>	Ripete il testo un determinato numero di volte.
<a href="#">RIGHT</a>	RIGHT restituisce uno o più caratteri finali in una stringa di testo, in base al numero di caratteri specificato.
<a href="#">SEARCH</a>	Restituisce il numero del carattere in corrispondenza del quale viene individuato per la prima volta un carattere o una stringa di testo specifica,

Funzione	Descrizione
	leggendo da sinistra a destra.
SUBSTITUTE	Sostituisce il nuovo testo a quello esistente in una stringa di testo.
TRIM	Rimuove tutti gli spazi dal testo esclusi gli spazi singoli che dividono le parole.
UNICHAR	Restituisce il carattere Unicode a cui fa riferimento il valore numerico.
UNICODE	Restituisce il codice numerico corrispondente al primo carattere della stringa di testo.
UPPER	Converte una stringa di testo in caratteri maiuscoli.
VALUE	Converte una stringa di testo che rappresenta un numero in un numero.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# COMBINEVALUES

Articolo • 20/10/2023

unisce due o più stringhe di testo in una sola. Lo scopo principale di questa funzione è supportare le relazioni a più colonne nei modelli DirectQuery. Per altri dettagli, vedere la sezione **Osservazioni**.

## Sintassi

DAX

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[ , <expression>]...)
```

## Parametri

Termine	Definizione
delimiter	Separatore da usare durante la concatenazione. Deve essere un valore costante.
expression	Espressione DAX il cui valore verrà combinato in una singola stringa di testo.

## Valore restituito

Stringa concatenata.

## Osservazioni:

- La funzione COMBINEVALUES presuppone, ma non convalida, il fatto che quando i valori di input sono diversi, anche le stringhe di output sono diverse. In base a questo presupposto, quando si usa COMBINEVALUES viene per creare colonne calcolate, allo scopo di creare una relazione che unisca più colonne di due tabelle DirectQuery, durante l'esecuzione della query viene generata una condizione di join ottimizzata. Ad esempio, se gli utenti vogliono creare una relazione tra Table1(Column1, Column2) e Table2(Column1, Column2), possono creare due colonne calcolate, una per ogni tabella, come segue:

DAX

```
Table1[CalcColumn] = COMBINEVALUES(", ", Table1[Column1],
```

```
Table1[Column2])
```

e

```
DAX
```

```
Table2[CalcColumn] = COMBINEVALUES(", ", Table2[Column1],  
Table2[Column2])
```

e quindi creare una relazione tra `Table1[CalcColumn]` e `Table2[CalcColumn]`. A differenza di altre funzioni e operatori DAX, che vengono convertiti letteralmente nelle funzioni e negli operatori SQL corrispondenti, la relazione precedente genera un predicato di join SQL come:

```
DAX
```

```
(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND  
Table2.Column1 IS NULL)
```

e

```
DAX
```

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND  
Table2.Column2 IS NULL)
```

- Potenzialmente, il predicato di join può offrire prestazioni di query molto migliori rispetto a quelle che coinvolgono funzioni e operatori SQL complessi.
- La funzione COMBINEVALUES si basa sul presupposto che l'utente scelga il delimitatore appropriato per garantire che combinazioni univoche di valori di input producano stringhe di output distinte, ma non convalida che il presupposto sia veritiero. Ad esempio, se l'utente sceglie `"| "` come delimitatore, ma una riga in Tabella1 ha `Table1[Column1] = "| "` e `Table2[Column2] = " "`, mentre una riga in Tabella2 ha `Table2[Column1] = " "` e `Table2[Column2] = "| "`, i due output concatenati saranno entrambi `"|| "`, il che sembra indicare che le due righe rappresentano una corrispondenza nell'operazione di join. Le due righe non vengono unite in join se entrambe le tabelle appartengono alla stessa origine DirectQuery, sebbene vengano unite in join se entrambe le tabelle vengono importate.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
DISTINCT (  
    SELECTCOLUMNS ( Date, "Month", COMBINEVALUES ( ", ", [MonthName],  
[CalendarYear] ) )  
)
```

Restituisce la tabella seguente con una singola colonna:

[Month]
Gennaio 2020
Febbraio 2020
Marzo 2020
Aprile 2020
Maggio 2020
Giugno 2020
Luglio 2020
Agosto 2020
Settembre 2020
Ottobre 2020
Novembre 2020
Dicembre 2020
Gennaio 2021
Gennaio 2021
Febbraio 2021
Marzo 2021
Aprile 2021
Maggio 2021
Giugno 2021

**[Month]**

Luglio 2021

Agosto 2021

Settembre 2021

Ottobre 2021

Novembre 2021

Dicembre 2021

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# CONCATENATE

Articolo • 20/10/2023

Unisce due stringhe di testo in una sola.

## Sintassi

DAX

```
CONCATENATE(<text1>, <text2>)
```

## Parametri

Termine	Definizione
text1	Prima stringa di testo da unire in una singola stringa di testo. La stringa può includere testo o numeri. È anche possibile usare riferimenti a colonne.
text2	Seconda stringa di testo da unire in una singola stringa di testo. La stringa può includere testo o numeri. È anche possibile usare riferimenti a colonne.

## Valore restituito

Stringa concatenata.

## Osservazioni:

- La funzione CONCATENATE unisce due o più stringhe di testo in una sola. Gli elementi uniti possono includere testo, numeri o valori booleani rappresentati come testo o una combinazione di tali elementi. È anche possibile usare un riferimento a colonna se questa contiene valori appropriati.
- La funzione CONCATENATE in DAX accetta solo due argomenti, mentre la funzione CONCATENA di Excel accetta fino a 255 argomenti. Se è necessario concatenare più colonne, è possibile creare una serie di calcoli, o meglio, usare l'operatore di concatenazione (&) per unirle tutte in un'espressione più semplice.
- Se si vogliono usare direttamente stringhe di testo anziché un riferimento a colonna, è necessario racchiudere ogni stringa tra virgolette doppie.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio: Concatenazione di valori letterali

La formula di esempio crea un nuovo valore stringa combinando due valori stringa forniti come argomenti.

DAX

```
= CONCATENATE("Hello ", "World")
```

## Esempio: Concatenazione di stringhe nelle colonne

La formula di esempio restituisce il nome completo del cliente come elencato in una rubrica telefonica. Notare come una funzione annidata viene usata come secondo argomento. Questo è un modo per concatenare più stringhe se si hanno più di due valori che si vogliono usare come argomenti.

DAX

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

## Esempio: Concatenazione condizionale di stringhe nelle colonne

La formula di esempio crea una nuova colonna calcolata nella tabella Customer con il nome completo del cliente dato dalla combinazione di nome, iniziale del secondo nome e cognome. Se non è presente un secondo nome, il cognome seguirà immediatamente il nome. Se è presente un secondo nome, verrà usata solo l'iniziale di tale nome seguita da un punto.

DAX

```
= CONCATENATE( [FirstName]& " ", CONCATENATE( IF( LEN([MiddleName])>1, LEFT([MiddleName],1)& " ", ""), [LastName]))
```

Questa formula usa funzioni CONCATENATE e IF annidate insieme all'operatore e commerciale (&) per eseguire una concatenazione condizionale di tre valori stringa e per aggiungere spazi come separatori.

## Esempio: Concatenazione di colonne con tipi di dati diversi

L'esempio seguente mostra come concatenare valori in colonne che hanno tipi di dati diversi. Se il valore che si concatena è numerico, il valore sarà convertito in modo implicito in testo. Se i valori sono entrambi numerici, verrà eseguito il cast al testo di entrambi e saranno concatenati come se fossero stringhe.

Descrizione del prodotto.	Abbreviazione del prodotto (colonna 1 di chiave composta)	Numero di prodotto (colonna 2 di chiave composta)	Nuova colonna chiave generata
Mountain bike	MTN	40	MTN40
Mountain bike	MTN	42	MTN42

DAX

```
= CONCATENATE('Products'[Product abbreviation], 'Products'[Product number])
```

La funzione CONCATENATE in DAX accetta solo due argomenti, mentre la funzione CONCATENA di Excel accetta fino a 255 argomenti. Se è necessario aggiungere più argomenti, si può usare l'operatore e commerciale (&). Ad esempio, la formula seguente produce i risultati MTN-40 e MTN-42.

DAX

```
= [Product abbreviation] & "-" & [Product number]
```

## Vedi anche

[CONCATENATEX](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# CONCATENATEX

Articolo • 20/10/2023

Concatena il risultato di un'espressione valutata per ogni riga di una tabella.

## Sintassi

DAX

```
CONCATENATEX(<table>, <expression>[, <delimiter> [, <orderBy_expression> [, <order>]]...])
```

## Parametri

Termine	Definizione
table	Tabella contenente le righe per le quali verrà valutata l'espressione.
expression	Espressione da valutare per ogni riga della <i>tabella</i> .
delimiter	(Facoltativo) Separatore da usare durante la concatenazione.
orderBy_expression	(Facoltativo) Qualsiasi espressione DAX in cui il valore risultante viene usato per ordinare i valori concatenati nella stringa di output. Viene valutata per ogni riga della <i>tabella</i> .
order	(facoltativo) Valore che specifica come ordinare i valori di <i>orderBy_expression</i> , in senso crescente o decrescente.

Il parametro facoltativo **order** accetta i valori seguenti:

Valore	Valori alternativi	Descrizione
0 (zero)	FALSE, DESC	Ordinamenti in senso decrescente dei valori di <i>orderBy_expression</i> . Si tratta del valore predefinito quando il parametro <i>order</i> viene omesso.
1	TRUE, ASC	Ordinamenti in senso decrescente dei valori di <i>orderBy_expression</i> .

## Valore restituito

Stringa concatenata.

## Osservazioni:

- Questa funzione accetta come primo argomento una tabella o un'espressione che restituisce una tabella. Il secondo argomento è una colonna che contiene i valori da concatenare oppure un'espressione che restituisce un valore.
- I valori concatenati non vengono necessariamente ordinati in un ordine specifico, a meno che non venga specificato *orderBy\_expression*.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Tabella Employees

FirstName	LastName
Mattia	Brewer
Michael	Blythe

La formula seguente:

DAX

```
= CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Restituisce:

"Alan Brewer, Michael Blythe"

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# EXACT

Articolo • 20/10/2023

Confronta due stringhe di testo e restituisce TRUE se sono identiche. In caso contrario, FALSE. EXACT fa distinzione tra maiuscole e minuscole, ma ignora le differenze di formattazione. EXACT supporta la distinzione tra maiuscole e minuscole

## Sintassi

DAX

```
EXACT(<text1>,<text2>)
```

## Parametri

Termine	Definizione
text1	Prima stringa di testo o colonna contenente testo.
text2	Seconda stringa di testo o colonna contenente testo.

## Valore restituito

Vero o falso. (Boolean)

## Esempio

La formula seguente utilizzata in una colonna calcolata nella tabella Product controlla il valore di Product per la riga corrente rispetto al valore di Model per la riga corrente, e restituisce True se sono uguali, False se sono diversi.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
= EXACT([Product], [Model])
```

## Vedi anche

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FIND

Articolo • 20/10/2023

Restituisce la posizione iniziale di una stringa di testo all'interno di un'altra stringa. FIND fa distinzione tra maiuscole e minuscole.

## Sintassi

DAX

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

## Parametri

Termine	Definizione
find_text	Testo da trovare. Usare le virgolette doppie (testo vuoto) per la corrispondenza con il primo carattere in <code>within_text</code> .
within_text	Stringa di testo contenente il testo da trovare.
start_num	(facoltativo) Carattere dal quale iniziare la ricerca; se omesso, <code>start_num</code> = 1. Il primo carattere in <code>within_text</code> è il numero 1.
NotFoundValue	(facoltativo, ma fortemente consigliato) Valore che deve essere restituito quando l'operazione non trova una substring corrispondente, in genere 0, -1 o <code>BLANK()</code> . Se non specificato, viene restituito un errore.

## Valore restituito

Numero che specifica il punto iniziale della stringa di testo da trovare.

## Osservazioni:

- Mentre in Microsoft Excel sono disponibili più versioni della funzione FIND per la gestione delle lingue SBCS (Single Byte Character Set) e DBCS (Double Byte Character Set), DAX usa Unicode e conta ogni tipo di caratteri allo stesso modo, quindi non è necessario usare una versione diversa in base al tipo di carattere.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

- FIND non supporta i caratteri jolly. Per usare i caratteri jolly, usare [SEARCH](#).

## Esempio

La query DAX seguente trova la posizione della prima lettera di "Bike" nella stringa contenente il nome del rivenditore. Se non viene trovato, viene restituito Blank.

Tenere presente che FIND fa distinzione tra maiuscole e minuscole. In questo esempio, se "bike" è stato usato nell'argomento <find\_text>, non verrà restituito alcun risultato. Usare [SEARCH](#) per fare distinzione tra maiuscole e minuscole.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

```
DAX

EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of Bike", FIND ( "Bike", 'Reseller'[Reseller], 1, BLANK ())
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added Reseller", "Warehouse" }
)
```

Restituisce:

Reseller	Business Type	Posizione della bicicletta
Volume Bike Sellers	Magazzino	8
Mass Market Bikes	Value Added Reseller	13
Twin Cycles	Value Added Reseller	
Rich Department Store	Magazzino	
Rental Gallery	Specialty Bike Shop	
Budget Toy Store	Magazzino	
Global Sports Outlet	Magazzino	
Online Bike Catalog	Magazzino	8

Reseller	Business Type	Posizione della bicicletta
Helmets and Cycles	Value Added Reseller	
Jumbo Bikes	Specialty Bike Shop	7

## Vedi anche

[SEARCH](#)

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FIXED

Articolo • 20/10/2023

Arrotonda un numero al numero di decimali specificato e restituisce il risultato come testo. È possibile specificare che il risultato venga restituito con o senza virgole.

## Sintassi

DAX

```
FIXED(<number>, <decimals>, <no_commas>)
```

## Parametri

Termine	Definizione
number	Numero da arrotondare e convertire in testo o una colonna contenente un numero.
decimals	(facoltativo) Numero di cifre a destra del separatore decimale; se omesso, viene specificato 2.
no_commas	(facoltativo) Valore logico: se 1, non vengono visualizzate virgole nel testo restituito; se 0 o omesso, vengono visualizzate le virgole nel testo restituito.

## Valore restituito

Numero rappresentato come testo.

## Osservazioni:

- Se il valore usato per il parametro **decimals** è negativo, **number** viene arrotondato a sinistra del separatore decimale.
- Se si omette **decimals**, si presuppone che sia pari a 2.
- Se **no\_commas** è 0 o viene omesso, il testo restituito includerà le virgole come di consueto.
- La differenza principale tra la formattazione di una cella contenente un numero mediante un comando e la formattazione di un numero direttamente con la

funzione FIXED sta nel fatto che FIXED converte il risultato in testo. Un numero formattato con un comando del menu di formattazione rimane un numero.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula seguente usata in una colonna calcolata ottiene il valore numerico per la riga corrente in Product[List Price] e la restituisce come testo con 2 cifre decimali e senza virgole.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
= FIXED([List Price],2,1)
```

## Vedi anche

[CEILING](#)

[FLOOR](#)

[ISO.CEILING](#)

[MROUND](#)

[ROUND](#)

[ROUNDDOWN](#)

[ROUNDUP](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FORMAT

Articolo • 20/10/2023

Converte un valore in testo in base al formato specificato.

## Sintassi

DAX

```
FORMAT(<value>, <format_string>[, <locale_name>])
```

## Parametri

Termine	Definizione
value	Valore o espressione che restituisce un singolo valore.
format_string	Stringa con il modello di formattazione.
locale_name	(Facoltativo) Nome delle impostazioni locali da usare dalla funzione. I valori possibili sono stringhe accettate dalla funzione dell'API Windows <a href="#">LocaleNameToLCID()</a> .

## Valore restituito

Stringa contenente **value** formattata come definito da **format\_string**.

### ⓘ Nota

Se **value** è BLANK la funzione restituisce una stringa vuota.

Se **format\_string** è BLANK, il valore viene formattato con un formato "Numero generico" o "Data generale" (in base al tipo di dati di **value**).

## Osservazioni:

- Le stringhe di formato predefinite usano la proprietà della lingua del modello durante la formattazione del risultato. Per impostazione predefinita, la proprietà della lingua del modello viene impostata in base alle impostazioni locali dell'utente

del computer. Per i nuovi modelli di Power BI Desktop, la proprietà della lingua del modello può essere modificata in Opzioni > Impostazioni internazionali> Lingua del modello. Per Analysis Services, la lingua del modello è impostata in base alla proprietà Lingua definita inizialmente dall'istanza.

- Le stringhe di formato supportate come argomento della funzione DAX FORMAT sono basate sulle stringhe di formato usate da Visual Basic (Automazione OLE), non sulle stringhe di formato usate da .NET Framework. Quindi si potrebbero verificare risultati imprevisti o errori se l'argomento non corrisponde alle stringhe di formato definite. Ad esempio, "p" come abbreviazione di "percentuale" non è supportato. Le stringhe fornite come argomento alla funzione FORMAT che non sono incluse nell'elenco di stringhe di formato predefinite vengono gestite come parte di una stringa di formato personalizzata o come un valore letterale stringa.
- Per altre informazioni su come specificare le impostazioni locali con FORMAT, vedere questo [video](#).
- L'uso di FORMAT modifica il risultato di una misura in un tipo di dati di testo. Se il risultato della misura è originariamente di tipo numerico, con FORMAT la misura non può essere usata negli oggetti visivi in cui la sezione valori richiede un tipo di dati numerico, ad esempio con i grafici. In Power BI è possibile usare in alternativa **stringhe di formato dinamico per le misure** specificare una stringa di formato condizionale che gestisce il tipo di dati numerico della misura.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempi

### Stringhe di formato

DAX

```
= FORMAT( 12345.67, "General Number")
= FORMAT( 12345.67, "Currency")
= FORMAT( 12345.67, "Fixed")
= FORMAT( 12345.67, "Standard")
= FORMAT( 12345.67, "Percent")
= FORMAT( 12345.67, "Scientific")
```

Restituisce:

12345.67 "General Number" visualizza il numero senza formattazione.

`$12,345.67` "Currency" visualizza il numero con la formattazione delle impostazioni locali di valuta. Questo esempio mostra la formattazione predefinita della valuta degli Stati Uniti.

`12345.67` "Fixed" visualizza almeno una cifra a sinistra del separatore decimale e due cifre a destra del separatore decimale.

`12.345,67` "Standard" visualizza almeno una cifra a sinistra del separatore decimale e due cifre a destra del separatore decimale e include il separatore delle migliaia. Questo esempio mostra la formattazione predefinita dei numeri degli Stati Uniti.

`1,234,567.00 %` "Percent" visualizza il numero come percentuale (moltiplicato per 100) con la formattazione e il segno di percentuale a destra del numero separato da uno spazio singolo.

`1.23E+04` "Scientific" visualizza il numero in notazione scientifica con due cifre decimali.

## Datetime con locale\_name facoltativo

DAX

```
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-US" )  
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-GB" )  
= FORMAT( dt"2020-12-15T12:30:59", "mm/dd/yyyy", "en-GB" )
```

Restituisce:

`15/12/2020 12:30:59` Dove il mese precede il giorno e l'ora è nel formato di 12 ore.

`12/15/2020 12:30:59` Dove il giorno precede il mese e l'ora è nel formato di 24 ore.

`15/12/2020 12:30:59` Dove il mese precede il giorno e l'ora è nel formato di 24 ore.

Poiché viene specificata una stringa di formato non dipendente dalle impostazioni locali, le impostazioni locali non vengono applicate e viene restituito il formato non locale.

## Formati numerici predefiniti

È possibile specificare i formati numerici predefiniti seguenti nell'argomento `format_string`:

Formato	Descrizione
"General"	Visualizza il numero senza separatore delle migliaia.
Number"	

<b>Formato</b>	<b>Descrizione</b>
"Currency"	Visualizza il numero con separatore delle migliaia, se appropriato. Visualizza due cifre a destra del separatore decimale. L'output dipende dalle impostazioni locali del sistema.
"Fixed"	Il numero viene visualizzato con almeno una cifra a sinistra e due a destra del separatore decimale.
"Standard"	Visualizza il numero con il separatore decimale, almeno una cifra a sinistra e due cifre a destra del separatore decimale.
"Percent"	Visualizza il numero moltiplicato per 100 con un segno di percentuale (%) aggiunto immediatamente a destra; visualizza sempre due cifre a destra del separatore decimale.
"Scientific"	Usa la notazione scientifica standard, che specifica die cifre significative.
"Yes/No"	Visualizza No se il numero è 0; in caso contrario, visualizza Sì.
"True/False"	Visualizza False se il numero è 0; in caso contrario, visualizza True.
"On/Off"	Visualizza Off se il numero è 0; in caso contrario, visualizza On.

## Formati numerici personalizzati

Un'espressione di formato personalizzata per i numeri può includere da una a tre sezioni separate da punti e virgola. Se l'argomento della stringa di formato include un formato numerico predefinito, è consentita una sola sezione.

<b>Se si usa</b>	<b>Il risultato è</b>
Una sola sezione	L'espressione di formato viene applicata a tutti i valori.
Due sezioni	La prima sezione viene applicata ai valori positivi e agli zeri, la seconda ai valori negativi.
Tre sezioni	La prima sezione viene applicata ai valori positivi, la seconda ai valori negativi e la terza agli zeri.

DAX

"\$#,##0; (\$#,##0)"

Se si specificano punti e virgola consecutivi, la sezione mancante viene definita usando il formato dei valori positivi. La stringa di formato seguente, ad esempio, visualizza i valori

positivi e negativi usando il formato specificato nella prima sezione e visualizza "Zero" se il valore è zero.

DAX

```
"$#,##0"
```

Se si specificano punti e virgola consecutivi, la sezione mancante viene visualizzata usando il formato dei valori positivi.

## Caratteri di formato numerico personalizzati

È possibile specificare i caratteri di formato numerico personalizzati seguenti nell'argomento **format\_string**:

Carattere	Descrizione
None	Visualizzare il numero senza formattazione.
(0)	Segnaposto della cifra. Visualizzare una cifra o uno zero. Se nell'espressione è presente una cifra nella posizione in cui il valore 0 viene visualizzato nella stringa di formato, visualizzarla. In caso contrario, visualizzare uno zero in tale posizione. Se il numero include meno cifre di quanti sono gli zeri (su entrambi i lati del separatore decimale) nell'espressione di formato, visualizzare zeri iniziali o finali. Se il numero ha più cifre a destra del separatore decimale del numero di zeri a destra del separatore decimale nell'espressione di formato, arrotondare il numero al numero di cifre decimali corrispondente al numero di zeri. Se il numero ha più cifre a sinistra del separatore decimale del numero di zeri a sinistra del separatore decimale nell'espressione di formato, visualizzare le cifre aggiuntive senza alcuna modifica.
(#)	Segnaposto della cifra. Visualizzare una cifra o nessun elemento. Se nell'espressione è presente una cifra nella posizione di # nella stringa di formato, visualizzarla; in caso contrario, non visualizzare alcun elemento. Questo simbolo viene usato allo stesso modo del segnaposto 0 cifre, ad eccezione del fatto che gli zeri iniziali e finali non vengono visualizzati se il numero ha un numero di cifre uguale o inferiore al numero di caratteri # su entrambi i lati del separatore decimale nell'espressione di formato.
(.)	Segnaposto decimale. In alcune impostazioni locali viene usata la virgola come separatore decimale. Il segnaposto decimale determina il numero di cifre visualizzate a sinistra e a destra del separatore decimale. Se l'espressione di formato contiene solo cancelletti a sinistra di questo simbolo, i numeri inferiori a 1 iniziano con un separatore decimale. Per visualizzare uno zero iniziale con i numeri frazionari, usare 0 come primo segnaposto di cifra a sinistra del separatore decimale. Il carattere effettivo usato come segnaposto decimale nell'output formattato dipende dal formato numerico riconosciuto dal sistema.
(%)	Segnaposto percentuale. L'espressione viene moltiplicata per 100. Il carattere di

<b>Carattere</b>	<b>Descrizione</b>
	percentuale (%) viene inserito nella posizione in cui compare nella stringa di formato.
(.)	Separatore delle migliaia. In alcune impostazioni locali viene usato il punto come separatore delle migliaia. Il separatore delle migliaia separa le migliaia dalle centinaia all'interno di un numero con quattro o più posizioni a sinistra del separatore decimale. Se il formato include un separatore delle migliaia racchiuso da segnaposto di cifra (0 o #), il separatore delle migliaia viene usato in modo standard. Due separatori delle migliaia adiacenti o un separatore delle migliaia immediatamente a sinistra del separatore decimale (indipendentemente dal fatto che sia specificato un decimale) indica il "ridimensionamento del numero tramite divisione per 1000 con eventuale arrotondamento". Ad esempio, è possibile usare la stringa di formato "##0.," per rappresentare 100 milioni come 100. I numeri minori di 1 milione vengono visualizzati come 0. Due separatori delle migliaia adiacenti in una posizione diversa dalla posizione immediatamente a sinistra del separatore decimale vengono considerati semplicemente la specifica dell'utilizzo di un separatore delle migliaia. Il carattere effettivo usato come segnaposto delle migliaia nell'output formattato dipende dal formato numerico riconosciuto dal sistema.
(:)	Separatore dell'ora. In alcune impostazioni locali potrebbero essere usati altri caratteri per rappresentare il separatore dell'ora. Il separatore dell'ora separa le ore, i minuti e i secondi quando vengono formattati i valori di ora. Il carattere effettivo usato come separatore dell'ora nell'output formattato è determinato dalle impostazioni di sistema.
(/)	Separatore della data. In alcune impostazioni locali potrebbero essere usati altri caratteri per rappresentare il separatore della data. Il separatore di data separa il giorno, il mese e l'anno in cui vengono formattati i valori di data. Il carattere effettivo usato come separatore della data nell'output formattato è determinato dalle impostazioni di sistema.
(E- E+ e- e+)	Formato scientifico. Se l'espressione di formato contiene almeno un segnaposto di cifra (0 oppure #) a destra di E-, E+, e- o e+, il numero viene visualizzato in formato scientifico e viene inserito E o e tra il numero e il relativo esponente. Il numero di segnaposti di cifra a destra determina il numero di cifre nell'esponente. Usare E- o e- per inserire un segno meno accanto agli esponenti negativi. Usare E+ o e+ per inserire un segno meno accanto agli esponenti negativi e un segno più accanto agli esponenti positivi.
- + \$ ()	Visualizzare un carattere letterale. Per visualizzare un carattere diverso da uno di quelli elencati, anteporvi una barra rovesciata (\) o racchiuderlo tra virgolette doppie ("").
\	Visualizzare il carattere successivo nella stringa di formato. Per visualizzare un carattere che ha un significato speciale come carattere letterale, farlo precedere da una barra rovesciata (\). La barra rovesciata non viene visualizzata. Utilizzare una barra rovesciata equivale a racchiudere il carattere successivo tra virgolette doppie. Per visualizzare una barra rovesciata, usare due barre rovesciate (\ \). I caratteri di formattazione di data e ora (a, c, d, h, m, n, p, q, s, t, w, y, / e :), i caratteri di formattazione numerica (#, 0, %, E, e, virgola e punto) e il caratteri di formattazione di

Carattere	Descrizione
	stringa (@, &, <, > e !) sono esempi di caratteri che non possono essere visualizzati come caratteri letterali.
("ABC")	Visualizzare la stringa racchiusa tra virgolette doppie (" ") .

## Formati data/ora predefiniti

È possibile specificare i formati di data/ora predefiniti seguenti nell'argomento **format\_string**. Quando si usano formati diversi da questi, vengono interpretati come formato di data/ora personalizzato:

Formato	Descrizione
"General Date"	Vengono visualizzate una data e/o un'ora. Ad esempio: 3/12/2008 11:07:31. La visualizzazione relativa alla data è determinata dal valore delle impostazioni cultura correnti dell'applicazione.
"Long Date" oppure "Medium Date"	Visualizza una data in base al formato di data estesa delle impostazioni cultura correnti. Ad esempio: mercoledì 12 marzo 2008.
"Short Date"	Visualizza una data in base al formato di data breve delle impostazioni cultura correnti. Ad esempio: 3/12/2008.
"Long Time" oppure	Visualizza un'ora in base al formato ora estesa delle impostazioni cultura correnti; in genere comprende ore, minuti, secondi. Ad esempio: 11:07:31.
"Medium Time"	Visualizza l'ora nel formato 12 ore. Ad esempio: 11:07.
"Short Time"	Visualizza l'ora nel formato 24 ore. Ad esempio: 23:07.

## Formati di data/ora personalizzati

È possibile specificare i caratteri di formato seguenti in **format\_string** per creare formati di data/ora personalizzati:

Carattere	Descrizione
(:)	Separatore dell'ora. In alcune impostazioni locali potrebbero essere usati altri caratteri per rappresentare il separatore dell'ora. Il separatore dell'ora separa le ore, i minuti e i secondi quando vengono formattati i valori di ora. Il carattere effettivo usato come separatore dell'ora nell'output formattato è determinato dalle impostazioni di sistema.

<b>Carattere</b>	<b>Descrizione</b>
(/)	Separatore della data. In alcune impostazioni locali potrebbero essere usati altri caratteri per rappresentare il separatore della data. Il separatore di data separa il giorno, il mese e l'anno in cui vengono formattati i valori di data. Il carattere effettivo usato come separatore della data nell'output formattato è determinato dalle impostazioni di sistema.
(\)	Barra rovesciata. Visualizza il carattere successivo come carattere letterale. Non viene quindi interpretato come carattere di formattazione.
("")	Virgolette. Viene visualizzato il testo racchiuso tra virgolette. Non viene quindi interpretato come carattere di formattazione.
c	Visualizza la data come <code>ddddd</code> e l'ora come <code>ttttt</code> , in questo ordine. Visualizzare solo le informazioni sulla data se non esiste alcuna parte frazionaria per il numero seriale della data. Visualizzare solo le informazioni sull'ora se non esiste una parte intera.
g	Visualizza il giorno sotto forma di numero senza zero iniziale (1-31).
dd	Visualizza il giorno sotto forma di numero con uno zero iniziale (01-31).
ddd	Visualizza il nome abbreviato del giorno (lun-dom). Localizzato.
dddd	Visualizza il nome completo del giorno (lunedì-domenica). Localizzato.
ddddd	Visualizzare la data completa, inclusi il giorno, il mese e l'anno, formattata in base all'impostazione del formato di data breve del sistema in uso. Il formato di data breve predefinito è <code>mm/dd/yyyy</code> .
ddddddd	Visualizzare un numero seriale di data sotto forma di una data completa, inclusi il giorno, il mese e l'anno, formattata in base all'impostazione del formato di data estesa riconosciuta dal sistema in uso. Il formato di data estesa predefinito è <code>ddd, mmmm d, yyyy</code> .
w	Visualizzare il giorno della settimana sotto forma di numero (1 corrisponde a domenica e 7 a sabato).
ww	Visualizza la settimana dell'anno sotto forma di numero (1-54).
m	Visualizza il mese sotto forma di numero senza zero iniziale (1-12). Se è presente <code>m</code> immediatamente dopo <code>h</code> o <code>hh</code> , vengono visualizzati i minuti al posto del mese.
mm	Visualizza il mese sotto forma di numero con uno zero iniziale (01-12). Se è presente <code>mm</code> immediatamente dopo <code>h</code> o <code>hh</code> , vengono visualizzati i minuti al posto del mese.
mmm	Visualizza il mese sotto forma di abbreviazione (gen-dic). Localizzato.
mmmm	Visualizza il mese sotto forma di nome completo (gennaio-dicembre). Localizzato.
q	Visualizza il trimestre dell'anno sotto forma di numero (1-4).

<b>Carattere</b>	<b>Descrizione</b>
y	Visualizza il giorno dell'anno sotto forma di numero (1-366).
yy	Visualizza l'anno sotto forma di numero a 2 cifre (00-99).
yyyy	Visualizza l'anno sotto forma di numero a 4 cifre (100-9999).
h	Visualizza l'ora sotto forma di numero senza zero iniziale (0-23).
hh	Visualizza l'ora sotto forma di numero con uno zero iniziale (00-23).
n	Visualizza i minuti sotto forma di numero senza zero iniziale (0-59).
nn	Visualizza i minuti sotto forma di numero con uno zero iniziale (00-59).
s	Visualizza i secondi sotto forma di numero senza zero iniziale (0-59).
ss	Visualizza i secondi sotto forma di numero con uno zero iniziale (00-59).
tttt	Visualizzare l'ora completa, inclusi ora, minuti e secondi, formattata usando il separatore dell'ora definito dal formato di ora riconosciuto dal sistema in uso. Viene visualizzato uno zero iniziale se è selezionata l'opzione relativa allo zero iniziale e l'ora è anteriore alle 10.00 del mattutino o della sera. Il formato di ora predefinito è <code>h:mm:ss</code> .
AM/PM	Usare il formato a 12 ore e visualizzare AM in lettere maiuscole per le ore antecedenti a mezzogiorno; visualizzare PM in lettere maiuscole per le ore tra mezzogiorno e 11:59 P.M.
am/pm	Usare il formato a 12 ore e visualizzare AM in lettere minuscole per le ore antecedenti a mezzogiorno; visualizzare PM in lettere minuscole per le ore tra mezzogiorno e 11:59 P.M.
A/P	Usare il formato a 12 ore e visualizzare una A maiuscola per le ore antecedenti a mezzogiorno; visualizzare una P maiuscola per le ore tra mezzogiorno e 11:59 P.M.
a/p	Usare il formato a 12 ore e visualizzare una A minuscola per le ore antecedenti a mezzogiorno; visualizzare una P minuscola per le ore tra mezzogiorno e 11:59 P.M.
AMPM	Usare il formato a 12 ore e visualizzare il valore letterale stringa AM come definito dal sistema per le ore antecedenti a mezzogiorno; visualizzare il valore letterale stringa PM come definito dal sistema per le ore tra mezzogiorno e 11:59 P.M. AMPM può essere in lettere maiuscole o minuscole, ma la combinazione maiuscole/minuscole della stringa visualizzata corrisponderà alla stringa definita dalle impostazioni di sistema. Il formato predefinito è AM/PM. Se il sistema è impostato su un formato a 24 ore, la stringa viene in genere impostata su una stringa vuota.

Per la formattazione di data/ora vengono usate le impostazioni locali correnti dell'utente per formattare la stringa. Considerare ad esempio la data 25 giugno 2020. Quando viene formattata con la stringa di formato "m/d/yyyy", il risultato sarà:

- Le impostazioni locali dell'utente corrispondono a Stati Uniti d'America (en-US): "6/25/2020"
- Le impostazioni locali dell'utente corrispondono a Germania (de-DE): "6.25.2020"

## Esempi di formato di data/ora personalizzato

Gli esempi seguenti usano come data/ora giovedì 25 giugno 2020 alle 13:23:45. Germania (de-DE) usa un sistema a 24 ore. Non esiste un equivalente di AM/PM.

<b>Formatta</b>	<b>Risultato (en-US)</b>	<b>Risultato (de-DE)</b>
"c"	06/25/2020 13:23:45	25.06.2020 13:23:45
"d"	25	25
"dd"	25	25
"ddd"	Gio	Procedura consigliata
"dddd"	giovedì	Donnerstag
"ddddd"	06/25/2020	25.06.2020
"dddddd"	Thursday, June 25, 2020	Donnerstag, 25. Juni 2020
"w"	5	5
"ww"	26	26
"m"	6	6
"mm"	06	06
"mmm"	Giu	Giu
"mmmm"	Giugno	Juni
"q"	2	2
"y"	177	177
"yy"	20	20
"yyyy"	2020	2020
""""Year"" yyyy"	Year 2020	Year 2020
"yyyy \Qq"	2020 Q2	2020 Q2
"dd/mm/yyyy"	25/06/2020	25.06.2020

<b>Formatta</b>	<b>Risultato (en-US)</b>	<b>Risultato (de-DE)</b>
"mm/dd/yyyy"	06/25/2020	06.25.2020
"h:nn:ss"	13:23:45	13:23:45
"h:nn:ss AMPM"	1:23:45 PM	1:23:45
"hh:nn:ss"	13:23:45	13:23:45
"hh:nn:ss AMPM"	01:23:45 PM	01:23:45
"tttt"	13:23:45	13:23:45
"tttt AMPM"	13:23:45 PM	13:23:45
"mm/dd/yyyy hh:nn:ss AMPM"	06/25/2020 01:23:45 PM	6.25.2020 01:23:45

## Commenti e suggerimenti

Questa pagina è stata utile?



# LEFT

Articolo • 20/10/2023

Restituisce il numero specificato di caratteri dall'inizio di una stringa di testo.

## Sintassi

DAX

```
LEFT(<text>, <num_chars>)
```

## Parametri

Termine	Definizione
testo	Stringa di testo contenente i caratteri da estrarre oppure riferimento a una colonna contenente testo.
num_chars	(facoltativo) Numero di caratteri da estrarre tramite LEFT. Se omesso, viene specificato 1.

## Valore restituito

Stringa di testo.

## Osservazioni:

- Mentre Microsoft Excel contiene diverse funzioni per l'utilizzo di testo in lingue con caratteri a singolo e a doppio byte, DAX usa Unicode e pertanto tutti i caratteri vengono archiviati con la stessa lunghezza. Per questo motivo, è sufficiente una singola funzione.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

Nell'esempio seguente vengono restituiti i primi cinque caratteri del nome dell'azienda nella colonna [ResellerName] e le prime cinque lettere del codice geografico nella

colonna [GeographyKey] e questi valori vengono concatenati per creare un identificatore.

DAX

```
= CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))
```

Se l'argomento **num\_chars** è un numero maggiore del numero di caratteri disponibili, la funzione restituisce il numero massimo di caratteri disponibili e non genera un errore. La colonna [GeographyKey], ad esempio, contiene numeri come 1, 12 e 311, pertanto anche il risultato ha lunghezza variabile.

## Vedi anche

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# LEN

Articolo • 20/10/2023

Restituisce il numero di caratteri in una stringa di testo.

## Sintassi

DAX

```
LEN(<text>)
```

## Parametri

Termine	Definizione
testo	Testo di cui si vuole trovare la lunghezza o colonna contenente il testo. Gli spazi vengono conteggiati come caratteri.

## Valore restituito

Numero intero che specifica il numero di caratteri nella stringa di testo.

## Osservazioni:

- Mentre Microsoft Excel ha diverse funzioni per l'utilizzo di testo in lingue con caratteri a singolo e a doppio byte, DAX usa Unicode e pertanto tutti i caratteri vengono archiviati con la stessa lunghezza.
- LEN calcola sempre 1 per ogni carattere, a prescindere dall'impostazione della lingua predefinita.
- Se si usa LEN con una colonna che contiene valori non di testo, ad esempio date o valori booleani, la funzione esegue il cast implicito del valore in testo usando il formato della colonna corrente.

## Esempio

La formula seguente somma le lunghezze degli indirizzi nelle colonne [AddressLine1] e [AddressLine2].

DAX

```
= LEN([AddressLine1])+LEN([AddressLin2])
```

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# LOWER

Articolo • 20/10/2023

Converte tutte le lettere di una stringa di testo in caratteri minuscoli.

## Sintassi

DAX

```
LOWER(<text>)
```

## Parametri

Termine	Definizione
testo	Testo che si vuole convertire in minuscole o riferimento a una colonna contenente testo.

## Valore restituito

Testo in minuscolo.

## Osservazioni:

I caratteri diversi dalle lettere non vengono modificati. La formula `= LOWER("123ABC")`, ad esempio, restituisce `123abc`.

## Esempio

La formula seguente ottiene ogni riga della colonna [ProductCode] e converte il valore in minuscole. Questa operazione non influisce sui numeri nella colonna.

DAX

```
= LOWER('New Products'[ProductCode])
```

## Vedi anche

## Commenti e suggerimenti

Questa pagina è stata utile?



# MID

Articolo • 20/10/2023

Restituisce una stringa di caratteri dal centro di una stringa di testo, date una posizione di inizio e una lunghezza.

## Sintassi

DAX

```
MID(<text>, <start_num>, <num_chars>)
```

## Parametri

Termine	Definizione
testo	Stringa di testo da cui estrarre i caratteri o colonna contenente testo.
start_num	Posizione del primo carattere da estrarre. Le posizioni iniziano da 1.
num_chars	Il numero di caratteri da restituire.

## Valore restituito

Stringa di testo della lunghezza specificata.

## Osservazioni:

Mentre Microsoft Excel ha diverse funzioni per l'utilizzo di testo in lingue con caratteri a singolo e a doppio byte, DAX usa Unicode e pertanto tutti i caratteri vengono archiviati con la stessa lunghezza.

## Esempi

L'espressione seguente,

DAX

```
MID("abcde",2,3))
```

Restituisce "bcd".

L'espressione seguente,

DAX

```
MID('Reseller'[ResellerName],1,5)
```

Restituisce lo stesso risultato di `LEFT([ResellerName],5)`. Entrambe le espressioni restituiscono le prime 5 lettere della colonna, `[ResellerName]`.

## Vedi anche

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# REPLACE

Articolo • 20/10/2023

REPLACE sostituisce parte di una stringa di testo, in base al numero di caratteri specificati, con una stringa di testo diversa.

## Sintassi

DAX

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

## Parametri

Termine	Definizione
old_text	Stringa di testo che contiene i caratteri da sostituire oppure riferimento a una colonna contenente testo.
start_num	Posizione del carattere in <b>old_text</b> che si vuole sostituire con <b>new_text</b> .
num_chars	Numero di caratteri da sostituire. <b>Avviso:</b> se l'argomento <b>num_chars</b> è un valore vuoto o fa riferimento a una colonna che restituisce un valore vuoto, la stringa per <b>new_text</b> viene inserita nella posizione <b>start_num</b> , senza sostituire alcun carattere. Questo comportamento è identico a quello di Excel.
new_text	Testo sostitutivo per i caratteri specificati in <b>old_text</b> .

## Valore restituito

Stringa di testo.

## Osservazioni:

- Mentre Microsoft Excel ha diverse funzioni per l'uso di testo in lingue con caratteri a singolo e a doppio byte, DAX usa Unicode e quindi tutti i caratteri vengono archiviati con la stessa lunghezza.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula seguente crea una nuova colonna calcolata che sostituisce i primi due caratteri del codice prodotto nella colonna [ProductCode] con un nuovo codice di due lettere, OB.

DAX

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

## Vedi anche

[Funzioni di testo](#)

[funzione SUBSTITUTE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# REPT

Articolo • 20/10/2023

Ripete il testo un determinato numero di volte. Usare REPT per completare una cella con un numero di istanze di una stringa di testo.

## Sintassi

DAX

```
REPT(<text>, <num_times>)
```

## Parametri

Termine	Definizione
testo	Testo da ripetere.
num_times	Numero positivo che specifica quante volte ripetere il testo.

## Valore restituito

Stringa contenente le modifiche.

## Osservazioni:

- Se **number\_times** è 0 (zero), REPT restituisce un valore blank.
- Se **number\_times** non è un numero intero, la parte decimale viene troncata.
- Il risultato della funzione REPT non può contenere più di 32.767 caratteri. In caso contrario, REPT restituisce un errore.

## Esempio: Ripetizione di stringhe letterali

L'esempio seguente restituisce la stringa 85, ripetuta tre volte.

DAX

```
= REPT("85",3)
```

## Esempio: Ripetizione dei valori di colonna

L'esempio seguente restituisce la stringa nella colonna [MyText] ripetuta per il numero di volte indicato nella colonna [MyNumber]. Poiché la formula si estende per l'intera colonna, la stringa risultante dipende dal testo e dal valore del numero in ogni riga.

DAX

```
= REPT([MyText],[MyNumber])
```

MyText	MyNumber	CalculatedColumn1
Testo	2	TextText
Numero	0	
85	3	858585

## Vedi anche

[Funzioni di testo](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# RIGHT

Articolo • 20/10/2023

RIGHT restituisce uno o più caratteri finali in una stringa di testo, in base al numero di caratteri specificato.

## Sintassi

DAX

```
RIGHT(<text>, <num_chars>)
```

## Parametri

Termine	Definizione
testo	Stringa di testo che contiene i caratteri da estrarre oppure riferimento a una colonna che contiene testo.
num_chars	(facoltativo) Numero di caratteri da estrarre tramite RIGHT. Se omesso, viene specificato 1. È anche possibile usare un riferimento a una colonna contenente numeri.

Se il riferimento di colonna non contiene testo, viene trasmesso in modo implicito come testo.

## Valore restituito

Stringa di testo contenente i caratteri più a destra specificati.

## Osservazioni:

- RIGHT calcola sempre 1 per ogni carattere, che sia a un byte o a doppio byte, a prescindere dall'impostazione della lingua predefinita.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio: restituzione di un numero fisso di caratteri

La formula seguente restituisce le ultime due cifre del codice del prodotto nella tabella New Products.

DAX

```
= RIGHT('New Products'[ProductCode],2)
```

# Esempio: utilizzo di un riferimento di colonna per specificare il numero dei caratteri

La formula seguente restituisce un numero variabile di cifre dal codice del prodotto nella tabella New Products, a seconda del numero nella colonna MyCount. Se non è presente alcun valore nella colonna MyCount o il valore è vuoto, anche RIGHT restituisce un valore vuoto.

DAX

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

## Vedi anche

[Funzioni di testo](#)

[LEFT](#)

[MID](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# Cerca...

Articolo • 20/10/2023

Restituisce il numero del carattere in corrispondenza del quale viene individuato per la prima volta un carattere o una stringa di testo specifica, leggendo da sinistra a destra. La ricerca non distingue tra maiuscole e minuscole, ma distingue tra caratteri accentati e non accentati.

## Sintassi

DAX

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

## Parametri

Termine	Definizione
find_text	Testo da cercare.  In <b>find_text</b> è possibile usare caratteri jolly, ad esempio il punto interrogativo (?) e l'asterisco (*). Un punto interrogativo corrisponde a qualsiasi carattere singolo. Un asterisco corrisponde a qualsiasi sequenza di caratteri. Per trovare un punto interrogativo o un asterisco effettivo, digitare una tilde (~) prima del carattere.
within_text	Testo in cui cercare <b>find_text</b> o colonna contenente il testo.
start_num	(facoltativo) Posizione del carattere in <b>within_text</b> da cui iniziare la ricerca. Se omesso, il valore è 1.
NotFoundValue	(facoltativo, ma fortemente consigliato) Valore che deve essere restituito quando l'operazione non trova una substring corrispondente, in genere 0, -1 o BLANK(). Se non specificato, viene restituito un errore.

## Valore restituito

Numero della posizione iniziale della prima stringa di testo a partire dal primo carattere della seconda stringa di testo.

## Osservazioni:

- La funzione di ricerca non distingue tra maiuscole e minuscole. La ricerca di "N" troverà la prima occorrenza di "N" o di "n".
- La funzione di ricerca distingue tra caratteri accentati e non accentati. La ricerca di "á" troverà la prima occorrenza di "á", ma nessuna occorrenza di "a", "à" o delle versioni maiuscole "A", "Á".
- È possibile usare la funzione SEARCH per determinare la posizione di un carattere o di una stringa di testo all'interno di un'altra stringa di testo e quindi usare la funzione MID per restituire il testo oppure usare la funzione REPLACE per modificare il testo.
- Se non è possibile trovare **find\_text** in **within\_text**, la formula restituisce un errore. Questo comportamento è simile a quello di Excel, che restituisce #VALORE se la sottostringa non viene trovata. In questo contesto i valori Null in **within\_text** verranno interpretati come una stringa vuota.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente trova la posizione della prima lettera di "cycle" nella stringa che contiene il nome del rivenditore. Se non viene trovato, viene restituito BLANK.

SEARCH non fa distinzione tra maiuscole e minuscole. In questo esempio, se nell'argomento <find\_text> viene usato "cycle" o "Cycle", vengono restituiti risultati per entrambe le combinazioni di caratteri. Usare **FIND** per la distinzione tra maiuscole e minuscole.

*Gli esempi in questo articolo possono essere usati con il modello di esempio Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of cycle", SEARCH ( "cycle", 'Reseller'[Reseller], 1,
BLANK () )
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
```

```
    Reseller", "Warehouse"}  
)
```

Restituisce:

Reseller	Business Type	Posizione di "cycle"
Volume Bike Sellers	Magazzino	
Mass Market Bikes	Value Added Reseller	
Twin Cycles	Value Added Reseller	6
Rich Department Store	Magazzino	
Rental Gallery	Specialty Bike Shop	
Budget Toy Store	Magazzino	
Global Sports Outlet	Magazzino	
Online Bike Catalog	Magazzino	
Helmets and Cycles	Value Added Reseller	13
Jumbo Bikes	Specialty Bike Shop	

## Vedi anche

[FIND](#)

[REPLACE](#)

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì    No

# SUBSTITUTE

Articolo • 20/10/2023

Sostituisce il nuovo testo a quello esistente in una stringa di testo.

## Sintassi

DAX

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

## Parametri

Termino	Definizione
testo	Testo in cui sostituire i caratteri o riferimento a una colonna contenente testo.
old_text	Testo esistente che si vuole sostituire.
new_text	Testo con cui sostituire <b>old_text</b> .
instance_num	(facoltativo) Occorrenza di <b>old_text</b> che si vuole sostituire. Se omesso, viene sostituita ogni istanza di <b>old_text</b>

## Valore restituito

Stringa di testo.

## Osservazioni:

- Usare la funzione SUBSTITUTE quando si vuole sostituire un testo specifico in una stringa di testo. Usare la funzione REPLACE quando si vuole sostituire qualsiasi testo di lunghezza variabile presente in una posizione specifica in una stringa di testo.
- La funzione SUBSTITUTE fa distinzione tra maiuscole e minuscole. Se l'uso delle maiuscole e delle minuscole non corrisponde in **text** e **old\_text**, SUBSTITUTE non sostituirà il testo.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio: sostituzione all'interno di un stringa

La formula seguente crea una copia della colonna [Product Code] che sostituisce il nuovo codice prodotto **NW** per il codice prodotto precedente **PA** ovunque si trovi nella colonna.

DAX

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

## Vedi anche

[Funzioni di testo](#)

[REPLACE](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# TRIM

Articolo • 20/10/2023

Rimuove tutti gli spazi dal testo esclusi gli spazi singoli che dividono le parole.

## Sintassi

DAX

```
TRIM(<text>)
```

## Parametri

Termine	Definizione
text	Testo da cui si vogliono rimuovere gli spazi o colonna contenente il testo.

## Valore restituito

Stringa con gli spazi rimossi.

## Osservazioni:

- Usare TRIM sul testo ricevuto da un'altra applicazione che potrebbe avere una spaziatura irregolare.
- La funzione TRIM è stata originariamente progettata per tagliare lo spazio ASCII a 7 bit (valore 32) dal testo. Nel set di caratteri Unicode è presente un altro spazio denominato spazio unificatore con un valore decimale di 160. Questo carattere viene comunemente usato nelle pagine Web come entità HTML, &nbsp;. Di per sé, la funzione TRIM non rimuove questo spazio unificatore. Per un esempio di come tagliare entrambi gli spazi dal testo, vedere Rimuovere gli spazi e i caratteri non stampabili dal testo.

## Esempio

La formula seguente crea una nuova stringa senza spazi vuoti finali.

DAX

```
= TRIM("A column with trailing spaces.    ")
```

Quando si crea la formula, poiché la formula viene propagata nella riga esattamente come è stata digitata, in ogni formula viene visualizzata la stringa originale e i risultati non sono evidenti. Tuttavia, quando la formula viene valutata, la stringa viene tagliata.

È possibile verificare che la formula produca il risultato corretto controllando la lunghezza della colonna calcolata creata dalla formula precedente, come indicato di seguito:

DAX

```
= LEN([Calculated Column 1])
```

## Vedi anche

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# UNICHAR

Articolo • 20/10/2023

Restituisce il carattere Unicode a cui fa riferimento il valore numerico.

## Sintassi

DAX

```
UNICHAR(number)
```

## Parametri

Termine	Definizione
number	Numero Unicode che rappresenta il carattere.

## Valore restituito

Carattere rappresentato dal numero Unicode.

## Osservazioni:

- Se i caratteri XML non sono validi, UNICHAR restituisce un errore.
- Se i numeri Unicode sono surrogati parziali e i tipi di dati non sono validi, UNICHAR restituisce un errore.
- Se i numeri sono valori numerici non compresi nell'intervallo consentito, UNICHAR restituisce un errore.
- Se il numero è zero (0), UNICHAR restituisce un errore.
- Il carattere Unicode restituito può essere una stringa di caratteri, ad esempio nei codici UTF-8 o UTF-16.

## Esempio

L'esempio seguente restituisce il carattere rappresentato dal numero Unicode 66 (A maiuscola).

```
DAX  
= UNICHAR(65)
```

L'esempio seguente restituisce il carattere rappresentato dal numero Unicode 32 (spazio).

```
DAX  
= UNICHAR(32)
```

L'esempio seguente restituisce il carattere rappresentato dal numero Unicode 9733 (carattere ★).

```
DAX  
= UNICHAR(9733)
```

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# UNICODE

Articolo • 20/10/2023

Restituisce il numero (punto di codice) corrispondente al primo carattere del testo.

## Sintassi

DAX

```
UNICODE( <Text> )
```

## Parametri

Termine	Definizione
Testo	Il testo è il carattere per cui si vuole il valore Unicode.

## Valore restituito

Codice numerico del primo carattere di una stringa di testo.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# UPPER

Articolo • 20/10/2023

Converte una stringa di testo in caratteri maiuscoli.

## Sintassi

DAX

```
UPPER (<text>)
```

## Parametri

Termine	Definizione
testo	Testo da convertire in caratteri maiuscoli o riferimento a una colonna che contiene testo.

## Valore restituito

Lo stesso testo, in maiuscolo.

## Esempio

La formula seguente converte l'intera stringa della colonna, [ProductCode], in maiuscolo. I caratteri non alfabetici non sono interessati.

DAX

```
= UPPER([ 'New Products'[Product Code])
```

## Vedi anche

[Funzioni di testo](#)

[Funzione LOWER](#)

# Commenti e suggerimenti

Questa pagina è stata utile?



# VALORE

Articolo • 20/10/2023

Converte una stringa di testo che rappresenta un numero in un numero.

## Sintassi

DAX

`VALUE(<text>)`

## Parametri

Termine	Definizione
testo	Testo da convertire.

## Valore restituito

Numero convertito nel tipo di dati decimale.

## Osservazioni:

- Il valore passato come parametro **text** può essere in qualsiasi formato di costante, numero, data o ora riconosciuto dall'applicazione o dai servizi usati. Se **text** non è in uno di questi formati, viene restituito un errore.
- In genere non è necessario usare la funzione VALUE in una formula perché il motore converte in modo implicito il testo in numeri, se necessario.
- È anche possibile usare riferimenti a colonne. Se ad esempio è presente una colonna che contiene tipi di numeri misti, è possibile usare VALUE per convertire tutti i valori in un singolo tipo di dati numerico. Se tuttavia si usa la funzione VALUE con una colonna che contiene numeri e testo misti, l'intera colonna viene contrassegnata con un errore, perché non tutti i valori di tutte le righe possono essere convertiti in numeri.

## Esempio

La formula seguente converte la stringa tipizzata "3" nel valore numerico 3.

DAX

```
= VALUE("3")
```

## Vedi anche

[Funzioni di testo](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì     No

# Funzioni di Business Intelligence per le gerarchie temporali

Articolo • 20/10/2023

Data Analysis Expressions (DAX) include funzioni di Business Intelligence per le gerarchie temporali che consentono di modificare i dati usando periodi di tempo, tra cui giorni, mesi, trimestri e anni, e quindi di compilare e confrontare calcoli su tali periodi. Prima di usare qualsiasi funzione di Business Intelligence per le gerarchie temporali, assicurarsi di contrassegnare una delle tabelle contenenti la colonna data come Tabella data.

## In questa categoria

Funzione	Descrizione
<a href="#">CLOSINGBALANCEMONTH</a>	Valuta l'espressione in corrispondenza dell'ultima data del mese nel contesto corrente.
<a href="#">CLOSINGBALANCEQUARTER</a>	Valuta l'espressione in corrispondenza dell'ultima data del trimestre nel contesto corrente.
<a href="#">CLOSINGBALANCEYEAR</a>	Valuta l'espressione in corrispondenza dell'ultima data dell'anno nel contesto corrente.
<a href="#">DATEADD</a>	Restituisce una tabella che contiene una colonna di date, spostate in avanti o indietro nel tempo in base al numero specificato di intervalli rispetto alle date nel contesto corrente.
<a href="#">DATESBETWEEN</a>	Restituisce una tabella che contiene una colonna di date che inizia con una data di inizio specificata e continua fino a una data di fine specificata.
<a href="#">DATESINPERIOD</a>	Restituisce una tabella che contiene una colonna di date che inizia con una data di inizio specificata e continua in base al numero e al tipo di intervalli di data specificati.
<a href="#">DATESMTD</a>	Restituisce una tabella contenente una colonna delle date da inizio mese, nel contesto corrente.
<a href="#">DATESQTD</a>	Restituisce una tabella contenente una colonna delle date da inizio trimestre, nel contesto corrente.
<a href="#">DATESYTD</a>	Restituisce una tabella contenente una colonna delle date da inizio anno, nel contesto corrente.
<a href="#">ENDOFMONTH</a>	Restituisce l'ultima data del mese nel contesto corrente per la

Funzione	Descrizione
	colonna di date specificata.
ENDOFQUARTER	Restituisce l'ultima data del trimestre nel contesto corrente per la colonna di date specificata.
ENDOFYEAR	Restituisce l'ultima data dell'anno nel contesto corrente per la colonna di date specificata.
FIRSTDATE	Restituisce la prima data nel contesto corrente per la colonna di date specificata.
FIRSTNONBLANK	Restituisce il primo valore nella colonna column, filtrata in base al contesto corrente in cui l'espressione non è vuota
LASTDATE	Restituisce l'ultima data nel contesto corrente per la colonna di date specificata.
LASTNONBLANK	Restituisce l'ultimo valore nella colonna column, filtrata in base al contesto corrente, in cui l'espressione non è vuota.
NEXTDAY	Restituisce una tabella che contiene una colonna di tutte le date dal giorno successivo, in base alla prima data specificata nella colonna dates nel contesto corrente.
NEXTMONTH	Restituisce una tabella che contiene una colonna di tutte le date del mese successivo, in base alla prima data presente nella colonna dates nel contesto corrente.
NEXTQUARTER	Restituisce una tabella che contiene una colonna di tutte le date del trimestre successivo, basata sulla prima data specificata nella colonna dates, nel contesto corrente.
NEXTYEAR	Restituisce una tabella che contiene una colonna di tutte le date dell'anno successivo, in base alla prima data presente nella colonna dates, nel contesto corrente.
OPENINGBALANCEMONTH	Valuta l'espressione in corrispondenza della prima data del mese nel contesto corrente.
OPENINGBALANCEQUARTER	Valuta l'espressione in corrispondenza della prima data del trimestre, nel contesto corrente.
OPENINGBALANCYEAR	Valuta l'espressione in corrispondenza della prima data dell'anno nel contesto corrente.
PARALLELPERIOD	Restituisce una tabella che contiene una colonna di date che rappresenta un periodo parallelo alle date nella colonna dates specificata nel contesto corrente, con le date spostate di un numero di intervalli in avanti o indietro nel tempo.

<b>Funzione</b>	<b>Descrizione</b>
PREVIOUSDAY	Restituisce una tabella che contiene una colonna di tutte le date che rappresentano il giorno precedente alla prima data presente nella colonna dates, nel contesto corrente.
PREVIOUSMONTH	Restituisce una tabella che contiene una colonna di tutte le date del mese precedente, in base alla prima data presente nella colonna dates nel contesto corrente.
PREVIOUSQUARTER	Restituisce una tabella che contiene una colonna di tutte le date del trimestre precedente, in base alla prima data presente nella colonna dates, nel contesto corrente.
PREVIOUSYEAR	Restituisce una tabella che contiene una colonna di tutte le date dell'anno precedente, in base all'ultima data presente nella colonna dates, nel contesto corrente.
SAMEPERIODLASTYEAR	Restituisce una tabella che contiene una colonna di date spostate indietro di un anno rispetto alle date nella colonna dates specificata, nel contesto corrente.
STARTOFMONTH	Restituisce la prima data del mese nel contesto corrente per la colonna di date specificata.
STARTOFQUARTER	Restituisce la prima data del trimestre nel contesto corrente per la colonna di date specificata.
STARTOFYEAR	Restituisce la prima data dell'anno nel contesto corrente per la colonna di date specificata.
TOTALMTD	Valuta il valore dell'espressione da inizio mese, nel contesto corrente.
TOTALQTD	Valuta il valore dell'espressione per le date da inizio trimestre, nel contesto corrente.
TOTALYTD	Valuta il valore dell'espressione da inizio anno nel contesto corrente.

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# CLOSINGBALANCEMONTH

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza dell'ultima data del mese nel contesto corrente.

## Sintassi

DAX

```
CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

## Parametri

Parametro	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza dell'ultima data del mese nel contesto corrente.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

### ⚠ Nota

I vincoli per le espressioni booleane sono descritti in [Funzione CALCULATE](#).

### ⚠ Nota

Per l'espressione **filter** esistono le restrizioni descritte in [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore "Month End Inventory Value" dell'inventario dei prodotti.

DAX

```
=  
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product  
Inventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEYEAR](#)

[Funzione CLOSINGBALANCEQUARTER](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# CLOSINGBALANCEQUARTER

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza dell'ultima data del trimestre nel contesto corrente.

## Sintassi

DAX

```
CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

## Parametri

Termine	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza dell'ultima data del trimestre nel contesto corrente.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

### ① Nota

I vincoli per le espressioni booleane sono descritti in [Funzione CALCULATE](#).

### ① Nota

Per l'espressione **filter** esistono le restrizioni descritte in [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore "Quarter End Inventory Value" dell'inventario dei prodotti.

DAX

```
=  
CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*Produ  
ctInventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEYEAR](#)

[Funzione CLOSINGBALANCEMONTH](#)

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# CLOSINGBALANCEYEAR

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza dell'ultima data dell'anno nel contesto corrente.

## Sintassi

DAX

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

## Parametri

Termine	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza dell'ultima data dell'anno nel contesto corrente.

## Osservazioni:

- Il parametro **year\_end\_date** è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.

- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
- Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

① Nota

I vincoli per le espressioni booleane sono descritti in [Funzione CALCULATE](#).

① Nota

Per l'espressione **filter** esistono le restrizioni descritte in [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore "Year End Inventory Value" dell'inventario dei prodotti.

DAX

```
=  
CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI  
nventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEYEAR](#)

[Funzione CLOSINGBALANCEQUARTER](#)

[Funzione CLOSINGBALANCEMONTH](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# DATEADD

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di date, spostate in avanti o indietro nel tempo in base al numero specificato di intervalli rispetto alle date nel contesto corrente.

## Sintassi

DAX

```
DATEADD(<dates>,<number_of_intervals>,<interval>)
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.
number_of_intervals	Numero intero che specifica il numero di intervalli da aggiungere o sottrarre nelle date.
interval	Intervallo in base al quale spostare le date. I possibili valori per l'intervallo sono i seguenti: <code>year</code> , <code>quarter</code> , <code>month</code> , <code>day</code>

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:

- Un riferimento a una colonna di data/ora.
- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
- Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

### ⚠ Nota

I vincoli per le espressioni booleane sono descritti nell'argomento **Funzione CALCULATE**.

- Se il numero specificato per **number\_of\_intervals** è positivo, le date in **dates** vengono spostate avanti nel tempo. Se il numero è negativo, le date in **dates** vengono spostate indietro nel tempo.
- Il parametro **interval** è un'enumerazione, non un set di stringhe, pertanto i valori non devono essere racchiusi tra virgolette. Inoltre, quando vengono usati, i valori **year**, **quarter**, **month**, **day** devono essere digitati per esteso.
- La tabella dei risultati include solo le date presenti nella colonna **dates**.
- Se le date nel contesto corrente non formano un intervallo contiguo, la funzione restituisce un errore.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio: spostamento di un set di date

La formula seguente calcola le date che precedono di un anno le date nel contesto corrente.

DAX

```
= DATEADD(DateTime[DateKey], -1, year)
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATESBETWEEN

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di date che inizia con una data di inizio specificata e continua fino a una data di fine specificata.

Questa funzione è adatta per il passaggio come filtro alla funzione [CALCULATE](#). Usarla per filtrare un'espressione in base a un intervallo di date personalizzato.

## ① Nota

Se si usano intervalli di date standard, ad esempio giorni, mesi, trimestri o anni, è consigliabile usare la funzione [DATESINPERIOD](#), che è più appropriata.

## Sintassi

DAX

```
DATESBETWEEN(<Dates>, <StartDate>, <EndDate>)
```

## Parametri

Termine	Definizione
Date	Colonna di data.
StartDate	Espressione data.
EndDate	Espressione data.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Nel caso d'uso più comune, **Dates** è un riferimento alla colonna di data di una tabella di date contrassegnata.

- Se **StartDate** è BLANK, **StartDate** corrisponderà al valore meno recente nella colonna **Dates**.
- Se **EndDate** è BLANK, **EndDate** corrisponderà al valore più recente nella colonna **Dates**.
- Le date utilizzate come **StartDate** ed **EndDate** sono incluse. Se, ad esempio, il valore **StartDate** è 1° luglio 2019, tale data sarà inclusa nella tabella restituita (a condizione che sia presente nella colonna **Dates**).
- La tabella restituita può contenere solo date archiviate nella colonna **Dates**. Ad esempio, se la colonna **Dates** inizia dal 1° luglio 2017 e il valore **StartDate** è 1° luglio 2016, la tabella restituita inizierà dal 1° luglio 2017.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La definizione di misura della tabella **Sales** seguente usa la funzione DATESBETWEEN per produrre un calcolo *dall'inizio* (LTD, Life-To-Date). Il valore calcolato dall'inizio rappresenta l'accumulo di una misura nel tempo dall'inizio del tempo.

Si noti che la formula usa la funzione **MAX**. Questa funzione restituisce la data più recente inclusa nel contesto del filtro. La funzione DATESBETWEEN restituisce quindi una tabella di date a partire dalla data meno recente fino alla più recente.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

### DAX

```
Customers_LTD =
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    )
)
```

Tenere presente che la data meno recente archiviata nella tabella **Date** è il 1° luglio 2017. Pertanto, quando un report filtra la misura in base al mese di giugno 2020, la

funzione DATESBETWEEN restituisce un intervallo di date compreso tra il 1° luglio 2017 e il 30 giugno 2020.

## Vedi anche

- [Funzioni di Business Intelligence per le gerarchie temporali \(DAX\)](#)
  - [Funzioni di data e ora \(DAX\)](#)
  - [Funzione DATESINPERIOD \(DAX\)](#)
- 

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATESINPERIOD

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di date che inizia con una data di inizio specificata e continua in base al numero e al tipo di intervalli di data specificati.

Questa funzione è adatta per il passaggio come filtro alla funzione [CALCULATE](#). Usarla per filtrare un'espressione in base a intervalli di date standard, ad esempio giorni, mesi, trimestri o anni.

## Sintassi

DAX

`DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)`

## Parametri

Termine	Definizione
date	Colonna di data.
start_date	Espressione data.
number_of_intervals	Numero intero che specifica il numero di intervalli da aggiungere o sottrarre nelle date.
interval	Intervallo in base al quale spostare le date. I possibili valori per l'intervallo sono i seguenti: <code>DAY</code> , <code>MONTH</code> , <code>QUARTER</code> e <code>YEAR</code>

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Nel caso d'uso più comune, **dates** è un riferimento alla colonna di data di una tabella di date contrassegnata.
- Se il numero specificato per **number\_of\_intervals** è positivo, le date vengono spostate avanti nel tempo. Se il numero è negativo, le date vengono spostate

indietro nel tempo.

- Il parametro **interval** è un'enumerazione. I valori validi sono **DAY**, **MONTH**, **QUARTER** e **YEAR**. Trattandosi di un'enumerazione, i valori non vengono passati come stringhe. Evitare quindi di racchiuderli tra virgolette.
- La tabella restituita può contenere solo date archiviate nella colonna **dates**. Ad esempio, se la colonna **dates** inizia dal 1° luglio 2017 e il valore **start\_date** è 1 luglio 2016, la tabella restituita inizierà dal 1° luglio 2017.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La definizione della misura di tabella **Sales** seguente usa la funzione **DATESINPERIOD** per calcolare i ricavi per l'anno precedente (PY).

Si noti che la formula usa la funzione **MAX**. Questa funzione restituisce la data più recente inclusa nel contesto del filtro. La funzione **DATESINPERIOD** restituisce quindi una tabella di date a partire dalla data più recente per l'anno precedente.

*Gli esempi in questo articolo possono essere usati con il modello campione di Adventure Works DW 2020 di Power BI Desktop. Per ottenere il modello, vedere [modello campione DAX](#).*

DAX

```
Revenue PY =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    DATESINPERIOD(  
        'Date'[Date],  
        MAX('Date'[Date]),  
        -1,  
        YEAR  
    )  
)
```

Si supponga che il report sia filtrato in base al mese di giugno 2020. La funzione MAX restituisce il 30 giugno 2020. La funzione **DATESINPERIOD** restituisce quindi l'intervallo di date dal 1° luglio 2019 al 30 giugno 2020. Si tratta di un anno di valori di data a partire dal 30 giugno 2020 per l'anno precedente.

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali \(DAX\)](#)

[Funzioni di data e ora \(DAX\)](#)

[Funzione DATESBETWEEN \(DAX\)](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATESMTD

Articolo • 20/10/2023

Restituisce una tabella contenente una colonna delle date da inizio mese, nel contesto corrente.

## Sintassi

DAX

**DATESMTD(<dates>)**

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:

- Riferimento a una colonna di data/ora.
- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
- Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

### ⓘ Nota

I vincoli per le espressioni booleane sono descritti nell'argomento **Funzione CALCULATE**.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore di Month To Date Total per Internet Sales.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESMTD(DateTime[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione DATESYTD](#)

[Funzione DATESQTD](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATESQTD

Articolo • 20/10/2023

Restituisce una tabella contenente una colonna delle date da inizio trimestre, nel contesto corrente.

## Sintassi

DAX

`DATESQTD(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:

- Riferimento a una colonna di data/ora.
- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
- Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

### ⚠ Nota

I vincoli per le espressioni booleane sono descritti nell'argomento **Funzione CALCULATE**.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore di Quarterly Running Total per Internet Sales.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESQTD(DateTime[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione DATESYTD](#)

[Funzione DATESMTD](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# DATESYTD

Articolo • 20/10/2023

Restituisce una tabella contenente una colonna delle date da inizio anno, nel contesto corrente.

## Sintassi

DAX

```
DATESYTD(<dates> [,<year_end_date>])
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:

- Un riferimento a una colonna di data/ora.
- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
- Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

(!) Nota

I vincoli per le espressioni booleane sono descritti nell'argomento **Funzione CALCULATE**.

- Il parametro **year\_end\_date** è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata. A seconda delle impostazioni locali, il formato potrebbe essere simile a "m-dd" o "dd-m".
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore di Running Total per Internet Sales.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESYTD(DateTime[DateKey]))
```

La formula di esempio seguente crea una misura che calcola il valore "Fiscal Year Running Total" per le vendite Internet, usando le impostazioni locali degli Stati Uniti per il formato Data.

DAX

```
= CALCULATE(  
    SUM(InternetSales_USD[SalesAmount_USD]),  
    DATESYTD(DateTime[DateKey],  
        "6-30"  
    )  
)
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione DATESMTD](#)

[Funzione DATESQTD](#)

# Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ENDOFMONTH

Articolo • 20/10/2023

Restituisce l'ultima data del mese nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

`ENDOFMONTH(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula di esempio seguente crea una misura che restituisce la fine del mese per il contesto corrente.

DAX

```
= ENDOFMONTH(DateTime[DateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione ENDOFYEAR](#)

[Funzione ENDOFQUARTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ENDOFQUARTER

Articolo • 20/10/2023

Restituisce l'ultima data del trimestre nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

`ENDOFQUARTER(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula di esempio seguente crea una misura che restituisce la fine del trimestre per il contesto corrente.

DAX

```
= ENDOFQUARTER(DateTime[DateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione ENDOFYEAR](#)

[Funzione ENDOFMONTH](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# ENDOFYEAR

Articolo • 20/10/2023

Restituisce l'ultima data dell'anno nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

```
ENDOFYEAR(<dates> [,<year_end_date>])
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- Il parametro `year_end_date` è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che restituisce la fine dell'anno fiscale che termina il 30 giugno, per il contesto corrente.

DAX

```
= ENDOFYEAR(DateTime[DateKey], "06/30/2004")
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione ENDOFMONTTH](#)

[Funzione ENDOFQUARTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FIRSTDATE

Articolo • 20/10/2023

Restituisce la prima data nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

```
FIRSTDATE(<dates>)
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Quando il contesto corrente è una singola data, la data restituita dalle funzioni FIRSTDATE e LASTDATE sarà uguale.
- Il valore restituito è una tabella contenente una singola colonna e un singolo valore. Questa funzione può pertanto essere usata come argomento di qualsiasi

funzione nei cui argomenti sia richiesta una tabella. Il valore restituito può inoltre essere usato quando è richiesto un valore di data.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che ottiene la prima data in cui è stata effettuata una vendita tramite il canale di vendita Internet per il contesto corrente.

DAX

```
= FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione LASTDATE](#)

[Funzione FIRSTNONBLANK](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FIRSTNONBLANK

Articolo • 20/10/2023

Restituisce l'ultimo valore nella colonna **column**, filtrata in base al contesto corrente in cui l'espressione non è vuota.

## Sintassi

DAX

```
FIRSTNONBLANK(<column>,<expression>)
```

## Parametri

Termine	Definizione
colonna	Espressione di colonna.
expression	Espressione valutata per i valori vuoti di ogni valore di <b>column</b> .

## Valore restituito

Tabella contenente una singola colonna e una singola riga con il primo valore calcolato.

## Osservazioni:

- L'argomento **column** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a qualsiasi colonna.
  - Una tabella con una sola colonna.
- Un'espressione booleana che definisce una tabella con una sola colonna.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Questa funzione viene in genere usata per restituire il primo valore di una colonna per cui l'espressione non è vuota. Ad esempio, è possibile ottenere l'ultimo valore per cui sono state effettuate vendite di un prodotto.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[Funzione LASTNONBLANK](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# FIRSTNONBLANKVALUE

Articolo • 20/10/2023

Valuta un'espressione filtrata in base ai valori ordinati di una colonna e restituisce il primo valore dell'espressione che non è vuoto.

## Sintassi

DAX

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

## Parametri

Termine	Definizione
colonna	Colonna o espressione che restituisce una tabella a colonna singola.
expression	Un'espressione valutata per ogni valore di <colonna>.

## Valore restituito

Il primo valore non vuoto di <expression> corrispondente ai valori ordinati di <column>.

## Osservazioni:

- L'argomento column può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a qualsiasi colonna.
  - Una tabella con una sola colonna.
- Questa funzione è diversa da FIRSTNONBLANK in quanto il parametro <column> viene aggiunto al contesto di filtro per la valutazione di <expression>.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
SUMMARIZECOLUMNS(  
    DimProduct[Class],  
    "FNBV",  
    FIRSTNONBLANKVALUE(  
        DimDate[Date],  
        SUM(FactInternetSales[SalesAmount])  
    )  
)
```

Restituisce:

DimProduct[Class]	[FNBV]
L	699,0982
H	13778,24
M	1000,4375
	533,83

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# LASTDATE

Articolo • 20/10/2023

Restituisce l'ultima data nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

```
LASTDATE(<dates>)
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Quando il contesto corrente è una singola data, la data restituita dalle funzioni FIRSTDATE e LASTDATE sarà uguale.
- Tecnicamente, il valore restituito è una tabella contenente una singola colonna e un singolo valore. Questa funzione può pertanto essere usata come argomento di

qualsiasi funzione nei cui argomenti sia richiesta una tabella. Il valore restituito può inoltre essere usato quando è richiesto un valore di data.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che ottiene l'ultima data, per il contesto corrente, in cui è stata effettuata una vendita tramite il canale di vendita Internet.

DAX

```
= LASTDATE('InternetSales_USD'[SaleDateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione FIRSTDATE](#)

[Funzione LASTNONBLANK](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# LASTNONBLANK

Articolo • 20/10/2023

Restituisce l'ultimo valore nella colonna **column**, filtrata in base al contesto corrente, in cui l'espressione non è vuota.

## Sintassi

DAX

```
LASTNONBLANK(<column>,<expression>)
```

## Parametri

Termine	Definizione
colonna	Espressione di colonna.
expression	Espressione valutata per i valori vuoti di ogni valore di <b>column</b> .

## Valore restituito

Tabella contenente una singola colonna e una singola riga con l'ultimo valore calcolato.

## Osservazioni:

- L'argomento **column** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a qualsiasi colonna.
  - Una tabella con una sola colonna.
  - Un'espressione booleana che definisce una tabella con una sola colonna.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Questa funzione viene in genere usata per restituire l'ultimo valore di una colonna per cui l'espressione non è vuota. Ad esempio, è possibile ottenere l'ultimo valore per cui sono state effettuate vendite di un prodotto.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Vedi anche

[Funzione FIRSTNONBLANK](#)

[Funzioni statistiche](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# LASTNONBLANKVALUE

Articolo • 20/10/2023

Valuta un'espressione filtrata in base ai valori ordinati di una colonna e restituisce l'ultimo valore dell'espressione che non è vuoto.

## Sintassi

DAX

```
LASTNONBLANKVALUE(<column>, <expression>)
```

## Parametri

Termine	Definizione
colonna	Colonna o espressione che restituisce una tabella a colonna singola.
expression	Un'espressione valutata per ogni valore di <colonna>.

## Valore restituito

L'ultimo valore non vuoto di <espressione> corrispondente ai valori ordinati di <colonna>.

## Osservazioni:

- L'argomento column può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a qualsiasi colonna.
  - Una tabella con una sola colonna.
- Questa funzione è diversa da LASTNONBLANK in quanto la <colonna> è aggiunta al contesto di filtro per la valutazione di <espressione>.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La query DAX seguente:

DAX

```
EVALUATE  
SUMMARIZECOLUMNS(  
    DimProduct[Class],  
    "LNBV",  
    LASTNONBLANKVALUE(  
        DimDate[Date],  
        SUM(FactInternetSales[SalesAmount])  
    )  
)
```

Restituisce:

DimProduct[Class]	[LNBV]
L	132,44
H	137,6
M	84,97
	2288,6

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# NEXTDAY

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date dal giorno successivo, in base alla prima data specificata nella colonna **dates** nel contesto corrente.

## Sintassi

DAX

`NEXTDAY(<dates>)`

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date dal giorno successivo fino alla prima data nel parametro di input. Se ad esempio la prima data nell'argomento **dates** si riferisce al 10 giugno 2009, questa funzione restituisce tutte le date uguali all'11 giugno 2009.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del giorno successivo.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTDAY('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione NEXTQUARTER](#)

[Funzione NEXTMONTH](#)

[Funzione NEXTYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NEXTMONTH

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date del mese successivo, in base alla prima data presente nella colonna **dates** nel contesto corrente.

## Sintassi

DAX
<code>NEXTMONTH(&lt;dates&gt;)</code>

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date dal giorno successivo fino alla prima data nel parametro di input. Se ad esempio la prima data nell'argomento **dates** si riferisce al 10 giugno 2009, questa funzione restituisce tutte le date per il mese di luglio 2009.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del mese successivo.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTMONTH('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione NEXTDAY](#)

[Funzione NEXTQUARTER](#)

[Funzione NEXTYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 **Sì**     **No**

# NEXTQUARTER

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date del trimestre successivo, basata sulla prima data specificata nella colonna **dates**, nel contesto corrente.

## Sintassi

DAX
<code>NEXTQUARTER(&lt;dates&gt;)</code>

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date del trimestre successivo, in base alla prima data nel parametro di input. Se ad esempio la prima data nella colonna **dates** si riferisce al 10 giugno 2009, questa funzione restituisce tutte le date per il trimestre da luglio a settembre 2009.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del trimestre successivo.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTQUARTER('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione NEXTDAY](#)

[Funzione NEXTMONTH](#)

[Funzione NEXTYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# NEXTYEAR

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date dell'anno successivo, in base alla prima data presente nella colonna **dates**, nel contesto corrente.

## Sintassi

DAX

```
NEXTYEAR(<dates>[,<year_end_date>])
```

## Parametri

Termine	Definizione
dates	Colonna contenente date.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date dell'anno successivo, in base alla prima data nella colonna di input. Se ad esempio la prima data nella colonna **dates** si riferisce all'anno 2007, questa funzione restituisce tutte le date per l'anno 2008.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- Il parametro **year\_end\_date** è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet dell'anno successivo.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTYEAR('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione NEXTDAY](#)

[Funzione NEXTQUARTER](#)

[Funzione NEXTMONTH](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# OPENINGBALANCEMONTH

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza della prima data del mese nel contesto corrente.

## Sintassi

DAX

```
OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

## Parametri

Termine	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza della prima data del mese nel contesto corrente.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- L'espressione **filter** ha le restrizioni descritte nell'argomento [Funzione CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore "Month Start Inventory Value" dell'inventario dei prodotti.

```
DAX  
=  
OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product  
Inventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzione OPENINGBALANCEYEAR](#)

[Funzione OPENINGBALANCEQUARTER](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEMONTH](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# OPENINGBALANCEQUARTER

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza della prima data del trimestre, nel contesto corrente.

## Sintassi

DAX

```
OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

## Parametri

Termine	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter*	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza della prima data del trimestre nel contesto corrente.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- L'espressione **filter** ha le restrizioni descritte nell'argomento [Funzione CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola "Quarter Start Inventory Value" dell'inventario dei prodotti.

```
DAX  
=  
OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzione OPENINGBALANCEYEAR](#)

[Funzione OPENINGBALANCEMONTH](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEQUARTER](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# OPENINGBALANCEYEAR

Articolo • 20/10/2023

Valuta il valore di **expression** in corrispondenza della prima data dell'anno nel contesto corrente.

## Sintassi

DAX

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

## Parametri

Termine	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Valore scalare che rappresenta il valore di **expression** valutato in corrispondenza della prima data dell'anno nel contesto corrente.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
- L'espressione **filter** ha le restrizioni descritte nell'argomento [Funzione CALCULATE](#).
- Il parametro **year\_end\_date** è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il valore "Year Start Inventory Value" dell'inventario dei prodotti.

DAX

```
=  
OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI  
nventory[UnitsBalance]),DateTime[DateKey])
```

## Vedi anche

[Funzione OPENINGBALANCEQUARTER](#)

[Funzione OPENINGBALANCEMONTH](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzione CLOSINGBALANCEYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì	 No
--	--

# PARALLELPERIOD

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di date che rappresenta un periodo parallelo alle date nella colonna **dates** specificata nel contesto corrente, con le date spostate di un numero di intervalli in avanti o indietro nel tempo.

## Sintassi

DAX

```
PARALLELPERIOD(<dates>, <number_of_intervals>, <interval>)
```

## Parametri

Termine	Definizione
date	Colonna che contiene date.
number_of_intervals	Numero intero che specifica il numero di intervalli da aggiungere o sottrarre nelle date.
interval	Intervallo in base al quale spostare le date. I possibili valori per l'intervallo sono i seguenti: <code>year</code> , <code>quarter</code> , <code>month</code> .

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione accetta il set di date corrente della colonna specificata in base a **dates**, sposta la prima e l'ultima data in base al numero di intervalli specificato e quindi restituisce tutte le date contigue comprese tra le due date spostate. Se l'intervallo è una parte di mese, trimestre o anno, nel risultato verranno riempiti anche tutti i mesi parziali per completare l'intero intervallo.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a una colonna di data/ora.

- Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).
  - Se il numero specificato per **number\_of\_intervals** è positivo, le date in **dates** vengono spostate avanti nel tempo. Se il numero è negativo, le date in **dates** vengono spostate indietro nel tempo.
  - Il parametro **interval** è un'enumerazione, non un set di stringhe, pertanto i valori non devono essere racchiusi tra virgolette. Inoltre, quando vengono usati, i valori **year**, **quarter**, **month** devono essere digitati per esteso.
  - La tabella dei risultati include solo le date visualizzate nei valori della colonna della tabella sottostante.
  - La funzione PARALLELPERIOD è simile alla funzione DATEADD, con la differenza che PARALLELPERIOD restituisce sempre periodi completi al livello di granularità specificato, mentre DATEADD restituisce periodi parziali. Se ad esempio si ha una selezione di date con inizio il 10 giugno e fine il 21 giugno dello stesso anno e si vuole spostare tale selezione in avanti di un mese, la funzione PARALLELPERIOD restituirà tutte le date del mese successivo, dall'1 fino al 31 luglio. Se invece si usa la funzione DATEADD, il risultato includerà solo le date dal 10 al 21 luglio.
  - Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet dell'anno precedente.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PARALLELPERIOD(DateTime[DateKey], -1, year))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione DATEADD](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PREVIOUSDAY

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date che rappresentano il giorno precedente alla prima data presente nella colonna **dates**, nel contesto corrente.

## Sintassi

DAX
<code>PREVIOUSDAY(&lt;dates&gt;)</code>

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione determina la prima data nel parametro di input, quindi restituisce tutte le date corrispondenti al giorno che precede tale prima data. Se ad esempio la prima data nell'argomento **dates** si riferisce al 10 giugno 2009, questa funzione restituisce tutte le date uguali al 9 giugno 2009.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [Funzione CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del giorno precedente.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSDAY('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[Funzione PREVIOUSMONTH](#)

[Funzione PREVIOUSQUARTER](#)

[Funzione PREVIOUSYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PREVIOUSMONTH

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date del mese precedente, in base alla prima data presente nella colonna **Dates** nel contesto corrente.

## Sintassi

DAX
<code>PREVIOUSMONTH(&lt;Dates&gt;)</code>

## Parametri

Termine	Definizione
Date	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date del mese precedente, usando la prima data nella colonna usata come input. Se ad esempio la prima data nell'argomento **Dates** si riferisce al 10 giugno 2009, questa funzione restituisce tutte le date per il mese di maggio 2009.
- L'argomento **Dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del mese precedente.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSMONTH('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

[PREVIOUSYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PREVIOUSQUARTER

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date del trimestre precedente, in base alla prima data presente nella colonna **dates**, nel contesto corrente.

## Sintassi

DAX

`PREVIOUSQUARTER(<dates>)`

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date del trimestre precedente, usando la prima data nella colonna di input. Ad esempio, se la prima data dell'argomento **date** fa riferimento al 10 giugno 2009, questa funzione restituisce tutte le date per il trimestre da gennaio a marzo 2009.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet del trimestre precedente.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSQUARTER('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

[PREVIOUSYEAR](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# PREVIOUSYEAR

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di tutte le date dell'anno precedente, in base all'ultima data presente nella colonna **dates**, nel contesto corrente.

## Sintassi

DAX

```
PREVIOUSYEAR(<dates>[,<year_end_date>])
```

## Parametri

Termine	Definizione
dates	Colonna contenente date.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Tabella contenente una singola colonna di valori di data.

## Osservazioni:

- Questa funzione restituisce tutte le date dell'anno precedente, in base all'ultima data nel parametro di input. Se ad esempio l'ultima data nell'argomento **dates** si riferisce all'anno 2009, questa funzione restituisce tutte le date per l'anno 2008, fino al valore di **year\_end\_date** specificato.
- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.

- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- Il parametro `year_end_date` è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno viene ignorata.
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola le vendite Internet dell'anno precedente.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# SAMEPERIODLASTYEAR

Articolo • 20/10/2023

Restituisce una tabella che contiene una colonna di date spostate indietro di un anno rispetto alle date nella colonna **dates** specificata, nel contesto corrente.

## Sintassi

DAX

`SAMEPERIODLASTYEAR(<dates>)`

## Parametri

Termine	Definizione
dates	Colonna contenente date.

## Valore restituito

Tabella a una sola colonna di valori di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Un riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- Le date restituite sono le stesse restituite da questa formula equivalente:  
`DATEADD(dates, -1, year)`
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula di esempio seguente crea una misura che calcola le vendite dei rivenditori dell'anno precedente.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),  
SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

## Vedi anche

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[Funzioni di data e ora](#)

[PREVIOUSYEAR](#)

[PARALLELPERIOD](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# STARTOFTMONTH

Articolo • 20/10/2023

Restituisce la prima data del mese nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

`STARTOFTMONTH(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che restituisce l'inizio del mese per il contesto corrente.

DAX

```
= STARTOFMONTH(DateTime[DateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[STARTOFTYEAR](#)

[STARTOFOQUARTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# STARTOFQUARTER

Articolo • 20/10/2023

Restituisce la prima data del trimestre nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

`STARTOFQUARTER(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che restituisce l'inizio del trimestre per il contesto corrente.

DAX

```
= STARTOFCQUARTER(DateTime[DateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[STARTOFCYEAR](#)

[STARTOFCMONTH](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 [Sì](#)    [No](#)

# STARTOFTYEAR

Articolo • 20/10/2023

Restituisce la prima data dell'anno nel contesto corrente per la colonna di date specificata.

## Sintassi

DAX

`STARTOFTYEAR(<dates>)`

## Parametri

Termine	Definizione
date	Colonna che contiene date.
YearEndDate	(facoltativo) Valore di data di fine anno.

## Valore restituito

Tabella contenente una singola colonna e una singola riga con un valore di data.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

# Esempio

La formula di esempio seguente crea una misura che restituisce l'inizio dell'anno per il contesto corrente.

DAX

```
= STARTOFTIME(DateTime[DateKey])
```

## Vedi anche

[Funzioni di data e ora](#)

[Funzioni di Business Intelligence per le gerarchie temporali](#)

[STARTOFTQUARTER](#)

[STARTOFTMONTH](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# TOTALMTD

Articolo • 20/10/2023

Valuta il valore di **expression** da inizio mese, nel contesto corrente.

## Sintassi

DAX

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

## Parametri

Parametro	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta l'**espressione** valutata per le date dall'inizio del mese corrente, specificate le date in **dates**.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- L'espressione **filter** prevede restrizioni descritte nell'argomento [CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il "totale parziale del mese" o la "somma parziale del mese" delle vendite Internet.

DAX

```
= TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## Vedi anche

[ALL](#)

[CALCULATE](#)

[TOTALYTD](#)

[TOTALQTD](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?



# TOTALQTD

Articolo • 20/10/2023

Valuta il valore di **expression** per le date da inizio trimestre, nel contesto corrente.

## Sintassi

DAX

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

## Parametri

Parametro	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.

## Valore restituito

Valore scalare che rappresenta l'**espressione** valutata per tutte le date dall'inizio del trimestre corrente, specificate le date in **dates**.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).
- L'espressione **filter** prevede restrizioni descritte nell'argomento [CALCULATE](#).

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il "totale parziale del trimestre" o la "somma totale del trimestre" delle vendite Internet.

DAX

```
= TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## Vedi anche

[ALL](#)

[CALCULATE](#)

[TOTALYTD](#)

[TOTALMTD](#)

---

## Commenti e suggerimenti

Questo articolo è stato utile?

 Sì  No

# TOTALYTD

Articolo • 20/10/2023

Valuta il valore di **expression** da inizio anno nel contesto corrente.

## Sintassi

DAX

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

## Parametri

Parametro	Definizione
expression	Espressione che restituisce un valore scalare.
date	Colonna che contiene date.
filter	(facoltativo) Espressione che specifica un filtro da applicare al contesto corrente.
year_end_date	(Facoltativo) Stringa letterale con una data che definisce la data di fine anno. Il valore predefinito è 31 dicembre.

## Valore restituito

Valore scalare che rappresenta l'elemento **expression** valutato per l'elemento **dates** corrente da inizio anno.

## Osservazioni:

- L'argomento **dates** può essere rappresentato da uno qualsiasi degli elementi seguenti:
  - Riferimento a una colonna di data/ora.
  - Un'espressione di tabella che restituisce una singola colonna di valori di data/ora.
  - Un'espressione booleana che definisce una tabella con una sola colonna di valori di data/ora.
- I vincoli per le espressioni booleane sono descritti nell'argomento [CALCULATE](#).

- L'espressione **filter** prevede restrizioni descritte nell'argomento [CALCULATE](#).
- Il parametro **year\_end\_date** è un valore letterale di stringa per una data, nelle stesse impostazioni locali del client in cui è stata creata la cartella di lavoro. La parte della data relativa all'anno non è obbligatoria e viene ignorata. La formula seguente, ad esempio, specifica l'elemento **year\_end\_date** (fiscale) 6/30 in una cartella di lavoro con le impostazioni locali EN-US.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey],
ALL('DateTime'), "6/30")
```

In questo esempio **year\_end\_date** può essere specificato come "6/30", "Jun 30", "30 June" o qualsiasi stringa che restituisce un mese/giorno. È tuttavia consigliabile specificare **year\_end\_date** usando il formato "mese/giorno" (come illustrato qui) per assicurarsi che la stringa restituisca una data.

- Questa funzione non è supportata per l'uso nella modalità DirectQuery se usata in colonne calcolate o nelle regole di sicurezza a livello di riga.

## Esempio

La formula di esempio seguente crea una misura che calcola il "totale parziale dell'anno" o la "somma parziale dell'anno" delle vendite Internet.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## Vedi anche

[ALL](#)  
[CALCULATE](#)  
[DATESYTD](#)  
[TOTALMTD](#)  
[TOTALQTD](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Rendiconti

Articolo • 20/10/2023

## In questa categoria

Istruzione	Descrizione
DEFINE	(Parola chiave) Introduce una o più definizioni di entità che possono essere applicate a una o più istruzioni EVALUATE.
EVALUATE	(Parola chiave) Introduce un'istruzione contenente un'espressione di tabella necessaria per eseguire una query DAX.
MEASURE	(Parola chiave) Introduce una definizione di misura che può essere usata in una o più istruzioni EVALUATE in una query.
ORDER BY	(Parola chiave) Introduce un'istruzione che definisce l'ordinamento dei risultati delle query restituiti da un'istruzione EVALUATE.
START AT	(Parola chiave) Introduce un'istruzione che definisce il valore iniziale in corrispondenza del quale vengono restituiti i risultati della query di un'istruzione ORDER BY.
VAR	(Parola chiave) Archivia il risultato di un'espressione come variabile denominata, che può quindi essere passata come argomento ad altre espressioni di misura.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì  No

# DEFINE

Articolo • 20/10/2023

Introduce un'istruzione con una o più definizioni di entità che possono essere applicate a una o più istruzioni EVALUATE di una [query DAX](#).

## Sintassi

```
DAX

[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
) +
]

(EVALUATE <table expression>) +
```

## Parametri

Termine	Definizione
Entity	MISURA, VAR, TABELLA <sup>1</sup> o COLONNA <sup>1</sup> .
name	Nome di una definizione di misura, var, tabella o colonna. Non può essere un'espressione. Il nome non deve essere univoco. Il nome esiste solo per la durata della query.
expression	Qualsiasi espressione DAX che restituisce una tabella o un valore scalare. L'espressione può usare una qualsiasi delle entità definite. Se è necessario convertire un'espressione scalare in un'espressione di tabella, eseguire il wrapping dell'espressione all'interno di un costruttore di tabella con parentesi graffe {} oppure usare la funzione ROW() per restituire una singola tabella di righe.

[1]**Attenzione:** le definizioni TABLE e COLUMN con ambito query sono destinate solo all'uso interno. Sebbene sia possibile definire espressioni TABLE e COLUMN per una query senza errori di sintassi, possono generare errori di runtime e non sono consigliate.

## Osservazioni:

- Una query DAX può avere più istruzioni EVALUATE, ma può avere una sola istruzione DEFINE. Le definizioni nell'istruzione DEFINE possono essere applicate a qualsiasi istruzione EVALUATE nella query.
- In un'istruzione DEFINE è necessaria almeno una definizione.
- Le definizioni di misura per un modello di override della query hanno lo stesso nome.
- I nomi VAR hanno restrizioni univoche. Per altre informazioni, vedere [VAR - Parametri](#).
- Per altre informazioni sull'uso di un'istruzione DEFINE, vedere [query DAX](#).

## Vedi anche

[EVALUATE](#)

[VAR](#)

[MEASURE](#)

[Query DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# EVALUATE

Articolo • 20/10/2023

Mostra un'istruzione contenente un'espressione di tabella richiesta in una [query DAX](#).

## Sintassi

DAX

```
EVALUATE <table>
```

## Parametri

Termine	Definizione
table	Espressione di tabella

## Valore restituito

Il risultato di un'espressione di tabella.

## Osservazioni:

- Una query DAX può contenere più istruzioni EVALUATE.
- Per ulteriori informazioni sul come si usano le istruzioni EVALUATE, vedere [query DAX](#).

## Esempio

DAX

```
EVALUATE  
    'Internet Sales'
```

Restituire tutte le righe e le colonne dalla tabella Internet Sales, sotto forma di tabella.

## Vedi anche

[ORDER BY](#)

[START AT](#)

[DEFINE](#)

[VAR](#)

[Query DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# MEASURE

Articolo • 20/10/2023

Introduce una definizione di misura in un'istruzione DEFINE di una [query DAX](#).

## Sintassi

DAX

```
[DEFINE
(
    MEASURE <table name>[<measure name>] = <scalar expression>
) +
]

(EVALUATE <table expression>) +
```

## Parametri

Termine	Definizione
Nome tabella	Nome di una tabella contenente la misura.
nome misura	Nome della misura. Non può essere un'espressione. Il nome non deve essere univoco. Il nome esiste solo per la durata della query.
scalar_expression	Un'espressione DAX che restituisce un valore scalare.

## Valore restituito

Risultato calcolato dell'espressione di misura.

## Osservazioni:

- Le definizioni di misura per un modello di override della query hanno lo stesso nome per la durata della query. Non influiranno sulla misura del modello.
- L'espressione di misura può essere usata con qualsiasi altra espressione nella stessa query.
- Per ulteriori informazioni sull'uso delle istruzioni MEASURE, vedere le [query DAX](#).

## Vedi anche

[DEFINE](#)

[EVALUATE](#)

[VAR](#)

[Query DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# ORDER BY

Articolo • 20/10/2023

Introduce un'istruzione che definisce l'ordinamento dei risultati delle query restituiti da un'istruzione EVALUATE in una [query DAX](#).

## Sintassi

DAX

```
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]]
```

## Parametri

Termine	Definizione
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare.
ASC	(predefinito) Ordinamento crescente.
DESC	Ordinamento decrescente.

## Valore restituito

Risultato di un'istruzione EVALUATE in ordine crescente (ASC) o decrescente (DESC).

## Osservazioni:

Per ulteriori informazioni sul come si usano le istruzioni ORDER BY, vedere [query DAX](#).

## Vedi anche

[START AT](#)

[EVALUATE](#)

[VAR](#)

[DEFINE](#)

[Query DAX](#)

# Commenti e suggerimenti

Questa pagina è stata utile?



# START AT

Articolo • 20/10/2023

Introduce un'istruzione che definisce il valore iniziale in cui vengono restituiti i risultati della query di una clausola ORDER BY in un'istruzione EVALUATE in una [query DAX](#).

## Sintassi

DAX

```
[START AT {<value>|<parameter>} [, ...]]
```

## Parametri

Termine	Definizione
value	Valore costante di . Non può essere un'espressione.
parameter	Nome di un parametro in un'istruzione XMLA preceduto da un carattere @.

## Osservazioni:

- Gli argomenti START AT presentano una corrispondenza uno-a-uno con le colonne nell'istruzione ORDER BY. Nell'istruzione START AT possono essere presenti lo stesso numero d argomenti dell'istruzione ORDER BY, ma non di più. Il primo argomento nell'istruzione START AT definisce il valore iniziale nella colonna 1 delle colonne ORDER BY. Il secondo argomento nell'istruzione START AT definisce il valore iniziale nella colonna 2 delle colonne ORDER BY all'interno delle righe che soddisfano il primo valore per la colonna 1.
- Per ulteriori informazioni sulle modalità di utilizzo delle istruzioni START AT, vedere le [query DAX](#).

## Vedi anche

[ORDER BY](#)

[EVALUATE](#)

[VAR](#)

DEFINE

Query DAX

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# VAR

Articolo • 20/10/2023

Archivia il risultato di un'espressione come variabile denominata, che può quindi essere passata come argomento ad altre espressioni di misura. Dopo aver calcolato i valori risultanti per un'espressione variabile, non tali valori non cambiano, anche se viene fatto riferimento alla variabile in un'altra espressione.

## Sintassi

DAX

```
VAR <name> = <expression>
```

## Parametri

Termine	Definizione
name	<p>Nome della variabile (identificatore).</p> <p>I delimitatori non sono supportati. "varName" o [varName], ad esempio, genererà un errore.</p> <p>Set di caratteri supportato: a-z, A-Z, 0-9.</p> <p>I caratteri compresi tra 0 e 9 non possono essere usati come primo carattere.</p> <p>_ (carattere di sottolineatura doppio) è consentito come prefisso per il nome dell'identificatore.</p> <p>Non sono supportati altri caratteri speciali.</p> <p>Le parole chiave riservate non sono consentite.</p> <p>I nomi delle tabelle esistenti non sono consentiti.</p> <p>Gli spazi vuoti non sono consentiti.</p>
expression	Espressione DAX che restituisce un valore scalare o di tabella.

## Valore restituito

Variabile denominata contenente il risultato dell'argomento dell'espressione.

## Osservazioni:

- Un'espressione passata come argomento a VAR può contenere un'altra dichiarazione VAR.

- Quando si fa riferimento a una variabile:
  - Le misure non possono fare riferimento a variabili definite all'esterno dell'espressione di misura, ma possono fare riferimento alle variabili di ambito funzionale definite nell'espressione.
  - Le variabili possono fare riferimento alle misure.
  - Le variabili possono fare riferimento a variabili definite in precedenza.
  - Non è possibile fare riferimento alle colonne nelle variabili di tabella tramite la sintassi `TableName[ColumnName]`.
- Per le procedure consigliate quando si usa VAR, vedere [Usare le variabili per migliorare le formule DAX](#).
- Per ulteriori informazioni sull'uso di VAR all'interno di una query DAX, vedere le [query DAX](#).

## Esempio

Per calcolare una percentuale di crescita rispetto all'anno precedente senza usare una variabile, è possibile creare tre misure separate. Questa prima misura calcola la somma dell'importo delle vendite:

DAX

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

Una seconda misura calcola l'importo delle vendite per l'anno precedente:

DAX

```
SalesAmount PreviousYear =
CALCULATE([Sum of SalesAmount],
SAMEPERIODLASTYEAR(Calendar[Date])
)
```

È quindi possibile creare una terza misura che combina le altre due misure per calcolare una percentuale di crescita. Si noti che la misura Sum of SalesAmount viene usata due volte: prima per determinare se è presente una vendita e in seguito per calcolare una percentuale.

DAX

```
Sum of SalesAmount YoY%: =
IF([Sum of SalesAmount] ,
DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of
```

```
SalesAmount])  
)
```

Usando una variabile, è possibile creare una singola misura che calcola lo stesso risultato:

DAX

```
YoY% = VAR Sales = SUM(SalesTable[SalesAmount])  
  
VAR SalesLastYear =  
    CALCULATE ( SUM ( SalesTable[SalesAmount] ), SAMEPERIODLASTYEAR ( 'Calendar'[Date] ) )  
  
    return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

Usando una variabile, è possibile ottenere lo stesso risultato, ma in modo più leggibile. Poiché il risultato dell'espressione viene archiviato nella variabile, le prestazioni della misura possono essere notevolmente migliorate perché non è necessario effettuare il ricalcolo ogni volta che si usa.

## Vedi anche

[Usare le variabili per migliorare le formule DAX](#)

[Query DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Glossario di DAX

Articolo • 20/10/2023

## Query di analisi

Gli oggetti visivi di Power BI eseguono query su un modello di dati usando una *query di analisi*. Una query analitica cerca di ridurre volumi di dati potenzialmente di grandi dimensioni e complessità del modello usando tre fasi distinte: filtrare, raggruppare e riepilogare. Una query di analisi viene creata automaticamente quando vengono assegnati campi alle aree degli oggetti visivi del report. Gli autori del report possono controllare il comportamento delle assegnazioni dei campi rinominando i campi, modificando la tecnica di riepilogo oppure disabilitando il riepilogo per ottenere il raggruppamento. In fase di progettazione del report, è possibile aggiungere filtri al report, a una pagina del report o a un oggetto visivo. Nella visualizzazione di lettura è possibile modificare i filtri nel riquadro **Filtri** o tramite interazioni con filtri dei dati e altri oggetti visivi (filtro incrociato).

## BLANK

DAX definisce l'assenza di un valore come BLANK. Ciò equivale a NULL in SQL, ma il comportamento non è esattamente lo stesso. Il comportamento è più simile a Excel e al modo in cui viene definita una cella vuota. In combinazione con altre operazioni, BLANK viene valutato come zero o come una stringa vuota. Ad esempio,  $\text{BLANK} + 20 = 20$ . Usare sempre lettere maiuscole.

## Colonna calcolata

Calcolo del modello usato per aggiungere una colonna a un modello tabulare scrivendo una formula DAX. La formula deve restituire un valore scalare e viene valutata per ogni riga della tabella. Una colonna calcolata può essere aggiunta a una tabella in modalità di archiviazione di tipo importazione o DirectQuery.

## Misura calcolata

Nella modellazione tabulare non esiste un concetto di *misura calcolata*. Usare invece il concetto di *misura*. La parola *calcolata* viene usata per descrivere le tabelle calcolate e le colonne calcolate. Questi elementi vengono così distinti dalle tabelle e dalle colonne che derivano da Power Query. Power Query non prevede il concetto di misura.

# Tabella calcolata

Calcolo del modello usato per aggiungere una tabella a un modello tabulare scrivendo una formula DAX. La formula deve restituire un oggetto tabella. Il risultato è una tabella che usa la modalità di archiviazione di tipo importazione.

## Calcolo

Processo intenzionale che trasforma uno o più input in uno o più risultati. In un modello di dati tabulare un calcolo può essere un oggetto del modello, ovvero una tabella calcolata, una colonna calcolata o una misura.

## Contesto

Describe l'ambiente in cui viene valutata una formula DAX. Esistono due tipi di contesto: *contesto di riga* e *contesto di filtro*. Il contesto di riga rappresenta la "riga corrente" e viene usato per valutare le formule delle colonne calcolate e le espressioni usate dagli iteratori di tabella. Il contesto di filtro viene usato per valutare le misure e rappresenta i filtri applicati direttamente alle colonne del modello e i filtri propagati dalle relazioni del modello.

## DAX

Il linguaggio DAX (Data Analysis Expressions) è un linguaggio delle formule per Power Pivot in Excel, Power BI, Azure Analysis Services e la modellazione tabulare in SQL Server Analysis Services. È anche possibile usare DAX per aggiungere calcoli del modello di dati e definire regole di sicurezza a livello di riga.

## Sicurezza dinamica

Applicazione delle regole di sicurezza a livello di riga usando l'identità dell'utente del report. Le regole filtrano le tabelle del modello usando il nome dell'account dell'utente, tramite la funzione USERNAME o USERPRINCIPALNAME. Vedere [Sicurezza a livello di riga](#).

## Expression

Unità di logica DAX che viene valutata e restituisce un risultato. Le espressioni possono dichiarare variabili e in questo caso viene assegnata loro una sottoespressione e devono

includere un'istruzione RETURN che restituisce un'espressione finale. Le espressioni vengono create usando oggetti del modello (tabelle, colonne o misure), funzioni, operatori o costanti.

## Campo

Risorsa del modello di dati visualizzata nel riquadro **Campi**. I campi vengono usati per configurare oggetti visivi e filtri dei report. I campi sono costituiti da colonne del modello, livelli di gerarchia e misure.

## Formula

Una o più espressioni DAX usate per definire un calcolo del modello. Le espressioni interne sono dette sottoespressioni. Il plurale è *formule*.

## Funzione

Le funzioni DAX hanno argomenti che consentono il passaggio di parametri. Le formule possono usare molte chiamate di funzione, annidando le funzioni all'interno di altre funzioni. In una formula i nomi di funzione devono essere seguiti da parentesi. All'interno delle parentesi, vengono passati i parametri.

## Misura implicita

Calcolo generato automaticamente ottenuto configurando un oggetto visivo di Power BI per riepilogare i valori di colonna. Le colonne **numeriche** supportano l'intervallo massimo di riepiloghi, tra cui: somma, media, minimo, massimo, conteggio (Distinct), conteggio, deviazione standard, varianza o mediana. È possibile riepilogare anche le colonne di altri tipi di dati. Le colonne di tipo **testo** possono essere riepilogate in base a: Prima (alfabeticamente), Ultima (alfabeticamente), Conteggio (Distinct) o Conteggio. Le colonne di tipo **data** possono essere riepilogate in base a: più vecchio, più recente, conteggio (Distinct) o conteggio. Le colonne **booleans** possono essere riepilogate usando: conteggio (Distinct) o conteggio.

## Funzione di iterazione

Funzione DAX che enumera tutte le righe di una determinata tabella e valuta un'espressione specifica per ogni riga. Fornisce flessibilità e controllo sulle modalità di riepilogo dei dati nei calcoli del modello.

# MDX

Il linguaggio MDX (Multidimensional Expressions) è un linguaggio delle formule per i modelli multidimensionali di SQL Server Analysis Services, noti anche come *cubi*. È possibile usare MDX per eseguire query sui modelli tabulari, ma non per definire misure esplicite. È possibile eseguire query solo sulle misure già definite nel modello.

## Misura

Calcolo che consente di ottenere il riepilogo. Le misure possono essere *esplicite* o *esplicite*. Una misura esplicita è un calcolo aggiunto a un modello di dati tabulare scrivendo una formula DAX. Una formula della misura deve restituire un valore scalare. Nel riquadro **Campi** le misure esplicite sono contrassegnate con un'icona di calcolatrice. Le misure esplicite sono necessarie quando si esegue una query sul modello usando MDX (Multidimensional Expressions), come avviene quando si usa Analizza in Excel. Una misura esplicita viene comunemente chiamata semplicemente misura.

## Gruppo di misure

Tabella del modello che contiene almeno una misura e non ha gerarchie o colonne visibili. Nel riquadro **Campi** ogni gruppo di misure è contrassegnato con un'icona di più calcolatrici. I gruppi di misure sono elencati insieme nella parte superiore del riquadro **Campi** e ordinati alfabeticamente in base al nome.

## Calcolo del modello

Formula denominata usata per aggiungere una tabella calcolata, una colonna calcolata o una misura a un modello di dati tabulare. La struttura è <NAME> = <FORMULA>. La maggior parte dei calcoli viene aggiunta dai responsabili della modellazione dei dati in Power BI Desktop, ma è possibile aggiungere misure anche a un report di connessione dinamica. Vedere [Misure del report](#).

## Misure rapide

Funzionalità di Power BI Desktop che elimina la necessità di scrivere formule DAX per misure comunemente definite. Le misure rapide includono quelle relative a media per categoria, classifica e differenza dalla linea di base.

# Misure del report

Dette anche *misure a livello di report*. Vengono aggiunte a un report di connessione dinamica in Power BI Desktop scrivendo una formula DAX, ma solo per le connessioni ai modelli di Power BI o ai modelli tabulari di Analysis Services.

# Sicurezza a livello di riga

Viene usato anche l'acronimo *RLS* (Row-Level Security). Tecnica di progettazione per limitare l'accesso ai subset di dati per utenti specifici. In un modello tabulare si ottiene creando ruoli del modello. I ruoli prevedono regole, ovvero espressioni DAX per filtrare le righe della tabella.

# Scalare

In DAX, un valore scalare è un singolo valore. Un valore scalare può essere di qualsiasi tipo di dati: Decimale, Intero, Ora/Data, Stringa, Valuta, Booleano. Un valore scalare può essere il risultato di un'espressione calcolata da più valori. Ad esempio, una funzione di aggregazione come MAX() restituisce un singolo valore massimo da un set di valori da cui valutare.

# Riepilogo

Operazione applicata ai valori di una colonna. Vedere [misura](#).

# Business Intelligence per gerarchie temporali

La funzionalità di Business Intelligence per le gerarchie temporali si riferisce ai calcoli nel tempo, ad esempio il valore da inizio anno.

# Funzione di Business Intelligence per le gerarchie temporali

DAX include numerose funzioni di Business Intelligence per le gerarchie temporali. Ogni funzione di Business Intelligence per le gerarchie temporali ottiene il risultato modificando il contesto di filtro per i filtri di data. Funzioni di esempio: TOTALYTD e SAMEPERIODLASTYEAR.

# Valore, valori

Dati da visualizzare.

## Parametro analisi di simulazione

Funzionalità di Power BI Desktop che consente di accettare l'input utente tramite filtri dei dati. Ogni parametro crea una tabella calcolata a colonna singola e una misura che restituisce un singolo valore selezionato. La misura può essere usata nei calcoli del modello per rispondere all'input dell'utente.

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Operatori DAX

Articolo • 20/10/2023

Nel linguaggio DAX (Data Analysis Expressions) vengono usati gli operatori per creare espressioni che confrontano valori, eseguono calcoli aritmetici o utilizzano stringhe.

## Tipi di operatori

Sono disponibili quattro tipi diversi di operatori di calcolo: aritmetico, di confronto, di concatenazione di testo e logico.

### Operatori aritmetici

Per eseguire operazioni matematiche di base, ad esempio l'addizione, la sottrazione o la moltiplicazione, per combinare numeri e per produrre risultati numerici, usare gli operatori aritmetici seguenti.

Operatore aritmetico	Significato	Esempio
+ (segno di addizione)	Addizione	3+3
- (segno di sottrazione)	Sottrazione o segno	3-1-1
* (asterisco)	Moltiplicazione	3*3
/ (barra)	Divisione	3/3
^ (accento circonflesso)	Elevamento a potenza	16^4

#### ⓘ Nota

Il segno più può essere usato sia come *operatore binario* sia come *operatore unario*. Un operatore binario richiede numeri su entrambi i lati ed esegue un'addizione. Quando si usano valori in una formula DAX su entrambi i lati dell'operatore binario, DAX tenta di eseguire il cast dei valori in tipi di dati numerici, se questi non sono già numeri. Al contrario, l'operatore unario può essere applicato a qualsiasi tipo di argomento. Il simbolo più non influisce sul tipo o sul valore e viene semplicemente ignorato, mentre l'operatore di sottrazione crea un valore negativo, se applicato a un valore numerico.

## Operatori di confronto

È possibile confrontare due valori con gli operatori seguenti. Quando vengono messi a confronto due valori tramite questi operatori, il risultato corrisponde a un valore logico, ovvero TRUE o FALSE.

<b>Operatore di confronto</b>	<b>Significato</b>	<b>Esempio</b>
=	Uguale a	[Region] = "USA"
==	Strettamente uguale a	[Region] == "USA"
>	Maggiore di	[Sales Date] > "Jan 2009"
<	Minore di	[Sales Date] < "Jan 1 2009"
>=	Maggiore di o uguale a	[Amount] >= 20000
<=	Minore di o uguale a	[Amount] <= 100
<>	Diverso da	[Region] <> "USA"

Tutti gli operatori di confronto tranne == trattano BLANK come valore 0, stringa vuota "", DATE(1899, 12, 30) o FALSE. Di conseguenza, [Column] = 0 è true quando il valore di [Column] è 0 o BLANK. Al contrario, [Column] == 0 è true solo quando il valore di [Column] è 0.

## Operatore di concatenazione del testo

Usare la e commerciale (&) per unire, o concatenare, due o più stringhe di testo in modo da ottenere un'unica stringa.

<b>Operatore di testo</b>	<b>Significato</b>	<b>Esempio</b>
& (e commerciale)	Connette, o concatena, due valori per generare un valore di testo continuo	[Region] & ", " & [City]

## Operatori logici

Utilizzare gli operatori logici (&&) e (||) per combinare espressioni per produrre un singolo risultato.

<b>Operatore di testo</b>	<b>Significato</b>	<b>Esempi</b>
&& (doppia e commerciale)	Crea una condizione AND tra due espressioni, ciascuna con un risultato booleano. Se entrambe le	([Region] = "France") && ([BikeBuyer] =

<b>Operatore di testo</b>	<b>Significato</b>	<b>Esempi</b>
	espressioni restituiscono TRUE, anche la combinazione delle espressioni restituisce TRUE; in caso contrario, la combinazione restituisce FALSE.	"yes"))
(simbolo della doppia barra verticale)	Crea una condizione OR tra due espressioni logiche. Se una delle due espressioni restituisce TRUE, il risultato è TRUE. Il risultato è FALSE solo quando sono FALSE entrambe le espressioni.	(([Region] = "France")    ([BikeBuyer] = "yes"))
IN	Crea una condizione OR logica tra ogni riga confrontata rispetto a una tabella. Nota: per la sintassi del costruttore della tabella vengono usate le parentesi graffe.	'Product'[Color] IN { "Red", "Blue", "Black" }

## Operatori e ordine di precedenza

In alcuni casi, l'ordine in cui viene eseguito il calcolo può influire sul valore restituito. È quindi importante comprendere come viene determinato l'ordine e come è possibile modificare l'ordine per ottenere i risultati desiderati.

### Ordine di calcolo

Un'espressione valuta gli operatori e i valori in un ordine specifico. Tutte le espressioni iniziano sempre con un segno di uguale (=). Il segno di uguale indica che i caratteri che seguono costituiscono un'espressione.

Dopo il segno di uguale sono riportati gli elementi da calcolare (gli operandi), che sono separati da operatori di calcolo. Le espressioni vengono lette sempre da sinistra a destra, ma l'ordine in cui gli elementi sono raggruppati può essere controllato in una certa misura usando le parentesi.

### Precedenza tra gli operatori

Se si combinano più operatori in una stessa formula, le operazioni vengono ordinate in base alla tabella seguente. Se gli operatori hanno la stessa precedenza, vengono ordinati da sinistra verso destra. Se, ad esempio, un'espressione contiene sia un operatore di moltiplicazione che un operatore di divisione, questi vengono valutati nell'ordine in cui compaiono nell'espressione, da sinistra a destra.

Operatore	Descrizione
$^$	Elevamento a potenza
$-$	Segno (come in $-1$ )
$* \text{ e } /$	Moltiplicazione e divisione
$+ \text{ e } -$	Addizione e sottrazione
$\&$	Connette due stringhe di testo (concatenazione)
$=, ==, <, >, <=, >=, <>, IN$	Confronto
NOT	NOT (operatore unario)

## Uso delle parentesi per controllare l'ordine di calcolo

Per modificare l'ordine di valutazione, è necessario racchiudere tra parentesi la parte della formula che deve essere calcolata per prima. Ad esempio, la formula seguente ha come risultato 11 perché la moltiplicazione viene calcolata prima dell'addizione. La formula moltiplica 2 per 3 e quindi somma 5 al risultato.

DAX

```
=5+2*3
```

Al contrario, se si usano le parentesi per modificare la sintassi, l'ordine di calcolo cambia. Vengono prima sommati 5 e 2 e il risultato dell'addizione viene moltiplicato per 3, restituendo così 21 come risultato finale.

DAX

```
=(5+2)*3
```

Nell'esempio seguente, le parentesi che racchiudono la prima parte della formula richiedono che l'espressione  $(3 + 0.25)$  venga valutata per prima e che il risultato così ottenuto venga diviso per quello dell'espressione  $(3 - 0.25)$ .

DAX

```
=(3 + 0.25)/(3 - 0.25)
```

Nell'esempio seguente viene applicato per primo l'operatore di elevamento a potenza, in base alle regole di precedenza per gli operatori, quindi viene applicato l'operatore di

segno. Il risultato di questa espressione è -4.

```
DAX
```

```
--2^2
```

Per assicurarsi che l'operatore di segno venga applicato per primo al valore numerico, è possibile usare le parentesi per controllare gli operatori, come illustrato nell'esempio seguente. Il risultato di questa espressione è 4.

```
DAX
```

```
= (-2)^2
```

## Compatibilità

DAX è in grado di gestire e confrontare facilmente diversi tipi di dati, in modo analogo a Microsoft Excel. Tuttavia, il motore di calcolo sottostante si basa su SQL Server Analysis Services e offre funzionalità avanzate aggiuntive di un archivio dati relazionale, incluso un supporto più esteso per i tipi relativi a data e ora. In alcuni casi, pertanto, il comportamento delle funzioni o i risultati dei calcoli potrebbero non essere uguali a quelli di Excel. DAX include inoltre il supporto per più tipi di dati rispetto a Excel. In questa sezione vengono descritte le differenze principali.

## Coercizione dei tipi di dati degli operandi

In linea di massima, il tipo di dati dei due operandi a destra e a sinistra di qualsiasi operatore deve essere lo stesso. Se tuttavia i tipi di dati sono diversi, in alcuni casi DAX li convertirà in un tipo comune per applicare l'operatore:

1. Entrambi gli operandi vengono convertiti nel tipo di dati comune più ampio possibile.
2. Viene applicato l'operatore, se possibile.

Si supponga, ad esempio, di avere due numeri da combinare. Un numero deriva da una formula, ad esempio = [Price] \* .20, e il risultato può contenere molte posizioni decimali. L'altro è un numero intero specificato come valore stringa.

In questo caso, DAX convertirà entrambi i numeri in numeri reali in formato numerico, usando il formato numerico più ampio, in grado di archiviare i due tipi di numeri. Dopo la conversione, DAX applicherà la moltiplicazione.

A seconda della combinazione di tipi di dati, è possibile che la coercizione dei tipi non sia applicabile per le operazioni di confronto. Per un elenco completo dei tipi di dati supportati da DAX, vedere [Tipi di dati supportati nei modelli tabulari](#) e [Tipi di dati in Power BI Desktop](#).

Integer, Real Number, Currency, Date/time e Blank sono considerati tipi di dati numerici ai fini delle operazioni di confronto. Quando si esegue un confronto, Blank restituisce zero. Per le operazioni di confronto sono supportate le combinazioni di tipi di dati seguenti.

<b>Tipo di dati per il lato sinistro</b>	<b>Tipo di dati per il lato destro</b>
Numeric	Numeric
Boolean	Boolean
Stringa	Stringa

Altri confronti di tipi di dati misti restituiranno un errore. Ad esempio, una formula come `= "1" > 0` restituisce un errore che indica che le *operazioni di confronto DAX non supportano il confronto di valori di tipo Text con valori di tipo Integer*.

<b>Tipi di dati usati in DAX</b>	<b>Tipi di dati usati in Excel</b>
Numeri (I8, R8)	Numeri (R8)
Stringa	Stringa
Boolean	Boolean
Data/Ora	Variante
Valuta	Valuta

## Differenze nell'ordine di precedenza

L'ordine di precedenza delle operazioni nelle formule DAX è fondamentalmente uguale a quello usato da Microsoft Excel. Tuttavia, alcuni operatori di Excel, come quello di percentuale, non sono supportati. Inoltre, non sono supportati nemmeno gli intervalli.

Di conseguenza, ogni volta che si copiano e si incollano formule da Excel, esaminarle con attenzione, poiché alcuni operatori o elementi delle formule potrebbero non essere validi. In caso di dubbi sull'ordine di esecuzione delle operazioni, è consigliabile usare le parentesi per controllare l'ordine di calcolo ed escludere qualsiasi ambiguità riguardo al risultato.

## Vedi anche

[Sintassi DAX](#)

[Denominazione dei parametri DAX](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?



# Query DAX

Articolo • 20/10/2023

I client di creazione di report come Power BI ed Excel eseguono query DAX ogni volta che un campo viene inserito in un report o quando viene applicato un filtro. Usando [SQL Server Management Studio \(SSMS\)](#), [Generatore report di Power BI](#) e strumenti open source come [DAX Studio](#), è possibile creare ed eseguire query DAX personalizzate. Le query DAX restituiscono risultati come tabella direttamente all'interno dello strumento, consentendo di creare e testare rapidamente le prestazioni delle formule DAX.

Prima di acquisire familiarità con le query, è importante avere una conoscenza approfondita delle nozioni di base di DAX. Se non è già stato fatto, vedere [Panoramica di DAX](#).

## Parole chiave

Le query DAX hanno una sintassi semplice costituita da una sola parola chiave obbligatoria, **EVALUATE** e diverse parole chiave facoltative: **ORDER BY**, **START AT**, **DEFINE**, **MEASURE**, **VAR**, **TABLE** e **COLUMN**. Ogni parola chiave definisce un'istruzione usata per la durata della query.

### EVALUATE (obbligatoria)

Al livello più elementare, una query DAX è un'istruzione **EVALUATE** che contiene un'espressione di tabella. È necessaria almeno un'istruzione EVALUATE, tuttavia, una query può contenere un numero qualsiasi di istruzioni EVALUATE.

#### Sintassi EVALUATE

DAX
EVALUATE <table>

#### Parametri EVALUATE

Termine	Definizione
table	Espressione di tabella.

## Esempio EVALUATE

```
DAX

EVALUATE
    'Internet Sales'
```

Restituire tutte le righe e le colonne dalla tabella Internet Sales, sotto forma di tabella.

The screenshot shows the Microsoft Power BI DAX Query Editor interface. The query window contains the DAX code:

```
EVALUATE
    'Internet Sales'
)
```

The results pane displays a table with the following data:

Internet Sales[ProductKey]	Internet Sales[CustomerKey]	Internet Sales[OrderID]	Internet Sales[OrderQuantity]	Internet Sales[UnitPrice]	Internet Sales[LineTotal]	Internet Sales[SalesAmount]	Internet Sales[CommissionAmount]	Internet Sales[DiscountAmount]	Internet Sales[Freight]
528	25839	1	100	4	S052301	1	1	1	
528	11260	1	100	4	S052314	1	1	1	
528	23695	1	100	4	S052342	1	1	1	
528	15198	1	100	4	S052387	1	1	1	
528	15414	1	100	4	S052499	1	1	1	
528	15469	1	100	4	S052500	1	1	1	
528	15901	1	100	4	S052545	1	1	1	

Below the table, a message bar indicates "Query executed successfully." and shows connection details: "asazure://westus.asazure.w... | adventureworks | 00:00:08".

## ORDER BY (facoltativa)

La parola chiave **ORDER BY** facoltativa definisce una o più espressioni usate per ordinare i risultati della query. Qualsiasi espressione valutabile per ogni riga del risultato è valida.

### Sintassi ORDER BY

```
DAX

EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]]
```

### Parametri ORDER BY

Termine	Definizione
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare.
ASC	(predefinito) Ordinamento crescente.

Termine	Definizione
DESC	Ordinamento decrescente.

## Esempio ORDER BY

```
DAX

EVALUATE
    'Internet Sales'

ORDER BY
    'Internet Sales'[Order Date]
```

Restituisce tutte le righe e le colonne della tabella Internet Sales, in ordine crescente in base alla data ordine, come tabella.

The screenshot shows the Microsoft Power BI Data Editor interface. The top navigation bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main area has tabs for 'Object Explorer' and 'MSDAXQuery1.msd...'. The code editor contains the following DAX query:

```
EVALUATE
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Order Date]
```

Below the code, the 'Results' tab is selected, displaying a table with the following data:

Internet Sales[SalesID]	Internet Sales[TerritoryID]	Internet Sales[Region]	Internet Sales[CountryRegionCode]	Internet Sales[CurrencyCode]	Internet Sales[Order Date]	Internet Sales[Due Date]	Internet Sales[Ship Date]	Internet Sales[Status]
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
78.27	286.2616	89.4568			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1406.97€
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
9.0982	55.9279	17.4775			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	285.951€
99.99	271.9992	84.9998			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1487.83€
74.99	269.9992	84.3748			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1476.89€
78.27	286.2616	89.4568			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1406.97€

At the bottom, a message indicates 'Query executed successfully.' and shows connection details: 'asazure://westus.asazure.wi...', 'adventureworks', and '00:00:08'.

## START AT (facoltativa)

La parola facoltativa **START AT** viene usata all'interno della clausola **ORDER BY**. Definisce il valore in corrispondenza del quale iniziano i risultati della query.

## Sintassi START AT

```
DAX
```

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value>|<parameter>} [, ...]]]
```

## Parametri START AT

Termine	Definizione
value	Valore costante di . Non può essere un'espressione.
parameter	Nome di un parametro in un'istruzione XMLA preceduto da un carattere @.

## Osservazioni START AT

Gli argomenti START AT presentano una corrispondenza uno-a-uno con le colonne nella clausola ORDER BY. Nella clausola START AT può essere presente lo stesso numero di argomenti della clausola ORDER BY, ma non di più. Il primo argomento in START AT definisce il valore iniziale nella colonna 1 delle colonne ORDER BY. Il secondo argomento in START AT definisce il valore iniziale nella colonna 2 delle colonne ORDER BY all'interno delle righe che soddisfano il primo valore per la colonna 1.

## Esempio START AT

```
DAX

EVALUATE
    'Internet Sales'

ORDER BY
    'Internet Sales'[Sales Order Number]
START AT "SO7000"
```

Restituisce tutte le righe e le colonne della tabella Internet Sales, in ordine crescente in base al numero ordine di vendita, a partire da SO7000.

The screenshot shows the Microsoft Power BI Data Editor interface. The top bar has tabs for File, Edit, View, Query, Project, Debug, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main area has tabs for 'MSDAXQuery4.msd...' and 'Object Explorer'. The 'MSDAXQuery4.msd...' tab contains the following DAX code:

```
EVALUATE
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Sales Order Number]
START AT "S07000"
```

The 'Results' tab displays a table with the following data:

Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[R...]	Internet Sales[O...
604	28153	1	19	6	S070000	1	1	1
538	28153	1	19	6	S070000	2	1	1
225	28153	1	19	6	S070000	3	1	1
222	21797	1	100	1	S070001	2	1	1
384	21797	1	100	1	S070001	1	1	1

Below the table, a message says 'Query executed successfully.' and shows connection details: 'asazure://westus.asazure.wi...', 'adventureworks', and '00:00:03'. The status bar at the bottom shows 'Ready', 'Ln 6', 'Col 18', 'Ch 18', and 'INS'.

## DEFINE (facoltativa)

La parola chiave **DEFINE** facoltativa introduce una o più definizioni di entità calcolate esistenti solo per la durata della query. Le definizioni precedono l'istruzione **EVALUATE** e sono valide per tutte le istruzioni **EVALUATE** nella query. Le definizioni possono essere variabili, misure, tabelle<sup>1</sup> e colonne<sup>1</sup>. Le definizioni possono fare riferimento ad altre definizioni visualizzate prima o dopo la definizione corrente. Almeno una definizione è necessaria se la parola chiave **DEFINE** è inclusa in una query.

## Sintassi DEFINE

DAX

```
[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
)
+
]
(EVALUATE <table expression>) +
```

## Parametri DEFINE

Termino	Definizione
Entity	MISURA, VAR, TABELLA <sup>1</sup> o COLONNA <sup>1</sup> .

Termino	Definizione
name	Nome di una definizione di misura, var, tabella o colonna. Non può essere un'espressione. Il nome non deve essere univoco. Il nome esiste solo per la durata della query.
expression	Qualsiasi espressione DAX che restituisce una tabella o un valore scalare. L'espressione può usare una qualsiasi delle entità definite. Se è necessario convertire un'espressione scalare in un'espressione di tabella, eseguire il wrapping dell'espressione all'interno di un costruttore di tabella con parentesi graffe {} oppure usare la funzione <code>ROW()</code> per restituire una singola tabella di righe.

[1] **Attenzione:** le definizioni TABLE e COLUMN con ambito query sono destinate solo all'uso interno. Sebbene sia possibile definire espressioni TABLE e COLUMN per una query senza errori di sintassi, possono generare errori di runtime e non sono consigliate.

## Osservazioni DEFINE

- Una query DAX può avere più istruzioni EVALUATE, ma può avere una sola istruzione DEFINE. Le definizioni nell'istruzione DEFINE possono essere applicate a qualsiasi istruzione EVALUATE nella query.
- In un'istruzione DEFINE è necessaria almeno una definizione.
- Le definizioni di misura per un modello di override della query hanno lo stesso nome, ma vengono usate solo all'interno della query. Non influiranno sulla misura del modello.
- I nomi VAR hanno restrizioni univoche. Per altre informazioni, vedere [VAR - Parametri](#).

## Esempio DEFINE

DAX

```
DEFINE
    MEASURE 'Internet Sales'[Internet Total Sales] =
        SUM ( 'Internet Sales'[Sales Amount] )

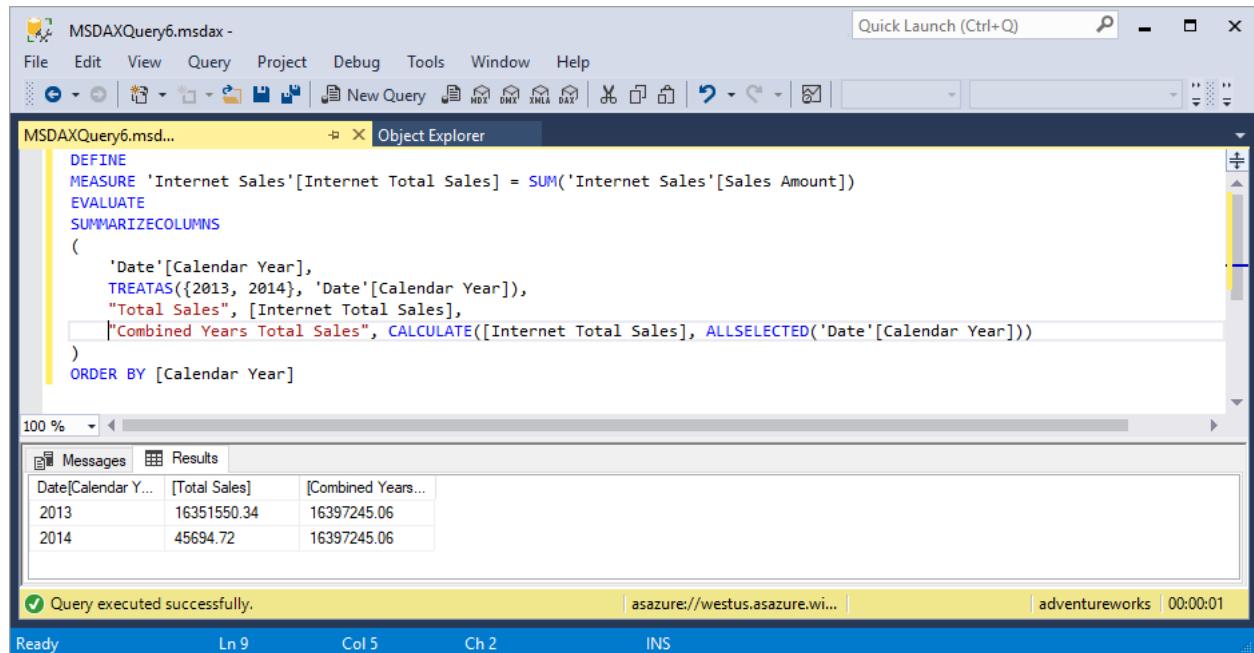
EVALUATE
SUMMARIZECOLUMNS (
    'Date'[Calendar Year],
    TREATAS (
        {
            2013,
            2014
        },
        'Date'[Calendar Year]
    )
) ON COLUMNS
```

```

'Date'[Calendar Year]
),
"Total Sales", [Internet Total Sales],
"Combined Years Total Sales",
CALCULATE (
    [Internet Total Sales],
    ALLSELECTED ( 'Date'[Calendar Year] )
)
)
ORDER BY [Calendar Year]

```

Restituire le vendite totali calcolate per gli anni 2013 e 2014 e le vendite totali calcolate combinate per gli anni 2013 e 2014, sotto forma di tabella. La misura nell'istruzione DEFINE, Internet Total Sales, viene usata in entrambe le espressioni Total Sales e Combined Years Total Sales.



The screenshot shows the Microsoft Power BI Data Editor interface. The top menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The title bar says "MSDAXQuery6.msdax -". The main area displays a DAX query:

```

DEFINE
MEASURE 'Internet Sales'[Internet Total Sales] = SUM('Internet Sales'[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    'Date'[Calendar Year],
    TREATAS({2013, 2014}, 'Date'[Calendar Year]),
    "Total Sales", [Internet Total Sales],
    ["Combined Years Total Sales", CALCULATE([Internet Total Sales], ALLSELECTED('Date'[Calendar Year]))]
)
ORDER BY [Calendar Year]

```

Below the query, the "Results" pane shows a table with the following data:

Date[Calendar Y...	[Total Sales]	[Combined Years...
2013	16351550.34	16397245.06
2014	45694.72	16397245.06

At the bottom, a status bar indicates "Query executed successfully.", the URL "asazure://westus.asazure.wi...", the database "adventureworks", and the time "00:00:01".

## Parametri nelle query DAX

Un'istruzione di query DAX definita in modo corretto può essere parametrizzata e quindi usata più volte modificando solo i valori dei parametri.

Il [metodo Execute \(XMLA\)](#) dispone di un [elemento Parameters \(XMLA\)](#) di tipo raccolta che consente la definizione e l'assegnazione di un valore per i parametri. All'interno della raccolta ogni [elemento Parameter \(XMLA\)](#) definisce il nome e un valore per il parametro.

Fare riferimento ai parametri XMLA aggiungendo un carattere @ come prefisso del nome del parametro. In qualsiasi posizione nella sintassi in cui è consentito un valore, il

valore può essere sostituito con una chiamata al parametro. Tutti i parametri XMLA sono tipizzati come testo.

### Importante

I parametri definiti nella sezione dei parametri e non usati nell'elemento <STATEMENT> generano una risposta di errore in XMLA. I parametri usati e non definiti nell'elemento <Parameters> generano una risposta di errore in XMLA.

## Vedi anche

[Istruzioni DAX](#)

[SUMMARIZECOLUMNS](#)

[TREATAS](#)

[FILTER](#)

---

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No

# Convenzioni di denominazione dei parametri DAX

Articolo • 20/10/2023

Nelle informazioni di riferimento di DAX i nomi dei parametri sono standardizzati per semplificare l'uso e la comprensione delle funzioni.

## Nomi dei parametri

Termine	Definizione
expression	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata più volte (per ogni riga/contesto).
value	Qualsiasi espressione DAX che restituisce un singolo valore scalare, dove l'espressione deve essere valutata esattamente una volta prima di tutte le altre operazioni.
table	Qualsiasi espressione DAX che restituisce una tabella di dati.
tableName	Nome di una tabella esistente con la sintassi DAX standard. Non può essere un'espressione.
columnName	Nome di una colonna esistente con la sintassi DAX standard, di solito completo. Non può essere un'espressione.
name	Costante di tipo stringa che verrà usata per fornire il nome di un nuovo oggetto.
order	Enumerazione usata per determinare l'ordinamento.
tà	Enumerazione usata per determinare la gestione dei valori equivalenti.
Tipo	Enumerazione usata per determinare il tipo di dati per PathItem e PathItemReverse.

## Aggiunta di un prefisso ai nomi di parametro o uso del solo prefisso

Termine	Definizione
prefixing	I nomi dei parametri possono essere ulteriormente qualificati con un prefisso descrittivo per indicare la modalità d'uso dell'argomento ed evitare la lettura ambigua dei parametri. Ad esempio:

Termine	Definizione
	<p>Result_ColumnName: fa riferimento a una colonna esistente usata per ottenere i valori dei risultati nella funzione LOOKUPVALUE().</p> <p>Search_ColumnName: fa riferimento a una colonna esistente usata per cercare un valore nella funzione LOOKUPVALUE().</p>
omitting	<p>I nomi dei parametri verranno omessi se il prefisso è sufficientemente chiaro per descrivere il parametro.</p> <p>Ad esempio, rispetto a DATE (Year_Value, Month_Value, Day_Value), la sintassi DATE (Year, Month, Day) risulta di più facile lettura per l'utente. La ripetizione del valore del suffisso non contribuisce in alcun modo alla comprensione della funzione e disturba inutilmente lettura.</p> <p>Se tuttavia il parametro con prefisso è Year_columnName, il nome del parametro e il prefisso rimarranno invariati affinché l'utente sia in grado di comprendere che il parametro richiede un riferimento a una colonna Years esistente.</p>

## Commenti e suggerimenti

Questa pagina è stata utile?



# Sintassi DAX

Articolo • 20/10/2023

Questo articolo descrive la sintassi e i requisiti del linguaggio di espressioni delle formule DAX.

## Requisiti della sintassi

Una formula DAX inizia sempre con un segno di uguale (=). Dopo il segno di uguale, è possibile specificare qualsiasi espressione che restituisce un valore scalare o un'espressione che può essere convertita in un valore scalare. Di seguito sono elencate le quattro opzioni disponibili.

- Una costante scalare o un'espressione che usa un operatore scalare (+,-,\*,/,>=,...,&&, ...)
- Riferimenti a colonne o tabelle. Come input per le funzioni, nel linguaggio DAX vengono sempre usate tabelle e colonne e mai matrici o set arbitrari di valori.
- Operatori, costanti e valori forniti come parte di un'espressione.
- Il risultato di una funzione e i relativi argomenti obbligatori. Alcune funzioni DAX restituiscono una tabella anziché un valore scalare ed è quindi necessario eseguirne wrapping in una funzione in grado di valutare la tabella e restituire un valore scalare. Fa eccezione il caso in cui la tabella ha una sola colonna e una sola riga e viene quindi trattata come un valore scalare.

La maggior parte delle funzioni DAX richiede uno o più argomenti che possono includere tabelle, colonne, espressioni e valori. Tuttavia, alcune funzioni, ad esempio PI, non richiedono argomenti, ma richiedono sempre le parentesi per indicare l'argomento null. È ad esempio necessario digitare sempre PI() e non PI. È inoltre possibile annidare le funzioni all'interno di altre funzioni.

- Espressioni. Un'espressione può contenere uno o più dei seguenti elementi: operatori, costanti o riferimenti a colonne.

Ad esempio, le formule riportate di seguito sono tutte valide.

Formula	Result
= 3	3
= "Sales"	Sales

Formula	Result
= 'Sales'[Amount]	Se si usa questa formula all'interno della tabella Sales, si otterrà il valore della colonna Amount nella tabella Sales per la riga corrente.
= (0.03 * [Amount])	Il 3% del valore nella colonna Amount della tabella corrente.
= 0.03 * [Amount]	Anche se questa formula può essere usata per calcolare una percentuale, il risultato non viene visualizzato come percentuale a meno che non si applichi la formattazione nella tabella.
= PI()	Valore della costante pi.

Le formule possono comportarsi in modo diverso a seconda della modalità di utilizzo. È necessario tenere sempre presente il contesto e il modo in cui i dati usati nella formula sono correlati ad altri dati che potrebbero essere usati nel calcolo.

## Requisiti per la denominazione

Un modello di dati contiene spesso più tabelle. Nel loro complesso, le tabelle e le relative colonne costituiscono un database archiviato nel motore di analisi in memoria (VertiPaq). All'interno di tale database, tutte le tabelle devono avere nomi univoci. Inoltre, le colonne devono avere nomi univoci all'interno di ogni tabella. Per tutti i nomi di oggetto *non viene fatta distinzione tra maiuscole e minuscole*. I nomi **SALES** e **Sales**, ad esempio, rappresentano la stessa tabella.

Ogni colonna e misura aggiunta a un modello di dati esistente deve appartenere a una tabella specifica. La tabella che contiene la colonna può essere specificata in modo implicito, quando si crea una colonna calcolata all'interno di una tabella, oppure in modo esplicito, quando si crea una misura e si specifica il nome della tabella in cui deve essere archiviata la relativa definizione.

Quando si usa una tabella o una colonna come input per una funzione, è in genere necessario *specificare il nome completo* della colonna. Il nome *completo* di una colonna è costituito dal nome della tabella, seguito da quello della colonna tra parentesi quadre: ad esempio, 'U.S. Sales'[Products]. L'uso di un nome completo è sempre obbligatorio quando si fa riferimento a una colonna nei contesti seguenti:

- Come argomento per la funzione, VALUES
- Come argomento per le funzioni, ALL o ALLEXCEPT
- In un argomento di filtro per le funzioni, CALCULATE o CALCULATETABLE
- Come argomento per la funzione, RELATEDTABLE

- Come argomento per qualsiasi funzione di Business Intelligence per le gerarchie temporali

Il nome *non completo* corrisponde solo al nome della colonna, racchiuso tra parentesi quadre, ad esempio [Sales Amount]. Ad esempio, quando si fa riferimento a un valore scalare dalla stessa riga della tabella corrente, è possibile usare il nome di colonna non completo.

Se il nome di una tabella contiene spazi, parole chiave riservate o caratteri non consentiti, è necessario racchiuderlo tra virgolette singole. È inoltre necessario racchiudere i nomi di tabella tra virgolette se contengono caratteri al di fuori dell'intervallo di caratteri alfanumerici ANSI, indipendentemente dal fatto che il set di caratteri sia supportato dalle impostazioni locali. Ad esempio, se si apre una cartella di lavoro che contiene nomi di tabella scritti in caratteri cirillici, come 'Таблица', tali nomi devono essere racchiusi tra virgolette, anche se non contengono spazi.

#### Nota

Per immettere più facilmente i nomi completi di colonne, usare la funzionalità di completamento automatico nell'editor delle formule.

## Tabelle

- I nomi delle tabelle sono necessari ogni volta che la colonna proviene da una tabella diversa da quella corrente e devono essere univoci all'interno del database.
- I nomi delle tabelle devono essere racchiusi tra virgolette singole se contengono spazi, altri caratteri speciali o caratteri alfanumerici non in lingua inglese.

## Misure

- I nomi delle misure devono essere sempre racchiusi tra parentesi quadre.
- I nomi delle misure possono contenere spazi.
- Ogni nome di misura deve essere univoco all'interno di un modello. Pertanto, quando si fa riferimento a una misura esistente, il nome della tabella davanti al nome della misura è facoltativo. Tuttavia, quando si crea una misura, è sempre necessario specificare il nome della tabella in cui verrà archiviata la definizione della misura.

## Colonne

I nomi delle colonne devono essere univoci nel contesto di una tabella. Possono tuttavia essere presenti più tabelle che contengono colonne con gli stessi nomi. In tal caso, l'ambiguità viene risolta specificando il nome della tabella.

In generale, è possibile fare riferimento alle colonne senza fare riferimento alla tabella di base a cui appartengono, tranne nel caso in cui si verifichi un conflitto di nome da risolvere o vengano usate funzioni che richiedono nomi di colonna completi.

## Parole chiave riservate

Se il nome usato per una tabella corrisponde a una parola chiave riservata di Analysis Services, viene restituito un errore ed è necessario rinominare la tabella. È tuttavia possibile usare parole chiave nei nomi di oggetto se il nome dell'oggetto è racchiuso tra parentesi quadre (per le colonne) o tra virgolette (per le tabelle).

### ⚠ Nota

Le virgolette possono essere rappresentate da caratteri diversi, a seconda dell'applicazione. Se si incollano formule da un documento esterno o da una pagina Web, controllare il codice ASCII del carattere usato per le virgolette di apertura e chiusura per assicurarsi che siano uguali, altrimenti DAX potrebbe non essere in grado di riconoscere i simboli come virgolette e il riferimento non risulterebbe valido.

## Caratteri speciali

I caratteri e tipi di carattere seguenti non sono validi per i nomi di tabelle, colonne o misure:

- Spazi iniziali o finali, a meno che gli spazi non siano racchiusi tra delimitatori di nome, parentesi quadre o apostrofi singoli.
- Caratteri di controllo
- I caratteri seguenti non sono validi per i nomi degli oggetti:  
`.,';^*|?&%$!+=()[]{}<>`

## Esempi di nomi di oggetto

Nella tabella seguente sono riportati alcuni esempi di nomi di oggetto:

<b>Tipi di oggetto</b>	<b>Esempi</b>	<b>Commento</b>
Nome tabella	<b>Sales</b>	Se il nome della tabella non contiene spazi o altri caratteri speciali, non è necessario racchiuderlo tra virgolette.
Nome tabella	<b>'Canada Sales'</b>	Se il nome contiene spazi, tabulazioni o altri caratteri speciali, racchiuderlo tra virgolette singole.
Nome di colonna completo	<b>Sales[Amount]</b>	Il nome della tabella precede quello della colonna e quest'ultimo è racchiuso tra parentesi quadre.
Nome di misura completo	<b>Sales[Profit]</b>	Il nome della tabella precede quello della misura e quest'ultimo è racchiuso tra parentesi quadre. In determinati contesti, un nome completo è sempre obbligatorio.
Nome di colonna non completo	<b>[Amount]</b>	Il nome non completo corrisponde al semplice nome della colonna, tra parentesi quadre. Tra i contesti in cui è possibile usare il nome non completo sono incluse le formule in una colonna calcolata all'interno della stessa tabella o una funzione di aggregazione che esegue l'analisi della stessa tabella.
Nome di colonna completo in tabella con spazi	<b>'Canada Sales'[Qty]</b>	Il nome della tabella contiene spazi e pertanto deve essere racchiuso tra virgolette singole.

## Altre limitazioni

La sintassi richiesta per ogni funzione e il tipo di operazione che può eseguire variano notevolmente a seconda della funzione. In generale, tuttavia, le regole seguenti si applicano a tutte le formule ed espressioni:

- Le formule e le espressioni DAX non possono modificare né inserire singoli valori nelle tabelle.
- Non è possibile creare righe calcolate tramite DAX. È possibile creare solo colonne calcolate e misure.
- Quando si definiscono le colonne calcolate, è possibile annidare le funzioni a qualsiasi livello.

- In DAX sono disponibili diverse funzioni che restituiscono una tabella. In genere, i valori restituiti da queste funzioni vengono usati come input per altre funzioni, che richiedono una tabella come input.

## Operatori e costanti DAX

Nella tabella seguente sono elencati gli operatori supportati da DAX. Per altre informazioni sulla sintassi dei singoli operatori, vedere [Operatori DAX](#).

Tipo di operatore	Simbolo e uso
Operatore di parentesi	() ordine di precedenza e raggruppamento di argomenti
Operatori aritmetici	+ (addizione)  - (sottrazione/  segno)  * (moltiplicazione)  / (divisione)  ^ (elevamento a potenza)
Operatori di confronto	= (uguale a)  > (maggiore di)  < (minore di)  >= (maggiore o uguale a)  <= (minore o uguale a)  <> (diverso da)
Operatore di concatenazione del testo	& (concatenazione)
Operatori logici	&& (e)     (or)

## Tipi di dati

Non è necessario sottoporre a cast, convertire o specificare in altro modo il tipo di dati di una colonna o di un valore usato in una formula DAX. Quando si usano dati in una formula DAX, DAX identifica automaticamente i tipi di dati nelle colonne a cui si fa riferimento e i valori digitati ed esegue conversioni implicite, quando necessario, per completare l'operazione specificata.

Se, ad esempio, si prova ad aggiungere un numero a un valore di data, il motore interpreterà l'operazione nel contesto della funzione e convertirà i numeri in un tipo di dati comune, quindi presenterà il risultato nel formato previsto, ovvero una data.

Esistono alcune limitazioni relative ai valori che possono essere convertiti correttamente. Se il tipo di dati di un valore o di una colonna è incompatibile con l'operazione corrente, DAX restituisce un errore. Inoltre, in DAX non sono disponibili funzioni che consentono di modificare, convertire o sottoporre a cast in modo esplicito il tipo di dati dei dati esistenti importati in un modello.

### Importante

DAX non supporta l'uso del tipo di dati Variant. Pertanto, quando si caricano o si importano dati in un modello, si presuppone che i dati di ogni colonna siano in genere di un tipo coerente.

Alcune funzioni restituiscono valori scalari, incluse stringhe, mentre altre funzioni utilizzano numeri, sia interi che reali, oppure date e ore. Il tipo di dati richiesto per ogni funzione è descritto nella sezione [Funzioni DAX](#).

Come argomento di una funzione è possibile usare tabelle contenenti più colonne e righe di dati. Alcune funzioni restituiscono anche tabelle, che vengono archiviate in memoria e possono essere usate come argomenti per altre funzioni.

## Data e ora

DAX archivia i valori di data e ora usando il tipo di dati datetime usato da Microsoft SQL Server. Il formato Datetime usa un numero a virgola mobile in cui i valori di data corrispondono alla parte intera che rappresenta il numero di giorni dal 30 dicembre 1899. I valori di ora corrispondono alla parte decimale di un valore di data in cui ore, minuti e secondi sono rappresentati da frazioni decimali di un giorno. Le funzioni di data e ora DAX convertono in modo implicito gli argomenti in un tipo di dati datetime.

### Nota

Il valore DateTime massimo esatto supportato da DAX è il 31 dicembre 9999 00:00:00.

## Valore letterale data e ora

A partire dalla versione di agosto 2021 di Power BI Desktop, i valori di data e datetime DAX possono essere specificati come valore letterale nel formato `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` o `dt"YYYY-MM-DD hh:mm:ss"`. Se specificato come valore letterale, non è necessario usare le funzioni `DATE`, `TIME`, `DATEVALUE`, `TIMEVALUE` nell'espressione.

Ad esempio, l'espressione seguente usa funzioni DATE e TIME per filtrare in OrderDate:

DAX

```
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > (DATE(2015,1,9) + TIME(2,30,0)) && [OrderDate] <  
    (DATE(2015,12,31) + TIME(11,59,59))  
)
```

La stessa espressione di filtro può essere specificata come valore letterale:

DAX

```
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > dt"2015-1-9T02:30:00" && [OrderDate] < dt"2015-12-  
31T11:59:59"  
)
```

### ⓘ Nota

Il formato letterale data e datetime tipizzato DAX non è supportato in tutte le versioni di Power BI Desktop, Analysis Services e Power Pivot in Excel. Le funzionalità DAX nuove e aggiornate vengono in genere introdotte in Power BI Desktop e successivamente incluse in Analysis Services e Power Pivot in Excel.

## Commenti e suggerimenti

Questa pagina è stata utile?

 Sì

 No