

Data Analysis Expressions (DAX)

Опис та приклади застосування - Функції дати та часу

Опис та приклади застосування - Логічні функції

Опис та приклади застосування - Функції зв'язків

Опис та приклади застосування - Текстові функції

CALENDAR - Повертає таблицю з одним стовпцем під назвою «Дата», який містить безперервний набір дати. Діапазон дат – від зазначеної дати початку до зазначеної дати завершення, включаючи ці дві дати.

DAX

CALENDAR(<start_date>, <end_date>)

Повертає таблицю з одним стовпцем під назвою «Дата», що містить безперервний набір дат. Діапазон дат — від зазначеної дати початку до зазначеної кінцевої дати включно ці дві дати

Приклад1. Наступна формула повертає таблицю з датами від 1 січня 2015 року до 31 грудня 2021 р.

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

Приклад2. Для моделі даних, яка включає фактичні дані про продажі та майбутні прогнози продажів, наступний вираз повертає таблицю дат, що охоплює діапазон дат в обох продажах і Прогнозні таблиці.

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

Функції дати та часу - Ці функції допомагають створювати обчислення на основі дат і часу. Багато з функції в DAX подібні до функцій дати й часу Excel. Однак DAX функції використовують тип даних datetime і можуть приймати значення зі стовпця як аргумент.

DATE - Повертає вказану дату у форматі дати й часу.

DAX

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

Функція ДАТА приймає цілі числа, які вводяться як аргументи, і генерує відповідну дату. Функція ДАТА найбільш корисна в ситуаціях, коли рік, місяць і день задаються формулами. Наприклад, основні дані може містити дати у форматі, який не розпізнається як дата, наприклад РРРРММДД. Ви можете використовувати функцію ДАТА в поєднанні з іншими функціями перетворити дати на число, яке можна розпізнати як дату. На відміну від Microsoft Excel, який зберігає дати як порядковий номер, дата DAX функції завжди повертають тип даних datetime. Однак ви можете використовувати форматування відображати дати як порядкові номери, якщо хочете. Дата й час також можуть бути вказані як літерали у форматі dt"РРРР-ММ-ДД", dt"YYYY-MM-DDThh:mm:ss" або dt"YYYY-MM-DD hh:mm:ss" . Коли вказано як літерал, використовувати функцію ДАТА у виразі не обов'язково.

Приклад 1. Роки до 1899 року

DATE(08,1,2)

Якщо значення, яке ви вводите для аргументу року, становить від 0 (нуль) до 1899 (включно), це значення додається до 1900 для обчислення року. Наступна формула повертається 2 січня 1908 р.: (1900+08).

Приклад 2. Якщо рік знаходиться між 1900 і 9999 (включно), це значення використовується як рік. Наступне формула повертає 2 січня 2008 р.:

DATE(2008,1,2)

Приклад 3.

Якщо місяць більше 12, місяць додає цю кількість місяців до першого місяця вказаний рік. Наступна формула повертає дату 2 лютого 2009 р.:

DATE(2008,14,2)

EDATE - Повертає дату, яка є вказаною кількістю місяців до або після дати початку. Використовуйте EDATE, щоб обчислити дати погашення або терміни, які припадають на той самий день місяць як дата видачі.

DAX

EDATE(<start_date>, <months>)

На відміну від Microsoft Excel, який зберігає дати як послідовні порядкові номери, DAX працює з датами у форматі дати й часу. Дати, збережені в інших форматах, конвертуються неявно. Якщо дата_початку недійсна, EDATE повертає помилку. Переконайтеся, що колонка посилання або дата, яку ви вказуєте як перший аргумент, є датою. Якщо місяці не є цілим числом, воно скорочується. Синтаксис EDATE(<дата_початку>, <місяці>) Параметри / Розгорнути таблицю Повернене значення Зауваження Зворотній зв'язок Чи була ця сторінка корисною? Якщо аргумент дати є текстовим представленням дати, функція EDATE використовує параметри мови та часу клієнтського комп'ютера для розуміння текстове значення, щоб виконати перетворення. Якщо поточні налаштування дати та часу представляють дату у форматі місяць/день/рік, а потім наступний рядок "1/8/2009" інтерпретується як значення дати й часу, еквівалентне 8 січня 2009 року. Однак, якщо поточні налаштування дати та часу представляють дату у форматі День/місяць/рік, той самий рядок інтерпретуватиметься як значення дати й часу еквівалентно 1 серпня 2009 року. Якщо запитана дата минула за останнім днем відповідного місяця, то останнім день місяця повертається. Наприклад, такі функції: EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) повернути 28 лютого 2009 рік; що відповідає одному місяцю після дати початку. Ця функція не підтримується для використання в режимі DirectQuery, коли використовується в обчислюванні стовпці або правила безпеки на рівні рядків (RLS)

Наступний приклад повертає дату через три місяці після дати замовлення, тобто зберігається в стовпці [TransactionDate].

EDATE([TransactionDate],3)

NOW() - Повертає поточну дату й час у форматі дати й часу. Функція NOW корисна, коли вам потрібно відобразити поточну дату й час на а робочий аркуш або обчислити значення на основі поточної дати та часу та отримати це значення оновлюється кожного разу, коли ви відкриваєте аркуш

Результат функції NOW змінюється лише тоді, коли стовпець, який містить формула оновлена. Він не оновлюється постійно. У службі Power BI результат функції NOW завжди відображається в UTC часовий пояс. Функція TODAY повертає ту саму дату, але не є точною щодо часу; повертається час завжди 12:00:00 AM і оновлюється лише дата.

UTCNOW() - Повертає поточну дату й час за UTC.

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

Повертає: 2/2/2018 4:48:08 AM

Логічні функції діють на вираз, щоб повернути інформацію про значення або набори у виразі. Наприклад, ви можете використовувати функцію IF, щоб перевірити результат вираз і створити умовні результати.

```
AND(<logical1>,<logical2>)
```

Функція I в DAX приймає лише два (2) аргументи. Якщо вам потрібно виконати ТА операції над кількома виразами, ви можете створити ряд обчислень або, краще, використовуйте оператор AND (&&), щоб об'єднати їх усі у простіший вираз.

У наведеному нижче прикладі використовується функція I з вкладеними формулами для порівняння двох наборів розрахунків одночасно. Для кожної категорії продукту формула визначає якщо продажі поточного року та продажі минулого року Інтернет-каналу є більшими, ніж Реселлерський канал за ті ж періоди. Якщо обидві умови вірні, для кожної категорії формула повертає значення "Інтернет-хіт".

```
= 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" , ""  
)
```

`OR(<logical1>,<logical2>)`

Логічне значення. Значення TRUE, якщо будь-який із двох аргументів TRUE; значення є FALSE, якщо обидва аргументи є FALSE. Функція АБО в DAX приймає лише два (2) аргументи. Якщо вам потрібно виконати Операція АБО над кількома виразами, ви можете створити ряд обчислень або, Краще використовуйте оператор АБО (||), щоб об'єднати їх усі у простіший вираз. Функція обчислює аргументи до першого аргументу TRUE, а потім повертає ПРАВДА.

У наступному прикладі показано, як використовувати функцію АБО, щоб отримати продавців які належать до Кола Досконалості. The Circle of Excellence відзначає тих, хто Синтаксис АБО(<логічний1>,<логічний2>) Параметри / Розгорнути таблицю Повернене значення Зауваження приклад досягли понад мільйона доларів продажів туристичних велосипедів або продажів понад двох і півтора мільйона доларів у 2007 році.

```
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" , "" )
```

COALESCE - Повертає перший вираз, який не має значення ПУСТИЙ. Якщо всі вирази обчислюються на BLANK, повертається BLANK

DAX

```
COALESCE(<expression>, <expression>[, <expression>]...)
```

Приклад 1. Повертає 10, який є першим виразом, який не має значення ПУСТИЙ

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Приклад 2. Повертає суму всіх значень у стовпці SalesAmount у таблиці FactInternetSales, або 0. Це можна використовувати для перетворення ПУСТИХ значень загального обсягу продажів на 0.

```
COALESCE(SUM(FactInternetSales[SalesAmount]), 0)
```

DAX

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

Вираз, який потрібно обчислити, може бути постійним значенням або виразом. А звичайним використанням цієї функції є встановлення першого параметра на TRUE. Подивіться приклади нижче. Усі вирази результату та вираз else мають бути одного типу даних. Порядок умов має значення. Як тільки одне значення збігається, відповідне повертається результат, а інші наступні значення не оцінюються. Переконайтеся, що найбільш обмежувальні значення для оцінки вказуються перед менш обмежувальними значеннями. Дивіться приклади нижче.

Приклад 1.

```
= 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" )
```

Іншим поширеним використанням SWITCH є заміна кількох вкладених операторів IF. Це є виконується встановленням виразу значення TRUE, як показано в наступному прикладі, який порівнює точку повторного замовлення та рівень безпечного запасу продуктів, щоб визначити потенційні ризики закінчується запас:

```
= 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" )
```

Порядок значень має значення. У наступному прикладі другий результат ніколи повертається, оскільки перше значення є менш обмежувальним, ніж друге. Результат в цьому приклад завжди «А» або «С», але ніколи «В».

```
= SWITCH ( TRUE,  
Product[Standard Cost] < 100, "A",  
Product[Standard Cost] < 10, "B", "C" )
```

Функції зв'язків. Функції цієї категорії призначені для керування та використання зв'язків між таблицями

DAX

RELATED(<column>)

Функція RELATED вимагає наявності зв'язку між поточною таблицею і таблицею з відповідною інформацією. Ви вказуєте стовпець, який містить дані, які вам потрібні, і функція слідує існуючому відношенню «багато до одного». щоб отримати значення з указанного стовпця у відповідній таблиці. Якщо відношення не існує, ви повинні створити стосунки. Коли функція RELATED виконує пошук, вона перевіряє всі значення в зазначену таблицю незалежно від будь-яких застосованих фільтрів. Для функції RELATED потрібен контекст рядка; тому його можна використовувати лише в вираз обчислюваного стовпця, де контекст поточного рядка є однозначним, або як вкладену функцію у вираз, який використовує функцію сканування таблиці. стіл функція сканування, наприклад SUMX, отримує значення поточного значення рядка, а потім сканує іншу таблицю на наявність екземплярів цього значення.

У наступному прикладі показник Non USA Internet Sales створюється для створення звіту про продажі, який не включає продажі в Сполучених Штатах. Щоб створити міру, Таблицю InternetSales_USD потрібно відфільтрувати, щоб виключити всі продажі, які належать United Штати в таблиці SalesTerritory. Сполучені Штати, як країна, з'являються 5 разів у Таблиця SalesTerritory; один раз для кожного з наступних регіонів: північний захід, північний схід, Центральна, південно-західна та південно-східна. Першим підходом до фільтрації Інтернет-продажів для створення показника може бути додайте вираз фільтра, як показано нижче

```
FILTER('InternetSales_USD' ,  
'InternetSales_USD'[SalesTerritoryKey]<>1 &&  
'InternetSales_USD'[SalesTerritoryKey]<>2 &&  
'InternetSales_USD'[SalesTerritoryKey]<>3 &&  
'InternetSales_USD'[SalesTerritoryKey]<>4 &&  
'InternetSales_USD'[SalesTerritoryKey]<>5)
```

Однак цей підхід є неінтуїтивним, схильний до помилок друку та може не працювати, якщо будь-який із існуючих регіонів буде розділений у майбутньому. Кращим підходом було б використовувати існуючий зв'язок між InternetSales_USD і SalesTerritory і прямо вказують, що країна має відрізнитися від США держави. Для цього створіть вираз фільтра, як показано нижче

```
FILTER( 'InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

Цей вираз використовує функцію RELATED для пошуку значення країни в Таблиця SalesTerritory, починаючи зі значення стовпця ключа, SalesTerritoryKey, у Таблиця InternetSales_USD. Результат пошуку використовується функцією фільтра для визначити, чи фільтрується рядок InternetSales_USD.

Приклад 2. У наведеній нижче таблиці показано лише підсумки для кожного регіону, щоб підтвердити, що вираз фільтра за показником «Інтернет-продажі за межами США» працює належним чином.

= SUMX(FILTER('InternetSales_USD' , RELATED('SalesTerritory'[SalesTerritoryCountry]) <>"United States") , 'InternetSales_USD'[SalesAmount_USD])

Для таблиці зліва – результат показано в таблиці з права

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
France	\$2,490,944.57	\$2,490,944.57
Germany	\$2,775,195.60	\$2,775,195.60
United Kingdom	\$5,057,076.55	\$5,057,076.55
United States	\$9,389,479.79	
Grand Total	\$26,054,827.45	\$16,665,347.67

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

RELATEDTABLE - Обчислює табличний вираз у контексті, зміненому вказаними фільтрами.

```
DAX

RELATEDTABLE(<tableName>)
```

Функція RELATEDTABLE змінює контекст, у якому дані фільтруються, і обчислює вираз у новому контексті, який ви вказуєте. Ця функція є скороченням для функції CALCULATETABLE без логіки вираз. Ця функція не підтримується для використання в режимі DirectQuery, коли використовується в обчислювані стовпці або правила безпеки на рівні рядків (RLS).
У наступному прикладі для створення обчислюваного стовпця використовується функція RELATEDTABLE з Інтернет-продажами в таблиці Категорія продукту:
= SUMX(RELATEDTABLE('InternetSales_USD') , [SalesAmount_USD])

Product Category Key	Product Category AlternateKey	Product Category Name	Internet Sales
1	1	Bikes	\$28,318,144.65
2	2	Components	
3	3	Clothing	\$339,772.61
4	4	Accessories	\$700,759.96

CROSSFILTER - Визначає напрямок перехресної фільтрації, який буде використовуватися в обчисленні для зв'язку, який знаходиться між двома колонками.

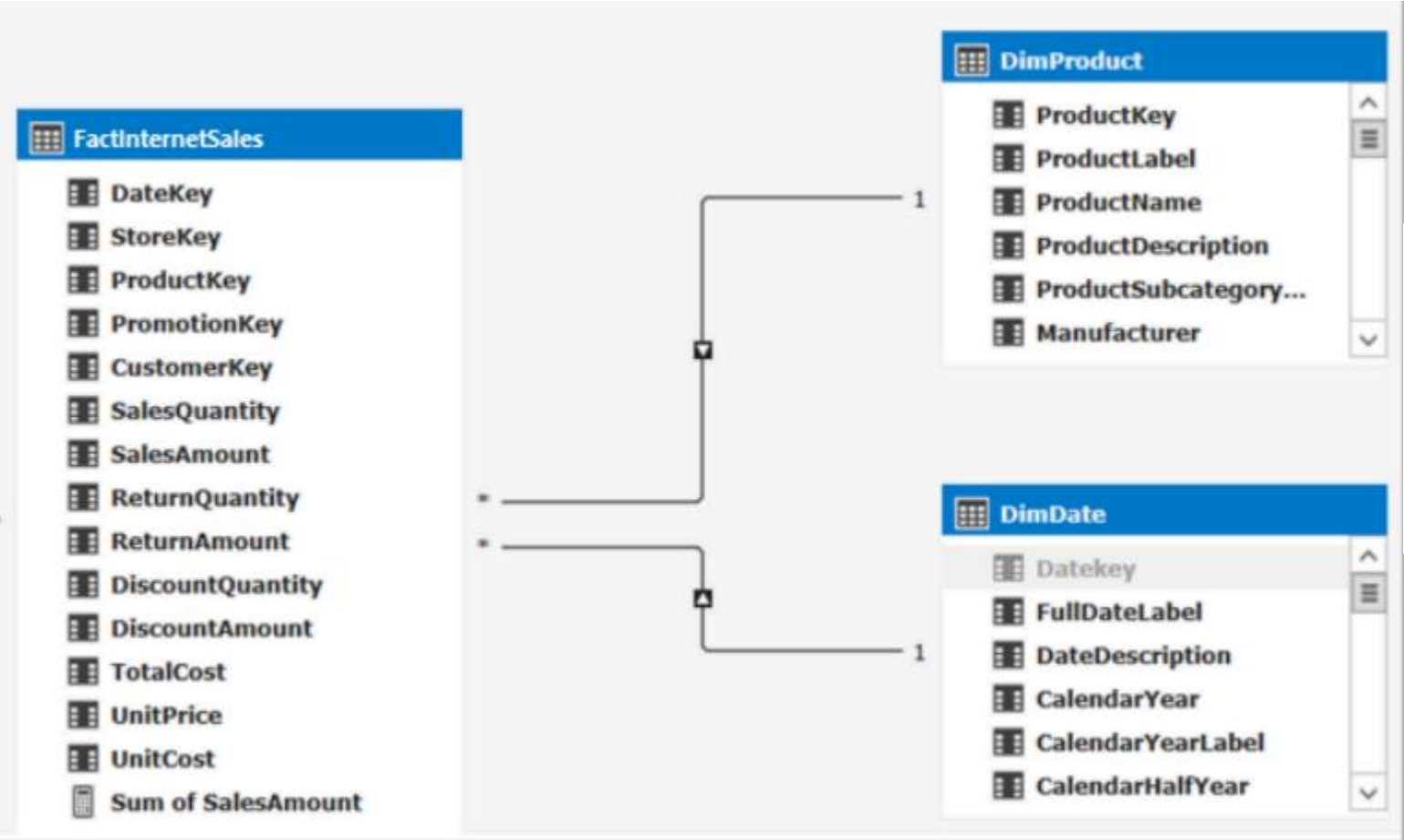
У випадку відношення 1:1 немає різниці між одним і обома напрямком.

CROSSFILTER можна використовувати лише у функціях, які приймають фільтр як аргумент, для приклад: **CALCULATE**, **CALCULATETABLE**, **CLOSINGBALANCEMONTH**, **КІНЦЕВИЙ БАЛАНС КВАРТАЛ**, **КІНЦЕВИЙ БАЛАНС РІК**, **ПОЧАТНИЙ БАЛАНС МІСЯЦЬ**, **OPENINGBALANCEQUARTER**, **OPENINGBALANCEYEAR**, **TOTALMTD**, **TOTALQTD** і Функції **TOTALYTD**.

CROSSFILTER використовує існуючі зв'язки в моделі, ідентифікуючи зв'язки за їхні стовпці кінцевої точки. У **CROSSFILTER** налаштування перехресної фільтрації зв'язку не є важливим; тобто, чи встановлено зв'язок для фільтрації в одному чи в обох напрямках у моделі не впливає на використання функції.

CROSSFILTER перевизначає будь-які існуючі налаштування перехресної фільтрації. Повертається помилка, якщо будь-який із стовпців, названих як аргумент, не є частиною а або аргументи належать до різних відносин. Якщо вирази **CALCULATE** є вкладеними, і більше ніж один вираз **CALCULATE** містить функцію **CROSSFILTER**, тоді внутрішній **CROSSFILTER** є тим, що переважає у випадку конфлікту чи неясності. Ця функція не підтримується для використання в режимі **DirectQuery**, коли використовується в обчислювані стовпці або правила безпеки на рівні рядків (RLS).

На наведеній нижче діаграмі моделі DimProduct і DimDate мають один напрямок відносини з FactInternetSales.



За умовчанням ми не можемо отримати кількість проданих продуктів за роками

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
Grand Total	\$29,358,677.22	606

Є два способи отримати кількість продуктів за роками: Увімкніть двонаправлену перехресну фільтрацію на зв'язку. Це змінить спосіб фільтри працюють для всіх даних між цими двома таблицями. Використовуйте функцію CROSSFILTER, щоб змінити спосіб роботи зв'язків саме для цього міра.

Під час використання DAX ми можемо використовувати функцію CROSSFILTER, щоб змінити спосіб перехресного фільтра напрямок поводить ся між двома стовпцями, визначеними зв'язком. У цьому випадку DAX вираз виглядає так:

```
BiDi:= CALCULATE([Distinct Count of ProductKey], CROSSFILTER(FactInternetSales[ProductKey], DimProduct[ProductKey] , Both))
```

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	
Grand Total	\$39,359,677.20	606	306

USERELATIONSHIP - Вказує зв'язок, який буде використовуватися в певному обчисленні, як той, що існує між columnName1 і columnName2.

DAX

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

USERELATIONSHIP можна використовувати лише у функціях, які приймають фільтр як аргумент, наприклад: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD and TOTALYTD

USERELATIONSHIP не можна використовувати, якщо для таблиці визначено захист на рівні рядка до якого входить міра. Наприклад, CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey])) буде повертає помилку, якщо для DimCustomer визначено захист на рівні рядка.

USERELATIONSHIP використовує існуючі зв'язки в моделі, ідентифікуючи відносини за стовпцями кінцевих точок. У USERELATIONSHIP статус відносин не важливий; тобто чи активний чи ні зв'язок не впливає на використання функції. Навіть якщо зв'язок неактивний, він буде використаний і перекриває будь-який інший активний відносини, які можуть бути присутніми в моделі, але не згадуються у функції аргументи.

Помилка повертається, якщо будь-який із стовпців, названих як аргумент, не є частиною зв'язку або аргументи належать до різних зв'язків.

Якщо для приєднання таблиці А до таблиці В під час обчислення потрібні кілька зв'язків, кожен зв'язок має бути вказано в іншій функції USERELATIONSHIP.

Якщо вирази CALCULATE є вкладеними, і більше ніж один вираз CALCULATE містить функцію USERELATIONSHIP, тоді внутрішній USERELATIONSHIP є тим, який переважає у випадку конфлікту чи неоднозначності.

Можна вкладати до 10 функцій USERELATIONSHIP; однак ваш вираз може мати глибший рівень вкладеності, тобто. наступний приклад виразу вкладено на 3 рівні, але лише на 2 для USERELATIONSHIP: =CALCULATE(CALCULATE(CALCULATE(, USERELATIONSHIP(t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP(t1[colA], t2[colA])) .

Для зв'язків 1-до-1 USERELATIONSHIP активує зв'язок лише в одному напрямку. Зокрема, фільтри зможуть перетікати лише з таблиці columnName2 до таблиці columnName1.

У наведеному нижче прикладі показано, як перевизначити активний зв'язок між Таблиці InternetSales і DateTime.

Щоб обчислити суму інтернет-продажів і дозволити нарізку за датою доставки замість традиційний OrderDate, створіть показник [InternetSales by ShippingDate] за допомогою наступний вираз:

```
= CALCULATE(SUM(InternetSales[SalesAmount]), USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Текстові функції - Вирази аналізу даних (DAX) містять набір текстових функцій на основі бібліотеки рядкові функції в Excel, але які були змінені для роботи з таблицями та стовпцями в табличних моделях. У цьому розділі описано текстові функції, доступні мовою DAX.

DAX

TRIM(<text>)

Видаляє всі пробіли з тексту, крім одиночних пробілів між словами.

Використовуйте TRIM для тексту, який ви отримали від іншої програми, яка могла бути нерегулярний інтервал. Функція TRIM спочатку була розроблена для обрізання 7-бітового пробілу ASCII (значення 32) з тексту. У наборі символів Unicode є додатковий пробіл символ, який називається нерозривним пробілом і має десяткове значення 160. Цей символ зазвичай використовується на веб-сторінках як об'єкт HTML .; за сама по собі функція TRIM не видаляє цей нерозривний пробіл. Для ан приклад того, як видалити обидва пробіли з тексту, див. Видалення пробілів і недруковані символи з тексту.

= TRIM("A column with trailing spaces. ")

COMBINEVALUES - Об'єднує два або більше текстових рядків в один текстовий рядок. Основне призначення цієї функції полягає в підтримці зв'язків із кількома стовпцями в моделях DirectQuery. Докладніше див. у розділі Примітки.

DAX

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[, <expression>]...)
```

The COMBINEVALUES function assumes, but does not validate, that when the input values are different, the output strings are also different. Based on this assumption, when COMBINEVALUES is used to create calculated columns in order to build a relationship that joins multiple columns from two DirectQuery tables, an optimized join condition is generated at query time. For example, if users want to create a relationship between Table1(Column1, Column2) and Table2(Column1, Column2), they can create two calculated columns, one on each table, as:

```
Table1[CalcColumn] = COMBINEVALUES(",", Table1[Column1], Table1[Column2])
```

```
Table2[CalcColumn] = COMBINEVALUES(",", Table2[Column1], Table2[Column2])
```

А потім створіть зв'язок між Table1[CalcColumn] і Таблиця2[CalcColumn] . На відміну від інших функцій і операторів DAX, які є буквально перекладено на відповідні оператори та функції SQL, наведені вище зв'язок генерує предикат з'єднання SQL як:

```
(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND Table2.Column1 IS NULL)
```

Приклад1. EVALUATE
DISTINCT (
SELECTCOLUMNS (Date,
"Month", COMBINEVALUES (
",", [MonthName],
[CalendarYear])))

[Month]
January, 2020
February, 2020
March, 2020
April, 2020
May, 2020
June, 2020
July, 2020
August, 2020
September, 2020
October, 2020
November, 2020
December, 2020
January, 2021
January, 2021