

Contents

DAX(Data Analysis Expressions) 참조

Learn

[DAX 개요](#)

[동영상](#)

[Power BI Desktop Learn 경로에서 DAX 사용](#)

DAX 함수

[DAX 함수 참조](#)

[새로운 DAX 함수](#)

[날짜 및 시간 함수](#)

[날짜 및 시간 함수 개요](#)

[CALENDAR](#)

[CALENDARAUTO](#)

[DATE](#)

[DATEDIFF](#)

[DATEVALUE](#)

[DAY](#)

[EDATE](#)

[EOMONTH](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[NOW](#)

[QUARTER](#)

[SECOND](#)

[TIME](#)

[TIMEVALUE](#)

[TODAY](#)

[UTCNOW](#)

[UTCTODAY](#)

WEEKDAY

WEEKNUM

YEAR

YEARFRAC

필터 함수

필터 함수 개요

ALL

ALLCROSSFILTERED

ALLEXCEPT

ALLNOBLANKROW

ALLSELECTED

CALCULATE

CALCULATETABLE

EARLIER

EARLIEST

FILTER

KEEPFILTERS

LOOKUPVALUE

REMOVEFILTERS

SELECTEDVALUE

재무 함수

재무 함수 개요

ACCRINT

ACCRINTM

AMORDEGRC

AMORLINC

COUPDAYBS

COUPDAYS

COUPDAYSNC

COUPNCD

COUPNUM

COUPPCD

CUMIPMT
CUMPRINC
DB
DDB
DISC
DOLLARDE
DOLLARFR
DURATION
EFFECT
FV
INTRATE
IPMT
ISPMT
MDURATION
NOMINAL
NPER
ODDFPRICE
ODDFYIELD
ODDLPRICE
ODDLYIELD
PDURATION
PMT
PPMT
PRICE
PRICEDISC
PRICEMAT
PV
RATE
RECEIVED
RRI
SLN
SYD

TBILLEQ
TBILLPRICE
TBILLYIELD
VDB
XIRR
XNPV
YIELD
YIELDDISC
YIELDMAT

정보 함수

정보 함수 개요

CONTAINS
CONTAINSROW
CONTAINSSTRING
CONTAINSSTRINGEXACT
CUSTOMDATA
HASONEFILTER
HASONEVALUE
ISBLANK
ISCROSSFILTERED
ISEMPTY
ISERROR
ISEVEN
ISFILTERED
ISINSCOPE
ISLOGICAL
ISNONTEXT
ISNUMBER
ISODD
ISONORAFTER
ISSELECTEDMEASURE
ISSUBTOTAL

ISTEXT

NONVISUAL

SELECTEDMEASURE

SELECTEDMEASUREFORMATSTRING

SELECTEDMEASURENAME

USERNAME

USEROBJECTID

USERPRINCIPALNAME

논리 함수

논리 함수 개요

AND

COALESCE

FALSE

IF

IF.EAGER

IFERROR

NOT

또는

SWITCH

TRUE

삼각함수 및 수학 함수

삼각함수 및 수학 함수 개요

ABS

ACOS

ACOSH

ACOT

ACOTH

ASIN

ASINH

ATAN

ATANH

CEILING

COMBIN

COMBINA

CONVERT

COS

COSH

통화

DEGREES

DIVIDE

EVEN

EXP

FACT

FLOOR

GCD

INT

ISO.CEILING

LCM

LN

LOG

LOG10

MOD

MROUND

ODD

PI

POWER

제품

PRODUCTX

QUOTIENT

RADIANS

RAND

RANDBETWEEN

ROUND

ROUNDDOWN

ROUNDUP

SIGN

SQRT

합계

SUMX

TRUNC

기타 함수

기타 함수 개요

BLANK

오류

부모 및 자식 함수

부모 및 자식 함수 개요

부모-자식 계층 구조를 위한 함수 이해

PATH

PATHCONTAINS

PATHITEM

PATHITEMREVERSE

PATHLENGTH

관계 함수

관계 함수

CROSSFILTER

RELATED

RELATEDTABLE

USERELATIONSHIP

통계 함수

통계 함수 개요

APPROXIMATEDISTINCTCOUNT

AVERAGE

AVERAGEA

AVERAGEX

BETA.DIST

BETA.INV

CHISQ.DIST
CHISQ.DIST.RT
CHISQ.INV
CHISQ.INV.RT
CONFIDENCE.NORM
CONFIDENCE.T
COT
COTH
개수
COUNTA
COUNTAX
COUNTBLANK
COUNTROWS
COUNTX
DISTINCTCOUNT
EXPON.DIST
GEOMEAN
GEOMEANX
최대
MAXA
MAXX
MEDIAN
MEDIANX
최소
MINA
MINX
NORM.DIST
NORM.INV
NORM.S.DIST
NORM.S.INV
PERCENTILE.EXC
PERCENTILE.INC

PERCENTILEX.EXC

PERCENTILEX.INC

PERMUT

POISSON.DIST

RANK.EQ

RANKX

SAMPLE

SIN

SINH

STDEV.S

STDEV.P

STDEVX.S

STDEVX.P

SQRTPI

T.DIST

T.DIST.2T

T.DIST.RT

T.INV

T.INV.2T

TAN

TANH

VAR.S

VAR.P

VARX.S

VARX.P

테이블 조작 함수

테이블 조작 함수 개요

ADDCOLUMNS

ADDMISSINGITEMS

CROSSJOIN

CURRENTGROUP

DATATABLE

DETAILROWS

DISTINCT(열)

DISTINCT(테이블)

EXCEPT

FILTERS

GENERATE

GENERATEALL

GENERATESERIES

GROUPBY

IGNORE

INTERSECT

NATURALINNERJOIN

NATURALLEFTOUTERJOIN

ROLLUP

ROLLUPADDISBTOTAL

ROLLUPGROUP

ROLLUPISBTOTAL

ROW

SELECTCOLUMNS

SUBSTITUTEWITHINDEX

요약

SUMMARIZECOLUMNS

테이블 생성자

TOPN

TREATAS

UNION

Values

텍스트 함수

텍스트 함수 개요

COMBINEVALUES

CONCATENATE

CONCATENATEX

EXACT
FIND
FIXED
FORMAT
LEFT
LEN
LOWER
MID
REPLACE
REPT
RIGHT
SEARCH
SUBSTITUTE
TRIM
UNICHAR
UNICODE
UPPER
값

시간 인텔리전스 함수

시간 인텔리전스 함수 개요
CLOSINGBALANCEMONTH
CLOSINGBALANCEQUARTER
CLOSINGBALANCEYEAR
DATEADD
DATESBETWEEN
DATESINPERIOD
DATESMTD
DATESQTD
DATESYTD
ENDOFMONTH
ENDOFQUARTER
ENDOFYEAR

FIRSTDATE
FIRSTNONBLANK
FIRSTNONBLANKVALUE
LASTDATE
LASTNONBLANK
LASTNONBLANKVALUE
NEXTDAY
NEXTMONTH
NEXTQUARTER
NEXTYEAR
OPENINGBALANCEMONTH
OPENINGBALANCEQUARTER
OPENINGBALANCEYEAR
PARALLELPERIOD
PREVIOUSDAY
PREVIOUSMONTH
PREVIOUSQUARTER
PREVIOUSYEAR
SAMEPERIODLASTYEAR
STARTOFMONTH
STARTOFQUARTER
STARTOFYEAR
TOTALMTD
TOTALQTD
TOTALYTD

DAX 문

문 개요

정의

평가

ORDER BY

VAR

DAX 용어집

DAX 연산자

DAX 쿼리

DAX 매개 변수 명명 규칙

DAX 구문

DAX 개요

2021-03-16 • 92 minutes to read

DAX(Data Analysis Expressions)는 Analysis Services, Power BI 및 Excel의 파워 피벗에 사용되는 수식 식 언어입니다. DAX 수식에는 테이블 형식 데이터 모델의 관련 테이블 및 열에 있는 데이터에 대한 고급 계산과 쿼리를 수행하는 함수, 연산자 및 값이 포함됩니다.

이 문서는 DAX의 가장 중요한 개념에 대한 기본적인 소개만 제공합니다. DAX를 사용하는 모든 제품에 적용되는 DAX에 대해 설명합니다. 일부 기능은 특정 제품 또는 사용 사례에 적용되지 않을 수 있습니다. DAX의 특정 구현을 설명하는 제품 설명서를 참조하세요.

계산

DAX 수식은 측정값, 계산 열, 계산된 테이블 및 행 수준 보안에 사용됩니다.

측정값

측정값은 컨텍스트에 따라 결과가 변경되는 동적 계산 수식입니다. 측정값은 Power BI 보고서, Excel 피벗 테이블 또는 피벗 차트와 같은 여러 특성을 사용하여 모델 데이터의 결합 및 필터링을 지원하는 보고에 사용됩니다. 측정값은 모델 디자이너의 DAX 수식 입력줄을 사용하여 만듭니다.

측정값의 수식에는 COUNT 또는 SUM과 같은 자동 집계 기능을 사용하여 자동으로 만든 표준 집계 함수를 사용할 수도 있고 DAX 수식 입력줄을 사용하여 고유한 수식을 정의할 수도 있습니다. 명명된 측정값은 다른 측정값에 인수로 전달될 수 있습니다.

수식 입력줄에서 측정값에 대한 수식을 정의하는 경우 도구 설명 기능에 총 현재 컨텍스트에 대한 결과의 미리 보기가 표시되지만 결과가 어느 곳에도 즉시 출력되지는 않습니다. (필터링된) 계산 결과를 즉시 볼 수 없는 이유는 측정값의 결과가 컨텍스트 없이 결정될 수 없기 때문입니다. 측정값을 계산하려면 각 셀과 관련된 데이터를 검색한 다음 각 셀에 대한 식을 계산하는 데 필요한 컨텍스트를 제공할 수 있는 보고 클라이언트 애플리케이션이 있어야 합니다. 이 클라이언트는 SSMS(SQL Server Management Studio)의 DAX 쿼리에 있는 테이블 식, Excel 피벗 테이블 또는 피벗 차트, Power BI 보고서일 수 있습니다.

클라이언트에 관계 없이 결과의 각 셀에 대해 별도의 쿼리가 실행됩니다. 즉, 피벗 테이블에 있는 행과 열 머릿글의 각 조합 또는 Power BI 보고서에서 슬라이서와 필터의 각 선택은 측정값이 계산되는 다른 데이터의 하위 집합을 생성합니다. 예를 들어 다음과 같은 매우 간단한 측정값 수식을 사용합니다.

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

사용자가 보고서에 TotalSales 측정값을 배치한 다음, Product 테이블의 Product Category 열을 필터에 배치하면 Sales Amount 집계 계산이 이루어져 각 제품 범주에 대해 표시됩니다.

계산 열과 달리 측정값의 구문에는 수식 앞에 측정값 이름이 포함됩니다. 제공된 예제에서 **Total Sales** 라는 이름이 수식 앞에 표시됩니다. 측정값을 만든 후에는 이름 및 해당 정의가 보고 클라이언트 애플리케이션 필드 목록에 표시되며 관점 및 역할에 따라 모델의 모든 사용자가 사용할 수 있습니다.

자세한 내용은 다음을 참조하세요.

[Power BI Desktop의 측정값](#)

[Analysis Services의 측정값](#)

[파워 피벗의 측정값](#)

계산 열

계산 열은 모델 디자이너에서 기존 테이블에 추가한 다음, 열 값을 정의하는 DAX 수식을 만드는 열입니다. 계산

열에 올바른 DAX 수식이 포함되어 있으면 수식이 입력되자마자 각 행에 대한 값이 계산됩니다. 그런 다음, 값이 메모리 내 데이터 모델에 저장됩니다. 예를 들어 Date 테이블에서 수식 입력줄에 수식이 입력되는 경우입니다.

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

테이블의 각 행에 대한 값은 동일한 Date 테이블의 Calendar Year 열에서 값을 가져와서 공백과 대문자 Q를 추가한 다음, 동일한 Date 테이블의 Calendar Quarter 열의 값을 추가하여 계산됩니다. 계산 열의 각 행에 대한 결과가 즉시 계산되어 표시됩니다(예: 2017 Q1). 열 값은 테이블 또는 관련 테이블을 처리(새로 고침)하거나 모델을 메모리에서 언로드했다가 다시 로드하는 경우에만 다시 계산됩니다. 예를 들어 Power BI Desktop 파일을 닫았다가 다시 여는 경우에 다시 계산됩니다.

자세히 알아보려면 다음을 참조하세요.

[Power BI Desktop의 계산 열](#)

[Analysis Services의 계산 열](#)

[파워 피벗의 계산 열](#)

계산된 테이블

계산된 테이블은 수식 식을 기반으로 계산된 개체이며 동일한 모델에 있는 다른 테이블의 전체 또는 일부에서 파생됩니다. 데이터 원본에서 새 테이블의 열에 값을 쿼리하고 로드하는 대신, DAX 수식은 테이블의 값을 정의합니다.

계산된 테이블은 롤플레이팅 차원에 유용할 수 있습니다. 예를 들어 외래 키 관계에 따라 Date 테이블은 OrderDate, ShipDate 또는 DueDate입니다. ShipDate에 대해 계산된 테이블을 명시적으로 만들면 다른 테이블처럼 완벽하게 작동할 수 있는 독립형 테이블을 얻을 수 있습니다. 계산된 테이블은 필터링된 행 집합을 구성하는 경우 또는 기존의 다른 테이블에서 열의 하위 집합이나 상위 집합을 구성하는 경우에도 유용합니다. 이렇게 하면 특정 시나리오를 지원하기 위해 테이블 변형을 만드는 동안 원래 테이블을 그대로 유지할 수 있습니다.

계산된 테이블은 다른 테이블과의 관계를 지원합니다. 계산된 테이블의 열에는 데이터 형식, 서식이 있으며 데이터 범주에 속할 수 있습니다. 계산된 테이블은 다른 테이블과 마찬가지로 이름을 지정하고 표시하거나 숨길 수 있습니다. 계산된 테이블은 데이터를 가져오는 테이블 중 하나라도 새로 고쳐지거나 업데이트되면 다시 계산됩니다.

자세히 알아보려면 다음을 참조하세요.

[Power BI Desktop의 계산된 테이블](#)

[Analysis Services의 계산된 테이블](#)

행 수준 보안

행 수준 보안에서 DAX 수식은 부울 TRUE/FALSE 조건(특정 역할의 멤버가 쿼리 결과로 반환할 수 있는 행을 정의하는)으로 평가되어야 합니다. 예를 들어 Sales 역할 멤버의 경우 Customers 테이블에 DAX 수식이 다음과 같이 나타납니다.

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

Sales 역할의 멤버는 USA 고객 데이터만 볼 수 있고 SUM과 같은 집계는 USA 고객에 대해서만 반환됩니다. 행 수준 보안은 Excel의 파워 피벗에서 사용할 수 없습니다.

DAX 수식을 사용하여 행 수준 보안을 정의하는 경우, 허용된 행 집합을 만들게 됩니다. 이렇게 해도 다른 행에 대한 액세스를 거부하지 않으며, 허용된 행 집합의 일부로 반환되지만 않습니다. 다른 역할을 통해 DAX 수식에서 제외된 행에 대한 액세스를 허용할 수 있습니다. 사용자가 다른 역할의 멤버인 경우 이 역할의 행 수준 보안에서 특정 행 집합에 대한 액세스를 허용하면 사용자가 해당 행에 대한 데이터를 볼 수 있습니다.

행 수준 보안 수식은 지정된 행은 물론 관련된 행에도 적용됩니다. 테이블에 여러 관계가 있으면 필터는 활성 관계에 대한 보안을 적용합니다. 행 수준 보안 수식은 관련 테이블에 대해 정의된 다른 수식과 교차됩니다.

자세히 알아보려면 다음을 참조하세요.

쿼리

DAX 쿼리는 SSMS(SQL Server Management Studio) 및 DAX Studio(daxstudio.org)와 같은 오픈 소스 도구에서 만들고 실행할 수 있습니다. 테이블 형식 데이터 모델에서만 만들 수 있는 DAX 계산 수식과 달리, DAX 쿼리는 Analysis Services 다차원 모델에 대해서도 실행할 수 있습니다. DAX 쿼리는 MDX(Multidimensional Data Expressions) 쿼리보다 더 쉽게 작성하고 효율적으로 사용할 수 있는 경우가 많습니다.

DAX 쿼리는 T-SQL의 SELECT 문과 유사한 명령문입니다. 가장 기본적인 유형의 DAX 쿼리는 *evaluate* 문입니다. 예제:

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

SafetyStockLevel이 200 미만인 제품만 EnglishProductName을 기준으로 오름차순으로 나열된 테이블을 결과로 반환합니다.

측정값은 쿼리의 일부로 만들 수 있습니다. 측정값은 쿼리 기간 동안만 존재합니다. 자세히 알아보려면 [DAX 쿼리](#)를 참조하세요.

수식

DAX 수식은 계산 열 및 측정값에서 계산을 만들고 행 수준 보안을 사용하여 데이터를 보호하는 데 필수적입니다. 계산 열 및 측정값에 대한 수식을 만들려면 모델 디자이너 창이나 DAX 편집기의 맨 위에 있는 수식 입력줄을 사용합니다. 행 수준 보안에 대한 수식을 만들려면 역할 관리자 또는 역할 관리 대화 상자를 사용합니다. 이 섹션에서는 DAX 수식의 기본 사항을 이해하는 데 활용할 수 있는 정보를 제공합니다.

수식 기본 사항

DAX 수식은 매우 간단하거나 아주 복잡할 수 있습니다. 다음 표에는 계산 열에 사용할 수 있는 간단한 수식의 예가 몇 가지 있습니다.

FORMULA	정의
<code>= TODAY()</code>	계산 열의 모든 행에 오늘 날짜를 삽입합니다.
<code>= 3</code>	계산 열의 모든 행에 값 3을 삽입합니다.
<code>= [Column1] + [Column2]</code>	[Column1]과 [Column2]의 같은 행에 값을 추가하고, 동일한 행의 계산 열에 값을 넣습니다.

만드는 수식이 간단하든 복잡하든 관계없이 수식을 작성할 때는 다음 단계를 사용할 수 있습니다.

1. 각 수식은 등호(=)로 시작해야 합니다.
2. 함수 이름을 입력 또는 선택하거나 식을 입력할 수 있습니다.
3. 원하는 함수 또는 이름의 처음 몇 글자를 입력하면 사용 가능한 함수, 테이블 및 열 목록이 자동 완성 기능을 통해 표시됩니다. TAB 키를 눌러서 자동 완성 목록에 있는 항목을 수식에 추가할 수 있습니다.

Fx 단추를 클릭하여 사용 가능한 함수 목록을 표시할 수도 있습니다. 드롭다운 목록에서 함수를 선택하려면 화살표 키를 사용하여 항목을 강조 표시하고 **확인** 을 클릭하여 수식에 함수를 추가합니다.

4. 가능한 테이블 및 열의 드롭다운 목록에서 인수를 선택하거나 값을 입력하여 함수에 인수를 제공합니다.

5. 구문 오류가 없는지 확인하고, 모든 괄호가 닫혀 있고 열, 테이블 및 값이 올바르게 참조되는지 확인합니다.

6. ENTER 키를 눌러 수식을 적용합니다.

NOTE

계산 열에 수식을 입력하는 즉시 수식의 유효성이 검사되고 열에 값이 채워집니다. 측정값에서 ENTER 키를 누르면 테이블과 함께 측정값 정의가 저장됩니다. 수식이 잘못된 경우에는 오류가 표시됩니다.

이 예에서는 **Days in Current Quarter** 라는 측정값의 수식을 살펴보겠습니다.

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

이 측정값은 불완전한 기간과 이전 기간 사이의 비교 비율을 만드는 데 사용됩니다. 수식은 경과된 기간의 비율을 고려하여 이전 기간의 동일한 규모와 비교해야 합니다. 이 경우 [Days Current Quarter to Date]/[Days in Current Quarter]는 현재 기간에서 경과된 비율을 제공합니다.

이 수식에는 다음과 같은 요소가 포함되어 있습니다.

수식 요소	설명
Days in Current Quarter	측정값의 이름입니다.
=	등호(=)는 수식을 시작합니다.
COUNTROWS	COUNTROWS는 Date 테이블의 행 수를 계산합니다.
()	열기 및 닫기 괄호는 인수를 지정합니다.
DATESBETWEEN	DATESBETWEEN 함수는 Date 테이블의 Date 열에 있는 각 값의 마지막 날짜 사이의 일 수를 반환합니다.
'Date'	Date 테이블을 지정합니다. 테이블은 작은따옴표로 묶여 있습니다.
[Date]	Date 테이블의 Date 열을 지정합니다. 열은 대괄호로 묶여 있습니다.
,	
STARTOFQUARTER	STARTOFQUARTER 함수는 분기의 시작 날짜를 반환합니다.
LASTDATE	LASTDATE 함수는 분기의 마지막 날짜를 반환합니다.
'Date'	Date 테이블을 지정합니다.
[Date]	Date 테이블의 Date 열을 지정합니다.
,	

수식 요소	설명
<code>ENDOFQUARTER</code>	ENDOFQUARTER 함수
<code>'Date'</code>	Date 테이블을 지정합니다.
<code>[Date]</code>	Date 테이블의 Date 열을 지정합니다.

수식 자동 완성 사용

자동 완성을 사용하면 수식의 각 요소에 대한 옵션이 제공되므로 올바른 수식 구문을 쉽게 입력할 수 있습니다.

- 중첩된 함수가 있는 기존 수식의 중간에 수식 자동 완성 기능을 사용할 수 있습니다. 삽입 지점 바로 전 텍스트는 드롭다운 목록의 값을 표시하는 데 사용되며 삽입 지점 이후의 모든 텍스트는 변경되지 않은 상태로 유지됩니다.
- 자동 완성 기능은 함수의 닫는 괄호를 추가하거나 괄호를 자동으로 일치시키지 않습니다. 각 함수의 구문이 올바른지 확인해야 합니다. 구문이 올바르지 않으면 수식을 저장하거나 사용할 수 없습니다.

수식에 여러 함수 사용

한 함수의 결과를 다른 함수의 인수로 사용하도록 함수를 중첩할 수 있습니다. 계산 열에서 함수의 64 수준까지 중첩할 수 있습니다. 그러나 함수를 중첩하면 수식을 만들거나 수식의 문제를 해결하기가 어려워질 수 있습니다. 여러 함수가 중첩 함수로만 사용되도록 설계되었습니다. 이러한 함수는 테이블을 반환합니다. 테이블은 결과로 직접 저장할 수 없으며 테이블 함수에 대한 입력으로 제공해야 합니다. 예를 들어 SUMX, AVERAGEX 및 MINX 함수의 경우 첫 번째 인수로 테이블을 사용해야 합니다.

Functions

함수는 식 내의 명명된 수식입니다. 대부분의 함수는 매개 변수라고도 하는 필수 및 선택적 인수를 입력으로 사용합니다. 함수가 실행되면 값이 반환됩니다. DAX에는 날짜와 시간을 사용하여 계산을 수행하고, 조건부 값을 만들고, 문자열을 사용하고, 관계를 기반으로 조회를 수행하는 데 사용할 수 있는 함수와 재귀적 계산을 수행하기 위해 테이블에서 반복하는 기능이 포함되어 있습니다. Excel 수식에 익숙한 경우 많은 함수가 비슷해 보일 것입니다. DAX 수식에는 다음과 같은 중요한 차이가 있습니다.

- DAX 함수는 항상 전체 열 또는 테이블을 참조합니다. 테이블 또는 열의 특정 값만 사용하려면 수식에 필터를 추가하면 됩니다.
- 행 단위로 계산을 사용자 지정해야 하는 경우 DAX에서 제공하는 함수를 통해 현재 행 값이나 관련 값을 일종의 매개 변수로 사용하여 컨텍스트에 따라 달라지는 계산을 수행할 수 있습니다. 이러한 함수의 작동 방식을 이해하려면 이 문서에서 [컨텍스트](#)를 참조하세요.
- DAX에는 값보다는 테이블을 반환하는 많은 함수가 포함되어 있습니다. 테이블은 보고 클라이언트에서 표시되지 않지만 다른 함수에 대한 입력을 제공하는 데 사용됩니다. 예를 들어 테이블을 검색한 다음 테이블에 있는 고유 값을 세거나 필터링된 테이블 또는 열에 대해 동적 합계를 계산할 수 있습니다.
- DAX 함수에는 다양한 *시간 인텔리전스* 함수가 포함되어 있습니다. 이러한 함수를 통해 날짜 범위를 정의하거나 선택하고 이러한 날짜 또는 범위를 기반으로 동적 계산을 수행할 수 있습니다. 예를 들어 병렬 기간에 대한 합계를 비교할 수 있습니다.

날짜 및 시간 함수

DAX의 날짜 및 시간 함수는 Microsoft Excel의 날짜 및 시간 함수와 비슷합니다. 그러나 DAX 함수는 1900년 3월 1일에 시작되는 **datetime** 데이터 형식을 기반으로 합니다. 자세히 알아보려면 [날짜 및 시간 함수](#)를 참조하세요.

필터 함수

DAX의 필터 함수를 사용하면 특정 데이터 형식을 반환하고, 관련 테이블의 값을 조회하고, 관련 값을 기준으로

필터링할 수 있습니다. 조회 함수는 데이터베이스와 같이 테이블 및 관계를 사용하여 작동됩니다. 필터링 함수를 사용하면 데이터 컨텍스트를 조작하여 동적 계산을 만들 수 있습니다. 자세히 알아보려면 [필터 함수](#)를 참조하세요.

재무 함수

DAX의 재무 함수는 순 현재 가치 및 수익률과 같은 재무 계산을 수행하는 수식에 사용됩니다. 이러한 함수는 Microsoft Excel에서 사용되는 재무 함수와 유사합니다. 자세히 알아보려면 [재무 함수](#)를 참조하세요.

정보 함수

정보 함수는 인수로 제공되는 셀 또는 행을 살펴보고 값이 필요한 형식과 일치하는지 여부를 알려 줍니다. 예를 들어 ISERROR 함수는 참조하는 값에 오류가 포함되어 있는 경우 TRUE를 반환합니다. 자세히 알아보려면 [정보 함수](#)를 참조하세요.

논리 함수

논리 함수는 식에 대해 실행되어 식의 값에 대한 정보를 반환합니다. 예를 들어 TRUE 함수를 사용하면 평가 중인 식이 TRUE 값을 반환하는지 여부를 알 수 있습니다. 자세히 알아보려면 [논리 함수](#)를 참조하세요.

수치 연산 및 삼각 함수

DAX의 수치 연산 함수는 Excel 수치 연산 및 삼각 함수와 매우 비슷합니다. DAX 함수에서 사용하는 숫자 데이터 형식에는 사소한 몇 가지 차이점이 있습니다. 자세히 알아보려면 [수학 및 삼각 함수](#)를 참조하세요.

기타 함수

이러한 함수는 대부분의 다른 함수가 속하는 범주로 정의할 수 없는 고유한 작업을 수행합니다. 자세히 알아보려면 [기타 함수](#)를 참조하세요.

관계 함수

DAX의 관계 함수를 사용하면 다른 관련 테이블의 값을 반환하고, 식에 사용할 특정 관계를 지정하고, 교차 필터링 방향을 지정할 수 있습니다. 자세한 내용은 [관계 함수](#)를 참조하세요.

통계 함수

DAX는 집계를 수행하는 통계 함수를 제공 합니다. DAX에서는 집계와 평균을 만들거나 최소값과 최대값을 구할 수 있을 뿐 아니라 관련 테이블을 기준으로 집계를 수행하거나 만들기 전에 열을 필터링할 수도 있습니다. 자세히 알아보려면 [통계 함수](#)를 참조하세요.

텍스트 함수

DAX의 텍스트 함수는 Excel의 텍스트 함수와 매우 유사합니다. 문자열의 일부를 반환하거나, 문자열 내의 텍스트를 검색하거나, 문자열 값을 연결할 수 있습니다. DAX에서는 날짜, 시간 및 숫자의 형식을 제어하기 위한 함수도 제공합니다. 자세히 알아보려면 [텍스트 함수](#)를 참조하세요.

시간 인텔리전스 함수

DAX에 제공되는 시간 인텔리전스 함수를 사용하면 달력과 날짜에 대한 기본 제공 정보를 사용하는 계산을 만들 수 있습니다. 집계 또는 계산과 함께 시간 및 날짜 범위를 사용하여 판매, 재고 등에 대해 의미 있는 기간별 비교를 빌드할 수 있습니다. 자세히 알아보려면 [시간 인텔리전스 함수\(DAX\)](#)를 참조하세요.

테이블 조작 함수

테이블 조작 함수는 테이블을 반환하거나 기존 테이블을 조작합니다. 예를 들어 ADDCOLUMNS를 사용하여 지정된 테이블에 계산 열을 추가하거나 SUMMARIZECOLUMNS 함수를 사용하여 일련의 그룹에 대한 요약 테이블을 반환할 수 있습니다. 자세히 알아보려면 [테이블 조작 함수](#)를 참조하세요.

변수

[VAR](#)을 사용하여 식 내에 변수를 만들 수 있습니다. VAR은 기술적으로 함수가 아니라 식의 결과를 명명된 변수로 저장하는 데 사용하는 키워드입니다. 그런 다음, 이 변수를 다른 측정값 식에 인수로 전달할 수 있습니다. 예를 들면 다음과 같습니다.

```

VAR
    TotalQty = SUM ( Sales[Quantity] )

Return

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

```

이 예제에서는 TotalQty를 다른 식에 명명된 변수로 전달할 수 있습니다. 변수는 테이블을 비롯한 모든 스칼라 데이터 형식일 수 있습니다. DAX 수식에 변수를 사용하면 매우 유용할 수 있습니다.

데이터 형식

다양한 데이터 형식을 지원하는 여러 데이터 원본의 데이터를 모델로 가져올 수 있습니다. 데이터를 모델로 가져올 때 데이터는 테이블 형식 모델 데이터 형식 중 하나로 변환됩니다. 계산에서 모델 데이터를 사용하면 해당 데이터는 계산 출력 및 지속을 위해 DAX 데이터 형식으로 변환됩니다. DAX 수식을 만들면 수식에 사용된 조건이 자동으로 반환되는 값 데이터 형식을 결정합니다.

DAX는 다음 데이터 형식을 지원합니다.

모델의 데이터 형식	DAX의 데이터 형식	설명
정수	64비트(8바이트) 정수 값 ^{1, 2}	소수 자릿수가 없는 숫자입니다. 정수는 양수나 음수가 될 수 있지만 -9,223,372,036,854,775,808(-2 ⁶³)부터 9,223,372,036,854,775,807(2 ⁶³ -1) 사이의 정수여야 합니다.
10진수	64비트(8바이트) 실수 ^{1, 2}	<p>실수는 소수 자리를 가질 수 있는 수입니다. 실수는 광범위한 값을 포함합니다.</p> <p>-1.79E +308에서 -2.23E -308 사이의 음수 값</p> <p>0</p> <p>2.23E -308에서 1.79E + 308 사이의 양수 값</p> <p>하지만 유효 숫자 수는 열일곱 자릿수로 제한됩니다.</p>
Boolean	Boolean	True 또는 False 값입니다.
텍스트	문자열	유니코드 문자 데이터 문자열입니다. 텍스트 형식으로 표시된 문자열, 숫자 또는 날짜가 될 수 있습니다.
날짜	날짜/시간	<p>허용되는 날짜-시간 표현의 날짜 및 시간</p> <p>유효한 날짜는 1900년 3월 1일 이후의 모든 날짜입니다.</p>

모델의 데이터 형식	DAX의 데이터 형식	설명
통화	통화	통화 데이터 형식에는 - 922,337,203,685,477.5808부터 922,337,203,685,477.5807 사이의 소수 자릿수가 고정된 네 자릿수를 사용할 수 있습니다.
해당 없음	비어 있음	공백은 SQL null을 나타내거나 대체하는 DAX의 데이터 형식입니다. BLANK 함수를 사용하여 공백을 만들고 논리 함수 ISBLANK를 사용하여 공백을 테스트할 수 있습니다.

테이블 형식 데이터 모델에는 많은 DAX 함수에 대한 입력 또는 출력으로 *테이블* 데이터 형식도 포함됩니다. 예를 들어 FILTER 함수는 테이블을 입력으로 사용하여 필터 조건을 충족하는 행만 포함하는 다른 테이블을 출력합니다. 집계 함수와 테이블 함수를 결합하면 동적으로 정의된 데이터 집합을 대상으로 복잡한 계산을 수행할 수 있습니다.

데이터 형식은 대개 자동으로 설정되지만, 데이터 형식과 해당 형식이 DAX 수식에 적용되는 방식을 이해하는 것이 중요합니다. 수식의 오류나 예기치 못한 결과는 인수에 지정된 데이터 형식에 사용할 수 없는 특정 연산자를 사용하여 발생하는 경우가 많습니다. 예를 들어 수식 `= 1 & 2` 는 결과로 문자열 12를 반환합니다. 하지만 수식 `= "1" + "2"` 는 정수 3을 반환합니다.

Context

컨텍스트는 DAX 수식을 만들 때 이해해야 하는 중요한 개념입니다. 컨텍스트를 사용하면 수식의 결과가 현재 행 또는 셀 선택과 모든 관련 데이터에 맞게 변경될 수 있는 동적 분석을 수행할 수 있습니다. 우수한 성능의 동적 분석을 작성하고 수식 문제를 해결하려면 컨텍스트를 이해하고 효과적으로 사용해야 합니다.

테이블 형식 모델의 수식은 다음과 같은 디자인 요소에 따라 각기 다른 컨텍스트에서 계산될 수 있습니다.

- 피벗 테이블 또는 보고서에서 적용된 필터
- 수식 내에 정의된 필터
- 수식 내에서 특수 함수를 사용하여 지정된 관계

컨텍스트의 유형으로는 *행 컨텍스트*, *쿼리 컨텍스트* 및 *필터 컨텍스트*가 있습니다.

행 컨텍스트

*행 컨텍스트*는 "현재 행"으로 생각할 수 있습니다. 계산 열에서 수식을 만드는 경우 해당 수식의 행 컨텍스트에 현재 행에 있는 모든 열의 값이 포함됩니다. 테이블이 다른 테이블과 관련되어 있으면 다른 테이블에서 현재 행과 관련된 모든 값도 내용에 포함됩니다.

예를 들어 Freight와 Tax라는 두 열의 값을 더하는 `= [Freight] + [Tax]` 계산 열을 만든다고 가정합니다. 이 수식은 자동으로 지정된 열에 있는 현재 행의 값만 가져옵니다.

또한 행 컨텍스트는 DAX 수식을 사용하여 계산 열에 정의된 관계를 비롯하여 테이블 간에 정의된 모든 관계를 따라 현재 행과 연결된 관련 테이블의 행을 확인합니다.

예를 들어 다음은 RELATED 함수를 사용하여 관련 테이블에서 주문이 발송된 지역을 기준으로 세금 값을 가져오는 수식입니다. 세금 값은 현재 테이블의 지역 값을 사용하고 관련 테이블에서 지역을 조회한 다음 관련 테이블에서 해당 지역의 세율을 가져와 결정됩니다.

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

이 수식은 Region 테이블에서 현재 지역의 세율을 가져와서 Freight 열의 값에 더합니다. DAX 수식에서 테이블

을 연결하는 특정 관계를 알거나 지정할 필요는 없습니다.

여러 행 컨텍스트

DAX에는 테이블에서 계산을 반복하는 함수가 포함되어 있습니다. 이러한 함수에는 각각 고유한 행 컨텍스트가 있는 여러 현재 행이 포함될 수 있습니다. 기본적으로 이러한 함수를 사용하여 내부 및 외부 루프에서 재귀적으로 연산을 수행하는 수식을 만들 수 있습니다.

예를 들어 모델에 **Products** 테이블과 **Sales** 테이블이 있다고 가정합니다. 사용자는 여러 제품과 관련된 트랜잭션으로 꼭 찬 판매량 테이블 전체를 살펴보고 한 트랜잭션에서 각 제품에 대해 주문된 수량 중 가장 많은 수량을 찾을 수 있습니다.

DAX에서는 올바른 값을 반환하는 단일 수식을 작성할 수 있으며, 사용자가 테이블에 데이터를 추가할 때마다 결과가 자동으로 업데이트됩니다.

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

이 수식의 자세한 예는 [EARLIER](#)를 참조하세요.

요약하자면 **EARLIER** 함수는 현재 연산보다 먼저 발생한 연산의 행 컨텍스트를 저장합니다. 이 함수는 항상 두 개의 컨텍스트 집합을 메모리에 저장합니다. 하나의 컨텍스트 집합은 수식의 내부 루프에 대한 현재 행을 나타내고, 다른 컨텍스트 집합은 수식의 외부 루프에 대한 현재 행을 나타냅니다. DAX는 두 루프 간에 값을 자동으로 공급하므로 복잡한 집계를 만들 수 있습니다.

쿼리 컨텍스트

*쿼리 컨텍스트*는 수식에 대해 암시적으로 검색되는 데이터의 하위 집합을 참조합니다. 예를 들어 사용자가 측정값 또는 필드를 보고서에 배치하면 엔진은 행 및 열 머리글, 슬라이서 및 보고서 필터를 검사하여 컨텍스트를 파악합니다. 그런 다음 모델 데이터에 대해 필요한 쿼리를 실행하여 올바른 일부 데이터를 가져오고, 수식으로 정의된 계산을 수행하며, 보고서의 값을 채웁니다.

컨텍스트는 수식을 배치하는 위치에 따라 변경되기 때문에 수식의 결과도 변경될 수 있습니다. 예를 들어 **Sales** 테이블의 **Profit** 열에 있는 값의 합계를 계산하는 수식인 `= SUM('Sales'[Profit])`을 만든다고 가정합니다.

Sales 테이블 내의 계산 열에 이 수식을 사용하면, 수식의 쿼리 컨텍스트가 항상 **Sales** 테이블의 전체 데이터 세트이기 때문에 수식에 대한 결과가 전체 테이블에서 동일합니다. 결과에는 모든 지역, 모든 제품, 모든 연도 등의 수익이 포함됩니다.

하지만, 일반적으로 동일한 결과를 수백 번 보려는 사용자는 없습니다. 그 대신 특정 연도, 특정 국가, 특정 제품 또는 이러한 항목의 조합에 대한 수익을 구한 후 총계를 보려고 합니다.

필드를 필터링, 추가 또는 제거하고 슬라이서를 사용하여 보고서에서 컨텍스트를 변경합니다. 각 변경 사항에 대해 쿼리 컨텍스트에서 측정값이 평가됩니다. 따라서 측정값에서 사용되는 동일한 수식은 셀에 따라 다른 *쿼리 컨텍스트*에서 계산됩니다.

필터 컨텍스트

*필터 컨텍스트*는 관련 테이블에서 검색된 값이나 각 열에서 허용되는 값의 집합입니다. 디자이너 또는 표시 계층(보고서 및 피벗 테이블)에서 열에 필터를 적용할 수 있습니다. 수식 내의 필터 식으로 필터를 명시적으로 정의할 수도 있습니다.

필터 컨텍스트는 수식에 인수를 사용하여 열이나 테이블에 허용되는 값 집합에 대한 필터 제약 조건을 지정하면 추가됩니다. 필터 컨텍스트는 행 컨텍스트 또는 쿼리 컨텍스트와 같은 다른 컨텍스트보다 먼저 적용됩니다.

테이블 형식 모델에서는 필터 컨텍스트를 만드는 다양한 방법이 있습니다. Power BI 보고서와 같이 모델을 사용할 수 있는 클라이언트의 컨텍스트 내에서 사용자는 행과 열 머리글에 슬라이서나 보고서 필터를 추가하여 즉석에서 필터를 만들 수 있습니다. 수식에서 직접 필터 식을 지정하여 관련 값을 지정하거나 입력으로 사용되는 테이블을 필터링하거나 계산에서 사용되는 값의 컨텍스트를 동적으로 가져올 수도 있습니다. 특정 열의 필터를 완전히 지우거나 선택적으로 지울 수도 있습니다. 이는 총합계를 계산하는 수식을 만들 때 매우 유용합니다.

수식 내에서 필터를 만드는 방법을 자세히 알아보려면 [FILTER 함수\(DAX\)](#)를 참조하세요.

필터를 지워서 총합계를 구하는 방법의 예는 [ALL 함수\(DAX\)](#)를 참조하세요.

수식 내에서 필터를 선택적으로 지우고 적용하는 방법에 대한 예는 [ALLEXCEPT](#)를 참조하세요.

수식의 컨텍스트 확인

DAX 수식을 만들 때 수식은 먼저 유효한 구문인지 테스트된 다음 수식에 포함된 열과 테이블의 이름을 현재 컨텍스트에서 찾을 수 있는지 테스트됩니다. 수식으로 지정된 열이나 테이블을 찾을 수 없는 경우 오류가 반환됩니다.

유효성 검사(및 재계산 작업) 중의 컨텍스트는 이전 섹션에 설명된 대로 모델에서 사용 가능한 테이블, 테이블 간의 관계 및 적용된 필터를 사용하여 결정됩니다.

예를 들어 다른 테이블과 관련되지 않은 일부 데이터를 새 테이블로 가져오고 필터를 적용하지 않은 경우 *현재 컨텍스트*는 테이블의 전체 열 집합입니다. 테이블이 다른 테이블과 관계로 연결되어 있으면 현재 컨텍스트에 관련 테이블이 포함됩니다. 테이블의 열을 슬라이서(및 경우에 따라 보고서 필터)가 있는 보고서에 추가하는 경우 수식의 컨텍스트는 보고서의 각 셀에 있는 데이터의 하위 집합입니다.

컨텍스트는 수식의 문제 해결을 어렵게 할 수도 있는 강력한 개념입니다. 간단한 수식 및 관계에서 시작하여 컨텍스트가 어떻게 작동하는지 확인해 보는 것이 좋습니다. 다음 섹션에서는 수식이 여러 유형의 컨텍스트를 사용하여 결과를 동적으로 반환하는 방식의 몇 가지 예를 제공합니다.

연산자

DAX 언어는 수식에서 다음과 같은 네 가지 유형의 계산 연산자를 사용합니다.

- 값을 비교하고 논리적 TRUE\FALSE 값을 반환하는 비교 연산자
- 숫자 값을 반환하는 산술 계산을 수행하는 산술 연산자
- 둘 이상의 텍스트 문자열을 결합하는 텍스트 연결 연산자
- 둘 이상의 식을 결합하여 단일 결과를 반환하는 논리 연산자

DAX 수식에 사용되는 연산자에 대한 자세한 내용은 [DAX 연산자](#)를 참조하세요.

테이블 및 열 작업

테이블 형식 데이터 모델의 테이블은 Excel 테이블처럼 보이지만 데이터와 수식 작업을 수행하는 방식이 다릅니다.

- 수식은 개별 셀, 범위 참조 또는 배열에서는 작동하지 않으며 테이블과 열에서만 작동합니다.
- 수식에서는 관계를 사용하여 관련 테이블에서 값을 가져올 수 있습니다. 검색되는 값은 항상 현재 행 값과 관련된 값입니다.
- Excel 워크시트에서처럼 불규칙하거나 "균일하지 않은" 데이터는 사용할 수 없습니다. 테이블의 각 행에는 같은 수의 열이 있어야 합니다. 그러나 일부 열에는 빈 값을 포함할 수 있습니다. Excel 데이터 테이블과 테이블 형식 모델 데이터 테이블은 서로 바꿔 사용할 수 없습니다.
- 각 열에 대해 데이터 형식이 설정되어 있지만 해당 열에 있는 각 값의 형식은 같아야 합니다.

수식에서 테이블 및 열 참조

이름을 사용하여 모든 테이블과 열을 참조할 수 있습니다. 예를 들어 다음의 수식은 *정규화된* 이름을 사용하여 두 개의 테이블에서 열을 참조하는 방법을 보여 줍니다.

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

수식을 계산할 때 모델 디자이너는 먼저 일반 구문을 확인한 다음 현재 컨텍스트에서 가능한 열과 테이블을 기준으로 제공된 열 및 테이블의 이름을 확인합니다. 이름이 모호하거나 열 또는 테이블을 찾을 수 없는 경우 수식

에서 오류가 발생하여 오류가 발생한 셀에 데이터 값 대신 #ERROR 문자열이 표시됩니다. 테이블, 열 및 기타 개체의 명명 요구 사항에 대해 자세히 알아보려면 [DAX 구문의 명명 요구 사항을 참조하세요](#).

테이블 관계

테이블 간의 관계를 만들면 다른 테이블의 관련 값을 계산에 사용할 수 있습니다. 예를 들어 계산 열을 사용하여 현재 재판매인과 관련된 모든 배송 레코드를 파악한 다음, 각각에 대한 배송 비용의 합계를 계산할 수 있습니다. 그러나 대부분의 경우 관계가 필요하지 않을 수 있습니다. 수식에 [LOOKUPVALUE](#) 함수를 사용하여 *search_column*과 *search_value* 인수에 지정된 조건을 충족하는 행에 대해 *result_columnName*의 값을 반환할 수 있습니다.

대부분의 DAX 함수에서 참조한 열을 찾고 의미 있는 결과를 반환하기 위해서는 두 테이블 간이나 여러 테이블 간에 관계가 있어야 합니다. 다른 함수에서는 관계를 확인하려고 시도하지만 최상의 결과를 얻기 위해서는 가능한 항상 관계를 만들어야 합니다. 테이블 형식 데이터 모델은 테이블 간의 여러 관계를 지원합니다. 혼동이나 잘못된 결과를 방지하기 위해 한 번에 하나의 관계만 활성 관계로 지정되지만, 계산 시 데이터의 다른 연결을 트래버스하기 위해 필요에 따라 활성 관계를 변경할 수 있습니다. [USERELATIONSHIP](#) 함수를 사용하면 특정 계산에 사용할 관계를 하나 이상 지정할 수 있습니다.

관계를 사용할 때는 다음과 같은 수식 디자인 규칙을 준수하는 것이 중요합니다.

- 테이블이 관계로 연결되는 경우에는 키로 사용된 두 열이 일치하는 값을 포함하도록 해야 합니다. 참조 무결성이 강제로 적용되지 않기 때문에, 키 열에 일치하지 않는 값을 포함할 수 있고 여전히 관계를 만들 수 있습니다. 이 경우 빈 값이나 일치하지 않는 값이 있으면 수식 결과에 영향을 줄 수 있다는 점을 명심해야 합니다.
- 관계를 사용하여 모델에서 테이블을 연결하면 수식이 계산되는 범위, 즉 *컨텍스트*가 확장됩니다. 새 테이블 또는 새 관계의 추가나 활성 관계의 변경으로 컨텍스트가 변경되면 예기치 않은 방식으로 결과가 변경될 수 있습니다. 자세히 알아보려면 이 문서의 [컨텍스트](#)를 참조하세요.

처리 및 새로 고침

처리(*process*) 및 다시 계산(*recalculation*)은 별도의 작업이지만 관련된 작업입니다. 복잡한 수식, 대용량 데이터 또는 외부 데이터 원본에서 가져온 데이터를 포함하는 데이터 모델을 디자인할 때 이러한 개념을 완벽하게 이해해야 합니다.

Process (refresh)(처리(새로 고침))는 모델의 데이터를 외부 데이터 원본의 새 데이터로 업데이트합니다.

*다시 계산*은 수식 변경 및 기본 데이터의 변경 내용을 반영하도록 수식의 결과를 업데이트하는 프로세스입니다. 다시 계산은 다음과 같은 방식으로 성능에 영향을 줄 수 있습니다.

- 계산 열의 값이 계산된 후 모델에 저장됩니다. 계산 열의 값을 업데이트하려면 세 가지 처리 명령인 전체 처리, 데이터 처리 및 다시 계산 처리 중 하나를 사용하여 모델을 처리해야 합니다. 식을 변경할 때마다 항상 전체 열에 대해 수식의 결과를 다시 계산해야 합니다.
- 측정값으로 계산되는 값은 사용자가 피벗 테이블에 측정값을 추가하거나 보고서를 열 때마다 동적으로 평가됩니다. 사용자가 컨텍스트를 수정하면 측정값에 의해 반환되는 값이 변경됩니다. 측정값 결과는 항상 메모리 내 캐시의 최신값을 반영합니다.

다시 계산 결과가 다른 값을 반환하여 역할 멤버가 행 쿼리를 가능하거나 불가능하게 만들지 않는 한, 처리와 다시 계산은 행 수준 보안 수식에 영향을 주지 않습니다.

Updates

DAX는 지속적으로 개선되고 있습니다. [새로운 함수 및 업데이트된 함수](#)는 사용 가능한 다음 업데이트(일반적으로 매달)에서 릴리스됩니다. 서비스가 먼저 업데이트된 다음 Power BI Desktop, Excel, SSMS(SQL Server Management Studio), Visual Studio용 Analysis Services 프로젝트 확장(SSDT) 같은 설치된 애플리케이션이 업데이트됩니다. SQL Server Analysis Services는 다음 누적 업데이트에서 업데이트됩니다. 새 함수는 Power BI Desktop 업데이트와 일치하는 DAX 함수 참조에서 먼저 선언되고 설명됩니다.

모든 함수가 이전 버전의 SQL Server Analysis Services 및 Excel에서 지원되는 것은 아닙니다.

문제 해결

수식을 정의할 때 오류가 발생하면 수식에 *구문 오류*, *의미 체계 오류* 또는 *계산 오류*가 있는 경우일 수 있습니다.

구문 오류는 해결하기가 쉽습니다. 대개는 괄호나 쉼표가 누락된 경우가 많습니다.

구문이 올바르지만 참조되는 열이나 값이 수식의 컨텍스트에 적합하지 않으면 다른 유형의 오류가 발생합니다. 다음과 같은 문제 때문에 이러한 의미 체계 오류와 계산 오류가 발생할 수 있습니다.

- 수식에서 존재하지 않는 열, 테이블 또는 함수를 참조합니다.
- 수식이 올바른 것 같지만 데이터 엔진이 데이터를 가져올 때 형식 불일치를 발견하여 오류가 발생합니다.
- 수식이 함수에 잘못된 수 또는 유형의 인수를 전달합니다.
- 수식에서 오류가 있는 다른 열을 참조하기 때문에 해당 값이 올바르지 않습니다.
- 수식은 처리되지 않은 열을 참조하므로 메타데이터는 있지만 계산에 사용할 실제 데이터가 없습니다.

처음 네 경우에서 DAX는 잘못된 수식이 포함된 열 전체에 플래그를 지정합니다. 마지막 경우 DAX는 열이 처리되지 않은 상태임을 나타내기 위해 열을 회색으로 표시합니다.

앱 및 도구

Power BI Desktop



[Power BI Desktop](#)는 무료 데이터 모델링 및 보고 애플리케이션입니다. 모델 디자이너에는 DAX 계산 수식을 작성하는 DAX 편집기가 포함되어 있습니다.

Excel의 파워 피벗



[Excel의 파워 피벗](#) 모델 디자이너에는 DAX 계산 수식을 작성하는 DAX 편집기가 포함되어 있습니다.

Visual Studio



Analysis Services 모델 프로젝트를 만들려면 [Analysis Services 프로젝트](#) 확장(VSIX)과 함께 Visual Studio를 사용합니다. 프로젝트 확장과 함께 설치된 테이블 형식 모델 디자이너에는 DAX 편집기가 포함되어 있습니다.

SQL Server Management Studio



SSMS([SQL Server Management Studio](#))는 Analysis Services 작업을 수행하기 위한 필수 도구입니다. SSMS에는 테이블 형식 모델과 다차원 모델을 쿼리하는 DAX 쿼리 편집기가 포함되어 있습니다.

DAX Studio



DAX Studio는 Excel 모델에서 Analysis Services, Power BI Desktop 및 파워 피벗에 대해 DAX 쿼리를 만들고 실행할 수 있는 오픈 소스 클라이언트 도구입니다.

테이블 형식 편집기



테이블 형식 편집기는 테이블 형식 모델 메타데이터의 모든 개체에 대해 직관적이고 계층적인 관점을 제공하는 오픈 소스 도구입니다. 테이블 형식 편집기에는 측정값, 계산 열, 계산된 테이블 식을 쉽게 편집하는 것을 지원하는 구문 강조 표시 기능을 가진 DAX 편집기가 포함되어 있습니다.

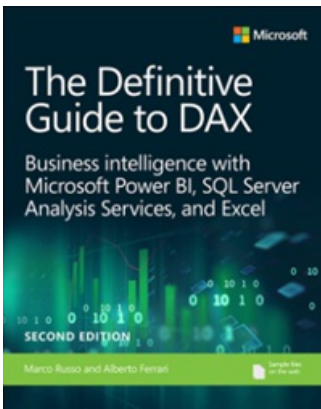
학습 리소스

DAX를 학습할 때는 사용자의 애플리케이션을 사용하여 데이터 모델을 만드는 것이 가장 좋습니다. Excel의 Analysis Services, Power BI Desktop 및 파워 피벗에는 DAX를 사용하여 측정값, 계산 열 및 행 필터를 만드는 단원을 포함하는 문서와 자습서가 있습니다. 다음은 몇 가지 추가 리소스입니다.

비디오

Microsoft Learn의 [Power BI Desktop](#) 경로에서 DAX를 사용합니다.

Alberto Ferrari와 Marco Russo(Microsoft Press)의 [Definitive Guide to DAX](#). 두 번째 버전에서 이 광범위한 가이드는 초보 데이터 모델러와 BI 전문가를 위한 혁신적인 고성능 기술에 대한 기본 사항을 제공합니다.



커뮤니티

DAX에는 항상 전문 지식을 공유하는 활기찬 커뮤니티가 있습니다. Microsoft [Power BI 커뮤니티](#)에는 DAX, [DAX 명령 및 팁](#)을 위한 특별 토론 포럼이 있습니다.

동영상

2021-03-16 • 2 minutes to read

Excel 또는 Analysis Services에서 Power BI Desktop, Power Pivot을 사용하는지에 관계 없이 유효한 데이터 모델을 만드는 데는 DAX(Data Analysis Expressions)를 배우는 것이 중요합니다. 이 강력한 식 언어를 사용하여 시작하는 데 도움이 되는 몇 가지 비디오는 다음과 같습니다.

DAX 101

이 DAX 101 비디오에서는 Microsoft 파트너, Alberto Ferrari는 DAX의 필수 개념을 소개합니다. 유용하고 명확한 예를 사용하여 DAX를 사용하는 측정값, 계산 열 및 기본 데이터 모델링 식에 대해 알아봅니다.

고급 DAX

이 고급 DAX 비디오에서 Microsoft 파트너, Alberto Ferrari는 DAX 이론, 필터 및 행 컨텍스트 및 DAX의 기타 필수 개념을 설명합니다.

DAX 함수 참조

2021-03-16 • 7 minutes to read

DAX 함수 참조는 DAX(Data Analysis Expression) 수식에서 사용되는 250개 이상의 각 함수에 대한 구문, 매개 변수, 반환 값, 예제를 포함하여 자세한 정보를 제공합니다.

IMPORTANT

이전 버전의 Power BI Desktop, Analysis Services 및 Excel 파워 피벗에서는 일부 DAX 함수가 지원되거나 포함되지 않습니다.

섹션 내용

새로운 DAX 함수 - 이러한 함수는 신규 함수이거나 크게 향상된 기존 함수입니다.

날짜 및 시간 함수 - DAX의 이 함수는 Microsoft Excel의 날짜 및 시간 함수와 비슷합니다. 하지만 DAX 함수는 Microsoft SQL Server에서 사용되는 날짜/시간 데이터 형식을 기반으로 합니다.

필터 함수 - 이러한 함수는 특정 데이터 형식을 반환하고, 관련된 테이블에서 값을 조회하고, 관련 값으로 필터링하는 데 유용합니다. 조회 함수는 테이블과 테이블 사이의 관계를 사용하여 작동합니다. 필터링 함수를 사용하면 데이터 컨텍스트를 조작하여 동적 계산을 만들 수 있습니다.

재무 함수 - 이러한 함수는 순 현재 가치 및 수익률과 같은 재무 계산을 수행하는 수식에 사용됩니다.

정보 함수 - 이러한 함수는 다른 함수에 대한 인수로 제공된 테이블 또는 열을 조회하고 값이 예상된 형식과 일치하는지 여부를 알려줍니다. 예를 들어 ISERROR 함수는 참조 값에 오류가 포함된 경우 TRUE를 반환합니다.

논리 함수 - 이러한 함수는 식에 있는 값에 대한 정보를 반환합니다. 예를 들어 TRUE 함수를 사용하면 평가 중인 식이 TRUE 값을 반환하는지 여부를 알 수 있습니다.

수학 및 삼각 함수 - DAX의 수학 함수는 Excel의 수학 및 삼각 함수와 비슷합니다. 하지만 DAX 함수에서 사용되는 숫자 데이터 형식에 일부 차이가 있습니다.

기타 함수 - 이러한 함수는 다른 대부분의 함수에 속하는 범주로 정의될 수 없는 고유한 작업을 수행합니다.

부모 및 자식 함수 - 사용자는 이러한 DAX(Data Analysis Expressions) 함수를 사용하여 해당 데이터 모델에 부모/자식 계층으로 표시된 데이터를 관리할 수 있습니다.

관계 함수 - 이러한 함수는 테이블 간의 관계를 관리하고 활용하기 위한 것입니다. 예를 들어 계산에 사용할 특정 관계를 지정할 수 있습니다.

통계 함수 - 이러한 함수는 집계를 수행합니다. 합계 및 평균을 만들거나, 최솟값 및 최댓값을 찾는 것 외에도 DAX에서는 관련 테이블을 기준으로 집계하거나 집계를 생성하기 전에 열을 필터링할 수도 있습니다.

테이블 조작 함수 - 이러한 함수는 테이블을 반환하거나 기존 테이블을 조작합니다.

텍스트 함수 - 이러한 함수를 사용하면 문자열의 일부를 반환하거나, 문자열 내에서 텍스트를 검색하거나, 문자열 값을 연결할 수 있습니다. 추가 함수는 날짜, 시간 및 숫자의 형식을 제어하기 위한 것입니다.

시간 인텔리전스 함수 - 이러한 함수는 달력과 날짜에 대해 기본 제공되는 정보를 사용하는 계산을 만들 때 유용합니다. 집계 또는 계산과 함께 시간 및 날짜 범위를 사용하여 판매, 재고 등에 대해 의미 있는 기간별 비교를 빌드할 수 있습니다.

참고 항목

[DAX 구문 참조](#)

[DAX 연산자 참조](#)

[DAX 매개 변수 명명 규칙](#)

새로운 DAX 함수

2021-04-22 • 4 minutes to read

DAX는 새로운 기능을 지원하기 위해 새로운 함수 및 기능으로 지속적으로 개선되고 있습니다. 새 함수 및 업데이트는 대부분의 경우 매월 제공되는 서비스, 애플리케이션 및 도구 업데이트에 포함됩니다.

함수 및 기능은 항상 업데이트되고 있지만 사용자에게 표시되고 노출되는 기능 변경 사항이 있는 업데이트만 설명서에서 설명합니다. 지난해의 새로운 함수 및 기존 함수 업데이트 내역은 여기에 표시됩니다.

IMPORTANT

일부 함수는 Power BI Desktop, Analysis Services 및 Excel의 파워 피벗 일부 버전에서 지원되지 않습니다. 새 함수 및 업데이트된 함수는 일반적으로 Power BI Desktop에서 처음 소개됩니다.

새로운 함수

함수	월
IF.EAGER	2021년 3월
ACCRINT	2020년 7월
ACCRINTM	2020년 7월
AMORDEGRC	2020년 7월
AMORLINC	2020년 7월
COUPDAYBS	2020년 7월
COUPDAYS	2020년 7월
COUPDAYSNC	2020년 7월
COUPNCD	2020년 7월
COUPNUM	2020년 7월
COUPPCD	2020년 7월
CUMIPMT	2020년 7월
CUMPRINC	2020년 7월
DB	2020년 7월
DDB	2020년 7월

함수	월
DISC	2020년 7월
DOLLARDE	2020년 7월
DOLLARFR	2020년 7월
DURATION	2020년 7월
EFFECT	2020년 7월
FV	2020년 7월
INTRATE	2020년 7월
IPMT	2020년 7월
ISPMT	2020년 7월
MDURATION	2020년 7월
NOMINAL	2020년 7월
NPER	2020년 7월
ODDFPRICE	2020년 7월
ODDFYIELD	2020년 7월
ODDLPRICE	2020년 7월
ODDLYIELD	2020년 7월
PDURATION	2020년 7월
PMT	2020년 7월
PPMT	2020년 7월
PRICE	2020년 7월
PRICEDISC	2020년 7월
PRICEMAT	2020년 7월
PV	2020년 7월
RATE	2020년 7월
RECEIVED	2020년 7월

함수	월
RRI	2020년 7월
SLN	2020년 7월
SYD	2020년 7월
TBILLEQ	2020년 7월
TBILLPRICE	2020년 7월
TBILLYIELD	2020년 7월
VDB	2020년 7월
YIELD	2020년 7월
YIELDDISC	2020년 7월
YIELDMAT	2020년 7월

업데이트된 함수

함수	월	설명
CROSSFILTER	2021년 4월	방향 매개 변수에 대한 추가 옵션.
CALCULATE	2021년 3월	필터가 여러 개 있는 경우 OR() 연산자를 지원합니다.

날짜 및 시간 함수

2021-03-16 • 4 minutes to read

이 함수를 사용하여 날짜 및 시간을 기반으로 계산합니다. DAX의 많은 함수는 Excel의 날짜 및 시간 함수와 비슷합니다. 그러나 DAX 함수는 **datetime** 데이터 형식을 사용하고 열 값을 인수로 사용할 수 있습니다.

이 범주의 함수

함수	DESCRIPTION
CALENDAR	연속된 날짜 집합이 포함된 'Date'라는 단일 열이 있는 테이블을 반환합니다.
CALENDARAUTO	연속된 날짜 집합이 포함된 'Date'라는 단일 열이 있는 테이블을 반환합니다.
DATE	지정한 날짜를 datetime 형식으로 반환합니다.
DATEDIFF	두 날짜 간에 겹치는 간격 경계 수를 반환합니다.
DATEVALUE	텍스트 형식의 날짜를 datetime 형식의 날짜로 변환합니다.
DAY	해당 월의 일(1~31 사이의 숫자)을 반환합니다.
EDATE	시작 날짜 이전이나 이후의 지정된 개월 수에 해당하는 날짜를 반환합니다.
EOMONTH	지정한 개월 수 이전이나 이후 달의 마지막 날 날짜를 datetime 형식으로 반환합니다.
HOUR	0(오전 12:00)부터 23(오후 11:00)까지의 숫자로 시간을 반환합니다.
MINUTE	날짜 및 시간 값이 지정된 경우, 분을 0에서 59 사이의 숫자로 반환합니다.
MONTH	1(1월)에서 12(12월) 사이의 숫자로 월을 반환합니다.
NOW	현재 날짜 및 시간을 datetime 형식으로 반환합니다.
QUARTER	분기를 1에서 4 사이의 숫자로 반환합니다.
SECOND	시간 값의 초를 0에서 59 사이의 숫자로 반환합니다.
TIME	수치로 제공된 시간, 분, 초를 datetime 형식의 시간으로 변환합니다.
TIMEVALUE	텍스트 형식의 시간을 datetime 형식의 시간으로 변환합니다.

함수	DESCRIPTION
TODAY	현재 날짜를 반환합니다.
UTCNOW	현재 UTC 날짜 및 시간을 반환합니다.
UTCTODAY	현재 UTC 날짜를 반환합니다.
WEEKDAY	날짜의 요일을 나타내는 1~7 사이의 수를 반환합니다.
WEEKNUM	return_type 값에 따라 지정된 날짜와 연도의 주 번호를 반환합니다.
YEAR	날짜의 연도를 1900~9999 범위의 4자리 정수로 반환합니다.
YEARFRAC	두 날짜 사이의 전체 일수로 표시되는 연도의 비율을 계산합니다.

CALENDAR

2021-03-16 • 2 minutes to read

연속된 날짜 집합이 포함된 'Date'라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 지정된 시작 날짜부터 지정된 종료 날짜(두 날짜 포함)까지입니다.

구문

```
CALENDAR(<start_date>, <end_date>)
```

매개 변수

용어	정의
start_date	날짜/시간 값을 반환하는 DAX 식입니다.
end_date	날짜/시간 값을 반환하는 DAX 식입니다.

반환 값

연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 지정된 시작 날짜부터 지정된 종료 날짜(두 날짜 포함)까지입니다.

설명

- start_date가 end_date보다 크면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

다음 수식은 2005년 1월 1일에서 2015년 12월 31일 사이의 날짜가 포함된 테이블을 반환합니다.

```
= CALENDAR (DATE (2005, 1, 1), DATE (2015, 12, 31))
```

실제 판매 데이터 및 향후 판매 예측을 포함하는 데이터 모델의 경우 다음 식은 이러한 두 테이블의 날짜 범위를 포함하는 날짜 테이블을 반환합니다.

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

CALENDARAUTO

2021-04-27 • 3 minutes to read

연속된 날짜 집합이 포함된 'Date'라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 모델 데이터를 기준으로 자동으로 계산됩니다.

구문

```
CALENDARAUTO([fiscal_year_end_month])
```

매개 변수

용어	정의
fiscal_year_end_month	1에서 12 사이의 정수를 반환하는 DAX 식입니다. 생략하면 현재 사용자의 달력 테이블 템플릿에 지정된 값이 기본값으로 사용됩니다(있는 경우). 없는 경우 기본값은 12입니다.

반환 값

연속된 날짜 집합이 포함된 'Date'라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 모델 데이터를 기준으로 자동으로 계산됩니다.

설명

- 날짜 범위는 다음과 같이 계산됩니다.
 - 모델에서 계산 열이나 계산 테이블에 없는 가장 이른 날짜가 MinDate로 사용됩니다.
 - 모델에서 계산 열이나 계산 테이블에 없는 가장 늦은 날짜가 MaxDate로 사용됩니다.
 - 반환되는 날짜 범위는 MinDate와 관련된 회계 연도 시작부터 MaxDate와 관련된 회계 연도 끝까지의 날짜입니다.
- 계산 열이나 계산 테이블에 있는 날짜/시간 값만 모델에 포함되어 있으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

이 예제에서 데이터 모델의 MinDate와 MaxDate는 2010년 7월 1일과 2011년 6월 30일입니다.

`CALENDARAUTO()` 는 2010년 1월 1일부터 2011년 12월 31일까지의 모든 날짜를 반환합니다.

`CALENDARAUTO(3)` 는 2010년 3월 1일부터 2012년 3월 31일까지의 모든 날짜를 반환합니다.

DATE

2021-03-16 • 9 minutes to read

지정한 날짜를 **datetime** 형식으로 반환합니다.

구문

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

매개 변수

용어	정의
연도	<p>연도를 나타내는 숫자입니다.</p> <p>year 인수의 값은 1~4 자리수를 포함할 수 있습니다. year 인수는 컴퓨터에서 사용하는 날짜 시스템에 따라 해석됩니다.</p> <p>1900년 3월 1일부터 시작되는 날짜가 지원됩니다.</p> <p>소수 자릿수가 있는 숫자를 입력하면 숫자가 반올림됩니다.</p> <p>9999보다 크거나 0보다 작은 값(음수 값)의 경우, 함수는 #VALUE! 을(를) 반환합니다. error.</p> <p>year 값이 0에서 1899 사이인 경우, 값이 1900에 추가되어 최종 값이 생성됩니다. 아래 예제를 참조하세요. 참고: 원치 않는 결과를 방지하려면 가능한 경우 year 인수에 4자리 숫자를 사용해야 합니다. 예를 들어, 07을 사용하면 연도 값으로 1907이 반환됩니다.</p>
월	<p>월을 나타내는 숫자 또는 다음 규칙으로 계산됩니다.</p> <p>음의 정수는 지원되지 않습니다. 유효한 값은 1~12입니다.</p> <p>month 가 1에서 12 사이의 숫자인 경우, 해당 연도의 월을 나타냅니다. 1은 1월, 2는 2월 등등 12는 12월을 나타냅니다.</p> <p>12보다 큰 정수를 입력하는 경우 다음과 같이 계산됩니다. month 값을 year 에 추가하여 날짜가 계산됩니다. 예를 들어, 2008 시작에 18개월이 추가되어 2009년 6월이 생성되므로, DATE(2008, 18, 1)가 있는 경우 함수는 2009년 6월 1일과 동등한 날짜/시간 값을 반환합니다. 아래 예제를 참조하세요.</p>

용어	정의
일	<p>일을 나타내는 숫자 또는 다음 규칙으로 계산됩니다.</p> <p>음의 정수는 지원되지 않습니다. 유효한 값은 1~31입니다.</p> <p>day 가 1에서 지정된 월의 마지막 날 사이의 숫자인 경우, 이는 해당 월의 일을 나타냅니다.</p> <p>지정된 월의 마지막 날보다 큰 정수를 입력하는 경우 다음과 같이 계산됩니다. month 에 day 값을 추가하여 날짜가 계산됩니다. 예를 들어, 3월 시작에 32일이 추가되어 4월 1일이 생성되므로, <code>DATE(2008, 3, 32)</code> 수식에서 DATE 함수는 2008년 4월 1일과 동등한 datetime 값을 반환합니다.</p> <p>day 에 소수 부분이 있는 경우, 가장 가까운 정수 값으로 반올림됩니다.</p>

반환 값

지정된 날짜(**datetime**) 를 반환합니다.

설명

- DATE 함수는 인수로 입력된 정수를 사용하여 해당 날짜를 생성합니다. DATE 함수는 수식에서 year, month 및 day를 제공하는 경우에 가장 유용합니다. 예를 들어, 기본 데이터에는 날짜로 인식되지 않는 형식의 날짜(예: YYYYMMDD)가 포함될 수 있습니다. DATE 함수를 다른 함수와 함께 사용하여 날짜로 인식될 수 있는 숫자로 날짜를 변환할 수 있습니다.
- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리, DAX 날짜 함수는 항상 **datetime** 데이터 형식을 반환합니다. 그러나 필요한 경우 서식을 사용하여 날짜를 일련 번호로 표시할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제: 단순 날짜 반환

설명

다음 수식은 2009년 7월 8일 날짜를 반환합니다.

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

예제: 1899 이전 연도

year 인수에 대해 입력하는 값이 0(영)에서 1899(포함) 사이인 경우, 이 값을 1900에 추가하여 연도를 계산합니다. 다음 수식은 1908년 1월 2일을 반환합니다. (1900+08).

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

예제: 1899 이후의 연도

year 가 1900에서 9999(포함) 사이인 경우 해당 값이 연도로 사용됩니다. 다음 수식은 2008년 1월 2일을 반환합니다.

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

예제: 월 작업

month 가 12보다 큰 경우, **month** 에서 지정된 연도의 첫 번째 월에 월 수를 추가합니다. 다음 수식은 2009년 2월 2일을 반환합니다.

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

예제: 일 작업

day 가 지정된 월의 일 수보다 큰 경우, **day** 에서 해당 월의 첫째 날에 해당 일 수를 추가합니다. 다음 수식은 2008년 2월 4일을 반환합니다.

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

참고 항목

[날짜 및 시간 함수](#)

[DAY 함수](#)

[TODAY 함수](#)

DATEDIFF

2021-03-16 • 2 minutes to read

두 날짜 간에 겹치는 간격 경계 수를 반환합니다.

구문

```
DATEDIFF(<start_date>, <end_date>, <interval>)
```

매개 변수

용어	정의
start_date	스칼라 일자/시간 값입니다.
end_date	스칼라 일자/시간 값 반환 값입니다.
interval	날짜를 비교할 때 사용할 간격입니다. 값은 다음 중 하나일 수 있습니다. - 초 - 분 - 시간 - 일 - 주 - 월 - 분기 - 연도

반환 값

두 날짜 간에 겹치는 간격 경계의 수입니다.

설명

start_date가 end_date보다 큰 경우, 오류가 반환됩니다.

예제

DATE
2012-12-31 23:59:59
2013-01-01 00:00:00

다음은 모두 1을 반환합니다.


```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), SECOND )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MINUTE )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), HOUR )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), DAY )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), WEEK )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MONTH )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), QUARTER )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), YEAR )
```

DATEVALUE

2021-03-16 • 4 minutes to read

텍스트 형식의 날짜를 날짜/시간 형식으로 변환합니다.

구문

```
DATEVALUE(date_text)
```

매개 변수

용어	정의
date_text	날짜를 나타내는 텍스트입니다.

속성 값/반환 값

datetime 형식의 날짜입니다.

설명

- 변환할 때, DATEVALUE는 모델의 로캘과 날짜/시간 설정을 사용해 날짜 값을 결정합니다. 모델 날짜/시간 설정이 월/일/년의 형식으로 날짜를 나타낼 경우, 문자열 "1/8/2009"은 2009년 1월 8일에 해당하는 **날짜/시간** 값으로 변환됩니다. 그러나, 모델 날짜/시간 설정이 일/월/년의 형식으로 날짜를 나타내는 경우, 동일한 문자열이 2009년 8월 1일에 해당하는 **날짜/시간** 값으로 변환됩니다.
- 모델의 로캘 및 날짜/시간 설정을 사용해 변환하지 못한다면, DATEVALUE는 다른 날짜 형식의 사용을 시도합니다. 이 경우 일부 행은 한 형식을 사용해 변환할 수 있으며, 다른 행들은 다른 형식을 사용해 변환됩니다. 예를 들어, "5/4/2018"은 2018년 5월 4일로, "20/4/2018"은 4월 20일로 변환될 수 있습니다.
- date_text** 인수의 연도 부분을 생략하면 DATEVALUE 함수는 컴퓨터의 기본 제공 클록에서 현재 연도를 사용합니다. **date_text** 인수의 시간 정보는 무시됩니다.
- 모델 로캘 및 날짜/시간 설정은 모델을 생성할 때 애플리케이션 및 컴퓨터가 초반에 결정합니다.

예제

다음 예제는 날짜 및 시간을 나타내는 방식에 대한 모델 로캘 및 설정에 따라 다른 **날짜/시간** 값을 반환합니다.

- 일이 월보다 앞에 오는 날짜/시간 설정에서는 예제에서 2009년 1월 8일에 해당하는 **datetime** 값이 반환됩니다.
- 월이 일보다 앞에 오는 날짜/시간 설정에서는 예제에서 2009년 8월 1일에 해당하는 **datetime** 값이 반환됩니다.

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

참고 항목

[날짜 및 시간 함수](#)

DAY

2021-03-16 • 4 minutes to read

1 ~ 31 사이의 해당 월의 일을 반환합니다.

구문

DAY(<date>)

매개 변수

용어	정의
날짜	datetime 형식의 날짜 또는 날짜의 텍스트 표현입니다.

반환 값

해당 월의 일을 나타내는 정수입니다.

설명

- DAY 함수는 찾으려는 날짜를 인수로 사용합니다. 날짜는 다른 날짜 함수를 사용하거나 날짜를 반환하는 식을 사용하거나 **datetime** 형식으로 날짜를 입력하여 함수에 제공할 수 있습니다. 날짜에 대해 허용되는 문자열 형식 중 하나로 날짜를 입력할 수도 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공된 날짜 값의 표시 형식과 관계없이 양력 값입니다. 예를 들어 제공된 날짜의 표시 형식이 회교식인 경우 YEAR, MONTH 및 DAY 함수에 대해 반환되는 값은 해당하는 양력 날짜와 연결된 값입니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 day 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 현재 날짜/시간 설정이 월/일/연도 형식의 날짜를 나타내는 경우 "1/8/2009" 문자열은 2009년 1월 8일에 해당하는 **datetime** 값으로 해석되고 함수는 8을 반환합니다. 그러나 현재 날짜/시간 설정이 날짜를 일/월/연도 형식으로 나타내는 경우 동일한 문자열이 2009년 8월 1일에 해당하는 **datetime** 값으로 해석되고 함수는 1을 반환합니다.

예제: 날짜 열에서 일 가져오기

다음 수식은 [Birthdate] 열의 날짜에서 일을 반환합니다.

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

예제 - 문자열 날짜에서 일 가져오기

다음 수식은 허용된 텍스트 형식의 문자열로 제공된 날짜를 사용하여 4일을 반환합니다.

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

예제 - 조건으로 day 값 사용

다음 식은 각 판매 주문이 발생한 날짜를 반환하고, 해당 월의 10일에 주문이 발생한 경우 행에 판촉 판매 항목으로 플래그를 설정합니다.

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

참고 항목

[날짜 및 시간 함수](#)

[TODAY 함수](#)

[DATE 함수](#)

EDATE

2021-03-16 • 4 minutes to read

시작 날짜 이전이나 이후의 지정된 개월 수에 해당하는 날짜를 반환합니다. 해당 월에서 발행 날짜와 같은 날짜인 만기일 또는 기한을 계산하려면 EDATE를 사용합니다.

구문

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

매개 변수

용어	정의
start_date	시작 날짜를 나타내는 datetime 또는 text 형식의 날짜입니다.
months	start_date 이전 또는 이후의 월 수를 나타내는 정수입니다.

반환 값

날짜(**datetime**)입니다.

설명

- 날짜를 순차적 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜를 **datetime** 형식으로 처리합니다. 다른 형식으로 저장된 날짜는 암시적으로 변환됩니다.
- start_date** 가 유효한 날짜가 아니면 EDATE는 오류를 반환합니다. 첫 번째 인수로 제공하는 열 참조 또는 날짜가 날짜인지 확인하세요.
- months** 는 정수가 아니면 잘립니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 EDATE 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜 시간 설정을 사용하여 텍스트 값을 이해합니다. 현재 날짜 시간 설정이 날짜를 월/일/연도 형식으로 표현하는 경우 다음 "1/8/2009" 문자열은 2009년 1월 8일에 해당하는 **datetime** 값으로 해석됩니다. 하지만 현재 날짜 시간 설정이 날짜를 일/월/연도 형식으로 표현하는 경우 동일한 문자열이 2009년 8월 1일에 해당하는 **datetime** 값으로 해석됩니다.
- 요청된 날짜가 해당 월의 마지막 날을 지난 경우 해당 월의 마지막 날이 반환됩니다. 예를 들어 EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) 함수는 시작 날짜의 한 달 후에 해당하는 2009년 2월 28일을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 주문 날짜로부터 3개월 후의 날짜를 반환하며 이 날짜는 [TransactionDate] 열에 저장됩니다.

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

참고 항목

[EOMONTH 함수](#)

[날짜 및 시간 함수](#)

EOMONTH

2021-03-16 • 4 minutes to read

지정한 개월 수 이전이나 이후 달의 마지막 날 날짜를 **datetime** 형식으로 반환합니다. EOMONTH를 사용하여 해당 월의 마지막 날에 속하는 만기일 또는 기한을 계산합니다.

구문

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

매개 변수

용어	정의
start_date	datetime 형식의 시작 날짜 또는 해당 날짜의 허용되는 텍스트 표현입니다.
월	start_date 의 이전 또는 이후의 월 수를 나타내는 숫자입니다. 참고: 정수가 아닌 숫자를 입력하는 경우 숫자는 정수 단위로 반올림됩니다.

반환 값

날짜(**datetime**)입니다.

설명

- 날짜를 순차적인 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜를 **datetime** 형식으로 처리합니다. EOMONTH 함수는 다른 형식의 날짜를 허용하며, 다음과 같은 제한 사항이 있습니다.
- start_date** 가 유효한 날짜가 아니면 EOMONTH는 오류를 반환합니다.
- start_date** 는 **datetime** 형식이 아닌 숫자 값인 경우 EOMONTH는 숫자를 날짜로 변환합니다. 예기치 않은 결과를 방지하려면 EOMONTH 함수를 사용하기 전에 숫자를 **datetime** 형식으로 변환합니다.
- start_date** + 월이 잘못된 날짜를 산출하는 경우 EOMONTH는 오류를 반환합니다. 1900년 3월 1일 이전 날짜와 9999년 12월 31일 이후의 날짜는 유효하지 않습니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 EDATE 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜 시간 설정을 사용하여 텍스트 값을 이해합니다. 현재 날짜 시간 설정이 월/일/연도 형식의 날짜를 나타내는 경우 다음 문자열 "1/8/2009"는 2009년의 1월 8일에 해당하는 **datetime** 값으로 해석됩니다. 그러나 현재 날짜 시간 설정이 날짜를 일/월/연도 형식으로 나타내는 경우에는 동일한 문자열이 2009년 8월 1일에 해당하는 **datetime** 값으로 해석됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 식은 **months** 인수가 2로 반올림되므로 2008년 5월 31일을 반환합니다.

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

참고 항목

[EDATE 함수](#)

[날짜 및 시간 함수](#)

hour

2021-03-16 • 3 minutes to read

0(오전 12:00)부터 23(오후 11:00)까지의 숫자로 시간을 반환합니다.

구문

```
hour(<datetime>)
```

매개 변수

용어	정의
Datetime	16:48:00 또는 4:48 PM과 같은 datetime 값입니다.

반환 값

0에서 23 사이의 정수입니다.

설명

- hour 함수는 찾으려는 시간을 포함하는 시간을 인수로 사용합니다. 날짜/시간 함수, **datetime** 을 반환하는 식을 사용하거나 허용된 시간 형식 중 하나에 직접 값을 입력하여 시간을 제공할 수 있습니다. 시간은 모든 허용된 텍스트 표현으로도 입력할 수 있습니다.
- datetime** 인수가 날짜 및 시간의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 대부분의 로캘에서는 콜론(:)을 시간 구분 기호로 사용하며, 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 모두 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 파악합니다.

예제 1

다음 예제에서는 **Orders** 라는 테이블의 **TransactionTime** 열에서 시간을 반환합니다.

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

예 2

다음 예제에서는 24시간제로 오후 3시에 해당하는 시간을 나타내는 15를 반환합니다. 텍스트 값이 자동으로 구문 분석되어 날짜/시간 값으로 변환됩니다.

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

참고 항목

[날짜 및 시간 함수](#)

[minute 함수](#)

YEAR 함수
SECOND 함수

MINUTE

2021-03-16 • 3 minutes to read

날짜 및 시간 값이 지정된 경우, 분을 0에서 59 사이의 숫자로 반환합니다.

구문

```
MINUTE(<datetime>)
```

매개 변수

용어	정의
Datetime	허용되는 시간 형식의 날짜/시간 값이나 텍스트입니다(예: 16:48:00 또는 오후 4:48).

반환 값

0에서 59 사이의 정수입니다.

설명

- 날짜와 시간을 일련 번호 형식으로 저장하는 Microsoft Excel과 달리, DAX는 날짜 및 시간에 날짜/시간 데이터 형식을 사용합니다. 날짜 및 시간을 저장하는 열을 참조하거나, 날짜/시간 함수를 사용하거나, 날짜 및 시간을 반환하는 식을 사용하여 MINUTE 함수에 날짜/시간 값을 제공할 수 있습니다.
- datetime** 인수가 날짜 및 시간의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 대부분의 로캘에서는 콜론(:)을 시간 구분 기호로 사용하며, 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 모두 올바르게 구문 분석됩니다. 로캘 설정을 확인하여 결과를 파악합니다.

예 1

다음 예제는 **Orders** 테이블의 **TransactionTime** 열에 저장된 값의 분을 반환합니다.

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

예제 2

다음 예제는 오후 1:45 시간의 분 수인 45를 반환합니다.

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

참고 항목

[날짜 및 시간 함수](#)

[HOUR 함수](#)

YEAR 함수
SECOND 함수

MONTH

2021-03-16 • 4 minutes to read

1(1월)에서 12(12월) 사이의 숫자로 월을 반환합니다.

구문

```
MONTH(<datetime>)
```

매개 변수

용어	정의
date	날짜/시간 또는 텍스트 형식의 날짜입니다.

반환 값

1에서 12 사이의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리, DAX는 날짜 작업 시 **날짜/시간** 형식을 사용합니다. 허용되는 **날짜/시간** 형식을 입력하거나, 날짜가 포함된 열에 대한 참조를 제공하거나, 날짜를 반환하는 식을 사용하여 MONTH 함수의 인수로 사용되는 날짜를 입력할 수 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공한 날짜 값의 표시 형식에 관계없이 양력 값입니다. 예를 들어 제공한 날짜의 표시 형식이 회교식이라도 YEAR, MONTH 및 DAY 함수의 반환 값은 해당하는 양력 날짜의 값입니다.
- date 인수가 날짜의 텍스트 표현인 경우, 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜 시간 설정을 사용하여 텍스트 값을 파싱합니다. 현재 날짜/시간 설정이 날짜를 월/일/연도 형식으로 나타내는 경우 "1/8/2009" 문자열은 2009년 1월 8일에 해당하는 날짜/시간 값으로 해석되고, 함수는 결과 1을 생성합니다. 그러나 현재 날짜/시간 설정이 날짜를 일/월/연도 형식으로 나타내는 경우 동일한 문자열이 2009년 8월 1일에 해당하는 날짜/시간 값으로 해석되고, 함수는 결과 8을 생성합니다.
- 날짜의 텍스트 표현을 날짜/시간 값으로 올바르게 변환할 수 없으면 함수는 오류를 반환합니다.

예 1

다음 식은 **date** 인수의 월인 3월에 해당하는 정수 3을 반환합니다.

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

예제 2

다음 식은 **Orders** 테이블의 **TransactionDate** 열에 있는 날짜의 월을 반환합니다.

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

참고 항목

날짜 및 시간 함수

HOUR 함수

MINUTE 함수

YEAR 함수

SECOND 함수

NOW

2021-03-16 • 2 minutes to read

현재 날짜와 시간을 **datetime** 형식으로 반환합니다.

NOW 함수는 현재 날짜 및 시간을 워크시트에 표시하거나 현재 날짜 및 시간을 기반으로 값을 계산하고 워크시트를 열 때마다 해당 값이 업데이트되도록 해야 하는 경우 유용합니다.

구문

```
NOW()
```

반환 값

날짜(**datetime**)입니다.

설명

- NOW 함수의 결과는 수식이 포함된 열을 새로 고치는 경우에만 변경됩니다. 지속적으로 업데이트되지 않습니다.
- Power BI 서비스에서 NOW 함수의 결과는 항상 UTC 표준 시간대입니다.
- TODAY 함수도 동일한 날짜를 반환하지만 시간 면에서는 정확하지 않습니다. 반환되는 시간은 항상 오전 12:00:00이고 날짜만 업데이트됩니다.

예제

다음 예제에서는 현재 날짜 및 시간에 3.5일을 더한 값을 반환합니다.

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

참고 항목

[UTCNOW 함수](#)

[TODAY 함수](#)

QUARTER

2021-03-16 • 2 minutes to read

1(1월~3월)에서 4(10월~12월) 사이의 숫자로 분기를 반환합니다.

구문

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

매개 변수

용어	정의
date	날짜입니다.

반환 값

1에서 4 사이의 정수입니다.

설명

입력 값이 비어 있으면 출력 값도 비어 있습니다.

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

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

그러면 다음 값이 반환됩니다.

[VALUE]
1
4

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.


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

그러면 다음 값이 반환됩니다.

FACTINTERNETSALES[ORDERDATE]	[QUARTER]
2008년 3월 31일	1
2008년 4월 1일	2

참조

[YEAR](#)

[MONTH](#)

[DAY](#)

SECOND

2021-03-16 • 5 minutes to read

시간 값의 초를 0에서 59 사이의 숫자로 반환합니다.

구문

SECOND(<time>)

매개 변수

용어	정의
time	datetime 형식(예: 16:48:23 또는 4:48:47 PM)의 시간입니다.

반환 값

0에서 59 사이의 정수입니다.

설명

- 날짜 및 시간을 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간으로 작업할 때 **datetime** 형식을 사용합니다. 원본 데이터가 이 형식이 아닌 경우 DAX는 암시적으로 데이터를 변환합니다. 서식 지정을 사용하여 날짜 및 시간을 필요한 일련 번호로 표시할 수 있습니다.
- SECOND 함수에 인수로 제공하는 날짜/시간 값을 따옴표 안에 텍스트 문자열로 입력할 수 있습니다(예: "6:45 PM"). 시간 값을 다른 식의 결과로 제공하거나 시간을 포함하는 열에 대한 참조로 제공할 수도 있습니다.
- 13.60과 같은 다른 데이터 형식의 숫자 값을 제공하는 경우 값은 일련 번호로 해석되고 초의 값을 추출하기 전에 **datetime** 데이터 형식으로 표시됩니다. 결과를 더 쉽게 이해할 수 있도록 SECOND 함수에서 사용하기 전에 해당 숫자를 날짜로 표시하는 것이 좋습니다. 예를 들어 **25.56** 과 같은 숫자 값을 포함하는 열에서 SECOND를 사용하는 경우 수식은 24를 반환합니다. 날짜로 형식이 지정된 경우 값 25.56은 January 25, 1900, 1:26:24 PM과 동일하기 때문입니다.
- time** 인수가 날짜 및 시간의 텍스트 표현인 경우 함수는 텍스트 값을 파싱하여 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용합니다. 대부분의 로캘에서는 콜론(:)을 시간 구분 기호로 사용하며, 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 모두 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 파악합니다.

예제 1

다음 수식은 **Orders** 라는 테이블의 **TransactionTime** 열에 포함된 시간에서 초의 숫자를 반환합니다.

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

예 2

다음 수식은 3을 반환하는데, 이는 **March 3, 2008 12:00:03** 에서 초의 숫자를 값으로 표현한 것입니다.

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

참고 항목

[날짜 및 시간 함수](#)

[HOUR](#)

[MINUTE](#)

[YEAR](#)

TIME

2021-03-16 • 5 minutes to read

숫자로 제공된 시, 분, 초를 **datetime** 형식의 시간으로 변환합니다.

구문

TIME(hour, minute, second)

매개 변수

용어	정의
hour	0부터 23까지 시간을 나타내는 수입니다. 23보다 큰 값은 24로 나누고, 나머지는 시 값으로 처리됩니다.
minute	0부터 59까지 분을 나타내는 수입니다. 59보다 큰 값은 시 및 분으로 변환됩니다.
second	0부터 59까지 초를 나타내는 수입니다. 59보다 큰 값은 시, 분 및 초로 변환됩니다.

반환 값

시간(**datetime**)입니다.

설명

- 날짜 및 시간을 일련 번호로 저장하는 Microsoft Excel과 달리, DAX는 날짜 및 시간 값을 **datetime** 형식으로 처리합니다. DAX 함수에서 날짜/시간 값을 사용하는 경우 다른 형식의 숫자는 암시적으로 변환됩니다. 일련 번호를 사용해야 하는 경우 서식 지정을 사용하여 숫자가 표시되는 방식을 변경할 수 있습니다.
- 시간 값은 날짜 값의 일부이며 10진수로 표시되는 일련 번호 시스템에 속합니다. 따라서 **datetime** 값 12:00 PM은 하루 절반이기 때문에 0.5와 동일합니다.
- TIME 함수에 대한 인수를 직접 입력하는 값으로, 다른 식의 결과로 또는 숫자 값이 포함된 열에 대한 참조로 제공할 수 있습니다. 다음 제한 사항이 적용됩니다.
 - 23보다 큰 **hours** 값은 24로 나누고, 나머지는 시간 값으로 처리됩니다.
 - 59보다 큰 **minutes** 값은 시 및 분으로 변환됩니다.
 - 59보다 큰 **seconds** 값은 시, 분 및 초로 변환됩니다.
 - 분 또는 초의 경우 24시보다 큰 값은 24로 나누고, 나머지를 시 값으로 처리합니다. 24시를 초과하는 값은 날짜 부분을 변경하지 않습니다.
- 이 함수가 반환하는 시간 값의 가독성을 높이려면 Microsoft Excel에서 제공하는 시간 형식 중 하나를 사용하여 수식의 결과가 포함된 열 또는 피벗 테이블 셀의 서식을 지정하는 것이 좋습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

니다.

예 1

다음 예제에서는 모두 시간 3:00 AM을 반환합니다.

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

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

예제 2

다음 예제에서는 모두 시간 12:30 PM을 반환합니다.

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

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

예제 3

다음 예제에서는 `intHours`, `intMinutes`, `intSeconds` 열의 값을 기반으로 시간을 만듭니다.

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

참고 항목

[DATE](#)

[날짜 및 시간 함수](#)

TIMEVALUE

2021-03-16 • 2 minutes to read

텍스트 형식의 시간을 `datetime` 형식의 시간으로 변환합니다.

구문

```
TIMEVALUE(time_text)
```

매개 변수

용어	정의
time_text	하루 중 특정 시간을 나타내는 텍스트 문자열입니다. time_text 인수에 포함된 날짜 정보는 무시됩니다.

반환 값

날짜(**datetime**)입니다.

설명

- 시간 값은 날짜 값의 일부이며 10진수로 표시됩니다. 예를 들어 오후 12:00은 하루의 절반이기 때문에 0.5로 표시됩니다.
- time_text** 인수가 날짜 및 시간의 텍스트 표현인 경우, 함수는 텍스트 값을 해석하여 변환하기 위해 모델의 로캘 및 날짜/시간 설정을 사용합니다. 대부분의 로캘은 콜론(:)을 시간 구분 기호로 사용하며, 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 모두 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 파악합니다.

예제

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

참고 항목

[날짜 및 시간 함수](#)

TODAY

2021-03-16 • 2 minutes to read

현재 날짜를 반환합니다.

구문

```
TODAY()
```

반환 값

날짜(datetime)입니다.

설명

- TODAY 함수는 통합 문서를 열 때와 상관 없이 워크시트에 현재 날짜를 표시해야 하는 경우에 유용합니다. 간격을 계산하는 데에도 유용합니다.
- TODAY 함수에서 날짜를 업데이트하지 않을 경우 열 또는 통합 문서를 새로 고치는 시기를 제어하는 설정을 변경해야 할 수 있습니다.
- NOW 함수는 비슷하지만 정확한 시간을 반환하는 반면 TODAY는 모든 날짜의 시간 값 12:00:00 PM을 반환합니다.

예제

누군가가 1963년에 태어났다는 것을 아는 경우 다음 수식을 사용하여 올해의 생일을 기준으로 해당 사람의 나이를 찾을 수 있습니다.

```
= YEAR(TODAY())-1963
```

이 수식은 TODAY 함수를 YEAR 함수에 대한 인수로 사용하여 현재 연도를 가져온 다음, 1963을 빼서 사용자의 나이를 반환합니다.

참고 항목

[날짜 및 시간 함수](#)

[NOW](#)

UTCNOW

2021-03-16 • 2 minutes to read

현재 UTC 날짜 및 시간을 반환합니다.

구문

```
UTCNOW()
```

반환 값

A (datetime) .

설명

UTCNOW 함수의 결과는 수식을 새로 고칠 때만 변경됩니다. 지속적으로 업데이트되지 않습니다.

예제

함수는 다음과 같이 사용됩니다.

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

HRESULT = NO_ERROR를

[VALUE]

2018년 2월 2일 오전 4:48:08

참고 항목

[NOW 함수](#)

[UTCTODAY 함수](#)

UTCTODAY

2021-03-16 • 2 minutes to read

현재 UTC 날짜를 반환합니다.

구문

```
UTCTODAY()
```

반환 값

날짜입니다.

설명

- UTCTODAY는 모든 날짜에 대해 오후 12:00:00 시간 값을 반환합니다.
- UTCNOW 함수도 유사하지만 정확한 시간 및 날짜를 반환합니다.

예제

함수는 다음과 같이 사용됩니다.

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

HRESULT = NO_ERROR를

[VALUE]

2/2/2018

참고 항목

[NOW 함수](#)

[UTCNOW 함수](#)

WEEKDAY

2021-03-16 • 5 minutes to read

날짜의 요일을 나타내는 1에서 7 사이의 숫자를 반환합니다. 기본적으로 요일은 1(일요일)에서 7(토요일)까지입니다.

구문

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

매개 변수

용어	정의
날짜	datetime 형식의 날짜입니다. 날짜는 DATE 함수를 사용하거나 날짜를 반환하는 식을 사용하거나 기타 수식의 결과로 입력되어야 합니다.
return_type	반환 값을 결정하는 숫자입니다. 반환 형식: 1 이면 한 주가 일요일(1)에 시작하여 토요일(7)에 끝납니다. 1에서 7 사이의 번호가 지정됩니다. 반환 형식: 2 이면 한 주가 월요일(1)에 시작하여 일요일(7)에 끝납니다. 반환 형식: 3 이면 한 주가 월요일(0)에 시작하여 일요일(6)에 끝납니다. 1에서 7 사이의 번호가 지정됩니다.

반환 값

1에서 7 사이의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간을 **datetime** 형식으로 처리합니다. 날짜를 일련 번호로 표시해야 하는 경우 Excel의 서식 옵션을 사용할 수 있습니다.
- 날짜에 허용되는 텍스트 표현으로 날짜를 입력할 수도 있지만 예기치 않은 결과를 방지하려면 먼저 텍스트 날짜를 **datetime** 형식으로 변환하는 것이 좋습니다.
- date 인수가 날짜의 텍스트 표현인 경우 이 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 현재 날짜/시간 설정이 날짜를 월/일/연도 형식으로 표현하는 경우 "1/8/2009" 문자열은 2009년 1월 8일에 해당하는 **datetime** 값으로 해석됩니다. 하지만 현재 날짜/시간 설정이 날짜를 일/월/연도 형식으로 표현하는 경우 동일한 문자열이 2009년 8월 1일에 해당하는 **datetime** 값으로 해석됩니다.

예제

다음 예제에서는 [HireDate] 열의 날짜를 가져와서 1을 더하여 해당 날짜를 나타내는 요일을 표시합니다.

return_type 인수가 생략되었으므로 1은 일요일이고 7은 토요일인 기본 형식이 사용됩니다. 결과가 4이면 요

일은 수요일입니다.

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

참고 항목

[날짜 및 시간 함수](#)

[WEEKNUM 함수](#)

[YEARFRAC 함수](#)

WEEKNUM

2021-03-16 • 3 minutes to read

return_type 값에 따라, 지정된 날짜와 연도의 주 번호를 반환합니다. 주 번호는 연도 내에서 주를 번호로 표현할 경우 몇 번째에 해당하는지를 나타냅니다.

구문

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

매개 변수

용어	정의
날짜	datetime 형식의 날짜입니다.
return_type	<p>반환 값을 결정하는 수입니다. 주가 일요일에 시작하는 경우 1을 사용하고, 주가 월요일에 시작하는 경우 2를 사용합니다. 기본값은 1입니다.</p> <p>반환 형식: 1 은 주가 일요일에 시작합니다. 요일은 1부터 7 까지 번호가 지정됩니다.</p> <p>반환 형식: 2 는 주가 월요일에 시작합니다. 요일은 1부터 7 까지 번호가 지정됩니다.</p>

반환 값

정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 **datetime** 데이터 형식을 사용하여 날짜 및 시간으로 작업합니다. 원본 데이터가 다른 형식인 경우 DAX는 암시적으로 데이터를 **datetime** 으로 변환하여 계산을 수행합니다.
- 기본적으로 WEEKNUM 함수는 1월 1일이 포함된 주가 연도의 첫 번째 주로 간주되는 달력 규칙을 사용합니다. 하지만 유럽에서 많이 사용되는 ISO 8601 달력 표준에서는 다수 일수(4일 이상)가 새 연도에 속하는 주를 첫 번째 주로 정의합니다. 따라서 1월 첫 번째 주에 3일 이하의 일수가 포함된 연도에는 WEEKNUM 함수가 ISO 8601 정의와 다른 주 번호를 반환합니다.

예제 1

다음 예제에서는 2010년 2월 14일의 주 번호를 반환합니다.

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

예 2

다음 예제에서는 **Employees** 테이블에 있는 **HireDate** 열에 저장된 날짜의 주 번호를 반환합니다.

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

참고 항목

[날짜 및 시간 함수](#)

[YEARFRAC 함수](#)

[WEEKDAY 함수](#)

YEAR

2021-03-16 • 3 minutes to read

날짜의 연도를 1900~9999 범위의 4자리 정수로 반환합니다.

구문

YEAR(<date>)

매개 변수

용어	정의
date	찾으려는 연도가 포함된 datetime 또는 text 형식의 날짜입니다.

반환 값

1900~9999 범위의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 **datetime** 데이터 형식을 사용하여 날짜 및 시간으로 작업합니다.
- 날짜는 DATE 함수를 사용하거나 기타 수식 또는 함수의 결과로 입력되어야 합니다. 2007년 3월 3일 또는 2003-3-3과 같이 날짜에 대해 허용되는 텍스트 표현으로 날짜를 입력할 수도 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공된 날짜 값의 표시 형식과 관계없이 양력 값입니다. 예를 들어 제공된 날짜의 표시 형식이 회교식 달력인 경우 YEAR, MONTH 및 DAY 함수에 대해 반환되는 값은 해당하는 양력 날짜와 연결된 값입니다.
- date 인수가 날짜의 텍스트 표현인 경우, 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜 시간 설정을 사용하여 텍스트 값을 파싱합니다. 문자열 형식이 현재 로캘 설정과 호환되지 않는 경우 오류가 발생할 수 있습니다. 예를 들어 날짜가 월/일/연도로 형식 지정되도록 로캘에 정의되어 있으나 날짜가 일/월/연도로 입력되면 25/1/2009는 2009년 1월 25일이 아니라 잘못된 날짜로 해석됩니다.

예제

다음 예제에서는 2007을 반환합니다.

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

예 - 식의 결과인 날짜

Description

다음 예제에서는 오늘 날짜의 연도를 반환합니다.

= YEAR(TODAY())

참고 항목

[날짜 및 시간 함수](#)

[HOUR 함수](#)

[MINUTE 함수](#)

[YEAR 함수](#)

[SECOND 함수](#)

YEARFRAC

2021-03-16 • 4 minutes to read

두 날짜 사이의 전체 일수로 표시되는 연도의 비율을 계산합니다. YEARFRAC 워크시트 함수를 사용하여 특정 기간에 할당할 전체 연도의 혜택 또는 의무의 비율을 식별합니다.

구문

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

매개 변수

용어	정의
start_date	datetime 형식의 시작 날짜입니다.
end_date	datetime 형식의 종료 날짜입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. 모든 인수는 정수로 잘립니다. 기준 - 설명 0 - 미국(NASD) 30/360 1 - 실제/실제 2 - 실제/360 3 - 실제/365 4 - 유럽 30/360

반환 값

10진수입니다. 내부 데이터 형식은 부호 있는 IEEE 64비트(8바이트) 배정밀도 부동 소수점 숫자입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간으로 작업하는 데 **datetime** 형식을 사용합니다. 날짜를 일련 번호로 표시해야 하는 경우 Excel의 서식 지정 옵션을 사용할 수 있습니다.
- start_date** 또는 **end_date** 가 유효한 날짜가 아닌 경우 YEARFRAC는 오류를 반환합니다.
- basis** < 0 또는 **basis** > 4인 경우 YEARFRAC는 오류를 반환합니다.

예 1

다음 예에서는 두 개의 열, `TransactionDate` 및 `ShippingDate` 에 있는 날짜 간 차이로 표시되는 연도의 비율을 반환합니다.


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

예제 2

다음 예에서는 1월 1일과 3월 1일 간 차이로 표시되는 연도의 비율을 반환합니다.

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

예기치 않은 결과가 발생하지 않도록 하려면 가능한 한 네 자리 연도를 사용하세요. 연도가 잘리는 경우 현재 연도를 가정합니다. 날짜가 생략된 경우 해당 월의 첫 번째 날짜를 가정합니다.

두 번째 인수인 **basis** 도 생략되었습니다. 따라서 연간 비율은 미국(NASD) 30/360 표준에 따라 계산됩니다.

참고 항목

[날짜 및 시간 함수](#)

[WEEKNUM 함수](#)

[YEARFRAC 함수](#)

[WEEKDAY 함수](#)

필터 함수

2021-03-16 • 4 minutes to read

DAX의 필터 및 값 함수는 가장 복잡하고 강력한 함수에 속하며 Excel 함수와 크게 다릅니다. 조회 함수는 데이터 베이스와 같이 테이블 및 관계를 사용하여 작동됩니다. 필터링 함수를 사용하면 데이터 컨텍스트를 조작하여 동적 계산을 만들 수 있습니다.

이 범주의 함수

함수	DESCRIPTION
ALL	적용되었을 수 있는 필터를 무시하고 테이블의 모든 행 또는 열의 모든 값을 반환합니다.
ALLCROSSFILTERED	테이블에 적용된 필터를 모두 지웁니다.
ALLEXCEPT	지정된 열에 적용된 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다.
ALLNOBLANKROW	관계의 부모 테이블에서 빈 행을 제외한 모든 행을 반환하거나 빈 행을 제외한 열의 모든 고유 값을 반환하고, 존재할 수 있는 모든 컨텍스트 필터를 무시합니다.
ALLSELECTED	현재 쿼리의 열 및 행에서 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터 또는 명시적 필터는 그대로 유지합니다.
CALCULATE	수정된 필터 컨텍스트에서 식을 평가합니다.
CALCULATETABLE	수정된 필터 컨텍스트에서 테이블 식을 평가합니다.
EARLIER	언급된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.
EARLIEST	지정된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.
FILTER	다른 테이블이나 식의 하위 집합을 나타내는 테이블을 반환합니다.
KEEPFILTERS	CALCULATE 또는 CALCULATETABLE 함수를 계산하는 동안 필터가 적용되는 방식을 수정합니다.
LOOKUPVALUE	검색 조건에 지정된 모든 조건을 충족하는 행의 값을 반환합니다. 함수는 하나 이상의 검색 조건을 적용할 수 있습니다.
REMOVEFILTERS	지정된 테이블 또는 열에서 필터를 지웁니다.
SELECTEDVALUE	columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 해당 값을 반환합니다. 그 외의 경우 alternateResult를 반환합니다.

ALL

2021-03-16 • 20 minutes to read

적용되었을 수 있는 필터를 무시하고 테이블의 모든 행 또는 열의 모든 값을 반환합니다. 이 함수는 필터를 지우고 테이블의 모든 행에 대한 계산을 만들 때 유용합니다.

구문

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

매개 변수

용어	정의
테이블	필터를 지우려는 테이블입니다.
열	필터를 지우려는 열입니다.

ALL 함수의 인수는 기본 테이블에 대한 참조 또는 기본 열에 대한 참조여야 합니다. ALL 함수에는 테이블 식 또는 열 식을 사용할 수 없습니다.

반환 값

필터가 제거된 테이블 또는 열입니다.

설명

- 이 함수는 단독으로 사용되지 않지만 일부 다른 계산을 수행하는 결과 세트를 변경하는 데 사용할 수 있는 중간 함수 역할을 합니다.
- ALL() 함수가 포함된 DAX 식의 일반적인 동작은 적용된 필터가 무시된다는 것입니다. 그러나 특정 DAX 쿼리에 필요한 처리량을 줄이기 위해 필터링을 최적화하는 DAX 기술인 *auto-exist* 때문에 그렇지 않은 경우가 있습니다. auto-exist와 ALL()이 예기치 않은 결과를 제공하는 예로는 슬라이서를 사용하는 경우처럼 동일한 테이블의 두 개 이상의 열을 필터링하는 경우와 ALL()을 사용하는 동일한 테이블에 측정값이 있는 경우입니다. 이 경우 auto-exist는 여러 필터를 하나로 *병합*하고 기존의 값 조합만 필터링합니다. 이 병합으로 인해 기존 값 조합에서 측정값이 계산되고 결과는 예상대로 모든 값이 아니라 필터링된 값을 기반으로 합니다. auto-exist와 계산에 미치는 영향에 대해 자세히 알아보려면 sql.bi.com에 있는 Microsoft MVP Alberto Ferrari의 [Understanding DAX Auto-Exist](#)(DAX Auto-Exist 이해) 문서를 참조하세요.
- 다음 표는 여러 시나리오에서 ALL 및 ALLEXCEPT 함수를 사용하는 방법을 설명합니다.

함수 및 사용법	설명
ALL()	모든 위치에 있는 모든 필터를 제거합니다. ALL()은 테이블 반환이 아닌 필터 지우기 용도로만 사용될 수 있습니다.

함수 및 사용법	설명
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다. 실제로 ALL(Table)은 필터가 적용되었을 수 있는 컨텍스트에서 모든 필터를 제거하여, 테이블의 모든 값을 반환합니다. 이 함수는 여러 그룹화 수준을 사용 중이고, 전체 값 대비 집계 값의 비율을 가져오는 계산을 만들 때 유용합니다. 첫 번째 예제는 이러한 시나리오를 보여줍니다.
ALL (Column[, Column[, ...]])	테이블의 지정된 열에서 모든 필터를 제거합니다. 테이블의 다른 열에 있는 다른 모든 필터는 여전히 적용됩니다. 모든 열 인수는 같은 테이블에서 가져와야 합니다. ALL(Column) 변형은 하나 이상의 특정 열에서 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터를 유지하려는 경우에 유용합니다. 두 번째 및 세 번째 예제는 이 시나리오를 보여줍니다.
ALLEXCEPT(Table, Column1 [,Column2]...)	지정된 열에 적용된 필터를 제외하고 테이블에 있는 모든 컨텍스트 필터를 제거합니다. 이는 테이블의 여러 열 (모든 열이 아님)에서 필터를 제거하려는 경우에 편리한 바로 가기입니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

총 판매액 대비 범주 판매액 비율을 계산합니다.

여기에서는 피벗 테이블에서 모든 재판매인에 대한 총 판매액으로 분할된 현재 셀에 대한 판매액을 찾는다고 가정합니다. 피벗 테이블 사용자가 데이터를 필터링하거나 그룹화하는 방법에 관계없이 분모가 동일한지 확인하기 위해서는 ALL을 사용하여 올바른 총계를 만드는 수식을 정의합니다.

다음 표에서는 새로운 측정값인 **전체 재판매인 판매 비율** 이 코드 섹션에 표시된 수식을 사용하여 생성되었을 때의 결과를 보여줍니다. 작동 방식을 보려면 CalendarYear 필드를 피벗 테이블의 **행 레이블** 영역에 추가하고 ProductCategoryName 필드를 **열 레이블** 영역에 추가합니다. 그런 다음, 측정값 **전체 재판매인 판매 비율** 을 피벗 테이블의 **값** 영역으로 끌어옵니다. 결과를 백분율로 보려면 Excel의 서식 지정 기능을 사용해서 측정값이 포함된 셀에 백분율 숫자 형식을 적용합니다.

행 레이블	액세서리	자전거	의류	구성 요소	총합계
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%
총합계	0.70%	82.47%	2.18%	14.65%	100.00%

수식

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

이 수식은 다음과 같이 구성됩니다.

1. 분자 `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])` 는 피벗 테이블에서 현재 셀에 대해 CalendarYear 및 ProductCategoryName에 컨텍스트 필터가 적용된 ResellerSales_USD[SalesAmount_USD]에 있는 값의 합계입니다.
2. 분모에 대해서는 먼저 테이블 ResellerSales_USD를 지정하고 ALL 함수를 사용해서 테이블에서 모든 컨텍스트 필터를 제거할 수 있습니다.
3. 그런 다음, SUMX 함수를 사용하여 ResellerSales_USD[SalesAmount_USD] 열의 값을 합산합니다. 즉, 모든 재판매인 판매에 대해 ResellerSales_USD[SalesAmount_USD] 합계를 가져옵니다.

예제 2

현재 연도의 총 판매액 대비 제품 판매 비율 계산

여기에서는 각 제품 범주(ProductCategoryName)에 대해 몇 년 동안의 판매액 비율을 보여주는 테이블을 만든다고 가정합니다. ProductCategoryName의 각 값에 대해 각 연도별 백분율을 가져오려면 해당 연도 및 제품 범주의 판매액 합계를 모든 연도의 동일 제품 범주에 대한 판매액 합계로 나뉘야 합니다. 즉, 백분율의 분모를 계산할 때 ProductCategoryName에 대해서는 필터를 유지하고, 연도에 대한 필터는 제거해야 합니다.

다음 표에서는 코드 섹션에 표시된 수식을 사용하여 새로운 측정값 **재판매인 판매 연도** 를 만들 때의 결과를 보여줍니다. 작동 방식을 보려면 CalendarYear 필드를 피벗 테이블의 **행 레이블** 영역에 추가하고 ProductCategoryName 필드를 **열 레이블** 영역에 추가합니다. 결과를 백분율로 보려면 Excel의 서식 지정 기능을 사용해서 **재판매인 판매 연도** 측정값이 포함된 셀에 백분율 숫자 형식을 적용합니다.

행 레이블	액세서리	자전거	의류	구성 요소	총합계
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%
총합계	100.00%	100.00%	100.00%	100.00%	100.00%

수식

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

이 수식은 다음과 같이 구성됩니다.

1. 분자 `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])` 는 피벗 테이블에서 현재 셀에 대해 CalendarYear 및 ProductCategoryName 열에 컨텍스트 필터가 적용된 ResellerSales_USD[SalesAmount_USD]에 있는 값의 합계입니다.
2. 분모에 대해 ALL(Column) 함수를 사용해서 CalendarYear에 대한 기존 필터를 제거합니다. 그러면 열 레이블에서 기존 컨텍스트 필터를 적용한 후 ResellerSales_USD 테이블에 있는 남은 행에 대한 합계를 계산합니다. 최종적으로 분모에 대해서는 선택된 ProductCategoryName(암시적 컨텍스트 필터)에 대해 그리고 해당 연도의 모든 값에 대해 합계가 계산됩니다

예제 3

연도별 전체 판매액에 대한 제품 범주 기여도 계산

여기에서는 연도별 기준으로 각 제품 범주에 대한 판매 백분율을 표시하는 테이블을 만든다고 가정합니다. 특정 연도에서 각 제품 범주의 백분율을 가져오려면 n 연도에 특정 제품 범주(ProductCategoryName)에 대한 판매 합계를 계산한 후 결과 값을 모든 제품 범주에 대한 n 연도의 판매 합계로 나뉘어야 합니다. 즉, 백분율의 분모를 계산할 때 연도에 대해서는 필터를 유지하고, ProductCategoryName에 대한 필터는 제거해야 합니다.

다음 표에서는 코드 섹션에 표시된 수식을 사용하여 새로운 측정값 **재판매인 판매 범주 이름** 을 만들 때의 결과를 보여줍니다. 작동 방식을 보려면 CalendarYear 필드를 피벗 테이블의 **행 레이블** 영역에 추가하고 ProductCategoryName 필드를 **열 레이블** 영역에 추가합니다. 그런 다음, 피벗 테이블의 **값** 영역에 새 측정값을 추가합니다. 결과를 백분율로 보려면 Excel의 서식 지정 기능을 사용해서 새로운 측정값인 **재판매인 판매 범주 이름** 이 포함된 셀에 백분율 숫자 형식을 적용합니다.

행 레이블	액세서리	자전거	의류	구성 요소	총합계
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%
총합계	0.70%	82.47%	2.18%	14.65%	100.00%

수식

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

이 수식은 다음과 같이 구성됩니다.

- 분자 `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])` 는 피벗 테이블의 현재 셀에 대해 CalendarYear 및 ProductCategoryName 필드에 컨텍스트 필터가 적용된 ResellerSales_USD[SalesAmount_USD]에 있는 값의 합계입니다.
- 분모에 대해서는 행 레이블의 기존 컨텍스트 필터를 적용한 후 ALL(Column) 함수를 사용하여 ProductCategoryName에서 필터를 제거하고 ResellerSales_USD 테이블에서 남은 행에 대한 합계를 계산합니다. 최종적으로 분모에 대해서는 선택된 연도(암시적 컨텍스트 필터)에 대해 그리고 ProductCategoryName의 모든 값에 대해 합계가 계산됩니다.

참고 항목

[필터 함수](#)

[ALL 함수](#)

[ALLEXCEPT 함수](#)

[FILTER 함수](#)

ALLCROSSFILTERED

2021-03-16 • 2 minutes to read

테이블에 적용된 필터를 모두 지웁니다.

구문

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

매개 변수

용어	정의
테이블	필터를 지우려는 테이블입니다.

반환 값

해당 없음. 설명을 참조하세요.

설명

- ALLCROSSFILTERED는 필터를 지우는 데만 사용할 수 있고 테이블을 반환하는 데는 사용할 수 없습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
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]
```

그러면 다음 값이 반환됩니다.

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
데모 이벤트	60398	
잡지 광고	60398	
제조업체	60398	1818

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
프로모션 시	60398	7390
기타	60398	3653
가격	60398	47733
품질	60398	1551
검토	60398	1640
스폰서쉽	60398	
텔레비전 광고	60398	730

NOTE

FactInternetSales 테이블과 DimSalesReason 테이블 간에는 직접 또는 간접 다대다 관계가 있습니다.

ALLEXCEPT

2021-03-16 • 7 minutes to read

지정된 열에 적용된 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다.

구문

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

매개 변수

용어	정의
테이블	후속 인수에 지정된 열에 대한 필터를 제외하고 모든 컨텍스트 필터가 제거되는 테이블입니다.
열	컨텍스트 필터를 유지해야 하는 열입니다.

ALLEXCEPT 함수의 첫 번째 인수는 기본 테이블에 대한 참조여야 합니다. 모든 후속 인수는 기본 열에 대한 참조여야 합니다. 테이블 식이나 열 식은 ALLEXCEPT 함수와 함께 사용할 수 없습니다.

반환 값

지정된 열에 대한 필터를 제외하고 모든 필터가 제거된 테이블입니다.

설명

- 이 함수는 단독으로 사용되지 않지만 일부 다른 계산을 수행하는 결과 세트를 변경하는 데 사용할 수 있는 중간 함수 역할을 합니다.
- 다음 표에 설명된 것처럼 여러 시나리오에서 ALL 및 ALLEXCEPT 함수를 사용할 수 있습니다.

함수 및 사용법	DESCRIPTION
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다. 실제로 ALL(Table)은 필터가 적용되었을 수 있는 컨텍스트에서 모든 필터를 제거하여, 테이블의 모든 값을 반환합니다. 이 함수는 여러 그룹화 수준을 사용 중이고, 전체 값 대비 집계 값의 비율을 가져오는 계산을 만들 때 유용합니다.
ALL (Column[, Column[, ...]])	테이블의 지정된 열에서 모든 필터를 제거합니다. 테이블의 다른 열에 있는 다른 모든 필터는 여전히 적용됩니다. 모든 열 인수는 같은 테이블에서 가져와야 합니다. ALL(Column) 변형은 하나 이상의 특정 열에서 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터를 유지하려는 경우에 유용합니다.
ALLEXCEPT(Table, Column1 [,Column2]...)	지정된 열에 적용된 필터를 제외하고 테이블에 있는 모든 컨텍스트 필터를 제거합니다. 이는 테이블의 여러 열 (모든 열이 아님)에서 필터를 제거하려는 경우에 편리한 바로 가기입니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 측정값에 사용할 수 있는 수식을 보여 줍니다.

이 수식은 SalesAmount_USD를 합산하고 ALLEXCEPT 함수를 사용하여, 필터가 CalendarYear 열에 적용된 경우를 제외하고 DateTime 테이블에서 컨텍스트 필터를 제거합니다.

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

수식은 ALLEXCEPT를 사용하므로 DateTime 테이블에서 CalendarYear를 제외한 모든 열이 피벗 테이블을 슬라이스하는 데 사용될 때마다 수식은 모든 슬라이서 필터를 제거하여 표 1에 나와 있는 것처럼 열 레이블 값에 SalesAmount_USD의 합과 동일한 값을 제공합니다.

그러나 CalendarYear 열을 사용하여 피벗 테이블을 슬라이스하는 경우에는 결과가 다릅니다. CalendarYear는 ALLEXCEPT에 대한 인수로 지정되므로 연도에 데이터가 슬라이스되면 표 2에 나와 있는 것처럼 행 수준에서 필터가 연도에 적용됩니다. 사용자는 이러한 테이블을 비교하여 ALLEXCEPT()의 동작을 파악하는 것이 좋습니다.

참고 항목

[필터 함수](#)

[ALL 함수](#)

[FILTER 함수](#)

ALLNOBLANKROW

2021-03-16 • 12 minutes to read

관계의 부모 테이블에서 빈 행을 제외한 모든 행을 반환하거나 빈 행을 제외한 열의 모든 고유 값을 반환하고, 존재할 수 있는 모든 컨텍스트 필터를 무시합니다.

구문

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

매개 변수

용어	정의
테이블	모든 컨텍스트 필터가 제거된 테이블입니다.
열	모든 컨텍스트 필터가 제거된 열입니다.

매개 변수 하나만 전달해야 합니다. 매개 변수는 테이블 또는 열입니다.

반환 값

전달된 매개 변수가 테이블인 경우에는 테이블이고, 전달된 매개 변수가 열인 경우에는 값이 포함된 열입니다.

설명

- ALLNOBLANKROW 함수는 자식 테이블에 부모 열에 대해 일치하지 않는 값이 포함된 행이 하나 이상 있는 관계에서 부모 테이블이 표시하는 빈 행만 필터링합니다. 자세한 설명은 아래 예제를 참조하세요.
- 다음 표에서는 DAX에 제공된 ALL의 변형 및 이러한 각 변형의 차이점을 요약해서 보여줍니다.

함수 및 사용법	설명
ALL(Column)	테이블에 있는 지정된 열에서 모든 필터를 제거합니다. 다른 열에 대한 테이블의 다른 모든 필터는 계속 적용됩니다.
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다.
ALLEXCEPT(Table,Col1,Col2...)	지정된 열을 제외하고 테이블에 있는 모든 컨텍스트 필터를 재정의합니다.
ALLNOBLANK(table column)	관계의 부모 테이블에서 빈 행을 제외한 모든 행을 반환하거나 빈 행을 제외한 열의 모든 고유 값을 반환하고, 존재할 수 있는 모든 컨텍스트 필터를 무시합니다.

ALL 함수 작동 방법에 대한 일반적인 설명과 ALL(Table) 및 ALL(Column)을 사용하는 단계별 예제는 [ALL 함수](#)를 참조하세요.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

샘플 데이터에서 ResellerSales_USD 테이블에는 값이 없고, 따라서 통합 문서 내의 관계에서 어떤 부모 테이블과도 연관될 수 없는 하나의 행이 포함되어 있습니다. 피벗 테이블에서 이 테이블을 사용하여 빈 행 동작을 보고 관련되지 않은 데이터에서 카운트를 처리하는 방법을 확인할 수 있습니다.

1단계: 관련 없는 데이터 확인

파워 피벗 창을 연 다음, ResellerSales_USD 테이블을 선택합니다. ProductKey 열에서 빈 값을 필터링합니다. 하나의 행이 남습니다. 이 행에서 SalesOrderLineNumber를 제외하고 모든 열 값이 비어 있어야 합니다.

2단계: 피벗 테이블 만들기

새 피벗 테이블을 만든 후 datetime.[Calendar Year] 열을 행 레이블 창으로 끌어옵니다. 다음 표에서는 예상 결과를 보여줍니다.

행 레이블
2005
2006
2007
2008
총합계

2008 과 총합계 사이의 빈 레이블에 주의하세요. 이 빈 레이블은 부모 테이블에 일치하는 값이 없는 자식 테이블의 모든 값을 고려하기 위해 생성된 특별 그룹인 알 수 없는 멤버를 나타냅니다(이 예에서는 datetime.[Calendar Year] 열).

피벗 테이블에서 이 빈 레이블이 표시되면 datetime.[Calendar Year] 열에 연결된 일부 테이블에 빈 값 또는 일치하지 않는 값이 있다는 것을 알 수 있습니다. 부모 테이블에 빈 레이블이 표시되지만, 일치하지 않는 행이 하나 이상의 자식 테이블에 있습니다.

이 빈 레이블 그룹에 추가되는 행은 부모 테이블에 있는 값과 일치하지 않는 값(예: 날짜/시간 테이블에 존재하지 않는 날짜)이거나 null 값일 수 있습니다(날짜에 대한 값이 전혀 없음). 이 예에서는 자식 판매 테이블의 모든 열에 빈 값을 두었습니다. 자식 테이블보다 부모 테이블에 더 많은 값을 포함해도 문제가 되지 않습니다.

3단계: ALL 및 ALLNOBLANK를 사용하여 행 계산

테이블 행을 계산하기 위해 datetime 테이블에 다음 두 개의 측정값을 추가합니다. **datetime의 Countrows ALLNOBLANK**, **datetime의 Countrows ALL**. 이러한 측정값을 정의하기 위해 사용할 수 있는 수식은 다음과 같습니다.

```
// 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'))
```

빈 피벗 테이블에서 행 레이블에 datetime.[Calendar Year] 열을 추가한 다음, 새로 만든 측정값을 추가합니다. 결과는 다음 테이블과 비슷하게 표시됩니다.

행 레이블	DATETIME의 COUNTROWS ALLNOBLANK	DATETIME의 COUNTROWS ALL
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
총합계	1280	1281

결과는 테이블 행 개수에서 1 행의 차이를 보여줍니다. 하지만 **파워 피벗 창** 을 열고 datetime 테이블을 선택하면 여기에 언급된 특별한 빈 행이 알 수 없는 멤버이기 때문에 테이블에서 빈 행을 찾을 수 없습니다.

4단계: 개수가 정확한지 확인

ALLNOBLANKROW가 실제로 비어 있는 행을 계산하지 않고, 부모 테이블에서 특별한 빈 행만 처리하는지 증명하기 위해 ResellerSales_USD 테이블에 다음 두 가지 측정값을 추가합니다. **ResellerSales_USD의 Countrows ALLNOBLANKROW, ResellerSales_USD의 Countrows ALL.**

새 피벗 테이블을 만들고 datetime.[Calendar Year] 열을 행 레이블 창으로 끌어옵니다. 이제 바로 전에 만든 측정값을 추가합니다. 결과는 다음과 비슷하게 표시됩니다.

행 레이블	RESELLERSALES_USD의 COUNTROWS ALLNOBLANKROW	RESELLERSALES_USD의 COUNTROWS ALL
2005	60856	60856
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856
총합계	60856	60856

이제 두 개의 측정값에 같은 결과가 포함됩니다. 그 이유는 관계에 포함된 하나 이상의 자식 테이블에 일치하지 않는 값 또는 빈 값이 포함되어 있을 때 ALLNOBLANKROW 함수가 테이블에서 실제로 비어 있는 행은 계산하지 않고 부모 테이블에서 생성된 특별한 사례의 빈 행만 처리하기 때문입니다.

참고 항목

[필터 함수](#)

[ALL 함수](#)

[FILTER 함수](#)

ALLSELECTED

2021-03-16 • 10 minutes to read

현재 쿼리의 열 및 행에서 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터 또는 명시적 필터는 그대로 유지합니다.

ALLSELECTED 함수는 행 및 열 필터 이외의 명시적 필터 및 컨텍스트를 유지하면서 쿼리의 모든 행과 열을 나타내는 컨텍스트를 가져옵니다. 이 함수는 쿼리에서 보이는 값 합계를 가져오는 데 사용할 수 있습니다.

구문

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

매개 변수

용어	정의
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 이 매개 변수는 식이 될 수 없습니다. 이 매개 변수는 선택적 요소입니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식을 사용할 수 없습니다. 이 매개 변수는 선택적 요소입니다.

반환 값

열 및 행 필터를 사용하지 않는 쿼리의 컨텍스트입니다.

설명

- 인수가 하나인 경우 인수는 *tableName* 또는 *columnName*입니다. 인수가 둘 이상인 경우에는 동일한 테이블의 열이어야 합니다.
- 이 함수는 쿼리 내에서 명시적으로 설정된 모든 필터를 유지하고 행 및 열 필터 이외의 모든 컨텍스트 필터를 유지하기 때문에 ALL()과 다릅니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 DAX 식을 사용하여 테이블 보고서에서 서로 다른 수준의 보이는 값 합계를 생성하는 방법을 보여 줍니다. 보고서에서 이전 필터 2개가 Reseller Sales 데이터에 적용되었습니다. 각각 Sales Territory Group = Europe 및 Promotion Type = Volume Discount입니다. 필터가 적용되면 전체 보고서, 모든 연도 또는 모든 제품 범주에 대해 보이는 값 합계를 계산할 수 있습니다. 또한 설명을 위해 모든 Reseller Sales에 대한 총계를 가져와 보고서의 모든 필터를 제거합니다. 다음 DAX 식을 계산하면 보이는 값이 포함된 테이블을 작성하는 데 필요한 모든 정보가 포함된 테이블이 생성됩니다.

```

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]

```

SQL Server Management Studio에서 AdventureWorks DW 테이블 형식 모델에 대해 위의 식을 실행하면 다음과 같은 결과를 얻을 수 있습니다.

[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]
액세서리	2000		80450596.9823	877006.7987	38786.018	
액세서리	2001		80450596.9823	877006.7987	38786.018	
액세서리	2002	625.7933	80450596.9823	877006.7987	38786.018	91495.3104
액세서리	2003	26037.3132	80450596.9823	877006.7987	38786.018	572927.0136
액세서리	2004	12122.9115	80450596.9823	877006.7987	38786.018	212584.4747
액세서리	2005		80450596.9823	877006.7987	38786.018	

[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]
액세서리	2006		80450596.98 23	877006.7987	38786.018	
자전거	2000		80450596.98 23	877006.7987	689287.7939	
자전거	2001		80450596.98 23	877006.7987	689287.7939	
자전거	2002	73778.938	80450596.98 23	877006.7987	689287.7939	91495.3104
자전거	2003	439771.4136	80450596.98 23	877006.7987	689287.7939	572927.0136
자전거	2004	175737.4423	80450596.98 23	877006.7987	689287.7939	212584.4747
자전거	2005		80450596.98 23	877006.7987	689287.7939	
자전거	2006		80450596.98 23	877006.7987	689287.7939	
의류	2000		80450596.98 23	877006.7987	95090.7757	
의류	2001		80450596.98 23	877006.7987	95090.7757	
의류	2002	12132.4334	80450596.98 23	877006.7987	95090.7757	91495.3104
의류	2003	58234.2214	80450596.98 23	877006.7987	95090.7757	572927.0136
의류	2004	24724.1209	80450596.98 23	877006.7987	95090.7757	212584.4747
의류	2005		80450596.98 23	877006.7987	95090.7757	
의류	2006		80450596.98 23	877006.7987	95090.7757	
구성 요소	2000		80450596.98 23	877006.7987	53842.2111	
구성 요소	2001		80450596.98 23	877006.7987	53842.2111	

[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]
구성 요소	2002	4958.1457	80450596.9823	877006.7987	53842.2111	91495.3104
구성 요소	2003	48884.0654	80450596.9823	877006.7987	53842.2111	572927.0136
구성 요소	2004		80450596.9823	877006.7987	53842.2111	212584.4747
구성 요소	2005		80450596.9823	877006.7987	53842.2111	
구성 요소	2006		80450596.9823	877006.7987	53842.2111	

보고서의 열은 다음과 같습니다.

Reseller Sales Amount

연도 및 제품 범주에 대한 Reseller Sales의 실제 값입니다. 이 값은 연도 및 범주의 교차점에서 보고서 중심의 셀에 표시됩니다.

Reseller Visual Total for All of Calendar Year

모든 연도의 제품 범주에 대한 합계 값입니다. 이 값은 특정 제품 범주 및 보고서의 모든 연도에 대해 열 또는 행의 끝에 표시됩니다.

Reseller Visual Total for All of Product Category Name

모든 제품 범주에 대한 연간 합계 값입니다. 이 값은 특정 연도 및 보고서의 모든 제품 범주에 대해 열 또는 행의 끝에 표시됩니다.

Reseller Visual Total

모든 연도 및 제품 범주에 대한 합계 값입니다. 이 값은 일반적으로 테이블의 맨 오른쪽 아래 모퉁이에 표시됩니다.

Reseller Grand Total

필터가 적용되기 전 모든 재판매인 매출의 총계입니다. [Reseller Visual Total]과의 차이를 확인해야 합니다. 이 보고서에는 두 개(2)의 필터가 포함되어 있습니다. 하나는 제품 범주 그룹에 대한 것이고 다른 하나는 프로모션 유형에 대한 것입니다.

NOTE

식에 명시적 필터가 있는 경우 해당 필터도 식에 적용됩니다.

CALCULATE

2021-03-24 • 12 minutes to read

수정된 필터 컨텍스트에서 식을 평가합니다.

NOTE

[CALCULATE](#) 함수도 있습니다. '테이블 개체'를 반환하는 식에 적용된 [필터 컨텍스트](#)를 수정한다는 점을 제외하면 정확히 동일한 기능을 수행합니다.

구문

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])
```

매개 변수

용어	정의
expression	평가할 식입니다.
filter1, filter2,...	(선택 사항) 필터 또는 필터 한정자 함수를 정의하는 부울 식 또는 테이블 식입니다.

첫 번째 매개 변수로 사용되는 식은 기본적으로 측정값과 동일합니다.

필터는 다음과 같을 수 있습니다.

- 부울 필터 식
- 테이블 필터 식
- 필터 수정 함수

필터가 여러 개 있는 경우 AND(&&) [논리 연산자](#)(모든 조건이 TRUE여야 함) 또는 OR(||) 논리 연산자(두 조건 중 하나가 TRUE여야 함)를 사용하여 필터를 평가할 수 있습니다.

부울 필터 식

부울 식 필터는 TRUE 또는 FALSE로 평가되는 식입니다. 다음과 같은 몇 가지 규칙을 준수해야 합니다.

- 단일 테이블의 열을 참조할 수 있습니다.
- 측정값을 참조할 수 없습니다.
- 중첩된 CALCULATE 함수를 사용할 수 없습니다.
- 집계 함수를 포함하여 테이블을 검색하거나 반환하는 함수를 사용할 수 없습니다.

테이블 필터 식

테이블 식 필터는 테이블 개체를 필터로 적용합니다. 모델 테이블에 대한 참조일 수도 있지만, 테이블 개체를 반환하는 함수일 가능성이 더 높습니다. 부울 필터 식으로 정의할 수 없는 필터를 포함하여 복잡한 필터 조건을 적용하려면 [FILTER](#) 함수를 사용할 수 있습니다.

필터 한정자 함수

필터 수정 함수를 사용하면 단순히 필터를 추가하는 것 이상의 작업을 수행할 수 있습니다. 필터 컨텍스트를 수정할 때 추가 제어 기능을 제공합니다.

함수	용도
REMOVEFILTERS	모든 필터를 제거하거나, 단일 테이블에 있는 하나 이상의 열 또는 모든 열에서 필터를 제거합니다.
ALL ¹ , ALLEXCEPT , ALLNOBLANKROW	하나 이상의 열 또는 단일 테이블의 모든 열에서 필터를 제거합니다.
KEEPFILTERS	같은 열에서 기존 필터를 제거하지 않은 채 필터를 추가합니다.
USERELATIONSHIP	관련된 열 간에 비활성 관계로 전환합니다. 그러면 활성 관계가 자동으로 비활성화됩니다.
CROSSFILTER	필터 방향을 수정하거나(둘 다에서 하나로 또는 하나에서 둘 다로) 관계를 사용하지 않도록 설정합니다.

¹ ALL 함수와 해당 변형은 필터 한정자로서도 작동하고 테이블 개체를 반환하는 함수로서도 작동합니다. 도구에서 REMOVEFILTERS 함수를 지원하는 경우 이 함수를 사용하여 필터를 제거하는 것이 더 좋습니다.

반환 값

식의 결과 값입니다.

설명

- 필터 식이 제공되면 CALCULATE 함수는 식을 평가하기 위해 필터 컨텍스트를 수정합니다. 필터 식이 KEEPFILTERS 함수에서 래핑되지 않은 경우 각 필터 식에 대해 다음과 같은 두 가지 표준 결과가 나타날 수 있습니다.
 - 열(또는 테이블)이 필터 컨텍스트에 없으면 식을 평가하기 위해 새 필터가 필터 컨텍스트에 추가됩니다.
 - 열(또는 테이블)이 필터 컨텍스트에 이미 있는 경우 CALCULATE 식을 평가하기 위해 새 필터로 기존 필터를 덮어씁니다.
- ‘필터 없이’ 사용되는 CALCULATE 함수는 특정 요구 사항을 실현합니다. 행 컨텍스트를 필터 컨텍스트로 전환합니다. 모델 데이터를 요약하는 식(모델 측정값이 아님)을 행 컨텍스트에서 평가해야 하는 경우 필요합니다. 이 시나리오는 계산 열 수식에서 발생하거나 반복기 함수의 식이 평가될 때 발생할 수 있습니다. 행 컨텍스트에서 모델 측정값을 사용하는 경우에는 컨텍스트 전환이 자동으로 진행됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

다음의 **Sales** 테이블 측정값 정의는 파란색으로 된 제품에만 수익 결과를 생성합니다.

이 문서의 예는 *Power BI Desktop* 샘플 모델에 추가할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

```
Blue Revenue =
CALCULATE(
    SUM(Sales[Sales Amount]),
    'Product'[Color] = "Blue"
)
```

범 주	SALES AMOUNT	BLUE REVENUE
Accessories	\$1,272,057.89	\$165,406.62
Bikes	\$94,620,526.21	\$8,374,313.88
Clothing	\$2,117,613.45	\$259,488.37
구성 요소	\$11,799,076.66	\$803,642.10
합계	\$109,809,274.20	\$9,602,850.97

CALCULATE 함수는 지정된 필터 컨텍스트에서 **Sales** 테이블 **Sales Amount** 열의 합계를 평가합니다. 새 필터는 **Product** 테이블 **Color** 열에 추가되거나, 열에 이미 적용된 필터를 덮어씁니다.

다음 **Sales** 테이블 측정값 정의는 모든 판매 채널에 대한 매출 대비 수익률을 생성합니다.

채널	SALES AMOUNT	REVENUE % TOTAL CHANNEL
인터넷	\$29,358,677.22	26.74%
재 판매인	\$80,450,596.98	73.26%
합계	\$109,809,274.20	100.00%

```
Revenue % Total Channel =
DIVIDE(
    SUM(Sales[Sales Amount]),
    CALCULATE(
        SUM(Sales[Sales Amount]),
        REMOVEFILTERS('Sales Order'[Channel])
    )
)
```

DIVIDE 함수는 **Sales** 테이블 **Sales Amount** 열 값(필터 컨텍스트에서)의 합계를 수정된 필터 컨텍스트의 동일한 식으로 나눕니다. 필터 한정자 함수인 **REMOVEFILTERS** 함수를 사용하여 필터 컨텍스트를 수정하는 **CALCULATE** 함수입니다. **Sales Order** 테이블 **Channel** 열에서 필터를 제거합니다.

다음 **Customer** 테이블 계산 열 정의는 고객을 충성도 클래스로 분류합니다. 시나리오의 간단합니다. 고객에 의해 창출된 수익이 \$2500 미만이면 *Low* 로 분류되고, 이상이면 *High* 로 분류됩니다.

```
Customer Segment =
IF(
    CALCULATE(SUM(Sales[Sales Amount]), ALLEXCEPT(Customer, Customer[CustomerKey])) < 2500,
    "Low",
    "High"
)
```

이 예제에서는 행 컨텍스트가 필터 컨텍스트로 변환됩니다. 이를 '컨텍스트 전환'이라고 합니다. **ALLEXCEPT** 함수는 **CustomerKey** 열을 제외하고 모든 **Customer** 테이블 열에서 필터를 제거합니다.

참고 항목

[필터 컨텍스트](#)

[행 컨텍스트](#)

CALCULATE TABLE 함수

필터 함수

CALCULATETABLE

2021-03-16 • 8 minutes to read

수정된 필터 컨텍스트에서 테이블 식을 평가합니다.

NOTE

CALCULATE 함수도 있습니다. '스칼라 값'을 반환하는 식에 적용된 **필터 컨텍스트**를 수정한다는 점을 제외하면 정확히 동일한 기능을 수행합니다.

구문

```
CALCULATETABLE(<expression>[, <filter1> [, <filter2> [, ...]])
```

매개 변수

용어	정의
expression	평가할 테이블 식입니다.
filter1, filter2,...	(선택 사항) 필터 또는 필터 한정자 함수를 정의하는 부울 식 또는 테이블 식입니다.

첫 번째 매개 변수로 사용되는 식은 모델 테이블 또는 테이블을 반환하는 함수여야 합니다.

필터는 다음과 같을 수 있습니다.

- 부울 필터 식
- 테이블 필터 식
- 필터 수정 함수

필터가 여러 개 있는 경우 AND **논리 연산자**를 사용하여 평가됩니다. 즉, 모든 조건이 동시에 TRUE여야 합니다.

부울 필터 식

부울 식 필터는 TRUE 또는 FALSE로 평가되는 식입니다. 다음과 같은 몇 가지 규칙을 준수해야 합니다.

- 단일 열만 참조할 수 있습니다.
- 측정값을 참조할 수 없습니다.
- 중첩된 CALCULATE 함수를 사용할 수 없습니다.
- 집계 함수를 포함하여 테이블을 검색하거나 반환하는 함수를 사용할 수 없습니다.

테이블 필터 식

테이블 식 필터는 테이블 개체를 필터로 적용합니다. 모델 테이블에 대한 참조일 수도 있지만, 테이블 개체를 반환하는 함수일 가능성이 더 높습니다. 부울 필터 식으로 정의할 수 없는 필터를 포함하여 복잡한 필터 조건을 적용하려면 **FILTER** 함수를 사용할 수 있습니다.

필터 한정자 함수

필터 수정 함수를 사용하면 단순히 필터를 추가하는 것 이상의 작업을 수행할 수 있습니다. 필터 컨텍스트를 수정할 때 추가 제어 기능을 제공합니다.

함수	용도
REMOVEFILTERS	모든 필터를 제거하거나, 단일 테이블에 있는 하나 이상의 열 또는 모든 열에서 필터를 제거합니다.
ALL ¹ , ALLEXCEPT , ALLNOBLANKROW	하나 이상의 열 또는 단일 테이블의 모든 열에서 필터를 제거합니다.
KEEPFILTERS	같은 열에서 기존 필터를 제거하지 않은 채 필터를 추가합니다.
USERELATIONSHIP	관련된 열 간에 비활성 관계로 전환합니다. 그러면 활성 관계가 자동으로 비활성화됩니다.
CROSSFILTER	필터 방향을 수정하거나(둘 다에서 하나로 또는 하나에서 둘 다로) 관계를 사용하지 않도록 설정합니다.

¹ ALL 함수와 해당 변형은 필터 한정자로서도 작동하고 테이블 개체를 반환하는 함수로서도 작동합니다. 도구에서 REMOVEFILTERS 함수를 지원하는 경우 이 함수를 사용하여 필터를 제거하는 것이 더 좋습니다.

반환 값

값 테이블입니다.

설명

- 필터 식이 제공되면 CALCULATETABLE 함수는 식을 평가하기 위해 필터 컨텍스트를 수정합니다. 필터 식이 KEEPFILTERS 함수에서 래핑되지 않은 경우 각 필터 식에 대해 다음과 같은 두 가지 표준 결과가 나타날 수 있습니다.
 - 열(또는 테이블)이 필터 컨텍스트에 없으면 식을 평가하기 위해 새 필터가 필터 컨텍스트에 추가됩니다.
 - 열(또는 테이블)이 필터 컨텍스트에 이미 있는 경우 CALCULATETABLE 식을 평가하기 위해 새 필터로 기존 필터를 덮어씁니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 CALCULATETABLE 함수를 사용하여 2006년 인터넷 판매 합계를 구합니다. 이 값은 나중에 2006년 전체 판매와 비교하여 인터넷 판매 비율을 계산하는 데 사용합니다.

수식은 다음과 같습니다.

```
= SUMX(
    CALCULATETABLE(
        'InternetSales_USD',
        'DateTime'[CalendarYear] = 2006
    ),
    [SalesAmount_USD]
)
```

결과는 아래 테이블에 나와 있습니다.

행 레이블	INTERNET SALESAMOUNT_USD	CALCULATETABLE 2006년 인터넷 판매	2006년 판매 대비 인터넷 판 매 비율
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
총합계	\$26,054,827.45	\$5,681,440.58	4.59

참고 항목

- 필터 컨텍스트
- CALCULATE 함수(DAX)
- 필터 함수(DAX)

EARLIER

2021-03-16 • 10 minutes to read

언급된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.

EARLIER는 특정 값을 입력으로 사용하여 해당 입력을 기반으로 계산을 생성하려는 중첩된 계산에 유용합니다. Microsoft Excel에서는 현재 행의 컨텍스트 내에서만 이러한 계산을 수행할 수 있습니다. 하지만 DAX에서는 입력 값을 저장한 후 전체 테이블의 데이터를 사용하여 계산을 수행할 수 있습니다.

EARLIER는 계산 열의 컨텍스트에 주로 사용됩니다.

구문

```
EARLIER(<column>, <number>)
```

매개 변수

용어	정의
열	열 또는 열로 확인되는 식입니다.
num	(선택 사항) 외부 평가 패스에 대한 양수입니다. 다음 평가 수준으로의 외부 패스는 1로 나타내고 두 수준의 외부 패스는 2로 나타내는 방식입니다. 생략하는 경우 기본값은 1입니다.

반환 값

number 개의 외부 평가 패스에서 **column** 의 현재 행 값입니다.

예외

오류 설명

설명

- 테이블 검색 시작 전에 행 컨텍스트가 있으면 **EARLIER** 실행에 성공합니다. 그렇지 않으면 오류를 반환합니다.
- EARLIER** 는 이론적으로 해당 열의 전체 행 수에 동일한 수(식의 구문에 따라 다름)를 곱하는 것과 가까운 많은 연산을 수행해야 하므로 함수 성능이 느릴 수 있습니다. 예를 들어 열에 10개 행이 있는 경우 약 100 개의 연산을 수행해야 할 수 있습니다. 100개의 행이 있는 경우 10,000개에 가까운 연산을 수행해야 할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

NOTE

실제로 VertiPaq 메모리 내 분석 엔진은 최적화를 수행하여 실제 계산 수를 줄이지만 재귀가 포함된 수식을 만드는 경우에는 주의해야 합니다.

예제

EARLIER 사용에 대해 설명하려면 순위 값을 계산한 후 다른 계산에서 순위 값을 사용하는 시나리오를 작성해야 합니다.

다음 예제는 각 ProductSubcategory에 대한 총 판매를 보여 주는 간단한 테이블인 **ProductSubcategory** 를 기반으로 합니다.

순위 열이 포함된 최종 테이블은 아래와 같습니다.

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
18	빔 쇼츠	\$156,167.88	18
26	자전거 랙	\$220,720.70	14
27	자전거 스탠드	\$35,628.69	30
28	물병 및 거치대	\$59,342.43	24
5	바텀 브라켓	\$48,643.47	27
6	브레이크	\$62,113.16	23
19	캡	\$47,934.54	28
7	Chains	\$8,847.08	35
29	클리너	\$16,882.62	32
8	크랭크세트	\$191,522.09	15
9	변속기	\$64,965.33	22
30	펜더	\$41,974.10	29
10	포크	\$74,727.66	21
20	장갑	\$228,353.58	12
4	핸들바	\$163,257.06	17
11	헤드세트	\$57,659.99	25
31	헬멧	\$451,192.31	9
32	하이드레이션 팩	\$96,893.78	20

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
21	저지	\$699,429.78	7
33	조명		36
34	잠금	\$15,059.47	33
1	산악용 자전거	\$34,305,864.29	2
12	산악용 프레임	\$4,511,170.68	4
35	패니어		36
13	페달	\$140,422.20	19
36	펌프	\$12,695.18	34
2	도로용 자전거	\$40,551,696.34	1
14	도로용 프레임	\$3,636,398.71	5
15	안장	\$52,526.47	26
22	반바지	\$385,707.80	10
23	양말	\$28,337.85	31
24	타이즈	\$189,179.37	16
37	타이어 및 튜브	\$224,832.81	13
3	여행용 자전거	\$13,334,864.18	3
16	여행용 프레임	\$1,545,344.02	6
25	조끼	\$240,990.04	11
17	바퀴	\$648,240.04	8

순위 값 만들기

행에 있는 지정된 값의 순위 값을 가져오는 한 가지 방법은 동일한 테이블에서 비교 중인 행보다 크거나 작은 값이 있는 행의 수를 계산하는 것입니다. 이 방법은 테이블의 가장 높은 값에 대해 공백 또는 0 값을 반환하지만 같은 값은 순위 값이 동일하며 다음 값(같은 값 다음)의 순위 값은 연속되지 않습니다. 아래 샘플을 참조하세요.

다음 수식을 사용하여 새 계산 열 **SubCategorySalesRanking** 을 만듭니다.

```
= COUNTROWS(FILTER(ProductSubcategory, EARLIER(ProductSubcategory[TotalSubcategorySales])<ProductSubcategory[TotalSubcategorySales]))+1
```

다음 단계에서는 계산 방법을 자세히 설명합니다.

1. **EARLIER** 함수는 테이블의 현재 행에 대해 *TotalSubcategorySales* 값을 가져옵니다. 이 경우 프로세스를 시작하는 것이므로 테이블의 첫 번째 행입니다.
2. **EARLIER**([*TotalSubcategorySales*])는 외부 루프의 현재 행인 \$156,167.88로 평가합니다.
3. **FILTER** 함수는 이제 \$156,167.88(**EARLIER**의 현재 값)보다 큰 *TotalSubcategorySales* 값이 모든 행에 있는 테이블을 반환합니다.
4. **COUNTROWS** 함수는 필터링된 테이블의 행 수를 계산하고 해당 값에 1을 더하여 현재 행의 새 계산 열에 할당합니다. 최상위 순위 값이 공백이 되는 것을 방지하기 위해 1을 더해야 합니다.
5. 계산 열 수식은 다음 행으로 이동하여 1~4단계를 반복합니다. 이 단계는 테이블의 끝에 도달할 때까지 반복됩니다.

EARLIER 함수는 항상 현재 테이블 연산 전에 열의 값을 가져옵니다. 그 전에 루프에서 값을 가져와야 하는 경우 두 번째 인수를 2로 설정합니다.

참고 항목

[EARLIEST 함수](#)

[필터 함수](#)

EARLIEST

2021-03-16 • 2 minutes to read

지정된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.

구문

```
EARLIEST(<column>)
```

매개 변수

용어	정의
열	열에 대한 참조입니다.

반환 값

필터가 제거된 열입니다.

설명

- EARLIEST 함수는 EARLIER와 유사하지만 재귀를 한 수준 더 지정할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

현재 샘플 데이터에서는 이 시나리오를 지원하지 않습니다.

```
= EARLIEST(<column>)
```

참조

[EARLIER 함수](#)

[필터 함수](#)

FILTER

2021-03-16 • 6 minutes to read

다른 테이블이나 식의 하위 집합을 나타내는 테이블을 반환합니다.

구문

```
FILTER(<table>,<filter>)
```

매개 변수

용어	정의
테이블	필터링할 테이블입니다. 테이블은 테이블을 생성하는 식일 수도 있습니다.
filter	테이블의 각 행에 대해 평가할 부울 식입니다. 예를 들어 <code>[Amount] > 0</code> 또는 <code>[Region] = "France"</code> 와 같습니다.

반환 값

필터링된 행만 포함된 테이블입니다.

설명

- FILTER를 사용하여 작업 중인 테이블의 행 수를 줄이고 특정 데이터만 계산에서 사용할 수 있습니다. FILTER는 독립적으로 사용되지 않습니다. table을 인수로 사용해야 하는 다른 함수에 포함된 함수로 사용됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 미국의 판매를 필터링하여 제외하는 측정값을 사용한 후 판매를 연도 및 제품 범주별로 분할하여 미국 이외 지역의 인터넷 판매에 대한 보고서를 만듭니다. 이 측정값을 만들려면 판매 지역을 사용하여 Internet Sales USD 테이블을 필터링한 후 필터링된 테이블을 SUMX 함수에 사용합니다.

이 예제에서

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

식은 미국 판매 지역에 속하는 모든 행을 뺀 Internet Sales의 하위 집합인 테이블을 반환합니다. RELATED 함수는 Internet Sales 테이블의 Territory 키를 SalesTerritory 테이블의 SalesTerritoryCountry에 연결합니다.

다음 테이블에서는 미국 이외 지역 인터넷 판매 측정값에 대한 개념 증명을 볼 수 있습니다. 해당 수식은 아래 코드 섹션에 제공되어 있습니다. 이 테이블은 계산에서 미국 판매를 제외하는 방식으로 전체 인터넷 판매와 미국 이외 지역 인터넷 판매를 비교하여 필터 식이 작동하는 방식을 보여 줍니다.

이 테이블을 다시 만들려면 보고서 또는 PivotTable의 행 레이블 영역에 SalesTerritoryCountry 필드를 추가합니

다.

표 1. 미국의 및 다른 모든 지역의 총 판매 비교

행 레이블	INTERNET SALES	미국 이외 지역 인터넷 판매
오스트레일리아	\$4,999,021.84	\$4,999,021.84
캐나다	\$1,343,109.10	\$1,343,109.10
프랑스	\$2,490,944.57	\$2,490,944.57
독일	\$2,775,195.60	\$2,775,195.60
영국	\$5,057,076.55	\$5,057,076.55
미국	\$9,389,479.79	
총합계	\$26,054,827.45	\$16,665,347.67

최종 보고서 테이블은 미국 이외 지역 인터넷 판매 측정값을 사용하여 피벗 테이블을 만드는 경우 얻는 결과를 보여 줍니다. CalendarYear 필드를 피벗 테이블의 **행 레이블** 영역에 추가하고 ProductCategoryName 필드를 **열 레이블** 영역에 추가합니다.

표 2. 제품 범주별로 미국 이외 지역 판매 비교

행 레이블	액세서리	자전거	의류	총합계
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
총합계	\$384,639.63	\$16,107,890.23	\$172,817.81	\$16,665,347.67

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

참고 항목

[필터 함수](#)

[ALL 함수](#)

[ALLEXCEPT 함수](#)

KEEPFILTERS

2021-03-16 • 9 minutes to read

CALCULATE 또는 CALCULATETABLE 함수를 평가하는 동안 필터가 적용되는 방식을 수정합니다.

구문

```
KEEPFILTERS(<expression>)
```

매개 변수

용어	정의
식	임의의 식입니다.

반환 값

값 테이블입니다.

설명

- KEEPFILTERS는 CALCULATE 및 CALCULATETABLE 함수의 표준 동작을 재정의하기 위해 이러한 함수의 컨텍스트 내에서 사용됩니다.
- 기본적으로 CALCULATE 같은 함수의 필터 인수는 식을 계산하기 위한 컨텍스트로 사용되며, 이에 따라 CALCULATE의 필터 인수가 동일한 열에 대한 기존 필터를 모두 대체합니다. CALCULATE 필터 인수의 영향을 받는 새 컨텍스트는 필터 인수의 일부로 언급된 열의 기존 필터에만 영향을 줍니다. CALCULATE의 인수에 언급되지 않은 열의 필터나 기타 관련 함수는 계속 적용되며 변경되지 않은 상태로 유지됩니다.
- KEEPFILTERS 함수를 사용하면 이 동작을 수정할 수 있습니다. KEEPFILTERS를 사용하면 현재 컨텍스트에 있는 모든 기존 필터가 필터 인수의 열과 비교되며, 이러한 인수의 교집합이 식을 계산하기 위한 컨텍스트로 사용됩니다. 결국 임의의 한 열에 두 인수 세트, 즉 CALCULATE에 사용되는 필터 인수와 KEEPFILTERS 함수의 인수에 사용되는 필터가 모두 적용됩니다. 즉, CALCULATE 필터는 현재 컨텍스트를 대체하지만 KEEPFILTERS는 현재 컨텍스트에 필터를 추가합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 CALCULATE 또는 CALCULATETABLE 수식의 일부로 KEEPFILTERS 함수를 사용하는 일반적인 시나리오를 몇 개 보여 줍니다.

처음 세 개 식은 비교에 사용할 간단한 데이터를 가져옵니다.

- 워싱턴주의 인터넷 판매
- 워싱턴 및 오리곤주의 인터넷 판매(두 주를 합함)
- 워싱턴 및 브리티시 콜롬비아주의 인터넷 판매(두 지역을 합함)

네 번째 식은 워싱턴 및 브리티시 콜롬비아에 대한 필터가 적용된 상태에서 워싱턴 및 오리곤의 인터넷 판매를

계산합니다.

다음 식은 워싱턴 및 오리곤의 인터넷 판매를 계산하지만 KEEPFILTERS를 사용합니다. 워싱턴 및 브리티시 콜롬비아에 대한 필터는 이전 컨텍스트에 포함됩니다.

```
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"  
    )  
  )  
)
```

AdventureWorks DW 샘플 데이터베이스에 대해 이 식을 계산하면 다음과 같은 결과를 얻을 수 있습니다.

열	값
[\$\$ 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

NOTE

위 결과는 교육을 위해 단일 행 대신 테이블로 형식이 지정되었습니다.

먼저 **[\$\$ in WA and OR ??]** 식을 살펴봅니다. 외부 CALCULATE 식에 워싱턴 및 브리티시 콜롬비아에 대한 필터

터가 포함되어 있는데 이 수식으로 어떻게 워싱턴 및 오리곤의 매출 값을 반환할 수 있는지 잘 이해가 가지 않을 것입니다. 'Geography'[State Province Code]의 외부 필터를 재정의하고 자체 필터 인수로 대체하는 것이 CALCULATE의 기본 동작이라는 사실을 기억하면 이해가 될 것입니다(두 필터가 동일한 열에 적용됨).

다음으로, **[\$\$ in WA !!]** 식을 살펴봅니다. 여기서도 인수 필터에 오리곤이 포함되어 있고 외부 CALCULATE 식에 워싱턴 및 브리티시 콜롬비아에 대한 필터가 포함되어 있는데 이 수식으로 어떻게 워싱턴의 매출 값만 반환할 수 있는지 잘 이해가 가지 않을 것입니다. 여기서는 KEEPFILTERS가 CALCULATE의 기본 동작을 수정하고 필터를 더 추가한다는 사실을 기억하면 이해가 될 것입니다. 필터의 교집합이 사용되므로 이제 외부 필터 'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC" 가 필터 인수 'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR" 에 추가됩니다. 두 필터 모두 동일한 열에 적용되므로 결과 필터 'Geography'[State Province Code]="WA" 가 식을 계산할 때 적용되는 필터가 됩니다.

참고 항목

[필터 함수](#)

[CALCULATE 함수](#)

[CALCULATETABLE 함수](#)

LOOKUPVALUE

2021-03-16 • 4 minutes to read

한 개 이상의 검색 조건으로 지정된 모든 기준을 충족하는 행의 값을 반환합니다.

구문

```
LOOKUPVALUE(  
    <result_columnName>,  
    <search_columnName>,  
    <search_value>  
    [, <search2_columnName>, <search2_value>]...  
    [, <alternateResult>]  
)
```

매개 변수

용어	정의
result_columnName	반환하려는 값이 포함된 기존 열의 이름입니다. 식을 사용할 수 없습니다.
search_columnName	기존 열의 이름입니다. result_columnName과 동일한 테이블 또는 관련 테이블에 있을 수 있습니다. 식을 사용할 수 없습니다.
search_value	search_columnName에서 검색할 값입니다.
alternateResult	(선택 사항) result_columnName의 컨텍스트가 0 또는 둘 이상의 고유한 값으로 필터링된 경우 반환되는 값입니다. 제공되지 않은 경우, result_columnName이 0으로 필터링되면 BLANK가 반환되거나 고유한 값이 둘 이상이면 오류가 반환됩니다.

반환 값

search_column 및 search_value 의 모든 쌍에 정확히 일치하는 항목이 있는 행에서 result_column 의 값입니다.

모든 검색 값을 충족하는 일치 항목이 없으면 BLANK 또는 alternateResult(제공된 경우)가 반환됩니다. 즉, 기준 중 일부만 일치하는 경우 함수는 조회 값을 반환하지 않습니다.

검색 값과 일치하는 행이 여러 개 있고 모든 경우에 result_column 값이 동일하면 해당 값이 반환됩니다. 그러나 result_column 에서 다른 값을 반환하는 경우에는 오류 또는 alternateResult(제공된 경우)가 반환됩니다.

설명

- 결과 테이블과 검색 테이블 간에 관계가 있는 경우, 대부분의 경우에는 LOOKUPVALUE 대신 [RELATED](#) 함수를 사용하는 것이 더 효율적이며 더 나은 성능을 제공합니다.
- 함수가 검색 테이블의 행을 반복하기 전에 search_value 및 alternateResult 매개 변수가 계산됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

니다.

예제

이 문서의 예는 *Power BI Desktop* 샘플 모델에 추가할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

Sales 테이블에 정의된 다음 계산 열은 LOOKUPVALUE 함수를 사용하여 **Sales Order** 테이블의 채널 값을 반환합니다.

```
CHANNEL = LOOKUPVALUE('Sales Order'[Channel], 'Sales Order'[SalesOrderLineKey], [SalesOrderLineKey])
```

그러나 이 경우 **Sales Order** 와 **Sales** 테이블 간에 관계가 있으므로 [RELATED](#) 함수를 사용하는 것이 더 효율적입니다.

```
CHANNEL = RELATED('Sales Order'[Channel])
```

참고 항목

[RELATED 함수\(DAX\)](#)

[정보 함수](#)

REMOVEFILTERS

2021-03-16 • 2 minutes to read

지정한 테이블 또는 열에서 필터를 지웁니다.

구문

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

매개 변수

용어	정의
테이블	필터를 지우려는 테이블입니다.
열	필터를 지우려는 열입니다.

반환 값

해당 없음. 설명을 참조하세요.

설명

- REMOVEFILTERS는 필터를 지우는 데만 사용할 수 있고 테이블을 반환하는 데는 사용할 수 없습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제 1

DAX 쿼리

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales], CALCULATE([TotalSales], REMOVEFILTERS()))

EVALUATE
    SUMMARIZECOLUMNS(
        ROLLUPADDISSUBTOTAL(DimProductCategory[EnglishProductCategoryName], "IsGrandTotal"),
        "TotalSales", [TotalSales],
        "%Sales", [%Sales]
    )
ORDER BY
    [IsGrandTotal] DESC, [TotalSales] DESC
```

반환

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	[ISGRANDTOTAL]	[TOTALSALES]	[%SALES]
Row1	참	29358677.2207	1
Bikes	거짓	28318144.6507	0.964557920570538
Accessories	False	700759.96	0.023868921434441
Clothing	False	339772.61	0.0115731579950215

예 2

DAX 쿼리

<pre> DEFINE MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount]) MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales], CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategoryName]))) EVALUATE SUMMARIZECOLUMNS(DimProductCategory[EnglishProductCategoryName], DimProductSubcategory[EnglishProductSubcategoryName], "TotalSales", [TotalSales], "%Sales", [%Sales]) ORDER BY DimProductCategory[EnglishProductCategoryName] ASC, DimProductSubcategory[EnglishProductSubcategoryName] ASC </pre>
--

반환

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY[ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Accessories	자전거 랙	39360	0.05616759
Accessories	자전거 스탠드	39591	0.05649723
액세서리	물병 및 거치대	56798.19	0.08105228
액세서리	클리너	7218.6	0.0103011
액세서리	펜더	46619.58	0.06652717
액세서리	헬멧	225335.6	0.3215589
액세서리	하이드레이션 팩	40307.67	0.05751994
액세서리	타이어 및 튜브	245529.32	0.35037578
Bikes	산악용 자전거	9952759.564	0.35146228

DIMPRODUCTCATEGORY [ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY [ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Bikes	도로용 자전거	14520584.04	0.51276608
Bikes	여행용 자전거	3844801.05	0.13577164
의류	캡	19688.1	0.05794493
의류	장갑	35020.7	0.10307099
의류	저지	172950.68	0.5090189
의류	반바지	71319.81	0.20990453
의류	양말	5106.32	0.01502864
의류	조끼	35687	0.10503201

SELECTEDVALUE

2021-03-16 • 2 minutes to read

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 해당 값을 반환합니다. 그 외의 경우 alternateResult를 반환합니다.

구문

```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.
alternateResult	(선택 사항) columnName의 컨텍스트가 0개 또는 둘 이상의 고유 값으로 필터링된 경우 반환되는 값입니다. 제공되지 않으면 기본값은 BLANK()입니다.

반환 값

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 해당 값입니다. 그렇지 않으면 alternateResult입니다.

설명

- SELECTEDVALUE(<columnName>, <alternateResult>)에 해당하는 식은 `IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>)`입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
DEFINE
    MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No Single Selection")
EVALUATE
    SUMMARIZECOLUMNS
        (ROLLUPADDISSUBTOTAL(DimProduct[Color], "Is Total"),
            "Selected Color", [Selected Color])ORDER BY [Is Total] ASC,
            [Color] ASC
```

그러면 다음 값이 반환됩니다.

DIMPRODUCT[COLOR]	[IS TOTAL]	[SELECTED COLOR]
검정	FALSE	검정
파랑	FALSE	파랑
회색	FALSE	회색
다중	FALSE	다중
해당 없음	FALSE	해당 없음
빨간색	FALSE	빨간색
은	FALSE	은
은색/검정	FALSE	은색/검정
흰색	FALSE	흰색
노란색	FALSE	노란색
	TRUE	단일 선택 항목 없음

재무 함수

2021-03-16 • 11 minutes to read

DAX의 재무 함수는 순 현재 가치 및 수익률과 같은 재무 계산을 수행하는 수식에 사용됩니다. 이러한 함수는 Microsoft Excel에서 사용되는 재무 함수와 유사합니다.

이 범주의 함수

함수	설명
ACCRINT	정기 이자를 지급하는 유가 증권에 경과 이자를 반환합니다.
ACCRINTM	만기 시에 이자를 지급하는 유가 증권에 경과 이자를 반환합니다.
AMORDEGRC	각 회계 기간의 감가상각액을 반환합니다. 자산 수명에 따라 계산에 감가상각 계수가 적용된다는 점을 제외하면 AMORLINC 와 유사합니다.
AMORLINC	각 회계 기간의 감가상각액을 반환합니다.
COUPDAYBS	이표 기간의 시작부터 결산일까지의 일수를 반환합니다.
COUPDAYS	결산일을 포함하여 이표 기간의 일수를 반환합니다.
COUPDAYSNC	결산일부터 다음 이표일까지의 일수를 반환합니다.
COUPNCD	결산일 이후 다음 이표일을 반환합니다.
COUPNUM	결산일과 만기일 사이에 지급할 이표 수를 가장 가까운 전체 이표로 반올림하여 반환합니다.
COUPPCD	결산일 전에 이전 이표일을 반환합니다.
CUMIPMT	시작 기간과 종료 기간 사이의 대출에 대해 납입하는 누적 이자를 반환합니다.
CUMPRINC	시작 기간과 종료 기간 사이의 대출에 대해 납입하는 누적 원금을 반환합니다.
DB	고정 체감 잔액법을 사용하여 지정한 기간의 자산의 감가상각액을 반환합니다.
DDB	이중 체감법 또는 사용자가 지정하는 다른 방법을 사용하여 지정한 기간의 자산에 대한 감가상각액을 반환합니다.
DISC	유가 증권에 할인율을 반환합니다.
DOLLARDE	정수 부분과 분수 부분으로 표시된 달러 가격(예: 1.02)을 10진수로 표시된 달러 가격으로 변환합니다.

함수	설명
DOLLARFR	정수 부분과 분수 부분으로 표시된 달러 가격(예: 1.02)을 10진수로 표시된 달러 가격으로 변환합니다.
DURATION	추정 액면가 \$100에 대한 Macauley 기간을 반환합니다.
EFFECT	연간 명목 이율 및 연간 복리 계산 기간 수를 고려하여 연간 실효 이율을 반환합니다.
FV	일정한 이율을 기준으로 미래 투자 가치를 계산합니다.
INTRATE	완전 투자 유가 증권의 이율을 반환합니다.
IPMT	정기적인 원리금 균등 상환 금액 및 일정한 이율을 기준으로 투자에 대해 지정된 기간의 이자 지급액을 반환합니다.
ISPMT	원금 균등 상환 금액을 사용하여 지정된 대출(또는 투자) 기간에 상환(또는 수령)되는 이자를 계산합니다.
MDURATION	추정 액면가가 \$100인 유가 증권의 수정된 Macauley 기간을 반환합니다.
NOMINAL	실효 이율 및 연간 복리 계산 기간 수를 고려하여 연간 명목 이율을 반환합니다.
NPER	정기적인 원리금 균등 상환 금액 및 일정한 이율을 기준으로 투자의 기간 수를 반환합니다.
ODDFPRICE	첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가 \$100당 가격을 반환합니다.
ODDFYIELD	첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권의 수익률을 반환합니다.
ODDLPRICE	마지막 이표 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가 \$100당 가격을 반환합니다.
ODDLYIELD	마지막 기간이 홀수(단기 또는 장기)인 유가 증권의 수익률을 반환합니다.
PDURATION	투자가 지정된 가치에 도달하는 데 필요한 기간 수를 반환합니다.
PMT	원리금 균등 상환 금액 및 일정한 이율을 기준으로 대출 상환 금액을 계산합니다.
PPMT	정기적인 원리금 균등 상환 금액 및 일정한 이율을 기준으로 투자에 대한 특정 기간의 원금 상환액을 반환합니다.
PRICE	정기 이자를 지급하는 유가 증권의 액면가 \$100당 가격을 반환합니다.
PRICEDISC	할인된 유가 증권의 액면가 \$100당 가격을 반환합니다.

함수	설명
PRICEMAT	만기 시 이자를 지급하는 유가 증권의 액면가 \$100당 가격을 반환합니다.
PV	일정한 이율을 기준으로 대출 또는 투자의 현재 가치를 계산합니다.
RATE	연금 기간당 이율을 반환합니다.
RECEIVED	완전 투자 유가 증권에 대해 만기 시 수령하는 금액을 반환합니다.
RRI	투자 증가액의 등가 이율을 반환합니다.
SLN	특정 기간 자산의 정액 감가상각액을 반환합니다.
SYD	지정 한 기간에 연수 합계법에 따른 자산 감가상각액을 반환합니다.
TBILLEQ	국채의 채권 상당 수익률을 반환합니다.
TBILLPRICE	국채의 액면가 \$100당 가격을 반환합니다.
TBILLYIELD	국채의 수익률을 반환합니다.
VDB	이중 체감법 또는 사용자가 지정하는 다른 방법을 사용하여 지정 한 기간(부분 기간 포함) 자산의 감가상각액을 반환합니다.
XIRR	반드시 정기적이 아니어도 되는 현금 흐름 일정의 내부 수익률을 반환합니다.
XNPV	반드시 정기적이 아니어도 되는 현금 흐름의 일정에 대한 현재 가치를 반환합니다.
YIELD	정기 이자를 지급하는 유가 증권의 수익률을 반환합니다.
YIELDDISC	할인된 유가 증권의 연간 수익률을 반환합니다.
YIELDMAT	만기 시에 이자를 지급하는 유가 증권의 연간 수익률을 반환합니다.

ACCRINT

2021-03-16 • 6 minutes to read

정기 이자를 지급하는 유가 증권의 경과 이자를 반환합니다.

구문

```
ACCRINT(<issue>, <first_interest>, <settlement>, <rate>, <par>, <frequency>[, <basis>[, <calc_method>]])
```

매개 변수

용어	정의
issue	유가 증권의 발행일입니다.
first_interest	유가 증권의 첫 번째 이자 지급일입니다.
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
요율	유가 증권의 연간 표면 금리입니다.
par	유가 증권의 액면가입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.
calc_method	(선택 사항) 결산일이 first_interest 날짜 이후인 경우 총 경과 이자를 계산하는 방법을 지정하는 논리 값입니다. calc_method가 생략되면 TRUE로 간주됩니다. - calc_method가 TRUE로 평가되거나 생략되면 ACCRINT는 issue부터 settlement까지의 총 경과 이자를 반환합니다. - calc_method가 FALSE로 평가되면 ACCRINT는 first_interest부터 settlement까지의 경과 이자를 반환합니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360

기준	일 수 기준
3	실제/365
4	유럽 30/360

반환 값

경과 이자입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- ACCRINT는 다음과 같이 계산됩니다.

$$\text{ACCRINT} = \text{par} \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}$$

여기서

- A_i = 홀수 기간 내 i^{th} 준이표 기간의 경과 일 수입니다.
 - NC = 홀수 기간에 속하는 준이표 기간의 수입니다. 이 숫자에 분수가 포함되어 있으면 다음 정수로 올립니다.
 - NL_i = 홀수 기간 내 준이표 기간의 정상 길이(일 수)입니다.
- issue, first_interest, settlement는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - issue, first_interest 또는 settlement가 유효한 날짜가 아닙니다.
 - issue \geq settlement.
 - rate \leq 0.
 - par \leq 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
2007년 3월 1일	발행일
2008년 8월 31일	첫 번째 이자 지급일
2008년 5월 1일	결산일
10%	표면 금리

DATA	설명
1000	액면가
2	frequency는 반기입니다(위 참조)
0	30/360 기준(위 참조)

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0)
}
```

위에 지정된 조건을 사용하여 유가 증권의 발행일부터 결산일까지의 경과 이자를 반환합니다.

[VALUE]
116.944444444444

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0, FALSE)
}
```

위에 지정된 조건을 사용하여 유가 증권의 first_interest부터 settlement까지의 경과 이자를 반환합니다.

[VALUE]
66.9444444444445

ACCRINTM

2021-03-16 • 4 minutes to read

만기 시 이자를 지급하는 유가 증권의 경과 이자를 반환합니다.

구문

```
ACCRINTM(<issue>, <maturity>, <rate>, <par>[, <basis>])
```

매개 변수

용어	정의
이슈	유가 증권의 발행일입니다.
maturity	유가 증권의 만기일입니다.
요율	유가 증권의 연간 표면 금리입니다.
par	유가 증권의 액면가입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

경과 이자입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- ACCRINTM은 다음과 같이 계산됩니다.

$$\text{\text{\text{ACCRINTM}}} = \text{\text{\text{par}}} \times \text{\text{\text{rate}}} \times \frac{\text{\text{\text{A}}}}{\text{\text{\text{D}}}}$$

여기서

- $\text{\text{\text{A}}}$ = 월 단위로 계산한 경과 일 수입니다. 만기 시 이자 항목의 경우 발행일부터 만기일까지의 일 수를 사용합니다.
- $\text{\text{\text{D}}}$ = 연 기준.
- issue와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - issue 또는 maturity가 유효한 날짜가 아닙니다.
 - $\text{issue} \geq \text{maturity}$.
 - $\text{rate} \leq 0$.
 - $\text{par} \leq 0$.
 - $\text{basis} < 0$ 또는 $\text{basis} > 4$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2008년 4월 1일	발행일
2008년 6월 15일	만기일
10%	표면 금리
1000	액면가
3	실제/365 기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    ACCRINTM(DATE(2008,4,1), DATE(2008,6,15), 0.1, 1000, 3)
}
```

위에 지정된 조건을 사용하여 유가 증권에의 경과 이자를 반환합니다.

[VALUE]
20.5479452054795

AMORDEGRC

2021-03-16 • 5 minutes to read

각 회계 기간의 감가상각을 반환합니다. 이 함수는 프랑스 회계 시스템용으로 제공됩니다. 회계 기간 중간에 자산을 구매하는 경우 비례 감가상각이 고려됩니다. 자산 수명에 따라 계산에 감가상각 계수가 적용된다는 점을 제외하면 AMORLINC와 유사합니다.

구문

```
AMORDEGRC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

매개 변수

용어	정의
cost	자산의 비용입니다.
date_purchased	자산의 구매 날짜입니다.
first_period	첫 번째 기간의 종료 날짜입니다.
salvage	자산 수명 종료 시의 잔존 가치입니다.
period	기간입니다.
요일	감가상각률입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	날짜 시스템
0 또는 생략됨	360일(NASD 방식)
1	Actual
3	연간 365일
4	연간 360일(유럽 방식)

반환 값

각 회계 기간의 감가상각입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 이 함수는 자산의 마지막 기간 또는 감가상각의 누적 가치가 자산 비용에서 잔존 가치를 뺀 값보다 클 때 까지 감가상각을 반환합니다.
- 감가상각 계수는 다음과 같습니다.

자산 수명(1/RATE)	감가상각 계수
3년에서 4년 사이	1.5
5년에서 6년 사이	2
7년 이상	2.5

- 감가상각률은 마지막 기간 이전 기간 동안 50%로 증가하고, 마지막 기간 동안 100%까지 증가합니다.
- period 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $cost < 0$.
 - first_period 또는 date_purchased가 유효한 날짜가 아닙니다.
 - $date_purchased > first_period$.
 - $salvage < 0$ 또는 $salvage > cost$.
 - $period < 0$.
 - $rate \leq 0$.
 - 자산의 수명이 0~1, 1~2, 2~3 또는 4~5입니다.
 - basis가 0, 1, 3 또는 4 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2400	Cost
2008년 8월 19일	구매 날짜
2008년 12월 31일	첫 번째 기간 종료
300	잔존 가치
1	기간
15%	감가상각률
1	실제 기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

EVALUATE

```
{  
  AMORDEGRC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)  
}
```

위에 지정된 조건에 따라 첫 번째 기간의 감가상각을 반환합니다.

[VALUE]

776

AMORLINC

2021-03-16 • 4 minutes to read

각 회계 기간의 감가상각을 반환합니다. 이 함수는 프랑스 회계 시스템용으로 제공됩니다. 회계 기간 중간에 자산을 구매하는 경우 비례 배분 감가상각이 고려됩니다.

구문

```
AMORLINC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

매개 변수

용어	정의
cost	자산의 비용입니다.
date_purchased	자산의 구매 날짜입니다.
first_period	첫 번째 기간의 종료 날짜입니다.
salvage	자산 수명 종료 시의 잔존 가치입니다.
period	기간입니다.
요일	감가상각률입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	날짜 시스템
0 또는 생략됨	360일(NASD 방식)
1	Actual
3	연간 365일
4	연간 360일(유럽 방식)

반환 값

각 회계 기간의 감가상각입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일

이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- period 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $\text{cost} < 0$.
 - first_period 또는 date_purchased가 유효한 날짜가 아닙니다.
 - $\text{date_purchased} > \text{first_period}$.
 - $\text{salvage} < 0$ 또는 $\text{salvage} > \text{cost}$.
 - $\text{period} < 0$.
 - $\text{rate} \leq 0$.
 - basis가 0, 1, 3 또는 4 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2400	Cost
2008년 8월 19일	구매 날짜
2008년 12월 31일	첫 번째 기간 종료
300	잔존 가치
1	기간
15%	감가상각률
1	실제 기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

위에 지정된 조건에 따라 첫 번째 기간의 감가상각을 반환합니다.

[VALUE]
360

COUPDAYBS

2021-03-16 • 5 minutes to read

이표 기간의 시작부터 결산일까지의 일 수를 반환합니다.

구문

```
COUPDAYBS(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

이표 기간의 시작부터 결산일까지의 일 수입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2011년 1월 25일	결산일
2015년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPDAYBS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

위의 조건이 적용되는 채권의 경우 이표 기간의 시작부터 결산일까지의 일 수를 반환합니다.

[VALUE]
71

COUPDAYS

2021-03-16 • 5 minutes to read

결산일을 포함하는 이자 지급 기간의 일수를 반환합니다.

구문

```
COUPDAYS(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

결산일을 포함하는 이자 지급 기간의 일수입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2011년 1월 25일	결산일
2015년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPDAYS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

위에 지정된 조건의 채권의 경우 결산일을 포함하는 이표 기간의 일 수를 반환합니다.

[VALUE]
181

COUPDAYSNC

2021-03-16 • 5 minutes to read

결산일부터 다음 이표일까지의 일 수를 반환합니다.

구문

```
COUPDAYSNC(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

결산일부터 다음 이표일까지의 일 수입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2011년 1월 25일	결산일
2015년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPDAYSNC(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

위에 지정된 조건의 채권에 대해 결산일부터 다음 이표일까지의 일 수를 반환합니다.

[VALUE]
110

COUPNCD

2021-03-16 • 5 minutes to read

결산일 이후의 다음 이표일을 반환합니다.

구문

```
COUPNCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

결산일 이후의 다음 이표일입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2011년 1월 25일	결산일
2015년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPNCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

위에 지정된 조건의 채권에 대해 결산일 이후의 다음 이표일을 반환합니다.

[VALUE]
5/15/2011 12:00:00 AM

COUPNUM

2021-03-16 • 5 minutes to read

결산일과 만기일 사이에 지급할 이표 수를 가장 가까운 전체 이표로 반올림하여 반환합니다.

구문

```
COUPNUM(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

결산일과 만기일 사이에 지급할 이표 수입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2007년 1월 25일	결산일
2008년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPNUM(DATE(2007,1,25), DATE(2008,11,15), 2, 1)
}
```

위에 지정된 조건을 사용하여 채권의 이표 지급 횟수를 반환합니다.

[VALUE]
4

COUPPCD

2021-03-16 • 5 minutes to read

결산일 이전의 이표일을 반환합니다.

구문

```
COUPPCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

결산일 이전의 이표일입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2011년 1월 25일	결산일
2015년 11월 15일	만기일
2	반기 이표(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    COUPPCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

위에 지정된 조건을 사용하여 채권의 결산일 이전 이표일을 반환합니다.

[VALUE]
2010년 11월 15일 오전 12:00:00

CUMIPMT

2021-03-16 • 4 minutes to read

start_period와 end_period 사이에 상환하는 대출 누적 이자를 반환합니다.

구문

```
CUMIPMT(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

매개 변수

용어	정의
요율	이자율입니다.
nper	총 지급 기간 수입니다.
pv	현재 가치입니다.
start_period	계산의 첫 번째 기간입니다. 1에서 end_period 사이여야 합니다(두 값 모두 포함).
end_period	계산의 마지막 기간입니다. start_period와 nper 사이여야 합니다(두 값 모두 포함).
형식	지급 시기입니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

유형 매개 변수는 다음 값을 허용합니다.

유형	타이밍
0(영)	기간 종료 시 지급
1	기간 시작 시 지급

반환 값

지정된 기간에 상환되는 누적 이자입니다.

설명

- rate와 nper을 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 10%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.1/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.1을, nper에 4를 사용합니다.
- start_period, end_period, type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.

- $rate \leq 0$.
- $nper < 1$.
- $pv \leq 0$.
- $start_period < 1$ 또는 $start_period > end_period$.
- $end_period < start_period$ 또는 $end_period > nper$.
- type이 0 또는 1 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
9%	연간 이자율
30	대출 연수
125000	현재 가치

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

매월 초에 상환한다고 가정할 때 13기부터 24기까지 두 번째 상환 연도에 상환하는 총 이자를 반환합니다.

[VALUE]
-11052.3395838718

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

월말에 상환한다고 가정할 때 첫 번째 달에 단일 상환으로 상환되는 이자를 반환합니다.

[VALUE]
-937.5

CUMPRINC

2021-03-16 • 4 minutes to read

start_period와 end_period 사이에 지급되는 대출 누적 원금을 반환합니다.

구문

```
CUMPRINC(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

매개 변수

용어	정의
요율	이자율입니다.
nper	총 지급 기간 수입니다.
pv	현재 가치입니다.
start_period	계산의 첫 번째 기간입니다. 1에서 end_period 사이여야 합니다(두 값 모두 포함).
end_period	계산의 마지막 기간입니다. start_period와 nper 사이여야 합니다(두 값 모두 포함).
형식	지급 시기입니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

유형 매개 변수는 다음 값을 허용합니다.

유형	타이밍
0(영)	기간 종료 시 지급
1	기간 시작 시 지급

반환 값

지정된 기간에 지급되는 누적 원금입니다.

설명

- rate와 nper을 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 10%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.1/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.1을, nper에 4를 사용합니다.
- start_period, end_period, type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.

- $rate \leq 0$.
- $nper < 1$.
- $pv \leq 0$.
- $start_period < 1$ 또는 $start_period > end_period$.
- $end_period < start_period$ 또는 $end_period > nper$.
- type이 0 또는 1 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
9%	연간 이자율
30	기간(년)
125000	현재 가치

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

매월 초에 상환한다고 가정할 때 13기부터 24기까지 두 번째 상환 연도에 상환하는 총 원금을 반환합니다.

[VALUE]
-927.153472378062

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

월말에 상환한다고 가정할 때 첫 번째 달에 단일 상환으로 상환되는 원금을 반환합니다.

[VALUE]
-68.2782711809784

DB

2021-03-16 • 4 minutes to read

고정 채감 잔액법을 사용하여 지정한 기간 동안 자산의 감가상각을 반환합니다.

구문

```
DB(<cost>, <salvage>, <life>, <period>[, <month>])
```

매개 변수

용어	정의
cost	초기 자산 비용입니다.
salvage	감가상각 종료 시의 값입니다(자산의 잔존 가치라고도 함). 이 값은 0일 수 있습니다.
응용해 보세요.	자산이 감가상각된 기간 수입니다(자산의 내용연수라고도 함).
period	감가상각을 계산할 기간입니다. period는 life와 동일한 단위를 사용해야 합니다. 1과 life 사이여야 합니다(두 값 모두 포함).
month	(선택 사항) 첫 해의 개월 수입니다. month가 생략된 경우 12로 간주됩니다.

반환 값

지정된 기간 동안의 감가상각입니다.

설명

- 고정 채감 잔액법은 고정된 비율로 감가상각을 계산합니다. DB는 다음 수식을 사용하여 특정 기간의 감가상각을 계산합니다.

$$(\text{cost} - \text{total depreciation from prior periods}) \times \text{rate}$$

여기서

- $\text{rate} = 1 - (\frac{\text{salvage}}{\text{cost}})^{\frac{1}{\text{life}}}$, rounded to three decimal places
- 첫 번째 및 마지막 기간의 감가상각은 특별한 경우입니다.
 - 첫 번째 기간의 경우 DB는 다음 수식을 사용합니다.
$$\frac{\text{cost}}{12} \times \text{rate} \times \text{month}$$
 - 마지막 기간의 경우 DB는 다음 수식을 사용합니다.
$$\frac{\text{cost} - \text{total depreciation from prior periods}}{12 - \text{month}} \times \text{rate} \times \text{month}$$

\text{month}}\}{12}\$\$

- period 및 month는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - cost < 0.
 - salvage < 0.
 - life < 1.
 - period < 1 또는 period > life.
 - month < 1 또는 month > 12.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

예 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DB(1000000, 0, 6, 1, 2)
}
```

6년 후 가치가 \$0라고 가정하여 첫 번째 해의 마지막 두 달의 자산 감가상각을 반환합니다.

[VALUE]

166666.666666667

예제 2

다음은 자산 수명의 서로 다른 연도의 모든 자산의 총 감가상각을 계산합니다. 여기서 첫 번째 해에는 7개월의 감가상각만 포함되고 마지막 해에는 5개월의 감가상각만 포함됩니다.

```
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)
    )
)
```

DDB

2021-03-16 • 4 minutes to read

이중 체감 잔액법 또는 사용자가 지정하는 다른 방법을 사용하여 지정한 기간 동안 자산의 감가상각을 반환합니다.

구문

```
DDB(<cost>, <salvage>, <life>, <period>[, <factor>])
```

매개 변수

용어	정의
cost	초기 자산 비용입니다.
salvage	감가상각 종료 시의 값입니다(자산의 잔존 가치라고도 함). 이 값은 0일 수 있습니다.
응용해 보세요.	자산이 감가상각된 기간 수입니다(자산의 내용연수라고도 함).
period	감가상각을 계산할 기간입니다. period는 life와 동일한 단위를 사용해야 합니다. 1과 life 사이여야 합니다(두 값 모두 포함).
요소	(선택 사항) 잔액이 체감하는 비율입니다. factor가 생략되면 2로 간주됩니다(이중 체감 잔액법).

반환 값

지정된 기간 동안의 감가상각입니다.

설명

- 이중 체감 잔액법은 가속화된 비율로 감가상각을 계산합니다. 첫 번째 기간의 감가상각이 가장 높고 이후 기간에는 감소합니다. DDB는 다음 수식을 사용하여 특정 기간의 감가상각을 계산합니다.

$$\text{Min}((\text{cost} - \text{total depreciation from prior periods}) \times \frac{\text{factor}}{\text{life}}, (\text{cost} - \text{salvage} - \text{total depreciation from prior periods}))$$

- 이중 체감 잔액법을 사용하지 않으려면 factor를 변경하세요.
- 감가상각이 정률법 계산보다 큰 경우 정액법에 의한 감가상각으로 전환하려면 VDB 함수를 사용합니다.
- period는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - cost < 0.
 - salvage < 0.
 - life < 1.

- $\text{period} < 1$ 또는 $\text{period} > \text{life}$.
- $\text{factor} \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

예 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DDB(1000000, 0, 10, 5, 1.5)
}
```

10년 후 가치가 ₩0이라고 가정하여 5%의 자산 감가상각을 반환합니다. 이 계산에서 사용하는 인수는 1.5입니다.

[VALUE]

78300.9375

예제 2

다음은 자산 수명의 서로 다른 연도의 모든 자산의 총 감가상각을 계산합니다. 이 계산은 기본 인수인 2(이중 체감 잔액법)를 사용합니다.

```
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])
        )
    )
```

DISC

2021-03-16 • 5 minutes to read

유가 증권의 할인율을 반환합니다.

구문

```
DISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
pr	유가 증권의 액면가 \ \$100당 가격입니다.
redemption	유가 증권의 액면가 \ \$100당 상환액입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

할인율.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2018년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2018년 1월 1일이 되고, 결산일은 2018년 7월 1일이 되며, 만기일은 발행일인 2018년 1월 1일에서 30년 후인 2048년 1월 1일이 됩니다.

- DISC는 다음과 같이 계산됩니다.

$$\text{DISC} = \frac{\text{redemption} - \text{par}}{\text{redemption}} \times \frac{\text{B}}{\text{DSM}}$$

여기서

- B = 연 기준에 따른 연간 일 수입입니다.
- DSM = settlement와 maturity 사이의 일 수입입니다.
- settlement와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - $\text{pr} \leq 0$.
 - redemption ≤ 0 .
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2018년 7월 1일	결산일
2048년 1월 1일	만기일
97.975	가격
100	상환액
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DISC(DATE(2018,7,1), DATE(2048,1,1), 97.975, 100, 1)
}
```

위에 지정된 조건이 적용되는 채권의 채권 할인율을 반환합니다.

[VALUE]

0.000686384169121348

DOLLARDE

2021-03-16 • 2 minutes to read

정수 부분과 분수 부분으로 표시된 달러 가격(예: 1.02)을 10진수로 표시된 달러 가격으로 변환합니다. 분수 달러 숫자는 이따금 유가 증권 가격에 사용되기도 합니다.

구문

```
DOLLARDE(<fractional_dollar>, <fraction>)
```

매개 변수

용어	정의
fractional_dollar	소수점 기호로 구분된 정수 부분과 분수 부분으로 표시되는 숫자입니다.
분수	분수의 분모에 사용할 정수입니다.

반환 값

*fractional_dollar*의 10진수 값입니다.

설명

- 값의 분수 부분은 지정한 정수로 나눕니다. 예를 들어 가격을 1달러의 1/16 정밀도로 표시하려면 분수 부분을 16으로 나눕니다. 이 경우 1.02는 $\$1.125$ ($\$1 + 2/16 = \1.125)를 나타냅니다.
- fraction은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - fraction < 1.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DOLLARDE(1.02, 16)
}
```

1과 2/16로 읽는 원래 분수 가격 1.02의 10진수 가격인 1.125를 반환합니다. 분수 값이 16이므로 가격은 1달러의 1/16의 정밀도를 갖습니다.

DOLLARFR

2021-03-16 • 2 minutes to read

10진수로 표시된 달러 가격을 정수 부분과 분수 부분으로 표시된 달러 가격(예: 1.02)으로 변환합니다. 분수 달러 숫자는 이따금 유가 증권 가격에 사용되기도 합니다.

구문

```
DOLLARFR(<decimal_dollar>, <fraction>)
```

매개 변수

용어	정의
decimal_dollar	10진수입니다.
분수	분수의 분모에 사용할 정수입니다.

반환 값

정수 부분과 분수 부분으로 표현되는 *decimal_dollar*의 분수 값입니다.

설명

- fraction은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - fraction < 1.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DOLLARFR(1.125, 16)
}
```

1과 2/16으로 읽는 1.02를 반환합니다. 이 값은 원래 10진수 가격 1.125에 해당하는 분수 가격입니다. 분수 값이 16이므로 가격은 1달러의 1/16의 정밀도를 갖습니다.

DURATION

2021-03-16 • 5 minutes to read

추정 액면가 \ \$100에 대한 Macauley 기간을 반환합니다. 기간은 현금 흐름의 현재 가치 가중 평균으로 정의되며, 수익률 변화에 대한 채권 가격 반응의 측정값으로 사용됩니다.

구문

```
DURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
coupon	유가 증권의 연간 표면 금리입니다.
yld	유가 증권의 연간 수익률입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

Macauley 기간입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - coupon < 0 .
 - yld < 0
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4 .
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2018년 7월 1일	결산일
2048년 1월 1일	만기일
8.0%	표면 금리
9.0%	수익률
2	frequency는 반기입니다(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    DURATION(DATE(2018,7,1), DATE(2048,1,1), 0.08, 0.09, 2, 1)
}
```

위에 지정된 조건이 적용되는 채권의 Macauley 기간을 반환합니다.

[VALUE]
10.9191452815919

EFFECT

2021-03-16 • 2 minutes to read

연간 명목 이자율 및 연간 복리 계산 기간 수를 고려하여 연간 실질 이자율을 반환합니다.

구문

```
EFFECT(<nominal_rate>, <npery>)
```

매개 변수

용어	정의
nominal_rate	명목 이자율입니다.
npery	연간 복리 계산 기간의 수입니다.

반환 값

연간 실질 이자율입니다.

설명

- EFFECT는 다음과 같이 계산됩니다.

$$\text{EFFECT} = \text{bigg}(1 + \frac{\text{nominal_rate}}{\text{npery}} \text{bigg})^{\text{npery}} - 1$$

- npery는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nominal_rate ≤ 0.
 - npery < 1.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
5.25%	명목 이자율
4	연간 복리 계산 기간의 수

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
  EFFECT(0.0525, 4)
}
```

위에 지정된 조건을 사용하여 실질 이자율을 반환합니다.

[VALUE]

0.0535426673707584

FV

2021-03-16 • 5 minutes to read

일정한 이자율을 기준으로 미래 투자 가치를 계산합니다. 정기 상환, 균등 상환 또는 단일 총액 상환에 FV를 사용할 수 있습니다.

구문

```
FV(<rate>, <nper>, <pmt>[, <pv>[, <type>]])
```

매개 변수

용어	정의
요율	기간당 이자율입니다.
nper	연금의 총 지급 기간 수입니다.
pmt	각 기간의 지급이며 연금의 수명 동안 변경될 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료 또는 세금은 포함되지 않습니다.
pv	(선택 사항) 현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총액입니다. pv가 생략된 경우 0으로 간주됩니다.
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같게 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

참고: FV의 인수에 대한 자세한 설명 및 연금 함수에 대한 자세한 내용은 PV 함수를 참조하세요.

반환 값

투자의 미래 가치입니다.

설명

- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를, nper에 4를 사용합니다.
- 모든 인수에서 저축 예금과 같이 지급하는 현금은 음수로 표시됩니다. 배당 수표와 같이 수신하는 현금

은 양수로 표시됩니다.

- type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $nper < 1$
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
6%	연간 이자율
10	지급 횟수
-200	지급 금액
-500	현재 가치
1	지급 기한은 기간 초입니다(0은 지급 기한이 기간 말임을 나타냅니다).

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    FV(0.06/12, 10, -200, -500, 1)
}
```

위에 지정된 조건을 사용하여 투자의 미래 가치를 반환합니다.

[VALUE]
2581.40337406012

INTRATE

2021-03-16 • 5 minutes to read

완전 투자 유가 증권의 이자율을 반환합니다.

구문

```
INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
investment	유가 증권의 투자액입니다.
redemption	만기 시 상환액입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

이자율입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.

- INTRATE는 다음과 같이 계산됩니다.

$$\text{INTRATE} = \frac{\text{redemption} - \text{investment}}{\text{investment}} \times \frac{\text{B}}{\text{DIM}}$$

여기서

- B = 연 기준에 따른 연간 일 수입입니다.
- DSM = settlement부터 maturity까지의 일 수입입니다.
- settlement와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - $\text{settlement} \geq \text{maturity}$.
 - $\text{investment} \leq 0$.
 - $\text{redemption} \leq 0$.
 - $\text{basis} < 0$ 또는 $\text{basis} > 4$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2008년 2월 15일	결산일
2008년 5월 15일	만기일
\\$1,000,000	투자
\\$1,014,420	상환액
2	실제/360 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    INTRATE(DATE(2008,2,15), DATE(2008,5,15), 1000000, 1014420, 2)
}
```

위에 지정된 조건을 사용하여 채권의 할인율을 반환합니다.

[VALUE]
0.05768

IPMT

2021-03-16 • 5 minutes to read

정기적인 원리금 균등 상환 및 일정한 이자율을 기준으로 투자에 대한 특정 기간의 이자 상환액을 반환합니다.

구문

```
IPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

매개 변수

용어	정의
요율	기간당 이자율입니다.
per	이자를 찾을 기간입니다. 1과 nper 사이여야 합니다(두 값 모두 포함).
nper	연금의 총 지급 기간 수입니다.
pv	현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총액입니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv가 생략된 경우 BLANK로 간주됩니다.
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type 이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같게 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

반환 값

지정된 기간의 이자 상환액입니다.

설명

- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를, nper에 4를 사용합니다.
- 모든 인수에서 저축 예금과 같이 지급하는 현금은 음수로 표시됩니다. 배당 수표와 같이 수신하는 현금 은 양수로 표시됩니다.

- type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $per < 1$ 또는 $per > nper$
 - $nper < 1$
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
10.00%	연간 이자율
3	대출 기간(년)
\\$8,000	대출의 현재 가치

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    IPMT(0.1/12, 1, 3*12, 8000)
}
```

위에서 지정한 조건의 대출의 첫 달 월 이자 상환액을 반환합니다.

[VALUE]
-66.6666666666667

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    IPMT(0.1, 3, 3, 8000)
}
```

상환이 매년 이루어지는 경우 위에 지정된 조건의 대출 마지막 해에 상환할 연간 이자를 반환합니다.

[VALUE]
-292.447129909366

ISPMT

2021-03-16 • 5 minutes to read

원금 균등 상환을 사용하여 지정된 대출(또는 투자) 기간 동안 지급(또는 수신)된 이자를 계산합니다.

구문

ISPMT(<rate>, <per>, <nper>, <pv>)

매개 변수

용어	정의
요율	투자의 이자율입니다.
per	이자를 찾을 기간입니다. 0과 nper-1 사이여야 합니다(두 값 모두 포함).
nper	투자의 총 지급 기간 수입니다.
pv	투자의 현재 가치입니다. 대출의 경우 pv는 대출 금액입니다.

반환 값

지정된 기간 동안 지급(또는 수신)된 이자입니다.

설명

- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를, nper에 4를 사용합니다.
- 모든 인수에서 저축 예금이나 인출과 같이 지급하는 현금은 음수로 표시됩니다. 배당 수표 및 기타 예치금 같이 수신하는 현금은 양수로 표시됩니다.
- ISPMT는 각 기간이 1이 아닌 0으로 시작되는 것으로 계산합니다.
- 대부분의 대출은 정기 상환액이 균등한 상환 일정을 사용합니다. IPMT 함수는 이 대출 유형의 지정된 기간 동안의 이자 지급을 반환합니다.
- 일부 대출에서는 원금 상환액이 균등한 상환 일정을 사용합니다. ISPMT 이 대출 유형의 지정된 기간 동안의 이자 지급을 반환합니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nper = 0.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
\\$4,000	현재 가치
4	기간 수
10%	비용

ISPMT를 언제 사용해야 하는지 보여 주기 위해 아래 할부 상환 테이블은 위에 지정된 조건으로 원금 균등 상환 일정을 사용합니다. 각 기간의 이자는 이자율과 이전 기간의 미납금을 곱한 값과 같습니다. 각 기간의 지급 금액은 해당 기간의 균등 원리금과 같습니다.

기간	원금 상환	이자 상환	총 상환금	BALANCE
				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
4	1,000.00	100.00	1,100.00	0.00

다음 DAX 쿼리를 실행한다고 가정합니다.

```
DEFINE
VAR NumPaymentPeriods = 4
VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1)
EVALUATE
ADDCOLUMNS (
    PaymentPeriods,
    "Interest Payment",
    ISPMT(0.1, [Value], NumPaymentPeriods, 4000)
)
```

원금 균등 상환 일정 및 위에서 지정한 조건을 사용하여 각 기간 동안 상환하는 이자를 반환합니다. 수신한 이자가 아니라 지급한 이자임을 나타내기 위해 값은 음수입니다.

[VALUE]	[INTEREST PAYMENT]
0	-400
1	-300
2	-200
3	-100

MDURATION

2021-03-16 • 5 minutes to read

추정 액면가가 \$100인 유가 증권의 수정된 Macauley 기간을 반환합니다.

구문

```
MDURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
coupon	유가 증권의 연간 표면 금리입니다.
yld	유가 증권의 연간 수익률입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

수정된 Macauley 기간입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이고, 결산일은 2008년 7월 1일이며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일입니다.

- 수정된 기간은 다음과 같이 정의됩니다.

$$MDURATION = \frac{DURATION}{1 + (\frac{Market\ yield}{Coupon\ payments\ per\ year})}$$

- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement ≥ maturity.
 - coupon < 0.
 - yld < 0
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2008년 1월 1일	결산일
2016년 1월 1일	만기일
8%	표면 금리
9%	수익률
2	frequency는 반기입니다(위 참조)
1	실제/실제기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    MDURATION(DATE(2008,1,1), DATE(2016,1,1), 0.08, 0.09, 2, 1)
}
```

위에 지정된 조건을 사용하여 채권의 수정된 Macauley 기간을 반환합니다.

[VALUE]

5.73566981391884

NOMINAL

2021-03-16 • 2 minutes to read

실질 이자율 및 연간 복리 계산 기간 수를 고려하여 연간 명목 이자율을 반환합니다.

구문

```
NOMINAL(<effect_rate>, <npery>)
```

매개 변수

용어	정의
effect_rate	실질 이자율입니다.
npery	연간 복리 계산 기간의 수입니다.

반환 값

연간 명목 이자율입니다.

설명

- 다음 수식에 NOMINAL과 EFFECT 간의 관계가 나와 있습니다.

$$\text{EFFECT} = \left(1 + \frac{\text{nominal_rate}}{\text{npery}} \right)^{\text{npery}} - 1$$

- npery는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - effect_rate ≤ 0.
 - npery < 1.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
5.3543%	실질 이자율
4	연간 복리 계산 기간의 수

다음 DAX 쿼리를 실행한다고 가정합니다.


```
EVALUATE
{
  NOMINAL(0.053543, 4)
}
```

위에 지정된 조건을 사용하여 명목 이자율을 반환합니다.

[VALUE]
0.052500319868356

NPER

2021-03-16 • 3 minutes to read

정기적인 원리금 균등 상환 및 일정한 이율을 기준으로 투자의 기간 수를 반환합니다.

구문

```
NPER(<rate>, <pmt>, <pv>[, <fv>[, <type>]])
```

매개 변수

용어	정의
요율	기간당 이자율입니다.
pmt	각 기간의 지급이며 연금의 수명 동안 변경될 수 없습니다. 일반적으로 pmt 에는 원금과 이자가 포함되지만 다른 수수료 또는 세금은 포함되지 않습니다.
pv	현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총액입니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv 가 생략된 경우 BLANK 로 간주됩니다.
형식	(선택 사항) 숫자 0 또는 1이며 납입 시점을 나타냅니다. type 이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같게 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

반환 값

투자 기간의 수입니다.

설명

- type**은 가장 가까운 정수로 반올림됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
12%	연간 이자율
-100	각 기간의 지급
-1000	현재 가치
10000	미래 가치
1	기간 시작 시 지급합니다(위 참조).

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    NPER(0.12/12, -100, -1000, 10000, 1)
}
```

위에서 지정한 조건에 설명된 투자의 기간 수를 반환합니다.

[VALUE]
59.6738656742946

ODDFPRICE

2021-03-16 • 10 minutes to read

첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

구문

```
ODDFPRICE(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <yld>, <redemption>, <frequency>[,  
<basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
이슈	유가 증권의 발행일입니다.
first_coupon	유가 증권의 첫 번째 이표일입니다.
요율	유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률입니다.
redemption	액면가 \ \$100당 유가 증권의 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365

기준	일 수 기준
4	유럽 30/360

반환 값

액면가\\$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- ODDFPRICE는 다음과 같이 계산됩니다.

홀수 단기 첫 번째 이표:

$$\text{ODDFPRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} + \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{\text{E}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}}}{\text{Big} [100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}]}$$

여기서

- A = 이표 기간 시작부터 결산일까지의 일수입니다(경과 일수).
- DSC = 결산일부터 다음 이표일까지의 일수입니다.
- DFC = 첫 번째 홀수 이표 시작부터 첫 번째 이표일까지의 일수입니다.
- E = 이표 기간의 일수입니다.
- N = 결산일과 만기일 사이에 지급할 이표 수입니다. (이 숫자에 소수 부분이 포함되어 있으면 다음 정수로 올림됩니다.)

홀수 장기 첫 번째 이표:

$$\text{ODDFPRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N} + \text{N}_q + \frac{\text{DSC}}{\text{E}})}} + \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \text{Big} [\sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i}]}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N}_q + \frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=1}^{\text{N}} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - \text{N}_q + \frac{\text{DSC}}{\text{E}})}}}{\text{Big} [100 \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}]}$$

여기서

- A_i = 홀수 기간 내 i^{th} 또는 마지막 준이표 기간 시작부터의 일수입니다.
- DC_i = 지정 날짜(또는 발행일)로부터 첫 번째 준이표까지의 일수($i = 1$) 또는 준이표의 일수($i = 2, \dots, i = \text{NC}$)입니다.
- DSC = 결산일부터 이표일까지의 일수입니다.
- E = 이표 기간의 일수입니다.

- $\text{\textit{N}}$ = 첫 번째 실제 이표일과 상환일 간에 지급 가능한 이표의 수입입니다. (이 숫자에 소수 부분이 포함되어 있으면 다음 정수로 올림됩니다.)
- $\text{\textit{NC}}$ = 홀수 기간에 해당하는 준이표 기간의 수입입니다. (이 숫자에 소수 부분이 포함되어 있으면 다음 정수로 올림됩니다.)
- $\text{\textit{NL}}_i$ = 홀수 기간 내 전체 i^{th} 또는 마지막 준이표 기간의 정상 길이(일수)입니다.
- $\text{\textit{N}}_q$ = 결산일과 첫 이표 사이 전체 준이표 기간의 수입입니다.
- settlement, maturity, issue, first_coupon은 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, issue 또는 first_coupon이 유효한 날짜가 아닙니다.
 - maturity > first_coupon > settlement > issue가 충족되지 않습니다.
 - rate < 0.
 - yld < 0.
 - redemption ≤ 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	인수 설명
11/11/2008	결산일
3/1/2021	만기일
10/15/2008	발행일
3/1/2009	첫 번째 이표 날짜
7.85%	이표율
6.25%	수익률
\\$100.00	상환 금액
2	빈도는 반기입니다.
1	실제/실제 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
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
    )
}
```

위에 지정된 조건을 사용하여, 첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권 액면가 \\$100당 가격을 반환

합니다.

[VALUE]
113.597717474079

ODDFYIELD

2021-03-16 • 7 minutes to read

첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권 수익률을 반환합니다.

구문

```
ODDFYIELD(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
이슈	유가 증권 발행일입니다.
first_coupon	유가 증권 첫 번째 이표일입니다.
요율	유가 증권 이자율입니다.
pr	유가 증권 가격입니다.
redemption	액면가 \$100당 유가 증권 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365

기준	일 수 기준
4	유럽 30/360

반환 값

유가 증권 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- ODDFYIELD는 반복 메서드를 사용하여 계산됩니다. ODDFPRICE 함수에 사용되는 수식을 기반으로 뉴턴 법을 사용합니다. yield는 지정된 수익률의 예상 가격이 가격에 근접할 때까지 100회 반복을 통해 변경됩니다. ODDFYIELD에서 사용하는 수식은 ODDFPRICE를 참조하세요.
- settlement, maturity, issue 및 first_coupon은 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, issue 또는 first_coupon이 유효한 날짜가 아닙니다.
 - maturity > first_coupon > settlement > issue가 충족되지 않습니다.
 - rate < 0.
 - pr ≤ 0.
 - redemption ≤ 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	인수 설명
2008년 11월 11일	결산일
2021년 3월 1일	만기일
2008년 10월 15일	발행일
2009년 3월 1일	첫 번째 이표일
5.75%	표면 금리
84.50	가격

DATA	인수 설명
100	상환 금액
2	frequency는 반기입니다
0	30/360 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
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)
}
```

위에 지정된 조건을 사용하여 첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권의 수익률을 반환합니다.

[VALUE]
0.0772455415972989

ODDLPRICE

2021-03-16 • 6 minutes to read

마지막 이표 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

구문

```
ODDLPRICE(<settlement>, <maturity>, <last_interest>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
last_interest	유가 증권의 마지막 이표일입니다.
요일	유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률입니다.
redemption	액면가 \ \$100당 유가 증권의 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

액면가 \ \$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement, maturity, last_interest는 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 last_interest가 유효한 날짜가 아닙니다.
 - maturity > settlement > last_interest 조건이 충족되지 않습니다.
 - rate < 0.
 - yld < 0.
 - redemption ≤ 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

DATA	인수 설명
2008년 2월 7일	결산일
2008년 6월 15일	만기일
2007년 10월 15일	마지막 이자 지급일
3.75%	표면 금리
4.05%	수익률
\ \$100	상환 금액
2	frequency는 반기입니다
0	30/360 기준

EVALUATE

```
{  
  ODDLPRICE( DATE(2008,2,7), DATE(2008,6,15), DATE(2007,10,15), 0.0375, 0.0405, 100, 2, 0)  
}
```

위에 지정된 조건을 사용하여 마지막 이표 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가\\$100당 가격을 반환합니다.

[VALUE]

99.8782860147213

ODDLYIELD

2021-03-16 • 7 minutes to read

마지막 기간이 홀수(단기 또는 장기)인 유가 증권의 수익률을 반환합니다.

구문

```
ODDLYIELD(<settlement>, <maturity>, <last_interest>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
last_interest	유가 증권의 마지막 이표일입니다.
요율	유가 증권의 이자율입니다.
pr	유가 증권의 가격입니다.
redemption	액면가 \$100당 유가 증권의 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

유가 증권 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- ODDLYIELD는 다음과 같이 계산됩니다.

$$\begin{aligned} \text{ODDLYIELD} = & \frac{(\text{redemption}) + ((\sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}}) - (\text{par}) +}{(\sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}}} \\ & \times (\text{par}) + ((\sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}}) \times \frac{100 \times \text{rate}}{\text{frequency}} \times \frac{1}{(\sum_{i=1}^{\text{NC}} \frac{\text{DSC}_i}{\text{NL}_i})} \end{aligned}$$

여기서

- A_i = i^{th} 의 경과 일수 또는 마지막 이자 지급일부터 상환 전까지 계산하는 홀수 기간 내 마지막 준이표 기간입니다.
- DC_i = i^{th} 에서 계산한 일 수 또는 실제 이표 기간 길이로 구분한 마지막 준이표 기간입니다.
- NC = 홀수 기간에 맞는 준이표 기간의 수입니다. 이 숫자에 분수가 포함되어 있으면 다음 정수로 올림됩니다.
- NL_i = i^{th} 의 날짜의 정상 길이 또는 홀수 이표 기간 내 마지막 준이표 기간입니다.
- settlement, maturity, last_interest는 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, last_interest가 유효한 날짜가 아닙니다.
 - maturity > settlement > last_interest 조건이 충족되지 않습니다.
 - rate < 0.
 - pr ≤ 0.
 - redemption ≤ 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

DATA	인수 설명
2008년 4월 20일	결산일
2008년 6월 15일	만기일
2007년 12월 24일	마지막 이자 지급일
3.75%	표면 금리
\\$99.875	가격
\\$100	상환액
2	frequency는 반기입니다
0	30/360 기준

```
EVALUATE
{
  ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375, 99.875, 100, 2, 0)
}
```

위에 지정된 조건을 사용하여 마지막 기간이 홀수(단기 또는 장기)인 유가 증권 수익률을 반환합니다.

[VALUE]
0.0451922356291692

PDURATION

2021-03-16 • 2 minutes to read

지정된 가치에 투자가 도달하는 데 필요한 기간 수를 반환합니다.

구문

```
PDURATION(<rate>, <pv>, <fv>)
```

매개 변수

용어	정의
요율	기간당 이자율입니다.
pv	투자의 현재 가치입니다.
fv	원하는 투자의 미래 가치입니다.

반환 값

기간 수입니다.

설명

- PDURATION은 다음 수식을 사용합니다.

$$\text{PDURATION} = \frac{\log(\text{fv}) - \log(\text{pv})}{\log(1 + \text{rate})}$$

- 다음과 같은 경우 오류가 반환됩니다.
 - $\text{rate} \leq 0$.
 - $\text{pv} \leq 0$.
 - $\text{fv} \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PDURATION(0.025, 2000, 2200)
}
```

연간 수익률 2.5%의 \$2000 투자가 \$2200에 도달하는 데 필요한 연수를 반환합니다.

[VALUE]
3.85986616262266

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PDURATION(0.025/12, 1000, 1200)
}
```

연간 수익률 2.5%의 \$1000 투자가 \$1200에 도달하는 데 필요한 개월 수를 반환합니다.

[VALUE]
87.6054764193714

PMT

2021-03-16 • 5 minutes to read

원리금 균등 상환 및 일정한 이자율을 기준으로 한 대출 상환 금액을 계산합니다.

구문

```
PMT(<rate>, <nper>, <pv>[, <fv>[, <type>]])
```

매개 변수

용어	정의
요율	대출의 이자율입니다.
nper	대출의 총 상환 횟수입니다.
pv	현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총 금액입니다. 원금이라고도 합니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv가 생략된 경우 BLANK로 간주됩니다.
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type 이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같이 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

참고: PMT의 인수에 대한 자세한 설명은 PV 함수를 참조하세요.

반환 값

단일 대출 상환 액수입니다.

설명

- PMT에서 반환하는 상환 액수에는 원리금이 포함되지만 경우에 따라 대출에 수반되는 세금, 준비금 지급 또는 수수료는 포함되지 않습니다.
- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를, nper에 4를 사용합니다.
- type은 가장 가까운 정수로 반올림됩니다.

- 다음과 같은 경우 오류가 반환됩니다.

- $nper < 1$

팁: 대출 기간 동안 상환된 총 금액을 찾으려면 반환된 PMT 값을 $nper$ 로 곱합니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

예 1

DATA	설명
8%	연간 이자율
10	상환 개월 수
\\$10,000	대출 금액

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PMT(0.08/12, 10, 10000, 0, 1)
}
```

위에서 지정된 조건의 대출에 대해 월초에 상환하는 월 상환 액수를 반환합니다.

[VALUE]
-1030.16432717797

참고: 1030.16432717797은 기간당 상환액입니다. 따라서 대출 기간 동안 상환된 총 금액은 약 $1030.16 * 10 = \$10,301.60$ 입니다. 다시 말해 약 \$301.60의 이자가 상환됩니다.

예제 2

DATA	설명
6%	연간 이자율
18	상환 연수
\\$50,000	대출 금액

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PMT(0.06/12, 18*12, 0, 50000)
}
```

[VALUE]
-129.081160867991

위에 지정된 조건을 사용하여 18년이 끝날 때 \ \$50,000이 있도록 매월 저축할 금액을 반환합니다.

PPMT

2021-03-16 • 5 minutes to read

정기적인 원리금 균등 상환 및 일정한 이자율을 기준으로 투자에 대한 특정 기간의 원금 상환액을 반환합니다.

구문

```
PPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

매개 변수

용어	정의
요율	대출의 이자율입니다.
per	기간을 지정합니다. 1과 nper 사이여야 합니다(두 값 모두 포함).
nper	연금의 총 지급 기간 수입니다.
pv	현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총 금액입니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv가 생략된 경우 BLANK로 간주됩니다.
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type 이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같이 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

참고: PPMT의 인수에 대한 자세한 설명은 PV 함수를 참조하세요.

반환 값

지정된 기간 동안의 원금 상환입니다.

설명

- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를, nper에 4를 사용합니다.

- type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $per < 1$ 또는 $per > nper$
 - $nper < 1$
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

DATA	인수 설명
10%	연간 이자율
2	대출 기간(년)
\\$2,000.00	대출 금액

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PPMT(0.1/12, 1, 2*12, 2000.00)
}
```

위에서 지정한 조건의 대출의 첫 달 원금 상환액을 반환합니다.

[VALUE]
-75.6231860083663

예제 2

DATA	인수 설명
8%	연간 이자율
10	대출 기간(년)
\\$200,000.00	대출 금액

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PPMT(0.08, 10, 10, 200000.00)
}
```

위에서 지정한 조건의 대출의 10번째 해 원금 상환액을 반환합니다.

[VALUE]

-27598.0534624214

PRICE

2021-03-16 • 7 minutes to read

정기 이자를 지급하는 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

구문

```
PRICE(<settlement>, <maturity>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
요율	유가 증권의 연간 표면 금리입니다.
yld	유가 증권의 연간 수익률입니다.
redemption	액면가 \ \$100당 유가 증권의 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

액면가 \ \$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement ≥ maturity.
 - rate < 0.
 - yld < 0.
 - redemption ≤ 0.
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

중요:

- N > 1(N은 결산일과 상환일 사이에 지급할 이표 수)인 경우 **PRICE** 는 다음과 같이 계산됩니다.

$$\text{PRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=1}^N \frac{100 \times \text{rate}}{\text{frequency}} (1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}}{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}}$$

- N = 1(N은 결산일과 상환일 사이에 지급할 이표 수)인 경우 **PRICE** 는 다음과 같이 계산됩니다.

$$\text{DSR} = \text{E} - \text{A}$$

$$\text{T1} = 100 \times \frac{\text{rate}}{\text{frequency}} + \text{redemption}$$

$$\text{T2} = \frac{\text{yld}}{\text{frequency}} \times \frac{\text{DSR}}{\text{E}} + 1$$

$$\text{T3} = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}$$

$$\text{PRICE} = \frac{\text{T1}}{\text{T2}} - \text{T3}$$

여기서

- DSC = 결산일부터 이표일까지의 일 수입니다.
- E = 결산일이 속한 이표 기간의 일 수입니다.
- A = 이표 기간 시작부터 결산일까지의 일 수입니다.

예제

DATA	인수 설명
2008년 2월 15일	결산일
2017년 11월 15일	만기일
5.75%	반기 표면 금리
6.50%	수익률
\\$100	상환액
2	frequency는 반기입니다
0	30/360 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PRICE(DATE(2008,2,15), DATE(2017,11,15), 0.0575, 0.065, 100, 2, 0)
}
```

위에 지정된 조건을 사용하여 채권의 채권 가격을 반환합니다.

[VALUE]
94.6343616213221

PRICEDISC

2021-03-16 • 5 minutes to read

할인된 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

구문

```
PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
할인	유가 증권의 할인율입니다.
redemption	유가 증권의 액면가 \ \$100당 상환액입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

액면가 \ \$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2018년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2018년 1월 1일이 되고, 결산일은 2018년 7월 1일이 되며, 만기일은 발행일인 2018년 1월 1일에서 30년 후인 2048년 1월 1일이 됩니다.

- PRICEDISC는 다음과 같이 계산됩니다.

$$\text{PRICEDISC} = \text{redemption} - \text{discount} \times \text{redemption} \times \frac{\text{DSM}}{\text{B}}$$

여기서

- B = 연 기준에 따른 연간 일 수입니다.
- DSM = settlement부터 maturity까지의 일 수입니다.
- settlement와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - $\text{settlement} \geq \text{maturity}$.
 - $\text{discount} \leq 0$.
 - $\text{redemption} \leq 0$.
 - $\text{basis} < 0$ 또는 $\text{basis} > 4$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	인수 설명
2008년 2월 16일	결산일
2008년 3월 1일	만기일
5.25%	할인율
\\$100	상환액
2	실제/360 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    PRICEDISC(DATE(2008,2,16), DATE(2008,3,1), 0.0525, 100, 2)
}
```

위의 조건이 적용되는 채권의 액면가 \\$100당 채권 가격을 반환합니다.

[VALUE]
99.7958333333333

PRICEMAT

2021-03-16 • 6 minutes to read

만기 시 이자를 지급하는 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

구문

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
이슈	유가 증권의 발행일입니다.
요율	발행일을 기준으로 한 유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

액면가 \ \$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- PRICEMAT은 다음과 같이 계산됩니다.

$$\text{PRICEMAT} = \frac{100 + (\frac{\text{DIM}}{\text{B}} \times \text{rate} \times 100)}{1 + (\frac{\text{DSM}}{\text{B}} \times \text{yld})} - (\frac{\text{A}}{\text{B}} \times \text{rate} \times 100)$$

여기서

- B = 연 기준에 따른 연간 일 수입니다.
- DSM = settlement부터 maturity까지의 일 수입니다.
- DIM = issue부터 maturity까지의 일 수입니다.
- A = issue부터 settlement까지의 일 수입니다.
- settlement, maturity, issue는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 issue가 유효한 날짜가 아닙니다.
 - maturity > settlement > issue 조건이 충족되지 않습니다.
 - rate < 0.
 - yld < 0.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

DATA	설명
2008년 2월 15일	결산일
2008년 4월 13일	만기일
2007년 11월 11일	발행일
6.10%	반기 표면 금리
6.10%	수익률
0	30/360 기준

EVALUATE

```
{  
  PRICEMAT(DATE(2008,2,15), DATE(2008,4,13), DATE(2007,11,11), 0.061, 0.061, 0)  
}
```

위에 지정된 조건을 사용하여 유가 증권의 액면가 \ \$100당 가격을 반환합니다.

[VALUE]

99.9844988755569

PV

2021-03-16 • 8 minutes to read

일정한 이율을 기준으로 대출 또는 투자의 현재 가치를 계산합니다. 주기적인 일정한 지불(예: 모기지 또는 기타 대출) 및/또는 투자 목표인 미래 가치와 함께 PV를 사용할 수 있습니다.

구문

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

매개 변수

용어	정의
요율	기간당 이자율입니다. 예를 들어 연이율 10%로 자동차 대출을 받고 매월 상환하는 경우 월별 이율은 0.1/12 또는 0.0083입니다. 이 경우 수식의 rate에 0.1/12 또는 0.0083을 입력합니다.
nper	연금의 총 지급 기간 수입니다. 예를 들어, 4년 자동차 대출을 받고 매월 상환하는 경우 대출 기간은 4*12(즉 48)입니다. 수식의 nper에 48을 입력합니다.
pmt	각 기간의 상환액이며 연금의 수명 동안 변경될 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료 또는 세금은 포함되지 않습니다. 예를 들어 \$10,000에 대한 월 상환액은 4년 자동차 대출, 12퍼센트 이율의 경우 \$263.33입니다. 수식의 pmt에 -263.33을 입력합니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv가 생략된 경우 BLANK로 간주됩니다. 예를 들어 18년의 특별 프로젝트 비용을 지불하기 위해 \$50,000를 저축하려는 경우 \$50,000가 미래 가치입니다. 그런 다음 이율을 보수적으로 추정하고 매월 저축해야 하는 금액을 결정할 수 있습니다.
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같게 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

반환 값

대출 또는 투자의 현재 가치입니다.

설명

- rate와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 rate에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 rate에 0.12를 사용하고 nper에 4를 사용합니다.
- 다음 함수는 연금에 적용됩니다.
 - CUMIPMT
 - CUMPRINC
 - FV
 - IPMT
 - PMT
 - PPMT
 - PV
 - RATE
 - XIRR
 - XNPV
- 연금은 연속된 기간에 걸쳐 일련의 현금을 일정하게 지불하는 것입니다. 예를 들어 자동차 대출 또는 모기지 연금입니다. 자세한 내용은 각 연금 함수에 대한 설명을 참조하세요.
- 연금 함수에서, 저축 예금과 같이 지불하는 현금음수로 표시됩니다. 배당 수표와 같이 수령하는 현금양수로 표시됩니다. 예를 들어 은행에 \$1,000가 예치될 때 예금자에게는 인수 -1000으로 표시되고 은행에는 인수 1000으로 표시됩니다.
- 하나의 재무 인수는 다른 인수들과 관련하여 해결됩니다.
 - rate가 0이 아닌 경우:
$$\text{pv} \times (1 + \text{rate})^{\text{nper}} + \text{pmt} \times \left(\frac{1 - (1 + \text{rate})^{\text{nper}}}{\text{rate}} \right) + \text{fv} = 0$$
 - rate가 0인 경우:
$$\text{pmt} \times \text{nper} + \text{pv} + \text{fv} = 0$$
- type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $\text{nper} < 1$
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
\$500.00	매월 말에 보험 연금에서 지불되는 금액입니다.
8%	지불되는 금액에 대한 이율입니다.
20	금액이 지불되는 기간(년)입니다.

다음 DAX 쿼리를 실행한다고 가정합니다.

EVALUATE

```
{  
  PV(0.08/12, 12*20, 500.00, 0, 0)  
}
```

위에 지정된 조건을 사용하여 연금의 현재 가치를 반환합니다.

[VALUE]

-59777.1458511878

RATE

2021-03-16 • 5 minutes to read

연금 기간당 이자율을 반환합니다. RATE는 반복으로 계산되며 0개 이상의 해가 있을 수 있습니다. 연속된 RATE 결과가 20회 반복 후에도 0.0000001 내로 수렴하지 않으면 오류가 반환됩니다.

구문

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]]])
```

매개 변수

용어	정의
nper	연금의 총 지급 기간 수입니다.
pmt	각 기간의 지급이며 연금의 수명 동안 변경될 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료 또는 세금은 포함되지 않습니다.
pv	현재 가치, 즉 일련의 미래 지급액이 지금 갖고 있는 가치의 총 금액입니다.
fv	(선택 사항) 미래 가치 또는 최종 납입 후 받고자 하는 현금 잔액입니다. fv를 생략하면 0으로 간주됩니다(예를 들어 대출금의 미래 가치가 0).
형식	(선택 사항) 지급 시점을 나타내는 숫자 0 또는 1입니다. type이 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.
guess	(선택 사항) 이자율에 대한 추측입니다. <ul style="list-style-type: none">- 생략되면 10%로 간주됩니다.- RATE가 수렴되지 않으면 다른 guess 값을 시도해 보세요. guess가 0에서 1 사이인 경우 일반적으로 RATE가 수렴됩니다.

type 매개 변수는 다음 값을 허용합니다.

TYPE을 다음과 같게 설정	납입 기한인 경우
0 또는 생략됨	기간 종료 시
1	기간 시작 시

반환 값

기간당 이자율입니다.

설명

- guess와 nper를 지정하는 데 사용하는 단위가 일관되어야 합니다. 연간 이자율이 12%인 4년짜리 대출을 매월 상환하는 경우 guess에 0.12/12를 사용하고 nper에 4*12를 사용합니다. 동일한 대출을 매년 상환하는 경우 guess에 0.12를, nper에 4를 사용합니다.
- type은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - $nper \leq 0$.
 - 20회 반복 후에 RATE가 0.0000001 이내로 수렴되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
4	대출 연수
-200	월 납입금
8000	대출 금액

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    RATE(4*12, -200, 8000)
}
```

위에 지정된 조건을 사용하여 대출의 월 이자율을 반환합니다.

[VALUE]
0.00770147248820137

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    RATE(4*12, -200, 8000) * 12
}
```

위에 지정된 조건을 사용하여 대출의 연간 이자율을 반환합니다.

[VALUE]
0.0924176698584164

RECEIVED

2021-03-16 • 5 minutes to read

완전 투자 유가 증권에 대해 만기 시 수령하는 금액을 반환합니다.

구문

```
RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
investment	유가 증권의 투자액입니다.
할인	유가 증권의 할인율입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

만기 시 수령 금액입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.

- RECEIVED는 다음과 같이 계산됩니다.

$$RECEIVED = \frac{\text{investment}}{1 - (\text{discount} \times \frac{\text{DIM}}{\text{B}})}$$

여기서

- B = 연 기준에 따른 연간 일수입니다.
- DIM = issue부터 maturity까지의 일수입니다.
- settlement와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - $settlement \geq maturity$.
 - $investment \leq 0$.
 - $discount \leq 0$.
 - $basis < 0$ 또는 $basis > 4$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

DATA	설명
15-Feb-08	결산일(발행일)
15-May-08	만기일
\\$1,000,000.00	투자
5.75%	할인율
2	실제/360 기준

```
EVALUATE
{
    RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

위에 지정된 조건을 사용하여 채권에 대해 만기 시 수령할 총금액을 반환합니다.

[VALUE]
1014584.6544071

RRI

2021-03-16 • 2 minutes to read

투자 증가에 상응하는 이자율을 반환합니다.

구문

```
RRI(<nper>, <pv>, <fv>)
```

매개 변수

용어	정의
nper	투자 기간의 수입입니다.
pv	투자의 현재 가치입니다.
fv	투자의 미래 가치입니다.

반환 값

상응하는 이자율입니다.

설명

- RRI는 다음 수식을 사용하여 계산된 $\text{RRI}(\text{nper}, \text{pv}, \text{fv})$ (기간 수), $\text{RRI}(\text{pv}, \text{nper}, \text{fv})$ (현재 가치), $\text{RRI}(\text{fv}, \text{nper}, \text{pv})$ (미래 가치)를 고려하여 이자율을 반환합니다.

$$\text{RRI}(\text{nper}, \text{pv}, \text{fv}) = \left(\frac{\text{fv}}{\text{pv}} \right)^{\frac{1}{\text{nper}}} - 1$$

- 다음과 같은 경우 오류가 반환됩니다.
 - $\text{nper} \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
$\$10,000$	현재 가치
$\$21,000$	미래 가치
4	투자한 기간(년)

다음 DAX 쿼리를 실행한다고 가정합니다.


```
EVALUATE
{
  RRI(4*12, 10000, 21000)
}
```

위에 지정된 조건을 사용하여 투자 증가에 상응하는 이자율을 반환합니다.

[VALUE]

0.0155771057566627

SLN

2021-03-16 • 2 minutes to read

특정 기간 동안의 자산 정액 감가상각을 반환합니다.

구문

```
SLN(<cost>, <salvage>, <life>)
```

매개 변수

용어	정의
cost	초기 자산 비용입니다.
salvage	감가상각 종료 시의 값입니다(자산의 잔존 가치라고도 함).
응용해 보세요.	자산이 감가상각된 기간 수입니다(자산의 내용연수라고도 함).

반환 값

한 기간 동안의 정액 감가상각입니다.

설명

- 다음과 같은 경우 오류가 반환됩니다.
life = 0.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
\\$30,000	Cost
\\$7,500	잔존 가치
10	내용연수

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    SLN(30000, 7500, 10)
}
```

위에 지정된 조건을 사용하여 연간 감가상각 총당액을 반환합니다.

[VALUE]
2250

SYD

2021-03-16 • 3 minutes to read

연수 합계법에 따른 지정한 기간 동안의 자산 감가상각을 반환합니다.

구문

SYD(<cost>, <salvage>, <life>, <per>)

매개 변수

용어	정의
cost	초기 자산 비용입니다.
salvage	감가상각 종료 시의 값입니다(자산의 잔존 가치라고도 함).
응용해 보세요.	자산이 감가상각된 기간 수입니다(자산의 내용연수라고도 함).
per	기간입니다. life와 동일한 단위를 사용해야 합니다. 1과 life 사이여야 합니다(두 값 모두 포함).

반환 값

연수 합계법에 따른 지정한 기간 동안의 자산 감가상각입니다.

설명

- SYD는 다음과 같이 계산됩니다.

$$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life} + 1)}$$

- 다음과 같은 경우 오류가 반환됩니다.
 - life < 1.
 - per < 1 또는 per > life.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
\\$30,000.00	초기 비용
\\$7,500.00	잔존 가치

DATA	설명
10	수명(년)

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 1)
}
```

위에 지정된 조건을 사용하여 자산의 연수 합계법에 따른 첫 번째 해의 감가상각 총당액을 반환합니다.

[VALUE]

4090.90909090909

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 10)
}
```

위에 지정된 조건을 사용하여 자산의 연수 합계법에 따른 10번째(최종) 해의 감가상각 총당액을 반환합니다.

[VALUE]

409.090909090909

2021-03-16 • 3 minutes to read

구문

- 여 제

DATA	설명
2008년 3월 31일	결산일
2008년 6월 1일	만기일
9.14%	할인율

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    TBILLEQ(
        DATE(2008,3,31),
        DATE(2008,6,1),
        0.0914
    )
}
```

위에 지정된 조건을 사용하여 국채의 채권 상당 수익률을 반환합니다.

[VALUE]
0.094151493565943

TBILLPRICE

2021-03-16 • 3 minutes to read

국채의 액면가 \ \$100당 가격을 반환합니다.

구문

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

매개 변수

용어	정의
settlement	국채의 결산일입니다. 유가 증권 결산일은 국채가 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	국채의 만기일입니다. 만기일은 국채가 만료되는 날짜입니다.
할인	국채의 할인율입니다.

반환 값

국채의 액면가 \ \$100당 가격입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- TBILLPRICE는 다음과 같이 계산됩니다.

$$\text{TBILLPRICE} = 100 \times (1 - \frac{\text{discount} \times \text{DSM}}{360})$$

여기서

- $\text{DSM} = \text{settlement}$ 부터 maturity 까지의 일 수입니다. 결산 후 1년이 넘는 만기일은 제외됩니다.
- settlement 와 maturity 는 정수로 잘립니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity 가 유효한 날짜가 아닙니다.
 - $\text{settlement} \geq \text{maturity}$ 이거나 maturity 가 settlement 후 1년을 넘습니다.
 - $\text{discount} \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2008년 3월 31일	결산일
2008년 6월 1일	만기일
9.0%	할인율

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    TBILLPRICE(DATE(2008,3,31), DATE(2008,6,1), 0.09)
}
```

위에 지정된 조건을 사용하여 국채의 액면가 \ \$100당 가격을 반환합니다.

[VALUE]
98.45

TBILLYIELD

2021-03-16 • 3 minutes to read

국채의 수익률을 반환합니다.

구문

```
TBILLYIELD(<settlement>, <maturity>, <pr>)
```

매개 변수

용어	정의
settlement	국채의 결산일입니다. 유가 증권 결산일은 국채가 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	국채의 만기일입니다. 만기일은 국채가 만료되는 날짜입니다.
pr	액면가 \ \$100당 국채의 가격입니다.

반환 값

국채의 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- TBILLYIELD는 다음과 같이 계산됩니다.

$$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

여기서

- DSM = settlement부터 maturity까지의 일 수입니다. 결산 후 1년이 넘는 만기일은 제외됩니다.
- settlement와 maturity는 정수로 잘립니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity이거나 maturity가 settlement 후 1년을 넘습니다.
 - pr \leq 0.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

DATA	설명
2008년 3월 31일	결산일
2008년 6월 1일	만기일
\\$98.45	액면가 \\$100당 가격

<pre>EVALUATE { TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45) }</pre>
--

위에 지정된 조건을 사용하여 국채의 수익률을 반환합니다.

[VALUE]
0.0914169629253426

VDB

2021-03-16 • 6 minutes to read

이중 체감법 또는 사용자가 지정하는 다른 방법을 사용하여 지정한 기간(부분 기간 포함) 자산의 감가상각액을 반환합니다. VDB는 가변 체감 잔액(Variable Declining Balance)을 나타냅니다.

구문

```
VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[, <no_switch>]])
```

매개 변수

용어	정의
cost	초기 자산 비용입니다.
salvage	감가상각 종료 시의 값입니다(자산의 잔존 가치라고도 함). 이 값은 0일 수 있습니다.
응용해 보세요.	자산이 감가상각된 기간 수입니다(자산의 수명이라고도 함).
start_period	감가상각을 계산하려는 시작 기간입니다. Start_period에는 life와 같은 단위를 사용해야 합니다. 1과 life 사이여야 합니다(두 값 모두 포함).
end_period	감가상각을 계산하려는 종료 기간입니다. End_period에는 life와 같은 단위를 사용해야 합니다. Start_period와 life 사이여야 합니다(두 값 모두 포함).
요소	(선택 사항) 잔액이 체감하는 비율입니다. Factor가 생략된 경우 2로 간주됩니다(이중 체감 잔액법). 이중 체감 잔액법을 사용하지 않으려면 factor를 변경하세요. 이중 체감 잔액법에 대한 자세한 내용은 DDB를 참조하세요.
no_switch	(선택 사항) 감가상각액이 잔존가액보다 큰 경우 정액 감가상각액으로 전환할지 여부를 지정하는 논리값입니다. 생략된 경우 FALSE로 간주됩니다. - no_switch가 TRUE로 평가되면, 감가상각액이 잔존가액보다 큰 경우에도 VDB는 정액 감가상각액으로 전환되지 않습니다. - no_switch가 FALSE로 평가되거나 생략되면, 감가상각액이 잔존가액보다 큰 경우 VDB는 정액 감가상각액으로 전환됩니다.

반환 값

지정된 기간 동안의 감가상각입니다.

설명

- 다음과 같은 경우 오류가 반환됩니다.

- $cost < 0$.
- $salvage < 0$.
- $life < 1$.
- $start_period < 1$ 또는 $start_period > end_period$.
- $end_period < start_period$ 또는 $end_period > life$.
- $factor < 0$.
- `no_switch`는 TRUE 또는 FALSE로 평가되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

DATA	설명
2400	초기 비용
300	잔존 가치
10	수명(년)

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    VDB(2400, 300, 10*365, 0, 1)
}
```

계수 2를 사용하여 첫날의 자산 감가상각액을 반환합니다.

[VALUE]
1.31506849315068

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    VDB(2400, 300, 10*12, 6, 18, 3)
}
```

6th개월과 18th개월 사이의 자산 감가상각액을 반환합니다. 이 계산에서는 계수 3을 사용합니다.

[VALUE]
540.185558199698

예제 3

다음 DAX 쿼리를 실행한다고 가정합니다.

EVALUATE

```
{  
  VDB(2400, 300, 10, 0, 0.875, 1.5)  
}
```

소유하고 있는 첫 번째 회계 연도에 자산의 감가상각액을 반환합니다. 세법에서 감가상각액을 체감 잔액의 150%로 제한한다고 가정합니다. 자산은 회계 연도의 1사분기 중에 구매됩니다.

[VALUE]

315

XIRR

2021-03-16 • 2 minutes to read

반드시 정기적이 아니어도 되는 현금 흐름 일정의 내부 수익률을 반환합니다.

구문

```
XIRR(<table>, <values>, <dates>, [guess])
```

매개 변수

용어	정의
테이블	값 및 날짜 식을 계산해야 하는 테이블입니다.
값	테이블의 각 행에 대한 현금 흐름 값을 반환하는 식입니다.
날짜	테이블의 각 행에 대해 현금 흐름 날짜를 반환하는 식입니다.
guess	(선택 사항) 내부 수익률에 대한 초기 추측입니다. 생략할 경우 기본 추측 0.1이 사용됩니다.

반환 값

지정된 입력의 내부 수익률입니다. 계산에서 올바른 결과가 반환되지 않으면 오류가 반환됩니다.

설명

- 값은 다음 함수를 충족하는 비율로 계산됩니다.

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

위치:

- P_j 는 j^{th} 지급
- d_j 는 j^{th} 지급일
- d_1 는 첫 번째 지급일
- 일련의 현금 흐름 값에는 양수와 음수가 적어도 하나씩 포함되어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

아래 수식은 CashFlows 테이블의 내부 수익률을 계산합니다.

```
= XIRR( CashFlows, [Payment], [Date] )
```

DATE	결제
2014/1/1	-10000
2014/3/1	2750
10/30/2014	4250
2/15/2015	3250
2015/4/1	2750

수익률 = 37.49%

XNPV

2021-03-16 • 2 minutes to read

반드시 정기적이 아니어도 되는 현금 흐름의 일정에 대한 현재 가치를 반환합니다.

구문

```
XNPV(<table>, <values>, <dates>, <rate>)
```

매개 변수

용어	정의
테이블	값 및 날짜 식을 계산해야 하는 테이블입니다.
값	테이블의 각 행에 대한 현금 흐름 값을 반환하는 식입니다.
날짜	테이블의 각 행에 대해 현금 흐름 날짜를 반환하는 식입니다.
요율	테이블의 각 행에 대한 현금 흐름에 적용할 할인율입니다.

반환 값

순 현재 가치입니다.

설명

- 값은 다음 총합으로 계산됩니다.

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

위치:

- P_j 는 j^{th} 지급
- d_j 는 j^{th} 지급일
- d_1 는 첫 번째 지급일
- 일련의 현금 흐름 값에는 양수와 음수가 적어도 하나씩 포함되어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 CashFlows 테이블의 현재 가치를 계산합니다.

```
= XNPV( CashFlows, [Payment], [Date], 0.09 )
```

DATE	결제
2014/1/1	-10000
2014/3/1	2750
10/30/2014	4250
2/15/2015	3250
2015/4/1	2750

현재 가치 = 2086.65

YIELD

2021-03-16 • 7 minutes to read

정기 이자를 지급하는 유가 증권 수익률을 반환합니다. YIELD를 사용하여 채권 수익률을 계산합니다.

구문

```
YIELD(<settlement>, <maturity>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
요율	유가 증권 연간 확정 금리입니다.
pr	유가 증권 액면가 \$100당 가격입니다.
redemption	액면가 \$100당 유가 증권 상환액입니다.
frequency	연간 이표 지급 횟수입니다. 연간 지급인 경우 frequency = 1 이고, 반기 지급은 frequency = 2, 분기 지급은 frequency = 4 입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

유가 증권 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매 한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- 상환까지 남은 기간이 1 이표 기간 이하인 경우 YIELD는 다음과 같이 계산됩니다.

$$\text{YIELD} = \frac{\left(\frac{\text{redemption}}{100} + \frac{\text{rate}}{\text{frequency}} \right) - \left(\frac{\text{par}}{100} + \left(\frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}} \right) \right)}{\left(\frac{\text{par}}{100} + \left(\frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}} \right) \right) \times \frac{\text{frequency}}{\text{E}} \times \text{DSR}}$$

여기서

- A = 이표 기간 시작부터 결산일까지의 일 수입니다(경과 일 수).
- DSR = 결산일부터 상환일까지의 일 수입니다.
- E = 이표 기간의 일 수입니다.
- 상환까지 남은 기간이 1 이표 기간을 넘는 경우 YIELD는 100회 반복을 통해 계산됩니다. 이 해결 방법은 PRICE 함수에 사용되는 수식을 기반으로 뉴턴법을 사용합니다. 수익률이 가격에 근접한다고 할 때 수익률은 예상 가격까지 변합니다.
- settlement와 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - rate < 0 .
 - pr ≤ 0 .
 - redemption ≤ 0 .
 - frequency가 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4 .
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
15-Feb-08	결산일
15-Nov-16	만기일
5.75%	확정 금리
95.04287	가격

DATA	설명
\\$100	상환액
2	frequency는 반기입니다(위 참조)
0	30/360 기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
  YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

위에 지정된 조건이 적용되는 채권의 수익률을 반환합니다.

[VALUE]
0.06500000068807314

YIELDDISC

2021-03-16 • 5 minutes to read

할인된 유가 증권의 연간 수익률을 반환합니다.

구문

```
YIELDDISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
pr	유가 증권의 액면가 \ \$100당 가격입니다.
redemption	유가 증권의 액면가 \ \$100당 상환액입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

연간 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.

- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement와 maturity는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity가 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - pr \leq 0.
 - redemption \leq 0.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	-
2008년 2월 16일	결산일
2008년 3월 1일	만기일
99.795	가격
\\$100	상환액
2	실제/360 기준

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2)
}
```

위에 지정된 조건에 따라 유가 증권의 연간 수익률을 반환합니다.

[VALUE]
0.0528225719868583

YIELDMAT

2021-03-16 • 5 minutes to read

만기 시에 이자를 지급하는 유가 증권의 연간 수익률을 반환합니다.

구문

```
YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

매개 변수

용어	정의
settlement	유가 증권의 결산일입니다. 유가 증권 결산일은 유가 증권이 구매자에게 판매되는 발행일 이후의 날짜입니다.
maturity	유가 증권의 만기일입니다. 만기일은 유가 증권이 만료되는 날짜입니다.
이슈	유가 증권의 발행일입니다.
요율	발행일을 기준으로 한 유가 증권의 이자율입니다.
pr	유가 증권의 액면가 \$100당 가격입니다.
basis	(선택 사항) 사용할 일수 기준 유형입니다. basis가 생략된 경우 0으로 간주됩니다. 허용되는 값은 이 테이블 아래 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기준	일 수 기준
0 또는 생략됨	미국(NASD) 30/360
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

반환 값

연간 수익률입니다.

설명

- 날짜는 계산에 사용될 수 있도록 순차적인 일련 번호로 저장됩니다. DAX에서는 1899년 12월 30일이 0일 이고 2008년 1월 1일은 1899년 12월 30일 후 39,448일째이므로 39,448일입니다.
- 결산일은 구매자가 채권 등의 이표를 구매한 날짜입니다. 만기일은 이표가 만료되는 날짜입니다. 예를 들어 30년 만기 채권이 2008년 1월 1일에 발행되고 6개월 후에 구매자가 구매한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일이 되고, 결산일은 2008년 7월 1일이 되며, 만기일은 발행일인 2008년 1월 1일에서 30년 후인 2038년 1월 1일이 됩니다.
- settlement, maturity, issue는 정수로 잘립니다.
- basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 issue가 유효한 날짜가 아닙니다.
 - maturity > settlement > issue 조건이 충족되지 않습니다.
 - rate < 0.
 - pr ≤ 0.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DATA	설명
2008년 3월 15일	결산일
2008년 11월 3일	만기일
2007년 11월 8일	발행일
6.25%	반기 표면 금리
100.0123	가격
0	30/360 기준(위 참조)

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
{
    YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625, 100.0123, 0)
}
```

위에 지정된 조건을 사용하여 유가 증권의 수익률을 반환합니다.

[VALUE]
0.0609543336915387

정보 함수

2021-03-16 • 7 minutes to read

DAX 정보 함수는 인수로 제공된 셀이나 행을 확인하고 값이 필요한 형식과 일치하는지 여부를 알려줍니다. 예를 들어 ISERROR 함수는 참조하는 값에 오류가 포함되어 있는 경우 TRUE를 반환합니다.

이 범주의 함수

함수	DESCRIPTION
CONTAINS	참조된 모든 열의 값이 존재하거나 해당 열에 포함되어 있으면 true를 반환하고, 그렇지 않으면 함수는 false를 반환합니다.
CONTAINSROW	값의 행이 존재하거나 테이블에 포함되면 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.
CONTAINSSTRING	한 문자열에 다른 문자열이 포함되어 있는지를 나타내는 TRUE 또는 FALSE를 반환합니다.
CONTAINSSTRINGEXACT	한 문자열에 다른 문자열이 포함되어 있는지를 나타내는 TRUE 또는 FALSE를 반환합니다.
CUSTOMDATA	연결 문자열에 있는 CustomData 속성의 내용을 반환합니다.
HASONEFILTER	<i>columnName</i> 에서 직접 필터링된 값의 수가 1인 경우 TRUE를 반환합니다. 1이 아닌 경우 FALSE를 반환합니다.
HASONEVALUE	<i>columnName</i> 에 대한 컨텍스트가 고유 값 하나로만 필터링된 경우 TRUE를 반환합니다. 그 외의 경우 FALSE입니다.
ISBLANK	값이 공백인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.
ISCROSSFILTERED	<i>columnName</i> 또는 동일한 테이블이나 관련 테이블의 다른 열이 필터링될 때 TRUE를 반환합니다.
ISEMPTY	테이블이 비어 있는지 확인합니다.
ISERROR	값이 오류인지를 확인하고 TRUE 또는 FALSE를 반환합니다.
ISEVEN	숫자가 짝수이면 TRUE를, 숫자가 홀수이면 FALSE를 반환합니다.
ISFILTERED	<i>columnName</i> 을 직접 필터링하는 경우 TRUE를 반환합니다.
ISINSCOPE	지정한 열이 수준 계층 구조의 수준인 경우 true를 반환합니다.
ISLOGICAL	값이 논리 값(TRUE 또는 FALSE)인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

함수	DESCRIPTION
ISNONTEXT	값이 텍스트가 아닌지 여부(빈 셀은 텍스트가 아님)를 확인하고 TRUE 또는 FALSE를 반환합니다.
ISNUMBER	값이 숫자인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.
ISODD	숫자가 홀수이면 TRUE를, 숫자가 짝수이면 FALSE를 반환합니다.
ISONORAFTER	Start At 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 충족하는 행에 대해 true를 반환하는 부울 함수입니다.
ISSELECTEDMEASURE	컨텍스트에 있는 측정값이 측정값 목록에 지정된 측정값 중 하나임을 확인하기 위해 계산 항목의 식에서 사용됩니다.
ISSUBTOTAL	인수로 지정된 열의 소계 값이 행에 포함되어 있으면 True를 반환하고 포함되어 있지 않으면 False를 반환하는 또 다른 열을 SUMMARIZE 식에 만듭니다.
ISTEXT	값이 텍스트인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.
NONVISUAL	SUMMARIZECOLUMNS 식의 값 필터를 비시각적 개체로 표시합니다.
SELECTEDMEASURE	컨텍스트에 있는 측정값을 참조하기 위해 계산 항목의 식에 사용됩니다.
SELECTEDMEASUREFORMATSTRING	컨텍스트에 있는 측정값의 형식 문자열을 검색하기 위해 계산 항목의 식에서 사용됩니다.
SELECTEDMEASURENAME	컨텍스트에 있는 측정값을 이름으로 확인하기 위해 계산 항목의 식에서 사용됩니다.
USERNAME	연결 시 시스템에 제공된 자격 증명의 도메인 이름과 사용자 이름을 반환합니다.
USEROBJECTID	현재 사용자의 개체 ID 또는 SID를 반환합니다.
USERPRINCIPALNAME	사용자 계정 이름을 반환합니다.

CONTAINS

2021-03-16 • 3 minutes to read

참조된 모든 열의 값이 존재하거나 해당 열에 포함되어 있으면 **true**를 반환하고, 그렇지 않으면 함수는 **false**를 반환합니다.

구문

```
CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)
```

매개 변수

용어	정의
테이블	데이터의 테이블을 반환하는 DAX 식입니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.
값	<i>columnName</i> 에서 찾으려는 단일 스칼라 값을 반환하는 DAX 식입니다. 식은 인수 목록에 전달되기 전에 정확히 한번 계산되어야 합니다.

반환 값

지정된 각 *value*가 해당 열에 포함되어 있거나 *columnName*에서 찾을 수 있으면 **TRUE** 값이고, 그렇지 않으면 함수는 **FALSE**를 반환합니다.

설명

- *columnName* 및 *value* 인수는 쌍으로 제공해야 합니다. 그렇지 않으면 오류가 반환됩니다.
- *columnName*은 지정된 *table*이나 *table*과 관련된 테이블에 속해 있어야 합니다.
- *columnName*이 관련 테이블의 열을 참조하는 경우 정규화된 이름을 사용해야 합니다. 그렇지 않으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제는 제품 214의 인터넷 판매와 고객 11185에 대한 인터넷 판매가 동시에 발생했는지 여부를 알려주는 측정값을 만듭니다.

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

CONTAINSROW 함수

2021-03-16 • 3 minutes to read

값의 행이 존재하거나 테이블에 포함되면 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.

구문

```
CONTAINSROW(<tableExpr>, <scalarExpr>[, <scalarExpr>, ...])
```

매개 변수

용어	정의
scalarExprN	스칼라 값을 반환하는 모든 유효한 DAX 식입니다.
tableExpr	데이터 테이블을 반환하는 모든 유효한 DAX 식입니다.

반환 값

TRUE 또는 FALSE입니다.

설명

- 구문을 제외하고는 **IN** 연산자와 CONTAINSROW 함수는 기능적으로 동일합니다.

```
<scalarExpr> IN <tableExpr>  
( <scalarExpr1>, <scalarExpr2>, ... ) IN <tableExpr>
```

- scalarExprN의 수는 tableExpr의 열 수와 일치해야 합니다.
- NOT IN은 DAX의 연산자가 아닙니다. IN 연산자의 논리 부정을 수행하려면 전체 식 앞에 NOT을 추가합니다. 예를 들어, NOT [Color] IN { "Red", "Yellow", "Blue" }와 같습니다.
- = 연산자와 달리 IN 연산자와 CONTAINSROW 함수는 엄격한 비교를 수행합니다. 예를 들어 BLANK 값은 0과 일치하지 않습니다.

예 1

다음과 같은 동등한 DAX 쿼리:

```
EVALUATE FILTER(ALL(DimProduct[Color]), [Color] IN { "Red", "Yellow", "Blue" })  
ORDER BY [Color]
```

그리고

```
EVALUATE FILTER(ALL(DimProduct[Color]), ([Color]) IN { "Red", "Yellow", "Blue" })  
ORDER BY [Color]
```

그리고

```
EVALUATE FILTER(ALL(DimProduct[Color]), CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

단일 열을 포함하는 다음과 같은 테이블을 반환합니다.

DIMPRODUCT[COLOR]
파랑
빨간색
노란색

예제 2

다음과 같은 동등한 DAX 쿼리:

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), ([Color], [Size]) IN { ("Black", "L") })
```

및

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), CONTAINSROW({ ("Black", "L") }, [Color], [Size]))
```

반환:

DIMPRODUCT[COLOR]	DIMPRODUCT[SIZE]
검정	L

예제 3

다음과 같은 동등한 DAX 쿼리:

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT [Color] IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

및

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

단일 열을 포함하는 다음과 같은 테이블을 반환합니다.

DIMPRODUCT[COLOR]
검정
회색
다중

DIMPRODUCT[COLOR]
해당 없음
은
은색\검정
흰색

CONTAINSSTRING

2021-03-16 • 2 minutes to read

한 문자열에 다른 문자열이 포함되어 있는지를 나타내는 TRUE 또는 FALSE를 반환합니다.

구문

```
CONTAINSSTRING(<within_text>, <find_text>)
```

매개 변수

용어	정의
within_text	find_text를 검색하려는 텍스트입니다.
find_text	찾으려는 텍스트입니다.

반환 값

find_text가 within_text의 부분 문자열이면 TRUE이고, 부분 문자열이 아니면 FALSE입니다.

설명

- CONTAINSSTRING은 대/소문자를 구분하지 않습니다.
- [?] 및 [*] 와일드카드 문자를 사용할 수 있습니다. 와일드카드 문자를 이스케이프하려면 [~]를 사용합니다.

예제

DAX 쿼리

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRING("abcd", "bc"),
        "Case 2", CONTAINSSTRING("abcd", "BC"),
        "Case 3", CONTAINSSTRING("abcd", "a*d"),
        "Case 4", CONTAINSSTRING("abcd", "ef")
    )
```

반환

[CASE 1]	[CASE 2]	[CASE 3]	[CASE 4]
TRUE	TRUE	TRUE	FALSE

CONTAINSSTRINGEXACT

2021-03-16 • 2 minutes to read

한 문자열에 다른 문자열이 포함되어 있는지를 나타내는 TRUE 또는 FALSE를 반환합니다.

구문

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

매개 변수

용어	정의
within_text	find_text를 검색하려는 텍스트입니다.
find_text	찾으려는 텍스트입니다.

반환 값

find_text가 within_text의 부분 문자열이면 TRUE이고, 부분 문자열이 아니면 FALSE입니다.

설명

CONTAINSSTRINGEXACT는 대/소문자를 구분합니다.

예제

DAX 쿼리

```
EVALUATE
ROW(
    "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),
    "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),
    "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),
    "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
)
```

반환

[CASE 1]	[CASE 2]	[CASE 3]	[CASE 4]
TRUE	FALSE	FALSE	FALSE

CUSTOMDATA

2021-03-16 • 2 minutes to read

연결 문자열에 있는 **CustomData** 속성의 내용을 반환합니다.

구문

```
CUSTOMDATA()
```

반환 값

연결 문자열에 있는 **CustomData** 속성의 내용입니다.

연결 시 **CustomData** 속성이 정의되지 않은 경우에는 공백입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 수식은 CustomData 특성이 "OK" 로 설정되었는지 확인합니다.

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No custom data in connection string property or unexpected value")
```

HASONEFILTER

2021-03-16 • 2 minutes to read

*columnName*에서 직접 필터링된 값의 수가 1일 경우 **TRUE** 를 반환합니다. 그렇지 않으면 **FALSE** 를 반환합니다.

구문

```
HASONEFILTER(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

*columnName*에서 직접 필터링된 값의 수가 1일 경우 **TRUE** 입니다. 그렇지 않으면 **FALSE** 를 반환합니다.

설명

- 이 함수는 HASONEVALUE()와 비슷하지만 HASONEVALUE()는 교차 필터를 기반으로 작동하는 반면 HASONEFILTER()는 직접 필터로 작동하는 차이점이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 HASONEFILTER()를 사용하여 필터가 있는 경우에는 ResellerSales_USD[ProductKey])에 대한 필터를 반환하거나, ResellerSales_USD[ProductKey])에 필터가 없거나 둘 이상의 필터가 있는 경우 BLANK를 반환하는 방법을 보여 줍니다.

```
= IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[ProductKey]),BLANK())
```

HASONEVALUE

2021-03-16 • 2 minutes to read

*columnName*에 대한 컨텍스트가 고유 값 하나로만 필터링된 경우 **TRUE**를 반환합니다. 그렇지 않으면 **FALSE**입니다.

구문

```
HASONEVALUE(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

*columnName*의 컨텍스트가 고유 값 하나로만 필터링된 경우 **TRUE**입니다. 그렇지 않으면 **FALSE**입니다.

설명

- HASONEVALUE()에 해당하는 식이 `COUNTROWS(VALUESEX(columnName))) = 1`입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 측정 수식은 미리 정의된 시나리오를 기준으로 백분율을 추정하기 위해 컨텍스트가 하나의 값으로 분할되는지 확인합니다. 이 경우, Reseller Sales를 2007년 매출과 비교하려면 컨텍스트가 1년 단위로 필터링되는지 확인해야 합니다. 또한 비교가 의미가 없는 경우 BLANK를 반환할 수 있습니다.

```
=  
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])/CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=2007),BLANK())
```

ISBLANK

2021-03-16 • 2 minutes to read

값이 공백인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISBLANK(<value>)
```

매개 변수

용어	정의
값	테스트하려는 값 또는 식입니다.

반환 값

값이 공백이면 부울 값이 TRUE이고 공백이 아니면 FALSE입니다.

예제

이 수식은 판매의 증가 또는 감소 비율을 작년과 비교하여 계산합니다. 이 예제에서는 0으로 나누기 오류를 방지하도록 IF 함수를 사용하여 작년의 판매 값을 확인합니다.

```
//Sales to Previous Year Ratio

= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])
, BLANK()
, ( 'CalculatedMeasures'[Total Sales]-'CalculatedMeasures'[PreviousYearTotalSales] )
/'CalculatedMeasures'[PreviousYearTotalSales])
```

결과,

행 레이블	TOTAL SALES	작년 총 판매액	작년 대비 판매 비율
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%
총합계	\$102,554,625.71		

참고 항목

[정보 함수](#)

ISCROSSFILTERED

2021-03-16 • 2 minutes to read

columnName 이나 동일한 테이블 또는 관련 테이블의 다른 열이 필터링된 경우 TRUE를 반환합니다.

구문

```
ISCROSSFILTERED(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

columnName 또는 동일한 테이블이나 관련 테이블의 다른 열이 필터링되는 경우 **TRUE** 입니다. 그렇지 않으면 **FALSE** 를 반환합니다.

설명

- 동일한 테이블 또는 관련 테이블의 다른 열에 적용된 필터가 필터링을 통해 *columnName*에 영향을 줄 경우 열이 교차 필터링된다고 말합니다. 필터가 해당 열에 적용될 때는 열이 **직접** 필터링된다고 합니다.
- 관련 함수 [ISFILTERED](#) 함수는 *columnName*이 직접 필터링될 때 TRUE를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[ISFILTERED](#) 함수

[FILTERS](#) 함수

[HASONEFILTER](#) 함수

[HASONEVALUE](#) 함수

IEMPTY

2021-04-26 • 2 minutes to read

테이블이 비어 있는지 확인합니다.

구문

```
IEMPTY(<table_expression>)
```

매개 변수

용어	정의
table_expression	테이블을 반환하는 테이블 참조 또는 DAX 식입니다.

반환 값

테이블이 비어 있으면(행이 없음) True이고, 아니면 False입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

아래의 'Info' 테이블에서,

국가	시스템 상태	국가	합계
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
미국	CA	5	500
미국	WA	10	900

```
EVALUATE  
ROW("Any countries with count > 25?", NOT(IEMPTY(FILTER(Info, [County]>25))))
```

반환 값: **FALSE**

ISERROR

2021-03-16 • 2 minutes to read

값이 오류인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISERROR(<value>)
```

매개 변수

용어	정의
값	테스트할 값입니다.

반환 값

값이 오류이면 TRUE의 부울 값, 아니면 FALSE입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 총 재판매인 매출 대비 총 인터넷 매출을 계산합니다. ISERROR 함수는 0으로 나누기와 같은 오류를 확인하는 데 사용됩니다. 오류가 있는 경우 빈 값이 반환되고, 아니면 비율이 반환됩니다.

```
= IF( ISERROR(
    SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
)
, BLANK()
, SUM('ResellerSales_USD'[SalesAmount_USD])
  /SUM('InternetSales_USD'[SalesAmount_USD])
)
```

참고 항목

[정보 함수](#)

[IFERROR 함수](#)

[IF 함수](#)

ISEVEN

2021-03-16 • 2 minutes to read

숫자가 짝수이면 TRUE를, 숫자가 홀수이면 FALSE를 반환합니다.

구문

ISEVEN(number)

매개 변수

용어	정의
숫자	테스트할 값입니다. 숫자가 정수가 아니면 잘립니다.

반환 값

숫자가 짝수이면 TRUE를, 숫자가 홀수이면 FALSE를 반환합니다.

설명

- 숫자가 비 숫자인 경우 ISEVEN은 #VALUE! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

ISFILTERED

2021-03-16 • 2 minutes to read

columnName 을 직접 필터링하는 경우 TRUE를 반환합니다. 열에 필터가 없거나 동일한 테이블 또는 관련 테이블의 다른 열이 필터링되어 필터링이 발생하는 경우 함수는 **FALSE** 를 반환합니다.

구문

```
ISFILTERED(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

columnName 을 직접 필터링하는 경우 TRUE입니다.

설명

- 열 위에 하나 이상의 필터가 적용될 때 *columnName* 이 필터링됩니다. 동일한 테이블 또는 관련 테이블의 다른 열에 적용되는 필터가 *columnName* 열도 필터링하여 영향을 미치는 경우 교차 필터링됩니다.
- 관련 함수 [ISCROSSFILTERED](#)는 *columnName* 또는 동일한 테이블이나 관련 테이블의 다른 열이 필터링될 때 TRUE를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[ISCROSSFILTERED](#) 함수

[FILTERS](#) 함수

[HASONEFILTER](#) 함수

[HASONEVALUE](#) 함수

ISINSCOPE

2021-03-16 • 2 minutes to read

지정된 열이 수준 계층 구조의 수준이면 true를 반환합니다.

구문

```
ISINSCOPE(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

지정한 열이 수준 계층 구조의 수준인 경우 True입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```

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]

```

그러면 다음 값이 반환됩니다.

DIMPRODUCT[CATEGORY]	DIMPRODUCT[SUBCATEGORY]	[CATEGORY SUBTOTAL]	[SUBCATEGORY SUBTOTAL]	[SALES]	[% OF PARENT]
		TRUE	TRUE	28,397,095.65	100.00
Accessories		FALSE	TRUE	78,951.00	0.28
Accessories	자전거 랙	FALSE	FALSE	39,360.00	49.85
Accessories	자전거 스탠드	FALSE	FALSE	39,591.00	50.15
Bikes		FALSE	TRUE	28,318,144.65	99.72
Bikes	산악용 자전거	FALSE	FALSE	9,952,759.56	35.15
Bikes	도로용 자전거	FALSE	FALSE	14,520,584.04	51.28
Bikes	여행용 자전거	FALSE	FALSE	3,844,801.05	13.58

참고 항목

SUMMARIZECOLUMNS 함수
CALCULATE 함수

ISLOGICAL

2021-03-16 • 2 minutes to read

값이 논리 값(TRUE 또는 FALSE)인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISLOGICAL(<value>)
```

매개 변수

용어	정의
값	테스트할 값입니다.

반환 값

값이 논리 값이면 TRUE이고, TRUE 또는 FALSE 이외의 값이면 FALSE입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 세 가지 샘플에서는 ISLOGICAL의 동작을 보여 줍니다.

```
//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")
```

참고 항목

[정보 함수](#)

ISNONTEXT

2021-03-16 • 2 minutes to read

값이 텍스트가 아닌지 여부(빈 셀은 텍스트가 아님)를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISNONTEXT(<value>)
```

매개 변수

용어	정의
값	확인할 값입니다.

반환 값

값이 텍스트가 아니거나 비어 있으면 TRUE이고, 값이 텍스트이면 FALSE입니다.

설명

- 빈 문자열은 텍스트로 간주됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 ISNONTEXT 함수의 동작을 보여 줍니다.

```
//RETURNS: Is Non-Text
= IF(ISNONTEXT(1), "Is Non-Text", "Is Text")

//RETURNS: Is Non-Text
= IF(ISNONTEXT(BLANK()), "Is Non-Text", "Is Text")

//RETURNS: Is Text
= IF(ISNONTEXT(""), "Is Non-Text", "Is Text")
```

참고 항목

[정보 함수](#)

ISNUMBER

2021-03-16 • 2 minutes to read

값이 숫자인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISNUMBER(<value>)
```

매개 변수

용어	정의
값	테스트할 값입니다.

반환 값

값이 숫자이면 TRUE이고, 그렇지 않으면 FALSE입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 세 가지 샘플에서는 ISNUMBER의 동작을 보여 줍니다.

```
//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")
```

참고 항목

[정보 함수](#)

ISODD

2021-03-16 • 2 minutes to read

숫자가 홀수이면 TRUE를, 숫자가 짝수이면 FALSE를 반환합니다.

구문

ISODD(number)

매개 변수

용어	정의
숫자	테스트할 값입니다. 숫자가 정수가 아니면 잘립니다.

반환 값

숫자가 홀수이면 TRUE를, 숫자가 짝수이면 FALSE를 반환합니다.

설명

- 숫자가 아닌 경우 ISODD는 #VALUE! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

ISONORAFTER

2021-03-16 • 3 minutes to read

'Start At' 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 충족하는 행에 대해 true를 반환하는 부울 함수입니다.

이 함수는 세 개로 구성된 변수를 사용합니다. 처음 두 값은 비교할 식이며 세 번째 매개 변수는 정렬 순서를 나타냅니다. 정렬 순서는 오름차순(기본값)이거나 내림차순일 수 있습니다.

정렬 순서에 따라 첫 번째 매개 변수가 두 번째 매개 변수와 비교됩니다. 정렬 순서가 오름차순이면 첫 번째 매개 변수가 두 번째 매개 변수보다 크거나 같은지를 비교합니다. 정렬 순서가 내림차순이면 두 번째 매개 변수가 첫 번째 매개 변수보다 작거나 같은지를 비교합니다.

구문

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>,  
<scalar_expression>[, sort_order]]...)
```

매개 변수

용어	정의
스칼라 식	열 참조, 정수 또는 문자열 값과 같은 스칼라 값을 반환하는 식입니다. 일반적으로 첫 번째 매개 변수는 열 참조이고 두 번째 매개 변수는 스칼라 값입니다.
정렬 순서	(옵션) 열이 정렬되는 순서입니다. 오름차순(ASC)이거나 내림차순(DEC)일 수 있습니다. 기본적으로 정렬 순서는 오름차순입니다.

반환 값

True 또는 False입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

테이블 이름: 'Info'

국가	시스템 상태	개수	합계
IND	JK	20	800
IND`	MH	25	1000
IND	WB	10	900

국가	시스템 상태	개수	합계
미국	CA	5	500
미국	WA	10	900

`FILTER(Info, ISONORAFTER(Info[Country], "IND", ASC, Info[State], "MH", ASC))`

ISSELECTEDMEASURE

2021-03-16 • 2 minutes to read

컨텍스트에 있는 측정값이 측정값 목록에 지정된 측정값 중 하나임을 확인하기 위해 계산 항목의 식에서 사용 됩니다.

구문

```
ISSELECTEDMEASURE( M1, M2, ... )
```

매개 변수

용어	정의
M1, M2, ...	측정값 목록입니다.

반환 값

현재 컨텍스트에 있는 측정값이 매개 변수 목록에 지정된 측정값 중 하나인지를 나타내는 부울입니다.

설명

- 계산 항목에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 계산 항목 식은 현재 측정값이 매개 변수 목록에 지정된 측정값 중 하나인지를 확인합니다. 측정값의 이름이 변경되면 수식 수정은 식의 이름 변경 내용을 반영합니다.

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

참고 항목

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

ISSUBTOTAL

2021-03-16 • 2 minutes to read

인수로 지정된 열의 소계 값이 행에 포함되어 있으면 True를 반환하고 포함되어 있지 않으면 False를 반환하는 또 다른 열을 [SUMMARIZE](#) 식에 만듭니다.

구문

```
ISSUBTOTAL(<columnName>)
```

[SUMMARIZE](#) 사용

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, {<expression>|ISSUBTOTAL(<columnName>)}]...)
```

매개 변수

용어	정의
columnName	SUMMARIZE 함수의 table에 있는 열 또는 관련 table-table에 있는 열의 이름입니다.

반환 값

인수로 지정된 열의 소계 값이 행에 있으면 True 값을 반환하고 없으면 False 값을 반환합니다.

설명

- 이 함수는 [SUMMARIZE](#) 함수의 식에서만 사용할 수 있습니다.
- 이 함수 앞에 부울 열의 이름이 와야 합니다.

예제

[SUMMARIZE](#)를 참조하세요.

ISTEXT

2021-03-16 • 2 minutes to read

값이 텍스트인지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

```
ISTEXT(<value>)
```

매개 변수

용어	정의
값	확인할 값입니다.

반환 값

값이 텍스트이면 TRUE이고, 텍스트가 아니면 FALSE입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 ISTEXT 함수의 동작을 보여 줍니다.

```
//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")
```

참고 항목

[정보 함수](#)

NONVISUAL

2021-03-16 • 2 minutes to read

[SUMMARIZECOLUMNS](#) 식의 값 필터를 비시각적 개체로 표시합니다. 이 함수는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다.

구문

```
NONVISUAL(<expression>)
```

매개 변수

용어	정의
expression	단일 값(테이블이 아님)을 반환하는 DAX 식입니다.

반환 값

값 테이블입니다.

설명

- [SUMMARIZECOLUMNS](#)의 값 필터를 측정값에 영향을 주지 않고 group-by 열에만 적용하도록 표시합니다.
- 이 함수는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다. [SUMMARIZECOLUMNS](#) 함수의 filterTable 인수 또는 [ROLLUPADDISSUBTOTAL](#) 또는 [ROLLUPISSUBTOTAL](#) 함수의 groupLevelFilter 인수 중 하나로 사용됩니다.

예제

[SUMMARIZECOLUMNS](#)를 참조하세요.

SELECTEDMEASURE

2021-03-16 • 2 minutes to read

컨텍스트에 있는 측정값을 참조하기 위해 계산 항목의 식에 사용됩니다.

구문

```
SELECTEDMEASURE()
```

매개 변수

None

반환 값

계산 항목을 계산할 때 현재 컨텍스트에 있는 측정값에 대한 참조입니다.

설명

- 계산 항목에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 계산 항목 식은 컨텍스트에 있는 측정값에 대한 연간 누계를 계산합니다.

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

참고 항목

[SELECTEDMEASURENAME](#)

[ISSELECTEDMEASURE](#)

SELECTEDMEASUREFORMATSTRING

2021-03-16 • 2 minutes to read

컨텍스트에 있는 측정값의 형식 문자열을 검색하기 위해 계산 항목의 식에서 사용됩니다.

구문

```
SELECTEDMEASUREFORMATSTRING()
```

매개 변수

None

반환 값

계산 항목을 계산할 때 현재 컨텍스트에 있는 측정값의 형식 문자열을 포함하는 문자열입니다.

설명

- 이 함수는 계산 그룹의 계산 항목에 대해서만 식에서 참조할 수 있습니다. 계산 항목의 **형식 문자열** 식 속성에서 사용하기 위해 작성되었습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 식은 계산 항목의 형식 문자열 식 속성에서 계산됩니다. 필터 컨텍스트에 단일 통화가 있으면 DimCurrency[FormatString] 열에서 형식 문자열이 검색됩니다. 그렇지 않으면 컨텍스트에 있는 측정값의 형식 문자열이 사용됩니다.

```
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

참고 항목

[SELECTEDMEASURE](#)
[ISSELECTEDMEASURE](#)

SELECTEDMEASURENAME

2021-03-16 • 2 minutes to read

컨텍스트에 있는 측정값을 이름으로 확인하기 위해 계산 항목의 식에서 사용됩니다.

구문

```
SELECTEDMEASURENAME()
```

매개 변수

None

반환 값

계산 항목이 평가될 때 현재 컨텍스트에 있는 측정값의 이름을 포함하는 문자열 값입니다.

설명

- 계산 항목에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 계산 그룹을 제작할 때 디버깅 목적으로 사용되는 경우가 많습니다.

예제

다음 계산 항목 식은 현재 측정값이 Expense Ratio인지를 확인하고 계산 논리를 조건부로 적용합니다. 이 확인은 문자열 비교를 기반으로 하기 때문에 수식 수정의 영향을 받지 않으며 개체 이름 바꾸기가 자동으로 반영되지 않습니다. 수식 수정을 활용하는 유사한 비교에 대해서는 ISSLECTEDMEASURE 함수를 대신 참조하세요.

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

참고 항목

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

USERNAME

2021-03-16 • 2 minutes to read

연결 시 시스템에 제공된 자격 증명의 도메인 이름과 사용자 이름을 반환합니다.

구문

```
USERNAME()
```

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

연결 시 시스템에 지정된 자격 증명의 사용자 이름입니다.

예제

다음 수식은 사용자 로그인인 UsersTable의 일부인지 확인합니다.

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

USEROBJECTID

2021-03-16 • 2 minutes to read

Azure AD 또는 보안 식별자(SID)에서 현재 사용자의 개체 ID를 반환합니다.

구문

```
USEROBJECTID()
```

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

Azure AD에서 Power BI 또는 Azure Analysis Services 모델에 대한 현재 사용자의 개체 ID를 반환하거나 SID에서 SQL Server Analysis Services 모델에 대한 현재 사용자 개체 ID를 반환합니다.

USERPRINCIPALNAME

2021-03-16 • 2 minutes to read

사용자 계정 이름을 반환합니다.

구문

```
USERPRINCIPALNAME()
```

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

연결 시의 userprincipalname입니다.

논리 함수

2021-03-17 • 2 minutes to read

논리 함수는 식에 대해 작동하여 식의 값 또는 집합에 대한 정보를 반환합니다. 예를 들어 IF 함수를 사용하여 식의 결과를 확인하고 조건부 결과를 만들 수 있습니다.

이 범주의 함수

함수	DESCRIPTION
및	인수가 둘 다 TRUE인지 확인하고 인수가 둘 다 TRUE인 경우 TRUE를 반환합니다.
COALESCE	BLANK로 평가되지 않는 첫 번째 식입니다.
FALSE	논리 값 FALSE를 반환합니다.
IF	조건을 검사하여 TRUE이면 값 하나를 반환하고 그렇지 않으면 두 번째 값을 반환합니다.
IF.EAGER	조건을 검사하여 TRUE이면 값 하나를 반환하고 그렇지 않으면 두 번째 값을 반환합니다. 조건식에 상관없이 분기 식을 항상 실행하는 '즉시' 실행 계획을 사용합니다.
IFERROR	식을 계산하고 식에서 오류를 반환하는 경우 지정된 값을 반환합니다.
다음이 아님	FALSE를 TRUE로 또는 TRUE를 FALSE로 변경합니다.
OR	인수 중 하나가 TRUE인지를 확인하여 TRUE를 반환합니다.
SWITCH	값 목록에 대해 식을 계산하고 가능한 여러 결과 식 중 하나를 반환합니다.
TRUE	논리 값 TRUE를 반환합니다.

AND

2021-03-16 • 3 minutes to read

인수가 둘 다 TRUE인지 확인하고 인수가 둘 다 TRUE인 경우 TRUE를 반환합니다. 그렇지 않으면 false를 반환합니다.

구문

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

매개 변수

용어	정의
logical_1, logical_2	테스트할 논리값입니다.

반환 값

테스트하는 값의 조합에 따라 true 또는 false를 반환합니다.

설명

DAX의 **AND** 함수에서는 인수가 두(2) 개만 허용됩니다. 여러 식에서 AND 연산을 수행해야 하는 경우 일련의 계산을 만들어 사용할 수도 있지만 AND 연산자(&&)를 사용하여 모든 항목을 더 간단한 식으로 조인하는 것이 좋습니다.

예 1

다음 수식은 AND 함수의 구문을 보여 줍니다.

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

AND 함수에 인수로 전달된 조건이 둘 다 True이므로 수식은 "All True"를 반환합니다.

예제 2

다음 샘플에서는 중첩된 수식에 AND 함수를 사용하여 두 계산 집합을 동시에 비교합니다. 이 수식은 각 제품 범주에 대해 올해와 작년의 인터넷 채널 판매가 동일 기간의 재판매인 채널 판매보다 높은지 확인합니다. 두 조건이 다 True이면 이 수식은 각 범주에 대해 "Internet hit" 값을 반환합니다.

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

반환

행 레이블	2005	2006	2007	2008	-	총 합계
빔 쇼츠						
자전거 랙						
자전거 스탠드				Internet Hit		
물병 및 거치대				Internet Hit		
바텀 브라켓						
브레이크						
캡						
Chains						
클리너						
크랭크세트						
변속기						
펜더				Internet Hit		
포크						
장갑						
핸들바						
헤드세트						
헬멧						
하이드레이션팩						
저지						
조명						
잠금						
산악용 자전거						

행 레이블	2005	2006	2007	2008	-	총 합계
산악용 프레임						
패니어						
페달						
펌프						
도로용 자전거						
도로용 프레임						
안장						
반바지						
양말						
타이즈						
타이어 및 튜브				Internet Hit		
여행용 자전거						
여행용 프레임						
조끼						
바퀴						
총 합계						

참고 항목

논리 함수

COALESCE

2021-03-16 • 2 minutes to read

BLANK로 평가되지 않는 첫 번째 식입니다. 모든 식이 BLANK로 평가되면 BLANK가 반환됩니다.

구문

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

매개 변수

용어	정의
expression	스칼라 식을 반환하는 DAX 식입니다.

반환 값

식 중 하나에서 오는 스칼라 값 또는 모든 식이 BLANK로 평가되는 경우 BLANK입니다.

설명

입력 식은 데이터 형식이 다를 수 있습니다.

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

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

BLANK로 평가되지 않는 첫 번째 식인 **10**을 반환합니다.

예제 2

다음 DAX 식을 실행한다고 가정합니다.

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

FactInternetSales 테이블의 SalesAmount 열에 있는 모든 값의 합계 또는 **0**을 반환합니다. 이 값을 사용하여 총 판매액의 BLANK 값을 **0**으로 변환할 수 있습니다.

FALSE

2021-03-16 • 2 minutes to read

논리 값 FALSE를 반환합니다.

구문

```
FALSE()
```

반환 값

항상 FALSE입니다.

설명

FALSE라는 단어도 논리 값 FALSE로 해석됩니다.

예제

이 수식은 'InternetSales_USD'[SalesAmount_USD] 열의 값이 200000보다 작거나 같은 경우 논리 값 FALSE를 반환합니다.

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

다음 테이블에서는 Row Labels의 'ProductCategory'[ProductCategoryName] 및 Column Labels의 'DateTime'[CalendarYear]에서 예제 수식이 사용된 경우의 결과를 보여 줍니다.

행 레이블	2005	2006	2007	2008	-	총 합계
Accessories	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Clothing	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
구성 요소	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
총 합계	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

참고 항목

[TRUE 함수](#)

[NOT 함수](#)

[IF 함수](#)

IF

2021-03-17 • 5 minutes to read

조건을 검사하여 TRUE이면 값 하나를 반환하고 그렇지 않으면 두 번째 값을 반환합니다.

구문

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

매개 변수

용어	정의
logical_test	TRUE 또는 FALSE로 계산할 수 있는 값 또는 식입니다.
value_if_true	논리 테스트가 TRUE인 경우 반환되는 값입니다.
value_if_false	(선택 사항) 논리 테스트가 FALSE인 경우 반환되는 값입니다. 생략하면 BLANK가 반환됩니다.

반환 값

value_if_true, value_if_false 또는 BLANK 중 하나입니다.

설명

- IF 함수는 value_if_true 와 value_if_false 의 데이터 형식이 서로 다른 경우 variant 데이터 형식을 반환할 수 있지만 value_if_true 와 value_if_false 모두 숫자 데이터 형식인 경우 단일 데이터 형식을 반환하려고 합니다. 후자의 경우 IF 함수는 두 값에 맞게 데이터 형식을 암시적으로 변환합니다.

예를 들어 IF(<condition>, TRUE(), 0) 수식은 TRUE 또는 0을 반환하지만 IF(<condition>, 1.0, 0) 수식은 value_if_false 가 전체 데이터 유형임에도 불구하고 10진수 값만 반환합니다. 암시적 데이터 형식 변환에 대한 자세한 내용은 [데이터 형식](#)을 참조하세요.

- 조건식에 상관없이 분기 식을 실행하려면 IFEAGER를 대신 사용합니다.

예

다음 Product 테이블 계산 열 정의는 정가에 따라 각 제품을 분류하는 다양한 방법으로 IF 함수를 사용합니다.

첫 번째 예제는 List Price 열 값이 500 미만인지 여부를 테스트합니다. 이 조건이 true이면 Low 값이 반환됩니다. value_if_false 값이 없으므로 BLANK가 반환됩니다.

이 문서의 예는 Power BI Desktop 샘플 모델에 추가할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low"  
)
```

두 번째 예제도 동일한 테스트를 사용하지만 이번에는 **value_if_false** 값이 포함됩니다. 따라서 수식은 각 제품을 **Low** 또는 **High** 로 분류합니다.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    "High"  
)
```

세 번째 예제도 동일한 테스트를 사용하지만 이번에는 추가 테스트를 수행하기 위해 IF 함수가 중첩됩니다. 따라서 수식은 각 제품을 **Low**, **Medium** 또는 **High** 로 분류합니다.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    IF(  
        'Product'[List Price] < 1500,  
        "Medium",  
        "High"  
    )  
)
```

TIP

여러 IF 함수를 중첩해야 하는 경우 **SWITCH** 함수를 사용하는 것이 더 좋습니다. 이 함수는 가능한 값 셋 이상을 반환하는 식을 작성할 수 있는 세련된 방법을 제공합니다.

참고 항목

[IF.EAGER 함수](#)

[SWITCH 함수\(DAX\)](#)

[논리 함수](#)

IF.EAGER

2021-03-17 • 3 minutes to read

조건을 검사하여 TRUE이면 값 하나를 반환하고 그렇지 않으면 두 번째 값을 반환합니다. 조건식에 상관없이 분기 식을 항상 실행하는 '즉시' 실행 계획을 사용합니다.

구문

```
IF.EAGER(<logical_test>, <value_if_true>[, <value_if_false>])
```

매개 변수

용어	정의
logical_test	TRUE 또는 FALSE로 계산할 수 있는 값 또는 식입니다.
value_if_true	논리 테스트가 TRUE인 경우 반환되는 값입니다.
value_if_false	(선택 사항) 논리 테스트가 FALSE인 경우 반환되는 값입니다. 생략하면 BLANK가 반환됩니다.

반환 값

value_if_true, value_if_false 또는 BLANK 중 하나입니다.

설명

- IF.EAGER 함수는 value_if_true와 value_if_false의 데이터 형식이 서로 다른 경우 variant 데이터 형식을 반환할 수 있지만 **value_if_true** 와 **value_if_false** 모두 숫자 데이터 형식인 경우 단일 데이터 형식을 반환하려고 합니다. 후자의 경우 IF.EAGER 함수는 두 값에 맞게 데이터 형식을 암시적으로 변환합니다.

예를 들어 IF.EAGER(<condition>, TRUE(), 0) 수식은 TRUE 또는 0을 반환하지만

IF.EAGER(<condition>, 1.0, 0) 수식은 **value_if_false** 가 전체 데이터 유형임에도 불구하고 10진수 값만 반환합니다. 암시적 데이터 형식 변환에 대한 자세한 내용은 [데이터 형식](#)을 참조하세요.

- IF.EAGER는 IF 함수와 기능 동작이 동일하지만, 실행 계획의 차이로 인해 성능이 다를 수 있습니다.

IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>) 는 다음 DAX 식과 실행 계획이 동일합니다.

```
VAR _value_if_true = <value_if_true>
VAR _value_if_false = <value_if_false>
RETURN
IF (<logical_test>, _value_if_true, _value_if_false)
```

참고: 두 분기 식은 조건식에 상관없이 평가됩니다.

예제

[IF 예제](#)를 참조하세요.

참고 항목

[IF 함수](#)

[논리 함수](#)

IFERROR

2021-03-16 • 2 minutes to read

식을 계산하고 식에서 오류를 반환하는 경우 지정된 값을 반환합니다. 그렇지 않으면 식 자체의 값을 반환합니다.

구문

```
IFERROR(value, value_if_error)
```

매개 변수

용어	정의
값	값 또는 식입니다.
value_if_error	값 또는 식입니다.

반환 값

값 과 동일한 형식의 스칼라입니다.

설명

- IFERROR 함수를 사용하여 식에서 오류를 트래핑하고 처리할 수 있습니다.
- value** 또는 **value_if_error** 가 빈 셀이면 IFERROR는 이 값을 빈 문자열 값("")으로 처리합니다.
- IFERROR 함수는 IF 함수를 기반으로 하며 동일한 오류 메시지를 사용하지만 인수가 더 적습니다. IFERROR 함수와 IF 함수 간의 관계는 다음과 같습니다.

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

A와 B에 대해 반환되는 값은 동일한 데이터 형식이어야 합니다. 따라서 **value** 에 사용되는 열 또는 식과 **value_if_error** 에 대해 반환되는 값은 동일한 데이터 형식이어야 합니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 25/0 식이 오류로 계산되는 경우 9999를 반환합니다. 식에서 오류 이외의 값을 반환하는 경우 해당 값은 호출하는 식으로 전달됩니다.

```
= IFERROR(25/0,9999)
```

참고 항목

[논리 함수](#)

NOT

2021-03-16 • 2 minutes to read

FALSE를 TRUE로 또는 TRUE를 FALSE로 변경합니다.

구문

```
NOT(<logical>)
```

매개 변수

용어	정의
논리	TRUE 또는 FALSE로 평가할 수 있는 값 또는 식입니다.

반환 값

TRUE 또는 FALSE입니다.

예제

다음 예제에서는 IF 함수를 설명하기 위해 만든 계산 열에서 값을 검색합니다. 이 예제에서 계산 열은 기본 이름 **Calculated Column1** 을 사용하여 이름을 지정했으며 `= IF([Orders]<300,"true","false")` 수식을 포함합니다.

이 수식은 [Orders] 열의 값을 확인하고 주문 수가 300개보다 적으면 "true"를 반환합니다."

이 새 계산 열 **Calculated Column2** 를 만들고 다음 수식을 입력합니다.

```
= NOT([CalculatedColumn1])
```

Calculated Column1 의 각 행에 대해 "true" 및 "false" 값은 논리 값 TRUE 또는 FALSE로 해석되고 NOT 함수는 해당 값의 논리적 반대를 반환합니다.

참고 항목

[TRUE 함수](#)

[FALSE 함수](#)

[IF 함수](#)

또는

2021-03-16 • 4 minutes to read

인수 중 하나가 TRUE인지를 확인하여 TRUE를 반환합니다. 이 함수는 인수가 둘 다 FALSE이면 FALSE를 반환합니다.

구문

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

매개 변수

용어	정의
logical_1, logical_2	테스트할 논리값입니다.

반환 값

부울 값입니다. 두 인수 중 하나라도 TRUE이면 값은 TRUE입니다. 인수가 둘 다 FALSE이면 값은 FALSE입니다.

설명

- DAX의 OR 함수에서는 인수가 두(2) 개만 허용됩니다. 여러 식에서 OR 연산을 수행해야 하는 경우 일련의 계산을 만들 수도 있으나 OR 연산자(||)를 사용하면 모든 항목을 더 간단한 식으로 조인할 수 있습니다.
- 이 함수는 첫 번째 TRUE 인수까지 인수를 평가하고 TRUE를 반환합니다.

예제

다음 예제에서는 OR 함수를 사용하여 Circle of Excellence에 속하는 판매 직원을 가져오는 방법을 보여 줍니다. Circle of Excellence는 백만 달러 이상의 여행용 자전거 판매를 달성했거나 2007년에 250만 달러 이상의 판매를 달성한 사원을 인정해 줍니다.

```
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"
, ""
)
```

반환

행 레이블	2005	2006	2007	2008	-	총 합계
Abbas, Syed E						
Alberts, Amy E						

행 레이블	2005	2006	2007	2008	-	총 합계
Ansman-Wolfe, Pamela O						
Blythe, Michael G	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Campbell, David R						
Carson, Jillian	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Ito, Shu K						
Jiang, Stephen Y						
Mensa-Annan, Tete A						
Mitchell, Linda C	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Pak, Jae B	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Reiter, Tsvi Michael						
Saraiva, José Edvaldo	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Tsofilas, Lynn N						
Valdez, Rachel B						
Vargas, Garrett R						
Varkey Chudukatil, Ranjit R						Circle of Excellence
총 합계	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence

참고 항목

[논리 함수](#)

SWITCH

2021-03-16 • 2 minutes to read

값 목록에 대해 식을 계산하고 가능한 여러 결과 식 중 하나를 반환합니다.

구문

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

매개 변수

용어	정의
expression	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.
value	<i>expression</i> 의 결과와 일치하는 상수 값입니다.
result	<i>expression</i> 의 결과가 해당 <i>value</i> 와 일치하는 경우 계산할 스칼라 식입니다.
else	<i>expression</i> 의 결과가 <i>value</i> 인수와 일치하지 않는 경우 계산할 스칼라 식입니다.

반환 값

result 식 중 하나에서 오거나(*value*와 일치하는 값이 있는 경우) *else* 식에서 오는(*value*와 일치하는 값이 없는 경우) 스칼라 값입니다.

설명

모든 *result* 식 및 *else* 식은 데이터 형식이 동일해야 합니다.

예제

다음 예에서는 월 이름의 계산 열을 만듭니다.

```
= SWITCH([Month], 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" )
```

TRUE

2021-03-16 • 2 minutes to read

논리 값 TRUE를 반환합니다.

구문

```
TRUE()
```

반환 값

항상 TRUE입니다.

설명

TRUE라는 단어도 논리 값 TRUE로 해석됩니다.

예제

이 수식은 'InternetSales_USD'[SalesAmount_USD] 열의 값이 200000보다 큰 경우 논리 값 TRUE를 반환합니다.

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

다음 테이블에서는 Row Labels에 'ProductCategory'[ProductCategoryName]가 있고 Column Labels에 'DateTime'[CalendarYear]이 있는 피벗 테이블 또는 시각화에서 예제 수식이 사용된 경우의 결과를 보여 줍니다.

행 레이블	2005	2006	2007	2008	-	총 합계
Accessories	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Clothing	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
구성 요소	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
총 합계	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

참고 항목

[FALSE](#)

[다음이 아님](#)

[IF](#)

수학 및 삼각 함수

2021-03-16 • 7 minutes to read

DAX(Data Analysis Expressions)의 수학 함수는 Excel의 수학 및 삼각 함수와 매우 비슷합니다. 이 섹션에서는 DAX에서 제공하는 수학 함수에 대해 설명합니다.

이 범주의 함수

함수	DESCRIPTION
ABS	숫자의 절대값을 반환합니다.
ACOS	숫자의 아크코사인 또는 역코사인을 반환합니다.
ACOSH	숫자의 역쌍곡 코사인을 반환합니다.
ACOT	숫자의 아크코탄젠트 또는 역코탄젠트를 반환합니다.
ACOTH	숫자의 역쌍곡 코탄젠트를 반환합니다.
ASIN	숫자의 아크사인 또는 역사인을 반환합니다.
ASINH	숫자의 역쌍곡 사인을 반환합니다.
ATAN	숫자의 아크탄젠트 또는 역 탄젠트를 반환합니다.
ATANH	숫자의 역쌍곡 탄젠트를 반환합니다.
CEILING	숫자를 가장 가까운 정수 또는 가장 가까운 significance 배수로 반올림합니다.
COMBIN	지정된 항목 수의 조합 수를 반환합니다.
COMBINA	지정된 항목 수의 조합 수(반복 포함)를 반환합니다.
CONVERT	데이터 형식의 식을 다른 형식의 식으로 변환합니다.
COS	지정된 각도의 코사인을 반환합니다.
COSH	숫자의 쌍곡선 코사인을 반환합니다.
CURRENCY	인수를 계산하고 결과를 통화 데이터 형식으로 반환합니다.
DEGREES	라디안을 각도로 변환합니다.
DIVIDE	나누기를 수행하고 0으로 나누기에 대한 대체 결과 또는 BLANK()를 반환합니다.
EVEN	가장 가까운 짝수로 반올림된 수를 반환합니다.

함수	DESCRIPTION
EXP	지정된 숫자를 지수로 한 e의 거듭제곱을 반환합니다.
FACT	1*2*3*...* 계열과 같고 지정된 숫자로 끝나는, 숫자의 계승을 반환합니다.
FLOOR	숫자를 0에 가까워지도록 가장 가까운 significance 배수로 내림합니다.
GCD	둘 이상의 정수의 최대 공약수 반환합니다.
INT	숫자를 가장 가까운 정수로 내림합니다.
ISO.CEILING	숫자를 가장 가까운 정수 또는 가장 가까운 significance 배수로 반올림합니다.
LCM	정수의 최소 공배수를 반환합니다.
LN	숫자의 자연 로그를 반환합니다.
LOG	지정한 밑에 대한 로그를 반환합니다.
LOG10	밑이 10인 로그를 반환합니다.
MROUND	원하는 배수로 반올림된 수를 반환합니다.
ODD	가장 가까운 홀수로 반올림된 수를 반환합니다.
PI	원주율(파이:3.14159265358979) 값을 15자리까지 반환합니다.
POWER	숫자를 거듭제곱한 결과를 반환합니다.
PRODUCT	열에 있는 숫자의 곱을 반환합니다.
PRODUCTX	테이블의 각 행에 대해 계산된 식의 곱을 반환합니다.
QUOTIENT	나누기를 수행하고 나누기 결과의 정수 부분만 반환합니다.
RADIANS	도를 라디안으로 변환합니다.
RAND	0보다 크거나 같고 1보다 작은, 균등하게 분포된 난수를 반환합니다.
RANDBETWEEN	지정한 두 숫자 사이의 범위에서 난수를 반환합니다.
ROUND	숫자를 지정된 자릿수로 반올림합니다.
ROUNDDOWN	숫자를 0에 가까워지도록 내림합니다.
ROUNDUP	0에서 멀어지도록 숫자를 반올림합니다.

함수	DESCRIPTION
SIGN	수의 부호, 계산 결과 또는 열의 값을 결정합니다.
SQRT	숫자의 제곱근을 반환합니다.
SUM	열에 있는 모든 수를 더합니다.
SUMX	테이블의 각 행에 대해 평가된 식의 합계를 반환합니다.
TRUNC	숫자의 소수 부분을 제거하여 숫자를 정수로 자릅니다.

ABS

2021-03-16 • 2 minutes to read

숫자의 절대값을 반환합니다.

구문

```
ABS(<number>)
```

매개 변수

용어	정의
number	절대 값을 반환하는 숫자입니다.

반환 값

10진수입니다.

설명

숫자의 절대값은 부호가 없는 10진수(정수 또는 소수)입니다. 양수가 필요한 함수에 중첩된 경우 식에서 음수가 아닌 숫자만 반환되도록 하기 위해 ABS 함수를 사용할 수 있습니다.

예제

다음 예제는 새 계산 열인 **DealerMarkup**에 사용할 수 있는, 정가와 총판 가격 간 차액의 절대값을 반환합니다.

```
= ABS([DealerPrice]-[ListPrice])
```

참고 항목

[수학 및 삼각 함수](#)

[SIGN 함수](#)

ACOS

2021-03-16 • 2 minutes to read

숫자의 아크코사인 또는 역코사인을 반환합니다. 아크코사인은 코사인이 *number* 인 각도입니다. 반환된 각도는 0(영)~pi 범위의 라디안 단위로 지정됩니다.

구문

ACOS(*number*)

매개 변수

용어	정의
Number	반환하려는 각도의 코사인은 -1에서 1 사이여야 합니다.

반환 값

숫자의 아크코사인 또는 역코사인을 반환합니다.

설명

결과를 라디안에서 도로 변환하려면 180/PI()를 곱하거나 DEGREES 함수를 사용합니다.

예제

수식	설명	결과
= ACOS(-0.5)	-0.5의 아크코사인(라디안), $2\pi/3$.	2.094395102
= ACOS(-0.5)*180/PI()	-0.5의 아크코사인(도).	120

ACOSH

2021-03-16 • 2 minutes to read

숫자의 역쌍곡 코사인을 반환합니다. 숫자는 1보다 크거나 같아야 합니다. 역쌍곡 코사인은 쌍곡 코사인이 숫자
인 값이므로 ACOSH(COSH (숫자))는 숫자와 같습니다.

구문

ACOSH(number)

매개 변수

용어	정의
number	모든 실수는 1보다 크거나 같습니다.

반환 값

숫자의 역쌍곡 코사인을 반환합니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= ACOSH(1)	1의 역하이퍼볼릭 코사인입니다.	0
= ACOSH(10)	10의 역하이퍼볼릭 코사인입니다.	2.993228

ACOT

2021-03-16 • 2 minutes to read

숫자의 아크코탄젠트 또는 역코탄젠트의 주요 값을 반환합니다.

구문

`ACOT(number)`

매개 변수

용어	정의
Number	원하는 각도의 코사인입니다. 실수여야 합니다.

반환 값

단일 10진수 값입니다.

ACOTH

2021-03-16 • 2 minutes to read

숫자의 역쌍곡 코탄젠트를 반환합니다.

구문

```
ACOTH(number)
```

매개 변수

용어	정의
Number	숫자의 절대값은 1보다 커야 합니다.

반환 값

단일 10진수 값입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

ASIN

2021-03-16 • 2 minutes to read

숫자의 아크사인 또는 역사인을 반환합니다. 아크사인은 사인이 *number* 인 각도입니다. 반환된 각도는 $-\pi/2 \sim \pi/2$ 범위의 라디안 단위로 지정됩니다.

구문

ASIN(*number*)

매개 변수

용어	정의
<i>number</i>	반환하려는 각도의 사인은 -1에서 1 사이여야 합니다.

반환 값

숫자의 아크사인 또는 역사인을 반환합니다.

설명

아크사인을 도 단위로 표현하려면 결과에 $180/\pi()$ 를 곱하거나 DEGREES 함수를 사용합니다.

예제

수식	설명	결과
= ASIN(-0.5)	-0.5의 아크사인(라디안), $-\pi/6$	-0.523598776
= ASIN(-0.5)*180/PI()	-0.5의 아크사인(도)	-30
= DEGREES(ASIN(-0.5))	-0.5의 아크사인(도)	-30

ASINH

2021-03-16 • 2 minutes to read

숫자의 역쌍곡 사인을 반환합니다. 역쌍곡 사인은 쌍곡 사인이 *number* 인 값이므로 ASINH(SINH(number))는 *number* 와 같습니다.

구문

```
ASINH(number)
```

매개 변수

용어	정의
number	실수입니다.

반환 값

숫자의 역쌍곡 사인을 반환합니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= ASINH(-2.5)	-2.5의 역쌍곡 사인	-1.647231146
= ASINH(10)	10의 역쌍곡 사인	2.99822295

ATAN

2021-03-16 • 2 minutes to read

숫자의 아크탄젠트 또는 역 탄젠트를 반환합니다. 아크탄젠트는 탄젠트가 인수인 *number*입니다. 반환된 각도는 $-\pi/2 \sim \pi/2$ 범위의 라디안 단위로 지정됩니다.

구문

ATAN(*number*)

매개 변수

용어	정의
<i>number</i>	반환할 각도의 탄젠트입니다.

반환 값

숫자의 역쌍곡 탄젠트를 반환합니다.

설명

아크탄젠트를 각도로 표현하려면 결과에 $180/\pi()$ 를 곱하거나 DEGREES 함수를 사용합니다.

예제

수식	설명	결과
= ATAN(1)	라디안 1의 아크탄젠트, $\pi/4$	0.785398163
= ATAN(1)*180/PI()	각도 1의 아크탄젠트	45

ATANH

2021-03-16 • 2 minutes to read

숫자의 역쌍곡 탄젠트를 반환합니다. 숫자는 -1에서 1 사이여야 합니다(-1과 1은 제외). 역쌍곡 탄젠트는 쌍곡 탄젠트가 *number* 인 값이므로 ATANH(TANH(number))는 *number* 와 같습니다.

구문

ATANH(*number*)

매개 변수

용어	정의
<i>number</i>	1에서 -1 사이의 실수입니다.

반환 값

숫자의 역쌍곡 탄젠트를 반환합니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= ATANH(0.76159416)	0.76159416의 역쌍곡 탄젠트	1.00000001
= ATANH(-0.1)		-0.100335348

참고 항목

[ATAN 함수](#)

CEILING

2021-03-16 • 4 minutes to read

숫자를 가장 가까운 정수 또는 가장 가까운 significance 배수로 반올림합니다.

구문

CEILING(<number>, <significance>)

매개 변수

용어	정의
숫자	반올림할 숫자이거나 숫자가 포함된 열에 대한 참조입니다.
significance	반올림할 때 기준이 되는 significance의 배수입니다. 예를 들어 가장 가까운 정수로 반올림하려면 1을 입력합니다.

반환 값

지정된 대로 반올림된 숫자입니다.

설명

- DAX에는 다음과 같은 차이가 있는 두 개의 CEILING 함수가 있습니다.
 - CEILING 함수는 Excel의 CEILING 함수 동작을 에뮬레이트합니다.
 - ISO.CEILING 함수는 최대값을 확인하기 위해 ISO 표준 동작을 따릅니다.
- 두 함수는 양수에 대해 동일한 값을 반환하지만, 음수에 대해서는 서로 다른 값을 반환합니다. significance의 양의 배수를 사용할 경우 CEILING과 ISO.CEILING은 모두 음수를 양의 무한대 방향으로 올립니다. significance의 음의 배수를 사용할 경우 CEILING은 음수를 음의 무한대 방향으로 내림하는 반면, ISO.CEILING은 음수를 양의 무한대 방향으로 올립니다.
- 반환 형식은 일반적으로 significance 인수와 동일한 형식입니다. 단, 다음과 같은 경우는 예외입니다.
 - 숫자 인수 형식이 통화이면 반환 형식은 통화입니다.
 - significance 인수 형식이 부울이면 반환 형식은 정수입니다.
 - significance 인수 형식이 숫자가 아니면 반환 형식은 실수입니다.

예 1

다음 수식은 4.45를 반환합니다. 이 수식은 가격에 더 작은 단위를 사용하지 않으려는 경우에 유용할 수 있습니다. 기존 제품 가격이 \$4.42이면 CEILING을 사용하여 가격을 가장 가까운 5센트 단위로 반올림할 수 있습니다.

= CEILING(4.42,0.05)

예제 2

다음 수식은 이전 예제와 비슷한 결과를 반환하지만, **ProductPrice** 열에 저장된 숫자 값을 사용합니다.

```
= CEILING([ProductPrice],0.05)
```

참고 항목

[수학 및 삼각 함수](#)

[FLOOR 함수](#)

[ISO.CEILING 함수](#)

[ROUNDUP 함수](#)

COMBIN

2021-03-16 • 2 minutes to read

지정된 항목 수의 조합 수를 반환합니다. COMBIN을 사용하여 지정된 항목 수에 대해 가능한 총 그룹 수를 확인합니다.

구문

```
COMBIN(number, number_chosen)
```

매개 변수

용어	정의
number	항목 수입니다.
number_chosen	각 조합의 항목 수입니다.

반환 값

지정된 항목 수의 조합 수를 반환합니다.

설명

- 숫자 인수는 정수로 잘립니다.
- 두 인수 중 하나가 숫자가 아닌 경우 COMBIN은 #VALUE! 오류 값을 반환합니다.
- number < 0, number_chosen < 0 또는 number < number_chosen인 경우 COMBIN은 #NUM! 오류 값을 반환합니다.
- 조합은 내부 순서와 관계없이 항목의 세트 또는 하위 세트입니다. 조합은 내부 순서가 중요한 순열과는 다릅니다.
- 조합 수는 다음과 같으며, 여기서 number = n 이고 number_chosen = k 입니다.

$${}_n\text{choose } k = \frac{P_{\{k,n\}}}{k!} = \frac{n!}{k!(n-k)!}$$

Where

$$P_{\{k,n\}} = \frac{n!}{(n-k)!}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= COMBIN(8,2)	8명의 후보로 구성할 수 있는 2인 일조 팀입니다.	28

COMBINA

2021-03-16 • 2 minutes to read

지정된 항목 수의 조합 수(반복 포함)를 반환합니다.

구문

```
COMBINA(number, number_chosen)
```

매개 변수

용어	정의
number	0보다 크거나 같고 Number_chosen보다 크거나 같아야 합니다. 정수가 아닌 값은 잘립니다.
number_chosen	0보다 크거나 같아야 합니다. 정수가 아닌 값은 잘립니다.

반환 값

지정된 항목 수의 조합 수(반복 포함)를 반환합니다.

설명

- 두 인수 중 하나의 값이 제약 조건을 벗어나면 COMBINA는 #NUM! 오류 값을 반환합니다.
- 두 인수 중 하나가 숫자가 아닌 값이면 COMBINA는 #VALUE! 오류 값을 반환합니다.
- 다음 수식이 사용되며, 여기서 \$N\$은 숫자이고 \$M\$은 Number_chosen입니다.

$${}_{N+M-1}C_{N-1}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= COMBINA(4,3)	4와 3의 조합 수(반복 포함)를 반환합니다.	20
= COMBINA(10,3)	10과 3의 조합 수(반복 포함)를 반환합니다.	220

CONVERT

2021-03-16 • 2 minutes to read

한 데이터 형식의 식을 다른 데이터 형식으로 변환합니다.

구문

```
CONVERT(<Expression>, <Datatype>)
```

매개 변수

용어	정의
식	유효한 식입니다.
Datatype	다음에 포함하는 열거형입니다. INTEGER(정수), DOUBLE(10진수), STRING(텍스트), BOOLEAN(True/False), CURRENCY(고정 10진수), DATETIME(날짜, 시간 등).

반환 값

<Datatype>으로 변환된 <Expression> 값을 반환합니다.

설명

- 값을 지정된 데이터 형식으로 변환할 수 없으면 함수는 오류를 반환합니다.
- DAX 계산 열은 단일 데이터 형식이어야 합니다. 정수 열의 MEDIAN 및 MEDIANX 함수는 정수 또는 double인 혼합 데이터 형식을 반환하므로 다음 계산 열 식은 오류를 결과로 반환합니다.

```
MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned])
```
- 데이터 형식을 혼합하지 않으려면 항상 double 데이터 형식을 반환하도록 식을 변경합니다. 예를 들면 다음과 같습니다.

```
MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned], DOUBLE))
```
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DAX 쿼리

```
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

반환

[VALUE]

2

COS

2021-03-16 • 2 minutes to read

지정된 각도의 코사인을 반환합니다.

구문

`COS(number)`

매개 변수

용어	정의
number	필수 사항입니다. 코사인을 구하려는 라디안 단위 각도입니다.

반환 값

지정된 각도의 코사인을 반환합니다.

설명

각도가 도 단위이면 각도에 $\text{PI}()/180$ 를 곱하거나, RADIANS 함수를 사용하여 도를 라디안으로 변환합니다.

예제

수식	설명	결과
<code>= COS(1.047)</code>	1.047 라디안의 코사인	0.5001711
<code>= COS(60*PI()/180)</code>	60도의 코사인	0.5
<code>= COS(RADIANS(60))</code>	60도의 코사인	0.5

COSH

2021-03-16 • 2 minutes to read

숫자의 쌍곡 코사인을 반환합니다.

구문

COSH(number)

매개 변수

용어	정의
number	필수 사항입니다. 쌍곡선 코사인을 찾으려는 실수입니다.

반환 값

숫자의 쌍곡선 코사인입니다.

설명

- 쌍곡선 코사인의 수식은 다음과 같습니다.

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= COSH(4)	4의 쌍곡선 코사인입니다.	27.308233
= COSH(EXP(1))	자연 로그 밑의 쌍곡선 코사인입니다.	7.6101251

통화

2021-04-22 • 3 minutes to read

인수를 계산하고 결과를 통화 데이터 형식으로 반환합니다.

구문

```
CURRENCY(<value>)
```

매개 변수

용어	정의
값	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 다른 모든 작업 이전에 정확히 한 번 계산됩니다.

반환 값

계산되어 통화 형식 값으로 반환된 식의 값입니다.

설명

- CURRENCY 함수는 값에서 다섯 번째 유효 소수점 숫자를 반올림하여 네 번째 소수점 자리로 값을 반환합니다. 반올림은 다섯 번째 유효 소수점 숫자가 5보다 크거나 같은 경우에 수행됩니다. 예를 들어 값이 3.666666666666이면 통화 변환 시 \ \$3.6667가 반환됩니다. 하지만 값이 3.0123456789이면 통화 변환 시 \ \$3.0123가 반환됩니다.
- 식의 데이터 형식이 TrueFalse인 경우 CURRENCY(<TrueFalse>)는 True 값의 경우 \ \$1.0000을 반환하고 False 값의 경우 \ \$0.0000을 반환합니다.
- 식의 데이터 형식이 Text인 경우 CURRENCY(<Text>)는 텍스트를 숫자로 변환하려고 시도합니다. 변환이 성공하면 숫자가 통화로 변환되고, 그렇지 않으면 오류가 반환됩니다.
- 식의 데이터 형식이 DateTime인 경우 CURRENCY(<DateTime>)는 날짜/시간 값을 숫자 및 통화에 대한 해당 숫자로 변환합니다. DateTime 값은 지정된 날짜와 1900-03-01 사이의 일 수를 나타내는 정수 부분과 하루 중의 일부를 나타내는 부분(12시간 또는 정오는 0.5일)을 포함합니다. 식의 값이 올바른 DateTime 값이 아니면 오류가 반환됩니다.

예제

숫자 1234.56을 통화 데이터 형식으로 변환합니다.

```
= CURRENCY(1234.56)
```

\$1234.5600 값을 반환합니다.

DEGREES

2021-03-16 • 2 minutes to read

라디안을 각도로 변환합니다.

구문

DEGREES(*angle*)

매개 변수

용어	정의
각도	필수 사항입니다. 변환할 라디안 단위 각도입니다.

예제

수식	설명	결과
= DEGREES(PI())	pi 라디안의 각도	180

DIVIDE

2021-03-16 • 2 minutes to read

나누기를 수행하고 0으로 나누기에 대한 대체 결과 또는 BLANK()를 반환합니다.

구문

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

매개 변수

용어	정의
numerator	피제수 또는 나누어지는 숫자입니다.
denominator	제수 또는 나눌 숫자입니다.
alternateresult	(선택 사항) 0으로 나누면 오류가 발생하는 경우 반환되는 값입니다. 제공되지 않으면 기본값은 BLANK()입니다.

반환 값

10진수입니다.

설명

0으로 나누기의 대체 결과는 상수여야 합니다.

예제

다음 예제에서는 2.5를 반환합니다.

```
= DIVIDE(5,2)
```

예 1

다음 예제에서는 BLANK를 반환합니다.

```
= DIVIDE(5,0)
```

예제 2

다음 예제에서는 1을 반환합니다.

```
= DIVIDE(5,0,1)
```

참고 항목

QUOTIENT 함수
수학 및 삼각 함수

EVEN

2021-03-16 • 2 minutes to read

가장 가까운 짝수로 반올림된 수를 반환합니다. 2로 들어오는 항목을 처리하는 데 이 함수를 사용할 수 있습니다. 예를 들어 포장 상자는 한 개 또는 두 개의 항목에 대한 행을 허용합니다. 상자는 가장 가까운 2로 반올림된 항목의 수가 상자 용량과 일치하는 경우 가득 찬 것입니다.

구문

EVEN(number)

매개 변수

용어	정의
number	반올림할 값입니다.

반환 값

가장 가까운 짝수로 반올림된 수를 반환합니다.

설명

- number가 숫자가 아닌 경우 EVEN은 #VALUE! 오류 값을 반환합니다.
- number의 부호와 상관없이 값이 0에서 멀어지는 쪽으로 반올림됩니다. number가 짝수이면 반올림이 발생하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= EVEN(1.5)	1.5를 가장 가까운 짝수로 반올림합니다.	2
= EVEN(3)	3을 가장 가까운 짝수로 반올림합니다.	4
= EVEN(2)	2를 가장 가까운 짝수로 반올림합니다.	2
= EVEN(-1)	-1을 가장 가까운 짝수로 반올림합니다.	-2

EXP

2021-03-16 • 2 minutes to read

지정된 숫자를 지수로 한 e의 거듭제곱을 반환합니다. 상수 e는 자연 로그의 밑수인 2.71828182845904와 같습니다.

구문

EXP(<number>)

매개 변수

용어	정의
number	밑수 e에 적용되는 지수입니다. 상수 e는 자연 로그의 밑수인 2.71828182845904와 같습니다.

반환 값

10진수입니다.

예외

설명

- EXP는 지정된 숫자의 자연 로그인 LN의 역함수입니다.
- e가 아닌 밑수의 제곱을 계산하려면 지수화 연산자(^)를 사용합니다. 자세한 내용은 [DAX 연산자 참조](#)를 참조하세요.

예제

다음 수식은 [Power] 열에 포함된 숫자의 e 거듭제곱을 계산합니다.

= EXP([Power])

참고 항목

[수학 및 삼각 함수](#)

[LN 함수](#)

[EXP 함수](#)

[LOG 함수](#)

[LOG 함수](#)

FACT

2021-03-16 • 2 minutes to read

1*2*3*...* 계열과 같이 지정된 숫자로 끝나는 숫자의 계승값을 반환합니다.

구문

```
FACT(<number>)
```

매개 변수

용어	정의
number	계승을 계산하려는 음수가 아닌 숫자입니다.

반환 값

10진수입니다.

설명

- 숫자가 정수가 아닌 경우 잘리고 오류가 반환됩니다. 결과가 너무 크면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식은 열의 정수 계열 [Values]에 대한 계승을 반환합니다.

```
= FACT([Values])
```

다음 표에는 예상 결과가 나와 있습니다.

값	결과
0	1
1	1
2	2
3	6
4	24
5	120

값	결과
170	7.257415615308E+306

참고 항목

[수학 및 삼각 함수](#)

[TRUNC 함수](#)

FLOOR

2021-03-16 • 2 minutes to read

숫자를 0에 가까워지도록 가장 가까운 significance 배수로 내림합니다.

구문

```
FLOOR(<number>, <significance>)
```

매개 변수

용어	정의
number	반올림하려는 숫자 값입니다.
significance	반올림할 때 기준이 되는 배수입니다. 인수 number 와 significance 는 둘 다 양수이거나 둘 다 음수여야 합니다.

반환 값

10진수입니다.

설명

- 두 인수 중 하나가 숫자가 아닌 경우 FLOOR는 #VALUE! 오류 값을 반환합니다.
- number와 significance의 부호가 다르면 FLOOR는 #NUM! 오류 값을 반환합니다.
- number의 부호와 상관없이 값은 0에서 먼 방향으로 조정될 때 내림됩니다. number가 significance의 정 확한 배수이면 반올림이 발생하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습 니다.

예제

다음 수식에서는 InternetSales 테이블의 [Total Product Cost] 열에 있는 값을 사용하고 1의 가장 가까운 배수로 내림합니다.

```
= FLOOR(InternetSales[Total Product Cost],.5)
```

다음 표에서는 일부 샘플 값의 예상 결과를 보여 줍니다.

값	예상 결과
10.8423	10.8
8.0373	8

값	예상 결과
2.9733	2.9

참고 항목

[수학 및 삼각 함수](#)

GCD

2021-03-16 • 2 minutes to read

둘 이상의 정수의 최대 공약수를 반환합니다. 최대 공약수는 나머지 없이 number1과 number2를 모두 나누는 가장 큰 정수입니다.

구문

```
GCD(number1, [number2], ...)
```

매개 변수

용어	정의
number1, number2, ...	number1은 필수이며 후속 숫자는 선택 사항입니다. 1~255 값입니다. 값이 정수가 아닌 경우 잘립니다.

반환 값

둘 이상의 정수의 최대 공약수입니다.

설명

- 인수가 숫자가 아닌 경우 GCD는 #VALUE! 오류 값을 반환합니다.
- 인수가 0보다 작은 경우 GCD는 #NUM! 오류 값을 반환합니다.
- 하나는 모든 값을 균등하게 나눕니다.
- 소수에는 자기 자신과 짝수 제수가 있습니다.
- GCD에 대한 매개 변수가 $\geq 2^{53}$ 이면 GCD는 #NUM! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= GCD(5, 2)	5와 2의 최대 공약수입니다.	1
= GCD(24, 36)	24와 36의 최대 공약수입니다.	12
= GCD(7, 1)	7과 1의 최대 공약수입니다.	1

INT

2021-03-16 • 2 minutes to read

숫자를 가장 가까운 정수로 내림합니다.

구문

```
INT(<number>)
```

매개 변수

용어	정의
number	정수로 내림하려는 수입니다.

반환 값

정수입니다.

설명

TRUNC와 INT는 둘 다 정수를 반환한다는 점에서 유사합니다. TRUNC는 숫자의 소수 부분을 제거합니다. INT는 숫자의 소수 부분 값을 기준으로 숫자를 가장 가까운 정수로 내림합니다. INT 및 TRUNC는 음수를 사용하는 경우에만 다릅니다. `TRUNC(-4.3)`에서는 -4를 반환하지만 `INT(-4.3)`에서는 -5가 더 작은 수이므로 -5를 반환합니다.

예제

다음 식은 값을 1로 내림합니다. ROUND 함수를 사용하는 경우 결과는 2가 됩니다.

```
= INT(1.5)
```

참고 항목

[수학 및 삼각 함수](#)

[ROUND 함수](#)

[ROUNDUP 함수](#)

[ROUNDDOWN 함수](#)

[MROUND 함수](#)

ISO.CEILING

2021-03-16 • 4 minutes to read

숫자를 가장 가까운 정수 또는 가장 가까운 significance 배수로 반올림합니다.

구문

```
ISO.CEILING(<number>[, <significance>])
```

매개 변수

용어	정의
숫자	반올림할 숫자이거나 숫자가 포함된 열에 대한 참조입니다.
significance	(선택 사항) 반올림할 때 기준이 되는 significance의 배수입니다. 예를 들어 가장 가까운 정수로 반올림하려면 1을 입력합니다. significance 단위를 지정하지 않으면 숫자는 가장 가까운 정수로 반올림됩니다.

반환 값

지정된 대로 반올림된, *number* 인수와 동일한 형식의 숫자입니다.

설명

DAX에는 다음과 같은 차이가 있는 두 개의 CEILING 함수가 있습니다.

- CEILING 함수는 Excel의 CEILING 함수 동작을 에뮬레이트합니다.
- ISO.CEILING 함수는 최대값을 확인하기 위해 ISO 표준 동작을 따릅니다.

두 함수는 양수에 대해 동일한 값을 반환하지만, 음수에 대해서는 서로 다른 값을 반환합니다. significance의 양의 배수를 사용할 경우 CEILING과 ISO.CEILING은 모두 음수를 양의 무한대 방향으로 올립니다. significance의 음의 배수를 사용할 경우 CEILING은 음수를 음의 무한대 방향으로 내림하는 반면, ISO.CEILING은 음수를 양의 무한대 방향으로 올립니다.

결과 형식은 일반적으로 인수로 사용된 significance와 동일한 형식입니다. 단, 다음과 같은 경우는 예외입니다.

- 첫 번째 인수가 통화 형식이면 결과는 통화 형식이 됩니다.
- 선택적 인수를 포함하지 않으면 결과는 정수 형식이 됩니다.
- significance 인수가 부울 형식이면 결과는 정수 형식이 됩니다.
- significance 인수가 숫자 형식이 아니면 결과는 실수 형식이 됩니다.

예제: 양수

다음 수식은 4.45를 반환합니다. 이 수식은 가격에 더 작은 단위를 사용하지 않으려는 경우에 유용할 수 있습니다. 기존 제품 가격이 \$4.42이면 ISO.CEILING을 사용하여 가격을 가장 가까운 5센트 단위로 반올림할 수 있습니다.

= ISO.CEILING(4.42,0.05)

예제: 음수

다음 수식은 ISO 최대값인 -4.40을 반환합니다.

= ISO.CEILING(-4.42,0.05)

참고 항목

[수학 및 삼각 함수](#)

[FLOOR 함수](#)

[CEILING 함수](#)

[ROUNDUP 함수](#)

LCM

2021-03-16 • 2 minutes to read

정수의 최소 공배수를 반환합니다. 최소 공배수는 number1, number2 등 모든 정수 인수의 배수 중 가장 작은 양의 정수입니다. 분모가 다른 분수를 추가하려면 LCM을 사용합니다.

구문

```
LCM(number1, [number2], ...)
```

매개 변수

용어	정의
number1, number2,...	number1은 필수이며 후속 숫자는 선택 사항입니다. 최소 공배수를 구할 1~255 사이의 값입니다. 값이 정수가 아닌 경우 잘립니다.

반환 값

정수의 최소 공배수를 반환합니다.

설명

- 인수가 숫자가 아닌 경우 LCM은 #VALUE! 오류 값을 반환합니다.
- 인수가 0보다 작은 경우 LCM은 #NUM! 오류 값을 반환합니다.
- $\text{LCM}(a,b) \geq 2^{53}$ 이면 LCM은 #NUM! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= LCM(5, 2)	5와 2의 최소 공배수입니다.	10
= LCM(24, 36)	24와 36의 최소 공배수입니다.	72

LN

2021-03-16 • 2 minutes to read

숫자의 자연 로그를 반환합니다. 자연 로그는 상수 $e(2.71828182845904)$ 를 밑수로 합니다.

구문

```
LN(<number>)
```

매개 변수

용어	정의
number	자연 로그를 구하려는 양수입니다.

반환 값

10진수입니다.

설명

LN은 EXP 함수의 역수입니다.

예제

다음 예제에서는 `[Values]` 열에 있는 숫자의 자연 로그를 반환합니다.

```
= LN([Values])
```

참고 항목

[수학 및 삼각 함수](#)

[EXP 함수](#)

LOG

2021-03-16 • 2 minutes to read

지정한 밑에 대한 로그를 반환합니다.

구문

$\text{LOG}(\langle \text{number} \rangle, \langle \text{base} \rangle)$

매개 변수

용어	정의
number	로그를 구하려는 양수입니다.
base	로그의 밑입니다. 생략하면 밑은 10이 됩니다.

반환 값

10진수입니다.

설명

값이 너무 커서 표시할 수 없는 경우 오류가 발생할 수 있습니다.

LOG10 함수는 비슷하지만 항상 밑이 10인 상용 로그를 반환합니다.

예제

다음 수식은 동일한 결과 2를 반환합니다.

```
= LOG(100,10)
= LOG(100)
= LOG10(100)
```

참고 항목

[수학 및 삼각 함수](#)

[EXP 함수](#)

[LOG 함수](#)

[LOG 함수](#)

LOG10

2021-03-16 • 2 minutes to read

밑이 10인 로그를 반환합니다.

구문

```
LOG10(<number>)
```

매개 변수

용어	정의
number	밑이 10인 로그를 구하려는 양수입니다.

반환 값

10진수입니다.

설명

LOG 함수를 사용하여 밑수 10을 사용하는 대신 로그의 밑수를 변경할 수 있습니다.

예제

다음 수식은 동일한 결과 2를 반환합니다.

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

참고 항목

[수학 및 삼각 함수](#)

[EXP 함수](#)

[LOG 함수](#)

[LOG 함수](#)

MOD

2021-03-16 • 2 minutes to read

숫자를 제수로 나눈 나머지를 반환합니다. 결과의 부호는 항상 제수와 같습니다.

구문

```
MOD(<number>, <divisor>)
```

매개 변수

용어	정의
숫자	나눈 나머지를 구하려는 숫자입니다.
divisor	나누려는 숫자입니다.

반환 값

정수입니다.

설명

- 제수가 0이면 MOD는 오류를 반환합니다. 0으로 나눌 수는 없습니다.
- MOD 함수를 다음과 같이 INT 함수로 표현할 수 있습니다. $MOD(n, d) = n - d * INT(n/d)$

예 1

다음 수식은 3을 2로 나눈 나머지인 1을 반환합니다.

```
= MOD(3,2)
```

예제 2

다음 수식은 3을 2로 나눈 나머지인 -1을 반환합니다. 부호는 항상 제수의 부호와 같습니다.

```
= MOD(-3, -2)
```

참고 항목

[수학 및 삼각 함수](#)

[ROUND 함수](#)

[ROUNDUP 함수](#)

[ROUNDDOWN 함수](#)

[MROUND 함수](#)

[INT 함수](#)

MROUND

2021-03-16 • 2 minutes to read

원하는 배수로 반올림된 수를 반환합니다.

구문

```
MROUND(<number>, <multiple>)
```

매개 변수

용어	정의
숫자	반올림할 숫자입니다.
multiple	숫자를 반올림할 때 기준이 되는 significance의 배수입니다.

반환 값

10진수입니다.

설명

number 를 지정된 **multiple** 로 나눈 나머지가 **multiple** 값의 절반보다 크거나 같은 경우 MROUND는 반올림되며 영(0)에서 멀어집니다.

예제: 소수 자릿수

다음 식에서는 1.3을 가장 가까운 2의 배수로 반올림합니다. 예상 결과는 1.4입니다.

```
= MROUND(1.3,0.2)
```

예제: 음수

다음 식에서는 -10을 가장 가까운 -3의 배수로 반올림합니다. 예상 결과는 -9입니다.

```
= MROUND(-10,-3)
```

예제: Error

숫자의 부호가 다르므로 다음 식은 오류를 반환합니다.

```
= MROUND(5,-2)
```

참고 항목

수학 및 삼각 함수

ROUND 함수

ROUNDUP 함수

ROUNDDOWN 함수

MROUND 함수

INT 함수

ODD

2021-03-16 • 2 minutes to read

가장 가까운 홀수로 반올림된 수를 반환합니다.

구문

ODD(number)

매개 변수

용어	정의
number	필수 사항입니다. 반올림할 값입니다.

반환 값

가장 가까운 홀수로 반올림된 수를 반환합니다.

설명

- number가 숫자가 아닌 경우 ODD는 #VALUE! 오류 값을 반환합니다.
- number의 부호와 상관없이 값이 0에서 멀어지는 쪽으로 반올림됩니다. number가 홀수이면 반올림되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= ODD(1.5)	1.5를 가장 가까운 홀수로 반올림합니다.	3
= ODD(3)	3을 가장 가까운 홀수로 반올림합니다.	3
= ODD(2)	2를 가장 가까운 홀수로 반올림합니다.	3
= ODD(-1)	-1을 가장 가까운 홀수로 반올림합니다.	-1
= ODD(-2)	-2를(0에서 멀어지는 쪽) 가장 가까운 홀수로 반올림합니다.	-3

PI

2021-03-16 • 2 minutes to read

원주율(파이:3.14159265358979) 값을 15자리까지 반환합니다.

구문

```
PI()
```

반환 값

원주율(파이:3.14159265358979) 값을 포함하는 10진수이며, 15자리까지 정확하게 표시됩니다.

설명

Pi는 수학적 상수입니다. DAX에서 Pi는 Excel과 마찬가지로 15자리까지 정확하게 실수로 표시됩니다.

예제

다음 수식은 `[Radius]` 열에서 반지름이 지정된 원의 면적을 계산합니다.

```
= PI()*([Radius]^2)
```

참고 항목

[수학 및 삼각 함수](#)

POWER

2021-03-16 • 2 minutes to read

숫자를 거듭제곱한 결과를 반환합니다.

구문

```
POWER(<number>, <power>)
```

매개 변수

용어	정의
number	밑수로, 임의의 실수입니다.
power	밑수를 제공할 지수입니다.

반환 값

10진수입니다.

예제

다음 예제에서는 25를 반환합니다.

```
= POWER(5,2)
```

참고 항목

[수학 및 삼각 함수](#)

제품

2021-03-16 • 2 minutes to read

열에 있는 숫자의 곱을 반환합니다.

구문

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

매개 변수

용어	정의
column	제품을 계산할 숫자가 포함된 열입니다.

반환 값

10진수입니다.

설명

- 테이블의 각 행에 대해 계산되는 식의 곱을 반환하려면 [PRODUCTX 함수](#)를 사용합니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다. 예를 들면 다음과 같습니다.

```
PRODUCT( Table[Column] )은 PRODUCTX( Table, Table[Column] )와 동등합니다.
```

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 Annuity 테이블에서 AdjustedRates 열의 곱을 계산합니다.

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

참고 항목

[PRODUCTX](#)

PRODUCTX

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 계산된 식의 합계를 반환합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- 열에 있는 숫자의 곱을 반환하려면 **PRODUCT**를 사용합니다.
- PRODUCTX 함수는 첫 번째 인수로 테이블 또는 테이블을 반환하는 식을 사용합니다. 두 번째 인수는 곱을 계산하려는 숫자가 포함된 열이거나 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 투자의 미래 가치를 계산합니다.

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

참고 항목

[PRODUCT](#)

QUOTIENT

2021-03-16 • 2 minutes to read

나누기를 수행하고 나누기 결과의 정수 부분만 반환합니다. 나누기의 나머지를 버리려는 경우 이 함수를 사용합니다.

구문

```
QUOTIENT(<numerator>, <denominator>)
```

매개 변수

용어	정의
numerator	피제수 또는 나누어지는 수입니다.
denominator	제수 또는 나누는 수입니다.

반환 값

정수입니다.

설명

- 두 인수 중 하나가 숫자가 아닌 경우 QUOTIENT는 #VALUE 오류 값을 반환합니다.
- 두 인수에 대해 리터럴 값 대신 열 참조를 사용할 수 있습니다. 그러나 참조하는 열에 0이 포함되어 있으면 전체 값 열에 대해 오류가 반환됩니다.

예제

다음 수식은 동일한 결과 2를 반환합니다.

```
= QUOTIENT(5,2)
```

```
= QUOTIENT(10/2,2)
```

참고 항목

[수학 및 삼각 함수](#)

RADIANS

2021-03-16 • 2 minutes to read

도를 라디안으로 변환합니다.

구문

RADIANS(*angle*)

매개 변수

용어	정의
각도	필수 사항입니다. 변환할 도 단위의 각도입니다.

예제

수식	설명	결과
= RADIANS(270)	270도에 대한 라디안 값(4.712389 또는 $3\pi/2$ 라디안)	4.712389

RAND

2021-03-16 • 3 minutes to read

0보다 크거나 같고 1보다 작으며 고르게 분산된 난수를 반환합니다. 이 함수가 포함된 셀을 다시 계산할 때마다 반환되는 숫자가 변경됩니다.

구문

```
RAND()
```

반환 값

10진수입니다.

설명

- 다시 계산은 모델이 수동 다시 계산 모드로 설정되어 있는지 자동 다시 계산 모드로 설정되어 있는지 및 데이터를 새로 고쳤는지 여부를 비롯한 다양한 요인에 따라 달라집니다.
- RAND 및 고정 값이 없는 기타 일시적 함수는 다시 계산되지 않는 경우도 있습니다. 예를 들어 쿼리나 필터링을 실행하는 경우 일반적으로 함수가 다시 계산되지 않습니다. 그러나 전체 열을 다시 계산할 때는 함수의 결과가 다시 계산됩니다. 외부 데이터 소스에서 새로 고침, 함수가 포함된 수식이 다시 계산되도록 하는 데이터 수동 편집 등이 이러한 경우에 해당합니다.
- 측정값 정의에 함수를 사용하는 경우에는 RAND가 항상 다시 계산됩니다.
- RAND 함수는 0으로 나누기와 같은 오류를 방지하기 위해 결과 값 0을 반환할 수 없습니다.

예

다른 두 숫자 사이에 실수 난수를 생성하려면 다음을 사용합니다.

```
= RAND()*(b-a)+a
```

0보다 크고 1보다 작은 난수를 생성하려면 다음을 사용합니다.

```
= RAND()
```

0보다 크고 100보다 작은 난수를 생성하려면 다음을 사용합니다.

```
= RAND()*100
```

0보다 크고 100보다 작은 정수 난수를 생성하려면 다음을 사용합니다.

INT(RAND()*100)

참고 항목

[수학 및 삼각 함수](#)

[통계 함수](#)

RANDBETWEEN

2021-03-16 • 2 minutes to read

지정한 두 숫자 사이의 범위에서 난수를 반환합니다.

구문

```
RANDBETWEEN(<bottom>,<top>)
```

매개 변수

용어	정의
아래쪽	함수가 반환할 가장 작은 정수입니다.
상위	함수가 반환할 가장 큰 정수입니다.

반환 값

정수입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식은 1에서 10 사이의 난수를 반환합니다.

```
= RANDBETWEEN(1,10)
```

참고 항목

[수학 및 삼각 함수](#)

[통계 함수](#)

ROUND

2021-03-16 • 2 minutes to read

숫자를 지정된 자릿수로 반올림합니다.

구문

```
ROUND(<number>, <num_digits>)
```

매개 변수

용어	정의
number	반올림하려는 수입니다.
num_digits	반올림하려는 자릿수입니다. 음수 값인 경우 소수점 왼쪽으로 반올림하고 0인 경우 가장 가까운 정수로 반올림합니다.

반환 값

10진수입니다.

설명

- **num_digits** 가 0보다 크면 숫자를 지정된 소수 자릿수로 반올림합니다.
- **num_digits** 가 0이면 가장 가까운 정수로 반올림합니다.
- **num_digits** 가 0보다 작으면 소수점 왼쪽으로 반올림합니다.
- 관련 함수
 - 0에서 멀어지도록 항상 반올림하려면 ROUNDUP 함수를 사용합니다.
 - 0에 가까워지도록 항상 내림하려면 ROUNDDOWN 함수를 사용합니다.
 - 지정한 배수로 숫자를 반올림하려면(예: 0.5의 가장 가까운 배수로 반올림하려는 경우) MROUND 함수를 사용합니다.
 - 숫자의 정수 부분을 가져오려면 TRUNC 및 INT 함수를 사용합니다.

예 1

다음 수식은 2.15를 한 자리의 소수 자릿수로 반올림합니다. 예상 결과는 2.2입니다.

```
= ROUND(2.15,1)
```

예제 2

다음 수식은 21.5를 소수점 왼쪽 한자리로 반올림합니다. 예상 결과는 20입니다.

= ROUND(21.5, -1)

참고 항목

수학 및 삼각 함수

ROUND

ROUNDDOWN

MROUND

INT

TRUNC

ROUNDDOWN

2021-03-16 • 2 minutes to read

0에 가까워지도록 숫자를 내림합니다.

구문

```
ROUNDDOWN(<number>, <num_digits>)
```

매개 변수

용어	정의
number	내림하려는 실수입니다.
num_digits	반올림하려는 자릿수입니다. 음수인 경우 소수점 왼쪽으로 내림하고 0인 경우 가장 가까운 정수로 내림합니다.

반환 값

10진수입니다.

설명

- **num_digits** 가 0(영)보다 크면 **number** 의 값을 지정된 소수 자리까지 내림합니다.
- **num_digits** 가 0이면 **number** 의 값을 가장 가까운 정수로 내림합니다.
- **num_digits** 가 0보다 작으면 **number** 의 값을 소수점 왼쪽으로 내림합니다.
- ROUNDDOWN은 항상 숫자를 내림한다는 점을 제외하고는 ROUND와 비슷하게 동작합니다. INT 함수도 내림하지만 INT를 사용하면 결과는 항상 정수이며 ROUNDDOWN을 사용하면 결과의 정밀도를 제어할 수 있습니다.

예 1

다음 예제에서는 3.14159를 세 번째 소수 자리까지 내림합니다. 예상 결과는 3.141입니다.

```
= ROUNDDOWN(3.14159, 3)
```

예제 2

다음 예제에서는 31415.92654 값을 소수점 왼쪽으로 두 번째 소수 자리까지 내림합니다. 예상 결과는 31400입니다.

```
= ROUNDDOWN(31415.92654, -2)
```

참고 항목

수학 및 삼각 함수

ROUND

ROUNDUP

ROUNDDOWN

MROUND

INT

ROUNDUP

2021-03-16 • 3 minutes to read

숫자를 0(영)에서 멀어지는 쪽으로 반올림합니다.

구문

```
ROUNDUP(<number>, <num_digits>)
```

매개 변수

용어	정의
숫자	반올림하려는 실수입니다.
num_digits	반올림하려는 자릿수입니다. num_digits 값이 음수이면 소수점 왼쪽으로 반올림되고 num_digits 가 0이거나 생략되면 number 가 가장 가까운 정수로 반올림됩니다.

반환 값

10진수입니다.

설명

- **num_digits** 가 0(영)보다 크면 숫자를 지정된 소수 자리까지 반올림합니다.
- **num_digits** 가 0이면 숫자를 가장 가까운 정수로 반올림합니다.
- **num_digits** 가 0보다 작으면 숫자를 소수점 왼쪽으로 반올림합니다.
- ROUNDUP은 항상 숫자를 반올림한다는 점을 제외하고는 ROUND와 비슷하게 동작합니다.

예제

다음 수식은 Pi를 네 번째 소수 자리까지 반올림합니다. 예상 결과는 3.1416입니다.

```
= ROUNDUP(PI(),4)
```

예제: 두 번째 인수인 10진수

다음 수식은 1.3을 가장 가까운 0.2의 배수로 반올림합니다. 예상 결과는 2입니다.

```
= ROUNDUP(1.3,0.2)
```

예제: 두 번째 인수인 음수

다음 수식은 **FreightCost** 열의 값을 반올림하며, 예상 결과는 다음 테이블에 나와 있습니다.

= ROUNDUP([Values],-1)

num_digits 가 0보다 작은 경우 소수점 왼쪽 자릿수가 사용자가 지정한 값만큼 증가합니다.

FREIGHTCOST	예상 결과
13.25	20
2.45	10
25.56	30
1.34	10
345.01	350

참고 항목

[수학 및 삼각 함수](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

SIGN

2021-03-16 • 2 minutes to read

수의 부호, 계산 결과 또는 열의 값을 결정합니다. 이 함수는 숫자가 양수이면 1을, 0이면 0을, 음수이면 -1을 반환합니다.

구문

`SIGN(<number>)`

매개 변수

용어	정의
number	실수, 숫자가 포함된 열 또는 숫자로 계산되는 식입니다.

반환 값

정수입니다. 가능한 반환 값은 1, 0 및 -1입니다.

반환 값	설명
1	숫자가 양수입니다.
0	숫자가 0입니다.
-1	숫자가 음수입니다.

예제

다음 수식은 판매 가격에서 비용을 뺀 값을 계산하는 식 결과의 부호를 반환합니다.

`= SIGN(([Sale Price] - [Cost]))`

참고 항목

[수학 및 삼각 함수](#)

SQRT

2021-03-16 • 2 minutes to read

숫자의 제곱근을 반환합니다.

구문

```
SQRT(<number>)
```

매개 변수

용어	정의
number	제곱근을 계산할 숫자, 숫자가 포함된 열 또는 숫자로 계산되는 식입니다.

반환 값

10진수입니다.

설명

숫자가 음수이면 SQRT 함수는 오류를 반환합니다.

예제

수식은 다음과 같습니다.

```
= SQRT(25)
```

참고 항목

[수학 및 삼각 함수](#)

합계

2021-03-16 • 2 minutes to read

열에 있는 모든 수를 더합니다.

구문

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

매개 변수

용어	정의
열	합계를 구할 수가 포함되어 있는 열입니다.

반환 값

10진수입니다.

설명

합계를 구할 값을 필터링하려는 경우 SUMX 함수를 사용하여 합계를 구할 식을 지정할 수 있습니다.

예제

다음 예제에서는 Sales 테이블에서 Amt 열에 포함된 모든 숫자를 더합니다.

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

참조

[SUMX](#)

SUMX

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 평가된 식의 합계를 반환합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- SUMX 함수는 첫 번째 인수로 테이블 또는 테이블을 반환하는 식을 사용합니다. 두 번째 인수는 합계를 계산하려는 숫자가 포함된 열이거나 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 수식에 사용되는 SUMX의 더 복잡한 예를 보려면 [ALL](#) 및 [CALCULATETABLE](#)을 참조하세요.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 먼저 `ShippingTerritoryID = 5` 식에서 InternetSales 테이블을 필터링한 다음 Freight 열에 있는 모든 값의 합계를 반환합니다. 즉, 식은 지정된 판매 지역에 대해서만 운임 요금 합계를 반환합니다.

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

열을 필터링하지 않아도 되는 경우 SUM 함수를 사용합니다. SUM 함수는 열을 참조로 사용한다는 점을 제외하고 동일한 이름의 Excel 함수와 비슷합니다.

참고 항목

[SUM](#)

[통계 함수](#)

TRUNC

2021-03-16 • 2 minutes to read

숫자의 일부 또는 소수점 이하를 잘라 정수로 맞춥니다.

구문

```
TRUNC(<number>,<num_digits>)
```

매개 변수

용어	정의
number	자르려는 숫자입니다.
num_digits	잘라낼 자릿수를 지정하는 숫자입니다. 생략하면 0이 됩니다.

반환 값

정수입니다.

설명

TRUNC와 INT는 둘 다 정수를 반환한다는 점에서 유사합니다. TRUNC는 숫자의 소수 부분을 제거합니다. INT는 숫자의 소수 부분 값을 기준으로 숫자를 가장 가까운 정수로 내림합니다. INT와 TRUNC는 음수를 사용하는 경우에만 차이가 있습니다. `TRUNC(-4.3)`은 -4를 반환하지만 `INT(-4.3)`은 -5가 더 작은 숫자이므로 -5를 반환합니다.

예 1

다음 수식은 pi의 정수 부분인 3을 반환합니다.

```
= TRUNC(PI())
```

예제 2

다음 수식은 -8.9의 정수 부분인 -8을 반환합니다.

```
= TRUNC(-8.9)
```

참고 항목

[수학 및 삼각 함수](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

MROUND
INT

기타 함수

2021-03-16 • 2 minutes to read

이 함수는 어떤 범주로도 정의할 수 없는 고유한 작업을 수행합니다.

이 범주의 함수

함수	DESCRIPTION
BLANK	공백을 반환합니다.
ERROR	오류 메시지와 함께 오류가 발생합니다.

공백

2021-03-16 • 3 minutes to read

공백을 반환합니다.

구문

```
BLANK()
```

반환 값

공백입니다.

설명

- 공백은 Null과 동일하지 않습니다. DAX는 데이터베이스 Null 및 Excel의 공백 셀 둘 다에 공백을 사용합니다.
- 일부 DAX 함수는 Microsoft Excel과 약간 다르게 공백 셀을 처리합니다. 공백 및 빈 문자열("")이 항상 동일한 것은 아니지만 일부 연산에서는 동일하게 처리할 수 있습니다.

예제

다음 예제는 수식에서 공백으로 작업할 수 있는 방법을 보여 줍니다. 이 수식에서는 재판매인 및 인터넷 채널 간 판매 비율을 계산합니다. 하지만 비율을 계산하기 전에 0 값이 있는지 분모를 확인해야 합니다. 분모가 0이면 공백 값을 반환해야 하고 0이 아니면 비율을 계산합니다.

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,  
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_USD]) )
```

테이블에서는 이 수식을 사용하여 피벗 테이블을 만드는 경우의 예상 결과를 보여 줍니다.

행 레이블	액세서리	자전거	의류	총합계
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
총합계	0.83	2.51	5.45	2.94

원래 데이터 원본에서 BLANK 함수를 통해 평가되는 열에는 텍스트, 빈 문자열 또는 Null이 포함될 수 있습니다. 원래 데이터 원본이 SQL Server 데이터베이스인 경우 Null 및 빈 문자열은 서로 다른 종류의 데이터입니다. 하지만 이 연산에서는 암시적 형식의 캐스팅이 수행되며 DAX가 데이터를 동일하게 처리합니다.

참고 항목

[텍스트 함수](#)

[ISBLANK 함수](#)

오류

2021-03-16 • 2 minutes to read

오류 메시지와 함께 오류가 발생합니다.

구문

```
ERROR(<text>)
```

매개 변수

용어	정의
text	오류 메시지를 포함하는 텍스트 문자열입니다.

반환 값

없음

설명

- ERROR 함수는 스칼라 값이 필요한 모든 곳에서 DAX식에 배치할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제 1

다음 DAX 쿼리를 실행한다고 가정합니다.

```
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]
```

"빨간색이 발견되었습니다"라는 오류 메시지가 표시되고 실패합니다.

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

```

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]

```

다음 테이블을 반환합니다.

DIMPRODUCT[COLOR]	[측정값]
검정	검정
파랑	파랑
회색	회색
다중	다중
해당 없음	해당 없음
빨강	빨강
은	은
Silver\Black	Silver\Black
흰색	흰색
노란색	노란색

Magenta는 제품 색 중 하나가 아니기 때문에 ERROR 함수는 실행되지 않습니다.

부모 및 자식 함수

2021-03-16 • 2 minutes to read

이 함수는 부모/자식 계층 구조로 표시되는 데이터를 관리합니다. 자세한 내용은 [DAX의 부모-자식 계층 구조 함수 이해](#)를 참조하세요.

이 범주의 함수

함수	설명
PATH	현재 식별자의 모든 부모 식별자가 포함된, 구분된 텍스트 문자열을 반환합니다.
PATHCONTAINS	지정된 <i>item</i> 이 지정된 <i>path</i> 에 있으면 TRUE를 반환합니다.
PATHITEM	PATH 함수의 계산에서 나온 문자열의 지정된 <i>위치</i> 에 있는 항목을 반환합니다.
PATHITEMREVERSE	PATH 함수의 계산에서 나온 문자열의 지정된 <i>위치</i> 에 있는 항목을 반환합니다.
PATHLENGTH	항목 자체를 포함하여, 지정된 PATH 결과에서 지정된 항목의 부모 수를 반환합니다.

DAX의 부모-자식 계층 구조 함수 이해

2021-03-16 • 8 minutes to read

DAX는 사용자가 모델에서 부모-자식 계층 구조로 표시된 데이터를 관리할 수 있도록 5가지 함수를 제공합니다. 이 함수를 통해 사용자는 행에 포함된 부모의 전체 계보, 최상위 부모까지 계보의 수준 수, 현재 행 상위 수준의 부모, 현재 행 계층 구조의 최상위에서 n 하위 항목, 특정 부모가 현재 행 계층 구조의 부모인지 여부를 확인할 수 있습니다.

DAX의 부모-자식 함수

다음 표에서 **EmployeeKey** 및 **ParentEmployeeKey** 열에 포함된 부모-자식 계층 구조가 모든 함수 예제에서 사용됩니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

위의 표에서 직원 112는 부모가 정의되어 있지 않고, 직원 14의 관리자(ParentEmployeeKey)는 직원 112이고, 직원 3의 관리자는 직원 14이며, 직원 11, 13, 162의 관리자는 직원 3임을 확인할 수 있습니다. 위 정보는 직원 112는 상위 관리자가 없고 표에 나와 있는 모든 직원의 최상위 관리자이며, 직원 3은 직원 14에게 보고하고 직원 11, 13, 162는 직원 3에게 보고한다는 것을 파악하는 데 도움이 됩니다.

다음 표에는 사용 가능한 함수, 함수에 대한 간략한 설명, 위와 동일한 데이터를 사용한 함수 예제가 나와 있습니다.

PATH 함수 - 가장 오래되었거나 가장 위에 있는 항목부터 시작하여 현재 항목까지 현재 행의 모든 부모 식별자가 포함된 구분된 텍스트를 반환합니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로
112		112
14	112	112 14

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로
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

PATHLENGTH 함수 - 현재 수준에서 시작하여 가장 오래되었거나 가장 위에 있는 부모 수준까지 지정된 PATH()의 수준 수를 반환합니다. 다음 예제에서 PathLength 열은 ' = PATHLENGTH([Path]) '로 정의됩니다. 이 함수의 작동 방식을 이해하기 쉽도록, 예제에는 Path() 예제의 모든 데이터가 포함되어 있습니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	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
117	162	112 14 3 162 117	5
221	162	112 14 3 162 221	5
81	162	112 14 3 162 81	5

PATHITEM 함수 - 왼쪽부터 오른쪽으로 개수를 계산한 결과처럼 PATH()에서 지정된 위치에 있는 항목을 반환합니다. 다음 예제에서 왼쪽부터 네 번째 열인 PathItem 열은 ' = PATHITEM([Path], 4) '으로 정의됩니다. 이 예제는 Path() 예제와 동일한 샘플 데이터를 사용하여 Path 문자열에서 왼쪽부터 네 번째 위치에 있는 EmployeeKey를 반환합니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	PATHITEM - 왼쪽부터 4번째
112		112	
14	112	112 14	
3	14	112 14 3	

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	PATHITEM - 왼쪽부터 4번째
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

PATHITEMREVERSE 함수 - 오른쪽부터 왼쪽으로 개수를 역방향으로 계산한 결과처럼 PATH()에서 *position*에 있는 항목을 반환합니다.

다음 예제에서 오른쪽부터 세 번째 열인 PathItemReverse 열은 '`= PATHITEMREVERSE([Path], 3)`'로 정의됩니다. 이 예제는 Path() 예제와 동일한 샘플 데이터를 사용하여 Path 문자열에서 오른쪽부터 세 번째 위치에 있는 EmployeeKey를 반환합니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	PATHITEMREVERSE - 오른쪽부터 세 번째
112		112	
14	112	112 14	
3	14	112 14 3	112
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

PATHCONTAINS 함수 - 지정된 *item*이 지정된 *path*에 있으면 **TRUE**를 반환합니다. 다음 예제에서 PathContains - employee 162 열은 '`= PATHCONTAINS([Path], "162")`'로 정의됩니다. 이 예제는 지정된 경로에 직원 162가 포함되어 있으면 **TRUE**를 반환합니다. 이 예제는 위 Path() 예제의 결과를 사용합니다.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	PATHCONTAINS - EMPLOYEE 162
112		112	FALSE
14	112	112 14	FALSE
3	14	112 14 3	FALSE

EMPLOYEEKEY	PARENTEMPLOYEEKEY	경로	PATHCONTAINS - EMPLOYEE 162
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

PATH

2021-03-16 • 6 minutes to read

가장 오래된 값부터 현재 값까지 계속되는, 현재 식별자의 모든 부모 식별자가 포함된, 구분된 텍스트 문자열을 반환합니다.

구문

```
PATH(<ID_columnName>, <parent_columnName>)
```

매개 변수

용어	정의
ID_columnName	테이블에 있는 행의 고유 식별자가 포함된 기존 열의 이름입니다. 이 매개 변수는 식이 될 수 없습니다. <i>ID_columnName</i> 의 값에 대한 데이터 형식은 텍스트 또는 정수여야 하며 <i>parent_columnName</i> 에서 참조된 열과 데이터 형식이 동일해야 합니다.
parent_columnName	현재 행 부모의 고유 식별자가 포함된 기존 열의 이름입니다. 이 매개 변수는 식이 될 수 없습니다. <i>parent_columnName</i> 데이터 형식의 값에 대한 데이터 형식은 텍스트 또는 정수여야 하며 <i>ID_columnName</i> 의 값과 데이터 형식이 동일해야 합니다.

반환 값

현재 식별자의 모든 부모 식별자가 포함된, 구분된 텍스트 문자열입니다.

설명

- 이 함수를 일종의 내부 계층 구조를 가진 테이블에서 사용하면 현재 행 값과 관련된 항목을 반환합니다. 예를 들어 직원, 직원의 관리자, 관리자의 관리자가 포함된 Employees 테이블에서 직원과 직원의 관리자를 연결하는 경로를 반환할 수 있습니다.
- 이러한 경로는 부모-자식 관계의 단일 수준으로 제한되지 않습니다. 지정된 시작 행으로부터 여러 수준 위에 있는 관련 행을 반환할 수 있습니다.
 - 상위 항목을 구분하는 데 사용되는 구분 기호는 세로 막대인 '|'입니다.
 - ID_columnName* 및 *parent_columnName*의 값은 데이터 형식이 텍스트 또는 정수로 동일해야 합니다.
 - parent_columnName*의 값은 *ID_columnName*에도 있어야 합니다. 즉, 자식 수준에 값이 없으면 부모를 조회할 수 없습니다.
 - parent_columnName*이 BLANK이면 PATH()는 *ID_columnName* 값을 반환합니다. 즉, 직원의 관리자를 찾지만 *parent_columnName* 열에 데이터가 없으면 PATH 함수는 직원 ID만 반환합니다.
 - ID_columnName*에 중복 항목이 있고 *parent_columnName*이 해당 중복 항목과 동일하면 PATH()는 공통 *parent_columnName* 값을 반환합니다. 하지만 *parent_columnName* 값이 해당 중복 항목과 다르면 PATH()는 오류를 반환합니다. 즉, 동일한 직원 ID에 대해 두 개의 목록이 있고 관리자 ID가 동일하면 PATH 함수는 해당 관리자의 ID를 반환합니다. 하지만 동일한 직원 ID가 두 개이고 해당 관리자 ID가 서

로 다르면 PATH 함수는 오류를 반환합니다.

- *ID_columnName* 이 BLANK이면 PATH()는 BLANK를 반환합니다.
- *ID_columnName*에 세로 막대 '|'가 포함되어 있으면 PATH()는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 각 직원의 모든 관리자가 나열되는 계산 열을 만듭니다.

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

PATHCONTAINS

2021-03-16 • 2 minutes to read

지정된 *item* 이 지정된 *path* 에 있으면 **TRUE** 를 반환합니다.

구문

```
PATHCONTAINS(<path>, <item>)
```

매개 변수

용어	정의
path	PATH 함수 평가 결과 만들어지는 문자열입니다.
항목	경로 결과에서 찾을 텍스트 식입니다.

반환 값

path 에 *item* 이 있으면 값이 **TRUE** 이고 없으면 **FALSE** 입니다.

설명

- *item* 이 정수이면 텍스트로 변환된 후 함수가 평가됩니다. 변환이 실패하면 함수에서 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 관리자 ID를 사용하여 직원 집합을 확인하는 계산 열을 만듭니다. PATH 함수에서 반환하는 관리자 목록에 관리자 ID가 있으면 PATHCONTAINS 함수에서 true를 반환합니다. 없으면 false를 반환합니다.

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), "23")
```


PATHITEM

2021-03-16 • 3 minutes to read

PATH 함수의 계산에서 나온 문자열의 지정된 *위치*에 있는 항목을 반환합니다. 위치는 왼쪽에서 오른쪽으로 계산됩니다.

구문

```
PATHITEM(<path>, <position>[, <type>])
```

매개 변수

용어	정의
path	PATH 함수 결과의 형식으로 된 텍스트 문자열입니다.
position	반환될 항목의 위치가 포함된 정수 식입니다.
형식	(선택 항목) 결과의 데이터 형식을 정의하는 열거형입니다.

형식 열거형

열거형	교대 열거형	설명
TEXT	0	결과가 데이터 형식 텍스트와 함께 반환됩니다. 않습니다(기본값).
정수	1	결과가 정수로 반환됩니다.

반환 값

식별자 목록의 지정된 위치에 있는 PATH 함수에 의해 반환되는 식별자입니다. PATH 함수에서 반환되는 항목은 현재에 대해 가장 먼 순서로 정렬됩니다.

설명

- PATH 함수에 의해 반환되는 계층 구조에서 특정 수준을 반환하려면 이 함수를 사용할 수 있습니다. 예를 들어 모든 직원의 스킵(skip) 수준 관리자만 반환할 수 있습니다.
- 1보다 작거나 *path*의 요소 수보다 큰 수를 *position*에 지정하면 PATHITEM 함수는 BLANK를 반환합니다.
- type*이 올바른 열거형 요소가 아닌 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 현재 직원의 세 번째 계층 관리자를 반환합니다. 현재 직원에 대한 부모의 계층 구조와 함께 문자열을 반환하는 PATH 함수에 대한 입력으로 employee 및 manager ID를 사용합니다. 이 문자열로부터 PATHITEM은 세 번째 항목을 정수로 반환합니다.

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

PATHITEMREVERSE

2021-03-16 • 4 minutes to read

PATH 함수의 계산에서 나온 문자열의 지정된 *위치*에 있는 항목을 반환합니다. 위치는 오른쪽에서 왼쪽까지 역방향으로 계산됩니다.

구문

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

매개 변수

용어	정의
path	PATH 함수 평가로 얻은 텍스트 문자열입니다.
position	반환될 항목의 위치가 포함된 정수 식입니다. 위치는 오른쪽에서 왼쪽까지 역방향으로 계산됩니다.
형식	(선택 항목) 결과의 데이터 형식을 정의하는 열거형입니다.

형식 열거형

열거형	교대 열거형	설명
TEXT	0	결과가 데이터 형식 텍스트와 함께 반환됩니다. 않습니다(기본값).
정수	1	결과가 정수로 반환됩니다.

반환 값

현재 항목부터 가장 오래된 항목까지 계산하여 지정된 경로의 n 위치에 있는 상위 항목을 반환합니다.

설명

- 이 함수는 PATH 함수의 결과로 얻은 계층 구조에서 개별 항목을 가져오는 데 사용할 수 있습니다.
- 이 함수는 가장 가까운 항목이 처음에 나열되도록 계층 구조의 표준 순서를 역순으로 만듭니다. 예를 들어 PATH 함수가 계층 구조에서 한 직원 위에 있는 관리자의 목록을 반환하는 경우 PATHITEMREVERSE 함수는 위치 2에 이 직원의 직속 관리자를 반환합니다. 위치 1에는 이 직원의 ID가 있기 때문입니다.
- position*에 지정된 숫자가 1(일)보다 작거나 *path*의 요소 수보다 큰 경우 PATHITEM 함수에서 BLANK를 반환합니다.
- type*이 올바른 열거형 요소가 아닌 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 PATH 함수에 대한 입력으로 직원 ID 열을 사용하고 반환된 최상위 요소의 목록을 역순으로 만듭니다. 지정된 위치는 3이고 반환 형식은 1입니다. 따라서 PATHITEMREVERSE 함수는 해당 직원으로부터 두 수준 위에 있는 관리자를 나타내는 정수를 반환합니다.

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

PATHLENGTH

2021-03-16 • 2 minutes to read

항목 자체를 포함하여, 지정된 PATH 결과에서 지정된 항목의 부모 수를 반환합니다.

구문

```
PATHLENGTH(<path>)
```

매개 변수

용어	정의
path	PATH 함수의 계산 결과인 텍스트 식입니다.

반환 값

지정된 항목을 포함하여, 지정된 PATH 결과에서 지정된 항목의 부모 항목 수입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제는 직원 ID를 PATH 함수의 입력으로 사용하고 계층 구조에서 해당 직원 위에 있는 관리자 목록을 반환합니다. PATHLENGTH 함수는 이 결과를 사용하여 처음 직원을 비롯한 여러 수준의 직원 및 관리자 수를 계산합니다.

```
= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```

관계 함수

2021-03-16 • 2 minutes to read

이 범주의 함수는 테이블 간의 관계를 관리하고 활용하기 위한 것입니다.

이 범주의 함수

함수	설명
CROSSFILTER	두 열 사이에 존재하는 관계에 대한 계산에서 사용할 교차 필터링 방향을 지정합니다.
RELATED	다른 테이블에서 관련 값을 반환합니다.
RELATEDTABLE	지정된 필터에 의해 수정된 컨텍스트에서 테이블 식을 평가합니다.
USERELATIONSHIP	특정 계산에서 columnName1 및 columnName2 간에 존재하는 관계로 사용할 관계를 지정합니다.

CROSSFILTER

2021-04-22 • 8 minutes to read

두 열 사이에 존재하는 관계에 대한 계산에서 사용할 교차 필터링 방향을 지정합니다.

구문

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

매개 변수

용어	정의
columnName1	일반적으로 사용할 관계의 다 측을 나타내는 기존 열의 이름으로, 표준 DAX 구문을 사용하며 정규화되어 있습니다. 인수가 역순으로 주어진 경우에는 함수에서 해당 인수를 사용하기 전에 바꿉니다. 이 인수는 식이 될 수 없습니다.
columnName2	일반적으로 사용할 관계의 일 측 또는 조회 측을 나타내는 기존 열의 이름으로, 표준 DAX 구문을 사용하며 정규화되어 있습니다. 인수가 역순으로 주어진 경우에는 함수에서 해당 인수를 사용하기 전에 바꿉니다. 이 인수는 식이 될 수 없습니다.
방향	<p>사용할 교차 필터 방향입니다. 다음 중 하나여야 합니다.</p> <p>None - 이 관계에 따라 교차 필터링이 수행되지 않습니다.</p> <p>Both - 둘 중 한 측의 필터로 다른 측을 필터링합니다.</p> <p>OneWay - 관계의 한 측 또는 조회 측에 있는 필터로 다른 측을 필터링합니다. 이 옵션은 일 대 일 관계에서 사용할 수 없습니다. 다 대 다 관계에서는 조회 측이 명확하지 않으므로 이 옵션을 사용하지 마세요. 대신 OneWay_LeftFiltersRight 또는 OneWay_RightFiltersLeft를 사용하세요.</p> <p>OneWay_LeftFiltersRight - <columnName1> 측의 필터로 <columnName2> 측을 필터링합니다. 이 옵션은 일 대 일 또는 다 대 일 관계에서 사용할 수 없습니다.</p> <p>OneWay_RightFiltersLeft - <columnName2> 측의 필터로 <columnName1> 측을 필터링합니다. 이 옵션은 일 대 일 또는 다 대 일 관계에서 사용할 수 없습니다.</p>

반환 값

이 함수는 값을 반환하지 않으며, 쿼리 기간 동안 표시된 관계에 대해 교차 필터링 방향만 설정합니다.

설명

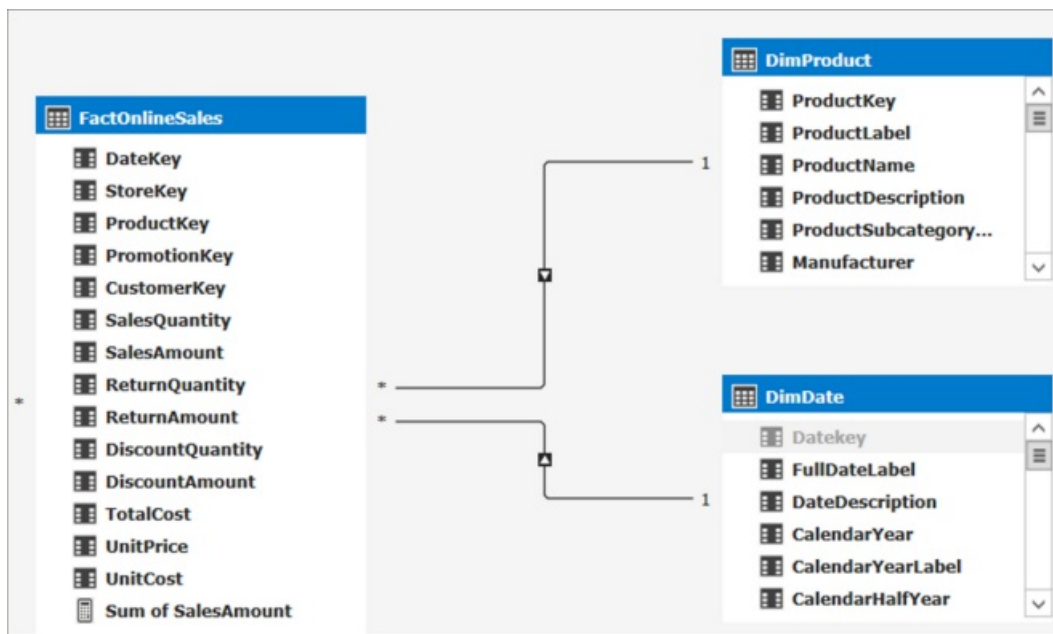
- 1:1 관계에서는 한 방향과 양방향 간에 차이가 없습니다.
- CROSSFILTER는 필터를 인수로 사용하는 함수에서만 사용할 수 있습니다. 예를 들어 CALCULATE,

CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD 및 TOTALYTD와 같은 함수가 있습니다.

- CROSSFILTER는 모델의 기존 관계를 사용하여 끝 지점 열로 관계를 식별합니다.
- CROSSFILTER에서는 관계의 교차 필터링 설정이 중요하지 않습니다. 즉, 관계가 모델의 한 방향으로 필터링하도록 설정되었는지 양방향으로 필터링하도록 설정되었는지 여부가 함수 사용에 영향을 주지 않습니다. CROSSFILTER는 기존의 모든 교차 필터링 설정을 재정의합니다.
- 인수로 명명된 열이 관계의 일부가 아니거나 인수가 다른 관계에 속해 있는 경우 오류가 반환됩니다.
- CALCULATE 식이 중첩되고 둘 이상의 CALCULATE 식에 CROSSFILTER 함수가 포함된 경우 충돌 또는 모호성 문제가 발생하면 가장 안쪽 CROSSFILTER가 우선 적용됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 모델 다이어그램에서 DimProduct와 DimDate는 모두 FactOnlineSales와 단방향 관계가 있습니다.



기본적으로 연도별로 판매된 제품 수를 가져올 수 없습니다.

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

다음과 같은 두 가지 방법으로 연도별 제품 수를 가져올 수 있습니다.

- 관계에서 양방향 교차 필터링을 설정합니다. 그러면 이러한 두 테이블 간의 모든 데이터에 대해 필터가 작동하는 방식이 변경됩니다.
- CROSSFILTER 함수를 사용하여 관계가 이 측정값에 대해서만 작동하는 방식을 변경합니다.

DAX를 사용하는 경우 CROSSFILTER 함수를 사용하여 관계에 정의된 두 열 간에 교차 필터 방향이 동작하는 방식을 변경할 수 있습니다. 이 경우 DAX 식은 다음과 같이 표시됩니다.

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

측정 식에서 CROSSFILTER 함수를 사용하여 다음 예상 결과를 얻습니다.

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	\$29,358,677.22	606	606

RELATED

2021-03-16 • 7 minutes to read

다른 테이블에서 관련 값을 반환합니다.

구문

RELATED(<column>)

매개 변수

용어	정의
열	검색하려는 값이 포함된 열입니다.

반환 값

현재 행과 관련된 단일 값입니다.

설명

- RELATED 함수를 사용하려면 현재 테이블과 관련 정보가 있는 테이블 간에 관계가 있어야 합니다. 원하는 데이터가 포함된 열을 지정하면 함수가 기존의 다 대 일 관계를 따라 관련 테이블의 지정된 열에서 값을 가져옵니다. 관계가 존재하지 않는 경우 관계를 만들어야 합니다.
- RELATED 함수가 조회를 수행하는 경우 적용되었을 수 있는 필터와 관계없이 지정된 테이블에서 모든 값을 검사합니다.
- RELATED 함수에는 행 컨텍스트가 필요하므로 현재 행 컨텍스트가 모호하지 않은 계산된 열 식에서만 사용하거나 테이블 검색 함수를 사용하는 식에서 중첩된 함수로만 사용할 수 있습니다. SUMX와 같은 테이블 검색 함수는 현재 행 값을 가져와서 다른 테이블에 해당 값이 있는지 검색합니다.
- RELATED 함수는 [제한된 관계](#)에서 열을 가져오는 데 사용할 수 없습니다.

예제

다음 예제에서는 미국의 매출이 제외된 판매 보고서를 생성하기 위해 Non USA Internet Sales라는 측정값을 만듭니다. 이 측정값을 만들려면 SalesTerritory 테이블에서 미국 관련 매출을 모두 제외하도록 InternetSales_USD 테이블을 필터링해야 합니다. United States라는 국가는 SalesTerritory 테이블에서 Northwest, Northeast, Central, Southwest 및 Southeast라는 지역에 대해 각각 한 번씩 총 다섯 번 나옵니다.

이 측정값을 만들기 위해 인터넷 매출을 필터링하는 첫 번째 방법으로 다음과 같은 필터 식을 추가할 수 있습니다.

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

그러나 이 방법은 직관적이지가 않고 입력 오류가 발생하기 쉬우며 향후 기존 지역 중 하나라도 분할될 경우 작

동하지 않을 수 있습니다.

InternetSales_USD와 SalesTerritory 간의 기존 관계를 사용하고 국가가 미국이 아니어야 함을 명시적으로 지정하는 방법이 더 낫습니다. 이렇게 하려면 다음과 같은 필터 식을 만듭니다.

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

이 식에서는 RELATED 함수를 사용하여 InternetSales_USD 테이블의 SalesTerritoryKey 키 열의 값부터 시작하여 SalesTerritory 테이블에 있는 국가 값을 조회합니다. 필터 함수는 이 조회 결과를 사용하여 InternetSales_USD 행이 필터링되었는지 여부를 확인합니다.

NOTE

예제가 작동하지 않을 경우 테이블 간에 관계를 만들어야 할 수 있습니다.

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

다음 테이블에서는 지역별 합계만 보여 줍니다. 이 테이블을 통해 Non USA Internet Sales 측정값의 필터 식이 의도한 대로 작동한다는 것을 확인할 수 있습니다.

행 레이블	INTERNET SALES	미국 이외 지역 인터넷 판매
오스트레일리아	\$4,999,021.84	\$4,999,021.84
캐나다	\$1,343,109.10	\$1,343,109.10
프랑스	\$2,490,944.57	\$2,490,944.57
독일	\$2,775,195.60	\$2,775,195.60
영국	\$5,057,076.55	\$5,057,076.55
미국	\$9,389,479.79	
총합계	\$26,054,827.45	\$16,665,347.67

다음 테이블에서는 이 측정값을 피벗 테이블에서 사용한 경우 얻을 수 있는 최종 보고서를 보여 줍니다.

행 레이블	액세서리	자전거	의류	총합계
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

행 레이블	액 세서리	자전거	의류	총 합계
총 합계	\$384,639.63	\$16,107,890.23	\$172,817.81	\$16,665,347.67

참고 항목

[RELATEDTABLE](#)

[필터 함수](#)

RELATEDTABLE

2021-03-16 • 2 minutes to read

지정된 필터로 수정된 컨텍스트에서 테이블 식을 평가합니다.

구문

```
RELATEDTABLE(<tableName>)
```

매개 변수

용어	정의
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 식이 될 수 없습니다.

반환 값

값 테이블입니다.

설명

- RELATEDTABLE 함수는 데이터가 필터링 되는 컨텍스트를 변경하고 사용자가 지정 한 새 컨텍스트에서 식을 계산합니다.
- 이 함수는 논리 식이 없는 CALCULATETABLE 함수에 대한 바로 가기입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 RELATEDTABLE 함수를 사용하여 Product Category 테이블에 있는 Internet Sales로 계산 열을 만듭니다.

다음 테이블에 결과가 표시됩니다.

제품 범주 키	제품 범주 대체 키	PRODUCT CATEGORY 이름	INTERNET SALES
1	1	Bikes	\$28,318,144.65
2	2	구성 요소	
3	3	Clothing	\$339,772.61
4	4	Accessories	\$700,759.96

```
= SUMX( RELATEDTABLE('InternetSales_USD')  
    , [SalesAmount_USD])
```

참고 항목

[CALCULATETABLE](#)

[필터 함수](#)

USERELATIONSHIP

2021-03-16 • 7 minutes to read

특정 계산에서 columnName1 및 columnName2 간에 존재하는 관계로 사용할 관계를 지정합니다.

구문

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

매개 변수

용어	정의
columnName1	일반적으로 사용할 관계의 다 측을 나타내는 기존 열의 이름으로, 표준 DAX 구문을 사용하며 정규화되어 있습니다. 인수가 역순으로 주어진 경우에는 함수에서 해당 인수를 사용하기 전에 바꿉니다. 이 인수는 식이 될 수 없습니다.
columnName2	일반적으로 사용할 관계의 일 측 또는 조회 측을 나타내는 기존 열의 이름으로, 표준 DAX 구문을 사용하며 정규화되어 있습니다. 인수가 역순으로 주어진 경우에는 함수에서 해당 인수를 사용하기 전에 바꿉니다. 이 인수는 식이 될 수 없습니다.

반환 값

이 함수는 값을 반환하지 않으며, 계산 기간 동안 지정된 관계를 사용하도록 설정만 합니다.

설명

- USERELATIONSHIP은 필터를 인수로 사용하는 함수에서만 사용할 수 있습니다. 이러한 함수로는 CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD 및 TOTALYTD 함수가 있습니다.
- 측정값이 포함된 테이블에 대해 행 수준 보안이 정의된 경우 USERELATIONSHIP을 사용할 수 없습니다. 예를 들어 DimCustomer에 대해 행 수준 보안이 정의된 경우

```
CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey],  
DimCustomer[CustomerKey]))
```

는 오류를 반환합니다.

- USERELATIONSHIP은 모델의 기존 관계를 사용하여 해당 끝 지점 열로 관계를 식별합니다.
- USERELATIONSHIP에서 관계의 상태는 중요하지 않습니다. 즉, 관계가 활성 상태인지 여부가 함수 사용에 영향을 주지 않습니다. 관계가 비활성 상태인 경우에도 관계는 사용되며 함수 인수에 언급되지 않았지만 모델에 존재할 수 있는 다른 모든 활성 관계를 재정의합니다.
- 인수로 명명된 열이 관계의 일부가 아니거나 인수가 다른 관계에 속해 있는 경우 오류가 반환됩니다.
- 계산에서 테이블 A를 테이블 B에 조인하는 데 여러 관계가 필요한 경우 각 관계를 서로 다른 USERELATIONSHIP 함수에서 지정해야 합니다.
- CALCULATE 식이 중첩되어 있고 둘 이상의 CALCULATE 식에 USERELATIONSHIP 함수가 포함된 경우 총

둘 또는 모호성 문제가 발생하면 가장 안쪽 USERELATIONSHIP이 우선 적용됩니다.

- 최대 10개의 USERELATIONSHIP 함수를 중첩할 수 있지만 식의 중첩 수준은 이보다 더 깊을 수 있습니다.
즉, 샘플 식

```
=CALCULATE(CALCULATE( CALCULATE( &lt;anyExpression>;, USERELATIONSHIP( t1[colA], t2[colB])),  
t99[colZ]=999), USERELATIONSHIP( t1[colA], t2[colA]))
```

에서 보면, 식은 3개 수준 깊이로 중첩되어 있지만 USERELATIONSHIP은 2개 수준 깊이로만 중첩되어 있습니다.

예제

다음 샘플에서는 InternetSales 및 DateTime 테이블 간에 기본 활성 관계를 재정의하는 방법을 보여 줍니다. 기본 관계는 InternetSales 테이블의 OrderDate 열과 DateTime 테이블의 Date 열 간에 존재합니다.

인터넷 판매 합계를 계산하고 기존 OrderDate 대신 ShippingDate별로 분리할 수 있도록 하려면 다음 식을 사용하여 [InternetSales by ShippingDate] 측정값을 만드세요.

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

InternetSales[ShipmentDate] 및 DateTime[Date] 간에 관계가 있어야 하며 활성 관계가 아니어야 합니다.

InternetSales[OrderDate] 및 DateTime[Date] 간에도 관계가 있어야 하며 활성 관계여야 합니다.

통계 함수

2021-03-16 • 10 minutes to read

DAX(Data Analysis Expressions)는 합계, 개수, 평균 등 집계를 만드는 데 사용되는 여러 함수를 제공합니다. 이 함수는 Microsoft Excel에서 사용되는 집계 함수와 매우 유사합니다. 이 섹션에는 DAX에 제공된 통계 및 집계 함수가 나열되어 있습니다.

이 범주의 함수

함수	설명
APPROXIMATEDISTINCTCOUNT	열에 고유한 값이 포함되어 있는 행의 <i>대략적인</i> 수를 반환합니다.
AVERAGE	한 열에 있는 모든 수의 평균(산술 평균)을 반환합니다.
AVERAGEA	한 열에 있는 값의 평균(산술 평균)을 반환합니다.
AVERAGEX	한 테이블에 대해 평가되는 식 집합의 평균(산술 평균)을 계산합니다.
BETA.DIST	베타 분포를 반환합니다.
BETA.INV	베타 누적 확률 밀도 함수(BETA.DIST)의 역수를 반환합니다.
CHISQ.DIST	카이 제곱 분포를 반환합니다.
CHISQ.DISTRT	카이 제곱 분포의 우측 확률의 역수를 반환합니다.
CHISQ.INV	카이 제곱 분포 좌측 확률의 역을 반환합니다.
CHISQ.INV.RT	카이 제곱 분포 우측 확률의 역을 반환합니다.
CONFIDENCE.NORM	신뢰 구간은 값 범위입니다.
CONFIDENCE.T	학생 t 분포를 사용하여 모집단 평균의 신뢰 구간을 반환합니다.
COT	라디안으로 지정된 각도의 코탄젠트를 반환합니다.
COTH	쌍곡각의 쌍곡 코탄젠트를 반환합니다.
COUNT	열에서 숫자가 포함된 셀 수를 계산합니다.
COUNTA	열에서 비어 있지 않은 셀 수를 계산합니다.
COUNTAX	테이블에 있는 식의 결과를 평가할 때 공백이 아닌 결과를 계산합니다.

함수	설명
COUNTBLANK	한 열에서 공백 셀의 수를 계산합니다.
COUNTROWS	지정된 테이블 또는 식으로 정의된 테이블의 행 수를 계산합니다.
COUNTX	테이블에 대해 식을 평가할 때 숫자 또는 숫자로 평가되는 식이 포함된 행 수를 계산합니다.
DATATABLE	인라인 데이터 값 집합을 선언하는 메커니즘을 제공합니다.
DISTINCTCOUNT	한 열의 고유 값 수를 계산합니다.
DISTINCTCOUNTNOBLANK	한 열의 고유 값 수를 계산합니다.
EXPON.DIST	지수 분포를 반환합니다.
GEOMEAN	한 열에 있는 숫자의 기하 평균을 반환합니다.
GEOMEANX	한 테이블의 각 행에 대해 평가된 식의 기하 평균을 반환합니다.
MAX	한 열에서 또는 두 스칼라 식 사이에서 가장 큰 숫자 값을 반환합니다.
MAXA	한 열에서 가장 큰 값을 반환합니다.
MAXX	테이블의 각 행에 대해 식을 평가하고 가장 큰 숫자 값을 반환합니다.
MEDIAN	한 열에서 숫자의 중앙값을 반환합니다.
MEDIANX	테이블의 각 행에 대해 계산되는 식의 중앙값 숫자를 반환합니다.
MIN	한 열에서 또는 두 스칼라 식 사이에서 가장 작은 숫자 값을 반환합니다.
MINA	텍스트로 표현된 논리 값과 숫자를 포함하여 한 열에서 가장 작은 값을 반환합니다.
MINX	테이블의 각 행에 대해 식을 평가하여 얻은 가장 작은 숫자 값을 반환합니다.
NORM.DIST	지정된 평균 및 표준 편차에 대한 정규 분포를 반환합니다.
NORM.INV	지정된 평균 및 표준 편차에 대한 정규 누적 분포의 역입니다.
NORM.S.DIST	표준 정규 분포(평균은 0이고 표준 편차는 1임)를 반환합니다.
NORM.S.INV	표준 정규 누적 분포의 역을 반환합니다.

함수	설명
PERCENTILE.EXC	범위에서 k번째 백분위수 값을 반환합니다. 여기서 k는 0과 1 사이(0과 1 제외)의 값입니다.
PERCENTILE.INC	범위에서 k번째 백분위수 값을 반환합니다. 여기서 k는 0과 1 사이(0과 1 포함)의 값입니다.
PERCENTILEX.EXC	테이블의 각 행에 대해 계산되는 식의 백분위수를 반환합니다.
PERCENTILEX.INC	테이블의 각 행에 대해 계산되는 식의 백분위수를 반환합니다.
POISSON.DIST	포아송 분포를 반환합니다.
RANK.EQ	숫자 목록에 있는 숫자의 순위를 반환합니다.
RANKX	<i>table</i> 인수의 각 행에 대해 숫자의 순위를 숫자 목록으로 반환합니다.
SAMPLE	지정된 테이블에서 N개 행 샘플을 반환합니다.
SIN	지정된 각도의 사인을 반환합니다.
SINH	숫자의 쌍곡선 사인을 반환합니다.
STDEV.P	전체 모집단의 표준 편차를 반환합니다.
STDEV.S	샘플 모집단의 표준 편차를 반환합니다.
STDEVX.P	전체 모집단의 표준 편차를 반환합니다.
STDEVX.S	샘플 모집단의 표준 편차를 반환합니다.
SQRTPI	(number * pi)의 제곱근을 반환합니다.
T.DIST	학생의 좌측 t 분포를 반환합니다.
T.DIST.2T	양측 학생 t 분포를 반환합니다.
T.DIST.RT	우측 학생 t 분포를 반환합니다.
T.INV	학생 t 분포의 좌측 역을 반환합니다.
T.INV.2t	학생 t 분포의 양측 역을 반환합니다.
TAN	지정된 각도의 탄젠트를 반환합니다.
TANH	숫자의 쌍곡선 탄젠트를 반환합니다.
VAR.P	전체 모집단의 분산을 반환합니다.

함수	설명
VAR.S	샘플 모집단의 분산을 반환합니다.
VARX.P	전체 모집단의 분산을 반환합니다.
VARX.S	샘플 모집단의 분산을 반환합니다.

APPROXIMATEDISTINCTCOUNT

2021-03-16 • 2 minutes to read

열에 고유한 값이 포함되어 있는 행의 *대략적인* 수를 반환합니다. 이 함수는 정확한 결과와 편차가 적어 DISTINCTCOUNT보다 향상된 성능으로 많은 양의 데이터를 쿼리할 수 있습니다.

구문

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

매개 변수

용어	DESCRIPTION
열	계산할 값이 포함되어 있는 열입니다. 이 매개 변수는 식이 될 수 없습니다.

반환 값

*column*의 대략적인 고유 값 수입니다.

설명

이 함수의 유일한 인수는 열입니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있습니다. 함수에서 개수를 계산할 행을 찾지 못하면 BLANK를 반환하고 그러지 않으면 고유 값의 개수를 반환합니다.

AVERAGE

2021-03-16 • 4 minutes to read

한 열에 있는 모든 수의 평균(산술 평균)을 반환합니다.

구문

```
AVERAGE(<column>)
```

매개 변수

용어	정의
열	평균을 구하려는 수가 포함되어 있는 열입니다.

반환 값

열에 있는 숫자의 산술 평균을 나타내는 10진수를 반환합니다.

설명

- 이 함수는 지정된 열을 인수로 사용하여 해당 열에 있는 값의 평균을 찾습니다. 일련의 숫자로 평가되는 식의 평균을 찾으려면 AVERAGEX 함수를 대신 사용합니다.
- 열의 숫자가 아닌 값은 다음과 같이 처리됩니다.
 - 열에 텍스트가 포함되어 있으면 집계를 수행할 수 없으며 함수가 빈 값을 반환합니다.
 - 열에 논리 값이나 빈 셀이 포함되어 있으면 해당 값은 무시됩니다.
 - 값이 0인 셀은 포함됩니다.
- 셀 평균을 계산할 때 빈 셀과 0 값을 포함하는 셀은 다르다는 점에 유의해야 합니다. 셀에 0이 포함된 경우 숫자 합계에 추가되고 해당 행이 피제수로 사용되는 행 수에 포함됩니다. 그러나 셀 값이 비어 있으면 해당 행은 포함되지 않습니다.
- 집계할 행이 없는 경우 이 함수는 빈 값을 반환합니다. 그러나 행이 있지만 지정된 기준을 충족하는 행이 없는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없는 경우 Excel도 0을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식에서는 InternetSales 테이블에서 ExtendedSalesAmount 열의 값에 대한 평균을 반환합니다.

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

관련 함수

AVERAGEX 함수는 테이블의 각 행에 대해 계산되는 식을 해당 인수로 사용할 수 있습니다. 따라서 사용자가 계산을 수행한 다음 계산된 값의 평균을 구할 수 있습니다.

AVERAGEA 함수는 열을 인수로 사용하지만 그 외에는 같은 이름의 Excel 함수와 유사합니다. AVERAGEA 함수를 사용하면 빈 값을 포함하는 열에서 평균을 계산할 수 있습니다.

AVERAGEA

2021-03-16 • 3 minutes to read

한 열에 있는 값의 평균(산술 평균)을 반환합니다. 텍스트 및 숫자가 아닌 값을 처리합니다.

구문

AVERAGEA(<column>)

매개 변수

용어	정의
열	평균을 구하려는 값이 포함되어 있는 열입니다.

반환 값

10진수입니다.

설명

- AVERAGEA 함수는 열을 사용하고 열에 있는 숫자의 평균을 구하지만 다음 규칙에 따라 숫자가 아닌 데이터 형식도 처리합니다.
 - TRUE로 평가되는 값은 1로 계산됩니다.
 - FALSE로 평가되는 값은 0(영)으로 계산됩니다.
 - 숫자가 아닌 텍스트가 포함된 값은 0(영)으로 계산됩니다.
 - 빈 텍스트("")는 0(영)으로 계산됩니다.
- 계산의 일부로 참조에 논리값 및 숫자의 텍스트 표현을 포함하지 않으려면 AVERAGE 함수를 사용합니다.
- 집계할 행이 없는 경우 이 함수는 빈 값을 반환합니다. 그러나 행이 있지만 지정된 기준을 충족하는 행이 없는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없는 경우 Microsoft Excel도 0을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 다음 테이블의 참조되는 열에서 비어 있지 않은 셀의 평균을 반환합니다. AVERAGE 함수를 사용한 경우 평균은 21/2이 되지만 AVERAGEA 함수를 사용하면 결과는 22/5가 됩니다.

TRANSACTION ID	금액	결과
0000123	1	1로 계산
0000124	20	20으로 계산
0000125	해당 없음	0으로 계산

TRANSACTION ID	금액	결과
0000126		0으로 계산
0000126	TRUE	1로 계산

= AVERAGEA([Amount])

참고 항목

[AVERAGE 함수](#)

[AVERAGEX 함수](#)

[통계 함수](#)

AVERAGEX

2021-03-16 • 3 minutes to read

한 테이블에 대해 평가되는 식 집합의 평균(산술 평균)을 계산합니다.

구문

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

매개 변수

용어	정의
테이블	테이블 이름 또는 집계를 수행할 수 있는 테이블을 지정하는 식입니다.
식	첫 번째 인수에서 테이블의 각 행에 대해 평가되는 스칼라 결과가 포함된 식입니다.

반환 값

10진수입니다.

설명

- AVERAGEX 함수를 사용하면 테이블의 각 행에 대한 식을 평가한 후 결과로 얻은 값 집합을 사용하여 산술 평균을 계산할 수 있습니다. 따라서 이 함수는 테이블을 첫 번째 인수로 사용하고, 식을 두 번째 인수로 사용합니다.
- 그 외의 모든 면에서 AVERAGEX는 AVERAGE와 동일한 규칙을 따릅니다. 숫자가 아닌 셀 또는 Null 셀은 포함할 수 없습니다. 테이블 인수 및 식 인수 둘 다 필수입니다.
- 집계할 행이 없으면 함수는 공백을 반환합니다. 행은 있지만 지정된 조건을 충족하는 행이 없으면 함수는 0을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 먼저 각 열의 Freight와 TaxAmt 합계를 구한 후 이 합계의 평균을 계산하는 방법으로 InternetSales 테이블의 각 주문에 대한 평균 운임과 세금을 계산합니다.

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

두 번째 인수로 사용되는 식에서 여러 연산을 사용하는 경우 괄호를 사용하여 계산 순서를 제어해야 합니다. 자세한 내용은 [DAX 구문 참조](#)를 확인해보세요.

참고 항목

AVERAGE 함수
AVERAGEA 함수
통계 함수

BETA.DIST

2021-03-16 • 2 minutes to read

베타 분포를 반환합니다. 베타 분포는 일반적으로 샘플에 있는 특정 항목의 백분율 편차(예: 하루 중 TV를 시청하는 시간)를 연구하는 데 사용됩니다.

구문

```
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

매개 변수

용어	정의
x	함수를 계산할 A와 B 사이의 값입니다.
알파	분포의 매개 변수입니다.
베타	분포의 매개 변수입니다.
A	선택 사항입니다. x 간격의 하한입니다.
b	선택 사항입니다. x 간격의 상한입니다.

반환 값

베타 분포를 반환합니다.

설명

- 인수가 숫자가 아닌 경우 BETA.DIST는 #VALUE! 오류 값을 반환합니다.
- 인수가 정수가 아니면 반올림됩니다.
- $\alpha \leq 0$ 또는 $\beta \leq 0$ 인 경우 BETA.DIST는 #NUM! 오류 값을 반환합니다.
- $x < A$, $x > B$ 또는 $A = B$ 인 경우 BETA.DIST는 #NUM! 오류 값을 반환합니다.
- A와 B의 값을 생략하면 BETA.DIST는 $A = 0$ 및 $B = 1$ 인 표준 누적 베타 분포를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

BETA.INV

2021-03-16 • 2 minutes to read

베타 누적 확률 밀도 함수(BETA.DIST)의 역수를 반환합니다.

$\text{probability} = \text{BETA.DIST}(x, \dots, \text{TRUE})$ 이면 $\text{BETA.INV}(\text{probability}, \dots) = x$ 입니다. 예상 완료 시간과 가변성이 주어질 경우, 프로젝트 계획에 베타 분포를 사용하여 가능한 완료 시간을 모델링할 수 있습니다.

구문

```
BETA.INV(probability, alpha, beta, [A], [B])
```

매개 변수

용어	정의
Probability	베타 분포와 관련된 확률입니다.
알파	분포의 매개 변수입니다.
베타	분포의 매개 변수입니다.
A	선택 사항입니다. x 간격의 하한입니다.
b	선택 사항입니다. x 간격의 상한입니다.

반환 값

베타 누적 확률 밀도 함수(BETA.DIST)의 역수를 반환합니다.

설명

- 인수가 숫자가 아닌 경우 BETA.INV는 #VALUE! 오류 값을 반환합니다.
- 인수가 정수가 아니면 반올림됩니다.
- $\alpha \leq 0$ 또는 $\beta \leq 0$ 이면 BETA.INV는 #NUM! 오류 값을 반환합니다.
- $\text{probability} \leq 0$ 또는 $\text{probability} > 1$ 이면 BETA.INV는 #NUM! 오류 값을 반환합니다.
- A와 B의 값을 생략하면 BETA.INV는 $A = 0$ 과 $B = 1$ 이 되도록 표준 누적 베타 분포를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

CHISQ.DIST

2021-03-16 • 2 minutes to read

카이 제곱 분포를 반환합니다.

카이 제곱 분포는 일반적으로 샘플에 있는 특정 항목의 백분을 편차(예: 하루 중 TV를 시청하는 시간)를 연구하는 데 사용됩니다.

구문

```
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

매개 변수

용어	정의
x	분포를 평가하려는 값입니다.
Deg_freedom	자유도 수입니다.
cumulative	함수의 형식을 결정하는 논리 값입니다. cumulative가 TRUE이면 CHISQ.DIST에서 누적 분포 함수를 반환하고 FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

카이 제곱 분포입니다.

설명

- x 또는 deg_freedom이 숫자가 아닌 경우 오류가 반환됩니다.
- deg_freedom이 정수가 아니면 반올림됩니다.
- x가 0보다 작은 경우 오류가 반환됩니다.
- deg_freedom이 1보다 작거나 10^{10} 보다 큰 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

반환

[VALUE]
0.632120558828558

CHISQ.DIST.RT

2021-03-16 • 2 minutes to read

카이 제곱 분포의 우측 확률의 역수를 반환합니다.

카이 제곱 분포는 카이 제곱 검정과 연결됩니다. 카이 제곱 검정을 사용하여 관찰 값과 예상 값을 비교합니다. 예를 들어, 유전자 실험은 식물의 다음 세대에 특정 색 집합이 나타난다는 가설을 제기할 수 있습니다. 관찰 결과와 예상 결과를 비교하여 원래 가설이 유효한지 결정할 수 있습니다.

구문

```
CHISQ.DIST.RT(<x>, <deg_freedom>)
```

매개 변수

용어	정의
x	분포를 평가하려는 값입니다.
Deg_freedom	자유도 수입니다.

반환 값

카이 제곱 분포의 우측 확률입니다.

설명

- x 또는 deg_freedom이 숫자가 아닌 경우 오류가 반환됩니다.
- deg_freedom이 정수가 아니면 반올림됩니다.
- x가 0보다 작은 경우 오류가 반환됩니다.
- deg_freedom이 1보다 작거나 10^10보다 큰 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

반환

[VALUE]

0.84914503608461

CHISQ.INV

2021-03-16 • 2 minutes to read

카이 제곱 분포 좌측 확률의 역을 반환합니다.

카이 제곱 분포는 일반적으로 샘플에 있는 특정 항목의 백분을 편차(예: 하루 중 TV를 시청하는 시간)를 연구하는 데 사용됩니다.

구문

```
CHISQ.INV(probability,deg_freedom)
```

매개 변수

용어	정의
Probability	카이 제곱 분포와 관련된 확률입니다.
Deg_freedom	자유도 수입니다.

반환 값

카이 제곱 분포 좌측 확률의 역을 반환합니다.

설명

- 인수가 숫자가 아니면 CHISQ.INV는 #VALUE! 오류 값을 반환합니다.
- probability < 0 or probability > 1이면 CHISQ.INV가 #NUM! 오류 값을 반환합니다.
- deg_freedom이 정수가 아니면 반올림됩니다.
- deg_freedom < 0 or deg_freedom > 10^10이면 CHISQ.INV는 #NUM! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= CHISQ.INV(0.93,1)	1 자유도를 사용하는 0.93에 대한 카이 제곱 분포의 좌측 검정 확률의 역함수 값입니다.	5.318520074
= CHISQ.INV(0.6,2)	2 자유도를 사용하는 0.6에 대한 카이 제곱 분포의 좌측 검정 확률의 역함수 값입니다.	1.832581464

CHISQ.INV.RT

2021-03-16 • 2 minutes to read

카이 제곱 분포 우측 확률의 역을 반환합니다.

$\text{probability} = \text{CHISQ.DIST.RT}(x, \dots)$ 이면 $\text{CHISQ.INV.RT}(\text{probability}, \dots) = x$ 입니다. 관찰된 결과와 예상되는 결과를 비교하여 원래 가설이 유효한지를 결정하려면 이 함수를 사용합니다.

구문

`CHISQ.INV.RT(probability, deg_freedom)`

매개 변수

용어	정의
Probability	카이 제곱 분포와 관련된 확률입니다.
Deg_freedom	자유도 수입니다.

반환 값

카이 제곱 분포 우측 확률의 역을 반환합니다.

설명

- 인수가 숫자가 아닌 경우 CHISQ.INV.RT는 #VALUE! 오류 값을 반환합니다.
- $\text{probability} < 0$ 또는 $\text{probability} > 1$ 이면 CHISQ.INV.RT는 #NUM! 오류 값을 반환합니다.
- deg_freedom이 정수가 아니면 반올림됩니다.
- If $\text{deg_freedom} < 1$ 이면 CHISQ.INV.RT는 #NUM! 오류 값을 반환합니다.
- 확률의 값이 지정되면 $\text{CHISQ.INV.RT}(\text{CHISQ.DIST.RT}(x, \text{deg_freedom}) = \text{probability})$ 와 같이 x 값을 구합니다. 따라서 CHISQ.INV.RT의 정밀도는 CHISQ.DIST.RT의 정밀도에 따라 달라집니다. CHISQ.INV.RT는 반복적인 검색 기법을 사용합니다. 64회 반복 후에 검색이 수렴되지 않은 경우 이 함수는 #N/A 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

CONFIDENCE.NORM

2021-03-16 • 4 minutes to read

신뢰 구간은 값 범위입니다. 샘플 평균인 x 는 이 범위의 중심에 있고, 범위는 $x \pm \text{CONFIDENCE.NORM}$ 입니다. 예를 들어 x 가 메일을 통해 주문한 제품 배달 시간의 샘플 평균인 경우 $x \pm \text{CONFIDENCE.NORM}$ 은 모집단 평균 범위입니다. 모집단 평균 μ_0 이 이 범위에 있으면 x 보다 μ_0 에서 추가 샘플 평균을 얻을 확률이 α 보다 더 높습니다. 모집단 평균 μ_0 이 이 범위에 없으면 x 보다 μ_0 에서 추가 샘플 평균을 얻을 확률이 α 보다 더 낮습니다. 즉, x , standard_dev 및 size 를 사용하여 모집단 평균이 μ_0 인 가설의 significance 수준 α 에서 양측 테스트를 구성한다고 가정합니다. μ_0 이 신뢰 구간에 있으면 가설을 거부하지 않고, μ_0 이 신뢰 구간에 없으면 가설을 거부합니다. 신뢰 구간에서 다음 패키지의 배달 시간이 신뢰 구간에 있을 확률이 $1-\alpha$ 라고 유추할 수 없습니다.

구문

```
CONFIDENCE.NORM(alpha,standard_dev,size)
```

매개 변수

용어	정의
alpha	신뢰 수준을 계산하는 데 사용되는 significance 수준입니다. 신뢰 수준은 $100 \times (1 - \alpha)\%$ 와 같습니다. 즉, α 0.05는 95% 신뢰 수준을 나타냅니다.
standard_dev	데이터 범위의 모집단 표준 편차이며, 알려진 것으로 가정됩니다.
standard_dev,size	샘플 크기입니다.

반환 값

값 범위

설명

- 인수가 숫자가 아니면 CONFIDENCE.NORM은 #VALUE! 오류 값을 반환합니다.
- $\alpha \leq 0$ 또는 $\alpha \geq 1$ 이면 CONFIDENCE.NORM은 #NUM! 오류 값을 반환합니다.
- $\text{standard_dev} \leq 0$ 이면 CONFIDENCE.NORM은 #NUM! 오류 값을 반환합니다.
- size 가 정수가 아니면 반올림됩니다.
- $\text{size} < 1$ 이면 CONFIDENCE.NORM은 #NUM! 오류 값을 반환합니다.
- α 가 0.05와 같다고 가정하면, 표준 정규 분포 곡선에서 $(1 - \alpha)$ 또는 95%와 같은 영역을 계산해야 합니다. 이 값은 ± 1.96 입니다. 따라서 신뢰 구간은 다음과 같습니다.

$$\overline{x} \pm 1.96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

CONFIDENCE.T

2021-03-16 • 2 minutes to read

학생 t 분포를 사용하여 모집단 평균의 신뢰 구간을 반환합니다.

구문

```
CONFIDENCE.T(alpha,standard_dev,size)
```

매개 변수

용어	정의
alpha	신뢰 수준을 계산하는 데 사용되는 significance 수준입니다. 신뢰 수준은 $100 \times (1 - \alpha)\%$ 와 같습니다. 즉, alpha 0.05는 95% 신뢰 수준을 나타냅니다.
standard_dev	데이터 범위의 모집단 표준 편차이며, 알려진 것으로 가정됩니다.
크기	샘플 크기입니다.

반환 값

학생 t 분포를 사용하여 모집단 평균의 신뢰 구간을 반환합니다.

설명

- 인수가 숫자가 아니면 CONFIDENCE.T는 #VALUE! 오류 값을 반환합니다.
- $\alpha \leq 0$ 또는 $\alpha \geq 1$ 이면 CONFIDENCE.T는 #NUM! 오류 값을 반환합니다.
- $\text{standard_dev} \leq 0$ 이면 CONFIDENCE.T는 #NUM! 오류 값을 반환합니다.
- size가 정수가 아니면 반올림됩니다.
- size가 1과 같으면 CONFIDENCE.T는 #DIV/0! 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= CONFIDENCE.T(0.05,1,50)	significance 수준이 5%이고 표준 편차가 1인 샘플 크기 50을 기준으로, 모집단 평균의 신뢰 구간입니다. 학생 t 분포를 기반으로 합니다.	0.284196855

COT

2021-03-16 • 2 minutes to read

라디안으로 지정된 각도의 코탄젠트를 반환합니다.

구문

```
COT (<number>)
```

매개 변수

용어	정의
숫자	코탄젠트를 반환하려는 라디안 단위 각도입니다.

반환 값

지정된 각도의 코탄젠트입니다.

설명

- 숫자의 절대값은 2^{27} 보다 작아야 하며 0일 수 없습니다.
- 숫자가 제한 범위를 초과하는 경우 오류가 반환됩니다.
- 숫자가 아닌 값인 경우 오류가 반환됩니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE { COT(30) }
```

반환

[VALUE]

-0.156119952161659

COTH

2021-03-16 • 2 minutes to read

쌍곡각의 쌍곡 코탄젠트를 반환합니다.

구문

COTH (<number>)

매개 변수

용어	정의
숫자	쌍곡 코탄젠트를 사용할 쌍곡 라디안 단위의 각도입니다.

반환 값

지정된 각도의 쌍곡 코탄젠트입니다.

설명

- 쌍곡 코탄젠트는 일반적인 (원형) 코탄젠트의 아날로그입니다.
- 숫자의 절대값은 2^{27} 보다 작아야 하며 0일 수 없습니다.
- 숫자가 제한 범위를 초과하는 경우 오류가 반환됩니다.
- 숫자가 아닌 값인 경우 오류가 반환됩니다.
- 다음 수식이 사용됩니다.

$$\text{COTH}(N) = \frac{1}{\text{TANH}(N)} = \frac{\text{COSH}(N)}{\text{SINH}(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE { COTH(2) }
```

반환

[VALUE]

1.03731472072755

개수

2021-03-16 • 2 minutes to read

COUNT 함수는 한 열에서 공백이 아닌 값이 포함된 셀의 개수를 계산합니다.

구문

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

매개 변수

용어	정의
열	계산할 값이 포함되어 있는 열입니다.

반환 값

정수입니다.

설명

- 이 함수의 인수에는 열만 허용됩니다. COUNT 함수는 다음과 같은 종류의 값이 포함된 행의 개수를 계산합니다.
 - 숫자
 - 날짜
 - 문자열
- 개수를 계산할 행을 찾을 수 없으면 함수는 공백을 반환합니다.
- 공백 값은 건너뜁니다. TRUE/FALSE 값은 지원되지 않습니다.
- TRUE/FALSE 값의 열을 계산하려면 COUNTA 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제는 ShipDate 열에 있는 값의 개수를 계산하는 방법을 보여 줍니다.

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

논리 값 또는 텍스트를 계산하려면 COUNTA 또는 COUNTAX 함수를 사용하세요.

참고 항목

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

COUNTA

2021-03-16 • 2 minutes to read

COUNTA 함수는 한 열에서 비어 있지 않은 셀의 개수를 계산합니다.

구문

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

매개 변수

용어	정의
열	개수를 계산할 값이 포함된 열입니다.

반환 값

정수입니다.

설명

- 개수를 계산할 행을 찾을 수 없으면 함수는 공백을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제는 전화번호를 저장하는 열에 임의 종류의 값이 있는 **Reseller** 테이블의 모든 행을 반환합니다. 테이블 이름에는 공백이 포함되지 않으므로 따옴표는 선택 사항입니다.

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

참고 항목

[COUNT 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

COUNTAX

2021-03-16 • 3 minutes to read

COUNTAX 함수는 테이블에 대해 식의 결과를 계산할 때 비어 있지 않은 결과를 계산합니다. 즉, COUNTA 함수와 작동 방식은 같지만 테이블의 행을 반복하고 지정된 식이 비어 있지 않은 결과를 반환하는 행 수를 계산하는 데 사용됩니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

정수입니다.

설명

- COUNTA 함수와 마찬가지로 COUNTAX 함수는 다른 식을 포함하여 모든 유형의 정보를 포함하는 셀을 계산합니다. 예를 들어 열에 빈 문자열로 계산되는 식이 포함된 경우 COUNTAX 함수는 해당 결과를 비어 있지 않은 것으로 처리합니다. 일반적으로 COUNTAX 함수는 빈 셀을 계산하지 않지만 이 경우 셀에 수식이 포함되어 있으므로 계산됩니다.
- 집계할 행을 찾지 못할 때마다 함수는 빈 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 [Status] = **Active** 에서 Reseller 테이블을 필터링하여 얻은 테이블을 사용하여 Phone 열에서 비어 있지 않은 행의 수를 계산합니다.

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

참고 항목

[COUNT 함수](#)
[COUNTA 함수](#)
[COUNTX 함수](#)
[통계 함수](#)

COUNTBLANK

2021-03-16 • 3 minutes to read

열에서 빈 셀 수를 계산합니다.

구문

```
COUNTBLANK(<column>)
```

매개 변수

용어	정의
열	계산할 빈 셀이 포함된 열입니다.

반환 값

정수입니다. 조건을 충족하는 행이 없으면 빈 값이 반환됩니다.

설명

- 이 함수의 인수에는 열만 허용됩니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있지만 빈 셀만 계산됩니다. 0(영) 값이 포함된 셀은 계산되지 않습니다. 0은 숫자 값이므로 비어 있지 않은 것으로 간주되기 때문입니다.
- 집계할 행이 없는 경우 이 함수는 빈 값을 반환합니다. 그러나 행이 있지만 지정된 기준을 충족하는 행이 없는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없는 경우 Microsoft Excel도 0을 반환합니다.
- 즉, COUNTBLANK 함수를 통해 비어 있는 셀을 찾지 못한 경우의 결과는 0이지만, 확인할 행이 없는 경우의 결과는 빈 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 Reseller 테이블에서 BankName의 값이 비어 있는 행의 수를 계산하는 방법을 보여 줍니다.

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

논리 값 또는 텍스트 개수를 계산하려면 COUNTA 또는 COUNTAX 함수를 사용합니다.

참고 항목

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

COUNTROWS

2021-03-16 • 4 minutes to read

COUNTROWS 함수는 지정된 테이블 또는 식으로 정의된 테이블의 행 수를 계산합니다.

구문

```
COUNTROWS(<table>)
```

매개 변수

용어	정의
테이블	계산할 행이 포함된 테이블의 이름 또는 테이블을 반환하는 식입니다.

반환 값

정수입니다.

설명

- 이 함수는 기본 테이블의 행 수를 계산하는 데 사용할 수 있지만 테이블을 필터링하거나 테이블에 컨텍스트를 적용한 결과로 생성되는 행 수를 계산하는 데 자주 사용됩니다.
- 집계할 행이 없는 경우 이 함수는 빈 값을 반환합니다. 그러나 행이 있지만 지정된 기준을 충족하는 행이 없는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없는 경우 Microsoft Excel도 0을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예에서는 Orders 테이블의 행 수를 계산하는 방법을 보여 줍니다. 예상 결과는 52761입니다.

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

예제 2

다음 예에서는 행 컨텍스트에서 COUNTROWS를 사용하는 방법을 보여 줍니다. 이 시나리오에는 주문 번호와 관련된 두 개의 데이터 세트가 있습니다. Reseller 테이블에는 각 재판매인에 대한 행이 하나씩 포함되어 있습니다. ResellerSales 테이블에는 각 주문에 대한 여러 행이 포함되어 있으며, 각 행에는 특정 재판매인에 대한 주문이 하나씩 있습니다. 테이블은 ResellerKey 열에 대한 관계로 연결됩니다.

수식은 ResellerKey의 값을 가져온 다음 동일한 재판매인 ID를 가진 관련 테이블의 행 수를 계산합니다. 결과는 CalculatedColumn1 열에 출력됩니다.

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

다음 표에서는 예상 결 과의 일부를 보여 줍니다.

RESELLERKEY	CALCULATEDCOLUMN1
1	73
2	70
3	394

참고 항목

[COUNT 함수](#)
[COUNTA 함수](#)
[COUNTAX 함수](#)
[COUNTX 함수](#)
[통계 함수](#)

COUNTX

2021-03-16 • 3 minutes to read

테이블에 대해 식을 계산할 때 비어 있지 않은 값으로 계산되는 식 또는 비어 있지 않은 값을 포함하는 행 수를 계산합니다.

구문

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

매개 변수

용어	정의
테이블	계산할 행이 포함되어 있는 테이블입니다.
식	개수를 계산할 값이 포함된 값 세트를 반환하는 식입니다.

반환 값

정수입니다.

설명

- COUNTX 함수는 두 개의 인수를 사용합니다. 첫 번째 인수는 항상 테이블이거나 테이블을 반환하는 식이어야 합니다. 두 번째 인수는 COUNTX에서 검색하는 열 또는 식입니다.
- COUNTX 함수는 값, 날짜 또는 문자열만 계산합니다. 함수에서 개수를 계산할 행을 찾지 못하면 빈 값을 반환합니다.
- 논리 값을 계산하려면 COUNTX 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 수식은 Product 테이블에서 정가가 포함된 모든 행의 개수를 반환합니다.

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

예제 2

다음 수식은 첫 번째 인수에 대해 필터링된 테이블을 COUNTX에 전달하는 방법을 보여 줍니다. 이 수식에서는 필터 식을 사용하여 Product 테이블에서 ProductSubCategory = "Caps" 조건을 충족하는 행만 가져온 다음 결과 테이블에서 정가가 포함된 행의 개수를 계산합니다. FILTER 식은 Products 테이블에 적용되지만 관련 테이블인 ProductSubCategory를 조회하는 값을 사용합니다.

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

참고 항목

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[통계 함수](#)

DISTINCTCOUNT

2021-03-16 • 3 minutes to read

한 열의 고유 값 수를 계산합니다.

구문

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

매개 변수

용어	DESCRIPTION
열	개수를 계산할 값이 포함된 열입니다.

반환 값

*column*에 있는 고유 값의 수입니다.

설명

- 이 함수의 인수에는 열만 허용됩니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있습니다. 함수에서 개수를 계산할 행을 찾지 못하면 BLANK를 반환하고 그러지 않으면 고유 값의 개수를 반환합니다.
- DISTINCTCOUNT 함수는 BLANK 값을 포함합니다. BLANK 값을 건너뛰려면 [DISTINCTCOUNTNOBLANK](#) 함수를 사용하세요.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 ResellerSales_USD[SalesOrderNumber] 열에서 고유한 판매 주문의 수를 계산하는 방법을 보여 줍니다.

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

측면에 달력 연도가 있고 위쪽에 제품 범주가 있는 테이블에서 위의 측정값을 사용하면 다음과 같은 결과가 반환됩니다.

행 레이블	액세서리	자전거	의류	구성 요소	-	총합계
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521

행 레이블	액세서리	자전거	의류	구성 요소	-	총합계
2008	293	724	561	601		894
					1	1
총합계	1315	3153	2410	2646	1	3797

위의 예제에서 총합계 행의 숫자가 맞지 않는 것을 알 수 있습니다. 그 이유는 동일 주문에 서로 다른 제품 범주의 품목이 포함될 수 있기 때문입니다.

참고 항목

- COUNT 함수
- COUNTA 함수
- COUNTAX 함수
- COUNTX 함수
- 통계 함수

EXPON.DIST

2021-03-16 • 2 minutes to read

지수 분포를 반환합니다. EXPON.DIST를 사용하여 은행의 자동 입출금기가 현금을 제공하는 데 걸리는 시간과 같은 이벤트 간 시간을 모델링합니다. 예를 들어 EXPON.DIST를 사용하여 프로세스가 최대 1분이 걸리는 확률을 알아볼 수 있습니다.

구문

EXPON.DIST(x,lambda,cumulative)

매개 변수

용어	정의
x	필수 사항입니다. 함수 값입니다.
람다	필수 사항입니다. 매개 변수 값입니다.
cumulative	필수 사항입니다. 제공할 지수 함수의 형식을 나타내는 논리 값입니다. cumulative가 TRUE이면 EXPON.DIST에서 누적 분포 함수를 반환하고 FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

지수 분포를 반환합니다.

설명

- x 또는 람다가 숫자가 아닌 경우 EXPON.DIST에서는 #VALUE! 오류 값을 반환합니다.
- x 또는 람다가 정수가 아닌 경우 반올림됩니다.
- $x < 0$ 이면 EXPON.DIST에서는 #NUM! 오류 값을 반환합니다.
- $\text{람다} \leq 0$ 이면 EXPON.DIST에서는 #NUM! 오류 값을 반환합니다.
- 확률 밀도 함수에 대한 수식은 다음과 같습니다.

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

- 누적 분포 함수에 대한 수식은 다음과 같습니다.

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

GEOMEAN

2021-03-16 • 2 minutes to read

열에 있는 숫자의 기하 평균을 반환합니다.

테이블의 각 행에 대해 계산되는 식의 기하 평균을 반환하려면 [GEOMEANX 함수](#)를 사용합니다.

구문

```
GEOMEAN(<column>)
```

매개 변수

용어	정의
열	기하 평균을 계산할 숫자가 포함된 열입니다.

반환 값

10진수입니다.

설명

- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- GEOMEAN(Table[Column])은 GEOMEANX(Table, Table[Column])와 동일합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 투자 테이블에서 Return 열의 기하 평균을 계산합니다.

```
= GEOMEAN( Investment[Return] )
```

참고 항목

[GEOMEANX 함수](#)

GEOMEANX

2021-03-16 • 2 minutes to read

한 테이블의 각 행에 대해 평가된 식의 기하 평균을 반환합니다.

열에 있는 숫자의 기하 평균을 반환하려면 [GEOMEAN 함수](#)를 사용합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- GEOMEANX 함수는 첫 번째 인수로 테이블 또는 테이블을 반환하는 식을 사용합니다. 두 번째 인수는 기하 평균을 계산하려는 숫자가 포함된 열이거나 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 투자 테이블에서 ReturnPct 열의 기하 평균을 계산합니다.

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

참고 항목

[GEOMEAN 함수](#)

최대

2021-03-16 • 2 minutes to read

한 열이나 두 스칼라 식 중에서 가장 큰 값을 반환합니다.

구문

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

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

매개 변수

용어	정의
열	가장 큰 값을 찾을 열입니다.
식	단일 값을 반환하는 DAX 식입니다.

반환 값

가장 큰 값입니다.

설명

- 두 식을 비교할 때 공백은 0으로 처리됩니다. 즉, Max(1,Blank())는 1을 반환하고 Max(-1, Blank())는 0을 반환합니다. 두 인수가 모두 공백이면 MAX는 공백을 반환합니다. 식 중 하나가 허용되지 않는 값을 반환하면 MAX는 오류를 반환합니다.
- TRUE/FALSE 값은 지원되지 않습니다. TRUE/FALSE 값의 열을 계산하려면 MAXA 함수를 사용합니다.

예제 1

다음 예제는 InternetSales 테이블의 ExtendedAmount 열에 있는 가장 큰 값을 반환합니다.

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

예 2

다음 예제는 두 식의 결과 중에서 가장 큰 값을 반환합니다.

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

참조

[MAXA 함수](#)

MAXX 함수
통계 함수

MAXA

2021-03-16 • 2 minutes to read

한 열에서 가장 큰 값을 반환합니다.

구문

```
MAXA(<column>)
```

매개 변수

용어	정의
열	가장 큰 값을 찾을 열입니다.

반환 값

가장 큰 값입니다.

설명

- MAXA 함수는 열을 인수로 사용하고, 다음 값 형식 중에서 가장 큰 값을 찾습니다.
 - 숫자
 - 날짜
- TRUE 및 FALSE와 같은 논리 값입니다. TRUE로 평가된 행은 1로 계산되고, FALSE로 평가된 행은 0으로 계산됩니다.
- 빈 셀은 무시됩니다. 열에 사용할 수 있는 값이 없으면 MAXA는 0을 반환합니다.
- 텍스트 값을 비교하려면 MAX 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제는 정가와 재판매인 가격 간의 차액을 계산하는 계산 열인 **ResellerMargin** 에서 가장 큰 값을 반환합니다.

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

예제 2

다음 예제는 날짜와 시간이 포함된 열에서 가장 큰 값을 반환합니다. 따라서 이 수식은 가장 최근 거래 날짜를 가져옵니다.

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

참고 항목

[MAX 함수](#)

[MAXX 함수](#)

[통계 함수](#)

MAXX

2021-03-16 • 3 minutes to read

테이블의 각 행에 대해 식을 계산하고 가장 큰 값을 반환합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

가장 큰 값입니다.

설명

- MAXX 함수의 **table** 인수는 테이블 이름이거나 테이블로 계산되는 식일 수 있습니다. 두 번째 인수는 테이블의 각 행에 대해 계산할 식을 나타냅니다.
- 계산할 값 중에서 다음 항목의 개수만 계산됩니다.
 - 숫자
 - 텍스트
 - 날짜
- 공백 값은 건너뜁니다. TRUE/FALSE 값은 지원되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 수식은 두 번째 인수로 식을 사용하여 InternetSales 테이블에 있는 각 주문의 세금 및 배송료 총액을 계산합니다. 예상 결과는 375.7184입니다.

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

예제 2

다음 수식은 먼저 FILTER 식으로 InternetSales 테이블을 필터링하여 [SalesTerritory] = 5로 정의된 특정 영업 지역의 주문 하위 집합을 반환합니다. 그런 다음, MAXX 함수는 두 번째 인수로 사용된 식을 필터링된 테이블의 각 행에 대해 계산하고 해당 주문의 세금 및 배송료 중에서 가장 높은 금액을 반환합니다. 예상 결과는 250.3724입니다.

니다.

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

참고 항목

[MAX 함수](#)

[MAXA 함수](#)

[통계 함수](#)

MEDIAN

2021-03-16 • 2 minutes to read

한 열에서 숫자의 중앙값을 반환합니다.

테이블의 각 행에 대해 계산되는 식의 중앙값을 반환하려면 [MEDIANX 함수](#)를 사용합니다.

구문

```
MEDIAN(<column>)
```

매개 변수

용어	정의
열	중앙값을 계산할 숫자가 포함된 열입니다.

반환 값

10진수입니다.

설명

- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- MEDIAN(Table[Column])은 MEDIANX(Table, Table[Column])과 동일합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식은 Customers 테이블에 있는 Age 열의 중앙값을 계산합니다.

```
= MEDIAN( Customers[Age] )
```

참고 항목

[MEDIANX 함수](#)

MEDIANX

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 계산되는 식의 중앙값 숫자를 반환합니다.

열에 있는 숫자의 중앙값을 반환하려면 [MEDIAN 함수](#)를 사용합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- MEDIANX 함수는 첫 번째 인수로 테이블이나 테이블을 반환하는 식을 사용합니다. 두 번째 인수는 중앙값을 계산하려는 숫자가 포함된 열이거나 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다.
- 논리 값과 텍스트는 무시됩니다.
- MEDIANX는 공백을 무시하지 않지만, MEDIAN은 공백을 무시합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음은 미국에 거주하는 고객의 중앙값 연령을 계산합니다.

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ), Customers[Age] )
```

참고 항목

[MEDIAN 함수](#)

최소

2021-03-16 • 2 minutes to read

한 열이나 두 스칼라 식 중에서 가장 작은 값을 반환합니다.

구문

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

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

매개 변수

용어	정의
열	가장 작은 값을 찾을 열입니다.
식	단일 값을 반환하는 DAX 식입니다.

반환 값

가장 작은 값입니다.

설명

- MIN 함수는 한 열이나 두 개의 식을 인수로 사용하고 가장 작은 값을 반환합니다. 열에서 다음 형식의 값 개수가 계산됩니다.
 - 숫자
 - 텍스트
 - 날짜
 - 공백
- 식을 비교할 때 공백은 0으로 처리됩니다. 즉, `Min(1,Blank())`은 0을 반환하고 `Min(-1, Blank())`은 -1을 반환합니다. 두 인수가 모두 공백이면 MIN은 공백을 반환합니다. 식 중 하나가 허용되지 않는 값을 반환하면 MIN은 오류를 반환합니다.
- TRUE/FALSE 값은 지원되지 않습니다. TRUE/FALSE 값의 열을 계산하려면 MINA 함수를 사용합니다.

예 1

다음 예제는 계산 열인 ResellerMargin에서 가장 작은 값을 반환합니다.

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

예제 2

다음 예제는 날짜와 시간인 TransactionDate가 포함된 열에서 가장 작은 값을 반환합니다. 따라서 이 수식은 가

장 이 큰 날짜를 반환합니다.

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

예제 3

다음 예제는 두 스칼라 식의 결과 중에서 가장 작은 값을 반환합니다.

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

참고 항목

[MINA 함수](#)

[MINX 함수](#)

[통계 함수](#)

MINA

2021-03-16 • 2 minutes to read

한 열에서 가장 작은 값을 반환합니다.

구문

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

매개 변수

용어	정의
열	최소값을 찾을 열입니다.

반환 값

가장 작은 값입니다.

설명

- MINA 함수는 숫자가 포함된 열을 인수로 사용하고, 가장 작은 값을 다음과 같이 결정합니다.
 - 열에 값이 없으면 MINA는 0을 반환합니다.
 - TRUE 및 FALSE와 같은 논리 값으로 계산되는 열의 행은 TRUE이면 1, FALSE이면 0으로 처리됩니다.
 - 빈 셀은 무시됩니다.
- 텍스트 값을 비교하려면 MIN 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 식은 InternetSales 테이블의 최소 운송비를 반환합니다.

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

예제 2

다음 식은 PostalCode 열의 최소값을 반환합니다. 열의 데이터 형식이 텍스트이기 때문에 함수는 값을 찾을 수 없으며 수식에서 0을 반환합니다.

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

참고 항목

[MIN 함수](#)

MINX

2021-03-16 • 3 minutes to read

테이블의 각 행에 대해 식을 평가하여 얻은 가장 작은 값을 반환합니다.

구문

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

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

가장 작은 값입니다.

설명

- MINX 함수는 첫 번째 인수로 테이블 또는 테이블을 반환하는 식을 사용합니다. 두 번째 인수에는 테이블의 각 행에 대해 평가되는 식이 포함됩니다.
- 공백 값은 건너뜁니다. TRUE/FALSE 값은 지원되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제에서는 InternetSales 테이블을 필터링하고 특정 영업 지역의 행만 반환합니다. 그런 다음 수식에서는 Freight 열의 최솟값을 찾습니다.

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

예제 2

다음 예제에서는 이전 예제와 동일한 필터링된 테이블을 사용하지만 필터링된 테이블의 각 행에서 열의 값을 조회만 하지 않고 Freight와 TaxAmt라는 두 열의 합계를 계산하고 해당 계산의 결과 중 가장 작은 값을 반환합니다.

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

첫 번째 예제에서는 열 이름이 정규화되지 않습니다. 두 번째 예제에서는 열 이름이 정규화됩니다.

참고 항목

[MIN 함수](#)

[MINA 함수](#)

[통계 함수](#)

NORM.DIST

2021-03-16 • 2 minutes to read

지정된 평균 및 표준 편차에 대한 정규 분포를 반환합니다.

구문

```
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

매개 변수

용어	정의
X	분포를 구하려는 값입니다.
평균	분포의 산술 평균입니다.
Standard_dev	분포의 표준 편차입니다.
Cumulative*	함수의 형식을 결정하는 논리 값입니다. Cumulative가 TRUE이면 NORM.DIST는 누적 분포 함수를 반환하고, FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

지정된 평균 및 표준 편차에 대한 정규 분포입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

반환

[VALUE]

0.908788780274132

참조

[NORM.S.DIST 함수](#)

[NORM.INV 함수](#)

[NORM.S.INV](#)

NORM.INV

2021-03-16 • 2 minutes to read

지정된 평균 및 표준 편차에 대한 정규 누적 분포의 역입니다.

구문

```
NORM.INV(Probability, Mean, Standard_dev)
```

매개 변수

용어	정의
Probability	정규 분포에 해당하는 확률입니다.
평균	분포의 산술 평균입니다.
Standard_dev	분포의 표준 편차입니다.

반환 값

지정된 평균 및 표준 편차에 대한 정규 누적 분포의 역을 반환합니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

반환

[VALUE]

42.00000200956628780274132

참고 항목

[NORM.S.INV](#)

[NORM.S.DIST](#) 함수

[NORM.DIST](#) 함수

NORM.S.DIST

2021-03-16 • 2 minutes to read

표준 정규 분포(평균은 0이고 표준 편차는 1임)를 반환합니다.

구문

NORM.S.DIST(Z, Cumulative)

매개 변수

용어	정의
Z	분포를 구하려는 값입니다.
Cumulative	Cumulative는 함수의 형식을 결정하는 논리 값입니다. Cumulative가 TRUE이면 NORM.S.DIST는 누적 분포 함수를 반환하고, FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

표준 정규 분포(평균은 0이고 표준 편차는 1임)입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

EVALUATE { NORM.S.DIST(1.333333, TRUE) }

반환

[VALUE]

0.908788725604095

참조

[NORM.INV 함수](#)

[NORM.DIST 함수](#)

[NORM.S.INV](#)

NORM.S.INV

2021-03-16 • 2 minutes to read

표준 정규 누적 분포의 역을 반환합니다. 분포의 평균은 0이고 표준 편차는 1입니다.

구문

```
NORM.S.INV(Probability)
```

매개 변수

용어	정의
Probability	정규 분포에 해당하는 확률입니다.

반환 값

표준 정규 누적 분포의 역입니다. 분포의 평균은 0이고 표준 편차는 1입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { NORM.S.INV(0.908789) }
```

반환

```
[VALUE]
```

```
1.33333467304411
```

참조

[NORM.INV](#)

[NORM.S.DIST](#) 함수

[NORM.DIST](#) 함수

PERCENTILE.EXC

2021-03-16 • 2 minutes to read

범위에서 k번째 백분위수 값을 반환합니다. 여기서 k는 0과 1 사이(0과 1 제외)의 값입니다.

테이블의 각 행에 대해 계산되는 식의 백분위수를 반환하려면 [PERCENTILEX.EXC 함수](#)를 사용합니다.

구문

```
PERCENTILE.EXC(<column>, <k>)
```

매개 변수

용어	정의
열	상대 순위를 정의하는 값이 포함된 열입니다.
k	0과 1 사이(0과 1 제외)의 백분위수 값입니다.

반환 값

범위에서 k번째 백분위수 값입니다. 여기서 k는 0과 1 사이(0과 1 제외)의 값입니다.

설명

- 열이 비어 있으면 BLANK()가 반환됩니다.
- k가 0이거나 공백이면 백분위수 순위 $1/(n+1)$ 은 가장 작은 값을 반환합니다. 0인 경우 범위를 벗어나므로 오류가 반환됩니다.
- k가 숫자가 아니거나 0과 1 사이의 범위를 벗어나면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 PERCENTILE.EXC는 보간하여 k번째 백분위수 값을 결정합니다.
- 지정된 백분위수 값이 배열의 두 값 사이에 있으면 PERCENTILE.EXC는 보간합니다. 지정된 k번째 백분위수를 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[PERCENTILEX.EXC](#)

PERCENTILE.INC

2021-03-16 • 2 minutes to read

범위에서 k번째 백분위수 값을 반환합니다. 여기서 k는 0과 1 사이(0과 1 포함)의 값입니다.

테이블의 각 행에 대해 계산되는 식의 백분위수를 반환하려면 [PERCENTILEX.INC](#)를 사용합니다.

구문

```
PERCENTILE.INC(<column>, <k>)
```

매개 변수

용어	정의
열	상대 순위를 정의하는 값이 포함된 열입니다.
k	0과 1 사이(0과 1 포함)의 백분위수 값입니다.

반환 값

범위에서 k번째 백분위수 값입니다. 여기서 k는 0과 1 사이(0과 1 포함)의 값입니다.

설명

- 열이 비어 있으면 BLANK()가 반환됩니다.
- k가 0이거나 공백이면 백분위수 순위 $1/(n+1)$ 은 가장 작은 값을 반환합니다. 0인 경우 범위를 벗어나므로 오류가 반환됩니다.
- k가 숫자가 아니거나 0과 1 사이의 범위를 벗어나면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 PERCENTILE.INC는 보간하여 k번째 백분위수 값을 결정합니다.
- 지정된 백분위수 값이 배열의 두 값 사이에 있으면 PERCENTILE.INC가 보간합니다. 지정된 k번째 백분위수를 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[PERCENTILEX.INC](#)

PERCENTILEX.EXC

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 계산되는 식의 백분위수를 반환합니다.

열에 있는 숫자의 백분위수를 반환하려면 [PERCENTILE.EXC 함수](#)를 사용합니다.

구문

```
PERCENTILEX.EXC(<table>, <expression>, k)
```

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
k	0과 1 사이(0과 1 제외)의 원하는 백분위수 값입니다.

반환 값

테이블의 각 행에 대해 계산되는 식의 백분위수입니다.

설명

- k가 0이거나 공백이면 백분위수 순위 $1/(n+1)$ 은 가장 작은 값을 반환합니다. 0인 경우 범위를 벗어나므로 오류가 반환됩니다.
- k가 숫자가 아니거나 0과 1 사이의 범위를 벗어나면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 PERCENTILEX.EXC는 보간하여 k번째 백분위수 값을 결정합니다.
- 지정된 백분위수 값이 배열의 두 값 사이에 있으면 PERCENTILEX.EXC는 보간합니다. 지정된 k번째 백분위수를 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[PERCENTILE.EXC](#)

PERCENTILEX.INC

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 계산되는 식의 백분위수를 반환합니다.

열에 있는 숫자의 백분위수를 반환하려면 [PERCENTILE.INC](#)를 사용합니다.

구문

```
PERCENTILEX.INC(<table>, <expression>;, k)
```

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
k	0과 1 사이(0과 1 포함)의 원하는 백분위수 값입니다.

반환 값

테이블의 각 행에 대해 계산되는 식의 백분위수입니다.

설명

- k가 0이거나 공백이면 $1/(n-1)$ 의 백분위수 순위는 가장 작은 값을 반환합니다. 0인 경우 범위를 벗어나므로 오류가 반환됩니다.
- k가 숫자가 아니거나 0과 1 사이의 범위를 벗어나면 오류가 반환됩니다.
- k가 $1/(n-1)$ 의 배수가 아니면 PERCENTILEX.EXC는 보간하여 k번째 백분위수의 값을 결정합니다.
- 지정된 백분위수 값이 배열의 두 값 사이에 있으면 PERCENTILEX.INC가 보간합니다. 지정된 k번째 백분위수를 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[PERCENTILE.INC](#)

PERMUT

2021-03-16 • 2 minutes to read

숫자 개체에서 선택할 수 있는 제공된 개체 수에 대한 순열 수를 반환합니다. 순열은 내부 순서가 중요한 개체 또는 이벤트의 집합이거나 하위 집합입니다. 순열은 내부 순서가 중요하지 않은 조합과는 다릅니다. 복권 스타일 확률 계산에는 이 함수를 사용합니다.

구문

PERMUT(number, number_chosen)

매개 변수

용어	정의
number	필수 사항입니다. 개체 수를 설명하는 정수입니다.
number_chosen	필수 사항입니다. 각 순열의 개체 수를 설명하는 정수입니다.

반환 값

숫자 개체에서 선택할 수 있는 제공된 개체 수에 대한 순열 수를 반환합니다.

설명

- 인수는 둘 다 정수로 잘립니다.
- number 또는 number_chosen이 숫자가 아닌 경우 PERMUT는 #VALUE! 오류 값을 반환합니다.
- number ≤ 0이거나 number_chosen < 0이면 PERMUT는 #NUM! 오류 값을 반환합니다.
- number < number_chosen이면 PERMUT는 #NUM! 오류 값을 반환합니다.
- 순열 수에 대한 수식은 다음과 같습니다.

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식에서 개체 3개 중 2개가 선택된 그룹에 가능한 순열은 다음과 같습니다.

= PERMUT(3,2)

결과,

POISSON.DIST

2021-03-16 • 2 minutes to read

포아송 분포를 반환합니다. 포아송 분포의 가장 일반적인 적용 사례는 특정 시간에 걸친 이벤트 수(예: 1분 안에 도로 요금소에 도착하는 차량 수)를 예상하는 것입니다.

구문

```
POISSON.DIST(x,mean,cumulative)
```

매개 변수

용어	정의
x	필수 사항입니다. 이벤트 수입니다.
평균	필수 사항입니다. 예상 숫자 값입니다.
cumulative	필수 사항입니다. 반환된 확률 분포의 형식을 결정하는 논리 값입니다. 누적이 TRUE이면 POISSON.DIST에서 발생하는 임의 이벤트 수가 0과 x(포함) 사이인 누적 포아송 확률을 반환하고, FALSE이면 발생하는 이벤트 수가 정확히 x인 포아송 확률 질량 함수를 반환합니다.

반환 값

포아송 분포를 반환합니다.

설명

- x가 정수가 아니면 반올림됩니다.
- x 또는 mean이 숫자가 아닌 경우 POISSON.DIST는 #VALUE! 오류 값을 반환합니다.
- $x < 0$ 인 경우 POISSON.DIST는 #NUM! 오류 값을 반환합니다.
- $mean < 0$ 인 경우 POISSON.DIST는 #NUM! 오류 값을 반환합니다.
- POISSON.DIST는 다음과 같이 계산됩니다.
 - cumulative = FALSE인 경우:
$$\text{POISSON} = \frac{e^{-\lambda} \lambda^x}{x!}$$
 - cumulative = TRUE인 경우:
$$\text{CUMPOISSON} = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

RANK.EQ

2021-03-16 • 4 minutes to read

숫자 목록에 있는 숫자의 순위를 반환합니다.

구문

```
RANK.EQ(<value>, <columnName>[, <order>])
```

매개 변수

용어	정의
값	순위를 찾으려는 단일 스칼라 값을 반환하는 DAX 식입니다. 식은 함수를 계산하기 전에 정확히 한 번 계산되고, 값이 인 수 목록에 전달됩니다.
columnName	순위 결정의 기준이 되는 기존 열의 이름입니다. 이 매개 변수는 ADDCOLUMNS, ROW 또는 SUMMARIZE와 같은 함수를 사용하여 만든 식 또는 열일 수 없습니다.
순서	(선택 사항) <i>number</i> 의 순위를 매기는 방식(오름차순 또는 내림차순)을 지정하는 값입니다.

순서 값

VALUE	대체 값	설명
0(영)	FALSE	<i>columnName</i> 의 순위를 내림차순으로 매깁니다. <i>value</i> 가 <i>columnName</i> 의 가장 높은 숫자와 같으면 RANK.EQ 는 1입니다.
1	TRUE	<i>columnName</i> 의 순위를 오름차순으로 매깁니다. <i>value</i> 가 <i>columnName</i> 의 가장 낮은 숫자와 같으면 RANK.EQ 는 1입니다.

반환 값

*columnName*의 숫자 중에서 *value*의 순위를 나타내는 숫자입니다.

설명

- columnName*은 다음 함수를 사용하여 만든 열을 참조할 수 없습니다. ADDCOLUMNS, ROW 또는 SUMMARIZE.I
- value*가 *columnName*에 없거나 값이 공백이면 **RANK.EQ**는 공백 값을 반환합니다.
- value*의 중복 값에는 동일한 순위 값이 지정됩니다. 할당되는 그다음 순위 값은 순위 값에 중복 값 개수를 더한 값이 됩니다. 예를 들어 5개의 값이 순위 11에 연결되어 있으면 그다음 값의 순위는 16(11 + 5)이 됩니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제는 *InternetSales_USD* 테이블에 있는 SalesAmount_USD 열의 모든 숫자에 대해 값의 순위를 매기는 계산 열을 만듭니다.

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD], InternetSales_USD[SalesAmount_USD])
```

예제 2

다음 예제는 지정된 샘플에 대해 값 하위 집합의 순위를 매깁니다. 지역 학생과 특정 국가 테스트 성적의 테이블이 있고, 해당 국가 테스트의 전체 점수 집합도 있다고 가정합니다. 다음 계산 열은 각 지역 학생의 전국 순위를 제공합니다.

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

RANKX

2021-03-16 • 6 minutes to read

table 인수의 각 행에 대해 숫자의 순위를 숫자 목록으로 반환합니다.

구문

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

매개 변수

table

식이 계산되는 데이터의 테이블을 반환하는 DAX 식입니다.

expression

단일 스칼라 값을 반환하는 DAX 식입니다. 식은 *table*의 각 행에 대해 계산되어 순위에 대한 모든 가능한 값을 생성합니다. *expression*이 BLANK로 계산되는 경우 함수 동작을 이해하려면 설명 섹션을 참조하세요.

value

(선택 사항) 순위를 찾을 단일 스칼라 값을 반환하는 DAX 식입니다. 식에서 *value*를 찾을 수 없는 경우 함수의 동작을 이해하려면 설명 섹션을 참조하세요.

value 매개 변수를 생략하면 현재 행에 있는 식의 값이 대신 사용됩니다.

order

(선택 사항) *value*의 순위를 매기는 방식(오름차순 또는 내림차순)을 지정하는 값입니다.

값	대체 값	설명
0(영)	FALSE	식의 값 순위를 내림차순으로 매깁니다. 값이 식에서 가장 높은 숫자와 같으면 RANKX는 1을 반환합니다. order 매개 변수가 생략되는 경우 이 값이 기본값입니다.
1	TRUE	식의 순위를 오름차순으로 매깁니다. 값이 식에서 가장 낮은 숫자와 같으면 RANKX는 1을 반환합니다.

ties

(선택 사항) 동률이 있는 경우 순위를 결정하는 방법을 정의하는 열거형입니다.

열거형	설명
Skip	동률 이후 다음 순위 값은 동률의 순위 값과 동률을 이룬 값의 수를 더한 값입니다. 예를 들어 5개의 값이 순위 11과 동률을 이룬 경우 다음 값은 순위 16(11 + 5)을 받습니다. 이는 <i>ties</i> 매개 변수가 생략된 경우 기본값입니다.

열거형	설명
Dense	동률 이후 다음 순위 값은 다음 순위 값입니다. 예를 들어 5개의 값이 순위 11과 동률을 이룬 경우 다음 값은 순위 12를 받습니다.

반환 값

table 모든 행 숫자에 대해 계산된 *expression*의 모든 가능한 값 중 *value*의 순위 숫자입니다.

설명

- *expression* 또는 *value*가 BLANK로 계산되면 숫자가 되는 모든 식에 대해서는 영(0)으로 취급되고 모든 텍스트 식에 대해서는 빈 텍스트로 취급됩니다.
- *value*가 *expression*의 모든 가능한 값 중에 없으면 RANKX는 *expression*의 값에 *value*를 추가하고 *value*의 적절한 순위를 결정하기 위해 RANKX를 다시 계산합니다.
- RANKX(Inventory, [InventoryCost],,, "Dense")와 같이 인수 목록에 빈 쉼표(,)를 넣어 선택적 인수를 건너뛸 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Products 테이블의 다음 계산 열은 인터넷 채널의 각 제품에 대한 판매 순위를 계산합니다.

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```


SAMPLE

2021-03-16 • 3 minutes to read

지정된 테이블에서 N개 행의 샘플을 반환합니다.

구문

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

매개 변수

용어	정의
n_value	반환할 행 수입니다. 단일 스칼라 값을 반환하는 DAX 식이며, 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다. 정수가 아닌 값 또는 식을 입력하면 결과는 정수로 캐스팅됩니다.
테이블	'n'개 샘플 행을 추출할 데이터 테이블을 반환하는 DAX 식입니다.
orderBy_expression	(옵션) <i>table</i> 의 각 행에 대해 결과 값이 계산되는 스칼라 DAX 식입니다.
순서	(옵션) <i>orderBy_expression</i> 값을 정렬하는 방법(오름차순 또는 내림차순)을 지정하는 값입니다. 0, <i>order_by</i> 값을 내림차순으로 정렬합니다. 1, <i>order_by</i> 의 오름차순으로 순위를 매깁니다.

반환 값

*table*의 N개 행 샘플로 구성된 테이블이거나 *n_value*가 0(영) 이하인 경우 빈 테이블입니다. OrderBy 인수를 지정하면 첫 번째 행, 마지막 행과 그 사이에 고르게 분산된 행을 반환하여 샘플이 안정적이고 결정적이게 됩니다. 순서를 지정하지 않으면 샘플이 임의로 구성되어 안정적이지 않고 결정적이지 않게 됩니다.

설명

- n_value가 0(영) 이하인 경우 SAMPLE은 빈 테이블을 반환합니다.
- 샘플에서 중복 값을 방지하려면 두 번째 인수로 지정된 테이블이 정렬에 사용된 열을 기준으로 그룹화되어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

SIN

2021-03-16 • 2 minutes to read

지정된 각도의 사인을 반환합니다.

구문

`SIN(number)`

매개 변수

용어	정의
number	필수 사항입니다. 사인 값을 구할 라디안 단위 각도입니다.

반환 값

지정된 각도의 사인을 반환합니다.

설명

인수의 단위가 도이면 $\text{PI}()/180$ 을 곱하거나 `RADIANS` 함수를 사용하여 도를 라디안으로 변환합니다.

예제

수식	설명	결과
<code>= SIN(PI())</code>	pi 라디안의 사인 값(약 0).	0.0
<code>= SIN(PI()/2)</code>	pi/2 라디안의 사인 값.	1.0
<code>= SIN(30*PI()/180)</code>	30도의 사인 값.	0.5
<code>= SIN(RADIANS(30))</code>	30도의 사인 값.	0.5

SINH

2021-03-16 • 2 minutes to read

숫자의 쌍곡선 사인을 반환합니다.

구문

`SINH(number)`

매개 변수

용어	정의
number	필수 사항입니다. 실수입니다.

반환 값

숫자의 쌍곡선 사인을 반환합니다.

설명

- 쌍곡 사인의 수식은 다음과 같습니다.

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

1.03초 미만의 결과를 얻을 확률입니다.

```
= 2.868*SINH(0.0342\*1.03)
```

0.1010491을 반환합니다.

STDEV.S

2021-03-16 • 2 minutes to read

샘플 모집단의 표준 편차를 반환합니다.

구문

```
STDEV.S(<ColumnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식을 사용할 수 없습니다.

반환 값

샘플 모집단의 표준 편차를 나타내는 숫자입니다.

예외

설명

- STDEV.S는 열이 모집단의 표본을 참조하는 것으로 가정합니다. 데이터가 전체 모집단을 나타내는 경우 STDEV.P를 사용하여 표준 편차를 계산합니다.
- STDEV.S는 다음 수식을 사용합니다.
$$\sqrt{\sum (x - \tilde{x})^2 / (n-1)}$$
여기서 \tilde{x} 는 샘플 모집단에서 x의 평균값이고 n은 모집단 크기입니다.
- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 InternetSales_USD 테이블이 샘플 모집단인 경우 SalesAmount_USD 열의 표준 편차를 계산하는 측정값에 대한 수식을 보여 줍니다.

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```

STDEV.P

2021-03-16 • 2 minutes to read

전체 모집단의 표준 편차를 반환합니다.

구문

```
STDEV.P(<ColumnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식을 사용할 수 없습니다.

반환 값

전체 모집단의 표준 편차를 나타내는 숫자입니다.

설명

- STDEV.P 열이 전체 모집단을 참조하는 것으로 가정합니다. 데이터가 모집단의 표본을 나타내는 경우 STDEV.S를 사용하여 표준 편차를 계산합니다.
- STDEV.P 다음 수식을 사용합니다.
$$\sqrt{\sum(x - \bar{x})^2/n}$$
여기서 \bar{x} 는 전체 모집단에서 x의 평균값이고 n은 모집단 크기입니다.
- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에서 공백이 아닌 행이 두 개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 InternetSales_USD 테이블이 전체 모집단인 경우 SalesAmount_USD 열의 표준 편차를 계산하는 측정값에 대한 수식을 보여 줍니다.

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

STDEVX.S

2021-03-16 • 3 minutes to read

샘플 모집단의 표준 편차를 반환합니다.

구문

```
STDEVX.S(<table>, <expression>)
```

매개 변수

용어	정의
테이블	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.
식	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

샘플 모집단의 표준 편차를 나타내는 숫자입니다.

예외

설명

- STDEVX.S는 *table*의 각 행에 대해 *expression*을 평가하고 *table*이 모집단 샘플을 참조하는 것으로 가정하여 *expression*의 표준 편차를 반환합니다. *table*이 전체 모집단을 나타내는 경우 STDEVX.P를 사용하여 표준 편차를 계산합니다.
- STDEVX.S는 다음 수식을 사용합니다.
$$\sqrt{\frac{\sum (x - \tilde{x})^2}{(n-1)}}$$
여기서 \tilde{x} 는 전체 모집단에서 x 의 평균값이고 n 은 모집단 크기입니다.
- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 Product 테이블에서 수식을 사용할 때 샘플 모집단의 제품별 단가 표준 편차를 예측하는 계산 열에 대한 수식을 보여 줍니다.

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

STDEVX.P

2021-03-16 • 3 minutes to read

전체 모집단의 표준 편차를 반환합니다.

구문

```
STDEVX.P(<table>, <expression>)
```

매개 변수

용어	정의
테이블	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.
식	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

전체 모집단의 표준 편차를 나타내는 숫자입니다.

설명

- STDEVX.P는 *table*의 각 행에 대해 *expression*을 계산하고 테이블이 전체 모집단을 참조하는 것으로 가정하여 식의 표준 편차를 반환합니다. *table*의 데이터가 모집단의 샘플을 나타내는 경우 대신 STDEVX.S를 사용하여 표준 편차를 계산해야 합니다.

- STDEVX.P 다음 수식을 사용합니다.

$$\sqrt{\sum (x - \tilde{x})^2 / n}$$

여기서 \tilde{x} 는 전체 모집단에서 x 의 평균값이고 n 은 모집단 크기입니다.

- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에서 공백이 아닌 행이 두 개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 *Product* 테이블에서 수식을 사용할 때 제품당 단가의 표준 편차를 계산하는 계산 열의 수식을 보여 줍니다.

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```


SQRTPI

2021-03-16 • 2 minutes to read

(number * pi)의 제곱근을 반환합니다.

구문

```
SQRTPI(number)
```

매개 변수

용어	정의
number	필수 사항입니다. pi를 곱한 수입니다.

반환 값

(number * pi)의 제곱근을 반환합니다.

예제

수식	설명	결과
= SQRTPI(1)	pi의 제곱근입니다.	1.772454
= SQRTPI(2)	2 * pi의 제곱근입니다.	2.506628

T.DIST

2021-03-16 • 2 minutes to read

학생의 좌측 t 분포를 반환합니다.

구문

```
T.DIST(X,Deg_freedom,Cumulative)
```

매개 변수

용어	정의
X	분포를 계산할 숫자 값입니다.
Deg_freedom	자유도 수를 나타내는 정수입니다.
Cumulative	함수의 형식을 결정하는 논리 값입니다. Cumulative가 TRUE이면 T.DIST는 누적 분포 함수를 반환하고, FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

학생의 좌측 t 분포입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

그러면 다음 값이 반환됩니다.

[VALUE]

0.994695326367377

참조

[T.DIST.2T](#)

[T.DIST.RT](#)

[T.INV](#)

[T.INV.2t](#)

T.DIST.2T

2021-03-16 • 2 minutes to read

양측 학생 t 분포를 반환합니다.

구문

```
T.DIST.2T(X,Deg_freedom)
```

매개 변수

용어	정의
X	분포를 계산할 숫자 값입니다.
Deg_freedom	자유도 수를 나타내는 정수입니다.

반환 값

양측 검정 학생 t 분포입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

반환

[VALUE]

0.054644929975921

참조

[T.DIST](#)

[T.DIST.RT](#)

[T.INV](#)

[T.INV.2t](#)

T.DIST.RT

2021-03-16 • 2 minutes to read

우측 학생 t 분포를 반환합니다.

구문

```
T.DIST.RT(X,Deg_freedom)
```

매개 변수

용어	정의
X	분포를 계산할 숫자 값입니다.
Deg_freedom	자유도 수를 나타내는 정수입니다.

반환 값

우측 검정 학생 t 분포입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

반환

[VALUE]

0.0273224649879605

참조

[T.DIST](#)

[T.DIST.2T](#)

[T.INV](#)

[T.INV.2t](#)

T.INV

2021-03-16 • 2 minutes to read

학생 t 분포의 좌측 검정 역을 반환합니다.

구문

```
T.INV(Probability,Deg_freedom)
```

매개 변수

용어	정의
Probability	학생 t 분포와 관련된 확률입니다.
Deg_freedom	분포의 특징을 지정하는 데 사용되는 자유도 수입니다.

반환 값

학생 t 분포의 좌측 검정 역입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { T.INV(0.75, 2) }
```

반환

[VALUE]

0.816496580927726

참고 항목

[T.INV.2T](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

T.INV.2T

2021-03-16 • 2 minutes to read

학생 t 분포의 양측 역을 반환합니다.

구문

```
T.INV.2T(Probability,Deg_freedom)
```

매개 변수

용어	정의
Probability	학생 t 분포와 관련된 확률입니다.
Deg_freedom	분포의 특징을 지정하는 데 사용되는 자유도 수입니다.

반환 값

학생 t 분포의 양측 역수입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

반환

[VALUE]

0.606533075825759

참조

[T.INV](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

TAN

2021-03-16 • 2 minutes to read

지정된 각도의 탄젠트를 반환합니다.

구문

TAN(number)

매개 변수

용어	정의
number	필수 사항입니다. 탄젠트를 반환하려는 라디안 단위 각도입니다.

반환 값

지정된 각도의 탄젠트를 반환합니다.

설명

인수의 단위가 도이면 PI()/180을 곱하거나 RADIANS 함수를 사용하여 도를 라디안으로 변환합니다.

예제

수식	설명	결과
= TAN(0.785)	0.785라디안의 탄젠트(0.99920)	0.99920
= TAN(45*PI()/180)	45도의 탄젠트(1)	1
= TAN(RADIANS(45))	45도의 탄젠트(1)	1

TANH

2021-03-16 • 2 minutes to read

숫자의 쌍곡선 탄젠트를 반환합니다.

구문

TANH(number)

매개 변수

용어	정의
number	필수 사항입니다. 실수입니다.

반환 값

숫자의 쌍곡선 탄젠트를 반환합니다.

설명

- 쌍곡선 탄젠트에 대한 수식은 다음과 같습니다.

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

수식	설명	결과
= TANH(-2)	-2(-0.96403)의 쌍곡선 탄젠트	-0.964028
= TANH(0)	0(0)의 쌍곡선 탄젠트	0
= TANH(0.5)	0.5(0.462117)의 쌍곡선 탄젠트	0.462117

VAR.S

2021-03-16 • 2 minutes to read

샘플 모집단의 분산을 반환합니다.

구문

```
VAR.S(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식을 사용할 수 없습니다.

반환 값

표본 모집단의 분산이 포함된 숫자입니다.

설명

- VAR.S는 열이 모집단의 표본을 참조하는 것으로 가정합니다. 데이터가 모집단 전체를 나타내는 경우 VAR.P를 사용하여 분산을 계산합니다.
- VAR.S는 다음 수식을 사용합니다.

$$\sum(x - \tilde{x})^2 / (n-1)$$

여기서 \tilde{x} 는 샘플 모집단에서 x의 평균값이고

n은 모집단 크기입니다.

- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예는 표본 모집단에 대한 InternetSales_USD에서 SalesAmount_USD 열의 분산을 계산하는 측정값에 대한 수식을 보여 줍니다.

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

VAR.P

2021-03-16 • 2 minutes to read

전체 모집단의 분산을 반환합니다.

구문

```
VAR.P(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식을 사용할 수 없습니다.

반환 값

전체 모집단의 분산을 나타내는 숫자입니다.

설명

- VAR.P 열이 전체 모집단을 참조한다고 가정합니다. 데이터가 모집단 샘플을 나타내는 경우 VAR.S를 사용하여 분산을 계산합니다.
- VAR.P 다음 수식을 사용합니다.

$$\sum(x - \tilde{x})^2/n$$

여기서 \tilde{x} 는 전체 모집단에서 x의 평균값이고

n은 모집단 크기입니다.

- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에서 공백이 아닌 행이 두 개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 전체 모집단의 InternetSales_USD 테이블에서 SalesAmount_USD 열 분산을 예측하는 측정값에 대한 수식을 보여 줍니다.

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

VARX.S

2021-03-16 • 2 minutes to read

샘플 모집단의 분산을 반환합니다.

구문

```
VARX.S(<table>, <expression>)
```

매개 변수

용어	정의
테이블	데이터의 테이블을 반환하는 DAX 식입니다.
식	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

표본 모집단의 분산을 나타내는 숫자입니다.

설명

- VARX.S는 *table* 각 행의 *expression* 을 계산하고 *expression* 의 분산을 반환하며, *table* 이 모집단의 표본을 참조한다고 가정합니다. *table* 이 전체 모집단을 나타내는 경우 VARX.P를 사용하여 분산을 계산해야 합니다.

- VAR.S는 다음 수식을 사용합니다.

$$\sum(x - \tilde{x})^2 / (n-1)$$

여기서 \tilde{x} 는 샘플 모집단에서 x 의 평균값이고

n 은 모집단 크기입니다.

- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 Product 테이블에서 수식을 사용할 때 표본 모집단에 대한 제품별 단가의 분산을 예측하는 계산 열의 수식을 보여 줍니다.

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

VARX.P

2021-03-16 • 2 minutes to read

전체 모집단의 분산을 반환합니다.

구문

```
VARX.P(<table>, <expression>)
```

매개 변수

용어	정의
테이블	데이터의 테이블을 반환하는 DAX 식입니다.
식	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

전체 모집단의 분산을 나타내는 숫자입니다.

설명

- VARX.P <table>의 각 행에 대해 <expression>을 평가하고 <table>이 전체 모집단을 참조하는 것으로 가정하여 <expression>의 분산을 반환합니다. <table>이 모집단 샘플을 나타내는 경우 VARX.S를 사용하여 분산을 계산합니다.

- VARX.P 다음 수식을 사용합니다.

$$\sum (x - \tilde{x})^2 / n$$

여기서 \tilde{x} 는 전체 모집단에서 x 의 평균값이고

n 은 모집단 크기입니다.

- 빈 행은 *columnName*에서 필터링되며 계산에서 고려되지 않습니다.
- columnName*에서 공백이 아닌 행이 두 개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 Product 테이블에서 수식을 사용할 때 제품별 단가 분산을 계산하는 계산 열의 수식을 보여 줍니다.

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

테이블 조작 함수

2021-03-16 • 7 minutes to read

이 함수는 테이블을 반환하거나 기존 테이블을 조작합니다.

이 범주의 함수

함수	설명
ADDCOLUMNS	지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.
ADDMISSINGITEMS	여러 열에 있는 항목의 조합이 아직 존재하지 않는 경우 이 조합을 테이블에 추가합니다.
CROSSJOIN	인수에 있는 모든 테이블의 모든 행에 대한 카티전 곱이 포함된 테이블을 반환합니다.
CURRENTGROUP	GROUPBY 식의 테이블 인수에서 행 집합을 반환합니다.
DATATABLE	인라인 데이터 값 집합을 선언하는 메커니즘을 제공합니다.
DETAILROWS	측정값에 대해 정의된 세부 정보 행 식을 계산하고 데이터를 반환합니다.
DISTINCT 열	지정된 열에서 고유한 값을 포함하는 열이 하나인 테이블을 반환합니다.
DISTINCT 테이블	다른 테이블 또는 식에서 중복 행을 제거하여 테이블을 반환합니다.
EXCEPT	다른 테이블에 표시되지 않는 한 테이블의 행을 반환합니다.
FILTERS	<i>columnName</i> 에 필터로 직접 적용되는 값의 테이블을 반환합니다.
GENERATE	<i>table1</i> 의 각 행 사이에 카티전 곱이 있는 테이블과 <i>table1</i> 의 현재 행 컨텍스트에서 <i>table2</i> 를 평가하여 얻은 테이블을 반환합니다.
GENERATEALL	<i>table1</i> 의 각 행 사이에 카티전 곱이 있는 테이블과 <i>table1</i> 의 현재 행 컨텍스트에서 <i>table2</i> 를 평가하여 얻은 테이블을 반환합니다.
GENERATESERIES	산술 계열의 값이 포함된 단일 열 테이블을 반환합니다.
GROUPBY	SUMMARIZE 함수와 비슷하지만 GROUPBY는 추가하는 확장 열에 대해 암시적 CALCULATE를 수행하지 않습니다.
IGNORE	BLANK/NULL 계산에서 특정 식을 생략하여 SUMMARIZECOLUMNS의 동작을 수정합니다.

함수	설명
INTERSECT	중복 행을 유지하여 두 테이블의 행 교집합을 반환합니다.
NATURALINNERJOIN	다른 테이블을 사용하여 한 테이블의 내부 조인을 수행합니다.
NATURALLEFTOUTERJOIN	다른 테이블을 사용하여 한 테이블의 내부 조인을 수행합니다.
ROLLUP	groupBy_columnName 매개 변수에 의해 정의되는 열에서 결과에 롤업 행을 추가하여 SUMMARIZE의 동작을 수정합니다.
ROLLUPADDISUBTOTAL	groupBy_columnName 열을 기반으로 결과에 롤업/소계 행을 추가하여 SUMMARIZECOLUMNS의 동작을 수정합니다.
ROLLUPISSUBTOTAL	ADDMISSINGITEMS 식 내에서 ROLLUPADDISUBTOTAL에 의해 추가된 열과 롤업 그룹을 페어링합니다.
ROLLUPGROUP	groupBy_columnName 매개 변수에 의해 정의되는 열에서 결과에 롤업 행을 추가하여 SUMMARIZECOLUMNS의 동작을 수정합니다.
ROW	각 열에 지정된 식에서 얻은 값이 포함된 단일 행이 있는 테이블을 반환합니다.
SELECTCOLUMNS	지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.
SUBSTITUTEWITHINDEX	인수로 제공된 두 테이블의 왼쪽 세미조인을 나타내는 테이블을 반환합니다.
SUMMARIZE	그룹 집합에 대해 요청된 합계의 요약 테이블을 반환합니다.
SUMMARIZECOLUMNS	그룹 세트에 대한 요약 테이블을 반환합니다.
테이블 생성자	하나 이상의 열이 있는 테이블을 반환합니다.
TOPN	지정된 테이블의 상위 N개 행을 반환합니다.
TREATAS	관련이 없는 테이블 열의 필터로 테이블 식의 결과를 적용합니다.
UNION	한 쌍의 테이블에서 합집합(조인) 테이블을 만듭니다.
VALUES	지정된 테이블 또는 열의 고유 값이 포함되어 있는 단일 열 테이블을 반환합니다.

ADDCOLUMNS

2021-03-16 • 2 minutes to read

지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.

구문

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

매개 변수

용어	정의
테이블	데이터 테이블을 반환하는 모든 DAX 식입니다.
name	열에 지정된 이름으로, 큰따옴표로 묶여 있습니다.
식	<i>테이블</i> 의 각 행에 대해 평가되는 스칼라 식을 반환하는 DAX 식입니다.

반환 값

원래 열과 추가된 열이 모두 포함된 테이블입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 재판매인 채널 및 인터넷 판매의 총 판매액 값이 포함되어 있는 Product Category 테이블의 확장 버전을 반환합니다.

```
ADDCOLUMNS(ProductCategory,
    , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD]))
```

다음 테이블에서는 테이블을 수신해야 하는 함수가 수신하는 데이터에 대한 미리 보기를 보여 줍니다.

PRODUCTCATEGORY[PRODUCTCATEGORY NAME]	PRODUCTCATEGORY[PRODUCTCATEGORY ALTERNATEKEY]	PRODUCTCATEGORY[PRODUCTCATEGORY KEY]	[INTERNET SALES]	[RESELLER SALES]
Bikes	1	1	25107749.77	63084675.04
구성 요소	2	2		11205837.96
의류	3	3	306157.5829	1669943.267

PRODUCTCATEGORY[PRODUCTCATEGORY NAME]	PRODUCTCATEGORY[PRODUCTCATEGORY ALTERNATEKEY]	PRODUCTCATEGORY[PRODUCTCATEGORY KEY]	[INTERNET SALES]	[RESELLER SALES]
액세서리	4	4	640920.1338	534301.9888

ADDMISSINGITEMS

2021-03-16 • 2 minutes to read

[SUMMARIZECOLUMNS](#)에서 반환되는 테이블에 빈 값이 포함된 행을 추가합니다.

구문

```
ADDMISSINGITEMS ( [ <showAll_columnName> [ , <showAll_columnName> [ , ... ] ] ], <table> [ , <groupBy_columnName> [ , [ <filterTable> ] [ , <groupBy_columnName> [ , [ <filterTable> ] [ , ... ] ] ] ] ] )
```

매개 변수

용어	정의
showAll_columnName	(선택 사항) 사용된 측정값에 대한 데이터가 없는 항목을 반환할 열입니다. 지정하지 않으면 모든 열이 반환됩니다.
테이블	SUMMARIZECOLUMNS 테이블입니다.
groupBy_columnName	(선택 사항) 제공된 테이블 인수에서 그룹화할 열입니다.
filterTable	(선택 사항) 반환되는 행을 정의하는 테이블 식입니다.

반환 값

하나 이상의 열이 있는 테이블입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

SUMMARIZECOLUMNS 포함

[SUMMARIZECOLUMNS](#)에서 반환되는 테이블에는 값이 있는 행만 포함됩니다. ADDMISSINGITEMS 식 내에 [SUMMARIZECOLUMNS](#) 식을 래핑하면 값을 포함하지 않는 행도 반환됩니다.

예제

ADDMISSINGITEMS 없이 다음 쿼리를 수행합니다.

```
SUMMARIZECOLUMNS(  
    'Sales'[CustomerId],  
    "Total Qty", SUM ( Sales[TotalQty] )  
)
```

그러면 다음 값이 반환됩니다.

CUSTOMERID	TOTALQTY
A	5

CUSTOMERID	TOTALQTY
b	3
C	3
E	2

ADDMISSINGITEMS를 사용하여 다음 쿼리를 수행 합니다.

```
EVALUATE
ADDMISSINGITEMS (
  'Sales'[CustomerId],
  SUMMARIZECOLUMNS(
    'Sales'[CustomerId],
    "Total Qty", SUM ( Sales[TotalQty] )
  ),
  'Sales'[CustomerId]
)
```

그러면 다음 값이 반환됩니다.

CUSTOMERID	TOTALQTY
A	5
b	3
C	3
D	
E	2
F	

CROSSJOIN

2021-03-16 • 4 minutes to read

인수에 있는 모든 테이블의 모든 행에 대한 카티전 곱이 포함된 테이블을 반환합니다. 새 테이블의 열은 모든 인수 테이블의 모든 열입니다.

구문

```
CROSSJOIN(<table>, <table>[, <table>]...)
```

매개 변수

용어	정의
테이블	데이터 테이블을 반환하는 DAX 식입니다.

반환 값

인수에 있는 모든 테이블의 모든 행에 대한 카티전 곱을 포함하는 테이블입니다.

설명

- `table` 인수의 열 이름은 모든 테이블에서 서로 달라야 합니다. 아니면 오류가 반환됩니다.
- `CROSSJOIN()`에서 반환된 총 행 수는 인수에 있는 모든 테이블의 행 수와 같습니다. 또한 결과 테이블의 총 열 수는 모든 테이블에 있는 열 수의 합계입니다. 예를 들어 **TableA**에 **rA** 개 행과 **cA** 개 열이 있고 **TableB**에 **rB** 개 행과 **cB** 개 열이 있고 **TableC**에 **rC** 개 행과 **cC** 개 열이 있으면 결과 테이블에는 **rA × rb × rC** 개 행과 **cA + cB + cC** 개 열이 포함됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예에서는 두 개의 테이블 **Colors** 및 **Stationery**에 `CROSSJOIN()`을 적용한 결과를 보여 줍니다.

Colors 테이블에는 색과 패턴이 포함됩니다.

색상	패턴
빨간색	가로 줄무늬
녹색	세로 줄무늬
파랑	크로스해치

Stationery 테이블에는 글꼴과 표시 방법이 포함됩니다.

글꼴	프레젠테이션
serif	embossed
sans-serif	engraved

크로스 조인을 생성하는 식은 다음과 같습니다.

CROSSJOIN(Colors, Stationery)

테이블 식이 필요한 곳에서 위의 식을 사용하면 식의 결과는 다음과 같습니다.

색상	패턴	글꼴	프레젠테이션
빨간색	가로 줄무늬	serif	embossed
녹색	세로 줄무늬	serif	embossed
파랑	크로스해치	serif	embossed
빨간색	가로 줄무늬	sans-serif	engraved
녹색	세로 줄무늬	sans-serif	engraved
파랑	크로스해치	sans-serif	engraved

CURRENTGROUP

2021-03-16 • 2 minutes to read

GROUPBY 결과의 현재 행에 속하는 **GROUPBY** 식의 table 인수로부터 행 집합을 반환합니다.

구문

```
CURRENTGROUP ( )
```

매개 변수

없음

반환 값

groupBy_columnName 인수의 값 그룹 하나에 해당하는 **GROUPBY** 함수의 table 인수에 있는 행입니다.

설명

- 이 함수는 **GROUPBY** 식 내에서만 사용할 수 있습니다.
- 이 함수는 인수를 사용하지 않으며 다음 집계 함수 중 하나에 대한 첫 번째 인수로만 지원됩니다.
AVERAGEX, **COUNTAX**, **COUNTX**, **GEOMEANX**, **MAXX**, **MINX**, **PRODUCTX**, **STDEVX.S**, **STDEVX.P**, **SUMX**, **VARX.S**, **VARX.P**.

예제

GROUPBY를 참조하세요.

DATATABLE

2021-03-16 • 3 minutes to read

인라인 데이터 값 집합을 선언하는 메커니즘을 제공합니다.

구문

```
DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}}, {ValueN, ValueN+1...}...)
```

매개 변수

용어	정의
ColumnName	테이블을 반환하는 DAX 식입니다.
DataType	다음에 포함하는 열거형입니다. INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
값	<p>Excel의 1차원 배열 상수 구문을 사용하고, 배열의 배열을 제공하기 위해 중첩된 단일 인수입니다. 이 인수는 테이블에 포함될 데이터 값 집합을 나타냅니다.</p> <p>예를 들면 다음과 같습니다. { {row1의 값}, {row2의 값}, {row3의 값}... } 여기서 {row1의 값}은 쉼표로 구분된 상수 식 집합으로, 상수 조합이 DATE, TIME, BLANK 등의 일부 기본 함수와 결합되었으며 DATE와 TIME 사이에는 더하기 연산자가 있고 음수 값을 표현할 수 있도록 단항 빼기 연산자도 있습니다.</p> <p>다음은 모두 유효한 값입니다. 3, -5, BLANK(), "2009-04-15 02:45:21". 값은 즉시 실행 식 이외의 항목을 참조할 수 없으며 열, 테이블, 관계 등을 참조할 수 없습니다.</p> <p>누락된 값은 BLANK()와 동일하게 처리됩니다. 예를 들어 {1,2,BLANK(),4}와 {1,2,,4}는 같습니다.</p>

반환 값

인라인 값 집합을 선언하는 테이블입니다.

설명

- DATATABLE과 달리, [테이블 생성자](#)는 스칼라 식을 입력 값으로 허용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

```
= DataTable("Name", STRING,  
            "Region", STRING  
            ,{  
                {" User1", "East"},  
                {" User2", "East"},  
                {" User3", "West"},  
                {" User4", "West"},  
                {" User4", "East"}  
            }  
        )
```

DETAILROWS

2021-03-16 • 2 minutes to read

측정값에 대해 정의된 세부 정보 행 식을 계산하고 데이터를 반환합니다.

구문

```
DETAILROWS([Measure])
```

매개 변수

용어	정의
측정값	측정값의 이름입니다.

반환 값

세부 정보 행 식이 반환한 데이터가 있는 테이블입니다. 세부 정보 행 식이 정의되지 않은 경우 측정값을 포함하는 테이블의 데이터가 반환됩니다.

DISTINCT(열)

2021-03-16 • 5 minutes to read

지정된 열의 고유 값이 포함되어 있는 단일 열 테이블을 반환합니다. 즉, 중복 값은 제거되고 고유 값만 반환됩니다.

NOTE

이 함수는 워크시트의 셀 또는 열에 값을 반환하는 데 사용할 수 없습니다. 대신, 다른 함수에 전달한 다음, 다른 작업에 대해 계산, 합계 또는 사용할 수 있는 고유 값 목록을 가져오기 위해 수식 내에 DISTINCT 함수를 중첩합니다.

구문

DISTINCT(<column>)

매개 변수

용어	정의
열	반환할 고유 값이 들어 있는 열입니다. 또는 열을 반환하는 식입니다.

반환 값

고유 값의 열입니다.

설명

- DISTINCT의 결과는 현재 필터 컨텍스트의 영향을 받습니다. 예를 들어 다음 예제의 수식을 사용하여 측정값을 만드는 경우 특정 지역 또는 특정 기간만 표시하도록 테이블을 필터링할 때마다 결과가 변경됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 함수

다른 테이블이나 식에서 중복 행을 제거하여 테이블을 반환하는 다른 버전의 DISTINCT 함수 [DISTINCT\(테이블\)](#)가 있습니다.

VALUES 함수는 DISTINCT와 비슷합니다. 또한 고유 값 목록을 반환하는 데 사용할 수 있으며, 일반적으로 DISTINCT와 정확히 동일한 결과를 반환합니다. 그러나 일부 컨텍스트에서 VALUES는 하나의 추가 특수 값을 반환합니다. 자세한 내용은 [VALUES 함수](#)를 참조하세요.

예제

다음 수식에서는 인터넷 채널을 통해 주문을 생성한 고유한 고객 수를 계산합니다. 다음 표에서는 수식이 피벗 테이블에 추가될 때 가능한 결과를 보여 줍니다.

```
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

DISTINCT가 반환하는 값 목록을 열에 직접 붙여넣을 수는 없습니다. 대신, 목록을 사용하여 값을 계산, 필터링 또는 집계하는 다른 함수에 DISTINCT 함수의 결과를 전달합니다. 예제를 가능한 간단하게 만들기 위해 여기서는 고유 값의 테이블이 COUNTROWS 함수에 전달되었습니다.

행 레이블	액세서리	자전거	의류	총합계
2005		1013		1013
2006		2677		2677
2007	6792	4875	2867	9309
2008	9435	5451	4196	11377
총합계	15114	9132	6852	18484

또한 결과가 합산되지 않습니다. 즉, *2007*에 속한 고유 고객의 총 수는 해당 연도의 *액세서리*, *자전거* 및 *의류*에 대한 고유 고객의 합계가 아닙니다. 그 이유는 고객이 여러 그룹에서 계산될 수 있기 때문입니다.

참고 항목

[필터 함수](#)

[FILTER 함수](#)

[RELATED 함수](#)

[VALUES 함수](#)

DISTINCT(테이블)

2021-03-16 • 2 minutes to read

다른 테이블 또는 식에서 중복 행을 제거하여 테이블을 반환합니다.

구문

```
DISTINCT(<table>)
```

매개 변수

용어	정의
테이블	반환할 고유 행이 들어 있는 테이블입니다. 테이블은 테이블을 생성하는 식일 수도 있습니다.

반환 값

고유 행만 포함하는 테이블입니다.

관련 함수

열 이름을 입력 매개 변수로 사용하는 다른 버전의 DISTINCT 함수, [DISTINCT\(열\)](#)가 있습니다.

예제

다음 쿼리:

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

테이블 반환:

[VALUE1]	[VALUE2]
1	A
2	b

참고 항목

[필터 함수](#)

[DISTINCT\(열\)](#)

[FILTER 함수](#)

[RELATED 함수](#)

[VALUES 함수](#)

EXCEPT

2021-03-16 • 3 minutes to read

다른 테이블에 표시되지 않는 한 테이블의 행을 반환합니다.

구문

```
EXCEPT(<table_expression1>, <table_expression2>)
```

매개 변수

용어	정의
Table_expression	테이블을 반환하는 DAX 식입니다.

반환 값

한 테이블의 행에서 다른 테이블의 모든 행을 뺀 값을 포함하는 테이블입니다.

설명

- 행이 두 테이블 모두에 표시되는 경우 해당 행과 그 중복은 결과 세트에 표시되지 않습니다. 행이 table_expression1에서만 표시되는 경우 해당 행과 그 중복은 결과 세트에 표시됩니다.
- 열 이름은 table_expression1의 열 이름과 일치합니다.
- 반환된 테이블에는 두 번째 테이블의 열 계보에 관계없이 table_expression1의 열을 기반으로 하는 계보가 있습니다. 예를 들어 첫 번째 table_expression의 첫 번째 열에 모델의 기본 열 C1에 대한 계보가 있는 경우, Except는 두 번째 table_expression의 첫 번째 열에 있는 값의 가용성을 기준으로 행을 줄이고 기본 열 C1에 계보를 그대로 유지합니다.
- 두 테이블에 동일한 개수의 열이 있어야 합니다.
- 열은 위치를 기반으로 비교되며 데이터 비교에서는 형식이 강제 변환되지 않습니다.
- 반환되는 행 세트는 두 식의 순서에 따라 달라집니다.
- 반환된 테이블에는 table_expression1과 관련된 테이블의 열이 포함되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

States1

상태
A
b

상태
b
b
C
D
D

States2

상태
b
C
D
D
D
E
E
E

Except(States1, States2)

상태
A

Except(States2, States1)

상태
E
E
E

FILTERS

2021-03-16 • 2 minutes to read

*columnName*에 필터로 직접 적용되는 값을 반환합니다.

구문

```
FILTERS(<columnName>)
```

매개 변수

용어	DESCRIPTION
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식을 사용할 수 없습니다.

반환 값

*columnName*에 필터로 직접 적용되는 값입니다.

설명

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 열에 포함된 직접 필터 수를 확인하는 방법을 보여 줍니다.

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

이 예제를 사용하면 식이 계산되는 컨텍스트에 적용된 ResellerSales_USD[ProductKey]의 직접 필터 수를 알 수 있습니다.

GENERATE

2021-03-16 • 4 minutes to read

*table1*의 각 행 사이에 카티전 곱이 있는 테이블과 *table1*의 현재 행 컨텍스트에서 *table2*를 평가하여 얻은 테이블을 반환합니다.

구문

```
GENERATE(<table1>, <table2>)
```

매개 변수

용어	정의
table1	테이블을 반환하는 DAX 식입니다.
table2	테이블을 반환하는 DAX 식입니다.

반환 값

*table1*의 각 행 사이에 카티전 곱이 있는 테이블과 *table1*의 현재 행 컨텍스트에서 *table2*를 평가하여 얻은 테이블입니다.

설명

- table1*의 현재 행에 대한 *table2*의 계산이 빈 테이블을 반환하는 경우 결과 테이블에 *table1*의 현재 행이 포함되지 않습니다. 이는 *table1*의 현재 행이 결과에 포함되고 *table2*에 해당하는 열에 해당 행에 대한 null 값이 포함되는 GENERATEALL()과는 다릅니다.
- table1* 및 *table2*의 모든 열 이름은 달라야 합니다. 그렇지 않으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서 사용자는 다음 표와 같이 대리점 채널에 대한 지역 및 제품 범주별 판매의 요약 테이블이 필요합니다.

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
유럽	Accessories	\$ 142,227.27
유럽	Bikes	\$ 9,970,200.44
유럽	Clothing	\$ 365,847.63
유럽	구성 요소	\$ 2,214,440.19

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
북아메리카	Accessories	\$ 379,305.15
북아메리카	Bikes	\$ 52,403,796.85
북아메리카	Clothing	\$ 1,281,193.26
북아메리카	구성 요소	\$ 8,882,848.05
태평양	Accessories	\$ 12,769.57
태평양	Bikes	\$ 710,677.75
태평양	Clothing	\$ 22,902.38
태평양	구성 요소	\$ 108,549.71

다음 수식은 위의 테이블을 생성합니다.

```
GENERATE(
  SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
, SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
)
```

- 첫 번째 요약 문인 `SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])` 은 다음과 같이 각 행이 지역 그룹인 지역 그룹 테이블을 생성합니다.

SALESTERRITORY[SALESTERRITORYGROUP]
북아메리카
유럽
태평양
해당 없음

- 두 번째 요약 문인

```
SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales",
SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

는 아래와 같이 각 그룹에 대한 대리점 판매가 포함된 제품 범주 그룹의 테이블을 생성합니다.

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
Bikes	\$ 63,084,675.04
구성 요소	\$ 11,205,837.96
Clothing	\$ 1,669,943.27

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
Accessories	\$ 534,301.99

3. 단, 위의 표를 사용하여 지역 그룹 테이블의 각 행 컨텍스트에서 계산할 경우 각 지역에 대해 다른 결과를 얻을 수 있습니다.

GENERATEALL

2021-03-16 • 4 minutes to read

*table1*의 각 행 사이에 카티전 곱이 있는 테이블과 *table1*의 현재 행 컨텍스트에서 *table2*를 평가하여 얻은 테이블을 반환합니다.

구문

```
GENERATEALL(<table1>, <table2>)
```

매개 변수

용어	정의
table1	테이블을 반환하는 DAX 식입니다.
table2	테이블을 반환하는 DAX 식입니다.

반환 값

*table1*의 각 행 사이에 카티전 곱이 있는 테이블과 *table1*의 현재 행 컨텍스트에서 *table2*를 평가하여 얻은 테이블입니다.

설명

- table1*의 현재 행에 대한 *table2*를 계산할 때 빈 테이블이 반환되는 경우 *table1*의 현재 행이 결과에 포함되고 *table2*에 해당하는 열에 해당 행에 대한 null 값이 포함됩니다. 이는 *table1*의 현재 행이 결과에 포함되지 않는 `GENERATE()`와는 다릅니다.
- table1* 및 *table2*의 모든 열 이름은 달라야 합니다. 그렇지 않으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서 사용자는 다음 표와 같이 대리점 채널에 대한 지역 및 제품 범주별 판매의 요약 테이블이 필요합니다.

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
유럽	Accessories	\$ 142,227.27
유럽	Bikes	\$ 9,970,200.44
유럽	Clothing	\$ 365,847.63
유럽	구성 요소	\$ 2,214,440.19

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
해당 없음	Accessories	
해당 없음	Bikes	
해당 없음	Clothing	
해당 없음	구성 요소	
북아메리카	Accessories	\$ 379,305.15
북아메리카	Bikes	\$ 52,403,796.85
북아메리카	Clothing	\$ 1,281,193.26
북아메리카	구성 요소	\$ 8,882,848.05
태평양	Accessories	\$ 12,769.57
태평양	Bikes	\$ 710,677.75
태평양	Clothing	\$ 22,902.38
태평양	구성 요소	\$ 108,549.71

다음 수식은 위의 테이블을 생성합니다.

```
GENERATEALL(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
,SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
)
)
```

1. 첫 번째 SUMMARIZE는 지역 그룹 테이블을 생성합니다. 각 행은 아래에 나열된 것과 같은 지역 그룹입니다.

SALESTERRITORY[SALESTERRITORYGROUP]
북아메리카
유럽
태평양
해당 없음

2. 두 번째 SUMMARIZE는 아래와 같이 각 그룹에 대한 대리점 판매가 포함된 제품 범주 그룹의 테이블을 생성합니다.

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[RESELLER SALES]
Bikes	\$ 63,084,675.04
구성 요소	\$ 11,205,837.96
Clothing	\$ 1,669,943.27
Accessories	\$ 534,301.99

- 단, 위의 표를 사용하여 지역 그룹 테이블의 각 행 컨텍스트에서 테이블을 계산할 경우 각 지역에 대해 다른 결과를 얻을 수 있습니다.

GENERATESERIES

2021-03-16 • 2 minutes to read

산술 계열의 값을 포함하는 단일 열 테이블을 반환합니다. 즉, 각 값이 상수 수량에 따라 앞과 다른 시퀀스입니다. 반환된 열의 이름은 Value입니다.

구문

```
GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

매개 변수

용어	정의
startValue	시퀀스를 생성하는 데 사용되는 초기 값입니다.
endValue	시퀀스를 생성하는 데 사용되는 끝 값입니다.
incrementValue	(선택 사항) 시퀀스의 증가값입니다. 이 값을 지정하지 않으면 기본값은 1입니다.

반환 값

산술 계열의 값을 포함하는 단일 열 테이블입니다. 열의 이름은 Value입니다.

설명

- startValue가 endValue보다 작은 경우 빈 테이블이 반환됩니다.
- incrementValue는 양수 값이어야 합니다.
- 시퀀스는 endValue보다 작거나 같은 마지막 값에서 중지됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

DAX 쿼리는 다음과 같습니다.

```
EVALUATE GENERATESERIES(1, 5)
```

단일 열을 포함하는 다음 테이블을 반환합니다.

[VALUE]
1
2

[VALUE]
3
4
5

예제 2

DAX 쿼리는 다음과 같습니다.

```
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)
```

단일 열을 포함하는 다음 테이블을 반환합니다.

[VALUE]
1.2
1.6
2
2.4

예제 3

DAX 쿼리는 다음과 같습니다.

```
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

단일 열을 포함하는 다음 테이블을 반환합니다.

[VALUE]
10
10.5
11
11.5
12

GROUPBY

2021-03-16 • 7 minutes to read

GROUPBY 함수는 [SUMMARIZE](#) 함수와 비슷합니다. 하지만 GROUPBY는 추가하는 확장 열에 대해 암시적 [CALCULATE](#)를 수행하지 않습니다. GROUPBY를 사용하면 추가하는 확장 열에 있는 집계 함수 내에서 새로운 [CURRENTGROUP](#) 함수를 사용할 수 있습니다. GROUPBY는 단일 테이블 검색에서 여러 집계를 수행하는 데 사용됩니다.

구문

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, ...]] [, <name>, <expression> [, <name>, <expression> [, ...]])
```

매개 변수

용어	정의
테이블	데이터 테이블을 반환하는 모든 DAX 식입니다.
groupBy_columnName	그룹화할 데이터를 기준으로 테이블(또는 관련 테이블)에 있는 기존 열의 이름입니다. 이 매개 변수는 식일 수 없습니다.
name	큰 따옴표로 묶어서 GroupBy 열 목록에 추가 중인 새 열에 지정된 이름입니다.
expression	첫 번째 인수가 CURRENTGROUP()인 X 집계 함수 중 하나입니다. 지원되는 X 집계 함수의 전체 목록은 아래의 CURRENTGROUP 사용 섹션을 참조하세요.

반환 값

groupBy_columnName 인수에 대해 선택된 열과 이름 인수로 지정된 확장 열이 포함된 테이블입니다.

설명

- GROUPBY 함수는 다음을 수행합니다.
 - 지정된 테이블(및 "to-one" 방향의 모든 관련 테이블)로 시작됩니다.
 - 모든 GroupBy 열(1단계에서 테이블에 존재해야 함)을 사용하여 그룹화를 만듭니다.
 - 각 그룹은 결과에서 하나의 행으로 표시되지만, 원래 테이블의 행 세트를 나타냅니다.
 - 각 그룹에 대해 추가되는 확장 열을 평가합니다. SUMMARIZE 함수와 달리 암시적 CALCULATE가 수행되지 않으며 그룹이 필터 컨텍스트에 배치되지 않습니다.
- 이름을 정의하는 각 열에는 해당 식이 있어야 합니다. 식이 없으면 오류가 반환됩니다. 첫 번째 인수인 name은 결과에 있는 열의 이름을 정의합니다. 두 번째 인수인 expression은 해당 열에서 각 행의 값을 가져오기 위해 수행되는 계산을 정의합니다.
- groupBy_columnName은 테이블 또는 관련 테이블에 있어야 합니다.

- 각 이름은 큰따옴표로 묶여 있어야 합니다.
- 이 함수는 하나 이상의 `groupBy_columnName` 열의 값에 따라 선택한 행 세트를 요약 행 집합으로 그룹화합니다. 그룹마다 하나의 행이 반환됩니다.
- GROUPBY는 DAX 테이블 식의 중간 결과에 대해 집계를 수행하는 데 주로 사용됩니다. 모델의 물리적 테이블을 효율적으로 집계하려면 [SUMMARIZECOLUMNS](#) 또는 [SUMMARIZE](#) 함수를 사용하는 것이 좋습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

CURRENTGROUP 사용

[CURRENTGROUP](#)은 GROUPBY 함수 내에서 확장 열을 정의하는 식에서만 사용할 수 있습니다. 실제로 [CURRENTGROUP](#)은 GROUPBY 결과의 현재 행에 속하는 GROUPBY의 table 인수로부터 행 집합을 반환합니다. [CURRENTGROUP](#) 함수는 인수를 사용하지 않으며 다음 집계 함수 중 하나에 대한 첫 번째 인수로만 지원됩니다. [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#), [VARX.P](#).

예제

다음 예제에서는 [SUMMARIZECOLUMNS](#) 함수를 사용하여 실제 테이블에서 국가 및 제품 범주 기준으로 그룹화된 총 판매액을 먼저 계산합니다. 그런 다음 GROUPBY 함수를 사용하여 첫 번째 단계의 중간 결과를 검색하여 제품 범주에서 각 국가의 최대 판매액을 찾습니다.

```
DEFINE
VAR SalesByCountryAndCategory =
SUMMARIZECOLUMNS(
    Geography[Country],
    Product[Category],
    "Total Sales", SUMX(Sales, Sales[Price] * Sales[Qty])
)

EVALUATE
GROUPBY(
    SalesByCountryAndCategory,
    Geography[Country],
    "Max Sales", MAXX(CURRENTGROUP(), [Total Sales])
)
```

참고 항목

[SUMMARIZE](#) 함수

[SUMMARIZECOLUMNS](#) 함수

IGNORE

2021-03-16 • 2 minutes to read

BLANK/NULL 계산에서 특정 식을 생략하여 [SUMMARIZECOLUMNS](#) 함수의 동작을 수정합니다. IGNORE를 사용하는 식이 BLANK/NULL로 계산되는지 여부와 관계없이, IGNORE를 사용하지 않는 모든 식이 BLANK/NULL 반환하는 행은 제외됩니다. 이 함수는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다.

구문

```
IGNORE(<expression>)
```

SUMMARIZECOLUMNS 포함

```
SUMMARIZECOLUMNS(<groupBy_columnName>[, <groupBy_columnName >]..., [<filterTable>]...[, <name>, IGNORE(...)]...)
```

매개 변수

용어	정의
expression	단일 값(테이블이 아님)을 반환하는 DAX 식입니다.

반환 값

함수가 값을 반환하지 않습니다.

설명

IGNORE는 [SUMMARIZECOLUMNS](#)에 대한 식 인수로만 사용할 수 있습니다.

예제

[SUMMARIZECOLUMNS](#)를 참조하세요.

INTERSECT

2021-03-16 • 3 minutes to read

중복 행을 유지하여 두 테이블의 행 교집합을 반환합니다.

구문

```
INTERSECT(<table_expression1>, <table_expression2>)
```

매개 변수

용어	정의
Table_expression	테이블을 반환하는 DAX 식입니다.

반환 값

table_expression2에도 있는 table_expression1의 모든 행이 포함된 테이블입니다.

예외

설명

- 교차는 가환성이 없습니다. 일반적으로 $\text{Intersect}(T1, T2)$ 는 $\text{Intersect}(T2, T1)$ 와 결과 집합이 다릅니다.
- 중복 행은 유지됩니다. 특정 행이 table_expression1 및 table_expression2에 표시되는 경우 해당 행과 table_expression_1의 모든 중복 행이 결과 집합에 포함됩니다.
- 열 이름은 table_expression1의 열 이름과 일치합니다.
- 반환된 테이블에는 두 번째 테이블의 열 계보에 관계없이 table_expression1의 열을 기반으로 하는 계보가 있습니다. 예를 들어 첫 번째 table_expression의 첫 번째 열에 모델의 기본 열 C1에 대한 계보가 있으면 교차에서는 두 번째 table_expression의 첫 번째 열에 대한 교차를 기반으로 행을 줄이고 기본 열 C1에 대한 계보를 그대로 유지합니다.
- 열은 위치를 기반으로 비교되며 데이터 비교에서는 형식이 강제 변환되지 않습니다.
- 반환된 테이블에는 table_expression1과 관련된 테이블의 열이 포함되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

States1

상태
A

상태
A
b
b
b
C
D
D

States2

상태
b
C
D
D
D
E

Intersect(States1, States2)

상태
b
b
b
C
D
D

Intersect(States2, States1)

상태

상태
b
C
D
D
D

NATURALINNERJOIN

2021-03-16 • 2 minutes to read

다른 테이블을 사용하여 한 테이블의 내부 조인을 수행합니다. 두 테이블의 공통 열(이름별)에서 조인됩니다. 두 테이블에 공통 열 이름이 없으면 오류가 반환됩니다.

구문

```
NATURALINNERJOIN(<leftJoinTable>, <rightJoinTable>)
```

매개 변수

용어	정의
leftJoinTable	조인의 왼쪽 테이블을 정의하는 테이블 식입니다.
rightJoinTable	조인의 오른쪽 테이블을 정의하는 테이블 식입니다.

반환 값

지정된 공통 열의 값이 두 테이블에 모두 있는 행만 포함하는 테이블입니다. 반환되는 테이블에는 왼쪽 테이블의 공통 열과 두 테이블의 다른 열이 포함됩니다.

설명

- 결과의 정렬 순서는 보장되지 않습니다.
- 조인되는 열의 데이터 형식은 두 테이블에서 동일해야 합니다.
- 동일한 소스 테이블의 열(계보가 동일함)만 조인됩니다. 예를 들어 WebSales와 StoreSales 간에 일 대 다 관계가 있고 Products 테이블이 ProductID 열과 WebSales 및 StoreSales 테이블을 기반으로 하는 Products[ProductID], WebSales[ProductID], StoreSales[ProductID]는 [ProductID]에서 조인됩니다.
- 조인하는 동안 엄격한 비교 의미 체계가 사용됩니다. 형식 강제 변환은 없습니다. 예를 들어 1은 1.0과 같지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

NATURALLEFTOUTERJOIN

2021-03-16 • 2 minutes to read

다른 테이블을 사용하여 한 테이블의 내부 조인을 수행합니다. 두 테이블의 공통 열(이름별)에서 조인됩니다. 두 테이블에 공통 열 이름이 없으면 오류가 반환됩니다.

구문

```
NATURALLEFTOUTERJOIN(<leftJoinTable>, <rightJoinTable>)
```

매개 변수

용어	정의
leftJoinTable	조인의 왼쪽 테이블을 정의하는 테이블 식입니다.
rightJoinTable	조인의 오른쪽 테이블을 정의하는 테이블 식입니다.

반환 값

지정된 공통 열의 값이 leftJoinTable에도 있는 rightJoinTable의 행만 포함하는 테이블입니다. 반환된 테이블에는 왼쪽 테이블의 공통 열과 두 테이블의 다른 열이 포함됩니다.

설명

- 결과의 정렬 순서는 보장되지 않습니다.
- 조인되는 열의 데이터 형식은 두 테이블에서 동일해야 합니다.
- 동일한 소스 테이블의 열(계보가 동일함)만 조인됩니다. 예를 들어 WebSales와 StoreSales 간에 일 대 다 관계가 있고 Products 테이블이 ProductID 열과 WebSales 및 StoreSales 테이블을 기반으로 하는 Products[ProductID], WebSales[ProductID], StoreSales[ProductID]는 [ProductID]에서 조인됩니다.
- 조인하는 동안 엄격한 비교 의미 체계가 사용됩니다. 형식 강제 변환은 없습니다. 예를 들어 1은 1.0과 같지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

ROLLUP

2021-03-16 • 2 minutes to read

groupBy_columnName 매개 변수에 의해 정의되는 열에서 결과에 롤업 행을 추가하여 [SUMMARIZE](#) 함수의 동작을 수정합니다. 이 함수는 [SUMMARIZE](#) 식 내에서만 사용할 수 있습니다.

구문

```
ROLLUP ( <groupBy_columnName> [, <groupBy_columnName> [, ... ] ] )
```

SUMMARIZE 사용

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, <expression>]...)
```

매개 변수

용어	정의
groupBy_columnName	찾은 값을 기반으로 요약 그룹을 만드는 데 사용할 기존 열 또는 ROLLUPGROUP 함수의 정규화된 이름입니다. 이 매개 변수는 식일 수 없습니다.

반환 값

이 함수는 값을 반환하지 않습니다. 소계가 될 열의 세트를 지정하기만 합니다.

설명

이 함수는 [SUMMARIZE](#) 식 내에서만 사용할 수 있습니다.

예제

[SUMMARIZE](#)를 참조하세요.

ROLLUPADDISSUBTOTAL

2021-03-16 • 2 minutes to read

groupBy_columnName 열을 기반으로 결과에 롤업/소계 행을 추가하여 [SUMMARIZECOLUMNS](#) 함수의 동작을 수정합니다. 이 함수는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다.

구문

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

매개 변수

용어	정의
grandtotalFilter	(선택 사항) 총합계 수준에 적용할 필터입니다.
groupBy_columnName	찾은 값을 기반으로 요약 그룹을 만드는 데 사용되는 기존 열의 이름입니다. 식이 될 수 없습니다.
name	IsSubtotal 열의 이름입니다. 열의 값은 ISSUBTOTAL 함수를 사용하여 계산됩니다.
groupLevelFilter	(선택 사항) 현재 수준에 적용할 필터입니다.

반환 값

함수가 값을 반환하지 않습니다.

설명

None

예제

[SUMMARIZECOLUMNS](#)를 참조하세요.

ROLLUPGROUP

2021-03-16 • 2 minutes to read

groupBy_columnName 매개 변수에 의해 정의되는 열에서 결과에 롤업 행을 추가하여 [SUMMARIZE](#) 및 [SUMMARIZECOLUMNS](#)의 동작을 수정합니다. 이 함수는 [SUMMARIZE](#) 또는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다.

구문

```
ROLLUPGROUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

매개 변수

용어	정의
groupBy_columnName	찾은 값을 기반으로 요약 그룹을 만드는 데 사용할 기존 열 또는 ROLLUPGROUP 함수의 정규화된 이름입니다. 이 매개 변수는 식일 수 없습니다.

반환 값

이 함수는 값을 반환하지 않습니다. [ROLLUP](#) 또는 [ROLLUPADDISSUBTOTAL](#)에 의한 소계 작업 중에 단일 그룹으로 취급할 열 집합을 표시합니다.

설명

ROLLUPGROUP은 [ROLLUP](#), [ROLLUPADDISSUBTOTAL](#), 또는 [ROLLUPISSUBTOTAL](#)의 groupBy_columnName 인수로만 사용할 수 있습니다.

예제

[SUMMARIZE](#) 및 [SUMMARIZECOLUMNS](#)를 참조하세요.

ROLLUPISSUBTOTAL

2021-03-16 • 2 minutes to read

[ROLLUPADDISSUBTOTAL](#)에서 추가한 열과 롤업 그룹을 연결합니다. 이 함수는 [ADDMISSINGITEMS](#) 식 내에서만 사용할 수 있습니다.

구문

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>, <isSubtotal_columnName> [,
[<groupLevelFilter>] [, <groupBy_columnName>, <isSubtotal_columnName> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

매개 변수

용어	정의
grandTotalFilter	(선택 사항) 총합계 수준에 적용할 필터입니다.
groupBy_columnName	찾은 값을 기반으로 요약 그룹을 만드는 데 사용되는 기존 열의 이름입니다. 식이 될 수 없습니다.
isSubtotal_columnName	IsSubtotal 열의 이름입니다. 열의 값은 ISSUBTOTAL 함수를 사용하여 계산됩니다.
groupLevelFilter	(선택 사항) 현재 수준에 적용할 필터입니다.

반환 값

없음

설명

이 함수는 [ADDMISSINGITEMS](#) 식 내에서만 사용할 수 있습니다.

ROW 함수

2021-03-16 • 2 minutes to read

각 열에 지정된 식에서 얻은 값이 포함된 단일 행이 있는 테이블을 반환합니다.

구문

```
ROW(<name>, <expression>[[,<name>, <expression>]...])
```

매개 변수

용어	정의
name	열에 지정된 이름으로, 큰따옴표로 묶여 있습니다.
식	채우려는 단일 스칼라 값을 반환하는 DAX 식입니다. <i>name</i> .

반환 값

단일 행 테이블

설명

- 인수는 항상 *name* 및 *expression* 의 쌍으로 제공되어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 인터넷 및 대리점 채널의 총 판매가 포함된 단일 행 테이블을 반환합니다.

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),  
    "Resellers Total Sales (USD)", SUM(ResellerSales_USD[SalesAmount_USD]))
```

SELECTCOLUMNS

2021-03-16 • 2 minutes to read

지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.

구문

```
SELECTCOLUMNS(<table>, <name>, <scalar_expression> [, <name>, <scalar_expression>]...)
```

매개 변수

용어	정의
테이블	테이블을 반환하는 DAX 식입니다.
name	열에 지정된 이름으로, 큰따옴표로 묶여 있습니다.
식	열 참조, 정수 또는 문자열 값과 같은 스칼라 값을 반환하는 식입니다.

반환 값

첫 번째 인수로 지정된 테이블과 행 수가 동일한 테이블입니다. 반환되는 테이블에는 <name>, <scalar_expression> 인수의 각 쌍마다 하나의 열이 있고, 각 식은 지정된 <table> 인수의 행 컨텍스트에서 계산됩니다.

설명

SELECTCOLUMNS는 ADDCOLUMNS와 시그니처 및 동작이 동일합니다. 단, SELECTCOLUMNS는 지정된 <table>에서 시작하지 않고 빈 테이블로 시작하여 열을 추가합니다.

이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 Info 테이블을 사용한다고 가정합니다.

국가	시스템 상태	개수	합계
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
미국	CA	5	500
미국	WA	10	900

```
SELECT COLUMNS(Info, "StateCountry", [State]&", "&[Country])
```

그러면 다음 값이 반환됩니다.

STATECOUNTRY
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

SUBSTITUTEWITHINDEX

2021-03-16 • 6 minutes to read

인수로 제공된 두 테이블의 왼쪽 세미조인을 나타내는 테이블을 반환합니다. 세미조인은 공통 열 이름 및 공통 데이터 형식에 의해 결정되는 공통 열을 사용하여 수행됩니다. 조인되는 열은 정수 형식이며 인덱스를 포함하는 반환된 테이블의 단일 열로 바뀝니다. 인덱스는 지정된 정렬 순서를 제공하는 경우 오른쪽 조인 테이블에 대한 참조입니다.

제공된 오른쪽/두 번째 테이블의 열은 제공된 왼쪽/첫 번째 테이블에 없으며, 반환된 테이블에 포함되지 않고 조인에 사용되지 않습니다.

인덱스는 0(0부터 시작)에서 시작되며, 제공된 오른쪽/두 번째 조인 테이블의 추가 행마다 하나씩 증가합니다. 인덱스는 오른쪽/두 번째 조인 테이블에 대해 지정된 정렬 순서를 기반으로 합니다.

구문

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>]][, <orderBy_expression>, [<order>]]...)
```

매개 변수

용어	정의
테이블	세 번째 인수(indexColumnsTable)로 지정된 테이블을 통해 왼쪽 세미조인을 수행하여 필터링할 테이블입니다. 이 테이블은 왼쪽 세미조인의 왼쪽에 있는 테이블이므로 반환되는 테이블에 이 테이블과 동일한 열이 포함되지만, 두 테이블의 모든 공통 열이 반환된 테이블의 단일 인덱스 열로 바뀝니다.
indexColumnName	이 함수에 인수로 제공된 두 테이블의 모든 공통 열을 바꾸는 인덱스 열의 이름을 지정하는 문자열입니다.
indexColumnsTable	왼쪽 세미조인의 두 번째 테이블입니다. 왼쪽 세미조인의 오른쪽에 있는 테이블입니다. 이 테이블에 있는 값만 함수를 통해 반환됩니다. 또한 이 테이블의 열(열 이름 기반)은 이 함수를 통해 반환되는 테이블의 단일 인덱스 열로 바뀝니다.
orderBy_expression	올바른 인덱스 값을 생성하기 위해 indexColumnsTable 테이블에 대해 원하는 정렬 순서를 지정하는 데 결과 값이 사용되는 DAX 식입니다. indexColumnsTable 테이블에 대해 지정된 정렬 순서는 테이블에 있는 각 행의 인덱스를 정의하고, 해당 인덱스는 반환된 테이블에서 이 함수의 첫 번째 인수로 제공된 테이블에 표시되는 indexColumnsTable의 값 조합을 나타내는 데 사용됩니다.

용어	정의
순서	<p>(선택 사항) orderBy_expression 값을 정렬하는 방법(오름차순 또는 내림차순)을 지정하는 값입니다.</p> <p>값: Desc. 대체 값: 0(영)/FALSE. orderBy_expression 값을 내림차순으로 정렬합니다. order 매개 변수가 생략되는 경우 이 값이 기본값입니다.</p> <p>값: ASC. 대체 값: 1/TRUE. orderBy_expression을 오름차순으로 순위 지정합니다.</p>

반환 값

indexColumnsTable 테이블에 있는 값만 포함하고, indexColumnsTable 테이블에 있는 모든 열(이름별) 대신 인덱스 열이 있는 테이블입니다.

설명

- 이 함수는 결과 정렬 순서를 보장하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

SUMMARIZE

2021-03-16 • 11 minutes to read

그룹 집합에 대해 요청된 합계의 요약 테이블을 반환합니다.

구문

```
SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

매개 변수

용어	정의
테이블	데이터 테이블을 반환하는 모든 DAX 식입니다.
groupBy_ColumnName	(선택 사항) 찾은 값을 기반으로 요약 그룹을 만드는 데 사용되는 기존 열의 정규화된 이름입니다. 이 매개 변수는 식이 될 수 없습니다.
name	합계 또는 요약 열에 지정된 이름으로, 큰따옴표로 묶여 있습니다.
expression	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

groupBy_columnName 인수에 대해 선택된 열과 *name* 인수를 통해 설계된 요약 열이 포함된 테이블입니다.

설명

- 이름을 정의하는 각 열에는 해당 식이 있어야 합니다. 식이 없으면 오류가 반환됩니다. 첫 번째 인수인 *name*은 결과에 있는 열의 이름을 정의합니다. 두 번째 인수인 *expression*은 해당 열에서 각 행의 값을 가져오기 위해 수행되는 계산을 정의합니다.
- groupBy_columnName*은 *table* 또는 관련 *table-table*에 있어야 합니다.
- 각 이름은 큰따옴표로 묶여 있어야 합니다.
- 이 함수는 하나 이상의 *groupBy_columnName* 열의 값에 따라 선택한 행 세트를 요약 행 집합으로 그룹화합니다. 그룹마다 하나의 행이 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 달력 연도 및 제품 범주 이름에 대해 그룹화된 재판매인 판매 요약을 반환합니다. 이 결과 테이블을 통해 연도 및 제품 범주별로 재판매인 판매에 대한 분석을 수행할 수 있습니다.


```
SUMMARIZE(ResellerSales_USD
, DateTime[CalendarYear]
, ProductCategory[ProductCategoryName]
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
, "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

다음 테이블에서는 테이블을 수신해야 하는 함수가 수신하는 데이터에 대한 미리 보기를 보여 줍니다.

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2008	자전거	12968255.42	36167.6592
2005	자전거	6958251.043	4231.1621
2006	자전거	18901351.08	178175.8399
2007	자전거	24256817.5	276065.992
2008	구성 요소	2008052.706	39.9266
2005	구성 요소	574256.9865	0
2006	구성 요소	3428213.05	948.7674
2007	구성 요소	5195315.216	4226.0444
2008	의류	366507.844	4151.1235
2005	의류	31851.1628	90.9593
2006	의류	455730.9729	4233.039
2007	의류	815853.2868	12489.3835
2008	액세서리	153299.924	865.5945
2005	액세서리	18594.4782	4.293
2006	액세서리	86612.7463	1061.4872
2007	액세서리	275794.8403	4756.6546

ROLLUP 사용

ROLLUP 구문이 추가되어 `groupBy_columnName` 열의 결과에 롤업 행을 추가하면 **SUMMARIZE** 함수의 동작이 수정됩니다. **ROLLUP**은 **SUMMARIZE** 식 내에서만 사용할 수 있습니다.

예제

다음 예제에서는 **SUMMARIZE** 함수 호출의 **Group-By** 열에 롤업 행을 추가합니다.

```

SUMMARIZE(ResellerSales_USD
, ROLLUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName])
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
, "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)

```

다음 테이블을 반환합니다.

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2008	자전거	12968255.42	36167.6592
2005	자전거	6958251.043	4231.1621
2006	자전거	18901351.08	178175.8399
2007	자전거	24256817.5	276065.992
2008	구성 요소	2008052.706	39.9266
2005	구성 요소	574256.9865	0
2006	구성 요소	3428213.05	948.7674
2007	구성 요소	5195315.216	4226.0444
2008	의류	366507.844	4151.1235
2005	의류	31851.1628	90.9593
2006	의류	455730.9729	4233.039
2007	의류	815853.2868	12489.3835
2008	액세서리	153299.924	865.5945
2005	액세서리	18594.4782	4.293
2006	액세서리	86612.7463	1061.4872
2007	액세서리	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

ROLLUPGROUP 사용

ROLLUP 구문 내에 ROLLUPGROUP을 추가하여 롤업 행에서 부분 소계를 방지하는 데 사용할 수 있습니다. ROLLUPGROUP은 ROLLUP, ROLLUPADDISSUBTOTAL 또는 ROLLUPISSUBTOTAL 식 내에서만 사용할 수 있습니다.

예제

다음 예제는 모든 범주에서 연도별 소계 없이 모든 연도 및 범주의 총합계만 보여 줍니다.

```
SUMMARIZE(ResellerSales_USD
, ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName]))
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
, "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

다음 테이블을 반환합니다.

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2008	자전거	12968255.42	36167.6592
2005	자전거	6958251.043	4231.1621
2006	자전거	18901351.08	178175.8399
2007	자전거	24256817.5	276065.992
2008	구성 요소	2008052.706	39.9266
2005	구성 요소	574256.9865	0
2006	구성 요소	3428213.05	948.7674
2007	구성 요소	5195315.216	4226.0444
2008	의류	366507.844	4151.1235
2005	의류	31851.1628	90.9593
2006	의류	455730.9729	4233.039
2007	의류	815853.2868	12489.3835
2008	액세서리	153299.924	865.5945
2005	액세서리	18594.4782	4.293
2006	액세서리	86612.7463	1061.4872
2007	액세서리	275794.8403	4756.6546
		76494758.25	527507.9262

ISSUBTOTAL 사용

ISSUBTOTAL을 사용하면 행에 ISSUBTOTAL에 대한 인수로 지정된 열의 소계 값이 포함된 경우 True를 반환하고 포함되지 않은 경우 False를 반환하는 또 다른 열을 SUMMARIZE 함수에서 만들 수 있습니다. ISSUBTOTAL은 SUMMARIZE 식 내에서만 사용할 수 있습니다.

예제

다음 샘플은 지정된 SUMMARIZE 함수 호출에 있는 각 ROLLUP 열에 대해 ISSUBTOTAL 열을 생성합니다.

```
SUMMARIZE(ResellerSales_USD
, ROLLUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName])
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
, "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
, "Is Sub Total for DateTimeCalendarYear", ISSUBTOTAL(DateTime[CalendarYear])
, "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductCategoryName])
)
```

다음 테이블을 반환합니다.

[IS SUB TOTAL FOR DATETIMECALENDAR YEAR]	[IS SUB TOTAL FOR PRODUCTCATEGORYNAME]	DATETIME[CALENDAR YEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
FALSE	FALSE				
FALSE	FALSE	2008	자전거	12968255.42	36167.6592
FALSE	FALSE	2005	자전거	6958251.043	4231.1621
FALSE	FALSE	2006	자전거	18901351.08	178175.8399
FALSE	FALSE	2007	자전거	24256817.5	276065.992
FALSE	FALSE	2008	구성 요소	2008052.706	39.9266
FALSE	FALSE	2005	구성 요소	574256.9865	0
FALSE	FALSE	2006	구성 요소	3428213.05	948.7674
FALSE	FALSE	2007	구성 요소	5195315.216	4226.0444
FALSE	FALSE	2008	의류	366507.844	4151.1235
FALSE	FALSE	2005	의류	31851.1628	90.9593
FALSE	FALSE	2006	의류	455730.9729	4233.039
FALSE	FALSE	2007	의류	815853.2868	12489.3835
FALSE	FALSE	2008	액세서리	153299.924	865.5945
FALSE	FALSE	2005	액세서리	18594.4782	4.293
FALSE	FALSE	2006	액세서리	86612.7463	1061.4872

[IS SUB TOTAL FOR DATETIMECALEN DARYEAR]	[IS SUB TOTAL FOR PRODUCTCATEG ORYNAME]	DATETIME[CALE NDARYEAR]	PRODUCTCATEG ORY[PRODUCTC ATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
FALSE	FALSE	2007	액세서리	275794.8403	4756.6546
FALSE	TRUE				
FALSE	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

참고 항목

[SUMMARIZECOLUMNS](#)

SUMMARIZECOLUMNS

2021-03-16 • 12 minutes to read

그룹 집합의 요약 테이블을 반환합니다.

구문

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, <groupBy_columnName >]..., [<filterTable>]...[, <name>, <expression>] ...)
```

매개 변수

용어	정의
groupBy_columnName	반환된 테이블에 별개의 값이 포함되는 기본 테이블에 대한 정규화된 열 참조 (Table[Column])입니다. 각 groupBy_columnName 열은 이후에 지정된 열과 교차 조인 (다른 테이블) 또는 자동 존재(같은 테이블)가 수행됩니다.
filterTable	groupBy_columnName 인수로 지정된 모든 열의 필터 컨텍스트에 추가되는 테이블 식입니다. 필터 테이블에 있는 값은 교차 조인/자동 존재가 수행되기 전에 필터링하는 데 사용됩니다.
name	지정된 후속 식에 사용할 열 이름을 나타내는 문자열입니다.
식	단일 값(테이블이 아님)을 반환하는 DAX 식입니다.

반환 값

지정된 그룹화를 기준으로, 제공된 열의 값 조합을 포함하는 테이블입니다. 하나 이상의 제공된 식이 비어 있지 않은 값을 반환하는 행만 반환되는 테이블에 포함됩니다. 모든 식이 행에 대해 BLANK/NULL로 계산되는 경우 반환되는 테이블에 해당 행이 포함되지 않습니다.

설명

- 이 함수는 결과에 대한 정렬 순서를 보장하지 않습니다.
- groupBy_columnName 매개 변수에 열을 두 번 이상 지정할 수 없습니다. 예를 들어 다음 수식은 잘못되었습니다.

```
SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
```

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

필터 컨텍스트

다음과 같은 쿼리를 고려해 보세요.

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

이 쿼리에서는 측정값이 없으면 `groupBy` 열에 필터 식(예: Customer 테이블의 필터 식)의 열이 포함되지 않습니다. 필터가 `groupBy` 열에 적용되지 않습니다. Sales Territory 테이블과 Customer 테이블은 Reseller Sales 팩트 테이블을 통해 간접적으로 관련될 수 있습니다. 직접적으로 관련되지 않으므로 필터 식은 작동하지 않으며 `groupBy` 열에는 영향을 주지 않습니다.

그러나 다음 쿼리를 사용하세요.

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

`groupBy` 열은 필터의 영향을 받는 열을 포함하며 해당 필터는 `groupBy` 결과에 적용됩니다.

IGNORE 포함

BLANK/NULL 계산에서 특정 식을 생략하면 **IGNORE** 구문을 사용하여 SUMMARIZECOLUMNS 함수의 동작을 수정할 수 있습니다. **IGNORE**를 사용하는 식이 BLANK/NULL로 계산되는지 여부와 관계없이 **IGNORE**를 사용하지 않는 모든 식이 BLANK/NULL을 반환하는 행은 제외됩니다. **IGNORE**는 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.

예제

```
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Total Qty",
    IGNORE( SUM( Sales[Qty] ) ),
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )
)
```

Sales[CustomerId] 열을 롤업하여, 지정된 그룹의 모든 고객에 대한 소계를 만듭니다. **IGNORE**가 없으면 결과는 다음과 같습니다.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
A	5	
b	3	3
C	3	3

IGNORE를 사용하면 다음과 같습니다.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
b	3	3
C	3	3

모든 식이 무시됩니다.

```
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Blank",
    IGNORE( Blank() ), "BlankIfTotalQtyIsNot5",
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )
)
```

두 식 모두 일부 행에 대해 blank를 반환하더라도 blank를 반환하는 무시되지 않는 식이 없기 때문에 둘 다 포함됩니다.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
A		5
b		
C		

NONVISUAL 사용

NONVISUAL 함수는 SUMMARIZECOLUMNS 함수의 값 필터를 측정값에 영향을 주지 않고 group-by 열에만 적용하도록 표시합니다. **NONVISUAL**는 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.

예제

```
DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    NONVISUAL(TREATAS({2007, 2008}, DimDate[CalendarYear])),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales], ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]
```

결과를 반환합니다. 여기서 [Visual Total Sales]는 모든 연도의 합계입니다.

DIMDATE[CALENDARYEAR]	[SALES]	[VISUAL TOTAL SALES]
2007	9,791,060.30	29,358,677.22
2008	9,770,899.74	29,358,677.22

반면, *NONVISUAL* 함수를 사용하지 않는 동일한 쿼리는 다음과 같습니다.


```

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]

```

결과를 반환합니다. 여기서 [Visual Total Sales]는 선택한 두 연도의 합계입니다.

DIMDATE[CALENDARYEAR]	[SALES]	[VISUAL TOTAL SALES]
2007	9,791,060.30	19,561,960.04
2008	9,770,899.74	19,561,960.04

ROLLUPADDISSUBTOTAL 포함

ROLLUPADDISSUBTOTAL 구문을 추가하면 groupBy_columnName 열을 기반으로 결과에 롤업/소계 행이 추가되어 SUMMARIZECOLUMNS 함수의 동작이 수정됩니다. **ROLLUPADDISSUBTOTAL**은 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.

소계가 하나인 예제

```

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]

```

다음 테이블을 반환합니다.

범주	하위 범주	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	총 수량
		True	True	60398
Accessories		False	True	36092
Accessories	자전거 랙	False	False	328

범주	하위 범주	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	총 수량
Bikes	산악용 자전거	False	False	4970
Clothing		False	True	9101

소계가 여러 개인 예제

```
SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSUBTOTAL ( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSUBTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM( Sales[Qty] )
)
```

매출은 1에 대한 소계와 함께 주별, 고객별, 날짜별로 그룹화됩니다. 주별 매출, 날짜 2 기준. 주별 매출, 고객 3 기준. 주별 매출로 이어지는 고객과 날짜에서 모두 롤업됩니다.

다음 테이블을 반환합니다.

CUSTOMERID	ISCUSTOMERSUBTOTAL	시스템 상태	총 수량	DATE	ISDATESUBTOTAL
A	FALSE	WA	5	7/10/2014	
b	FALSE	WA	1	7/10/2014	
b	FALSE	WA	2	7/11/2014	
C	FALSE	또는	2	7/10/2014	
C	FALSE	또는	1	7/11/2014	
	TRUE	WA	6	7/10/2014	
	TRUE	WA	2	7/11/2014	
	TRUE	또는	2	7/10/2014	
	TRUE	또는	1	7/11/2014	
A	FALSE	WA	5		TRUE
b	FALSE	WA	3		TRUE
C	FALSE	또는	3		TRUE
	TRUE	WA	8		TRUE
	TRUE	또는	3		TRUE

ROLLUPGROUP 사용

SUMMARIZE 함수와 마찬가지로 **ROLLUPGROUP**을 **ROLLUPADDISSUBTOTAL**과 함께 사용하면 포함할 요약 그룹/세분성(소계)을 지정할 수 있어 반환되는 소계 행의 수가 줄어듭니다. **ROLLUPGROUP**은

SUMMARIZECOLUMNS 또는 [SUMMARIZE](#) 식 내에서만 사용할 수 있습니다.

소계가 여러 개인 예제

```
SUMMARIZECOLUMNS(  
    ROLLUPADDISSUBTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),  
    ROLLUPADDISSUBTOTAL(ROLLUPGROUP(Regions[City], Regions[State]), "IsCityStateSubtotal"), "Total Qty", SUM(  
    Sales[Qty] )  
)
```

여전히 도시 및 주별로 그룹화되지만 소계를 보고할 때 함께 롤업되는 경우 다음 테이블을 반환됩니다.

시스템 상태	CUSTOMERID	ISCUSTOMERSUB TOTAL	총 수량	구/군/시	ISCITYSTATESUB TOTAL
WA	A	FALSE	2	Bellevue	FALSE
WA	b	FALSE	2	Bellevue	FALSE
WA	A	FALSE	3	Redmond	FALSE
WA	b	FALSE	1	Redmond	FALSE
또는	C	FALSE	3	Portland	FALSE
WA		TRUE	4	Bellevue	FALSE
WA		TRUE	4	Redmond	FALSE
또는		TRUE	3	Portland	FALSE
	A	FALSE	5		FALSE
	b	FALSE	3		TRUE
	C	FALSE	3		TRUE
		TRUE	11		TRUE

참고 항목

[SUMMARIZE](#)

테이블 생성자

2021-03-16 • 2 minutes to read

하나 이상의 열이 포함된 테이블을 반환합니다.

구문

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ), ... }
```

매개 변수

용어	정의
scalarExprN	스칼라 값을 반환하는 DAX 식입니다.

반환 값

하나 이상의 열이 있는 테이블입니다. 열이 하나만 있는 경우 열 이름은 Value입니다. 열이 N개 있는 경우(여기서 $N > 1$) 열 이름은 왼쪽에서 오른쪽으로 Value1, Value2, ..., ValueN이 됩니다.

설명

- 첫 번째 구문은 단일 열이 있는 테이블을 반환합니다. 두 번째 구문은 하나 이상의 열이 있는 테이블을 반환합니다.
- 스칼라 식의 숫자는 모든 행에 대해 동일해야 합니다.
- 열 값의 데이터 형식이 행마다 다른 경우 모든 값이 공통된 데이터 형식으로 변환됩니다.

예 1

DAX 쿼리는 다음과 같습니다.

```
EVALUATE { 1, 2, 3 }
```

및

```
EVALUATE { (1), (2), (3) }
```

단일 열에 있는 다음 테이블을 반환합니다.

[VALUE]
1
2

[VALUE]
3

예제 2

다음 DAX 쿼리를 실행한다고 가정합니다.

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")
}

그러면 다음 값이 반환됩니다.

[VALUE1]	[VALUE2]	[VALUE3]	[VALUE4]
1.5	2017/1/1	199.99	A
2.5	2017/1/2	249.99	b
3.5	2017/1/3	299.99	C

예제 3

DAX 쿼리는 다음과 같습니다.

EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }

문자열 데이터 형식의 단일 열이 있는 다음 테이블을 반환합니다.

[VALUE]
1
2017/1/1
TRUE
A

TOPN

2021-03-16 • 4 minutes to read

지정된 테이블의 상위 N개 행을 반환합니다.

구문

```
TOPN(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

매개 변수

반환할 행 수입니다. 단일 스칼라 값을 반환하는 DAX 식이며, 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환되는 행 수가 *n_value* 보다 클 수 있는 경우를 이해하려면 설명 섹션을 참조하세요.

빈 테이블이 반환되는 경우를 이해하려면 설명 섹션을 참조하세요.

table 상위 'n'개 행을 추출할 데이터의 테이블을 반환하는 DAX 식입니다.

orderBy_expression

결과 값을 사용하여 테이블을 정렬하고 *table*의 각 행에 대해 결과 값이 계산되는 DAX 식입니다.

order (선택 사항) *orderBy_expression* 값을 정렬하는 방법(오름차순 또는 내림차순)을 지정하는 값입니다.

VALUE	대체 값	설명
0(영)	FALSE	<i>order_by</i> 값을 내림차순으로 정렬합니다. 이는 <i>order</i> 매개 변수가 생략된 경우 기본값입니다.
1	TRUE	<i>order_by</i> 의 오름차순으로 순위를 매깁니다.

반환 값

*table*의 N개 행으로 구성된 테이블이거나 *n_value*가 0(영) 이하인 경우 빈 테이블입니다. 행은 특정 순서로 정렬되지 않을 수도 있습니다.

설명

- 테이블의 N번째 행에서 *order_by* 값의 동률이 있는 경우 모든 동률 행이 반환됩니다. 그런 다음 N번째 행에 동률이 있으면 함수가 n개보다 많은 행을 반환할 수 있습니다.
- n_value*가 0(영) 이하인 경우 TOPN은 빈 테이블을 반환합니다.
- TOPN은 결과에 대한 정렬 순서를 보장하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플에서는 상위 10개 판매 제품의 매출로 측정값을 만듭니다.

```
= SUMX(TOPN(10, SUMMARIZE(Product, [ProductKey], "TotalSales",  
SUMX(RELATED(InternetSales_USD[SalesAmount_USD]), InternetSales_USD[SalesAmount_USD]) +  
SUMX(RELATED(ResellerSales_USD[SalesAmount_USD]), ResellerSales_USD[SalesAmount_USD]))
```

TREATAS

2021-03-16 • 2 minutes to read

관련이 없는 테이블 열의 필터로 테이블 식의 결과를 적용합니다.

구문

```
TREATAS(table_expression, <column>[, <column>[, <column>[,...]]] )
```

매개 변수

용어	정의
table_expression	테이블을 생성하는 식입니다.
열	하나 이상의 기존 열입니다. 식이 될 수 없습니다.

반환 값

table_expression에도 있는, 열의 모든 행이 포함된 테이블입니다.

설명

- 지정된 열 수는 테이블 식의 열 수와 일치해야 하며, 순서도 같아야 합니다.
- 테이블 식에 반환된 값이 열에 없으면 무시됩니다. 예를 들어 TREATAS({"Red", "Green", "Yellow"}, DimProduct[Color])는 "Red", "Green" 및 "Yellow"라는 세 개의 값이 있는 DimProduct[Color] 열에 대한 필터를 설정합니다. DimProduct[Color]에 "Yellow"가 없으면 유효 필터 값은 "Red" 및 "Green"입니다.
- 테이블 간에 관계가 없는 경우 사용하는 것이 가장 좋습니다. 관련 테이블 간에 여러 관계가 있는 경우 [USERELATIONSHIP](#)을 대신 사용하는 것이 좋습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 관련이 없는 제품 테이블 두 개가 모델에 포함되어 있습니다. 사용자가 DimProduct1[ProductCategory]에 필터를 적용하여 자전거, 좌석, 타이어를 선택하는 경우 동일한 필터, 자전거, 좌석, 타이어가 DimProduct2[ProductCategory]에 적용됩니다.

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

참고 항목

[INTERSECT](#)

[FILTER](#)

UNION

2021-03-16 • 3 minutes to read

한 쌍의 테이블에서 통합(조인) 테이블을 만듭니다.

구문

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

매개 변수

용어	정의
table_expression	테이블을 반환하는 DAX 식입니다.

반환 값

두 테이블 식 각각의 행을 모두 포함하는 테이블입니다.

설명

- 두 테이블에 동일한 개수의 열이 있어야 합니다.
- 열은 각 테이블의 위치에 따라 결합됩니다.
- 반환 테이블의 열 이름은 table_expression1의 열 이름과 일치합니다.
- 중복 행은 유지됩니다.
- 반환된 테이블에는 가능한 경우 계보가 있습니다. 예를 들어 각 table_expression의 첫 번째 열에 모델의 동일한 기본 열 C1에 대한 계보가 있는 경우 UNION 결과의 첫 번째 열에 C1에 대한 계보가 포함됩니다. 그러나 결합된 열에 다른 기본 열에 대한 계보가 있거나 확장 열이 있는 경우에는 UNION의 결과 열에 계보가 포함되지 않습니다.
- 데이터 형식이 다른 경우 결과 데이터 형식은 데이터 형식 강제 변환 규칙에 따라 결정됩니다.
- 반환된 테이블에는 관련 테이블의 열이 포함되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 식은 USAInventory 테이블과 INDInventory 테이블을 단일 테이블로 결합 하여 공용 구조체를 만듭니다.

```
UNION(UsaInventory, IndInventory)
```

USAInventory

국가	시스템 상태	개수	합계
USA	CA	5	500
미국	WA	10	900

INDInventory

국가	시스템 상태	개수	합계
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

반환 테이블

국가	시스템 상태	개수	합계
USA	CA	5	500
미국	WA	10	900
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

VALUES

2021-03-16 • 8 minutes to read

입력 매개 변수가 열 이름이면 지정된 열의 고유한 값이 포함된 단일 열 테이블을 반환합니다. 중복 값은 제거되고 고유 값만 반환됩니다. BLANK 값을 추가할 수 있습니다. 입력 매개 변수가 테이블 이름이면 지정된 테이블의 행을 반환합니다. 중복 행은 유지됩니다. BLANK 행을 추가할 수 있습니다.

NOTE

이 함수는 워크시트의 셀 또는 열에 값을 반환하는 데 사용할 수 없습니다. 대신, 수식에 중첩된 중간 함수로 사용하여 다른 값을 필터링하거나 집계하는 데 계산 또는 사용될 수 있는 고유 값 목록을 가져옵니다.

구문

VALUES(<TableNameOrColumnName>)

매개 변수

용어	정의
TableName 또는 ColumnName	반환할 고유 값이 들어 있는 열 또는 반환할 행이 들어 있는 테이블입니다.

반환 값

입력 매개 변수가 열 이름이면 단일 열 테이블을 반환합니다. 입력 매개 변수가 테이블 이름이면 동일한 열 테이블을 반환합니다.

설명

- VALUES 함수를 필터링된 컨텍스트에서 사용하는 경우 VALUES에서 반환하는 고유 값이 해당 필터의 영향을 받습니다. 예를 들어 지역별로 필터링하여 도시 값 목록을 반환하는 경우 필터에서 허용되는 지역의 도시만 해당 목록에 포함됩니다. 기존 필터와 관계없이 도시를 모두 반환하려면 ALL 함수를 사용하여 테이블의 필터를 제거해야 합니다. 두 번째 예제는 VALUES와 함께 ALL을 사용하는 방법을 보여 줍니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 함수

대부분의 시나리오에서는 인수가 열 이름인 경우 VALUES 함수의 결과가 DISTINCT 함수의 결과와 동일합니다. 두 함수 다 중복을 제거하고 지정된 열의 가능한 값 목록을 반환합니다. 하지만 VALUES 함수는 공백 값을 반환할 수도 있습니다. 이 공백 값은 관련 테이블에서 고유 값을 조회하는데 관계에 사용된 값이 한 테이블에서 누락되어 있는 경우 유용합니다. 데이터베이스 용어에서는 참조 무결성 위반이라고 합니다. 이와 같은 데이터의 불일치는 한 테이블은 업데이트되고 관련 테이블은 업데이트되지 않는 경우 발생할 수 있습니다.

인수가 테이블 이름이면 참조 무결성 위반이 있는 경우 VALUES 함수 결과가 지정된 테이블의 모든 행과 공백 행을 반환합니다. DISTINCT 함수는 중복 행을 제거하고 지정된 테이블의 고유 행을 반환합니다.

NOTE

DISTINCT 함수에서는 열 이름 또는 유효한 테이블 식이 인수로 허용되지만, VALUES 함수에서는 열 이름 또는 테이블 이름만 인수로 허용됩니다.

다음 테이블에는 참조 무결성이 유지되지 않는 경우 두 개의 관련 테이블에서 발생할 수 있는 데이터 간 불일치가 요약되어 있습니다.

MYORDERS 테이블	MYSALES 테이블
6월 1일	6월 1일 판매
6월 2일	6월 2일 판매
(주문 날짜를 입력하지 않음)	6월 3일 판매

DISTINCT 함수를 사용하여 이러한 테이블이 포함된 피벗 테이블에서 날짜 목록을 반환하는 경우 두 개의 날짜만 반환됩니다. 하지만 VALUES 함수를 사용하면 두 개의 날짜와 공백 멤버를 추가로 함수에서 반환합니다. 또한 MyOrders 테이블에 있는 날짜와 일치하는 날짜가 없는 MySales 테이블의 모든 행은 이 알 수 없는 멤버에 "일치"됩니다.

예제

다음 수식은 고유 송장(판매 주문) 수를 계산하고 제품 범주 이름이 포함된 보고서에서 사용할 경우 다음과 같은 결과를 생성합니다.

```
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

반환

행 레이블	송장 계산
Accessories	18,208
Bikes	15,205
Clothing	7,461
총합계	27,659

참고 항목

[FILTER 함수](#)

[COUNTROWS 함수](#)

[필터 함수](#)

텍스트 함수

2021-03-16 • 5 minutes to read

DAX(Data Analysis Expressions)에는 Excel의 문자열 함수 라이브러리를 기반으로 하는 텍스트 함수 세트가 포함되어 있지만, 이러한 세트는 테이블 형식 모델의 테이블 및 열을 사용하도록 수정되었습니다. 이 섹션에서는 DAX 언어에서 사용할 수 있는 텍스트 함수에 대해 설명합니다.

이 범주의 함수

함수	DESCRIPTION
COMBINEVALUES	둘 이상의 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다.
CONCATENATE	두 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다.
CONCATENATEX	테이블의 각 행에 대해 계산된 식의 결과를 연결합니다.
EXACT	두 텍스트 문자열을 비교하고 정확히 같으면 TRUE를, 아니면 FALSE를 반환합니다.
FIND	다른 텍스트 문자열 내에서 한 텍스트 문자열의 시작 위치를 반환합니다.
FIXED	숫자를 지정된 소수 자릿수로 반올림하고 결과를 텍스트로 반환합니다.
FORMAT	지정된 형식에 따라 값을 텍스트로 변환합니다.
LEFT	텍스트 문자열의 시작부터 지정된 수만큼의 문자를 반환합니다.
LEN	텍스트 문자열의 문자 수를 반환합니다.
LOWER	텍스트 문자열의 모든 문자를 소문자로 변환합니다.
MID	시작 위치와 길이가 주어질 경우, 텍스트 문자열의 중간에 있는 문자열을 반환합니다.
REPLACE	REPLACE는 지정된 문자 수에 따라 텍스트 문자열의 일부를 다른 텍스트 문자열로 바꿉니다.
REPT	지정된 횟수만큼 텍스트를 반복합니다.
RIGHT	RIGHT는 지정된 문자 수에 따라 텍스트 문자열에 있는 하나 이상의 마지막 문자를 반환합니다.
SEARCH	특정 문자 또는 텍스트 문자열이 처음 발견되는 문자 수를 반환합니다. 이때 왼쪽에서 오른쪽으로 읽습니다.

함수	DESCRIPTION
SUBSTITUTE	기존 텍스트를 텍스트 문자열의 새 텍스트로 바꿉니다.
TRIM	단어 사이의 공백 하나를 제외하고 텍스트에서 모든 공백을 제거합니다.
UNICHAR	숫자 값에서 참조하는 유니코드 문자를 반환합니다.
UNICODE	텍스트 문자열의 첫 번째 문자에 해당하는 숫자 코드를 반환합니다.
UPPER	텍스트 문자열을 모두 대문자로 변환합니다.
VALUE	숫자를 나타내는 텍스트 문자열을 숫자로 변환합니다.

COMBINEVALUES

2021-03-16 • 5 minutes to read

둘 이상의 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다. 이 함수의 주요 목적은 DirectQuery 모델에서 다중 열 관계를 지원하는 것입니다. 자세한 내용은 [설명](#) 을 참조하세요.

구문

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

매개 변수

용어	정의
구분 기호	연결 중에 사용할 구분 기호입니다. 상수 값이어야 합니다.
expression	단일 텍스트 문자열에 조인할 값을 반환하는 DAX 식입니다.

반환 값

연결된 문자열입니다.

설명

- COMBINEVALUES 함수는 입력 값이 다르면 출력 문자열도 다르다고 가정하지만 검증하지는 않습니다. 이 가정에 따라 두 DirectQuery 테이블의 여러 열을 조인하는 관계를 구축하기 위해 COMBINEVALUES를 사용하여 계산 열을 만들면, 쿼리 시 최적화된 조인 조건이 생성됩니다. 예를 들어 Table1(Column1, Column2)과 Table2(Column1, Column2) 간에 관계를 만들려는 사용자는 각 테이블에 하나씩, 두 개의 계산 열을 다음과 같이 만들 수 있습니다.

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

및

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

그런 다음, Table1[CalcColumn] 과 Table2[CalcColumn] 간의 관계를 만듭니다. 해당 SQL 연산자와 함수로 리터럴 변환되는 다른 DAX 함수 및 연산자와 달리, 위의 관계는 SQL 조인 조건자를 다음과 같이 생성합니다.

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

및

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND Table2.Column2 IS NULL) .
```

- 조인 조건자는 복잡한 SQL 연산자 및 함수와 관련된 쿼리보다 훨씬 더 나은 쿼리 성능을 제공할 수 있습니다.
- COMBINEVALUES 함수는 사용자가 적절한 구분 기호를 선택하여 입력 값의 각 조합이 고유한 출력 문자열을 생성하도록 하지만, 가정이 true인지 확인하지는 않습니다. 예를 들어 사용자가 "|" 를 구분 기호로 선택하지만 Table1의 한 행에는 Table1[Column1] = "|" 와 Table2 [Column2] = " " 가 있고 Table2의 한

행에는 `Table2[Column1] = " "` 와 `Table2[Column2] = "| "` 가 있는 경우 두 개의 연결된 출력은 동일한 `"|| "` 가 되어 두 행이 조인 작업에서 일치하는 것처럼 보입니다. 두 테이블을 모두 가져오면 조인되지 만, 두 테이블이 동일한 DirectQuery 소스에 있는 경우에는 두 행이 조인되지 않습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE DISTINCT(SELECTCOLUMNS(DimDate, "Month", COMBINEVALUES(", ", [MonthName], [CalendarYear])))
```

그러면 다음과 같은 단일 열 테이블이 반환됩니다.

[MONTH]
1월,2007
2월,2007
3월,2007
4월,2007
5월,2007
6월,2007
7월,2007
8월,2007
9월,2007
10월,2007
11월,2007
12월,2007
1월,2008
1월,2008
2월,2008
3월,2008
4월,2008
5월,2008
6월,2008
7월,2008

[MONTH]
8월,2008
9월,2008
10월,2008
11월,2008
12월,2008

CONCATENATE

2021-03-16 • 7 minutes to read

두 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다.

구문

```
CONCATENATE(<text1>, <text2>)
```

매개 변수

용어	정의
text1, text2	단일 텍스트 문자열로 조인할 텍스트 문자열입니다. 문자열은 텍스트 또는 숫자를 포함할 수 있습니다. 열 참조를 사용할 수도 있습니다.

반환 값

연결된 문자열입니다.

설명

- CONCATENATE 함수는 두 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다. 조인된 항목은 텍스트로 표시된 텍스트, 숫자 또는 부울 값이거나 이러한 항목의 조합일 수 있습니다. 열에 적절한 값이 포함되어 있으면 열 참조를 사용할 수도 있습니다.
- DAX의 CONCATENATE 함수는 두 개의 인수만 허용하지만 Excel의 CONCATENATE 함수는 최대 255개의 인수를 허용합니다. 여러 개의 열을 연결해야 하는 경우 일련의 계산을 만들 수도 있지만, 연결 연산자(&)를 사용하면 더 간단한 식으로 모든 항목을 조인할 수 있습니다.
- 열 참조를 사용하는 대신 텍스트 문자열을 직접 사용하려면 각 문자열을 큰따옴표로 묶어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제: 리터럴 연결

샘플 수식은 인수로 제공된 두 문자열 값을 결합하여 새 문자열 값을 만듭니다.

```
= CONCATENATE("Hello ", "World")
```

예제: 열에서 문자열 연결

샘플 수식은 전화 번호부에 나열된 고객의 전체 이름을 반환합니다. 중첩된 함수가 두 번째 인수로 사용되는 방식을 확인합니다. 인수로 사용하려는 값이 세 개 이상일 때 여러 문자열을 연결하는 한 가지 방법입니다.

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

예제: 열에서 문자열의 조건부 연결

샘플 수식은 Customer 테이블에 이름, 중간 이니셜, 성의 조합인 전체 고객 이름으로 새 계산 열을 만듭니다. 중간 이름이 없으면 성이 이름 바로 뒤에 옵니다. 중간 이름이 있으면 중간 이름의 첫 문자만 사용하고, 이니셜 문자 뒤에 마침표가 옵니다.

```
= CONCATENATE( [FirstName]&" ", CONCATENATE( IF( LEN([MiddleName])>1, LEFT([MiddleName],1)&" ", ""), [LastName]))
```

이 수식은 앰퍼샌드(&) 연산자와 함께 중첩된 CONCATENATE 및 IF 함수를 사용하여 세 개의 문자열 값을 조건부로 연결하고 공백을 구분 기호로 추가합니다.

예제: 서로 다른 데이터 형식의 열 연결

다음 예제는 서로 다른 데이터 형식의 열 값을 연결하는 방법을 보여 줍니다. 연결하려는 값이 숫자이면 값이 텍스트로 암시적으로 변환됩니다. 두 값이 모두 숫자이면 두 값이 모두 텍스트로 캐스트되어 문자열인 것처럼 연결됩니다.

제품 설명	제품 약어(복합 키의 열 1)	제품 번호(복합 키의 열 2)	새로 생성된 키 열
산악 자전거	MTN	40	MTN40
산악 자전거	MTN	42	MTN42

```
= CONCATENATE('Products'[Product abbreviation],'Products'[Product number])
```

DAX의 CONCATENATE 함수는 두 개의 인수만 허용하지만 Excel의 CONCATENATE 함수는 최대 255개의 인수를 허용합니다. 인수를 더 추가해야 하는 경우 앰퍼샌드(&) 연산자를 사용할 수 있습니다. 예를 들어 다음 수식은 MTN-40 및 MTN-42 결과를 생성합니다.

```
= [Product abbreviation] & "-" & [Product number]
```

참고 항목

[텍스트 함수](#)

CONCATENATEX

2021-03-16 • 2 minutes to read

테이블의 각 행에 대해 계산되는 식의 결과를 연결합니다.

구문

```
CONCATENATEX(<table>, <expression>, [delimiter])
```

매개 변수

용어	정의
테이블	식을 계산할 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
구분 기호	(선택 사항) 연결 중에 사용할 구분 기호입니다.

반환 값

텍스트 문자열입니다.

설명

- 이 함수는 첫 번째 인수로 테이블이나 테이블을 반환하는 식을 사용합니다. 두 번째 인수는 연결하려는 값이나 값을 반환하는 식이 포함된 열입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Employees 테이블

FIRSTNAME	LASTNAME
Alan	Brewer
Michael	Blythe

수식은 다음과 같습니다.

```
CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

HRESULT = NO_ERROR를

"Alan Brewer, Michael Blythe"

EXACT

2021-03-16 • 2 minutes to read

두 텍스트 문자열을 비교하고 정확히 같으면 TRUE를, 아니면 FALSE를 반환합니다. EXACT는 대/소문자를 구분하지만 서식 차이는 무시합니다. EXACT를 사용하여 문서에 입력되는 텍스트를 테스트할 수 있습니다.

구문

```
EXACT(<text1>,<text2>)
```

매개 변수

용어	정의
text1	텍스트를 포함하는 첫 번째 텍스트 문자열 또는 열입니다.
text2	텍스트를 포함하는 두 번째 텍스트 문자열 또는 열입니다.

반환 값

True 또는 False입니다. (부울)

예제

다음 수식은 현재 행에 대한 Column2 값과 비교하여 현재 행에 대한 Column1 값을 확인하고 같으면 TRUE를 반환하고, 다르면 FALSE를 반환합니다.

```
= EXACT([Column1],[Column2])
```

참고 항목

[텍스트 함수](#)

FIND

2021-03-16 • 3 minutes to read

다른 텍스트 문자열 내에서 한 텍스트 문자열의 시작 위치를 반환합니다. FIND는 대/소문자를 구분합니다.

구문

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

매개 변수

용어	정의
find_text	찾으려는 텍스트입니다. within_text 의 첫 번째 문자를 일치시키려면 큰따옴표(빈 텍스트)를 사용합니다.
within_text	찾으려는 텍스트를 포함하는 텍스트입니다.
start_num	(선택 사항) 검색을 시작할 문자입니다. 생략하는 경우 start_num = 1이 됩니다. within_text 의 첫 번째 문자는 문자 번호 1입니다.
NotFoundValue	(선택 사항) 작업에서 일치하는 부분 문자열을 찾을 수 없는 경우에 반환되는 값으로, 일반적으로 0, -1 또는 BLANK()입니다.

반환 값

찾으려는 텍스트 문자열의 시작 지점을 표시하는 숫자입니다.

설명

- Microsoft Excel에는 SBCS(싱글바이트 문자 집합) 및 DBCS(더블 바이트 문자 집합) 언어를 수용하기 위한 여러 버전의 FIND 함수가 있지만 DAX에서는 유니코드를 사용하고 각 문자를 동일한 방식으로 계산합니다. 따라서 문자 형식에 따라 다른 버전을 사용할 필요가 없습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.
- FIND는 와일드카드를 지원하지 않습니다. 와일드카드를 사용하려면 [SEARCH](#)를 사용합니다.

예제

다음 수식에서는 제품 설명이 포함된 문자열에서 제품 지정인 BMX의 첫 글자 위치를 찾습니다.

```
= FIND("BMX","line of BMX racing goods")
```

참고 항목

FIXED

2021-03-16 • 3 minutes to read

숫자를 지정된 소수 자릿수로 반올림하고 결과를 텍스트로 반환합니다. 결과를 쉼표를 포함하거나 포함하지 않고 반환하도록 지정할 수 있습니다.

구문

```
FIXED(<number>, <decimals>, <no_commas>)
```

매개 변수

용어	정의
number	반올림하고 텍스트로 변환할 숫자 또는 숫자가 포함된 열입니다.
decimals	(선택 사항) 소수점 이하 자릿수입니다. 생략하면 2가 됩니다.
no_commas	(선택 사항) 논리값입니다. 1이면 반환되는 텍스트에 쉼표를 표시하지 않고, 0이거나 생략하면 반환되는 텍스트에 쉼표를 표시합니다.

반환 값

텍스트로 표현되는 숫자입니다.

설명

- **decimals** 매개 변수에 사용된 값이 음수이면 **number** 의 소수점 이하는 반올림됩니다.
- 소수 자릿수 를 생략하면 2로 간주됩니다.
- **no_commas** 가 0이거나 생략된 경우 반환되는 텍스트에는 일반적으로 쉼표가 포함됩니다.
- 명령을 사용하여 숫자가 포함된 셀의 서식을 지정할 때와 FIXED 함수를 사용하여 숫자의 형식을 지정할 때의 주요 차이점은 FIXED는 결과를 텍스트로 변환한다는 점입니다. 서식 지정 메뉴의 명령을 사용하여 서식이 지정된 숫자는 여전히 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 PctCost 열의 현재 행에 대한 숫자 값을 가져와 소수 자릿수가 4자리인 텍스트로 쉼표 없이 반환합니다.

```
= FIXED([PctCost],3,1)
```

숫자는 유효 자릿수가 15자리를 넘을 수 없지만, 소수 자릿수는 127자리까지 가능합니다.

참고 항목

[텍스트 함수](#)

[수학 및 삼각 함수](#)

FORMAT

2021-03-16 • 39 minutes to read

지정된 형식에 따라 값을 텍스트로 변환합니다.

구문

```
FORMAT(<value>, <format_string>)
```

매개 변수

용어	정의
값	단일 값으로 계산되는 값 또는 식입니다.
format_string	서식 지정 템플릿이 포함된 문자열입니다.

반환 값

format_string 에서 정의한 대로 형식이 지정된 **값** 을 포함하는 문자열입니다.

NOTE

value 가 BLANK인 경우 함수는 빈 문자열을 반환합니다.

format_string 이 BLANK인 경우 값은 "일반 숫자" 또는 "일반 날짜" 형식으로 지정됩니다(**value** 형식에 따라).

설명

- 미리 정의된 형식 문자열은 결과의 형식을 지정할 때 모델 culture 속성을 사용합니다. 기본적으로 모델 culture 속성은 컴퓨터의 사용자 로캘에 따라 설정됩니다. 새 Power BI Desktop 모델의 경우 모델 culture 속성은 옵션 > 국가별 설정 > 모델 언어에서 변경할 수 있습니다. Analysis Services의 경우 모델 culture 은 처음에 인스턴스에 의해 정의된 Language 속성에 따라 설정됩니다.
- DAX FORMAT 함수의 인수로 지원되는 형식 문자열은 .NET Framework에서 사용되는 형식 문자열이 아닌 Visual Basic(OLE Automation)에서 사용되는 형식 문자열을 기반으로 합니다. 따라서 인수가 정의된 형식 문자열과 일치하지 않으면 예기치 않은 결과나 오류가 발생할 수 있습니다. 예를 들어 "Percent"의 약어로 "p"는 지원되지 않습니다. 미리 정의된 형식 문자열 목록에 포함되지 않은 FORMAT 함수에 인수 로 제공되는 문자열은 사용자 지정 형식 문자열의 일부로 또는 문자열 리터럴로 처리됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예

```
= 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")
```

그러면 다음 값이 반환됩니다.

12345.67 "일반 숫자"는 형식 없이 숫자를 표시합니다.

\$12,345.67 "통화"는 통화 로캘 형식으로 숫자를 표시합니다. 이 샘플은 기본 미국 통화 형식을 보여 줍니다.

12345.67 "고정"은 소수 구분 기호 왼쪽에 한 자릿수 이상, 소수 구분 기호 오른쪽에 두 자릿수를 표시합니다.

12,345.67 "표준"은 소수 구분 기호 왼쪽에 한 자릿수 이상, 소수 구분 기호 오른쪽에 두 자릿수 이상을 표시하고 천 단위 구분 기호를 포함합니다. 이 샘플은 기본 미국 숫자 형식을 보여 줍니다.

1,234,567.00 % "퍼센트"는 숫자에 100을 곱하고 형식을 지정한 다음, 퍼센트 기호를 공백 하나로 구분해서 숫자 오른쪽에 추가하여 숫자를 백분율로 표시합니다.

1.23E+04 "과학적"은 숫자를 소수 두 자릿수의 과학적 표기법으로 표시합니다.

미리 정의된 숫자 형식

format_string 인수에서 다음과 같이 미리 정의된 숫자 형식을 지정할 수 있습니다.

형식	DESCRIPTION
"General Number"	천 단위 구분 기호 없이 숫자를 표시합니다.
"Currency"	해당하는 경우 천 단위 구분 기호를 사용하여 숫자를 표시합니다. 소수 구분 기호 오른쪽에 두 자릿수를 표시합니다. 출력은 시스템 로캘 설정에 따릅니다.
"Fixed"	소수 구분 기호 왼쪽으로 최소 한 자릿수, 오른쪽으로 두 자릿수를 표시합니다.
"Standard"	천 단위 구분 기호를 사용하여 숫자를 표시하고 소수 구분 기호 왼쪽에 한 자릿수 이상, 오른쪽에 두 자릿수를 표시합니다.
"Percent"	숫자에 100을 곱하고, 바로 오른쪽에 퍼센트 기호(%)를 추가하여 표시합니다. 소수 구분 기호 오른쪽에 항상 두 자릿수를 표시합니다.
"Scientific"	표준 과학적 표기법을 사용하고 유효 자릿수 2자리를 제공합니다.
"Yes/No"	숫자가 0이면 No를 표시하고, 0이 아니면 Yes를 표시합니다.
"True/False"	숫자가 0이면 False를 표시하고, 0이 아니면 True를 표시합니다.
"On/Off"	숫자가 0이면 Off를 표시하고, 0이 아니면 On을 표시합니다.

사용자 지정 숫자 형식

숫자에 대한 사용자 정의 형식 식은 세미콜론으로 구분된 1~3개의 섹션을 포함할 수 있습니다. 형식 문자열 인수가 명명된 숫자 형식 중 한 개만 포함하는 경우에는 한 개의 섹션만 사용할 수 있습니다.

사용	결과
섹션 1개만	형식 식이 모든 값에 적용됩니다.
두 섹션	첫 번째 절은 양수 값 및 0에 두 번째 절은 음수 값에 적용됩니다.
세 섹션	첫 번째 절은 양수 값에 두 번째 절은 음수 값에 세 번째 절은 0에 적용됩니다.

```
"$#,##0;($#,##0)"
```

세미콜론 사이에 아무것도 포함하지 않으면 누락된 섹션이 양수 값 형식을 사용하여 정의됩니다. 예를 들어 다음 형식은 첫 번째 섹션의 형식을 사용하여 양수 값 및 음수 값을 표시하고 값이 0이면 "Zero"를 표시합니다.

```
"$#,##0"
```

세미콜론 사이에 아무것도 포함하지 않으면 누락된 섹션이 양수 값 형식을 사용하여 표시됩니다.

사용자 지정 숫자 형식 문자

format_string 인수에서 다음과 같은 사용자 지정 숫자 형식 문자를 지정할 수 있습니다.

문자	DESCRIPTION
없음	형식을 지정하지 않고 숫자를 표시합니다.
(0)	숫자 자리 표시자. 숫자 또는 0을 표시합니다. 형식 문자열에서 0이 나타나는 식의 위치에 숫자가 있으면 해당 숫자를 표시하고, 그렇지 않으면 해당 위치에 0을 표시합니다. 숫자의 자릿수가 형식 식에 있는 0의 개수(소수점의 한쪽에 있는)보다 적은 경우 선행 또는 후행 0을 표시합니다. 소수 구분 기호 오른쪽에 있는 숫자의 자릿수가 형식 문자열에서 소수 구분 기호 오른쪽에 있는 0의 개수보다 많으면 소수 자릿수가 0 개수가 되도록 숫자를 반올림합니다. 소수 구분 기호 왼쪽에 있는 숫자의 자릿수가 형식 문자열에서 소수 구분 기호 왼쪽에 있는 0의 개수보다 많으면 수정하지 않고 남은 자릿수를 표시합니다.
(#)	숫자 자리 표시자. 숫자를 표시하거나 아무 것도 표시하지 않습니다. 형식 문자열에서 #이 나타나는 식의 위치에 숫자가 있으면 해당 숫자를 표시하고, 그렇지 않으면 해당 위치에 아무 것도 표시하지 않습니다. 이 기호는 0 숫자 자리 표시자와 동일하게 작동합니다. 단, 숫자의 자릿수가 형식 식의 소수 구분 기호 양쪽에 있는 # 문자 수보다 적거나 같을 때 선행 및 후행 0이 표시되지 않습니다.

문자	DESCRIPTION
(.)	소수 자리 표시자입니다. 일부 로캘에서는 소수 구분 기호로 쉼표를 사용합니다. 소수 자리 표시자는 소수 구분 기호 왼쪽 및 오른쪽에 표시되는 자릿수를 결정합니다. 형식 식에 이 기호 왼쪽에 있는 숫자 기호만 포함된 경우 1보다 작은 숫자는 소수 구분 기호로 시작합니다. 소수를 사용하여 선행 0을 표시하려면 소수 구분 기호 왼쪽의 첫 번째 숫자 자리 표시자로 0을 사용합니다. 형식이 지정된 출력에서 소수 자리 표시자로 사용되는 실제 문자는 시스템에서 인식하는 숫자 형식에 따라 다릅니다.
(%)	백분을 자리 표시자. 식에 100을 곱합니다. 퍼센트 문자(%)가 형식 문자열에서 나타나는 위치에 삽입됩니다.
(,)	천 단위 구분 기호. 일부 로캘에서는 천 단위 구분 기호로 마침표를 사용합니다. 천 단위 구분 기호는 소수 구분 기호 왼쪽에 네 개 이상의 자릿수가 있는 숫자 내에서 천 단위를 구분합니다. 숫자 자리 표시자(0 또는 #)로 묶인 천 단위 구분 기호가 형식에 포함되어 있으면 천 단위 구분 기호의 표준 사용이 지정됩니다. 소수 구분 기호 바로 왼쪽에 있는 2개의 인접한 천 단위 구분 기호나 천 단위 구분 기호(소수 구분 기호 지정 여부와 관계 없음)는 “숫자를 1000으로 나눈 후 필요에 따라 반올림하는 것”을 의미합니다. 예를 들어, 형식 문자열 “##0,”을 사용하여 1억을 100으로 나타낼 수 있습니다. 100만보다 작은 숫자는 0으로 표시됩니다. 소수 구분 기호 바로 왼쪽이 아닌 위치에 있는 두 개의 인접한 천 단위 구분 기호는 천 단위 구분 기호 사용을 지정하여 간단히 처리됩니다. 형식이 지정된 출력에서 천 단위 구분 기호로 사용되는 실제 문자는 시스템에서 인식하는 숫자 형식에 따라 다릅니다.
(:)	시간 구분 기호. 일부 로캘에서는 다른 문자를 사용하여 시간 구분 기호를 나타낼 수 있습니다. 시간 구분 기호는 시간 값의 형식을 지정할 때 시간, 분 및 초를 구분합니다. 형식이 지정된 출력에서 시간 구분 기호로 사용되는 실제 문자는 시스템 설정에 따라 결정됩니다.
(/)	날짜 구분 기호. 일부 로캘에서는 다른 문자를 사용하여 날짜 구분 기호를 나타낼 수 있습니다. 날짜 구분 기호는 날짜 값의 형식을 지정할 때 일, 월 및 년을 구분합니다. 형식이 지정된 출력에서 날짜 구분 기호로 사용되는 실제 문자는 시스템 설정에 따라 결정됩니다.
(E- E+ e- e+)	지수 형식. 형식 식에서 E-, E+, e- 또는 e+ 오른쪽에 한 자리 이상의 자리 표시자(0 또는 #)가 포함되어 있으면 숫자는 지수 형식으로 표시되고 E 또는 e가 숫자와 지수 사이에 삽입됩니다. 오른쪽의 숫자 자리 표시자의 수는 지수의 자릿수를 결정합니다. 음수 지수 옆에 빼기 기호를 추가하려면 E- 또는 e-를 사용합니다. 음수 지수 옆에 빼기 기호를, 양수 지수 옆에 더하기 기호를 추가하려면 E+ 또는 e+를 사용합니다.
- + \$ ()	리터럴 문자를 표시합니다. 나열된 문자가 아닌 다른 문자를 표시하려면 앞에 백슬래시(\)를 사용하거나 큰따옴표(“)로 묶습니다.

문자	DESCRIPTION
(\)	형식 문자열에서 다음 문자를 표시합니다. 특별한 의미가 있는 문자를 리터럴 문자로 표시하려면 앞에 백슬래시(\)를 사용합니다. 백슬래시 자체는 표시되지 않습니다. 백슬래시를 사용하는 것은 다음 문자를 큰따옴표로 묶는 것과 같습니다. 백슬래시를 표시하려면 백슬래시를 두 개(\\) 사용합니다. 리터럴 문자로 표시할 수 없는 문자의 예로 날짜 형식 및 시간 형식 문자(a, c, d, h, m, n, p, q, s, t, w, y, /, :), 숫자 형식 문자(#, 0, %, E, e, 쉼표, 마침표), 문자열 형식 문자(@, &, <, >, !) 등이 있습니다.
("ABC")	문자열을 큰따옴표(" ")로 묶어 표시합니다.

미리 정의된 날짜/시간 형식

`format_string` 인수에서 다음과 같이 미리 정의된 날짜/시간 형식을 지정할 수 있습니다. 이러한 형식 이외의 형식을 사용하는 경우 사용자 지정 날짜/시간 형식으로 해석됩니다.

형식	DESCRIPTION
"General Date"	날짜 및/또는 시간을 표시합니다. 예를 들어 2008/3/12 오전 11:07:31이 표시됩니다. 날짜 표시는 애플리케이션의 현재 문화권 값에 따라 결정됩니다.
"Long Date" 또는 "Medium Date"	현재 문화권의 긴 날짜 형식에 따라 날짜를 표시합니다. 예를 들어 2008년 3월 12일 수요일이 표시됩니다.
"Short Date"	현재 문화권의 짧은 날짜 형식을 사용하여 날짜를 표시합니다. 예를 들어 2008/3/12가 표시됩니다.
"Long Time" 또는	현재 문화권의 긴 시간 형식을 사용하여 시간을 표시합니다. 일반적으로 시, 분, 초를 포함합니다. 예를 들어 오전 11:07:31이 표시됩니다.
"Medium Time"	12시간 형식으로 시간을 표시합니다. 예를 들어 오전 11:07이 표시됩니다.
"Short Time"	24시간 형식으로 시간을 표시합니다. 예를 들어 11:07이 표시됩니다.

사용자 지정 날짜/시간 형식

`format_string`에서 다음과 같은 형식 문자를 지정하여 사용자 지정 날짜/시간 형식을 만들 수 있습니다.

문자	DESCRIPTION
(:)	시간 구분 기호. 일부 로캘에서는 다른 문자를 사용하여 시간 구분 기호를 나타낼 수 있습니다. 시간 구분 기호는 시간 값의 형식을 지정할 때 시간, 분 및 초를 구분합니다. 형식이 지정된 출력에서 시간 구분 기호로 사용되는 실제 문자는 시스템 설정에 따라 결정됩니다.

문자	DESCRIPTION
(/)	날짜 구분 기호. 일부 로캘에서는 다른 문자를 사용하여 날짜 구분 기호를 나타낼 수 있습니다. 날짜 구분 기호는 날짜 값의 형식을 지정할 때 일, 월 및 년을 구분합니다. 형식이 지정된 출력에서 날짜 구분 기호로 사용되는 실제 문자는 시스템 설정에 따라 결정됩니다.
(\)	백슬래시입니다. 다음 문자를 리터럴 문자로 표시합니다. 따라서 형식 문자로 해석되지 않습니다.
(")	큰따옴표입니다. 큰따옴표로 묶인 텍스트가 표시됩니다. 따라서 형식 문자로 해석되지 않습니다.
c	날짜를 <code>dddd</code> 형식으로 표시하고 시간을 <code>ttttt</code> 형식으로 순서대로 표시합니다. 날짜 일련 번호에 소수 부분이 없으면 날짜 정보만 표시합니다. 정수 부분이 없으면 시간 정보만 표시합니다.
일	일을 앞에 오는 0 없이 숫자로 표시합니다(1~31).
dd	일을 앞에 오는 0을 사용하여 숫자로 표시합니다(01~31).
ddd	약어로 요일을 표시합니다(Sun~Sat). 지역화됨.
dddd	요일을 전체 이름으로 표시합니다(Sunday~Saturday). 지역화됨.
dddddd	날짜를 시스템의 간단한 날짜 형식 설정에 따라 완전한 형태(년, 월, 일 포함)로 표시합니다. 기본 간단한 날짜 형식은 <code>mm/dd/yyyy</code> 입니다.
dddddd	날짜 일련 번호를 시스템에서 인식하는 자세한 날짜 설정에 따라 완전한 형태(년, 월, 일 포함)로 표시합니다. 기본 긴 날짜 형식은 <code>dddd, mmmm d, yyyy</code> 입니다.
w	요일을 숫자로 표시합니다(일요일은 1~토요일은 7).
ww	연중 각 주를 번호로 표시합니다(1~54).
분	월을 앞에 오는 0 없이 숫자로 표시합니다(1~12). <code>h</code> 또는 <code>hh</code> 바로 뒤에 <code>m</code> 이 오면 월이 아닌 분이 표시됩니다.
MM	월을 앞에 오는 0을 사용하는 숫자로 표시합니다(01~12). <code>h</code> 또는 <code>hh</code> 바로 뒤에 <code>mm</code> 이 오면 월이 아닌 분이 표시됩니다.
mmm	약어로 월을 표시합니다(Jan~Dec). 지역화됨.
mmmm	월을 전체 월 이름으로 표시합니다(January~December). 지역화됨.
q	연중 분기를 번호로 표시합니다(1~4).
y	연중 날짜를 숫자로 표시합니다(1~366).

문자	DESCRIPTION
yy	연도를 2자리 숫자로 표시합니다(00~99).
yyyy	연도를 4자리 숫자로 표시합니다(100~9999).
h	시간을 앞에 오는 0 없이 숫자로 표시합니다(0~23).
hh	시간을 앞에 오는 0을 사용하는 숫자로 표시합니다(00~23).
n	분을 앞에 오는 0 없이 숫자로 표시합니다(0~59).
nn	분을 앞에 오는 0을 사용하는 숫자로 표시합니다(00~59).
초	초를 앞에 오는 0 없이 숫자로 표시합니다(0~59).
ss	초를 앞에 오는 0을 사용하는 숫자로 표시합니다(00~59).
ttttt	시스템에서 인식하는 시간 형식으로 정의된 시간 구분 기호를 사용하여 완전한 형태로(시, 분, 초 포함) 시간을 표시합니다. 앞에 오는 0 옵션이 선택되고 시간이 오전 10:00 이전인 경우 앞에 0이 표시됩니다. 시간 범위 기본 시간 형식은 <code>h:mm:ss</code> 입니다.
AM/PM	12시간 시계를 사용하고 정오 이전의 모든 시간에 대문자 AM을 표시하고, 정오와 오후 11시 59분 사이의 모든 시간에 대문자 PM을 표시합니다.
AM/PM	12시간제를 사용하며 정오 이전의 모든 시간에 소문자 AM을 표시하고 정오와 오후 11시 59분 사이의 모든 시간에 소문자 PM을 표시합니다.
A/P	12시간제를 사용하며 정오 이전의 모든 시간에 대문자 A를 표시하고 정오와 오후 11시 59분 사이의 모든 시간에 대문자 P를 표시합니다.
A/P	12시간제를 사용하며 정오 이전의 모든 시간에 소문자 A를 표시하고 정오와 오후 11시 59분 사이의 모든 시간에 소문자 P를 표시합니다.
AMPM	12시간제를 사용하며 정오 이전의 모든 시간에 시스템에 정의된 AM 문자열 리터럴을 표시하고 정오와 오후 11시 59분 사이의 모든 시간에 시스템에 정의된 PM 문자열 리터럴을 표시합니다. AMPM은 대문자이거나 소문자일 수 있지만 표시된 문자열의 대/소문자는 시스템 설정에 정의된 문자열과 일치합니다. 기본 형식은 AM/PM입니다. 시스템이 24시간제로 설정된 경우 일반적으로 문자열은 빈 문자열로 설정됩니다.

날짜/시간 형식 지정에서는 현재 사용자 로캘을 사용하여 문자열의 형식을 지정합니다. 예를 들어 날짜 June 25, 2020을 생각해 보세요. 서식 문자열 "m/d/yyyy"를 사용하여 형식이 지정되면 다음과 같이 표시됩니다.

- 사용자 로캘이 미국(en-US)인 경우 "6/25/2020"
- 사용자 로캘이 독일(de-DE)인 경우 "6.25.2020"

사용자 지정 날짜/시간 형식 예제

다음 예에서는 날짜/시간 Thursday, June 25, 2020, at 1:23:45 PM을 사용합니다. 독일(de-DE)은 24시간제를 사

용하므로 AM/PM에 상응하는 것이 없습니다.

형식	결과(EN-US)	결과(DE-DE)
"c"	06/25/2020 13:23:45	25.06.2020 13:23:45
"d"	25	25
"dd"	25	25
"ddd"	Thu	수행
"dddd"	목요일	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"	6월	6월
"mmm"	6월	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
"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

형식	결과(EN-US)	결과(DE-DE)
"hh:nn:ss"	13:23:45	13:23:45
"hh:nn:ss AMPM"	01:23:45 PM	01:23:45
"ttttt"	13:23:45	13:23:45
"ttttt 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

LEFT

2021-03-16 • 2 minutes to read

텍스트 문자열의 시작부터 지정한 수만큼의 문자를 반환합니다.

구문

```
LEFT(<text>, <num_chars>)
```

매개 변수

용어	정의
text	추출할 문자를 포함하는 텍스트 문자열이나 텍스트가 포함된 열에 대한 참조입니다.
num_chars	(선택 사항) LEFT에서 추출할 문자 수입니다. 생략하면 1입니다.

반환 값

텍스트 문자열입니다.

설명

- Microsoft Excel에는 싱글바이트와 더블바이트 문자 언어의 텍스트 작업을 위한 다른 함수가 포함되어 있지만 DAX는 유니코드로 작동하고 모든 문자를 같은 길이로 저장하므로 단일 함수로 충분합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 예제에서는 [ResellerName] 열에 있는 회사 이름의 처음 5자와 [GeographyKey] 열에 있는 지리 코드의 처음 5자를 반환하고 이들을 연결하여 식별자를 만듭니다.

```
= CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))
```

num_chars 인수가 사용 가능한 문자 수보다 큰 경우 이 함수는 사용할 수 있는 최대 문자 수를 반환하고 오류를 발생시키지 않습니다. 예를 들어 [GeographyKey] 열에는 1, 12 및 311과 같은 숫자가 포함되므로 결과도 가변 길이입니다.

참고 항목

[텍스트 함수](#)

LEN

2021-03-16 • 2 minutes to read

텍스트 문자열의 문자 수를 반환합니다.

구문

```
LEN(<text>)
```

매개 변수

용어	정의
text	길이를 찾을 텍스트이거나 텍스트를 포함하는 열입니다. 공백은 문자로 계산됩니다.

반환 값

텍스트 문자열의 문자 수를 나타내는 정수입니다.

설명

- Microsoft Excel에는 싱글바이트와 더블바이트 문자 언어의 텍스트 작업을 위한 다른 함수가 있지만 DAX는 유니코드로 작동하고 모든 문자를 같은 길이로 저장합니다.
- LEN은 기본 언어 설정과 관계없이 항상 각 문자를 1로 계산합니다.
- 날짜 또는 부울과 같이 텍스트가 아닌 값이 포함된 열에 LEN을 사용하는 경우 이 함수는 현재 열 형식을 사용하여 값을 텍스트로 암시적으로 캐스트합니다.

예제

다음 수식은 [AddressLine1] 및 [AddressLine2] 열의 주소 길이를 합합니다.

```
= LEN([AddressLine1])+LEN([AddressLin2])
```

LOWER

2021-03-16 • 2 minutes to read

텍스트 문자열의 모든 문자를 소문자로 변환합니다.

구문

```
LOWER(<text>)
```

매개 변수

용어	정의
text	소문자로 변환하려는 텍스트 또는 텍스트가 포함된 열에 대한 참조입니다.

반환 값

소문자로 표기된 텍스트입니다.

설명

글자가 아닌 문자는 변경되지 않습니다. 예를 들어 `= LOWER("123ABC")` 수식은 **123abc** 를 반환합니다.

예제

다음 수식은 [ProductCode] 열의 각 행을 가져오고 값을 모두 소문자로 변환합니다. 열의 숫자는 영향을 받지 않습니다.

```
= LOWER('New Products'[ProductCode])
```

참고 항목

[텍스트 함수](#)

MID

2021-03-16 • 2 minutes to read

텍스트 문자열의 지정된 위치부터 지정된 수만큼의 문자를 반환합니다.

구문

```
MID(<text>, <start_num>, <num_chars>)
```

매개 변수

용어	정의
text	문자를 추출할 텍스트 문자열이나 텍스트가 포함된 열입니다.
start_num	추출하려는 첫 번째 문자의 위치입니다. 시작 위치는 1입니다.
num_chars	반환할 문자 수입니다.

반환 값

지정된 길이의 텍스트 문자열입니다.

설명

Microsoft Excel에는 싱글바이트와 더블바이트 문자 언어에 사용되는 함수가 각각 제공되지만, DAX는 유니코드를 사용하며 모든 문자를 동일한 길이로 저장합니다.

예

다음 식은

```
MID("abcde",2,3))
```

"bcd"를 반환합니다.

다음 식은

```
MID('Reseller'[ResellerName],1,5))
```

`LEFT([ResellerName],5)`과 같은 결과를 반환합니다. 두 식은 모두 `[ResellerName]` 열의 처음 5개 문자를 반환합니다.

참고 항목

[텍스트 함수](#)

REPLACE

2021-03-16 • 2 minutes to read

REPLACE는 지정된 문자 수에 따라 텍스트 문자열의 일부를 다른 텍스트 문자열로 바꿉니다.

구문

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

매개 변수

용어	정의
old_text	바꾸려는 문자가 포함된 텍스트 문자열이거나 텍스트가 포함된 열에 대한 참조입니다.
start_num	old_text 에서 new_text 로 바꾸려는 문자의 위치입니다.
num_chars	바꾸려는 문자 수입니다. 경고: <i>num_chars</i> 인수가 공백이거나 공백으로 계산되는 열을 참조하는 경우, 문자를 바꾸지 않고 <i>new_text</i> 의 문자열이 <i>start_num</i> 위치에 삽입됩니다. Excel에서와 동일한 동작입니다.
new_text	old_text 에 지정된 문자의 대체 텍스트입니다.

반환 값

텍스트 문자열입니다.

설명

- Microsoft Excel에는 싱글바이트와 더블바이트 문자 언어에 사용되는 함수가 각각 제공되지만, DAX는 유니코드를 사용하므로 모든 문자를 동일한 길이로 저장합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 수식은 [ProductCode] 열에 있는 제품 코드의 처음 두 문자를 두 문자로 된 새 코드 OB로 바꾸는 새 계산 열을 만듭니다.

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

참고 항목

[텍스트 함수](#)

[SUBSTITUTE 함수](#)

REPT

2021-03-16 • 2 minutes to read

지정된 횟수만큼 텍스트를 반복합니다. 셀을 텍스트 문자열의 여러 인스턴스로 채우려면 REPT를 사용합니다.

구문

```
REPT(<text>, <num_times>)
```

매개 변수

용어	정의
텍스트	반복할 텍스트입니다.
num_times	텍스트 반복 횟수를 지정하는 양수입니다.

반환 값

변경 내용이 포함된 문자열입니다.

설명

- **number_times** 가 0인 경우 REPT는 빈 값을 반환합니다.
- **number_times** 가 정수가 아니면 잘립니다.
- REPT 함수의 결과는 32,767자를 초과할 수 없으며, 초과하는 경우 REPT는 오류를 반환합니다.

예제: 리터럴 문자열 반복

다음 예에서는 문자열 85를 세 번 반복합니다.

```
= REPT("85",3)
```

예제: 열 값 반복

다음 예에서는 [MyText] 열의 문자열을 [MyNumber] 열의 횟수만큼 반복해서 반환합니다. 수식이 전체 열에 대해 확장되기 때문에 결과 문자열은 각 행의 텍스트 및 숫자 값에 따라 달라집니다.

```
= REPT([MyText],[MyNumber])
```

MYTEXT	MYNUMBER	CALCULATEDCOLUMN1
텍스트	2	TextText
번호	0	

MYTEXT	MYNUMBER	CALCULATEDCOLUMN1
85	3	858585

참고 항목

[텍스트 함수](#)

RIGHT

2021-03-16 • 3 minutes to read

RIGHT는 지정된 문자 수에 따라 텍스트 문자열에 있는 하나 이상의 마지막 문자를 반환합니다.

구문

```
RIGHT(<text>, <num_chars>)
```

매개 변수

용어	정의
텍스트	추출할 문자가 포함된 텍스트 문자열이거나 텍스트가 포함된 열에 대한 참조입니다.
num_chars	(선택 사항) RIGHT에서 추출할 문자 수입니다. 생략할 경우 1입니다. 숫자가 포함된 열에 대한 참조를 사용할 수도 있습니다.

열 참조에 텍스트가 포함되지 않은 경우 텍스트로 암시적으로 캐스팅됩니다.

반환 값

지정된 가장 오른쪽 문자가 포함된 텍스트 문자열입니다.

설명

- RIGHT는 기본 언어 설정이 무엇이든 관계없이 문자가 싱글바이트이거나 더블바이트이거나 항상 각 문자를 1로 계산합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제: 고정된 수의 문자 반환

다음 수식은 New Products 테이블에 있는 제품 코드의 마지막 두 자리 숫자를 반환합니다.

```
= RIGHT('New Products'[ProductCode],2)
```

예제: 열 참조를 사용하여 문자 수 지정

다음 수식은 MyCount 열의 숫자에 따라 해당 개수만큼 New Products 테이블에 있는 제품 코드의 숫자를 반환합니다. MyCount 열에 값이 없거나 값이 공백인 경우 RIGHT는 공백도 반환합니다.

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

참고 항목

텍스트 함수

LEFT

MID

SEARCH

2021-03-16 • 7 minutes to read

왼쪽에서 오른쪽으로 읽어 특정 문자 또는 텍스트 문자열이 처음 발견되는 문자 번호를 반환합니다. 검색은 대/소문자를 구분하지 않고 악센트를 구분합니다.

구문

```
SEARCH(<find_text>, <within_text>[, [<start_num>]][, <NotFoundValue>]])
```

매개 변수

용어	정의
find_text	찾으려는 텍스트입니다. find_text 에 물음표(?), 별표(*) 등의 와일드카드 문자를 사용할 수 있습니다. 물음표는 모든 단일 문자를 나타내고, 별표는 모든 문자 시퀀스를 나타냅니다. 실제 물음표나 별표를 찾으려면 문자 앞에 물결표(~)를 입력합니다.
within_text	find_text 를 검색하려는 텍스트이거나 텍스트가 포함된 열입니다.
start_num	(선택 사항) within_text 에서 검색을 시작하려는 문자 위치입니다. 생략하면 1이 됩니다.
NotFoundValue	(선택 사항) 작업에서 일치하는 부분 문자열을 찾을 수 없는 경우에 반환되는 값으로, 일반적으로 0, -1 또는 BLANK()입니다.

반환 값

두 번째 텍스트 문자열의 첫 번째 문자를 기준으로 하는 첫 번째 텍스트 문자열의 시작 위치 번호입니다.

설명

- search 함수는 대/소문자를 구분하지 않습니다. "N"을 검색하면 'N'이나 'n'의 첫 번째 항목을 찾습니다.
- search 함수는 악센트를 구분합니다. "á"를 검색하면 'á'의 첫 번째 항목을 찾지만 'a', 'à' 또는 대문자 버전인 'A', 'Á' 항목은 찾지 않습니다.
- 이 함수를 사용하면 두 번째 텍스트 문자열에서 한 텍스트 문자열을 찾은 다음, 첫 번째 문자열의 시작 위치를 반환할 수 있습니다.
- SEARCH 함수를 사용하여 다른 텍스트 문자열에서 한 문자나 텍스트 문자열의 위치를 확인한 다음, MID 함수를 사용하여 텍스트를 반환하거나 REPLACE 함수를 사용하여 텍스트를 변경할 수 있습니다.
- **within_text**에서 **find_text**를 찾을 수 없으면 수식은 오류를 반환합니다. 이 동작은 부분 문자열을 찾을 수 없는 경우 #VALUE를 반환하는 Excel과 비슷합니다. **within_text**의 null은 이 컨텍스트에서 빈 문자열로 해석됩니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제: 문자열 내에서 검색

다음 수식은 "printer" 단어에서 "n" 문자의 위치를 찾습니다.

```
= SEARCH("n","printer")
```

"n"은 "printer" 단어에서 네 번째 문자이므로 수식은 4를 반환합니다.

예제: 열 내에서 검색

열 참조를 SEARCH의 인수로 사용할 수 있습니다. 다음 수식은 [PostalCode] 열에서 "-"(하이픈) 문자의 위치를 찾습니다.

```
= SEARCH("-",[PostalCode])
```

반환 결과는 하이픈의 인덱스 위치를 나타내는 숫자 열입니다.

예제: SEARCH를 사용한 오류 처리

소스 열의 일부 행에서 검색 문자열을 찾을 수 없으면 위 예제의 수식은 실패합니다. 따라서 다음 예제에서는 SEARCH 함수와 함께 IFERROR를 사용하여 모든 행에 대해 유효한 결과가 반환되도록 하는 방법을 보여 줍니다.

다음 수식은 열에서 "-" 문자의 위치를 찾고, 문자열을 찾을 수 없으면 -1을 반환합니다.

```
= IFERROR(SEARCH("-",[PostalCode]),-1)
```

오류 출력으로 사용하는 값의 데이터 형식은 오류가 아닌 출력 형식의 데이터 형식과 일치해야 합니다. 이 예제에서는 SEARCH가 정수 값을 반환하므로 오류가 발생할 경우 출력할 숫자 값을 제공합니다. 그러나 IFERROR의 두 번째 인수로 `BLANK()`를 사용하여 공백(빈 문자열)을 반환할 수도 있습니다.

참고 항목

[MID](#)

[REPLACE](#)

[텍스트 함수](#)

SUBSTITUTE

2021-03-16 • 2 minutes to read

기존 텍스트를 텍스트 문자열의 새 텍스트로 바꿉니다.

구문

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

매개 변수

용어	정의
텍스트	문자를 대체하려고 하는 텍스트 또는 텍스트가 포함된 열에 대한 참조입니다.
old_text	바꾸려는 기존 텍스트입니다.
new_text	old_text 를 대체할 텍스트입니다.
instance_num	(선택 사항) 바꾸려는 old_text 의 인스턴스 수입니다. 생략하면 old_text 의 모든 인스턴스가 바뀝니다.

반환 값

텍스트 문자열입니다.

설명

- 텍스트 문자열에서 특정 텍스트를 바꾸려는 경우 SUBSTITUTE 함수를 사용합니다. 텍스트 문자열의 특정 위치에서 발생하는 가변 길이 텍스트를 바꾸려는 경우 REPLACE 함수를 사용합니다.
- SUBSTITUTE 함수는 대/소문자를 구분합니다. **text** 와 **old_text** 간에 대/소문자가 일치하지 않는 경우 SUBSTITUTE는 텍스트를 대체하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제: 문자열 내 대체

다음 수식에서는 이전 제품 코드 **PA** 가 열어 나타날 때마다 새 제품 코드 **NW** 로 대체하는 [Product Code] 열의 복사본을 만듭니다.

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

참고 항목

[텍스트 함수](#)

[REPLACE](#)

TRIM

2021-03-16 • 3 minutes to read

단어 사이의 하나의 공백을 제외하고 텍스트에서 모든 공백을 제거합니다.

구문

```
TRIM(<text>)
```

매개 변수

용어	정의
text	공백을 제거하려는 텍스트이거나 텍스트가 포함된 열입니다.

반환 값

공백이 제거된 문자열입니다.

설명

- 간격이 불규칙할 수 있는, 다른 애플리케이션에서 받은 텍스트에 대해 TRIM을 사용합니다.
- TRIM 함수는 원래 텍스트에서 7비트 ASCII 공백 문자(값 32)를 자르기 위해 작성되었습니다. 유니코드 문자 집합에는 10진수 값이 160인, 줄 바꿈하지 않는 공백 문자라는 추가 공백 문자가 있습니다. 이 문자는 주로 웹 페이지에서 HTML 엔터티 로 사용됩니다. TRIM 함수는 줄 바꿈하지 않는 공백 문자를 단독으로 제거하지 않습니다. 텍스트에서 두 공백 문자를 모두 자르는 방법의 예제는 텍스트에서 공백 및 인쇄되지 않는 문자 제거를 참조하세요.

예제

다음 수식은 후행 공백이 없는 새 문자열을 만듭니다.

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

수식을 만들 때는 수식이 입력한 대로 행에 전파되므로 각 수식에 원래 문자열이 표시되고 결과가 명확하지 않습니다. 그러나 수식을 계산하면 문자열이 잘립니다.

이전 수식에서 만들어진 계산 열의 길이를 다음과 같이 검사하여 수식이 올바른 결과를 생성하는지 확인할 수 있습니다.

```
= LEN([Calculated Column 1])
```

참고 항목

[텍스트 함수](#)

UNICHAR

2021-03-16 • 2 minutes to read

숫자 값에서 참조하는 유니코드 문자를 반환합니다.

구문

```
UNICHAR(number)
```

매개 변수

용어	정의
number	문자를 나타내는 유니코드 숫자입니다.

반환 값

유니코드 숫자가 나타내는 문자입니다.

설명

- XML 문자가 유효하지 않으면 UNICHAR는 오류를 반환합니다.
- 유니코드 숫자가 부분 서로게이트이고 데이터 형식이 유효하지 않으면 UNICHAR는 오류를 반환합니다.
- number가 허용 범위를 벗어난 숫자 값이면 UNICHAR는 오류를 반환합니다.
- number가 0이면 UNICHAR는 오류를 반환합니다.
- 반환되는 유니코드 문자는 UTF-8 또는 UTF-16 코드와 같은 문자열일 수 있습니다.

예제

다음 예제는 유니코드 숫자 66(대문자 A)이 나타내는 문자를 반환합니다.

```
= UNICHAR(65)
```

다음 예제는 유니코드 숫자 32(공백 문자)가 나타내는 문자를 반환합니다.

```
= UNICHAR(32)
```

다음 예제는 유니코드 숫자 9733(★ 문자)이 나타내는 문자를 반환합니다.

```
= UNICHAR(9733)
```

UNICODE

2021-03-16 • 2 minutes to read

텍스트의 첫 번째 문자에 해당하는 숫자(코드 포인트)를 반환합니다.

구문

```
UNICODE( <Text> )
```

매개 변수

용어	정의
텍스트	텍스트는 유니코드 값을 구하려는 문자입니다.

반환 값

텍스트 문자열의 첫 번째 문자에 대한 숫자 코드입니다.

UPPER

2021-03-16 • 2 minutes to read

텍스트 문자열을 모두 대문자로 변환합니다.

구문

```
UPPER (<text>)
```

매개 변수

용어	정의
text	대문자로 변환할 텍스트이거나 텍스트가 포함된 열에 대한 참조입니다.

반환 값

대문자로 표시된 동일한 텍스트입니다.

예제

다음 수식은 [ProductCode] 열의 문자열을 모두 대문자로 변환합니다. 영문자가 아닌 문자는 영향을 받지 않습니다.

```
= UPPER(['New Products'[Product Code]))
```

참고 항목

[텍스트 함수](#)

[LOWER 함수](#)

값

2021-03-16 • 2 minutes to read

숫자를 나타내는 텍스트 문자열을 숫자로 변환합니다.

구문

```
VALUE(<text>)
```

매개 변수

용어	정의
text	변환할 텍스트입니다.

반환 값

10진수 데이터 형식으로 변환된 숫자입니다.

설명

- **text** 매개 변수로 전달되는 값은 사용하는 애플리케이션 또는 서비스에서 인식되는 모든 상수, 숫자, 날짜 또는 시간 형식일 수 있습니다. **text** 가 이러한 형식 중 하나가 아니면 오류가 반환됩니다.
- 일반적으로 엔진에서 필요에 따라 암시적으로 텍스트를 숫자로 변환하므로 수식에서 VALUE 함수를 사용할 필요가 없습니다.
- 열 참조를 사용할 수도 있습니다. 예를 들어 열에 혼합된 숫자 형식이 포함되어 있는 경우 VALUE를 사용하여 모든 값을 단일 숫자 데이터 형식으로 변환할 수 있습니다. 하지만 혼합된 숫자 및 텍스트가 포함된 열에서 VALUE 함수를 사용해도 모든 행의 모든 값을 숫자로 변환할 수 있는 것은 아니므로 전체 열에 오류 플래그가 지정됩니다.

예제

다음 수식은 입력한 문자열 "3"을 숫자 값 3으로 변환합니다.

```
= VALUE("3")
```

참고 항목

[텍스트 함수](#)

시간 인텔리전스 함수

2021-03-16 • 10 minutes to read

또한 DAX(Data Analysis Expression)에는 시간 인텔리전스 함수가 포함되어 있습니다. 이러한 함수를 활용하면 일, 월, 분기, 년 등의 시간 단위를 사용하여 데이터를 조작한 다음 해당 기간을 대상으로 식을 작성하고 값을 비교할 수 있습니다.

이 범주의 함수

함수	설명
CLOSINGBALANCEMONTH	현재 컨텍스트에 있는 월의 마지막 날짜에서 식을 계산합니다.
CLOSINGBALANCEQUARTER	현재 컨텍스트에 있는 분기의 마지막 날짜에서 식을 계산합니다.
CLOSINGBALANCEYEAR	현재 컨텍스트에 있는 연도의 마지막 날짜에서 식을 계산합니다.
DATEADD	현재 컨텍스트의 날짜에서 지정된 간격 수만큼 앞으로 또는 뒤로 이동하는 날짜 열이 포함된 테이블을 반환합니다.
DATESBETWEEN	지정된 시작 날짜에 시작하고 지정된 종료 날짜까지 계속되는 날짜 열이 포함된 테이블을 반환합니다.
DATESINPERIOD	지정된 시작 날짜에 시작하고 지정된 수와 형식의 날짜 간격 동안 계속되는 날짜 열이 포함된 테이블을 반환합니다.
DATESMTD	현재 컨텍스트에 있는 월간 누계에 대해 날짜 열이 포함된 테이블을 반환합니다.
DATESQTD	현재 컨텍스트에서 분기별 날짜 열이 포함된 테이블을 반환합니다.
DATESYTD	현재 컨텍스트에서 연간 날짜 열이 포함된 테이블을 반환합니다.
ENDOFMONTH	지정된 날짜의 열에 대한 현재 컨텍스트에서 월의 마지막 날짜를 반환합니다.
ENDOFQUARTER	지정된 날짜의 열에 대한 현재 컨텍스트에서 분기의 마지막 날짜를 반환합니다.
ENDOFYEAR	지정된 날짜의 열에 대한 현재 컨텍스트에서 연도의 마지막 날짜를 반환합니다.
FIRSTDATE	지정된 날짜의 열에 대한 현재 컨텍스트에서 첫 번째 날짜를 반환합니다.

함수	설명
FIRSTNONBLANK	식이 비어 있지 않은 현재 컨텍스트에 의해 필터링된 열에서 첫 번째 값을 반환합니다.
LASTDATE	지정된 날짜의 열에 대한 현재 컨텍스트에서 마지막 날짜를 반환합니다.
LASTNONBLANK	식이 비어 있지 않은 현재 컨텍스트에 의해 필터링된 열에서 마지막 값을 반환합니다.
NEXTDAY	현재 컨텍스트의 dates 열에 지정된 첫 번째 날짜를 기준으로 다음 날부터 모든 날짜의 열이 포함된 테이블을 반환합니다.
NEXTMONTH	현재 컨텍스트의 dates 열에 있는 첫 번째 날짜를 기준으로 다음 달부터 모든 날짜의 열이 포함된 테이블을 반환합니다.
NEXTQUARTER	현재 컨텍스트의 dates 열에 지정된 첫 번째 날짜를 기준으로 다음 분기에서 모든 날짜의 열이 포함된 테이블을 반환합니다.
NEXTYEAR	현재 컨텍스트의 dates 열에 있는 첫 번째 날짜를 기준으로 다음 연도에서 모든 날짜의 열이 포함된 테이블을 반환합니다.
OPENINGBALANCEMONTH	현재 컨텍스트에 있는 월의 첫 번째 날짜에서 식을 계산합니다.
OPENINGBALANCEQUARTER	현재 컨텍스트에 있는 분기의 첫 번째 날짜에서 식을 계산합니다.
OPENINGBALANCEYEAR	현재 컨텍스트에 있는 연도의 첫 번째 날짜에서 식을 계산합니다.
PARALLELPERIOD	현재 컨텍스트에서 지정된 dates 열의 날짜와 평행한 기간을 나타내는 날짜의 열이 포함된 테이블을 반환합니다. 날짜는 기간의 숫자만큼 시간상 앞으로 또는 뒤로 이동합니다.
PREVIOUSDAY	현재 컨텍스트에서 dates 열에 있는 첫 번째 날짜의 이전 날짜를 나타내는 모든 날짜의 열이 포함된 테이블을 반환합니다.
PREVIOUSMONTH	현재 컨텍스트의 dates 열에 있는 첫 번째 날짜를 기준으로 이전 달부터 모든 날짜의 열이 포함된 테이블을 반환합니다.
PREVIOUSQUARTER	현재 컨텍스트의 dates 열에 있는 첫 번째 날짜를 기준으로 이전 분기부터 모든 날짜의 열이 포함된 테이블을 반환합니다.
PREVIOUSYEAR	현재 컨텍스트의 dates 열에 있는 마지막 날짜를 기준으로 이전 연도부터 모든 날짜의 열이 포함된 테이블을 반환합니다.
SAMEPERIODLASTYEAR	현재 컨텍스트에서 지정된 dates 열의 날짜로부터 1 년 뒤로 이동한 날짜의 열이 포함된 테이블을 반환합니다.

함수	설명
STARTOFMONTH	지정된 날짜의 열에 대한 현재 컨텍스트에서 월의 첫 번째 날짜를 반환합니다.
STARTOFQUARTER	지정된 날짜의 열에 대한 현재 컨텍스트에서 분기의 첫 번째 날짜를 반환합니다.
STARTOFYEAR	지정된 날짜의 열에 대한 현재 컨텍스트에서 연도의 첫 번째 날짜를 반환합니다.
TOTALMTD	현재 컨텍스트에 있는 월간 누계에 대해 식의 값을 계산합니다.
TOTALQTD	현재 컨텍스트에서 분기 누계의 날짜에 대해 식의 값을 계산합니다.
TOTALYTD	현재 컨텍스트에서 식의 연간 누계 값을 계산합니다.

CLOSINGBALANCEMONTH

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 해당 월의 마지막 날짜에 **expression** 을 계산합니다.

구문

```
CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

매개 변수

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 해당 월의 마지막 날짜에 계산되는 **expression** 을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#)에 설명되어 있습니다.

NOTE

filter 식에는 [CALCULATE 함수](#)에 설명된 제한 사항이 있습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 제품 인벤토리의 '월말 인벤토리 값'을 계산하는 측정값을 만듭니다.


```
=  
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTim  
e[DateKey])
```

참고 항목

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEQUARTER 함수](#)

CLOSINGBALANCEQUARTER

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 해당 분기의 마지막 날짜에 **expression** 을 계산합니다.

구문

```
CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

매개 변수

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 해당 분기의 마지막 날짜에 계산된 **expression** 을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#)에 설명되어 있습니다.

NOTE

filter 식에는 [CALCULATE 함수](#)에 설명된 제한 사항이 있습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 제품 인벤토리의 '분기말 인벤토리 값'을 계산하는 측정값을 만듭니다.

=

CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])

참고 항목

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEMONTH 함수](#)

CLOSINGBALANCEYEAR

2021-03-16 • 3 minutes to read

현재 컨텍스트에서 연도의 마지막 날짜에 **expression** 을 계산합니다.

구문

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][, <year_end_date>])
```

매개 변수

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

현재 컨텍스트에서 연도의 마지막 날짜에 계산되는 **expression** 을 나타내는 스칼라 값입니다.

설명

- **year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘의 날짜 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#)에 설명되어 있습니다.

NOTE

filter 식에는 [CALCULATE 함수](#)에 설명된 제한 사항이 있습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 제품 인벤토리의 'Year End Inventory Value'를 계산하는 측정값을 만듭니다.

```
=  
CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime  
[DateKey])
```

참고 항목

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEQUARTER 함수](#)

[CLOSINGBALANCEMONTH 함수](#)

DATEADD

2021-03-16 • 3 minutes to read

현재 컨텍스트의 날짜에서 지정된 간격 수만큼 앞으로 또는 뒤로 이동하는 날짜 열이 포함된 테이블을 반환합니다.

구문

```
DATEADD(<dates>,<number_of_intervals>,<interval>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
number_of_intervals	날짜에서 더하거나 뺀 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. interval 값은 <code>year</code> , <code>quarter</code> , <code>month</code> , <code>day</code> 중 하나일 수 있습니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

dates 인수는 다음 중 하나일 수 있습니다.

- 날짜/시간 열에 대한 참조
- 날짜/시간 값의 단일 열을 반환하는 테이블 식
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.

- number_of_intervals**에 지정된 숫자가 양수이면 **dates**의 날짜가 이후 시간으로 이동되고, 숫자가 음수이면 **dates**의 날짜가 이전 시간으로 이동됩니다.
- interval** 매개 변수는 문자열 세트가 아니라 열거형입니다. 따라서 값을 따옴표로 묶지 말아야 합니다. 또한 `year`, `quarter`, `month`, `day` 값을 사용할 때 철자 전체를 사용해야 합니다.
- 결과 테이블에는 **dates** 열에 있는 날짜만 포함됩니다.
- 현재 컨텍스트의 날짜가 연속된 간격을 형성하지 않는 경우, 함수는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 - 날짜 세트 전환

다음 수식에서는 현재 컨텍스트에 있는 날짜의 1년 전 날짜를 계산합니다.

```
= DATEADD(DateTime[DateKey], -1, year)
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

DATESBETWEEN

2021-03-16 • 5 minutes to read

지정된 시작 날짜에 시작하고 지정된 종료 날짜까지 계속되는 날짜 열이 포함된 테이블을 반환합니다.

이 함수는 [CALCULATE](#) 함수에 필터로 전달하기에 적합합니다. 사용자 지정 날짜 범위로 식을 필터링하는 데 사용됩니다.

NOTE

일, 월, 분기 또는 연도와 같은 표준 날짜 간격을 사용하는 경우 더 적합한 [DATESINPERIOD](#) 함수를 사용하는 것이 좋습니다.

구문

```
DATESBETWEEN(<dates>, <start_date>, <end_date>)
```

매개 변수

용어	정의
날짜	날짜 열입니다.
start_date	날짜 식입니다.
end_date	날짜 식입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 가장 일반적인 사용 사례에서 **dates** 는 표시된 날짜 테이블의 날짜 열에 대한 참조입니다.
- start_date** 가 BLANK이면 **start_date** 는 **dates** 열에서 가장 이른 값이 됩니다.
- If **end_date** 가 BLANK이면 **end_date** 는 **dates** 열에서 가장 늦은 값이 됩니다.
- start_date** 및 **end_date** 로 사용되는 날짜는 포함됩니다. 따라서 예를 들어 **start_date** 값이 2019년 7월 1일인 경우 해당 날짜는 반환되는 테이블에 포함됩니다(이 날짜가 **dates** 열에 있는 경우).
- 반환되는 테이블에는 **dates** 열에 저장된 날짜만 포함될 수 있습니다. 그러므로 예를 들어 **dates** 열이 2017년 7월 1일부터 시작하고 **start_date** 값이 2016년 7월 1일이면 반환되는 테이블은 2017년 7월 1일부터 시작합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 **Sales** 테이블 측정 정의에서는 DATESBETWEEN 함수를 사용하여 *life-to-date*(LTD) 계산을 생성합니다. LTD(life-to-date)는 시간이 시작된 후 시간이 지남에 따른 측정값의 누적을 나타냅니다.

수식은 **MAX** 함수를 사용합니다. 이 함수는 필터 컨텍스트에 있는 가장 늦은 날짜를 반환합니다. 따라서 DATESBETWEEN 함수는 가장 이른 날짜부터 시작하여 보고되는 가장 늦은 날짜까지의 테이블을 반환합니다.

이 문서의 예는 *Power BI Desktop* 샘플 모델에 추가할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

```
Customers LTD =  
CALCULATE(  
    DISTINCTCOUNT(Sales[CustomerKey]),  
    DATESBETWEEN(  
        'Date'[Date],  
        BLANK(),  
        MAX('Date'[Date])  
    )  
)
```

Date 테이블에 저장된 가장 이른 날짜가 2017년 7월 1일이라고 가정해 보겠습니다. 따라서 보고서가 2020년 6월을 기준으로 측정값을 필터링하는 경우 DATESBETWEEN 함수는 2017년 7월 1일부터 2020년 6월 30일까지의 날짜 범위를 반환합니다.

참고 항목

- [시간 인텔리전스 함수\(DAX\)](#)
- [날짜 및 시간 함수\(DAX\)](#)
- [DATESINPERIOD 함수\(DAX\)](#)

DATESINPERIOD

2021-03-16 • 5 minutes to read

지정된 시작 날짜에 시작하고 지정된 수와 형식의 날짜 간격 동안 계속되는 날짜 열이 포함된 테이블을 반환합니다.

이 함수는 [CALCULATE](#) 함수에 필터로 전달하기에 적합합니다. 일, 월, 분기 또는 연도와 같은 표준 날짜 간격으로 식을 필터링하는 데 사용하세요.

구문

```
DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)
```

매개 변수

용어	정의
날짜	날짜 열입니다.
start_date	날짜 식입니다.
number_of_intervals	날짜에서 더하거나 뺄 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. interval 값은 <code>DAY</code> , <code>MONTH</code> , <code>QUARTER</code> , <code>YEAR</code> 중 하나일 수 있습니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 가장 일반적인 사용 사례에서 **dates** 는 표시된 날짜 테이블의 날짜 열에 대한 참조입니다.
- number_of_intervals** 에 지정된 숫자가 양수이면 날짜가 앞으로 이동합니다. 숫자가 음수이면 날짜가 과거로 이동합니다.
- interval** 매개 변수는 열거형입니다. 유효한 값은 `DAY`, `MONTH`, `QUARTER` 및 `YEAR` 입니다. 열거형이므로 값이 문자열로 전달되지 않습니다. 따라서 따옴표로 묶지 마세요.
- 반환되는 테이블에는 **dates** 열에 저장된 날짜만 포함될 수 있습니다. 그러므로 예를 들어 **dates** 열이 2017년 7월 1일부터 시작하고 **start_date** 값이 2016년 7월 1일이면 반환되는 테이블은 2017년 7월 1일부터 시작합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 **Sales** 테이블 측정 정의에서는 DATESINPERIOD 함수를 사용하여 이전 연도(PY)의 수익을 계산합니다.

수식은 [MAX](#) 함수를 사용합니다. 이 함수는 필터 컨텍스트에 있는 가장 늦은 날짜를 반환합니다. 따라서 DATESINPERIOD 함수는 작년의 가장 늦은 날짜부터 시작하는 날짜 테이블을 반환합니다.

이 문서의 예는 Power BI Desktop 샘플 모델에 추가할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

```
Revenue PY =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    DATESINPERIOD(  
        'Date'[Date],  
        MAX('Date'[Date]),  
        -1,  
        YEAR  
    )  
)
```

보고서가 2020년 6월로 필터링된다고 생각해 보세요. MAX 함수는 2020년 6월 30일을 반환합니다. 그러면 DATESINPERIOD 함수는 2019년 7월 1일부터 2020년 6월 30일까지의 날짜 범위를 반환합니다. 작년의 경우 2020년 6월 30일부터 시작하는 날짜 값으로 이루어진 연도입니다.

참고 항목

[시간 인텔리전스 함수\(DAX\)](#)

[날짜 및 시간 함수\(DAX\)](#)

[DATESBETWEEN 함수\(DAX\)](#)

DATESMTD

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 월간 날짜 열이 포함된 테이블을 반환합니다.

구문

DATESMTD(<dates>)

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

dates 인수는 다음 중 하나일 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매에 대한 '월간 누계'를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESMTD(DateTime[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATESYTD 함수](#)

[DATESQTD 함수](#)

DATESQTD

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 분기별 날짜 열이 포함된 테이블을 반환합니다.

구문

DATESQTD(<dates>)

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

dates 인수는 다음 중 하나일 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '분기별 실행 누계'를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESQTD(DateTime[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATESYTD 함수](#)

[DATESMTD 함수](#)

DATESYTD

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 연간 날짜 열이 포함된 테이블을 반환합니다.

구문

```
DATESYTD(<dates> [,<year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

dates 인수는 다음 중 하나일 수 있습니다.

- 날짜/시간 열에 대한 참조
- 날짜/시간 값의 단일 열을 반환하는 테이블 식
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

NOTE

부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.

- **year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매에 대한 '실행 누계'를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESYTD(DateTime[DateKey]))
```

참고 항목

시간 인텔리전스 함수

날짜 및 시간 함수

DATESMTD 함수

DATESQTD 함수

ENDOFMONTH

2021-03-16 • 2 minutes to read

지정된 날짜 열의 현재 컨텍스트에서 해당 월의 마지막 날짜를 반환합니다.

구문

```
ENDOFMONTH(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 현재 컨텍스트에서 해당 월의 끝을 반환하는 측정값을 만듭니다.

```
= ENDOFMONTH(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFYEAR 함수](#)

[ENDOFQUARTER 함수](#)

ENDOFQUARTER

2021-03-16 • 2 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 분기의 마지막 날짜를 반환합니다.

구문

```
ENDOFQUARTER(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 현재 컨텍스트에 대해 분기의 끝을 반환하는 측정값을 만듭니다.

```
= ENDOFQUARTER(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFYEAR 함수](#)

[ENDOFMONTH 함수](#)

ENDOFYEAR

2021-03-16 • 2 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 연도의 마지막 날짜를 반환합니다.

구문

```
ENDOFYEAR(<dates> [, <year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- **year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 현재 컨텍스트에 대해 6월 30일에 끝나는 회계 연도의 끝을 반환하는 측정값을 만듭니다.

```
= ENDOFYEAR(DateTime[DateKey], "06/30/2004")
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFMONTH 함수](#)

[ENDOFQUARTER 함수](#)

FIRSTDATE

2021-03-16 • 3 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 첫 번째 날짜를 반환합니다.

구문

```
FIRSTDATE(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 현재 컨텍스트가 단일 날짜인 경우 FIRSTDATE 및 LASTDATE 함수에서 반환되는 날짜는 동일합니다.
- 반환 값은 단일 열과 단일 값을 포함하는 테이블입니다. 따라서 이 함수는 해당 인수에서 테이블이 필요한 함수에 대한 인수로 사용할 수 있습니다. 또한, 날짜 값이 필요할 때마다 반환된 값을 사용할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 현재 컨텍스트에 대해 인터넷 판매 채널에서 판매량이 만들어진 첫 번째 날짜를 가져오는 측정값을 만듭니다.

```
= FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[LASTDATE 함수](#)

FIRSTNONBLANK

2021-03-16 • 2 minutes to read

식이 비어 있지 않은 현재 컨텍스트에 의해 필터링된 열, **column** 에서 첫 번째 값을 반환합니다.

구문

```
FIRSTNONBLANK(<column>, <expression>)
```

매개 변수

용어	정의
열	열 식입니다.
식	열의 값마다 공백에 대해 평가되는 식입니다.

반환 값

계산된 첫 번째 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **column** 인수는 다음 중 하나일 수 있습니다.
 - 모든 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 일반적으로 식이 비어 있지 않은 열의 첫 번째 값을 반환하는 데 사용됩니다. 예를 들어 제품이 판매된 마지막 값을 가져올 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[LASTNONBLANK 함수](#)

[통계 함수](#)

FIRSTNONBLANKVALUE

2021-03-16 • 2 minutes to read

열의 정렬된 값으로 필터링된 식을 계산하고 비어 있지 않은 식의 첫 번째 값을 반환합니다.

구문

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

매개 변수

용어	정의
열	단일 열 테이블을 반환하는 열 또는 식입니다.
식	<column>의 각 값에 대해 평가되는 식입니다.

반환 값

<column>의 정렬된 값에 해당하는 비어 있지 않은 첫 번째 <expression> 값입니다.

설명

- column 인수는 다음 중 하나일 수 있습니다.
 - 모든 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 이 함수는 <expression> 평가를 위해 <column>이 필터 컨텍스트에 추가된다는 점에서 FIRSTNONBLANK와 다릅니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "FNBV",
    FIRSTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

그러면 다음 값이 반환됩니다.

DIMPRODUCT[CLASS]	[FNBV]
L	699.0982
H	13778.24
M	1000.4375
	533.83

LASTDATE

2021-03-16 • 3 minutes to read

지정된 날짜 열의 현재 컨텍스트에서 마지막 날짜를 반환합니다.

구문

```
LASTDATE(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 현재 컨텍스트가 단일 날짜인 경우 FIRSTDATE 및 LASTDATE 함수에서 반환되는 날짜는 동일합니다.
- 기술적으로 반환 값은 단일 열과 단일 값이 포함된 테이블입니다. 따라서 이 함수는 해당 인수에서 테이블이 필요한 함수에 대한 인수로 사용할 수 있습니다. 또한, 날짜 값이 필요할 때마다 반환된 값을 사용할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 인터넷 판매 채널을 통해 판매가 이루어진 경우 현재 컨텍스트에서 마지막 날짜를 가져 오는 측정값을 만듭니다.

작동 방식을 보려면 피벗 테이블을 만들고 CalendarYear 필드를 피벗 테이블의 **행 레이블** 영역에 추가합니다. 그런 다음, 코드 섹션에 정의된 수식을 사용하여 **LastSaleDate** 라는 측정값을 피벗 테이블의 **값** 영역에 추가합니다.

```
= LASTDATE('InternetSales_USD'[SaleDateKey])
```

참고 항목

날짜 및 시간 함수

시간 인텔리전스 함수

FIRSTDATE 함수

LASTNONBLANK 함수

LASTNONBLANK

2021-03-16 • 2 minutes to read

식이 공백이 아닌 현재 컨텍스트에서 필터링된 **column** 열의 마지막 값을 반환합니다.

구문

```
LASTNONBLANK(<column>,<expression>)
```

매개 변수

용어	정의
열	열 식입니다.
식	열의 값마다 공백에 대해 평가되는 식입니다.

반환 값

계산된 마지막 값이 있는 단일 열과 단일 행이 포함되어 있는 테이블입니다.

설명

- **column** 인수는 다음 중 하나일 수 있습니다.
 - 모든 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
 - 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 일반적으로 식이 공백이 아닌 열의 마지막 값을 반환하는 데 사용됩니다. 예를 들어 제품이 판매된 마지막 값을 가져올 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[FIRSTNONBLANK 함수](#)

[통계 함수](#)

LASTNONBLANKVALUE

2021-03-16 • 2 minutes to read

열의 정렬된 값으로 필터링된 식을 계산하고 비어 있지 않은 식의 마지막 값을 반환합니다.

구문

```
LASTNONBLANKVALUE(<column>, <expression>)
```

매개 변수

용어	정의
열	단일 열 테이블을 반환하는 열 또는 식입니다.
식	<column>의 각 값에 대해 평가되는 식입니다.

반환 값

<column>의 정렬된 값에 해당하는 비어 있지 않은 마지막 <expression> 값입니다.

설명

- column 인수는 다음 중 하나일 수 있습니다.
 - 모든 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 이 함수는 <expression> 평가를 위해 <column>이 필터 컨텍스트에 추가된다는 점에서 LASTNONBLANK와 다릅니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 DAX 쿼리를 실행한다고 가정합니다.

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "LNBV",
    LASTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

그러면 다음 값이 반환됩니다.

DIMPRODUCT[CLASS]	[LNBV]
L	132.44
H	137.6
M	84.97
	2288.6

NEXTDAY

2021-03-16 • 2 minutes to read

현재 컨텍스트의 **dates** 열에 지정된 첫 번째 날짜를 기준으로 다음 날부터 모든 날짜의 열이 포함된 테이블을 반환합니다.

구문

```
NEXTDAY(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 매개 변수에 나오는 첫 번째 날짜 그다음 날의 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 첫 번째 날짜가 2009년 6월 10일을 가리키는 경우 이 함수는 2009년 6월 11일과 동일한 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 인터넷 판매의 'next day sales'를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTDAY('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTQUARTER 함수](#)

[NEXTMONTH 함수](#)

NEXTMONTH

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 **dates** 열의 첫 번째 날짜를 기준으로 그다음 달의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
NEXTMONTH(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 매개 변수에 나오는 첫 번째 날짜 그다음 날의 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하면 이 함수는 2009년 7월의 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '다음달 매출'을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTMONTH('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTDAY 함수](#)

[NEXTQUARTER 함수](#)

[NEXTYEAR 함수](#)

NEXTQUARTER

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 **dates** 열에 지정된 첫 번째 날짜를 기준으로 그다음 분기의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
NEXTQUARTER(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 첫 번째 날짜를 기준으로 그다음 분기의 모든 날짜를 반환합니다. 예를 들어 **dates** 열의 첫 번째 날짜가 2009년 6월 10일을 참조하면 이 함수는 2009년 7월~9월 분기의 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '다음 분기 매출'을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTQUARTER('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTDAY 함수](#)

[NEXTMONTH 함수](#)

NEXTYEAR

2021-03-16 • 3 minutes to read

현재 컨텍스트에서 **dates** 열의 첫 번째 날짜를 기준으로 그다음 연도의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
NEXTYEAR(<dates>[, <year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 열의 첫 번째 날짜를 기준으로 그다음 연도의 모든 날짜를 반환합니다. 예를 들어 **dates** 열의 첫 번째 날짜가 2007년을 참조하면 이 함수는 2008년의 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '내년 매출'을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTYEAR('DateTime'[DateKey]))
```

참고 항목

시간 인텔리전스 함수

날짜 및 시간 함수

NEXTDAY 함수

NEXTQUARTER 함수

NEXTMONTH 함수

OPENINGBALANCEMONTH

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 해당 월 첫 번째 날짜에 **expression** 을 평가합니다.

구문

```
OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

매개 변수

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 해당 월의 첫 번째 날짜에 평가되는 **expression** 식을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- **filter** 식에는 [CALCULATE 함수](#) 항목에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 제품 인벤토리의 '월초 인벤토리 값'을 계산하는 측정값을 만듭니다.

```
=  
OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTim  
e[DateKey])
```

참고 항목

[OPENINGBALANCEYEAR 함수](#)

OPENINGBALANCEQUARTER 함수

시간 인텔리전스 함수

CLOSINGBALANCEMONTH 함수

OPENINGBALANCEQUARTER

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 해당 분기의 첫 번째 날짜에 **expression** 을 평가합니다.

구문

```
OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

매개 변수

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filte*	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 해당 분기의 첫 번째 날짜에 평가된 **expression** 을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- **filter** 식에는 [CALCULATE 함수](#) 항목에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 제품 인벤토리의 '분기 초 인벤토리 값'을 계산하는 측정값을 만듭니다.

```
=  
OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateT  
ime[DateKey])
```

참고 항목

[OPENINGBALANCEYEAR 함수](#)

OPENINGBALANCEMONTH 함수
시간 인텔리전스 함수
CLOSINGBALANCEQUARTER 함수

OPENINGBALANCEYEAR

2021-03-16 • 3 minutes to read

현재 컨텍스트에서 해당 연도의 첫 번째 날짜에 식 을 계산합니다.

구문

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

매개 변수

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

현재 컨텍스트에서 해당 연도의 첫 번째 날짜에 계산되는 식 을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- **filter** 식에는 [CALCULATE 함수](#) 항목에 설명된 제한 사항이 있습니다.
- **year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 제품 인벤토리의 '연초 인벤토리 값'을 계산하는 측정값을 만듭니다.

=

OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime
[DateKey])

참고 항목

[OPENINGBALANCEQUARTER 함수](#)

[OPENINGBALANCEMONTH 함수](#)

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

PARALLELPERIOD

2021-03-16 • 6 minutes to read

현재 컨텍스트에서 지정된 **dates** 열의 날짜와 평행한 기간(날짜가 특정 간격 수만큼 이전 시간이나 이후 시간으로 이동된 기간)을 나타내는 날짜 열이 포함된 테이블을 반환합니다.

구문

```
PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
number_of_intervals	날짜에서 더하거나 뺀 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. interval 값은 <code>year</code> , <code>quarter</code> , <code>month</code> 중 하나일 수 있습니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 **dates** 로 지정된 열의 현재 날짜 집합을 사용하고 첫 번째 날짜와 마지막 날짜를 지정된 간격 수만큼 이동한 다음, 이동된 두 날짜 사이의 모든 연속 날짜를 반환합니다. 간격이 월, 분기 또는 연도의 일부 범위이면 결과의 부분 월도 전체 간격을 완성하기 위해 채워집니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- number_of_intervals** 에 지정된 숫자가 양수이면 **dates** 의 날짜가 이후 시간으로 이동되고, 숫자가 음수이면 **dates** 의 날짜가 이전 시간으로 이동됩니다.
- interval** 매개 변수는 문자열 세트가 아니라 열거형입니다. 따라서 값을 따옴표로 묶지 말아야 합니다. 또한 `year`, `quarter`, `month` 값을 사용할 때는 전체 철자로 지정해야 합니다.
- 결과 테이블에는 기본 테이블 열의 값에 표시되는 날짜만 포함됩니다.
- PARALLELPERIOD 함수는 DATEADD 함수와 비슷합니다. 단, 부분 기간을 반환하는 DATEADD와 달리 PARALLELPERIOD는 항상 지정된 세분성 수준의 전체 기간을 반환합니다. 예를 들어 동일한 연도의 6월 10일부터 6월 21일까지의 날짜를 선택한 다음, 선택한 날짜를 1달 이후로 이동하려는 경우 PARALLELPERIOD 함수는 다음달의 모든 날짜(7월 1일~7월 31일)를 반환하지만 DATEADD를 대신 사용

하면 7월 10일부터 7월 21일까지의 날짜만 결과에 포함됩니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 작년 매출을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PARALLELPERIOD(DateTime[DateKey],-1,year))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATEADD 함수](#)

PREVIOUSDAY

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 **dates** 열에 있는 첫 번째 날짜의 전일을 나타내는 모든 날짜의 열이 포함된 테이블을 반환합니다.

구문

```
PREVIOUSDAY(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 첫 번째 날짜를 확인한 후 첫 번째 날짜의 전일에 해당하는 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 첫 번째 날짜가 2009년 6월 10일을 나타내는 경우 이 함수는 2009년 6월 9일에 해당하는 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE 함수](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 인터넷 판매의 '전일 판매'를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSDAY('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSMONTH 함수](#)

[PREVIOUSQUARTER 함수](#)

PREVIOUSMONTH

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 **dates** 열의 첫 번째 날짜를 기준으로 이전 달의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
PREVIOUSMONTH(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력으로 사용된 열의 첫 번째 날짜를 기준으로 이전 달의 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하면 이 함수는 2009년 5월의 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '전달 매출'을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSMONTH('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

PREVIOUSQUARTER

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 **dates** 열의 첫 번째 날짜를 기준으로 이전 분기의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
PREVIOUSQUARTER(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 열의 첫 번째 날짜를 사용하여 이전 분기의 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하면 이 함수는 2009년 1월~3월 분기의 모든 날짜를 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '전분기 매출'을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSQUARTER('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

PREVIOUSYEAR

2021-03-16 • 3 minutes to read

현재 컨텍스트에서 **dates** 열에 있는 마지막 날짜를 기준으로 작년의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

```
PREVIOUSYEAR(<dates>[, <year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

날짜 값의 단일 열이 포함된 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 마지막 날짜를 기준으로 작년의 모든 날짜를 반환합니다. 예를 들어 **dates** 인수의 마지막 날짜가 2009년을 나타내는 경우 이 함수는 2008년의 모든 날짜를 지정된 **year_end_date** 까지 반환합니다.
- dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 **CALCULATE** 항목에 설명되어 있습니다.
- year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 작년 매출을 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey]))
```

참고 항목

시간 인텔리전스 함수

날짜 및 시간 함수

PREVIOUSMONTH

PREVIOUSDAY

PREVIOUSQUARTER

SAMEPERIODLASTYEAR

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 지정된 **dates** 열의 날짜로부터 1년 전으로 이동한 시점의 날짜로 구성된 열이 포함되어 있는 테이블을 반환합니다.

구문

```
SAMEPERIODLASTYEAR(<dates>)
```

매개 변수

용어	정의
dates	날짜가 포함된 열입니다.

반환 값

날짜 값으로 구성된 단일 열 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 반환되는 날짜는 상응하는 다음 수식에서 반환되는 날짜와 동일합니다. `DATEADD(dates, -1, year)`
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 재판매인 판매의 작년 판매를 계산하는 측정값을 만듭니다.

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSYEAR](#)

[PARALLELPERIOD](#)

STARTOFMONTH

2021-03-16 • 2 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 월의 첫 번째 날짜를 반환합니다.

구문

```
STARTOFMONTH(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 현재 컨텍스트에 대해 월의 시작을 반환하는 측정값을 만듭니다.

```
= STARTOFMONTH(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[STARTOFYEAR](#)

[STARTOFQUARTER](#)

STARTOFQUARTER

2021-03-16 • 2 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 분기의 첫 번째 날짜를 반환합니다.

구문

```
STARTOFQUARTER(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 현재 컨텍스트에서 분기 시작을 반환하는 측정값을 만듭니다.

```
= STARTOFQUARTER(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[STARTOFYEAR](#)

[STARTOFMONTH](#)

STARTOFYEAR

2021-03-16 • 2 minutes to read

지정된 날짜의 열에 대한 현재 컨텍스트에서 연도의 첫 번째 날짜를 반환합니다.

구문

```
STARTOFYEAR(<dates>)
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
YearEndDate	(선택 사항) 연도 종료 날짜 값입니다.

반환 값

날짜 값이 있는 단일 열과 단일 행을 포함하는 테이블입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식에서는 현재 컨텍스트에 대해 연도의 시작을 반환하는 측정값을 만듭니다.

```
= STARTOFYEAR(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[STARTOFQUARTER](#)

[STARTOFMONTH](#)

TOTALMTD

2021-03-16 • 2 minutes to read

현재 컨텍스트에 있는 월간 누계에 대해 식의 값을 계산합니다.

구문

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

매개 변수

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

날짜의 날짜를 기준으로 현재 월간 누계의 날짜에 대해 계산된 식을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- **filter** 식에는 [CALCULATE](#) 항목에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매에 대한 '월간 실행 누계' 또는 '월간 실행 합계'를 계산하는 측정값을 만듭니다.

```
= TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

참고 항목

[ALL](#)

[CALCULATE](#)

[TOTALYTD](#)

TOTALQTD

2021-03-16 • 2 minutes to read

현재 컨텍스트에서 분기 누계의 날짜에 대해 식의 값을 계산합니다.

구문

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

매개 변수

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

dates의 날짜가 주어질 경우, 현재 분기 누계의 모든 날짜에 대해 계산된 **expression**을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 **CALCULATE** 항목에 설명되어 있습니다.
- **filter** 식에는 **CALCULATE** 항목에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 샘플 수식은 인터넷 판매의 '분기 누계' 또는 '분기 누적 합계'를 계산하는 측정값을 만듭니다.

```
= TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

참고 항목

[ALL](#)

[CALCULATE](#)

TOTALYTD
TOTALMTD

TOTALYTD

2021-03-16 • 4 minutes to read

현재 컨텍스트에서 **expression** 의 연간 누계 값을 계산합니다.

구문

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

매개 변수

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.
year_end_date	(선택 사항) 연도 종료 날짜를 정의하는 날짜가 포함된 리터럴 문자열입니다. 기본값은 12월 31일입니다.

반환 값

현재 연간 누계 **dates** 에 대해 계산된 **expression** 을 나타내는 스칼라 값입니다.

설명

- **dates** 인수는 다음 중 하나일 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식의 제약 조건은 [CALCULATE](#) 항목에 설명되어 있습니다.
- **filter** 식에는 [CALCULATE](#) 항목에 설명된 제한 사항이 있습니다.
- **year_end_date** 매개 변수는 통합 문서가 만들어진 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 필요하지 않으며 무시됩니다. 예를 들어 다음 수식은 EN-US 로캘 통합 문서에 (fiscal) year_end_date를 6/30으로 지정합니다.

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey], ALL('DateTime'), "6/30")
```

이 예제에서 year_end_date는 "6/30", "Jun 30", "30 June" 또는 월/일로 확인되는 문자열로 지정할 수 있습니다. 그러나 문자열이 날짜로 확인되도록 하려면 표시된 대로 "월/일"을 사용하여 year_end_date를 지정하는 것이 좋습니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용하는 경우 DirectQuery 모드에서 사용할 수 없습니다.

니다.

예제

다음 샘플 수식은 인터넷 판매의 '연간 누계' 또는 '연간 누적 합계'를 계산하는 측정값을 만듭니다.

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

참고 항목

[ALL](#)

[CALCULATE](#)

[DATESYTD](#)

[TOTALMTD](#)

[TOTALQTD](#)

문

2021-03-16 • 2 minutes to read

이 범주의 함수

문	설명
DEFINE	(키워드) DAX 쿼리 기간에만 존재하는 엔티티를 정의합니다.
EVALUATE	(키워드) DAX 쿼리를 실행하는 데 필요한 문입니다.
ORDER BY	(키워드) DAX 쿼리의 결과를 정렬하기 위해 사용되는 하나 이상의 식을 정의합니다.
VAR	(키워드) 식의 결과를 명명된 변수로 저장합니다. 그런 다음, 이 변수를 다른 측정값 식에 인수로 전달할 수 있습니다.

정의

2021-03-16 • 2 minutes to read

DAX 쿼리의 하나 이상의 EVALUATE 문에 적용할 수 있는 엔터티를 정의하는 키워드입니다.

구문

```
DEFINE { <entity> [<name>] = <expression> }
```

인수

용어	정의
엔터티	MEASURE, VAR, TABLE 또는 COLUMN.
이름	엔터티의 이름입니다. 식이 될 수 없습니다.
expression	단일 스칼라 값을 반환하는 모든 DAX 식입니다. 식은 정의된 모든 엔터티를 사용할 수 있습니다. 식은 테이블을 반환해야 합니다. 스칼라 값이 필요하면 ROW() 함수 내에서 스칼라를 래핑하여 테이블을 생성합니다.

설명

- 엔터티는 변수, 측정값, 테이블 및 열일 수 있습니다.
- 정의는 일반적으로 EVALUATE 문 앞에 있으며 모든 EVALUATE 문에 대해 유효합니다.
- 정의는 현재 정의 앞 또는 뒤에 표시되는 다른 정의를 참조할 수 있습니다.
- 정의는 쿼리 기간 동안만 존재합니다.

참고 항목

[DAX 쿼리](#)

[ORDER BY](#)

[VAR](#)

평가

2021-03-16 • 2 minutes to read

DAX 쿼리에 필요한 테이블 식을 포함하는 문입니다.

구문

```
EVALUATE <table>
```

매개 변수

용어	정의
테이블	테이블 식

반환 값

테이블 식의 결과입니다.

설명

쿼리는 여러 EVALUATE 문을 포함할 수 있습니다.

예제

```
EVALUATE(  
    'Internet Sales'  
)
```

인터넷 판매 테이블에서 모든 행 및 열을 테이블로 반환합니다.

참고 항목

[DAX 쿼리](#)

[DEFINE](#)

[ORDER BY](#)

ORDER BY

2021-03-16 • 2 minutes to read

DAX 쿼리에서 EVALUATE 문이 반환하는 쿼리 결과의 정렬 순서를 정의합니다.

구문

```
ORDER BY {<expression> [{ASC | DESC}]}
```

인수

용어	정의
expression	단일 스칼라 값을 반환하는 모든 DAX 식입니다.
ASC	(기본값) 오름차순으로 정렬합니다.
DESC	내림차순으로 정렬합니다.

반환 값

EVALUATE 문의 결과는 ASC(오름차순) 또는 DESC(내림차순)입니다.

예제

```
EVALUATE(  
    'Internet Sales'  
)  
ORDER BY  
    'Internet Sales'[Order Date]
```

주문 날짜로 정렬된 인터넷 판매 테이블의 모든 행 및 열을 테이블로 반환합니다.

참고 항목

[DAX 쿼리](#)

[EVALUATE](#)

VAR

2021-03-16 • 5 minutes to read

식의 결과를 명명된 변수로 저장합니다. 그런 다음, 이 변수를 다른 측정값 식에 인수로 전달할 수 있습니다. 일단 변수 식에 대한 결과 값이 계산되면 변수가 다른 식에서 참조되더라도 해당 값은 변경되지 않습니다.

구문

```
VAR <name> = <expression>
```

매개 변수

용어	정의
name	변수(식별자)의 이름입니다. 구분 기호는 지원되지 않습니다. 예를 들어 'varName' 또는 [varName]에서는 오류가 발생합니다. 지원되는 문자 집합은 a~z, A~Z, 0~9입니다. 0~9는 첫 번째 문자로 유효하지 않습니다. (이중 밑줄)은 식별자 이름 접두사로 사용할 수 있습니다. 다른 특수 문자는 지원되지 않습니다. 예약 키워드는 사용할 수 없습니다. 기존 테이블의 이름은 사용할 수 없습니다. 빈 공백은 사용할 수 없습니다.
식	스칼라 또는 테이블 값을 반환하는 DAX 식입니다.

반환 값

식 인수 결과가 포함된 명명된 변수입니다.

설명

- VAR에 인수로 전달되는 식에는 다른 VAR 선언이 포함될 수 있습니다.
- 변수를 참조하는 경우 다음 사항에 유의합니다.
 - 측정값은 측정값 식 외부에 정의된 변수를 참조할 수 없지만, 식 내에 정의된 함수 범위 변수를 참조할 수 있습니다.
 - 변수는 측정값을 참조할 수 있습니다.
 - 변수는 이전에 정의된 변수를 참조할 수 있습니다.
 - 테이블 변수의 열은 TableName[ColumnName] 구문을 통해 참조할 수 없습니다.

예제

변수를 사용하지 않고 전년동기대비 성장률을 계산하기 위해 별도의 측정값 3개를 만들 수 있습니다. 첫 번째 측정값은 판매액 합계를 계산합니다.

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

두 번째 측정값은 작년 판매액을 계산합니다.

```
SalesAmount PreviousYear =  
    CALCULATE([Sum of SalesAmount],  
        SAMEPERIODLASTYEAR(Calendar[Date]))  
    )
```

그런 다음, 다른 두 측정값을 결합하는 세 번째 측정값을 만들어 성장률을 계산할 수 있습니다. SalesAmount 합계 측정값은 두 곳에서 사용됩니다. 먼저 판매가 있는지 확인하는 데 사용되고, 백분율을 계산하는 데 다시 사용됩니다.

```
Sum of SalesAmount YoY%: =  
    IF([Sum of SalesAmount] ,  
        DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of SalesAmount])  
    )
```

변수를 사용하면 동일한 결과를 계산하는 단일 측정값을 만들 수 있습니다.

```
YoY% = VAR Sales = SUM(SalesTable[SalesAmount])  
  
VAR SalesLastYear =  
    CALCULATE ( SUM ( SalesTable[SalesAmount] ), SAMEPERIODLASTYEAR ( 'Calendar'[Date] ) )  
  
return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

변수를 사용하면 동일한 결과를 얻으면서도 더 읽기 쉬운 방식으로 나타낼 수 있습니다. 또한 식의 결과가 선언 시에 변수에 저장됩니다. 변수를 사용하지 않으므로 식을 사용할 때마다 다시 계산하지 않아도 됩니다. 이렇게 하면 측정값의 성능을 개선할 수 있습니다.

DAX 용어집

2021-03-16 • 22 minutes to read

분석 쿼리

Power BI 시각적 개체는 '분석 쿼리'를 사용하여 데이터 모델을 쿼리합니다. 분석 쿼리는 다음 세 가지 단계를 사용하여 잠재적으로 대용량 데이터 볼륨 및 모델 복잡성을 줄이기 위해 노력합니다. 세 단계는 바로 필터링, 그룹화 및 요약입니다. 분석 쿼리는 보고서 시각적 개체의 웹에 필드가 할당될 때 자동으로 생성됩니다. 보고서 작성자는 필드 이름을 바꾸거나, 요약 기법을 수정하거나, 그룹화를 위해 요약을 사용하지 않도록 설정하여 필드 할당의 동작을 제어할 수 있습니다. 보고서 디자인 타임에 보고서, 보고서 페이지 또는 시각적 개체에 필터를 추가할 수 있습니다. 읽기용 보기에서는 필터 창에서 또는 슬라이서 및 기타 시각적 개체(교차 필터링)와의 상호 작용에 의해 필터가 수정될 수 있습니다.

BLANK

DAX는 값이 없는 경우를 BLANK로 정의합니다. 이는 SQL NULL에 해당하지만 정확히 동일하게 동작하지는 않습니다. Excel에서 빈 셀을 정의하는 방법과 좀 더 유사합니다. BLANK는 다른 연산과 결합될 경우 0 또는 빈 문자열로 평가됩니다. 예를 들어 $BLANK + 20 = 20$ 입니다. 항상 대문자를 사용합니다. 복수는 BLANKs입니다(소문자 "s" 포함).

계산 열

DAX 수식을 작성하여 테이블 형식 모델에 열을 추가하는 데 사용되는 모델 계산입니다. 수식은 스칼라 값을 반환해야 하며 테이블의 각 행에 대해 평가됩니다. 계산 열은 가져오기 또는 DirectQuery 스토리지 모드 테이블에 추가할 수 있습니다.

계산 측정값

테이블 형식 모델링에서는 '계산 측정값' 같은 개념이 없습니다. 대신 '측정값'을 사용하세요. 계산이라는 단어가 계산된 테이블 및 계산 열을 설명하기 위해 사용됩니다. 파워 쿼리에서 시작되는 테이블 및 열과 구분됩니다. 파워 쿼리에는 측정값의 개념이 없습니다.

계산된 테이블

DAX 수식을 작성하여 테이블 형식 모델에 테이블을 추가하는 데 사용되는 모델 계산입니다. 수식은 테이블 개체를 반환해야 합니다. 그러면 가져오기 스토리지 모드를 사용하는 테이블이 생성됩니다.

계산

하나 이상의 입력을 하나 이상의 결과로 변환하는 고의적인 프로세스입니다. 표 형식 데이터 모델에서 계산은 모델 개체, 즉 계산된 테이블, 계산 열 또는 측정값 중 하나입니다.

Context

DAX 수식이 평가되는 환경에 대해 설명합니다. 두 가지 형식의 컨텍스트가 있습니다. 바로 '행 컨텍스트'와 '필터 컨텍스트'입니다. 행 컨텍스트는 "현재 행"을 나타내며 테이블 반복기에서 사용되는 계산 열 수식과 식을 평가하는 데 사용됩니다. 필터 컨텍스트는 측정값을 평가하는 데 사용되며, 모델 관계에 의해 전파된 필터 및 모델 열에 직접 적용되는 필터를 나타냅니다.

큐브

[다차원 모델](#)을 참조하세요.

데이터 모델

보고 및 분석을 위해 특별히 준비된 데이터 리소스입니다. 이 리소스를 통해 보고서 사용자는 직관적이고 단순하게 데이터를 찾아보고 탐색할 수 있습니다. 중요한 점은 대용량 데이터 볼륨에서도 고성능 쿼리 결과를 제공한다는 것입니다. 여러 원본의 데이터를 통합하고 계산을 사용하여 데이터를 변환할 수 있습니다. 데이터 모델은 행 수준 사용 권한을 적용하여 다양한 사용자가 서로 다른 데이터에 액세스하도록 할 수 있습니다. 경우에 따라 '의미 체계 모델' 또는 단순히 '모델'이라고도 합니다.

데이터 모델러

데이터 모델을 만드는 숙련된 데이터 전문가입니다. SSBI(셀프 서비스 BI)에서는 비즈니스 분석가라고 할 수 있습니다. 회사 BI에서는 BI 개발자라고 할 수 있습니다.

DAX

DAX(Data Analysis Expressions) 언어는 Excel의 파워 피벗, Power BI, Azure Analysis Services, 그리고 SQL Server Analysis Services의 테이블 형식 모델링을 위한 수식 언어입니다. DAX를 사용하여 데이터 모델 계산을 추가하고 RLS(행 수준 보안) 규칙을 정의할 수도 있습니다.

동적 보안

보고서 사용자의 ID를 통해 RLS(행 수준 보안) 규칙이 적용될 때 사용됩니다. 규칙은 사용자 계정 이름을 사용하여 모델 테이블을 필터링하며, 이는 USERNAME 또는 USERPRINCIPALNAME 함수로 수행할 수 있습니다. [행 수준 보안](#)을 참조하세요.

식

평가되어 결과를 반환하는 DAX 논리 단위입니다. 식은 변수를 선언할 수 있으며 이 경우 하위 식이 할당됩니다. 또한 식은 최종 식을 출력하는 RETURN 문을 포함해야 합니다. 식은 모델 개체(테이블, 열 또는 측정값), 함수, 연산자 또는 상수를 사용하여 생성됩니다.

필드

필드 창에 표시되는 데이터 모델 리소스입니다. 필드는 보고서 필터 및 시각적 개체를 구성하는 데 사용됩니다. 필드는 모델 열, 계층 구조 수준 및 측정값으로 구성됩니다.

수식

모델 계산을 정의하는 데 사용되는 하나 이상의 DAX 식입니다. 내부 식을 하위 식이라고 합니다. Formula(수식)의 복수형은 *formulas* 입니다.

함수

DAX 함수에는 매개 변수 전달을 허용하는 인수가 있습니다. 수식은 많은 함수 호출을 사용할 수 있으며, 다른 함수 내에 함수를 중첩할 수 있습니다. 수식에서 함수 이름 뒤에는 괄호가 와야 합니다. 괄호 안에 매개 변수가 전달됩니다.

암시적 측정값

열 값을 요약하도록 Power BI 시각적 개체를 구성하여 자동으로 생성된 계산입니다. 숫자 열은 가장 넓은 범위

의 요약을 지원하며 합계, 평균, 최솟값, 최댓값, 개수(고유), 개수, 표준 편차, 분산, 중앙값 등을 포함합니다. 다른 데이터 형식의 열도 요약할 수 있습니다. **텍스트** 열은 처음(사전순), 마지막(사전순), 개수(고유) 또는 개수를 사용하여 요약할 수 있습니다. **날짜** 열은 가장 이른 날짜, 가장 늦은 날짜, 개수(고유) 또는 개수를 사용하여 요약할 수 있습니다. **부울** 열은 개수(고유) 또는 개수를 사용하여 요약할 수 있습니다.

반복기 함수

지정된 테이블의 모든 행을 열거하고 각 행에 대해 지정된 식을 평가하는 DAX 함수입니다. 모델 계산이 데이터를 요약하는 방식을 유연하게 제어할 수 있습니다.

MDX

MDX(Multidimensional Expressions) 언어는 SQL Server Analysis Services 다차원 모델('큐브'라고도 함)용 수식 언어입니다. MDX는 테이블 형식 모델을 쿼리하는 데 사용할 수 있지만, 암시적 측정값을 정의할 수는 없습니다. 모델에 이미 정의되어 있는 측정값만 쿼리할 수 있습니다.

측정값

요약을 작성하는 계산입니다. 측정값은 '**암시적**'이거나 '명시적'입니다. 명시적 측정값은 DAX 수식을 작성하여 표 형식 데이터 모델에 추가되는 계산입니다. 측정값 수식은 스칼라 값을 반환해야 합니다. 필드 창에서 명시적 측정값은 계산기 아이콘으로 표시됩니다. Excel에서 분석을 사용할 때처럼, MDX(Multidimensional Expressions)를 사용하여 모델을 쿼리할 때에도 명시적 측정값이 필요합니다. 명시적 측정값을 일반적으로 그냥 측정값이라고 합니다.

측정값 그룹

하나 이상의 측정값을 포함하며, 계층 구조 또는 보이는 열이 없는 모델 테이블입니다. 필드 창에서 각 측정값 그룹은 여러 계산기 아이콘으로 표시됩니다. 측정값 그룹은 필드 창의 맨 위에 나열되며 이름에 따라 사전순으로 정렬됩니다.

모델

[데이터 모델](#)을 참조하세요.

모델러

[데이터 모델러](#)를 참조하세요.

모델 계산

계산된 테이블, 계산 열 또는 측정값을 표 형식 데이터 모델에 추가하는 데 사용되는 명명된 수식입니다. 구조는 <NAME> = <FORMULA>입니다. 데이터 모델러가 Power BI Desktop에서 대부분의 계산을 추가하지만, 라이브 연결 보고서에 측정값이 추가될 수도 있습니다. [보고서 측정값](#)을 참조하세요.

다차원 모델

SQL Server Analysis Services(다차원 모드)용으로 개발된 데이터 모델입니다. 차원과 측정값으로 구성됩니다. 종종 간단하게 '큐브'라고도 부릅니다.

빠른 측정

일반적으로 정의된 측정값에 대한 DAX 수식을 작성하지 않아도 되는 Power BI Desktop의 기능입니다. 빠른 측정에는 범주당 평균, 순위 및 기준선과의 차이가 포함됩니다.

보고서 측정값

‘보고서 수준 측정값’이라고도 합니다. 보고서 측정값은 DAX 수식 작성을 통해 Power BI Desktop의 라이브 연결 보고서에 추가되지만, Power BI 모델 또는 Analysis Services 테이블 형식 모델에 대한 연결에만 해당됩니다.

행 수준 보안

*RLS*라고도 합니다. 특정 사용자의 일부 데이터에 대한 액세스를 제한하는 디자인 기법입니다. 테이블 형식 모델에서는 모델 역할을 만들어 수행할 수 있습니다. 역할에는 테이블 행을 필터링하는 DAX 식인 규칙이 있습니다.

의미 체계 모델

[데이터 모델](#)을 참조하세요.

요약

열의 값에 적용되는 연산입니다. [측정값](#)을 참조하세요.

테이블 형식 큐브

‘테이블 형식 큐브’와 같은 개념은 없습니다. 대신 [테이블 형식 모델](#)을 사용하세요.

테이블 형식 모델

Excel의 파워 피벗, Power BI, Azure Analysis Services 또는 SQL Server Analysis Services(테이블 형식 모드)에서 개발된 데이터 모델입니다.

시간 인텔리전스

시간 인텔리전스는 YTD(연간 누계)처럼 시간 경과에 따른 계산과 관련이 있습니다.

시간 인텔리전스 함수

DAX에는 다양한 시간 인텔리전스 함수가 포함되어 있습니다. 각 시간 인텔리전스 함수는 날짜 필터에 대한 필터 컨텍스트를 수정하여 결과를 얻습니다. 예제 함수: TOTALYTD 및 SAMEPERIODLASTYEAR.

값

시각화할 데이터입니다.

가상 매개 변수

슬라이서를 통해 사용자 입력을 허용하는 기능을 제공하는 Power BI Desktop 기능입니다. 각 매개 변수는 단일 열 계산 테이블 및 단일 선택 값을 반환하는 측정값을 만듭니다. 측정값은 모델 계산에서 사용자의 입력에 응답하는 데 사용될 수 있습니다.

DAX 연산자

2021-03-16 • 19 minutes to read

DAX(Data Analysis Expression) 언어는 연산자를 사용하여 값을 비교하거나 산술 계산을 수행하거나 문자열에 대한 작업을 수행하는 식을 만듭니다.

연산자 유형

계산 연산자에는 산술, 비교, 텍스트 연결 및 논리라는 네 가지 유형이 있습니다.

산술 연산자

더하기, 빼기 또는 곱하기와 같은 기본 수치 연산을 수행하는 경우 다음 산술 연산자를 사용하여 숫자를 결합하고 수치 결과를 생성합니다.

산술 연산자	의미	예제
+(더하기 기호)	더하기	3+3
-(빼기 기호)	빼기 또는 기호	3-1-1
*(별표)	곱하기	3*3
/(슬래시)	나누기	3/3
^(캐럿)	지수	16^4

NOTE

더하기 기호는 이항 연산자와 단항 연산자로 작동 할 수 있습니다. 이항 연산자는 연산자의 양쪽 모두에 숫자가 필요하고 더하기를 수행합니다. 이항 연산자의 양쪽에서 DAX 수식에 값을 사용하는 경우 값이 숫자가 아니면 DAX는 값을 숫자 데이터 형식으로 캐스트하려고 합니다. 반면에 단항 연산자는 모든 형식의 인수에 적용될 수 있습니다. 더하기 기호는 형식이나 값에 영향을 주지 않고 간단히 무시되지만 빼기 연산자는 숫자 값에 적용되는 경우 음수 값을 만듭니다.

비교 연산자

다음 연산자를 사용하여 두 값을 비교할 수 있습니다. 이러한 연산자를 사용하여 두 값을 비교하면 결과는 논리 값(TRUE 또는 FALSE)입니다.

비교 연산자	의미	예제
=	같음	[Region] = "USA"
==	엄격한 같음	[Region] == "USA"
>	보다 큼	[Sales Date] > "Jan 2009"
<	보다 작음	[Sales Date] < "Jan 1 2009"
>=	크거나 같음	[Amount] >= 20000

비교 연산자	의미	예제
<=	작거나 같음	[Amount] <= 100
<>	같지 않음	[Region] <> "USA"

==를 제외한 모든 비교 연산자는 BLANK를 숫자 0, 빈 문자열 "", DATE(1899, 12, 30) 또는 FALSE로 간주합니다. 따라서, [Column] = 0은 [Column]의 값이 0 또는 BLANK이면 true입니다. 반면에 [Column] == 0은 [Column]의 값이 0인 경우에만 true입니다.

텍스트 연결 연산자

앰퍼샌드(&)를 사용하여 둘 이상의 텍스트 문자열을 조인 또는 연결하여 단일 텍스트를 생성합니다.

텍스트 연산자	의미	예제
&(앰퍼샌드)	두 값을 연결하여 하나의 연속 텍스트 값을 생성합니다.	[Region] & ", " & [City]

논리 연산자

논리 연산자(&&) 및(||)를 사용하여 식을 결합하여 단일 결과를 생성합니다.

텍스트 연산자	의미	예
&&(이중 앰퍼샌드)	각각 부울 결과가 있는 두 식 사이에 AND 조건을 만듭니다. 두 식이 모두 TRUE를 반환하면 식의 조합도 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.	(([Region] = "France") && ([BikeBuyer] = "yes"))
(이중 파이프 기호)	두 논리 식 사이에 OR 조건을 만듭니다. 두 식이 모두 TRUE를 반환하면 결과는 TRUE입니다. 두 식이 모두 FALSE인 경우에만 결과가 FALSE입니다.	((([Region] = "France") ([BikeBuyer] = "yes"))
IN	테이블과 비교할 각 행 사이에 논리적 OR 조건을 만듭니다. 참고: 테이블 생성자 구문은 중괄호를 사용합니다.	'Product'[Color] IN { "Red", "Blue", "Black" }

연산자 및 우선 순위

경우에 따라 계산이 수행되는 순서가 반환 값에 영향을 줄 수 있습니다. 따라서 순서가 결정되는 방식과 원하는 결과를 얻기 위해 순서를 변경하는 방법을 이해하는 것이 중요합니다.

계산 순서

식은 연산자와 값을 특정 순서로 계산합니다. 모든 식은 항상 등호(=)로 시작합니다. 등호는 뒤에 오는 문자가 식을 구성한다는 것을 나타냅니다.

등호 뒤에는 계산되는 요소(피연산자)가 계산 연산자로 구분되어 있습니다. 식은 항상 왼쪽에서 오른쪽으로 읽지만 요소를 그룹화하는 순서는 괄호를 사용하여 어느 정도 제어할 수 있습니다.

연산자 우선 순위

단일 수식에 여러 연산자를 결합하는 경우 연산되는 순서는 다음 표에 따라 정해집니다. 연산자의 우선 순위 값이 같으면 왼쪽에서 오른쪽으로 순서가 정해집니다. 예를 들어 식에 곱하기와 나누기 연산자가 모두 포함되어 있으면, 식에 표시되는 순서대로 왼쪽에서 오른쪽으로 계산됩니다.

연산자	설명
^	지수
-	기호(-1과 동일)
* 및 /	곱하기 및 나누기
!	NOT(단항 연산자)
+ 및 -	더하기 및 빼기
&	두 개의 텍스트 문자열을 연결합니다.
=,=, <, >, <=, >=, <>	비교

괄호를 사용하여 계산 순서 제어

계산 순서를 변경하려면 먼저 계산해야 하는 수식의 부분을 괄호로 묶어야 합니다. 예를 들어 다음 수식은 더하기 전에 곱하기가 계산되기 때문에 11을 생성합니다. 이 수식은 2에 3을 곱하고 그 결과에 5를 더합니다.

```
=5+2*3
```

그에 반해, 괄호를 사용하여 구문을 변경하면 순서가 변경되어 5와 2를 더하고 그 결과에 3을 곱하여 21을 생성합니다.

```
=(5+2)*3
```

다음 예는 수식의 첫 번째 부분을 괄호로 묶어서 `(3 + 0.25)` 식이 먼저 계산되도록 하고 이 결과를 `3 - 0.25` 식의 결과로 나누도록 합니다.

```
=(3 + 0.25)/(3 - 0.25)
```

다음 예는 연산자의 우선 순위 규칙에 따라 지수 연산자가 먼저 적용된 다음, 부호 연산자가 적용됩니다. 이 식의 결과는 -4입니다.

```
=-2^2
```

기호 연산자가 숫자 값에 먼저 적용되도록 하려면 다음 예제와 같이 괄호를 사용하여 연산자를 제어하면 됩니다. 이 식의 결과는 4입니다.

```
= (-2)^2
```

호환성

DAX는 Microsoft Excel과 같이 다양한 데이터 형식을 쉽게 처리하고 비교합니다. 하지만 기본 계산 엔진은 SQL Server Analysis Services를 기반으로 하며 관계형 데이터 저장소의 고급 기능을 추가로 제공합니다. 여기에는 날짜 및 시간 형식에 대한 다양한 지원이 포함됩니다. 따라서 경우에 따라 계산 결과나 함수의 동작이 Excel과 같지 않을 수 있습니다. 또한 DAX는 Excel 보다 많은 데이터 형식을 지원합니다. 이 섹션에서는 주요 차이점에 대해 설명합니다.

피연산자의 데이터 형식 강제 변환

일반적으로 연산자의 왼쪽과 오른쪽에 있는 피연산자 두 개는 동일한 데이터 형식이어야 합니다. 하지만 데이터 형식이 다르면, DAX는 데이터 형식을 공통 데이터 형식으로 변환하여 연산자를 적용합니다. 경우에 따라:

1. 양쪽 피연산자가 가장 큰 가능한 공통 데이터 형식으로 변환됩니다.
2. 가능한 경우 연산자가 적용됩니다.

예를 들어 두 개의 숫자를 결합하려고 한다고 가정합니다. 한 숫자는 `= [Price] * .20` 수식의 결과이며 소수점 이하 자릿수가 많이 포함될 수 있습니다. 다른 숫자는 문자열 값으로 제공된 정수입니다.

이 경우 DAX는 두 종류의 숫자를 모두 저장할 수 있는 가장 큰 숫자 형식을 사용하여 두 숫자를 숫자 형식의 실수로 변환합니다. 그런 다음, DAX는 곱셈을 적용합니다.

데이터 형식 조합에 따라서는 형식 강제 변환이 비교 연산에 적용되지 않을 수 있습니다. DAX에서 지원되는 데이터 형식의 전체 목록은 [테이블 형식 모델에서 지원되는 데이터 형식](#) 및 [Power BI Desktop의 데이터 형식](#)을 참조하세요.

정수, 실수, 통화, 날짜/시간 및 공백은 비교 시 숫자로 간주됩니다. 공백은 비교를 수행할 때 0으로 계산됩니다. 비교 작업에 대해 지원되는 데이터 형식 조합은 다음과 같습니다.

왼쪽 데이터 형식	오른쪽 데이터 형식
숫자	숫자
Boolean	Boolean
String	String

다른 혼합 데이터 형식 비교는 오류를 반환합니다. 예를 들어 `"1" > 0`과 같은 수식은 DAX 비교 연산은 텍스트 형식 값과 정수 형식 값의 비교를 지원하지 않는다는 오류를 반환합니다.

DAX에 사용되는 데이터 형식	EXCEL에 사용되는 데이터 형식
숫자(I8, R8)	숫자(R8)
String	String
Boolean	Boolean
DateTime	변형
통화	통화

우선 순위 차이

DAX 수식의 연산 우선 순위는 기본적으로 Microsoft Excel에 사용되는 순서와 동일하지만 일부 Excel 연산자(예: 퍼센트)는 지원되지 않습니다. 또한 범위는 지원되지 않습니다.

따라서 Excel에서 수식을 복사하여 붙여 넣을 때마다 수식을 신중하게 검토해야 합니다. 수식의 일부 연산자나 요소가 유효하지 않을 수 있기 때문입니다. 연산 수행 순서가 확실하지 않을 경우 괄호를 사용하여 연산 순서를 제어하고 결과에 대한 모호성을 제거하는 것이 좋습니다.

참고 항목

[DAX 구문](#)

[DAX 매개 변수 명명 규칙](#)

DAX 쿼리

2021-03-16 • 11 minutes to read

DAX 쿼리를 사용하면 테이블 식으로 정의된 데이터를 쿼리하고 반환할 수 있습니다. 보고 클라이언트는 필드가 보고서 표면에 배치되거나 필터 또는 계산이 적용될 때마다 DAX 쿼리를 생성합니다. 또한 [SSMS\(SQL Server Management Studio\)](#) 및 open-source tools like [DAX Studio](#)와 같은 오픈 소스 도구에서 DAX 쿼리를 만들고 실행할 수 있습니다. DAX 쿼리는 SSMS에서 실행되고 DAX Studio는 결과를 테이블로 반환합니다.

쿼리를 학습하기 전에 DAX 기본 사항을 확실하게 이해하는 것이 중요합니다. 아직 확실히 이해하지 못했다면, [DAX 개요](#)를 확인해야 합니다.

구문

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }
      { VAR <name> = <expression>}}]
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value>|<parameter>} [, ...]]]
```

키워드

EVALUATE(필수)

가장 기본적으로 DAX 쿼리는 테이블 식이 포함된 **EVALUATE** 문입니다. 하지만 쿼리는 여러 EVALUATE 문을 포함할 수 있습니다.

구문

```
EVALUATE <table>
```

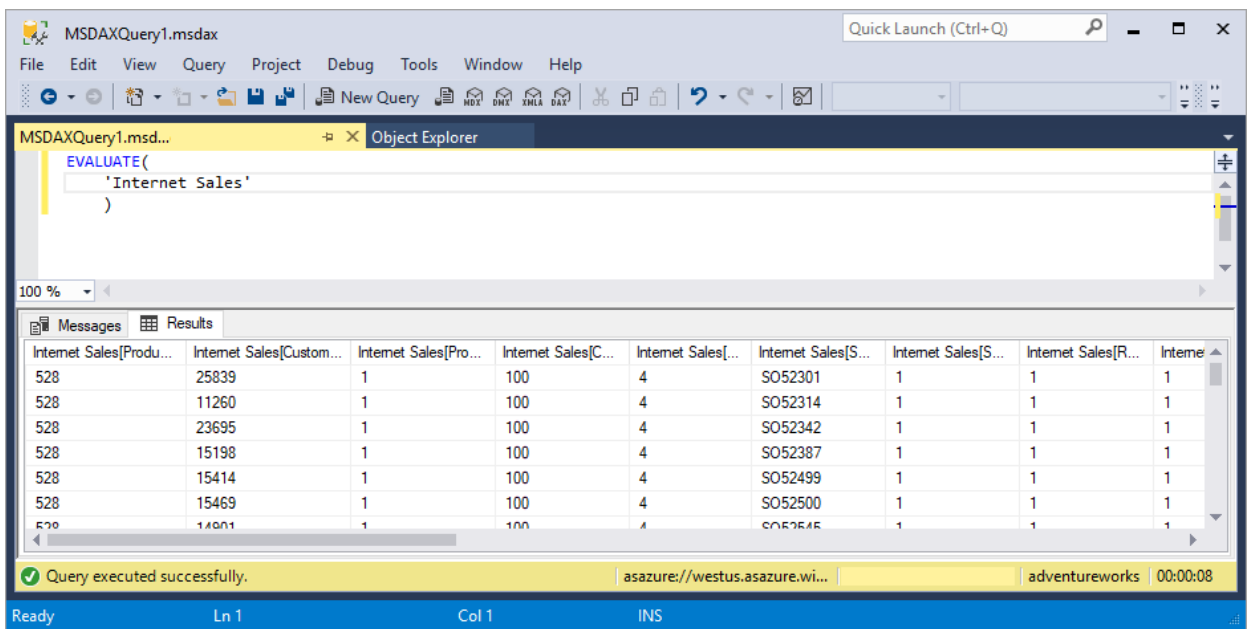
인수

용어	정의
테이블	테이블 식.

예제

```
EVALUATE(
    'Internet Sales'
)
```

인터넷 판매 테이블에서 모든 행 및 열을 테이블로 반환합니다.



ORDER BY(선택 사항)

선택적인 **ORDER BY** 키워드는 쿼리 결과를 정렬하기 위해 사용되는 하나 이상의 식을 정의합니다. 각 결과 행에 대해 평가될 수 있는 모든 식이 유효합니다.

구문

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
```

인수

용어	정의
expression	단일 스칼라 값을 반환하는 모든 DAX 식입니다.
ASC	(기본값) 오름차순으로 정렬합니다.
DESC	내림차순으로 정렬합니다.

예제

```
EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Order Date]
```

주문 날짜로 정렬된 인터넷 판매 테이블의 모든 행 및 열을 테이블로 반환합니다.

The screenshot shows the MSDAX Query tool interface. The query editor contains the following code:

```

EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Order Date]
  
```

The results pane displays a table with the following columns: Internet Sales[Sales Order Number], Internet Sales[Total Sales Amount], Internet Sales[Total Quantity], Internet Sales[Total Volume], Internet Sales[Total Weight], Internet Sales[Order Date], Internet Sales[Order Date], Internet Sales[Order Date], and Internet Sales[Order Date]. The data is sorted by Order Date in descending order.

Internet Sales[Sales Order Number]	Internet Sales[Total Sales Amount]	Internet Sales[Total Quantity]	Internet Sales[Total Volume]	Internet Sales[Total Weight]	Internet Sales[Order Date]	Internet Sales[Order Date]	Internet Sales[Order Date]	Internet Sales[Order Date]
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	1487.835
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	1406.975
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	1487.835
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	1487.835
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	285.9515
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	1487.835
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	1476.895
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	1406.975

The status bar at the bottom indicates: Query executed successfully. asazure://westus.asazure.wi... adventureworks 00:00:08

START AT(선택 사항)

선택적인 **START AT** 키워드는 **ORDER BY** 절 내부에서 사용됩니다. 쿼리 결과가 시작되는 값을 정의합니다.

구문

```

EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value><parameter>} [, ...]]
  
```

인수

용어	정의
값	상수 값입니다. 식이 될 수 없습니다.
매개 변수	XMLA 문에서 @ 문자가 접두어로 표시된 매개 변수의 이름입니다.

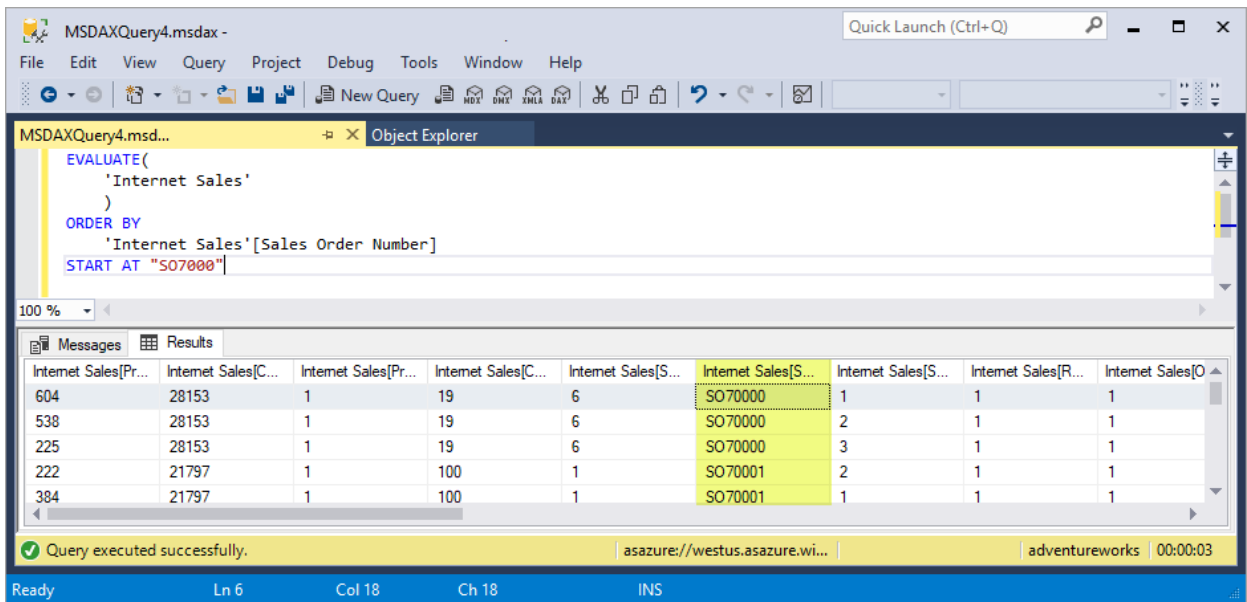
START AT 인수는 **ORDER BY** 절의 열과 일대일 대응을 갖습니다. **ORDER BY** 절과 같이 **START AT** 절에 여러 인수가 사용될 수 있지만, **ORDER BY** 절보다 많을 수 없습니다. **START AT**의 첫 번째 인수는 **ORDER BY** 열의 열 1에서 시작 값을 정의합니다. **START AT**의 두 번째 인수는 열 1의 첫 번째 값을 충족하는 행 내에서 **ORDER BY** 열의 열 2에 있는 시작 값을 정의합니다.

예제

```

EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Sales Order Number]
START AT "SO7000"
  
```

SO7000에서 시작하여 판매 주문 번호로 정렬된 인터넷 판매 테이블의 모든 행 및 열을 반환합니다.



여러 EVALUATE/ORDER BY/START AT 절을 단일 쿼리로 지정할 수 있습니다.

DEFINE(선택 사항)

선택적인 **DEFINE** 키워드는 쿼리 기간에만 존재하는 엔터티를 정의합니다. 정의는 모든 EVALUATE 문에 대해 유효합니다. 엔터티는 변수, 측정값, 테이블 및 열일 수 있습니다. 정의는 현재 정의 앞 또는 뒤에 표시되는 다른 정의를 참조할 수 있습니다. 일반적으로 정의는 EVALUATE 문 앞에 옵니다.

구문

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }
{ VAR <name> = <expression>}]
EVALUATE <table>
```

인수

용어	정의
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 식이 될 수 없습니다.
name	새 측정값의 이름입니다. 식이 될 수 없습니다.
expression	단일 스칼라 값을 반환하는 모든 DAX 식입니다. 식은 정의된 모든 측정 값을 사용할 수 있습니다. 식은 테이블을 반환해야 합니다. 스칼라 값이 필요하면 ROW() 함수 내에서 스칼라를 래핑하여 테이블을 생성합니다.
VAR	명명된 변수인 선택적인 식입니다. VAR 은 다른 식에 인수로 전달될 수 있습니다.

예제

```

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]

```

2013년과 2014년에 대해 계산된 총 판매액과 2013년 및 2014년에 해당하는 결합 계산된 총 판매액을 하나의 테이블로 반환합니다. DEFINE 문의 측정값인 인터넷 총 판매액은 총 판매액 및 결합 연도 총 판매액 식에 모두 사용됩니다.

The screenshot shows the SSDT interface with a DAX query in the Object Explorer. The query is as follows:

```

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]

```

The Results pane displays the following data:

Date[Calendar Y...	[Total Sales]	[Combined Years...
2013	16351550.34	16397245.06
2014	45694.72	16397245.06

The status bar at the bottom indicates "Query executed successfully." and "adventureworks | 00:00:01".

DAX 쿼리의 매개 변수

잘 정의된 DAX 쿼리 문을 매개 변수화하고 매개 변수 값만 바꿔서 계속해서 사용할 수 있습니다.

[Execute Method\(XMLA\)](#) 메서드에는 매개 변수를 정의하고 값을 할당할 수 있게 해주는 [Parameters Element\(XMLA\)](#) 컬렉션 요소가 있습니다. 컬렉션 내에서 각 [Parameter Element\(XMLA\)](#) 요소는 매개 변수의 이름 및 이에 대한 값을 정의합니다.

매개 변수 이름에 `@` 문자를 접두어로 사용하여 XMLA 매개 변수를 참조합니다. 값이 허용되는 구문 어디서나 매개 변수 호출로 값이 대체될 수 있습니다. 모든 XMLA 매개 변수는 텍스트로 입력됩니다.

IMPORTANT

매개 변수 섹션에 정의되었고 **<STATEMENT>** 요소에 사용되지 않은 매개 변수는 XMLA로 오류 응답을 생성합니다. 사용되었지만 **<Parameters>** 요소에 정의되지 않은 매개 변수는 XMLA에서 오류 응답을 생성합니다.

참고 항목

[FILTER](#)

[SUMMARIZECOLUMNS](#)

[TREATAS](#)

DAX 매개 변수-명명 규칙

2021-03-16 • 4 minutes to read

매개 변수 이름은 함수의 사용 및 이해를 용이하게 하기 위해 DAX 참조에서 표준화됩니다.

매개 변수 이름

용어	정의
expression	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.
value	단일 스칼라 값을 반환하는 DAX 식입니다. 식은 다른 모든 작업 이전에 정확히 한 번 계산됩니다.
테이블	데이터 테이블을 반환하는 모든 DAX 식입니다.
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 식이 될 수 없습니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름으로, 일반적으로 정규화된 이름입니다. 식이 될 수 없습니다.
name	새 개체의 이름을 제공하는 데 사용되는 문자열 상수입니다.
순서	정렬 순서를 결정하는 데 사용되는 열거형입니다.
ties	동률 값의 처리를 결정하는 데 사용되는 열거형입니다.
형식	PathItem 및 PathItemReverse에 대한 데이터 형식을 결정하는 데 사용되는 열거형입니다.

매개 변수 이름 접두사 사용 또는 접두사만 사용

용어	정의
prefixing	<p>매개 변수 이름은 인수 사용 방법을 설명하고 매개 변수의 모호한 판독을 방지하기 위해 접두사를 사용하여 추가로 한정할 수 있습니다. 예:</p> <p>Result_ColumnName - LOOKUPVALUE() 함수의 결과 값을 가져오는 데 사용되는 기존 열을 참조합니다.</p> <p>Search_ColumnName - LOOKUPVALUE() 함수의 값을 검색하는 데 사용되는 기존 열을 참조합니다.</p>

용어	정의
omitting	<p>접두사가 매개 변수를 설명하기에 충분한 경우 매개 변수 이름이 생략됩니다.</p> <p>예를 들어 다음 구문 DATE(Year_Value, Month_Value, Day_Value)를 사용하는 대신 사용자가 DATE(연도, 월, 일)를 읽는 것이 더 명확합니다. 접미사 값을 세 번 반복해도 함수를 더 잘 이해하는 데 도움이 되지 않으며 판독을 불필요하게 혼란스럽게 합니다.</p> <p>그러나 접두사가 지정된 매개 변수가 Year_columnName인 경우, 매개 변수 이름과 접두사를 통해 사용자는 매개 변수에 기존 연도의 열에 대한 참조가 필요하다는 점을 이해할 수 있습니다.</p>

DAX 구문

2021-03-16 • 25 minutes to read

이 문서에서는 DAX 수식 언어의 구문과 요구 사항에 대해 설명합니다.

구문 요구 사항

DAX 수식은 항상 등호(=)로 시작합니다. 등호 뒤에 스칼라로 계산되는 식이나 스칼라로 변환할 수 있는 식을 제공할 수 있습니다. 포함되는 내용은 다음과 같습니다.

- 스칼라 상수 또는 스칼라 연산자(+,-,*,/,>=,....&&, ...)를 사용하는 식
- 열 또는 테이블에 대한 참조. DAX 언어는 함수에 대한 입력으로 항상 테이블 및 열을 사용하지만 배열이나 임의의 값 집합은 사용하지 않습니다.
- 식의 일부로 제공되는 연산자, 상수 및 값.
- 함수의 결과와 필수 인수. 일부 DAX 함수는 스칼라 대신 테이블을 반환하며, 테이블을 평가하고 스칼라를 반환하는 함수로 묶어야 합니다. 테이블이 단일 열, 단일 행 테이블이 아니면 스칼라 값으로 처리됩니다.

대부분의 DAX 함수에는 하나 이상의 인수가 필요하며, 여기에는 테이블, 열, 식 및 값을 포함할 수 있습니다. 단, PI와 같은 일부 함수에는 인수가 필요하지 않지만 null 인수를 나타내기 위해 항상 괄호가 필요합니다. 예를 들어 PI가 아니라 항상 PI()를 입력해야 합니다. 다른 함수 내에 함수를 중첩할 수도 있습니다.

- 식. 식에는 연산자, 상수 또는 열에 대한 참조 중 일부 또는 전부가 포함될 수 있습니다.

예를 들어 다음은 모두 유효한 수식입니다.

FORMULA	결과
= 3	3
= 'Sales'	Sales
= 'Sales'[Amount]	Sales 테이블 내에서 이 수식을 사용하면, 현재 행에 대해 Sales 테이블의 Amount 열 값을 가져옵니다.
= (0.03 * [Amount]) =0.03 * [Amount]	현재 테이블의 Amount 열에 있는 값의 3%. 이 수식은 백분율을 계산하는 데 사용할 수 있지만 테이블에 서식을 적용하지 않으면 결과가 백분율로 표시되지 않습니다.
= PI()	상수 pi의 값.

NOTE

수식은 계산 열에 사용되는지 또는 피벗 테이블 내의 측정값에 사용되는지에 따라 다르게 동작할 수 있습니다. 컨텍스트를 항상 인식하고 수식에 사용하는 데이터가 계산에 사용될 수 있는 다른 데이터와 어떻게 관련되어 있는지를 항상 알고 있어야 합니다.

명명 요구 사항

데이터 모델에는 여러 테이블이 포함되는 경우가 많습니다. 테이블과 여기에 포함된 열은 메모리 내 분석 엔진 (VertiPaq)에 저장된 데이터베이스를 구성합니다. 해당 데이터베이스 내에서 모든 테이블의 이름은 고유해야 합니다. 열 이름도 각 테이블 내에서 고유해야 합니다. 모든 개체 이름은 대/소문자를 구분하지 않습니다. 예를 들어, **SALES** 와 **Sales** 라는 이름은 동일한 테이블을 나타냅니다.

기존 데이터 모델에 추가하는 각 열과 측정 값은 특정 테이블에 속해야 합니다. 테이블 내에 계산 열을 만드는 경우에는 암시적으로 또는 측정값을 만들고 측정값 정의를 저장할 테이블의 이름을 지정하는 경우에는 명시적으로, 열을 포함하는 테이블을 지정합니다.

테이블 또는 열을 함수에 대한 입력으로 사용하는 경우 열 이름을 일반적으로 정규화(*qualify*)해야 합니다. 열의 정규화된(*fully qualified*) 이름은 테이블 이름 뒤에 열 이름이 대괄호로 묶인 상태입니다. 예: 'U.S. Sales'[Products]. 다음 컨텍스트에서 열을 참조하는 경우에는 항상 정규화된 이름이 필요합니다.

- 함수에 대한 인수로, VALUES
- 함수에 대한 인수로, ALL 또는 ALLEXCEPT
- 함수에 대한 필터 인수에, CALCULATE 또는 CALCULATETABLE
- 함수에 대한 인수로, RELATEDTABLE
- 모든 시간 인텔리전스 함수에 대한 인수로

정규화되지 않은(*unqualified*) 열 이름은 대괄호로 묶은 열 이름(예: [Sales Amount])입니다. 예를 들어 현재 테이블의 같은 행에서 스칼라 값을 참조할 때는 정규화되지 않은 열 이름을 사용할 수 있습니다.

테이블 이름에 공백, 예약 키워드 또는 허용되지 않는 문자가 포함된 경우 테이블 이름을 작은따옴표로 묶어야 합니다. 또한 테이블 이름에 ANSI 영숫자 문자 범위를 벗어나는 문자가 포함된 경우에는 사용자의 로캘에서 해당 문자 집합을 지원하는지 여부와 관계없이 해당 이름을 따옴표로 묶어야 합니다. 예를 들어 키릴 문자(예: 'Таблица')로 작성된 테이블 이름을 포함하는 통합 문서를 여는 경우 공백을 포함하지 않더라도 테이블 이름을 따옴표로 묶어야 합니다.

NOTE

정규화된 열 이름을 쉽게 입력하려면 수식 편집기의 자동 완성 기능을 사용합니다.

테이블

- 열이 현재 테이블과 다른 테이블에 있을 때마다 테이블 이름이 필요합니다. 테이블 이름은 데이터베이스 내에서 고유해야 합니다.
- 공백, 기타 특수 문자 또는 영어가 아닌 영숫자 문자가 포함된 테이블 이름은 작은따옴표로 묶어야 합니다.

측정값

- 측정값 이름은 항상 대괄호 안에 있어야 합니다.
- 측정값 이름에는 공백이 포함될 수 있습니다.
- 각 측정값 이름은 모델 내에서 고유해야 합니다. 따라서 기존 측정값을 참조하는 경우 측정값 이름 앞에 테이블 이름은 선택 사항입니다. 단, 측정값을 만들 때는 측정값 정의가 저장되는 테이블을 항상 지정해야 합니다.

열

열 이름은 테이블의 컨텍스트에서 고유해야 합니다. 하지만 이름이 같은 열이 여러 테이블에 있을 수 있으며, 명확성은 테이블 이름을 통해 제공됩니다.

일반적으로 열이 속하는 기본 테이블을 참조하지 않고 열을 참조할 수 있습니다. 단, 분석할 때 이름이 충돌하거

나 열 이름을 정규화해야 하는 특정 함수인 경우는 예외입니다.

예약 키워드

테이블에 사용하는 이름이 Analysis Services 예약 키워드와 같으면 오류가 발생하며 테이블의 이름을 바꿔야 합니다. 하지만 개체 이름을 괄호(열의 경우)나 따옴표(테이블의 경우)로 묶으면 개체 이름에 키워드를 사용할 수 있습니다.

NOTE

따옴표는 애플리케이션에 따라 여러 가지 다른 문자로 나타낼 수 있습니다. 외부 문서나 웹 페이지에서 수식을 붙여 넣는 경우에는 여는 따옴표와 닫는 따옴표에 사용되는 문자의 ASCII 코드가 동일한지 확인해야 합니다. 그렇지 않으면 DAX에서 해당 기호를 따옴표로 인식하지 못해서 참조가 유효하지 않을 수 있습니다.

특수 문자

다음 문자 및 문자 형식은 테이블, 열 또는 측정값의 이름에 사용할 수 없습니다.

- 선행 또는 후행 공백, 공백이 이름 구분 기호, 대괄호 또는 단일 아포스트로피로 묶여 있지 않은 경우.
- 제어 문자
- 개체 이름에 사용할 수 없는 문자는 다음과 같습니다.

.,:~\/?&%\$!+=()[]{}<>

개체 이름 예

다음 표에는 일부 개체 이름의 예가 있습니다.

개체 유형	예	설명
테이블 이름	Sales	테이블 이름에 공백이나 기타 특수 문자가 포함되어 있지 않으면 해당 이름을 따옴표로 묶을 필요가 없습니다.
테이블 이름	'Canada Sales'	이름에 공백, 탭 또는 기타 특수 문자가 포함된 경우 이름을 작은따옴표로 묶습니다.
정규화된 열 이름	Sales[Amount]	테이블 이름이 열 이름 앞에 오고 열 이름은 대괄호로 묶습니다.
정규화된 측정값 이름	Sales[Profit]	테이블 이름이 측정값 이름 앞에 오고 측정값 이름은 대괄호로 묶습니다. 특정 컨텍스트에서는 정규화된 이름이 항상 필요합니다.
정규화되지 않은 열 이름	[Amount]	정규화되지 않은 이름은 열 이름만 대괄호 안에 있습니다. 정규화되지 않은 이름을 사용할 수 있는 컨텍스트는 동일한 테이블 내의 계산 열 또는 동일한 테이블을 검색하는 집계 함수의 수식입니다.
공백이 있는 테이블의 정규화된 열	'Canada Sales'[Qty]	테이블 이름에 공백이 포함되어 있으므로 작은따옴표로 묶어야 합니다.

기타 제한 사항

각 함수에 필요한 구문 및 함수가 수행할 수 있는 연산 유형은 함수에 따라 크게 달라집니다. 하지만 다음 규칙은 일반적으로 모든 수식과 식에 적용됩니다.

- DAX 수식 및 식은 테이블의 개별 값을 삽입하거나 수정할 수 없습니다.
- DAX를 사용하여 계산 행을 만들 수 없습니다. 계산 열과 측정값만 만들 수 있습니다.
- 계산 열을 정의할 때 함수를 모든 수준에 중첩할 수 있습니다.
- DAX에는 테이블을 반환하는 여러 함수가 있습니다. 일반적으로 이러한 함수가 반환하는 값은, 테이블을 입력으로 사용하는 다른 함수에 대한 입력으로 사용됩니다.

DAX 연산자 및 상수

다음 표에는 DAX에서 지원되는 연산자가 있습니다. 개별 연산자의 구문에 대한 자세한 내용은 [DAX 연산자](#)를 참조하세요.

연산자 유형	기호 및 사용
괄호 연산자	() 우선 순위 및 인수 그룹
산술 연산자	+ (더하기) - (빼기/ 부호) * (곱하기) / (나누기) ^ (지수)
비교 연산자	= (같음) > (보다 큼) < (보다 작음) >= (크거나 같음) <= (작거나 같음) <> (같지 않음)
텍스트 연결 연산자	& (연결)
논리 연산자	&& (및) (또는)

데이터 형식

DAX 수식에서 사용하는 열 또는 값의 데이터 형식을 캐스트, 변환 또는 지정할 필요가 없습니다. DAX 수식에서 데이터를 사용하면 DAX는 참조되는 열 및 사용자가 입력하는 값의 데이터 형식을 자동으로 식별하고, 지정된 연산을 완료하는 데 필요한 경우 암시적 변환을 수행합니다.

예를 들어 날짜 값에 숫자를 추가하려고 하면 엔진은 함수 컨텍스트에서 연산을 해석하고 숫자를 공통 데이터 형식으로 변환한 다음, 원하는 형식(날짜)으로 결과를 표시합니다.

단, 성공적으로 변환될 수 있는 값에는 몇 가지 제한이 있습니다. 값 또는 열에 현재 연산과 호환되지 않는 데이터 형식이 있으면 DAX는 오류를 반환합니다. 또한 DAX는 데이터 모델로 가져온 기존 데이터의 형식을 명시적

으로 변경, 변환 또는 캐스트할 수 있는 함수를 제공하지 않습니다.

IMPORTANT

DAX는 변형 데이터 형식을 사용하도록 지원하지 않습니다. 따라서 데이터 모델에 데이터를 로드하거나 가져올 때 각 열의 데이터는 일반적으로 일관된 데이터 형식이어야 합니다.

일부 함수는 문자열을 포함하는 스칼라 값을 반환하는 반면, 다른 함수는 숫자, 정수와 실수, 날짜와 시간을 사용합니다. 각 함수에 필요한 데이터 형식은 [DAX 함수](#) 섹션에 설명되어 있습니다.

여러 열과 여러 행의 데이터를 포함하는 테이블을 함수에 대한 인수로 사용할 수 있습니다. 또한 일부 함수는 메모리에 저장되어 다른 함수의 인수로 사용될 수 있는 테이블을 반환합니다.