

DAX(Data Analysis Expressions) 참조

DAX(데이터 분석 식)는 Excel 데이터 모델의 Power BI, Analysis Services 및 파워 피벗에서 수식과 식을 작성하기 위해 결합할 수 있는 함수 및 연산자의 라이브러리입니다.

함수



[DAX 함수](#)

[DAX 문](#)



[새로운 DAX 함수](#)

DAX에 대해 알아보기



[DAX 개요](#)



[동영상](#)



[Power BI Desktop에서 DAX 사용](#)

이해



[DAX 용어집](#)

[DAX 연산자](#)

[DAX 쿼리](#)

DAX 구문

DAX 개요

아티클 • 2023. 10. 20.

DAX(데이터 분석 식)는 Excel의 Analysis Services, Power BI 및 파워 피벗에서 사용되는 수식 식 언어입니다. DAX 수식에는 테이블 형식 데이터 모델의 관련 테이블 및 열에서 데이터에 대한 고급 계산 및 쿼리를 수행하는 함수, 연산자 및 값이 포함됩니다.

이 문서에서는 DAX에서 가장 중요한 개념에 대한 기본적인 소개만 제공합니다. DAX는 DAX를 사용하는 모든 제품에 적용되므로 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, 계산 테이블 및 행 수준 보안에 사용됩니다.

측정값 그룹

측정값은 컨텍스트에 따라 결과가 변경되는 동적 계산 수식입니다. 측정값은 Power BI 보고서 또는 Excel 피벗 테이블 또는 피벗 차트 같은 여러 특성을 사용하여 모델 데이터 결합 및 필터링을 지원하는 보고에 사용됩니다. 측정값은 모델 디자이너에서 DAX 수식 입력줄을 사용하여 생성됩니다.

측정값의 수식은 COUNT 또는 SUM과 같은 자동 합계 기능을 사용하여 자동으로 생성된 표준 집계 함수를 사용하거나 DAX 수식 입력줄을 사용하여 고유한 수식을 정의할 수 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 보기와 표시되지만 결과가 어느 곳에도 즉시 출력되지는 않습니다. 계산의 (필터링된) 결과를 즉시 볼 수 없는 이유는 컨텍스트 없이 측정값의 결과를 확인할 수 없기 때문입니다. 측정값을 평가하려면 각 셀과 관련된 데이터를 검색한 다음 각 셀에 대한 식을 평가하는 데 필요한 컨텍스트를 제공할 수 있는 보고 클라이언트 애플리케이션이 필요합니다. 해당 클라이언트는 Excel 피벗 테이블 또는 피벗 차트, Power BI 보고서 또는 SSMS(SQL Server Management Studio)의 DAX 쿼리에 있는 테이블 식일 수 있습니다.

클라이언트에 관계없이 결과의 각 셀에 대해 별도의 쿼리가 실행됩니다. 즉, 피벗 테이블의 행 머리글과 열 머리글의 각 조합 또는 Power BI 보고서에서 슬라이서와 필터를 선택할 때마다 측정값이 계산되는 데이터의 다른 하위 집합이 생성됩니다. 예를 들어 이 매우 간단한 측정값 수식을 사용합니다.

DAX

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

사용자가 보고서에 Total Sales 측정값을 배치한 다음 제품 테이블의 제품 범주 열을 필터에 배치하면 각 제품 범주에 대해 Sales Amount의 합계가 계산되고 표시됩니다.

계산 열과 달리 측정값의 구문에는 수식 앞에 측정값 이름이 포함됩니다. 방금 제공한 예제에서는 수식 앞에 Total Sales라는 이름이 나타납니다. 측정값을 만든 후에는 보고 클라이언트 애플리케이션 필드 목록에 이름과 해당 정의가 표시되며, 큐브 뷰 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 유효한 DAX 수식이 포함된 경우 수식을 입력하는 즉시 각 행에 대한 값이 계산됩니다. 그런 다음, 값이 메모리 내 데이터 모델에 저장됩니다. 예를 들어 Date 테이블에서 수식 입력줄에 수식을 입력할 때는 다음과 같습니다.

DAX

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

테이블의 각 행에 대한 값은 같은 날짜 테이블의 Calendar Year 열에서 값을 가져와서 공백과 대문자 Q를 추가한 다음 Calendar Quarter 열의 값(동일한 날짜 테이블)을 추가하여 계산됩니다. 계산 열의 각 행에 대한 결과는 즉시 계산되며 2017 Q1과 같이 표시됩니다. 열 값은 테이블 또는 관련 테이블이 처리(새로 고침)되거나 모델이 메모리에서 언로드된 후 Power BI Desktop 파일을 닫고 다시 열 때와 같이 다시 로드되는 경우에만 다시 계산됩니다.

자세한 내용은 다음을 참조하세요.

[Power BI Desktop의 계산 열](#)

[Analysis Services의 계산 열](#)

[파워 피벗의 계산 열입니다.](#)

계산된 테이블

계산된 테이블은 동일한 모델의 다른 테이블 전체 또는 일부에서 파생된 수식 식을 기반으로 계산된 개체입니다. DAX 수식은 데이터 원본에서 새 테이블의 열에 값을 쿼리하고 로드하는 대신 테이블 값을 정의합니다.

계산된 테이블은 역할 재생 차원에서 유용할 수 있습니다. 외래 키 관계에 따라 Date 테이블(OrderDate, ShipDate 또는 DueDate)이 그 예입니다. ShipDate에 대한 계산 테이블을 명시적으로 만들면 쿼리에 사용할 수 있는 독립 실행형 테이블이 다른 테이블과 마찬가지로 완전히 작동할 수 있습니다. 계산 테이블은 필터링된 행 집합 또는 다른 기준 테이블에서 열의 하위 집합 또는 상위 집합을 구성할 때도 유용합니다. 이렇게 하면 특정 시나리오를 지원하기 위해 해당 테이블의 변형을 만드는 동안 원래 테이블을 그대로 유지할 수 있습니다.

계산된 테이블은 다른 테이블과의 관계를 지원합니다. 계산 테이블의 열에는 데이터 형식, 서식이 있으며 데이터 범주에 속할 수 있습니다. 계산된 테이블의 이름은 다른 테이블과 마찬가지로 표시되거나 숨겨질 수 있습니다. 계산된 테이블은 데이터를 가져오는 테이블이 새로 고쳐지거나 업데이트되는 경우 다시 계산됩니다.

자세한 내용은 다음을 참조하세요.

[Power BI Desktop의 계산된 테이블](#)

[Analysis Services의 계산된 테이블입니다.](#)

행 수준 보안

행 수준 보안을 사용하면 DAX 수식이 부울 TRUE/FALSE 조건으로 평가되어야 하며, 특정 역할의 멤버가 쿼리 결과로 반환할 수 있는 행을 정의해야 합니다. 예를 들어 Sales 역할의 멤버의 경우 다음 DAX 수식이 포함된 Customers 테이블이 있습니다.

DAX

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

Sales 역할의 멤버는 미국 고객의 데이터만 볼 수 있으며 SUM과 같은 집계는 미국 내 고객에 대해서만 반환됩니다. Excel의 파워 피벗에서는 행 수준 보안을 사용할 수 없습니다.

DAX 수식을 사용하여 행 수준 security를 정의할 때 허용되는 행 집합을 만듭니다. 이렇게 하면 다른 행에 대한 액세스가 거부되지 않습니다. 오히려 허용된 행 집합의 일부로 반환되지 않습니다. 다른 역할은 DAX 수식에서 제외된 행에 대한 액세스를 허용할 수 있습니다. 사용자가 다른 역할의 멤버이고 해당 역할의 행 수준 보안을 통해 특정 행 집합에 액세스할 수 있는 경우 사용자는 해당 행에 대한 데이터를 볼 수 있습니다.

행 수준 보안 수식은 지정된 행과 관련 행에 적용됩니다. 테이블에 여러 관계가 있는 경우 필터는 활성 관계에 대한 보안을 적용합니다. 행 수준 보안 수식은 관련 테이블에 대해 정의된 다른 수식과 교차합니다.

자세한 내용은 다음을 참조하세요.
[Power BI를 사용하는 RLS\(행 수준 보안\)](#)
[Analysis Services의 역할](#)

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 종종 MDX(다차원 데이터 식) 쿼리보다 쓰기가 더 쉽고 효율적입니다.

DAX 쿼리는 T-SQL의 SELECT 문과 유사한 문입니다. DAX 쿼리의 가장 기본적인 형식은 evaluate 문입니다. 예를 들면 다음과 같습니다.

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

Results에서 EnglishProductName의 오름차순으로 SafetyStockLevel이 200 미만인 제품만 나열하는 테이블을 반환합니다.

쿼리의 일부로 측정값을 만들 수 있습니다. 측정값은 쿼리 기간 동안에만 존재합니다. 자세한 내용은 DAX 쿼리를 참조 [하세요](#).

제조법

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 창 또는 DAX 편집기의 위쪽에 있는 수식 입력줄을 사용합니다. 행 수준 보안에 대한 수식을 만들려면 역할 관리자 또는 역할 관리 대화 상자를 사용합니다. 이 섹션의 정보는 DAX 수식의 기본 사항을 이해하기 위한 것입니다.

수식 기본 사항

DAX 수식은 매우 간단하거나 매우 복잡할 수 있습니다. 다음 표에서는 계산 열에서 사용할 수 있는 간단한 수식의 몇 가지 예를 보여 줍니다.

수식	정의
= TODAY()	계산 열의 모든 행에 오늘 날짜를 삽입합니다.
= 3	계산 열의 모든 행에 값 3을 삽입합니다.
= [Column1] + [Column2]	[Column1]과 [Column2]의 같은 행에 값을 추가하고 결과를 같은 행의 계산 열에 넣습니다.

만드는 수식의 복잡도에 관계없이 다음 단계를 사용하여 수식을 작성할 수 있습니다.

- 각 수식은 등호(=)로 시작해야 합니다.
- 함수 이름을 입력하거나 선택하거나 식을 입력할 수 있습니다.
- 원하는 함수 또는 이름의 처음 몇 글자를 입력하기 시작하고 자동 완성은 사용 가능한 함수, 테이블 및 열 목록을 표시합니다. Tab 키를 눌러 자동 완성 목록의 항목을 수식에 추가합니다.
Fx 단추를 클릭하여 사용 가능한 함수의 목록을 표시할 수도 있습니다. 드롭다운 목록에서 함수를 선택하려면 화살표 키를 사용하여 항목을 강조 표시한 다음 **확인** 을 클릭하여 수식에 함수를 추가합니다.
- 가능한 테이블 및 열의 드롭다운 목록에서 인수를 선택하거나 값을 입력하여 함수에 인수를 제공합니다.
- 구문 오류를 확인합니다. 즉, 모든 괄호가 닫혀 있고 열, 테이블 및 값이 올바르게 참조되는지 확인합니다.
- Enter 키를 눌러 수식을 적용합니다.

① 참고

계산 열에서 수식을 입력하고 수식의 유효성을 검사하는 즉시 열이 값으로 채워집니다. 측정값에서 ENTER 키를 누르면 측정값 정의가 테이블에 저장됩니다. 수식이 잘 못되면 오류가 표시됩니다.

이 예제에서는 현재 분기의 Days라는 측정값에서 수식을 살펴보겠습니다.

DAX

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

이 측정값은 완료되지 않은 기간과 이전 기간 사이의 비교 비율을 만들기 위해 사용됩니다. 수식은 경과된 기간의 비율을 고려하여 이를 이전 기간의 동일한 부분과 비교해야 합니다. 이 경우 [일 현재 분기에서 날짜로]/[현재 분기의 일]은 현재 기간에 경과된 비율을 제공합니다.

이 수식에는 다음 요소가 포함됩니다.

Formula 요소	설명
Days in Current Quarter	측정값의 이름입니다.
=	등호(=)가 수식을 시작합니다.
COUNTROWS	COUNTROWS 는 날짜 테이블의 행 수를 계산합니다.
()	열기 및 닫는 괄호는 인수를 지정합니다.
DATESBETWEEN	DATESBETWEEN 함수는 Date 테이블의 Date 열에 있는 각 값의 마지막 날짜 사이의 날짜를 반환합니다.
'Date'	Date 테이블을 지정합니다. 테이블은 작은따옴표로 표시됩니다.
[Date]	Date 테이블의 Date 열을 지정합니다. 열은 대괄호 안에 있습니다.
,	
STARTOFQUARTER	STARTOFQUARTER 함수는 분기 시작 날짜를 반환합니다.
LASTDATE	LASTDATE 함수는 분기의 마지막 날짜를 반환합니다.
'Date'	Date 테이블을 지정합니다.
[Date]	Date 테이블의 Date 열을 지정합니다.
,	
ENDOFQUARTER	ENDOFQUARTER 함수
'Date'	Date 테이블을 지정합니다.
[Date]	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 함수를 사용하여 기존 수식 중간에 수식 자동 완성을 사용할 수 있습니다. 삽입 지점 바로 앞의 텍스트는 드롭다운 목록에 값을 표시하는 데 사용되며 삽입 지점 뒤의 모든 텍스트는 변경되지 기본.
- 자동 완성은 함수의 닫는 괄호를 추가하거나 자동으로 괄호와 일치하지 않습니다. 각 함수가 구문적으로 올바른지 또는 수식을 저장하거나 사용할 수 있는지 확인해야 합니다.

수식에서 여러 함수 사용

함수를 중첩할 수 있습니다. 즉, 한 함수의 결과를 다른 함수의 인수로 사용할 수 있습니다. 계산 열에 최대 64개 수준의 함수를 중첩할 수 있습니다. 그러나 중첩을 사용하면 수식을 만들거나 문제를 해결하기 어려울 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 함수는 모두 첫 번째 인수로 테이블이 필요합니다.

함수

함수는 식 내에서 명명된 수식입니다. 대부분의 함수에는 매개 변수라고도 하는 필수 인수와 선택적 인수가 입력으로 있습니다. 함수가 실행되면 값이 반환됩니다. DAX에는 날짜 및 시간을 사용하여 계산을 수행하고, 조건부 값을 만들고, 문자열로 작업하고, 관계에 따라 조회를 수행하고, 테이블을 반복하여 재귀 계산을 수행하는 데 사용할 수 있는 함수가 포함되어 있습니다. Excel 수식에 익숙한 경우 이러한 함수의 대부분은 매우 유사하게 표시됩니다. 그러나 DAX 수식은 다음과 같은 중요한 방법으로 다릅니다.

- DAX 함수는 항상 전체 열 또는 테이블을 참조합니다. 테이블 또는 열에서 특정 값만 사용하려는 경우 수식에 필터를 추가할 수 있습니다.
- 행 단위로 계산을 사용자 지정해야 하는 경우 DAX는 컨텍스트에 따라 달라지는 계산을 수행하기 위해 현재 행 값 또는 관련 값을 매개 변수의 일종으로 사용할 수 있는 함수를 제공합니다. 이러한 함수의 작동 방식을 이해하려면 이 문서의 컨텍스트를 참조하세요.
- DAX에는 값보다는 테이블을 반환하는 많은 함수가 포함되어 있습니다. 테이블은 보고 클라이언트에 표시되지 않지만 다른 함수에 대한 입력을 제공하는 데 사용됩니다. 예를 들어 테이블을 검색한 다음 테이블에 포함된 고유 값을 계산하거나 필터링된 테이블 또는 열 전체에서 동적 합계를 계산할 수 있습니다.
- DAX 함수에는 다양한 시간 인텔리전스 함수가 포함됩니다. 이러한 함수를 사용하면 날짜 범위를 정의하거나 선택하고 이러한 날짜 또는 범위에 따라 동적 계산을 수행할 수 있습니다. 예를 들어 병렬 기간 전체의 합계를 비교할 수 있습니다.

집계 함수

집계 함수는 식에 정의된 대로 열 또는 테이블의 모든 행에 대한 개수, 합계, 평균, 최소값 또는 최대값과 같은 (스칼라) 값을 계산합니다. 자세한 내용은 집계 함수를 참조 [하세요](#).

날짜 및 시간 함수

DAX의 날짜 및 시간 함수는 Microsoft Excel의 날짜 및 시간 함수와 비슷합니다. 그러나 DAX 함수는 1900년 3월 1일부터 날짜/시간 데이터 형식을 기반으로 합니다. 자세한 내용은 날짜 및 시간 함수를 참조 [하세요](#).

Filter 함수

DAX의 필터 함수는 특정 데이터 형식을 반환하고, 관련 이야기에서 값을 조회하고, 관련 값을 필터링합니다. 조회 함수는 데이터베이스와 같은 테이블 및 관계를 사용하여 작동합니다. 필터링 함수를 사용하면 데이터 컨텍스트를 조작하여 동적 계산을 만들 수 있습니다. 자세한 내용은 필터 함수를 참조 [하세요](#).

재무 기능

DAX의 재무 기능은 순 현재 가치 및 수익률과 같은 재무 계산을 수행하는 수식에 사용됩니다. 이러한 함수는 Microsoft Excel에서 사용되는 재무 함수와 유사합니다. 자세한 내용은 재무 기능을 참조 [하세요](#).

정보 함수

정보 함수는 인수로 제공되는 셀 또는 행을 살펴보고 값이 예상 형식과 일치하는지 여부를 알려줍니다. 예를 들어 ISERROR 함수는 참조하는 값에 오류가 포함되어 있는 경우 TRUE를 반환합니다. 자세한 내용은 정보 함수를 참조 [하세요](#).

논리 함수

논리 함수는 식에 따라 동작하여 식의 값에 대한 정보를 반환합니다. 예를 들어 TRUE 함수를 사용하면 평가 중인 식이 TRUE 값을 반환하는지 여부를 알 수 있습니다. 자세한 내용은 논리 함수를 참조 [하세요](#).

수치 연산 및 삼각 함수

DAX의 수치 연산 함수는 Excel 수치 연산 및 삼각 함수와 매우 비슷합니다. DAX 함수에서 사용하는 숫자 데이터 형식에는 몇 가지 사소한 차이점이 있습니다. 자세한 내용은 수학

및 삼각 함수를 참조 [하세요](#).

기타 함수

이러한 함수는 대부분의 다른 함수가 속한 범주에서 정의할 수 없는 고유한 작업을 수행합니다. 자세한 내용은 다른 함수를 참조 [하세요](#).

관계 함수

DAX의 관계 함수를 사용하면 다른 관련 테이블의 값을 반환하고 식에 사용할 특정 관계를 지정하며 교차 필터링 방향을 지정할 수 있습니다. 자세한 내용은 관계 함수를 참조 [하세요](#).

통계 함수

통계 함수는 표준 편차 및 순열 수와 같은 통계 분포 및 확률과 관련된 값을 계산합니다. 자세한 내용은 통계 함수를 참조 [하세요](#).

텍스트 함수

DAX의 텍스트 함수는 Excel의 텍스트 함수와 매우 유사합니다. 문자열의 일부를 반환하거나, 문자열 내에서 텍스트를 검색하거나, 문자열 값을 연결할 수 있습니다. DAX는 날짜, 시간 및 숫자의 형식을 제어하는 함수도 제공합니다. 자세한 내용은 Text 함수를 참조 [하세요](#).

시간 인텔리전스 함수

DAX에서 제공되는 시간 인텔리전스 함수를 사용하면 일정 및 날짜에 대한 기본 제공 지식을 사용하는 계산을 만들 수 있습니다. 시간 및 날짜 범위를 집계 또는 계산과 함께 사용하면 비교 가능한 기간에서 판매량, 재고 등에 대한 의미 있는 비교 자료를 작성할 수 있습니다. 자세한 내용은 DAX(시간 인텔리전스 함수)를 참조하세요.

테이블 조작 함수

이러한 함수는 테이블을 반환하거나 기존 테이블을 조작합니다. 예를 들어 ADDCOLUMNS를 사용하여 지정된 테이블에 계산 열을 추가하거나 SUMMARIZECOLUMNS 함수를 사용하여 그룹 집합에 대해 요약 테이블을 반환할 수 있습니다. 자세한 내용은 테이블 조작 함수를 참조 [하세요](#).

변수

VAR을 사용하여 식 내에서 변수를 만들 수 있습니다. VAR은 기술적으로 함수가 아니며 식의 결과를 명명된 변수로 저장하는 키워드(keyword). 그런 다음 해당 변수를 다른 측정 값 식에 인수로 전달할 수 있습니다. 예시:

```
DAX  
  
VAR  
    TotalQty = SUM ( Sales[Quantity] )  
  
Return  
  
IF (  
    TotalQty > 1000,  
    TotalQty * 0.95,  
    TotalQty * 1.25  
)
```

이 예제에서는 TotalQty를 명명된 변수로 다른 식에 전달할 수 있습니다. 변수는 테이블을 포함한 모든 스칼라 데이터 형식일 수 있습니다. DAX 수식에서 변수를 사용하는 것은 매우 강력할 수 있습니다.

데이터 유형

다양한 데이터 형식을 지원할 수 있는 다양한 데이터 원본에서 모델로 데이터를 가져올 수 있습니다. 데이터를 모델로 가져올 때 데이터는 테이블 형식 모델 데이터 형식 중 하나로 변환됩니다. 모델 데이터가 계산에 사용되는 경우 데이터는 계산 기간 및 출력에 대한 DAX 데이터 형식으로 변환됩니다. DAX 수식을 만들 때 수식에 사용된 용어에 따라 반환된 값 데이터 형식이 자동으로 결정됩니다.

DAX는 다음 데이터 형식을 지원합니다.

모델의 데이터 형식	DAX의 데이터 형식	설명
정수	64비트(8바이트) 정수 값 ^{1, 2}	소수 자릿수가 없는 숫자입니다. 정수는 양수 또는 음수일 수 있지만 -9,223,372,036,854,775,808(-2^63)과 9,223,372,036,854,775,807(2^63-1) 사이의 정수여야 합니다.
10진수	64비트(8바이트) 실수 ^{1, 2}	실수는 소수 자릿수를 가질 수 있는 숫자입니다. 실수는 다양한 값을 다룹니다. - 1.79E +308에서 -2.23E -308 사이의 음수 값
		0(Zero)

모델의 데이터 형식	DAX의 데이터 형식	설명
		2.23E -308에서 1.79E + 308까지의 양수 값 하지만 유효 숫자 수는 열일곱 자릿수로 제한됩니다.
Boolean	Boolean	True 또는 False 값입니다.
Text	문자열	유니코드 문자 데이터 문자열입니다. 텍스트 형식으로 표현되는 문자열, 숫자 또는 날짜일 수 있습니다.
날짜	날짜/시간	허용되는 날짜-시간 표현의 날짜 및 시간입니다. 유효한 날짜는 1900년 3월 1일 이후의 모든 날짜입니다.
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	Blank	공백은 SQL null을 나타내고 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 사용하고 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. 예를 들어 수식의 오류 또는 예기치 않은 결과는 인수에 지정된 데이터 형식과 함께 사용할 수 없는 특정 연산자를 사용하여 발생하는 경우가 많습니다. 예를 들어 수식 `= 1 & 2`은 12의 문자열 결과를 반환합니다. 하지만 수식 `= "1" + "2"`는 정수 3을 반환합니다.

컨텍스트

컨텍스트는 DAX 수식을 만들 때 이해해야 하는 중요한 개념입니다. 컨텍스트를 사용하면 수식의 결과가 현재 행 또는 셀 선택 영역과 관련된 데이터를 반영하도록 변경되므로 동적 분석을 수행할 수 있습니다. 컨텍스트를 이해하고 컨텍스트를 효과적으로 사용하는 것은 고성능 동적 분석을 빌드하고 수식의 문제를 해결하는 데 중요합니다.

테이블 형식 모델의 수식은 다른 디자인 요소에 따라 다른 컨텍스트에서 평가할 수 있습니다.

- 피벗 테이블 또는 보고서에 적용된 필터
- 수식 내에 정의된 필터
- 수식 내에서 특수 함수를 사용하여 지정한 관계

컨텍스트의 유형으로는 행 컨텍스트, 쿼리 컨텍스트 및 필터 컨텍스트가 있습니다.

행 컨텍스트

행 컨텍스트는 "현재 행"으로 간주할 수 있습니다. 계산 열에서 수식을 만드는 경우 해당 수식의 행 컨텍스트에 현재 행에 있는 모든 열의 값이 포함됩니다. 테이블이 다른 테이블과 관련된 경우 콘텐츠에는 현재 행과 관련된 다른 테이블의 모든 값도 포함됩니다.

예를 들어 동일한 테이블의 두 열 = [Freight] + [Tax] 인 Freight 및 Tax의 값을 함께 추가하는 계산 열을 만든다고 가정해 보겠습니다. 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 다음 수식은 RELATED 함수를 사용하여 주문이 배송된 지역에 따라 관련 테이블에서 세금 값을 가져옵니다. 세금 값은 현재 테이블의 지역 값을 사용하여 관련 테이블의 지역을 조회한 다음 관련 테이블에서 해당 지역의 세율을 가져오는 방식으로 결정됩니다.

DAX

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

이 수식은 Region 테이블에서 현재 지역의 세율을 가져와서 Freight 열의 값에 더합니다. DAX 수식에서 테이블을 연결하는 특정 관계를 알거나 지정할 필요는 없습니다.

여러 행 컨텍스트

DAX에는 테이블에 대한 계산을 반복하는 함수가 포함되어 있습니다. 이러한 함수에는 각각 고유한 행 컨텍스트가 있는 여러 개의 현재 행이 있을 수 있습니다. 기본적으로 이러한 함수를 사용하면 내부 및 외부 루프를 통해 재귀적으로 작업을 수행하는 수식을 만들 수 있습니다.

예를 들어 모델에 Products 테이블과 Sales 테이블이 포함되어 있다고 가정합니다. 사용자는 여러 제품과 관련된 트랜잭션으로 가득 찬 전체 판매 테이블을 살펴보고 한 트랜잭션에서 각 제품에 대해 주문된 최대 수량을 찾을 수 있습니다.

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

DAX

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

이 수식의 자세한 예제는 [EARLIER을 참조하세요](#).

요약하자면, EARLIER 함수는 현재 작업 이전의 작업에서 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 복잡한 집계를 만들 수 있도록 두 루프 간에 값을 자동으로 공급합니다.

쿼리 컨텍스트

쿼리 컨텍스트는 수식에 대해 암시적으로 검색되는 데이터의 하위 집합을 참조합니다. 예를 들어 사용자가 측정값 또는 필드를 보고서에 배치할 때 엔진은 행 및 열 머리글, 슬라이서 및 보고서 필터를 검사하여 컨텍스트를 확인합니다. 그런 다음, 모델 데이터에 대해 필요한 쿼리를 실행하여 데이터의 올바른 하위 집합을 얻고, 수식으로 정의된 계산을 수행한 다음, 보고서의 값을 채웁니다.

수식을 배치하는 위치에 따라 컨텍스트가 변경되므로 수식의 결과도 변경됩니다. 예를 들어 Sales 테이블 = `SUM('Sales'[Profit])`의 Profit 열에 있는 값의 합계를 계산하는 수식을 만듭니다. Sales 테이블 내의 계산 열에서 이 수식을 사용하는 경우 수식의 쿼리 컨텍스트는 항상 Sales 테이블의 전체 데이터 집합이므로 수식의 결과는 전체 테이블에 대해 동일합니다. 결과는 모든 지역, 모든 제품, 모든 연도 등에 대한 이익을 갖게 됩니다.

그러나 사용자는 일반적으로 동일한 결과를 수백 번 보고 싶지 않고 특정 연도, 특정 국가, 특정 제품 또는 이러한 조합에 대한 수익을 얻고 총합계를 구하려고 합니다.

보고서에서는 필드를 필터링, 추가 또는 제거하고 슬라이서를 사용하여 컨텍스트가 변경됩니다. 각 변경에 대해 측정값이 평가되는 쿼리 컨텍스트입니다. 따라서 측정값에 사용되는 동일한 수식은 각 셀에 대해 다른 쿼리 컨텍스트에서 평가됩니다.

필터 컨텍스트

필터 컨텍스트는 각 열 또는 관련 테이블에서 검색된 값에 허용되는 값 집합입니다. 필터는 디자이너의 열이나 프레젠테이션 계층(보고서 및 피벗 테이블)에 적용할 수 있습니다. 수식 내의 필터 식을 통해 필터를 명시적으로 정의할 수도 있습니다.

필터 컨텍스트는 수식에 인수를 사용하여 열이나 테이블에 적용되는 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 여러 가지 방법이 있습니다. Power BI 보고서와 같이 모델을 사용할 수 있는 클라이언트의 컨텍스트 내에서 사용자는 행 및 열 머리글에 슬라이서 또는 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 열에 대한 필터를 완전히 지우거나 선택적으로 지울 수도 있습니다. 이는 총합계를 계산하는 수식을 만들 때 매우 유용합니다.

수식 내에서 필터를 만드는 방법에 대한 자세한 내용은 [FILTER 함수\(DAX\)를 참조하세요](#). 총합계를 만들기 위해 필터를 지울 수 있는 방법의 예는 [ALL 함수\(DAX\)를 참조하세요](#).

수식 내에서 필터를 선택적으로 지우고 적용하는 방법의 예는 [ALLEXCEPT를 참조하세요](#).

수식에서 컨텍스트 확인

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

유효성 검사(및 다시 계산 작업) 중 컨텍스트는 모델의 사용 가능한 테이블, 테이블 간의 관계 및 적용된 필터를 사용하여 이전 섹션에 설명된 대로 결정됩니다.

예를 들어 다른 테이블과 관련되지 않은 일부 데이터를 새 테이블로 가져오고 필터를 적용하지 않은 경우 현재 컨텍스트는 테이블의 전체 열 집합입니다. 테이블이 다른 테이블과의 관계로 연결된 경우 현재 컨텍스트에는 관련 테이블이 포함됩니다. 테이블의 열을 슬라이서가 있고 일부 보고서 필터가 있는 보고서에 추가하는 경우 수식의 컨텍스트는 보고서의 각 셀에 있는 데이터의 하위 집합입니다.

컨텍스트는 수식 문제를 해결하기 어렵게 만들 수 있는 강력한 개념입니다. 컨텍스트의 작동 방식을 확인하려면 간단한 수식 및 관계로 시작하는 것이 좋습니다. 다음 섹션에서는 수식이 다양한 유형의 컨텍스트를 사용하여 결과를 동적으로 반환하는 방법에 대한 몇 가지 예를 제공합니다.

연산자

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

- 값을 비교하고 논리적 TRUE\FALSE 값을 반환하는 비교 연산자입니다.
- 숫자 값을 반환하는 산술 계산을 수행하는 산술 연산자입니다.

- 둘 이상의 텍스트 문자열을 결합하는 텍스트 연결 연산자
- 둘 이상의 식을 결합하여 단일 결과를 반환하는 논리 연산자입니다.

DAX 수식에 사용되는 연산자에 대한 자세한 내용은 DAX 연산자를 참조하세요.

테이블 및 열 작업

테이블 형식 데이터 모델의 테이블은 Excel 테이블과 비슷하지만 데이터와 수식으로 작업하는 방식은 다릅니다.

- 수식은 개별 셀, 범위 참조 또는 배열이 아닌 테이블 및 열에서만 작동합니다.
- 수식은 관계를 사용하여 관련 테이블에서 값을 가져올 수 있습니다. 검색되는 값은 항상 현재 행 값과 관련된 값입니다.
- Excel 워크시트에서처럼 불규칙하거나 "비정형" 데이터를 사용할 수 없습니다. 테이블의 각 행에는 동일한 수의 열이 포함되어야 합니다. 그러나 일부 열에는 빈 값이 있을 수 있습니다. Excel 데이터 테이블 및 테이블 형식 모델 데이터 테이블은 서로 교환할 수 없습니다.
- 각 열에 대해 데이터 형식이 설정되므로 해당 열의 각 값은 동일한 형식이어야 합니다.

수식에서 테이블 및 열 참조

이름을 사용하여 모든 테이블과 열을 참조할 수 있습니다. 예를 들어 다음 수식은 정규화된 이름을 사용하여 두 테이블의 열을 참조하는 방법을 보여 줍니다.

DAX

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

수식을 평가할 때 모델 디자이너는 먼저 일반 구문을 검사 다음 현재 컨텍스트에서 가능한 열 및 테이블에 대해 제공하는 열 및 테이블의 이름을 검사. 이름이 모호하거나 열 또는 테이블을 찾을 수 없는 경우 수식에서 오류가 발생하여 오류가 발생한 셀에 데이터 값 대신 #ERROR 문자열이 표시됩니다. 테이블, 열 및 기타 개체의 명명 요구 사항에 대한 자세한 내용은 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 확인한 다음 각각에 대한 배송 비용을 합산할 수 있습니다. 그러나 대부분의 경우 관계가 필요하지 않을 수 있

습니다. 수식에서 [LOOKUPVALUE](#) 함수를 사용하여 `search_column` 지정한 조건과 `search_value` 인수를 충족하는 행의 `resultColumnName` 값을 반환할 수 있습니다.

많은 DAX 함수는 참조한 열을 찾고 적합한 결과를 반환하기 위해 테이블 간 또는 여러 테이블 간에 관계가 있어야 합니다. 다른 함수는 관계를 식별하려고 시도합니다. 그러나 최상의 결과를 얻으려면 항상 가능한 경우 관계를 만들어야 합니다. 테이블 형식 데이터 모델은 테이블 간의 여러 관계를 지원합니다. 혼동 또는 잘못된 결과를 방지하기 위해 한 번에 하나의 관계만 활성 관계로 지정되지만 필요에 따라 활성 관계를 변경하여 계산에서 데이터의 서로 다른 연결을 트래버스할 수 있습니다. [USERELATIONSHIP](#) 함수를 사용하여 특정 계산에 사용할 하나 이상의 관계를 지정할 수 있습니다.

관계를 사용할 때는 다음 수식 디자인 규칙을 준수해야 합니다.

- 테이블이 관계로 연결된 경우 키로 사용되는 두 열에 일치하는 값이 있는지 확인해야 합니다. 참조 무결성은 적용되지 않으므로 키 열에 일치하지 않는 값을 갖고 관계를 만들 수 있습니다. 이 경우 빈 값 또는 일치하지 않는 값이 수식의 결과에 영향을 줄 수 있다는 점에 유의해야 합니다.
- 관계를 사용하여 모델의 테이블을 연결할 때 수식이 평가되는 범위 또는 컨텍스트를 확대합니다. 새 테이블 추가, 새 관계 또는 활성 관계의 변경으로 인한 컨텍스트 변경으로 인해 예상할 수 없는 방식으로 결과가 변경될 수 있습니다. 자세한 내용은 이 문서의 컨텍스트 [를 참조하세요](#).

프로세스 및 새로 고침

프로세스 및 다시 계산은 별도의 두 가지 작업이지만 관련된 작업입니다. 복잡한 수식, 대용량 데이터 또는 외부 데이터 원본에서 가져온 데이터를 포함하는 데이터 모델을 디자인할 때 이러한 개념을 완벽하게 이해해야 합니다.

프로세스(새로 고침)가 외부 데이터 원본의 새 데이터로 모델의 데이터를 업데이트하고 있습니다.

다시 계산은 수식 자체의 변경 내용을 반영하고 기본 데이터의 변경 내용을 반영하도록 수식의 결과를 업데이트하는 프로세스입니다. 다시 계산은 다음과 같은 방식으로 성능에 영향을 줄 수 있습니다.

- 계산 열의 값은 계산되고 모델에 저장됩니다. 계산 열의 값을 업데이트하려면 프로세스 전체, 프로세스 데이터 또는 프로세스 다시 계산의 세 가지 처리 명령 중 하나를 사용하여 모델을 처리해야 합니다. 수식을 변경할 때마다 항상 전체 열에 대해 수식의 결과를 다시 계산해야 합니다.
- 측정값으로 계산된 값은 사용자가 피벗 테이블에 측정값을 추가하거나 보고서를 열 때마다 동적으로 평가됩니다. 사용자가 컨텍스트를 수정하면 측정값에서 반환된 값

이 변경됩니다. 측정값의 결과는 항상 메모리 내 캐시의 최신 내용을 반영합니다.

다시 계산 결과가 다른 값을 반환하지 않는 한 처리 및 다시 계산은 행 수준 보안 수식에 영향을 주지 않으므로 역할 멤버가 행을 쿼리할 수 있거나 쿼리할 수 없습니다.

업데이트

DAX는 지속적으로 개선되고 있습니다. 새 함수와 업데이트된 함수는 일반적으로 매월 제 공되는 다음 업데이트와 함께 릴리스됩니다. 서비스가 먼저 업데이트되고 Power BI Desktop, Excel, SSMS(SQL Server Management Studio) 및 SSDT(Visual Studio용 Analysis Services 프로젝트 확장)와 같은 설치된 애플리케이션이 뒤따릅니다. SQL Server Analysis Services는 다음 누적 업데이트로 업데이트됩니다. 새 함수는 Power BI Desktop 업데이트 와 일치하는 DAX 함수 참조에서 먼저 발표되고 설명됩니다.

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

문제 해결

수식을 정의할 때 오류가 발생하면 수식에 구문 오류, 의미 체계 오류 또는 계산 오류가 포함될 수 있습니다.

구문 오류는 가장 쉽게 해결할 수 있습니다. 그들은 일반적으로 누락 된 괄호 또는 쉼표를 포함한다.

구문이 올바르지만 참조되는 값이나 열이 수식의 컨텍스트에서 의미가 없는 경우 다른 유형의 오류가 발생합니다. 이러한 의미 체계 및 계산 오류는 다음과 같은 문제로 인해 발생할 수 있습니다.

- 수식은 존재하지 않는 열, 테이블 또는 함수를 나타냅니다.
- 수식이 올바른 것처럼 보이지만 데이터 엔진이 데이터를 가져오면 형식이 일치하지 않는 것을 발견하고 오류를 발생합니다.
- 수식은 잘못된 수 또는 형식의 인수를 함수에 전달합니다.
- 수식은 오류가 있는 다른 열을 참조하므로 해당 값이 잘못되었습니다.
- 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[↗] 은 무료 데이터 모델링 및 보고 애플리케이션입니다. 모델 디자이너에는 DAX 계산 수식을 만들기 위한 DAX 편집기가 포함되어 있습니다.

Excel의 파워 피벗



Excel 모델 디자이너의[↗] 파워 피벗에는 DAX 계산 수식을 만들기 위한 DAX 편집기가 포함되어 있습니다.

Visual Studio



VSIX(Analysis Services 프로젝트[↗] 확장)가 있는 Visual Studio는 Analysis Services 모델 프로젝트를 만드는 데 사용됩니다. 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 위한 필수 도구입니다. SSMS에는 테이블 형식 모델과 다차원 모델을 모두 쿼리하기 위한 DAX 쿼리 편집기가 포함되어 있습니다.

DAX Studio



DAX Studio [☞](#) 는 Excel 모델에서 Analysis Services, Power BI Desktop 및 Power Pivot에 대해 DAX 쿼리를 만들고 실행하기 위한 오픈 소스 클라이언트 도구입니다.

테이블 형식 편집기



테이블 형식 편집 [☞](#) 기는 테이블 형식 모델 메타데이터의 모든 개체에 대한 직관적이고 계층적인 보기로 제공하는 오픈 소스 도구입니다. 테이블 형식 편집기에는 구문 강조 표시가 있는 DAX 편집기가 포함되어 있어 측정값, 계산 열 및 계산된 테이블 식을 쉽게 편집할 수 있습니다.

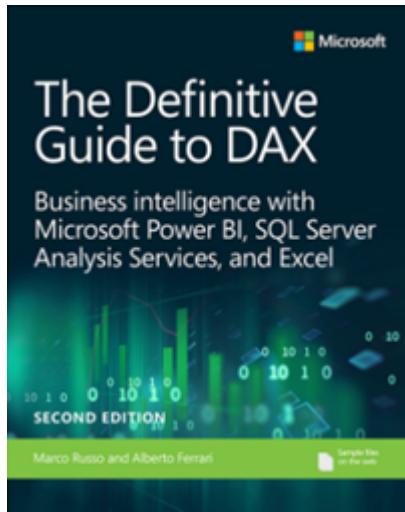
학습 리소스

DAX를 학습할 때 사용할 애플리케이션을 사용하여 데이터 모델을 만드는 것이 가장 좋습니다. Excel의 Analysis Services, Power BI Desktop 및 Power Pivot에는 모두 DAX를 사용하여 측정값, 계산 열 및 행 필터를 만드는 방법에 대한 단원을 포함하는 문서와 자습서가 있습니다. 다음은 몇 가지 추가 리소스입니다.

비디오

[Power BI Desktop에서 DAX 사용 학습 경로.](#)

[알베르토 페라리와 마르코 루소에 의해 DAX \[☞\]\(#\)에 대한 최종 가이드](#) (마이크로 소프트 프레스). 이제 두 번째 버전에서 이 광범위한 가이드는 데이터 모델러 및 BI 전문가를 위한 혁신적인 고성능 기술에 대한 기본 사항을 제공합니다.



커뮤니티

DAX는 항상 전문 지식을 공유하고자 하는 활기찬 커뮤니티를 보유하고 있습니다.

Microsoft Power BI 커뮤니티 [↗](#)에는 DAX, [DAX 명령 및 팁 ↗](#)을 위한 특별 토론 포럼이 있습니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

동영상

아티클 • 2023. 10. 20.

Power BI Desktop, Excel의 파워 피벗 또는 Analysis Services를 사용하는 경우 DAX(데이터 분석 식) 학습은 효과적인 데이터 모델을 만드는 데 필수적입니다. 이 강력한 식 언어 사용을 시작하는 데 도움이 되는 몇 가지 비디오는 다음과 같습니다.

DAX 101

이 DAX 101 비디오에서 Microsoft 파트너 알베르토 페라리는 DAX의 필수 개념을 소개합니다. 실용적이고 명확한 예제를 통해 DAX를 사용한 측정값, 계산 열 및 기본 데이터 모델링 식에 대해 알아봅니다.

<https://www.youtube-nocookie.com/embed/kIQAZLr5vxA> ↗

고급 DAX

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

<https://www.youtube-nocookie.com/embed/6ncHnWMEdic> ↗

Power BI에 대한 실제 DAX

이 비디오에서는 Microsoft 수석 프로그램 관리자인 Phil Seamark가 Power BI에서 DAX를 실제로 사용하기 위한 몇 가지 모범 사례를 안내합니다.

<https://www.youtube-nocookie.com/embed/1fGfqzS37qs> ↗

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX 샘플 모델

아티클 • 2023. 10. 20.

Adventure Works DW 2020 Power BI Desktop 샘플 모델은 DAX 학습을 지원하도록 설계되었습니다. 이 모델은 AdventureWorksDW2017용 Adventure Works 데이터 웨어하우스 샘플을 [기반으로](#) 하지만 샘플 모델의 목표에 맞게 데이터가 수정되었습니다.

샘플 모델에는 DAX 수식이 포함되어 있지 않습니다. 그러나 수백 또는 수천 개의 잠재적인 계산 수식 및 쿼리를 지원합니다. CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF 및 LOOKUPVALUE와 같은 일부 함수 예제는 수정 없이 샘플 모델에 추가할 수 있습니다. 샘플 모델에서 작동하는 다른 함수 참조 문서에 더 많은 예제를 포함하기 위해 노력하고 있습니다.

시나리오



Adventure Works 회사는 자전거와 액세서리를 글로벌 시장에 판매하는 자전거 제조업체를 대표합니다. 회사에는 Azure SQL Database에 저장된 데이터 웨어하우스 데이터가 있습니다.

모델 구조

모델에는 다음 7개의 테이블이 있습니다.

테이블	설명
고객	고객과 해당 지리적 위치를 설명합니다. 고객은 온라인으로 제품을 구매합니다(인터넷 판매).
날짜	주문 날짜, 배송 날짜 및 기한에 대한 Date 테이블과 Sales 테이블 사이에는 세 가지 관계가 있습니다. 주문 날짜 관계가 활성화되어 있습니다. 이 회사는 매년 7월 1일에 시작되는 회계 연도를 사용하여 매출을 보고합니다. 테이블은 날짜 열을 사용하여 날짜 테이블로 표시됩니다.
Product	완제품만 저장합니다.

테이블	설명
리셀러	재판매인 및 해당 자리적 위치를 설명합니다. 고객에게 제품을 판매하는 재판매인.
Sales	행을 판매 주문 라인 조직으로 저장합니다. 모든 재무 가치는 미국 달러(USD)입니다. 가장 빠른 주문 날짜는 2017년 7월 1일이며, 최신 주문 날짜는 2020년 6월 15일입니다.
판매 주문	판매 주문 및 주문 줄 번호와 재판매인 또는 인터넷인 판매 채널에 대해 설명합니다. 이 테이블은 Sales 테이블과 일대일 관계를 맺습니다.
판매 지역	판매 지역은 그룹(북아메리카, 유럽 및 태평양), 국가 및 지역으로 구성됩니다. 미국 지역 수준에서 제품을 판매합니다.

샘플 다운로드

여기에서 Power BI Desktop 샘플 모델 파일을 [다운로드합니다](#).

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다](#).
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ORDERBY, PARTITIONBY 및 MATCHBY 함수 이해

아티클 • 2023. 08. 27.

DAX의 ORDERBY, PARTITIONBY 및 MATCHBY 함수는 DAX 창 함수(INDEX, OFFSET, WINDOW, RANK, ROWNUMBER)와 함께 사용할 수 있는 특수 함수입니다.

WINDOW 함수를 성공적으로 사용하는 데는 ORDERBY, PARTITIONBY 및 MATCHBY를 이해하는 것이 중요합니다. 여기에 제공된 예제에서는 OFFSET을 사용하지만 다른 Window 함수에도 유사하게 적용할 수 있습니다.

시나리오

Window 함수를 전혀 사용하지 않는 예제부터 시작해 보겠습니다. 아래는 연간 총 판매액(색상별)을 반환하는 표입니다. 이 테이블을 정의하는 방법에는 여러 가지가 있지만 DAX에서 발생하는 작업을 이해하는 데 관심이 있으므로 계산된 테이블을 사용합니다. 테이블식은 다음과 같습니다.

DAX

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

이 계산 테이블 식은 SUMMARIZECOLUMNS를 사용하여 FactInternetSales 테이블에서 SalesAmount 열의 SUM, DimProduct 테이블의 Color 열 및 DimDate 테이블의 CalendarYear 열을 계산하는 것을 볼 수 있습니다. 결과는 다음과 같습니다.

색상	CalendarYear	CurrentYearSales
"Black"	2017	393885
"Black"	2018	1818835
"Black"	2019	3981638
"Black"	2020	2644054

색상	CalendarYear	CurrentYearSales
"파랑"	2019	994448
"파랑"	2020	1284648
"Multi"	2019	48622
"Multi"	2020	57849
"NA"를 입력하세요.	2019	207822
"NA"를 입력하세요.	2020	227295
"빨강"	2017	2961198
"빨강"	2018	3686935
"빨강"	2019	900175
"빨강"	2020	176022
"실버"	2017	326399
"실버"	2018	750026
"실버"	2019	2165176
"실버"	2020	1871788
"흰색"	2019	2517
"흰색"	2020	2589
"노란색"	2018	163071
"노란색"	2019	2072083
"노란색"	2020	2621602

이제 각 색상에 대해 전년 대비 매출 차이를 계산하는 비즈니스 문제를 해결하려고 한다고 상상해 보겠습니다. 효과적으로, 우리는 전년도에 동일한 색상에 대한 판매를 찾아 컨텍스트에서, 현재 연도의 판매에서 빼는 방법이 필요합니다. 예를 들어 [Red, 2019] 조합의 경우 [Red, 2018]의 매출을 찾고 있습니다. 이 작업이 완료되면 현재 판매에서 빼고 필요한 값을 반환할 수 있습니다.

OFFSET 사용

OFFSET은 위에서 설명한 비즈니스 질문에 답변하는 데 필요한 이전 유형의 계산과 일반적인 비교에 적합합니다. 상대 이동을 수행할 수 있기 때문에 적합합니다. 첫 번째 시도는

다음과 같습니다.

DAX

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

이 식에는 많은 일이 일어나고 있습니다. ADDCOLUMNS를 사용하여 **PreviousColorSales**라는 열을 사용하여 이전부터 테이블을 확장했습니다. 해당 열의 내용은 이전 색(OFFSET을 사용하여 검색됨)에 대해 SUM(FactInternetSales[SalesAmount])인 CurrentYearSales로 설정됩니다.

결과는 다음과 같습니다.

색상	CalendarYear	CurrentYearSales	PreviousColorSales
"Black"	2017	393885	
"Black"	2018	1818835	393885
"Black"	2019	3981638	1818835
"Black"	2020	2644054	3981638
"파랑"	2019	994448	2644054
"파랑"	2020	1284648	994448
"Multi"	2019	48622	1284648
"Multi"	2020	57849	48622
"NA"를 입력하세요.	2019	207822	57849

색상	CalendarYear	CurrentYearSales	PreviousColorSales
"NA"를 입력하세요.	2020	227295	207822
"빨강"	2017	2961198	227295
"빨강"	2018	3686935	2961198
"빨강"	2019	900175	3686935
"빨강"	2020	176022	900175
"실버"	2017	326399	176022
"실버"	2018	750026	326399
"실버"	2019	2165176	750026
"실버"	2020	1871788	2165176
"흰색"	2019	2517	1871788
"흰색"	2020	2589	2517
"노란색"	2018	163071	2589
"노란색"	2019	2072083	163071
"노란색"	2020	2621602	2072083

이것은 우리의 목표에 한 걸음 더 가까이, 하지만 우리가 자세히 보면 그것은 우리가 후 정확히 일치하지 않습니다. 예를 들어 [Silver, 2017]의 경우 PreviousColorSales는 [Red, 2020]으로 설정됩니다.

ORDERBY 추가

위의 정의는 다음과 같습니다.

DAX

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

```

SELECTCOLUMNS (
    OFFSET (
        -1,
        vRelation,
        ORDERBY ([Color], ASC, [CalendarYear], ASC,
        [CurrentYearSales], ASC)
    ),
    [CurrentYearSales]
)
)

```

이 경우 OFFSET 호출은 ORDERBY를 사용하여 오름차순으로 Color 및 CalendarYear별로 테이블을 정렬합니다. 그러면 반환된 이전 행으로 간주되는 항목이 결정됩니다.

이러한 두 결과가 동일한 이유는 ORDERBY가 PARTITIONBY에 없는 관계의 모든 열을 자동으로 포함하기 때문입니다. PARTITIONBY를 지정하지 않았기 때문에 ORDERBY는 Color, CalendarYear 및 CurrentYearSales로 설정됩니다. 그러나 관계의 Color 및 CalendarYear 쌍은 고유하므로 CurrentYearSales를 추가해도 결과는 변경되지 않습니다. 실제로 ORDERBY에서 색만 지정하더라도 CalendarYear가 자동으로 추가되므로 결과는 동일합니다. 이는 각 행이 ORDERBY 및 PARTITIONBY 열로 고유하게 식별될 수 있도록 하기 위해 함수가 ORDERBY에 필요한 만큼 열을 추가하기 때문입니다.

DAX

```

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

```

PARTITIONBY 추가

이제 다음 계산 테이블 식과 같이 PARTITIONBY를 사용할 수 있는 결과를 거의 얻을 수 있습니다.

DAX

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

ORDERBY에는 PARTITIONBY에 지정되지 않은 관계의 모든 열이 자동으로 포함되므로 ORDERBY 지정은 선택 사항입니다. 따라서 ORDERBY가 CalendarYear 및 CurrentYearSales로 자동으로 설정되므로 다음 식은 동일한 결과를 반환합니다.

DAX

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

```
[CurrentYearSales]
```

```
)  
)
```

① 참고

ORDERBY는 CalendarYear 및 CurrentYearSales로 자동으로 설정되지만 추가될 순서에 대한 보장은 없습니다. CalendarYear 앞에 CurrentYearSales가 추가되면 결과 순서가 예상과 인라인되지 않습니다. **ORDERBY 및 PARTITIONBY를 지정할 때 혼동과 예기치 않은 결과를 방지하려면 명시적이어야 합니다.**

두 식 모두 다음과 같은 결과를 반환합니다.

색상	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"Black"	2017	393885	
"Black"	2018	1818835	393885
"Black"	2019	3981638	1818835
"Black"	2020	2644054	3981638
"파랑"	2019	994448	
"파랑"	2020	1284648	994448
"Multi"	2019	48622	
"Multi"	2020	57849	48622
"NA"를 입력하세요.	2019	207822	
"NA"를 입력하세요.	2020	227295	207822
"빨강"	2017	2961198	
"빨강"	2018	3686935	2961198
"빨강"	2019	900175	3686935
"빨강"	2020	176022	900175
"실버"	2017	326399	
"실버"	2018	750026	326399
"실버"	2019	2165176	750026

색상	CalendarYear	CurrentYearSales	PreviousYearSalesForSameColor
"실버"	2020	1871788	2165176
"흰색"	2019	2517	
"흰색"	2020	2589	2517
"노란색"	2018	163071	
"노란색"	2019	2072083	163071
"노란색"	2020	2621602	2072083

이 표에서 볼 수 있듯이 PreviousYearSalesForSameColor 열에는 같은 색에 대한 이전 연도의 판매액이 표시됩니다. [Red, 2020]의 경우 [Red, 2019] 등의 매출을 반환합니다. 예를 들어 [Red, 2017]의 경우와 같이 이전 연도가 없으면 값이 반환되지 않습니다.

PARTITIONBY는 테이블을 OFFSET 계산을 실행할 부분으로 나누는 방법으로 생각할 수 있습니다. 위의 예제에서 표는 각 색에 대해 하나씩 색이 있는 만큼의 부분으로 나뉩니다. 그런 다음 각 파트 내에서 OFFSET이 CalendarYear별로 정렬되어 계산됩니다.

시각적으로, 무슨 일이 일어나고 있는지는 다음과 같습니다.

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

먼저 PARTITIONBY를 호출하면 테이블이 각 색에 대해 하나씩 파트로 나뉩니다. 테이블 이미지의 연한 파란색 상자로 표시됩니다. 다음으로 ORDERBY는 각 파트가 CalendarYear(주황색 화살표로 표시됨)로 정렬되도록 합니다. 마지막으로 정렬된 각 부분 내에서 각 행에 대해 OFFSET는 위의 행을 찾아 PreviousYearSalesForSameColor 열에서 해당 값을 반환합니다. 각 부분의 모든 첫 번째 행에 대해 동일한 부분에 이전 행이 없으므로 PreviousYearSalesForSameColor 열에 대한 해당 행의 결과는 비어 있습니다.

최종 결과를 얻으려면 OFFSET 호출에서 반환한 것과 동일한 색에 대해 이전 연도 판매에서 CurrentYearSales를 빼기만 하면 됩니다. 동일한 색에 대한 이전 연도 판매를 표시하는 데 관심이 없기 때문에 현재 연도 매출과 연도별 차이에만 관심이 있습니다. 최종 계산 테이블 식은 다음과 같습니다.

DAX

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

해당 식의 결과는 다음과 같습니다.

색상	CalendarYear	CurrentYearSales	YoYSalesForSameColor
"Black"	2017	393885	393885
"Black"	2018	1818835	1424950
"Black"	2019	3981638	2162803
"Black"	2020	2644054	-1337584

색상	CalendarYear	CurrentYearSales	YoYSalesForSameColor
"파랑"	2019	994448	994448
"파랑"	2020	1284648	290200
"Multi"	2019	48622	48622
"Multi"	2020	57849	9227
"NA"를 입력하세요.	2019	207822	207822
"NA"를 입력하세요.	2020	227295	19473
"빨강"	2017	2961198	2961198
"빨강"	2018	3686935	725737
"빨강"	2019	900175	-2786760
"빨강"	2020	176022	-724153
"실버"	2017	326399	326399
"실버"	2018	750026	423627
"실버"	2019	2165176	1415150
"실버"	2020	1871788	-293388
"흰색"	2019	2517	2517
"흰색"	2020	2589	72
"노란색"	2018	163071	163071
"노란색"	2019	2072083	1909012
"노란색"	2020	2621602	549519

MATCHBY 사용

MATCHBY를 전혀 지정하지 않은 것을 알 수 있습니다. 이 경우 필요하지 않습니다. ORDERBY 및 PARTITIONBY의 열(위 예제에서 지정한 한)은 각 행을 고유하게 식별하기에 충분합니다. MATCHBY를 지정하지 않았기 때문에 ORDERBY 및 PARTITIONBY에 지정된 열은 각 행을 고유하게 식별하는 데 사용되므로 OFFSET을 비교하여 의미 있는 결과를 제공할 수 있습니다. ORDERBY 및 PARTITIONBY의 열이 각 행을 고유하게 식별할 수 없는 경우 추가 열이 각 행을 고유하게 식별하도록 허용하는 경우 ORDERBY 절에 추가 열을 추가 할 수 있습니다. 가능하지 않으면 오류가 반환됩니다. 이 마지막 경우 MATCHBY를 지정하면 오류를 해결하는 데 도움이 될 수 있습니다.

MATCHBY를 지정하면 MATCHBY 및 PARTITIONBY의 열이 각 행을 고유하게 식별하는 데 사용됩니다. 가능하지 않으면 오류가 반환됩니다. MATCHBY가 필요하지 않더라도 혼동을 방지하기 위해 MATCHBY를 명시적으로 지정하는 것이 좋습니다.

위의 예제에서 계속하면 마지막 식은 다음과 같습니다.

DAX

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

행을 고유하게 식별하는 방법에 대해 명시적으로 지정하려면 다음 같은 식과 같이 MATCHBY를 지정할 수 있습니다.

DAX

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

```

        -1,
        vRelation,
        ORDERBY ([CalendarYear]),
        PARTITIONBY ([Color]),
        MATCHBY ([Color], [CalendarYear])
    ),
    [CurrentYearSales]
)
)

```

MATCHBY가 지정되었으므로 MATCHBY와 PARTITIONBY에 지정된 두 열은 모두 행을 고유하게 식별하는 데 사용됩니다. COLOR는 MATCHBY와 PARTITIONBY 모두에서 지정되므로 다음 식은 이전 식과 동일합니다.

DAX

```

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

```

지금까지 살펴본 예제에서는 MATCHBY를 지정할 필요가 없으므로 MATCHBY가 필요한 약간 다른 예제를 살펴보겠습니다. 이 경우 주문 줄 목록이 있습니다. 각 행은 주문에 대한 주문 줄을 나타냅니다. 주문에는 여러 주문 줄이 있을 수 있으며 주문 줄 1은 여러 주문에 표시됩니다. 또한 각 주문 라인에는 ProductKey 및 SalesAmount가 있습니다. 테이블의 관련 열 샘플은 다음과 같습니다.

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount
SO51900	1	528	4.99
SO51948	1	528	5.99
SO52043	1	528	4.99
SO52045	1	528	4.99
SO52094	1	528	4.99
SO52175	1	528	4.99
SO52190	1	528	4.99
SO52232	1	528	4.99
SO52234	1	528	4.99
SO52234	2	529	3.99

행을 고유하게 식별하려면 SalesOrderNumber와 SalesOrderLineNumber가 모두 필요합니다.

각 주문에 대해 SalesAmount에서 내림차순으로 주문한 동일한 제품의 이전 판매액 (ProductKey로 표시)을 반환하려고 합니다. 다음 식은 OFFSET에 전달될 때 vRelation에 잠재적으로 여러 행이 있기 때문에 작동하지 않습니다.

DAX

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

이 식은 "OFFSET의 Relation 매개 변수에 중복 행이 있을 수 있으며 이는 허용되지 않습니다."라는 오류를 반환합니다.

이 식을 작동하려면 MATCHBY를 지정해야 하며 행을 고유하게 정의하는 모든 열을 포함해야 합니다. 관계인 FactInternetSales에는 명시적 키 또는 고유 열이 없기 때문에 MATCHBY가 필요합니다. 그러나 SalesOrderNumber와 SalesOrderLineNumber 열은 복합 키를 형성합니다. 여기서 함께 존재하는 항목은 관계 내에서 고유하므로 각 행을 고유하게 식별할 수 있습니다. 두 열에 반복 값이 포함되어 있으므로 SalesOrderNumber 또는 SalesOrderLineNumber를 지정하는 것만으로는 충분하지 않습니다. 다음 식은 문제를 해결합니다.

DAX

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

그리고 이 식은 실제로 다음과 같은 결과를 반환합니다.

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount	이전 판매액
SO51900	1	528	5.99	
SO51948	1	528	4.99	5.99
SO52043	1	528	4.99	4.99
SO52045	1	528	4.99	4.99
SO52094	1	528	4.99	4.99
SO52175	1	528	4.99	4.99
SO52190	1	528	4.99	4.99
SO52232	1	528	4.99	4.99
SO52234	1	528	4.99	4.99

SalesOrderNumber	SalesOrderLineNumber	ProductKey	SalesAmount	이전 판매액
SO52234	2	529	3.99	

참고 항목

[Orderby](#)

[PARTITIONBY](#)

[MATCHBY](#)

[INDEX](#)

[OFFSET](#)

[WINDOW](#)

[순위](#)

[ROWNUMBER](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

[Microsoft Q&A에서 도움말 보기](#)

오류 함수를 적절하게 사용

아티클 • 2023. 08. 27.

데이터 모델러로서 평가 시간 오류를 발생시키는 DAX 식을 작성할 때 두 가지 유용한 DAX 함수를 사용하는 것이 좋습니다.

- ISERROR [함수는](#) 단일 식을 사용하고 해당 식에서 오류가 발생하는 경우 TRUE를 반환합니다.
- [두 식을 사용하는 IFERROR](#) 함수입니다. 첫 번째 식에서 오류가 발생하면 두 번째 식의 값이 반환됩니다. 실제로 ISERROR 함수를 IF [함수 내에 중첩하는 보다 최적화된 구현입니다.](#)

그러나 이러한 함수는 유용할 수 있으며 이해하기 쉬운 식을 작성하는 데 기여할 수 있지만 계산 성능을 크게 저하시킬 수도 있습니다. 이러한 함수는 필요한 스토리지 엔진 검색 수를 늘리기 때문에 발생할 수 있습니다.

대부분의 평가 시간 오류는 예기치 않은 BLANK 또는 0 값 또는 잘못된 데이터 형식 변환으로 인해 발생합니다.

권장 사항

ISERROR 및 IFERROR 함수를 사용하지 않는 것이 좋습니다. 대신 모델을 개발하고 식을 작성할 때 방어 전략을 적용합니다. 전략에는 다음이 포함될 수 있습니다.

- 품질 데이터가 모델에 로드되었는지 확인: 파워 쿼리 변환을 사용하여 유효하지 않거나 누락된 값을 제거하거나 대체하고 올바른 데이터 형식을 설정합니다. 파워 쿼리 변환을 사용하여 잘못된 데이터 변환과 같은 오류가 발생할 때 행을 필터링할 수도 있습니다.

모델 열 Is Nullable 속성을 Off로 설정하여 데이터 품질을 제어할 수도 있습니다. 그러면 BLANK가 발견되면 데이터 새로 고침이 실패합니다. 이 오류가 발생하면 성공적인 새로 고침의 결과로 로드된 데이터가 테이블에 유지됩니다.

- IF 함수 사용: IF 함수 논리 테스트 식은 오류 결과가 발생할지 여부를 결정할 수 있습니다. ISERROR 및 IFERROR 함수와 마찬가지로 이 함수는 추가 스토리지 엔진 검색을 발생시킬 수 있지만 오류가 발생하지 않을 경우 보다 성능이 향상될 수 있습니다.
- 오류 내성 함수 사용: 일부 DAX 함수는 오류 조건을 테스트하고 보정합니다. 이러한 함수를 사용하면 대신 반환될 대체 결과를 입력할 수 있습니다. DIVIDE 함수는 이러한 예제 중 하나입니다. 이 함수 [에 대한 추가 지침은 DAX: DIVIDE 함수와 나누기 연산자\(/\)](#) 문서를 참조하세요.

예시

다음 측정값 식은 오류가 발생하는지 여부를 테스트합니다. 이 인스턴스에서는 BLANK를 반환합니다(IF 함수에 value-if-false 식을 제공하지 않는 경우).

DAX

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

이 측정값 식의 다음 버전은 IF 및 ISERROR 함수 대신 IFERROR 함수를 사용하여 개선되었습니다.

DAX

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

그러나 이 측정값 식의 최종 버전은 동일한 결과를 달성하지만 더 효율적이고 우아하게 수행됩니다.

DAX

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

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 문의해 보세요.](#)
- 제안 사항은? [Power BI를 개선하기 위한 아이디어 기여](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

[Microsoft Q&A에서 도움말 보기](#)

공백을 값으로 변환하지 마세요.

아티클 • 2023. 10. 05.

데이터 모델러로서 측정값 식을 작성할 때 의미 있는 값을 반환할 수 없는 경우가 발생할 수 있습니다. 이러한 경우 0과 같은 값을 반환하는 것이 좋습니다. 이 디자인이 효율적이고 실용적인지 신중하게 결정하는 것이 좋습니다.

BLANK 결과를 0으로 명시적으로 변환하는 다음 측정값 정의를 고려합니다.

DAX

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

BLANK 결과를 0으로 변환하는 또 다른 측정값 정의를 고려합니다.

DAX

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

DIVIDE 함수는 Profit 측정값을 Sales 측정값으로 나눕니다. 결과가 0이거나 BLANK이면 세 번째 인수인 대체 결과(선택 사항)가 반환됩니다. 이 예제에서는 0이 대체 결과로 전달되므로 측정값은 항상 값을 반환하도록 보장됩니다.

이러한 측정값 디자인은 비효율적이며 보고서 디자인이 저하됩니다.

보고서 시각적 개체에 추가되면 Power BI는 필터 컨텍스트 내의 모든 그룹화 검색을 시도합니다. 대용량 쿼리 결과를 평가 및 검색하면 보고서 렌더링 속도가 느려지는 경우가 많습니다. 각 예제 측정값은 스파스 계산을 조밀한 계산으로 효과적으로 전환하여 Power BI가 필요한 것보다 더 많은 메모리를 사용하도록 합니다.

또한 너무 많은 그룹화가 보고서 사용자를 압도하는 경우가 많습니다.

Profit Margin 측정값이 테이블 시각적 개체에 추가되어 고객별로 그룹화되면 어떻게 되는지 살펴보겠습니다.

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

테이블 시각적 개체는 압도적인 수의 행을 표시합니다. (실제로 모델에는 18,484 명의 고객이 있으므로 테이블에서 모든 고객을 표시하려고 시도합니다.) 보기에서 고객은 어떤 판매도 달성하지 못했습니다. 그러나 Profit Margin 측정값은 항상 값을 반환하므로 표시됩니다.

① 참고

시각적 개체에 표시할 데이터 요소가 너무 많은 경우 Power BI는 데이터 감소 전략을 사용하여 큰 쿼리 결과를 제거하거나 요약할 수 있습니다. 자세한 내용은 시각적 유형별 데이터 요소 제한 및 전략을 참조 [하세요](#).

이익률 측정값 정의가 개선되면 어떻게 되는지 살펴보겠습니다. 이제 Sales 측정값이 BLANK(또는 0)가 아닌 경우에만 값을 반환합니다.

DAX

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

이제 테이블 시각적 개체는 현재 필터 컨텍스트 내에서 판매한 고객만 표시합니다. 향상된 측정값은 보고서 사용자에게 보다 효율적이고 실용적인 환경을 제공합니다.

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

💡 팁

필요한 경우 데이터 없는 항목 표시 옵션을 사용하도록 설정하여 필터 컨텍스트 내에서 모든 그룹화(값을 반환하거나 BLANK)를 표시하도록 시각적 개체를 구성할 수 있습니다.

권장

의미 있는 값을 반환할 수 없는 경우 측정값이 BLANK를 반환하는 것이 좋습니다.

이 디자인 방법은 효율적이므로 Power BI에서 보고서를 더 빠르게 렌더링할 수 있습니다. 또한 기본적으로 보고서 시각적 개체는 요약이 BLANK인 경우 그룹화를 제거하므로 BLANK를 반환하는 것이 좋습니다.

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다.](#)
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

FILTER를 필터 인수로 사용하지 마세요.

아티클 • 2023. 10. 05.

데이터 모델러는 일반적으로 수정된 필터 컨텍스트에서 평가해야 하는 DAX 식을 작성합니다. 예를 들어 측정값 정의를 작성하여 "고마진 제품"의 매출을 계산할 수 있습니다. 이 계산은 이 문서의 뒷부분에서 설명합니다.

① 참고

이 문서는 가져오기 테이블에 필터를 적용하는 모델 계산과 특히 관련이 있습니다.

CALCULATE 및 **CALCULATETABLE** DAX 함수는 중요하고 유용한 함수입니다. 필터를 제거하거나 추가하는 계산을 작성하거나 관계 경로를 수정할 수 있습니다. 부울 식, 테이블 식 또는 특수 필터 함수인 필터 인수를 전달하여 수행됩니다. 이 문서에서는 부울 및 테이블 식에 대해서만 설명합니다.

테이블 식을 사용하여 빨간색 제품 판매를 계산하는 다음 측정값 정의를 고려합니다.
Product 테이블에 적용할 수 있는 모든 필터를 대체합니다.

DAX

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

CALCULATE 함수는 PRODUCT 테이블의 각 행에 대한 필터 식을 평가하는 FILTER DAX 함수에서 반환된 테이블 식을 허용합니다. 올바른 결과(빨간색 제품에 대한 판매 결과)를 달성합니다. 그러나 부울 식을 사용하여 훨씬 더 효율적으로 달성할 수 있습니다.

다음은 테이블 식 대신 부울 식을 사용하는 향상된 측정값 정의입니다. KEEPFILTERS DAX 함수는 색 열에 적용된 기존 필터가 덮어쓰여지지 않고 유지되도록 합니다.

DAX

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

가능하면 필터 인수를 부울 식으로 전달하는 것이 좋습니다. 가져오기 모델 테이블이 메모리 내 열 저장소이기 때문입니다. 이러한 방식으로 열을 효율적으로 필터링하도록 명시적으로 최적화됩니다.

그러나 부울 식이 필터 인수로 사용될 때 적용되는 제한 사항이 있습니다. 제한은 다음과 같습니다.

- 여러 테이블의 열을 참조할 수 없습니다.
- 측정값을 참조할 수 없음
- 중첩된 CALCULATE 함수를 사용할 수 없음
- 테이블을 검색하거나 반환하는 함수를 사용할 수 없음

즉, 더 복잡한 필터 요구 사항에 대해 테이블 식을 사용해야 합니다.

이제 다른 측정값 정의를 고려합니다. 요구 사항은 수익을 달성한 월에 대해서만 판매를 계산하는 것입니다.

DAX

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

이 예제에서는 FILTER 함수를 사용해야 합니다. 수익을 달성하지 못한 달을 제거하기 위해 이의 **측정값을 평가해야** 하기 때문입니다. 필터 인수로 사용되는 경우 부울 식에서 측정값을 사용할 수 없습니다.

권장 사항

최상의 성능을 위해 가능한 한 부울 식을 필터 인수로 사용하는 것이 좋습니다.

따라서 FILTER 함수는 필요한 경우에만 사용해야 합니다. 이를 사용하여 복잡한 열 비교를 필터링할 수 있습니다. 이러한 열 비교에는 다음이 포함될 수 있습니다.

- 측정값 그룹
- 기타 열
- OR DAX 함수 또는 OR 논리 연산자 사용(||)

참고 항목

- 필터 함수(DAX)
 - 학습 경로: [Power BI Desktop](#)에서 DAX 사용
 - 궁금한 점이 더 있나요? [Power BI 커뮤니티](#)에 질문합니다. ↗
 - 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#) ↗
-

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

열 및 측정값 참조

아티클 • 2023. 10. 05.

데이터 모델러인 DAX 식은 모델 열 및 측정값을 참조합니다. 열 및 측정값은 항상 모델 테이블과 연결되지만 이러한 연결은 다르므로 식에서 참조하는 방법에 대한 권장 사항은 서로 다릅니다.

열

열은 테이블 수준 개체이며 열 이름은 테이블 내에서 고유해야 합니다. 따라서 동일한 열 이름이 모델에서 여러 번 사용되어 서로 다른 테이블에 속할 수 있습니다. 한 가지 규칙이 더 있습니다. 열 이름은 동일한 테이블에 있는 측정값 이름 또는 계층 이름과 같은 이름을 가질 수 없습니다.

일반적으로 DAX는 열에 대한 정규화된 참조를 강제로 사용하지 않습니다. 정규화된 참조는 테이블 이름이 열 이름 앞에 있음을 의미합니다.

다음은 열 이름 참조만 사용하는 계산 열 정의의 예입니다. Sales 및 Cost 열은 모두 Orders라는 테이블에 속합니다.

DAX

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

정규화된 열 참조를 사용하여 동일한 정의를 다시 작성할 수 있습니다.

DAX

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

그러나 Power BI가 모호성을 감지할 때 정규화된 열 참조를 사용해야 하는 경우도 있습니다. 수식을 입력하면 빨간색 물결선과 오류 메시지가 표시됩니다. 또한 LOOKUPVALUE DAX 함수와 같은 일부 DAX 함수에는 정규화된 열을 사용해야 합니다.

항상 열 참조를 정규화하는 것이 좋습니다. 이유는 권장 사항 [섹션](#)에 제공됩니다.

측정값 그룹

측정값은 모델 수준 개체입니다. 이러한 이유 때문에 측정 이름은 모델 내에서 고유해야 합니다. 그러나 필드 창에서 보고서 작성자는 단일 모델 테이블과 연결된 각 측정값을 볼 수 있습니다. 이 연결은 미용상의 이유로 설정되며 측정값에 대한 흄 테이블 속성을 설정

하여 구성할 수 있습니다. 자세한 내용은 [Power BI Desktop의 측정값\(측정값 구성\)](#)을 참조하세요.

식에서 정규화된 측정값을 사용할 수 있습니다. DAX intellisense는 제안도 제공합니다. 그러나 필수는 아니며 권장되는 방법은 아닙니다. 측정값에 대한 흄 테이블을 변경하면 정규화된 측정값 참조를 사용하는 식이 중단됩니다. 그런 다음 각 끊어진 수식을 편집하여 측정값 참조를 제거(또는 업데이트)해야 합니다.

측정값 참조를 한정하지 않는 것이 좋습니다. 이유는 권장 사항 [섹션](#)에 제공됩니다.

권장 사항

권장 사항은 간단하고 기억하기 쉽습니다.

- 항상 정규화된 열 참조 사용
- 정규화된 측정값 참조를 사용하지 마세요.

이유는 다음과 같습니다.

- **수식 항목:** 확인할 모호한 참조가 없으므로 식이 허용됩니다. 또한 정규화된 열 참조가 필요한 DAX 함수에 대한 요구 사항을 충족합니다.
- **견고성:** 측정값 흄 테이블 속성을 변경하더라도 식은 계속 작동합니다.
- **가독성:** 식은 빠르고 쉽게 이해할 수 있습니다. 정규화된 식인지 여부에 따라 열인지 측정값인지를 빠르게 확인할 수 있습니다.

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다.](#)
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

DIVIDE 함수 및 나누기 연산자(/)

아티클 • 2023. 10. 05.

데이터 모델러로서 분모를 분모로 나누는 DAX 식을 작성할 때 DIVIDE 함수 또는 나누기 연산자(/- 슬래시)를 사용하도록 선택할 수 있습니다.

DIVIDE 함수를 사용하는 경우 숫자 및 분모 식을 전달해야 합니다. 선택적으로 대체 결과를 나타내는 값을 전달할 수 있습니다.

DAX

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

DIVIDE 함수는 0으로 나누기 사례를 자동으로 처리하도록 설계되었습니다. 대체 결과가 전달되지 않고 분모가 0 또는 BLANK이면 함수는 BLANK를 반환합니다. 전달된 대체 결과가 있으면 BLANK 대신 반환됩니다.

DIVIDE 함수는 분모 값을 먼저 테스트할 필요가 없도록 식을 저장하기 때문에 편리합니다. 또한 이 함수는 IF 함수보다 분모 값을 테스트하는 데 더 적합합니다. 0으로 나누기 검사 비용이 많이 들기 때문에 성능 향상이 중요합니다. 또한 DIVIDE를 사용하면 더욱 간결하고 우아한 표현이 됩니다.

예시

다음 측정값 식은 안전한 나누기를 생성하지만 4개의 DAX 함수를 사용합니다.

DAX

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

이 측정값 식은 동일한 결과를 달성하지만 더 효율적이고 우아하게 수행됩니다.

DAX

```
Profit Margin =
```

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

권장 사항

분모가 0 또는 BLANK를 반환할 수 있는 식일 때마다 *DIVIDE* 함수를 사용하는 것이 좋습니다.

분모가 상수 값인 경우 나누기 연산자를 사용하는 것이 좋습니다. 이 경우 나누기가 성공으로 보장되며 불필요한 테스트를 방지하기 때문에 식이 더 잘 수행됩니다.

DIVIDE 함수가 대체 값을 반환해야 하는지 여부를 신중하게 고려합니다. 측정값의 경우 일반적으로 BLANK를 반환하는 것이 더 나은 디자인입니다. 기본적으로 보고서 시각적 개체는 요약이 BLANK일 때 그룹화를 제거하므로 BLANK를 반환하는 것이 좋습니다. 이를 통해 시각적 개체는 데이터가 있는 그룹에 집중할 수 있습니다. 필요한 경우 Power BI에서 데이터 옵션 없이 항목 표시를 사용하도록 설정하여 필터 컨텍스트 내에서 모든 그룹(값을 반환하거나 BLANK)을 표시하도록 시각적 개체를 구성할 수 있습니다.

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다.](#)
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

값 대신 SELECTEDVALUE 사용

아티클 • 2023. 10. 05.

데이터 모델러로서 열이 특정 값으로 필터링되는지 여부를 테스트하는 DAX 식을 작성해야 하는 경우가 있습니다.

이전 버전의 DAX에서는 세 가지 DAX 함수가 포함된 패턴을 사용하여 이 요구 사항을 안전하게 달성했습니다. [IF](#), [HASONEVALUE](#) 및 [VALUES](#). 다음 측정값 정의는 예제를 제공합니다. 그것은 판매세 금액을 계산하지만, 호주 고객에게 만든 판매에 대해서만.

DAX

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

이 예제에서 HASONEVALUE 함수는 현재 필터 컨텍스트에서 Country-Region 열의 단일 값이 표시되는 경우에만 TRUE를 반환합니다. TRUE이면 VALUES 함수가 리터럴 텍스트 "Australia"에 비교됩니다. VALUES 함수가 TRUE를 반환하면 Sales 측정값에 0.10(10%)을 곱합니다. 두 개 이상의 값이 열을 필터링하기 때문에 HASONEVALUE 함수가 FALSE를 반환하는 경우 첫 번째 IF 함수는 BLANK를 반환합니다.

HASONEVALUE의 사용은 방어 기술입니다. 여러 값이 국가-지역 열을 필터링할 수 있기 때문에 필요합니다. 이 경우 VALUES 함수는 여러 행의 테이블을 반환합니다. 여러 행의 테이블을 스칼라 값과 비교하면 오류가 발생합니다.

권장

SELECTEDVALUE [함수를](#) 사용하는 것이 좋습니다. 이 문서에서 설명하는 패턴과 동일한 결과를 얻을 수 있지만 더 효율적이고 우아하게 수행됩니다.

SELECTEDVALUE 함수를 사용하여 이제 측정값 정의 예제를 다시 작성합니다.

DAX

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

```
[Sales] * 0.10
```

```
)
```

💡 팁

SELECTEDVALUE 함수에 대체 결과 값을 전달할 수 있습니다. 필터 또는 여러 필터가 열에 적용되지 않을 때 대체 결과 값이 반환됩니다.

참고 항목

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다.](#) ↗
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#) ↗

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUNT 대신 COUNTROWS 사용

아티클 • 2023. 12. 14.

데이터 모델러로서 테이블 행을 계산하는 DAX 식을 작성해야 하는 경우도 있습니다. 테이블은 모델 테이블이거나 테이블을 반환하는 식일 수 있습니다.

요구 사항은 두 가지 방법으로 달성할 수 있습니다. COUNT 함수를 사용하여 열 값을 계산하거나 COUNTROWS 함수를 사용하여 테이블 행 수를 계산할 수 있습니다. 계산된 열에 BLANK가 없는 경우 두 함수 모두 동일한 결과를 얻을 수 있습니다.

다음 측정값 정의는 예제를 제공합니다. OrderDate 열 값의 수를 계산합니다.

DAX

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

Sales 테이블의 세분성이 판매 주문당 한 행이고 OrderDate 열에 BLANK가 포함되지 않은 경우 측정값은 올바른 결과를 반환합니다.

그러나 다음 측정값 정의는 더 나은 솔루션입니다.

DAX

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

두 번째 측정값 정의가 더 나은 세 가지 이유가 있습니다.

- 더 효율적이므로 더 나은 성능을 발휘할 수 있습니다.
- 테이블의 열에 포함된 BLANK는 고려하지 않습니다.
- 수식의 의도는 자기 설명의 시점까지 더 명확합니다.

권장

테이블 행을 계산하려는 경우 항상 COUNTROWS 함수를 사용하는 것이 좋습니다.

관련 콘텐츠

- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 질문합니다.](#)
- 제안 사항은? [Power BI 개선을 위한 아이디어 제공](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

변수를 사용하여 DAX 수식 개선

아티클 • 2023. 08. 27.

데이터 모델러로서 일부 DAX 계산을 작성하고 디버깅하는 것은 어려울 수 있습니다. 복잡한 계산 요구 사항에는 종종 복합 식이나 복잡한 식을 작성하는 것이 포함됩니다. 복합 식에는 많은 중첩 함수를 사용하고 식 논리를 다시 사용할 수 있습니다.

DAX 수식에서 변수를 사용하면 더 복잡하고 효율적인 계산을 작성하는 데 도움이 될 수 있습니다. 변수는 성능, 안정성, 가독성을 향상시키고 복잡성을 줄일 수 있습니다.

이 문서에서는 전년 대비(YoY) 매출 증가에 대한 예제 측정값을 사용하여 처음 세 가지 이점을 설명합니다. (YoY 매출 증가에 대한 공식은 작년 같은 기간의 매출을 뺀 기간 판매이며, 작년 같은 기간의 판매액으로 나뉩니다.)

다음 측정값 정의부터 시작해 보겠습니다.

DAX

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

측정값은 올바른 결과를 생성하지만 이제 개선할 수 있는 방법을 살펴보겠습니다.

성능 향상

수식은 "작년 같은 기간"을 계산하는 식을 반복합니다. 이 수식은 Power BI가 동일한 식을 두 번 평가해야 하므로 비효율적입니다. 측정값 정의는 변수 VAR을 사용하여 보다 효율적으로 만들 수 있습니다.

다음 측정값 정의는 향상된 것을 나타냅니다. 식을 사용하여 SalesPriorYear라는 변수에 "작년 같은 기간" 결과를 할당합니다. 그런 다음 변수는 RETURN 식에서 두 번 사용됩니다.

DAX

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

```
RETURN
```

```
DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
```

측정값은 계속해서 올바른 결과를 생성하며 쿼리 시간의 약 절반 정도에 해당합니다.

가독성 향상

이전 측정값 정의에서 변수 이름을 선택하면 RETURN 식을 더 쉽게 이해할 수 있습니다. 식은 짧고 자기 설명입니다.

디버깅 간소화

변수는 수식을 디버그하는 데 도움이 될 수도 있습니다. 변수에 할당된 식을 테스트하려면 RETURN 식을 일시적으로 다시 작성하여 변수를 출력합니다.

다음 측정값 정의는 SalesPriorYear **변수만** 반환합니다. 의도한 RETURN 식을 주석으로 처리하는 방법을 확인합니다. 이 기술을 사용하면 디버깅이 완료되면 쉽게 되돌릴 수 있습니다.

```
DAX
```

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

복잡성 감소

이전 버전의 DAX에서는 변수가 아직 지원되지 않았습니다. 새 필터 컨텍스트를 도입한 복잡한 식은 EARLIER 또는 EARLIEST **DAX 함수를 사용하여 외부 필터 컨텍스트를 참조해야** 했습니다. 아쉽게도 데이터 모델러는 이러한 함수를 이해하고 사용하기가 어려웠습니다.

변수는 항상 RETURN 식이 적용되는 필터 외부에서 평가됩니다. 이러한 이유로 수정된 필터 컨텍스트 내에서 변수를 사용하면 EARLIEST 함수와 동일한 결과를 얻을 수 있습니다. 따라서 EARLIER 또는 EARLIEST 함수를 사용하지 않도록 할 수 있습니다. 즉, 이제 덜 복잡하고 이해하기 쉬운 수식을 작성할 수 있습니다.

Subcategory 테이블에 추가된 다음 계산 열 정의를 고려합니다. Subcategory Sales 열 값은 기반으로 각 제품 하위 범주에 대한 순위를 평가합니다.

DAX

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

EARLIER 함수는 현재 행 컨텍스트에서 **Subcategory Sales** 열 값을 참조하는 데 사용됩니다.

EARLIER 함수 대신 변수를 사용하여 계산 열 정의를 개선할 수 있습니다.

CurrentSubcategorySales 변수는 **현재 행 컨텍스트 외 Subcategory Sales 열 값을 저장하고 RETURN 식은 수정된 필터 컨텍스트 내에서 사용합니다.**

DAX

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

참고 항목

- [VAR DAX 문서](#)
- 학습 경로: [Power BI Desktop에서 DAX 사용](#)
- 궁금한 점이 더 있나요? [Power BI 커뮤니티에 문의해 보세요.](#) ↗

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

[Microsoft Q&A에서 도움말 보기](#)

DAX 함수 참조

아티클 • 2023. 10. 20.

DAX 함수 참조는 DAX(데이터 분석 식) 수식에 사용되는 250개가 넘는 함수 각각에 대한 구문, 매개 변수, 반환 값 및 예제를 비롯한 자세한 정보를 제공합니다.

① 중요

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

이 섹션의 내용

[새 DAX 함수](#) - 이러한 함수는 새 함수이거나 크게 업데이트된 기존 함수입니다.

[집계 함수](#) - 이러한 함수는 식에 정의된 대로 열 또는 테이블의 모든 행에 대한 개수, 합계, 평균, 최소값 또는 최대값과 같은 (스칼라) 값을 계산합니다.

[날짜 및 시간 함수](#) - DAX의 이러한 함수는 Microsoft Excel의 날짜 및 시간 함수와 유사합니다. 하지만 DAX 함수는 Microsoft SQL Server에서 사용되는 datetime 데이터 형식을 기본으로 합니다.

[필터 함수](#) - 이러한 함수는 특정 데이터 형식을 반환하고, 관련 테이블에서 값을 조회하고, 관련 값을 필터링하는 데 도움이 됩니다. 조회 함수는 테이블과 테이블 간의 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 살펴보고 값이 예상 형식과 일치하는지 여부를 반환합니다. 예를 들어 참조하는 값에 오류가 포함된 경우 ISERROR 함수는 TRUE를 반환합니다.

[논리 함수](#) - 이러한 함수는 식의 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 유사합니다. 그러나 DAX 함수에서 사용하는 숫자 데이터 형식에는 몇 가지 차이점이 있습니다.

[다른 함수](#) - 이러한 함수는 대부분의 다른 함수가 속한 범주에서 정의할 수 없는 고유한 작업을 수행합니다.

부모 및 자식 함수 - 이러한 함수는 사용자가 데이터 모델에서 부모/자식 계층 구조로 표시되는 데이터를 관리하는 데 도움이 됩니다.

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

통계 함수 - 이러한 함수는 표준 편차 및 순열 수와 같은 통계 분포 및 확률과 관련된 값을 계산합니다.

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 일정 및 날짜에 대한 기본 제공 지식을 사용하는 계산을 만드는 데 도움이 됩니다. 시간 및 날짜 범위를 집계 또는 계산과 함께 사용하면 비교 가능한 기간에서 판매량, 재고 등에 대한 의미 있는 비교 자료를 작성할 수 있습니다.

참고 항목

[DAX 구문 참조](#)

[DAX 연산자 참조](#)

[DAX 매개 변수 명명 규칙](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

새로운 DAX 함수

아티클 • 2024. 02. 22.

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

함수와 기능이 항상 업데이트되는 동안 사용자에게 표시되는 기능 변경 내용이 있는 업데이트만 설명서에 설명되어 있습니다. 지난 1년 동안의 기존 함수에 대한 새로운 함수 및 업데이트가 여기에 표시됩니다.

① 중요

모든 버전의 Power BI Desktop, Analysis Services 및 Excel의 파워 피벗에서 모든 함수가 지원되는 것은 아닙니다. 새 함수와 업데이트된 함수는 일반적으로 Power BI Desktop에서 처음 도입된 다음 나중에 Analysis Services, Excel의 파워 피벗 및 도구에 도입됩니다.

새 함수

 테이블 확장

함수	Month	설명
첫 번째	2024년 1 월	시각적 계산에만 사용됩니다. 축의 첫 번째 행에서 시각적 행렬의 값을 검색합니다.
LAST	2024년 1 월	시각적 계산에만 사용됩니다. 축의 마지막 행에서 시각적 행렬의 값을 검색합니다.
다음	2024년 1 월	시각적 계산에만 사용됩니다. 시각적 행렬에서 축의 다음 행에 있는 값을 검색합니다.
이전	2024년 1 월	시각적 계산에만 사용됩니다. 시각적 행렬에서 축의 이전 행에 있는 값을 검색합니다.
MATCHBY	2023년 5 월	창 함수 식에서 데이터를 일치시키고 현재 행을 식별하는 데 사용되는 열을 정의합니다.
RANK	2023년 4 월	지정된 순서에 따라 정렬된 지정된 파티션 내의 현재 컨텍스트에 대한 순위를 반환합니다.
ROWNUMBER	2023년 4 월	지정된 순서에 따라 정렬된 지정된 파티션 내의 현재 컨텍스트에 대한

함수	Month	설명
	월	고유 순위를 반환합니다.
LINEST	2023년 2 월	최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산합니다.
LINESTX	2023년 2 월	최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산합니다. 테이블의 각 행에 대해 계산된 식의 데이터 결과입니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

집계 함수

아티클 • 2023. 10. 20.

집계 함수는 식에 정의된 대로 열 또는 테이블의 모든 행에 대한 개수, 합계, 평균, 최소값 또는 최대값과 같은 (스칼라) 값을 계산합니다.

이 범주에서

함수	설명
APPROXIMATEDISTINCTCOUNT	열의 예상 고유 값 수를 반환합니다.
평균	열에 있는 모든 숫자의 평균(산술 평균)을 반환합니다.
AVERAGEA	열에 있는 값의 평균(산술 평균)을 반환합니다.
Averagex	테이블에 대해 계산된 식 집합의 평균(산술 평균)을 계산합니다.
COUNT	비어 있지 않은 값이 포함된 지정된 열의 행 수를 계산합니다.
Counta	비어 있지 않은 값이 포함된 지정된 열의 행 수를 계산합니다.
COUNTAX	테이블에 있는 식의 결과를 평가할 때 비어 있지 않은 결과의 수를 계산합니다.
COUNTBLANK	열의 빈 셀 수를 계산합니다.
COUNTROWS	지정한 테이블이나 식으로 정의된 테이블의 행 수를 계산합니다.
COUNTX	테이블에 대한 식을 평가할 때 숫자 또는 숫자로 계산되는 식을 포함하는 행 수를 계산합니다.
DISTINCTCOUNT	열의 고유 값 수를 계산합니다.
DISTINCTCOUNTNOBLANK	열의 고유 값 수를 계산합니다.
MAX	열에서 또는 두 스칼라 식 사이에서 가장 큰 숫자 값을 반환합니다.
MAXA	열에서 가장 큰 값을 반환합니다.
맥스	테이블의 각 행에 대한 식을 계산하고 가장 큰 숫자 값을 반환합니다.
MIN	열에서 또는 두 스칼라 식 사이에서 가장 작은 숫자 값을 반환합니다.
Mina	텍스트로 표시되는 논리 값과 숫자를 포함하여 열에서 가장 작은 값을 반환합니다.

함수	설명
MINX	테이블의 각 행에 대한 식을 계산하여 발생하는 가장 작은 숫자 값을 반환합니다.
제품	열에 있는 숫자의 곱을 반환합니다.
PRODUCTX	테이블의 각 행에 대해 계산된 식의 곱을 반환합니다.
합계	열에 있는 모든 수를 더합니다.
SUMX	테이블의 각 행에 대해 계산된 식의 합계를 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

APPROXIMATEDISTINCTCOUNT

아티클 • 2023. 10. 20.

열의 예상 고유 값 수를 반환합니다. 이 함수는 쿼리 성능에 최적화되어 있지만 정확도가 약간 낮아지는 데이터 원본에서 해당 집계 작업을 호출합니다. 이 함수는 Azure SQL, Azure Synapse 전용 SQL 풀, BigQuery, Databricks 및 Snowflake 데이터 원본과 함께 사용 할 수 있습니다. 이 함수에는 DirectQuery 모드가 필요합니다. 가져오기 모드 및 이중 스토리지 모드는 지원되지 않습니다.

구문

DAX
APPROXIMATEDISTINCTCOUNT(<columnName>)

매개 변수

용어	설명
column	계산할 값이 들어 있는 열입니다. 식일 수 없습니다.

반환 값

열의 대략적인 고유 값 수입니다.

설명

이 함수의 유일한 인수는 열입니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있습니다. 계산할 행을 찾지 못한 함수는 BLANK를 반환하고, 그렇지 않으면 고유 값의 개수를 반환합니다.

피드백

이 페이지가 도움이 되었나요?

👍 Yes

👎 No

AVERAGE

아티클 • 2023. 10. 20.

열에 있는 모든 숫자의 평균(산술 평균)을 반환합니다.

구문

DAX

AVERAGE(<column>)

매개 변수

용어	정의
column	평균을 계산할 숫자가 들어 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. 숫자 집합으로 계산되는 식의 평균을 찾으려면 AVERAGEX 함수를 대신 사용합니다.
- 열의 숫자가 아닌 값은 다음과 같이 처리됩니다.
 - 열에 텍스트가 포함된 경우 집계를 수행할 수 없으며 함수는 공백을 반환합니다.
 - 열에 논리 값 또는 빈 셀이 포함된 경우 해당 값은 무시됩니다.
 - 값이 0인 셀이 포함됩니다.
- 셀의 평균을 계산할 때 빈 셀과 값이 0인 셀 간의 차이를 염두에 두어야 합니다. 셀에 0이 포함되면 숫자의 합계에 추가되고 행은 수로 사용되는 행 수 중에서 계산됩니다. 그러나 셀에 공백이 있는 경우 행은 계산되지 않습니다.
- 집계할 행이 없을 때마다 함수는 공백을 반환합니다. 그러나 행이 있지만 지정된 조건을 충족하지 않는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없으면 Excel에서도 0을 반환합니다.

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

예시

다음 수식은 Table의 ExtendedSalesAmount 열에 있는 값의 평균을 반환합니다.

DAX

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

관련된 함수

AVERAGEX 함수는 테이블의 각 행에 대해 계산되는 식을 인수로 사용할 수 있습니다. 이렇게 하면 계산을 수행한 다음 계산 값의 평균을 가져올 수 있습니다.

AVERAGEA 함수는 열을 인수로 사용합니다. 그렇지 않으면 같은 이름의 Excel 함수와 같습니다. AVERAGEA 함수를 사용하여 빈 값이 포함된 열의 평균을 계산할 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

AVERAGEA

아티클 • 2023. 12. 14.

열에 있는 값의 평균(산술 평균)을 반환합니다. 텍스트 및 숫자가 아닌 값을 처리합니다.

구문

DAX

`AVERAGEA(<column>)`

매개 변수

 테이블 확장

용어	정의
column	평균을 원하는 값이 들어 있는 열입니다.

반환 값

10진수입니다.

설명

- AVERAGEA 함수는 열을 사용하고 열의 숫자 평균을 계산하지만 다음 규칙에 따라 숫자가 아닌 데이터 형식도 처리합니다.
 - TRUE로 계산되는 값은 1로 계산됩니다.
 - FALSE로 계산되는 값은 0으로 계산됩니다.
 - 숫자가 아닌 텍스트가 포함된 값은 0으로 계산됩니다.
 - 빈 텍스트("")는 0으로 계산됩니다.
- 계산의 일부로 참조에 숫자의 논리적 값과 텍스트 표현을 포함하지 않으려면 AVERAGE 함수를 사용합니다.
- 집계할 행이 없을 때마다 함수는 공백을 반환합니다. 그러나 행이 있지만 지정된 조건을 충족하지 않는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없으면 Microsoft Excel에서도 0을 반환합니다.

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

예시

다음 예제에서는 다음 표에 지정된 참조된 열에 비어 있는 셀이 아닌 셀의 평균을 반환합니다. AVERAGE 함수를 사용한 경우 평균은 21/2입니다. AVERAGEA 함수를 사용하면 결과는 22/5입니다.

 테이블 확장

Transaction ID	수량	결과
0000123	1	1로 계산
0000124	20	20으로 계산
0000125	해당 없음	0으로 계산됩니다.
0000126		0으로 계산됩니다.
0000126	TRUE	1로 계산

DAX

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

관련 콘텐츠

[AVERAGE 함수](#)

[AVERAGEX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

AVERAGEX

아티클 • 2023. 12. 14.

테이블에 대해 계산된 식 집합의 평균(산술 평균)을 계산합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어 정의

table 테이블의 이름 또는 집계를 수행할 수 있는 테이블을 지정하는 식입니다.

식 스칼라 결과가 있는 식으로, 첫 번째 인수에서 테이블의 각 행에 대해 계산됩니다.

반환 값

10진수입니다.

설명

- AVERAGEX 함수를 사용하면 테이블의 각 행에 대한 식을 계산한 다음 결과 값 집합을 가져와서 산술 평균을 계산할 수 있습니다. 따라서 함수는 테이블을 첫 번째 인수로, 식을 두 번째 인수로 사용합니다.
- 다른 모든 면에서 AVERAGEX는 AVERAGE와 동일한 규칙을 따릅니다. 숫자가 아닌 셀이나 null 셀은 포함할 수 없습니다. 테이블 및 식 인수는 모두 필요합니다.
- 집계할 행이 없으면 함수는 공백을 반환합니다. 행이 있지만 지정된 조건을 충족하지 않는 경우 함수는 0을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 InternetSales 테이블의 각 주문에 대한 평균 화물 및 세금을 계산하고, 먼저 각 행에서 화물과 TaxAmt를 합산한 다음, 해당 합계를 평균화합니다.

DAX

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

두 번째 인수로 사용되는 식에서 여러 작업을 사용하는 경우 괄호를 사용하여 계산 순서를 제어해야 합니다. 자세한 내용은 [DAX 구문 참조](#)하세요.

관련 콘텐츠

[AVERAGE 함수](#)

[AVERAGEA 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

개수

아티클 • 2023. 10. 20.

비어 있지 않은 값이 포함된 지정된 열의 행 수를 계산합니다.

구문

DAX

`COUNT(<column>)`

매개 변수

용어	정의
column	계산할 값이 들어 있는 열입니다.

반환 값

정수입니다.

설명

- 이 함수에 허용되는 유일한 인수는 열입니다. COUNT 함수는 다음과 같은 종류의 값을 포함하는 행 수를 계산합니다.
 - 숫자
 - 날짜
 - 문자열
- 함수가 계산할 행을 찾지 않으면 공백이 반환됩니다.
- 빈 값은 건너뜁니다. TRUE/FALSE 값은 지원되지 않습니다.
- TRUE/FALSE 값의 열을 평가하려면 COUNTA 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.
- COUNT를 사용하는 경우 모범 사례는 COUNT 대신 COUNTROWS 사용을 참조 [하세요](#).

예시

다음 예제에서는 ShipDate 열의 값 수를 계산하는 방법을 보여 있습니다.

DAX

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

논리 값 또는 텍스트를 계산하려면 COUNTA 또는 COUNTAX 함수를 사용합니다.

참고 항목

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Counta

아티클 • 2023. 12. 14.

비어 있지 않은 값이 포함된 지정된 열의 행 수를 계산합니다.

구문

DAX

`COUNTA(<column>)`

매개 변수

 테이블 확장

용어	정의
column	계산할 값이 들어 있는 열입니다.

반환 값

정수입니다.

설명

- 함수가 계산할 행을 찾지 못하면 함수는 공백을 반환합니다.
- `COUNT`와 달리, COUNTA는 부울 데이터 형식을 지원합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 전화 번호를 저장하는 열에 `Reseller` 모든 종류의 값이 있는 테이블의 모든 행을 반환합니다.

DAX

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

관련 콘텐츠

[COUNT 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUNTAX

아티클 • 2023. 10. 20.

COUNTAX 함수는 테이블에 대해 식의 결과를 평가할 때 비어 있지 않은 결과의 수를 계산합니다. 즉, COUNTA 함수처럼 작동하지만 테이블의 행을 반복하고 지정된 식으로 인해 비어 없는 결과가 발생하는 행 수를 계산하는 데 사용됩니다.

구문

DAX

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

매개 변수

용어	정의
table	식이 계산될 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 계산됩니다.
- 함수가 집계할 행을 찾을 때마다 함수는 공백을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 [상태] = 활성에서 재판매인 테이블을 필터링한 결과 테이블을 사용하여 전화 열의 빌랭크 행 수를 계산합니다.

DAX

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

참고 항목

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUNTBLANK

아티클 • 2023. 12. 14.

열의 빈 셀 수를 계산합니다.

구문

DAX

`COUNTBLANK(<column>)`

매개 변수

 테이블 확장

용어	정의
column	계산할 빈 셀이 들어 있는 열입니다.

반환 값

정수입니다. 조건을 충족하는 행이 없으면 공백이 반환됩니다.

설명

- 이 함수에 허용되는 유일한 인수는 열입니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있지만 빈 셀만 계산됩니다. 값이 0인 셀은 계산되지 않습니다. 0은 공백이 아닌 숫자 값으로 간주됩니다.
- 집계할 행이 없을 때마다 함수는 공백을 반환합니다. 그러나 행이 있지만 지정된 조건을 충족하지 않는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없으면 Microsoft Excel에서도 0을 반환합니다.
- 즉, COUNTBLANK 함수가 공백을 찾지 못하면 결과는 0이 되지만 검사 행이 없으면 결과는 비어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 BankName에 대한 빈 값이 있는 테이블 재판매인의 행 수를 계산하는 방법을 보여 줍니다.

DAX

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

논리 값 또는 텍스트를 계산하려면 COUNTA 또는 COUNTAX 함수를 사용합니다.

관련 콘텐츠

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUNTROWS

아티클 • 2023. 10. 20.

COUNTROWS 함수는 지정된 테이블 또는 식으로 정의된 테이블의 행 수를 계산합니다.

구문

DAX

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

매개 변수

용어 정의

table (선택 사항) 계산할 행이 들어 있는 테이블의 이름 또는 테이블을 반환하는 식입니다. 제공되지 않은 경우 기본값은 현재 식의 홈 테이블입니다.

반환 값

정수입니다.

설명

- 이 함수는 기본 테이블의 행 수를 계산하는 데 사용할 수 있지만 테이블을 필터링하거나 테이블에 컨텍스트를 적용하여 발생하는 행 수를 계산하는 데 더 자주 사용됩니다.
- 집계할 행이 없을 때마다 함수는 공백을 반환합니다. 그러나 행이 있지만 지정된 조건을 충족하지 않는 경우 함수는 0을 반환합니다. 조건을 충족하는 행이 없으면 Microsoft Excel에서도 0을 반환합니다.
- COUNT 및 COUNTROWS를 사용하는 경우 모범 사례에 대한 자세한 내용은 DAX에서 COUNT 대신 COUNTROWS 사용을 참조하세요.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제에서는 테이블 Orders의 행 수를 계산하는 방법을 보여 줍니다. 예상 결과는 52761입니다.

DAX

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

예제 2

다음 예제에서는 행 컨텍스트에서 COUNTROWS를 사용하는 방법을 보여 줍니다. 이 시나리오에는 주문 번호와 관련된 두 가지 데이터 집합이 있습니다. 재판매인 테이블에는 각 재판매인에 대해 하나의 행이 포함됩니다. ResellerSales 테이블에는 각 주문에 대해 여러 행이 포함되며, 각 행에는 특정 재판매인에 대한 하나의 주문이 포함됩니다. 테이블은 ResellerKey 열의 관계로 연결됩니다.

수식은 ResellerKey의 값을 가져오고 동일한 재판매인 ID를 가진 관련 테이블의 행 수를 계산합니다. 결과는 CalculatedColumn1 열의 출력입니다.

DAX

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

다음 표에서는 예상 결과의 일부를 보여줍니다.

ResellerKey	CalculatedColumn1
1	73
2	70
3	394

참고 항목

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUNTX

아티클 • 2023. 12. 14.

테이블에 대한 식을 평가할 때 비어 있지 않은 값 또는 비어 있지 않은 값으로 계산되는 식을 포함하는 행의 수를 계산합니다.

구문

DAX

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

매개 변수

[+] 테이블 확장

용어	정의
table	계산할 행이 포함된 테이블입니다.
식	계산할 값이 들어 있는 값 집합을 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 검색되는 열 또는 식입니다.
- COUNTX 함수는 값, 날짜 또는 문자열만 계산합니다. 함수가 계산할 행을 찾지 못하면 공백이 반환됩니다.
- 논리 값을 계산하려면 COUNTAX 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 수식은 정가가 있는 Product 테이블의 모든 행 수를 반환합니다.

DAX

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

예제 2

다음 수식에서는 첫 번째 인수에 대해 필터링된 테이블을 COUNTX에 전달하는 방법을 보여 줍니다. 수식은 필터 식을 사용하여 Product 테이블의 ProductSubCategory = "Caps" 조건을 충족하는 행만 가져옵니다. 그런 다음 정가가 있는 결과 테이블의 행 수를 계산합니다. FILTER 식은 Products 테이블에 적용되지만 관련 테이블인 ProductSubCategory에서 조회하는 값을 사용합니다.

DAX

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

관련 콘텐츠

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DISTINCTCOUNT

아티클 • 2023. 12. 14.

열의 고유 값 수를 계산합니다.

구문

DAX

`DISTINCTCOUNT(<column>)`

매개 변수

 테이블 확장

용어	설명
column	계산할 값이 들어 있는 열입니다.

반환 값

열의 고유 값 수입니다.

설명

- 이 함수에 허용되는 유일한 인수는 열입니다. 모든 형식의 데이터를 포함하는 열을 사용할 수 있습니다. 계산할 행을 찾지 못한 함수는 BLANK를 반환하고, 그렇지 않으면 고유 값의 개수를 반환합니다.
- DISTINCTCOUNT 함수는 BLANK 값을 카운트합니다. BLANK 값을 건너뛰려면 DISTINCTCOUNTNOBLANK [함수를](#) 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 ResellerSales_USD[SalesOrderNumber] 열의 고유 판매 주문 수를 계산하는 방법을 보여 줍니다.

DAX

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

위에서 설명한 측정값을 옆면에 연도가 있고 맨 위에 제품 범주가 있는 테이블에서 위의 측정값을 사용하면 다음 결과가 반환됩니다.

테이블 확장

Row Labels	Accessories	Bikes	Clothing	Components	-	총합계
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521
2008	293	724	561	601		894
					1	1
총합계	1315	3153	2410	2646	1	3797

위의 예제에서는 총합계 번호 행이 합산되지 않습니다. 이는 동일한 주문에 여러 제품 범주의 품목이 동일한 순서로 포함될 수 있기 때문입니다.

관련 콘텐츠

[COUNT 함수](#)

[COUNTA 함수](#)

[COUNTAX 함수](#)

[COUNTX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DISTINCTCOUNTNOBLANK

아티클 • 2023. 12. 14.

열의 고유 값 수를 계산합니다.

구문

DAX

DISTINCTCOUNTNOBLANK (<column>)

매개 변수

 테이블 확장

용어	설명
column	계산할 값이 들어 있는 열입니다.

반환 값

열의 고유 값 수입니다.

설명

- [DISTINCTCOUNT](#) 함수와 달리 DISTINCTCOUNTNOBLANK는 빈 값의 수를 계산하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 ResellerSales_USD[SalesOrderNumber] 열의 고유 판매 주문 수를 계산하는 방법을 보여 줍니다.

DAX

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

DAX 쿼리

DAX

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

[] 테이블 확장

[DistinctCountNoBlank]	[DistinctCount]
2	3

관련 콘텐츠

[DISTINCTCOUNT](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MAX

아티클 • 2023. 10. 20.

열에서 또는 두 스칼라 식 사이에서 가장 큰 값을 반환합니다.

구문

DAX

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

DAX

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

매개 변수

용어	정의
column	가장 큰 값을 찾으려는 열입니다.
식	단일 값을 반환하는 DAX 식입니다.

반환 값

가장 큰 값입니다.

설명

- 두 식을 비교할 때 공백은 비교할 때 0으로 처리됩니다. 즉, Max(1, Blank())는 1을 반환하고 Max(-1, Blank())는 0을 반환합니다. 두 인수가 모두 비어 있으면 MAX는 공백을 반환합니다. 두 식 중 하나가 허용되지 않는 값을 반환하면 MAX는 오류를 반환합니다.
- TRUE/FALSE 값은 지원되지 않습니다. TRUE/FALSE 값의 열을 평가하려면 MAXA 함수를 사용합니다.

예 1

다음 예제에서는 InternetSales 테이블의 ExtendedAmount 열에 있는 가장 큰 값을 반환합니다.

DAX

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

예제 2

다음 예제에서는 두 식의 결과 간에 가장 큰 값을 반환합니다.

DAX

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

참고 항목

[MAXA 함수](#)

[MAXX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MAXA

아티클 • 2023. 12. 14.

열에서 가장 큰 값을 반환합니다.

구문

DAX
MAXA (<column>)

매개 변수

[+] 테이블 확장

용어	정의
column	가장 큰 값을 찾으려는 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 계산되는 행은 1로 계산됩니다. FALSE로 계산되는 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제에서는 정가와 재판매인 가격의 차이를 계산하는 ResellerMargin이라는 계산 열에서 가장 큰 값을 반환합니다.

DAX

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

예제 2

다음 예제에서는 날짜 및 시간을 포함하는 열에서 가장 큰 값을 반환합니다. 따라서 이 수식은 가장 최근의 트랜잭션 날짜를 가져옵니다.

DAX

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

관련 콘텐츠

[MAX 함수](#)

[MAXX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

맥스

아티클 • 2023. 12. 14.

테이블의 각 행에 대한 식을 계산하여 발생하는 가장 높은 값을 반환합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
variant	(선택 사항) TRUE이고 변형 또는 혼합 값 형식이 있는 경우 ORDER BY DESC를 기반으로 하는 가장 높은 값이 반환됩니다.

반환 값

가장 높은 값입니다.

설명

- MAXX 함수에 대한 테이블 **인수는** 테이블 이름 또는 테이블로 계산되는 식일 수 있습니다. 두 번째 인수는 테이블의 각 행에 대해 계산할 식을 나타냅니다.
- 평가할 값 중 다음 값만 계산됩니다.
 - 숫자
 - 텍스트
 - 날짜
- 빈 값은 건너뜁습니다. TRUE/FALSE 값은 지원되지 않습니다.

- 식에 텍스트 및 숫자와 같은 변형 또는 혼합 값 형식이 있는 경우 기본적으로 MAXX 는 숫자만 고려합니다. 이면 <variant> = TRUE 최대값이 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 수식에서는 식을 두 번째 인수로 사용하여 InternetSales 테이블의 각 주문에 대한 총 세금 및 배송액을 계산합니다. 예상 결과는 375.7184입니다.

DAX

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

예제 2

다음 수식은 먼저 FILTER 식을 사용하여 InternetSales 테이블을 필터링하여 [SalesTerritory] = 5로 정의된 특정 판매 지역에 대한 주문의 하위 집합을 반환합니다. MAXX 함수는 필터링된 테이블의 각 행에 대해 두 번째 인수로 사용되는 식을 평가하고 해당 주문에 대한 세금 및 배송에 대해 가장 높은 금액을 반환합니다. 예상 결과는 250.3724입니다.

DAX

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

관련 콘텐츠

[MAX 함수](#)

[MAXA 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MIN

아티클 • 2023. 10. 20.

열에서 또는 두 스칼라 식 사이에서 가장 작은 값을 반환합니다.

구문

DAX

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

DAX

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

매개 변수

용어	정의
column	가장 작은 값을 찾으려는 열입니다.
식	단일 값을 반환하는 DAX 식입니다.

반환 값

가장 작은 값입니다.

설명

- MIN 함수는 열 또는 두 개의 식을 인수로 사용하고 가장 작은 값을 반환합니다. 열의 값 유형은 다음과 같습니다.
 - 숫자
 - 텍스트
 - 날짜
 - 공백
- 식을 비교할 때 공백은 비교할 때 0으로 처리됩니다. 즉, Min(1,Blank())은 0을 반환하고 Min(-1, Blank())은 -1을 반환합니다. 두 인수가 모두 비어 있으면 MIN은 공백을

반환합니다. 두 식 중 하나가 허용되지 않는 값을 반환하면 MIN은 오류를 반환합니다.

- TRUE/FALSE 값은 지원되지 않습니다. TRUE/FALSE 값의 열을 평가하려면 MINA 함수를 사용합니다.

예 1

다음 예제에서는 계산 열 ResellerMargin에서 가장 작은 값을 반환합니다.

DAX

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

예제 2

다음 예제에서는 날짜 및 시간인 TransactionDate를 포함하는 열에서 가장 작은 값을 반환합니다. 따라서 이 수식은 가장 빠른 날짜를 반환합니다.

DAX

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

예 3

다음 예제에서는 두 스칼라 식의 결과에서 가장 작은 값을 반환합니다.

DAX

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

참고 항목

[MINA 함수](#)

[MINX 함수](#)

[통계 함수](#)

이 페이지가 도움이 되었나요?

 Yes

 No

MINA

아티클 • 2023. 10. 20.

열에서 가장 작은 값을 반환합니다.

구문

DAX

MINA(<column>)

매개 변수

용어	정의
column	최소값을 찾으려는 열입니다.

반환 값

가장 작은 값입니다.

설명

- MINA 함수는 숫자를 포함하는 열을 인수로 사용하고 다음과 같이 가장 작은 값을 결정합니다.
 - 열에 값이 없으면 MINA는 0을 반환합니다.
 - TRUE 및 FALSE와 같은 논리 값으로 계산되는 열의 행은 TRUE이면 1로 처리되고 FALSE인 경우 0으로 처리됩니다.
 - 빈 셀은 무시됩니다.
- 텍스트 값을 비교하려면 MIN 함수를 사용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 식은 InternetSales 테이블의 최소 화물 요금을 반환합니다.

DAX

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

예제 2

다음 식은 PostalCode 열의 최소값을 반환합니다. 열의 데이터 형식은 텍스트이므로 함수는 값을 찾지 못하고 수식은 0을 반환합니다.

DAX

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

참고 항목

[MIN 함수](#)

[MINX 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MINX

아티클 • 2023. 10. 20.

테이블의 각 행에 대한 식을 계산하여 발생하는 가장 낮은 값을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
variant	(선택 사항) TRUE이고 변형 또는 혼합 값 형식이 있는 경우 ORDER BY ASC를 기반으로 하는 가장 낮은 값이 반환됩니다.

반환 값

가장 낮은 값입니다.

설명

- MINX 함수는 테이블을 반환하는 식이나 테이블의 첫 번째 인수로 사용합니다. 두 번째 인수는 테이블의 각 행에 대해 계산되는 식을 포함합니다.
- 빈 값은 건너뜁니다. TRUE/FALSE 값은 지원되지 않습니다.
- 식에 텍스트 및 숫자와 같은 변형 또는 혼합 값 형식이 있는 경우 기본적으로 MINX는 숫자만 고려합니다. 이면 `<variant>` = TRUE 최소값이 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제에서는 InternetSales 테이블을 필터링하고 특정 판매 지역에 대한 행만 반환합니다. 그런 다음 수식은 Freight 열에서 최소값을 찾습니다.

DAX

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

예제 2

다음 예제에서는 이전 예제와 동일한 필터링된 테이블을 사용하지만 필터링된 테이블의 각 행에 대한 열의 값을 조회하는 대신, 함수는 두 열인 Freight 및 TaxAmt의 합계를 계산하고 해당 계산에서 생성된 가장 낮은 값을 반환합니다.

DAX

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

첫 번째 예제에서 열의 이름은 정규화되지 않습니다. 두 번째 예제에서는 열 이름이 정규화됩니다.

참고 항목

[MIN 함수](#)

[MINA 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PRODUCT

아티클 • 2023. 12. 14.

열에 있는 숫자의 곱을 반환합니다.

구문

DAX

`PRODUCT(<column>)`

매개 변수

 테이블 확장

용어	정의
column	제품을 계산할 숫자가 들어 있는 열입니다.

반환 값

10진수입니다.

설명

- 테이블의 각 행에 대해 계산된 식의 곱을 반환하려면 PRODUCTX 함수를 사용합니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다. 예를 들면 다음과 같습니다.

`PRODUCT(Table[Column])` 는 `PRODUCTX(Table, Table[Column])` 와 같습니다.

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

예시

다음은 연금 테이블에서 AdjustedRates 열의 곱을 계산합니다.

DAX

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

관련 콘텐츠

[PRODUCTX](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PRODUCTX

아티클 • 2023. 12. 14.

테이블의 각 행에 대해 계산된 식의 곱을 반환합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- 열에 있는 숫자의 곱을 반환하려면 PRODUCT를 사용합니다.
- PRODUCTX 함수는 테이블 또는 테이블을 반환하는 식의 첫 번째 인수로 사용합니다. 두 번째 인수는 제품을 계산하려는 숫자 또는 열로 계산되는 식을 포함하는 열입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 투자의 미래 가치를 계산합니다.

DAX

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

관련 콘텐츠

[제품](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SUM

아티클 • 2023. 12. 14.

열에 있는 모든 수를 더합니다.

구문

DAX

`SUM(<column>)`

매개 변수

[+] 테이블 확장

용어	정의
column	합계를 계산할 숫자가 들어 있는 열입니다.

반환 값

10진수입니다.

설명

합계를 계산하는 값을 필터링하려면 SUMX 함수를 사용하고 합계를 계산할 식을 지정할 수 있습니다.

예시

다음 예에서는 Sales 테이블에서 Amt 열에 포함된 모든 숫자를 추가합니다.

DAX

`= SUM(Sales[Amt])`

관련 콘텐츠

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SUMX

아티클 • 2023. 10. 20.

테이블의 각 행에 대해 계산된 식의 합계를 반환합니다.

구문

DAX

SUMX(<table>, <expression>)

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- SUMX 함수는 테이블 또는 테이블을 반환하는 식의 첫 번째 인수로 사용합니다. 두 번째 인수는 합계를 계산할 숫자가 들어 있는 열 또는 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 수식에서 SUMX의 더 복잡한 예제는 ALL 및 CALCULATETABLE을 참조하세요.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예에서는 먼저 'InternetSales[SalesTerritoryID] = 5` 식에서 InternetSales 테이블을 필터링한 다음 Freight 열에 있는 모든 값의 합계를 반환합니다. 즉, 식은 지정된 판매 영

역에 대해서만 화물 요금의 합계를 반환합니다.

DAX

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

열을 필터링할 필요가 없는 경우 SUM 함수를 사용합니다. SUM 함수는 열을 참조로 사용하는 것을 제외하고 동일한 이름의 Excel 함수와 비슷합니다.

참고 항목

[합계](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

날짜 및 시간 함수

아티클 • 2023. 10. 20.

이러한 함수는 날짜 및 시간을 기반으로 계산을 만드는 데 도움이 됩니다. DAX의 많은 함수는 Excel 날짜 및 시간 함수와 유사합니다. 그러나 DAX 함수는 datetime 데이터 형식을 사용하며 열의 값을 인수로 사용할 수 있습니다.

이 범주에서

함수	설명
일정	연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다.
CALENDARAUTO	연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다.
DATE	지정된 날짜를 datetime 형식으로 반환합니다.
DATEDIFF	두 날짜 간의 간격 경계 수를 반환합니다.
Datevalue	텍스트 형식의 날짜를 datetime 형식의 날짜로 변환합니다.
DAY	월의 날짜(1에서 31까지)를 반환합니다.
Edate	시작 날짜 이전 또는 이후의 지정된 개월 수인 날짜를 반환합니다.
EOMONTH	지정된 월 수 이전 또는 이후의 월 마지막 날의 날짜/시간 형식으로 날짜를 반환합니다.
시간	시간을 0(오전 12:00)에서 23(오후 11:00)으로 반환합니다.
분	날짜 및 시간 값이 지정된 경우 분을 0에서 59까지의 숫자로 반환합니다.
MONTH	월을 1(1월)에서 12월(12월)로 반환합니다.
NETWORKDAYS	두 날짜 사이의 전체 근무일 수를 반환합니다.
Nwo	현재 날짜 및 시간을 datetime 형식으로 반환합니다.
분기	분기를 1에서 4까지의 숫자로 반환합니다.
두 번째	시간 값의 초를 0에서 59까지의 숫자로 반환합니다.
시간	숫자로 지정된 시간, 분 및 초를 날짜/시간 형식의 시간으로 변환합니다.
TIMEVALUE	텍스트 형식의 시간을 날짜/시간 형식의 시간으로 변환합니다.
오늘	현재 날짜를 반환합니다.
UTCNOW	현재 UTC 날짜 및 시간을 반환합니다.

함수	설명
UTCTODAY	현재 UTC 날짜를 반환합니다.
평일	날짜의 요일을 식별하는 1에서 7까지의 숫자를 반환합니다.
WEEKNUM	return_type 값에 따라 지정된 날짜 및 연도의 주 번호를 반환합니다.
YEAR	날짜 연도를 1900-9999 범위의 4자리 정수로 반환합니다.
YEARFRAC	두 날짜 사이의 전체 일 수로 표시되는 연도의 비율을 계산합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CALENDAR

아티클 • 2023. 10. 20.

연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 지정된 시작 날짜부터 지정된 종료 날짜까지이며, 이 두 날짜를 포함합니다.

구문

DAX

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

매개 변수

용어	정의
start_date	datetime 값을 반환하는 DAX 식입니다.
end_date	datetime 값을 반환하는 DAX 식입니다.

반환 값

연속된 날짜 집합을 포함하는 "Date"라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 지정된 시작 날짜부터 지정된 종료 날짜까지이며, 이 두 날짜를 포함합니다.

설명

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

예제

다음 수식은 2015년 1월 1일에서 2021년 12월 31일 사이의 날짜가 포함된 테이블을 반환합니다.

DAX

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

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

DAX

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

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CALENDARAUTO

아티클 • 2023. 10. 20.

연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 모델의 데이터를 기반으로 자동으로 계산됩니다.

구문

DAX

`CALENDARAUTO([fiscal_year_end_month])`

매개 변수

용어	정의
fiscal_year_end_month	1에서 12까지의 정수 값을 반환하는 DAX 식입니다. 생략하면 현재 사용자의 달력 테이블 템플릿에 지정된 값(있는 경우)이 기본값으로 설정됩니다. 그렇지 않으면 기본값은 12입니다.

반환 값

연속된 날짜 집합이 포함된 "Date"라는 단일 열이 있는 테이블을 반환합니다. 날짜 범위는 모델의 데이터를 기반으로 자동으로 계산됩니다.

설명

- 날짜 범위는 다음과 같이 계산됩니다.
 - 계산 열 또는 계산 테이블에 없는 모델의 가장 빠른 날짜는 MinDate로 사용됩니다.
 - 계산 열 또는 계산 테이블에 없는 모델의 최신 날짜는 MaxDate로 사용됩니다.
 - 반환되는 날짜 범위는 MinDate와 연결된 회계 연도의 시작 날짜와 MaxDate와 연결된 회계 연도의 끝 사이의 날짜입니다.
- 모델에 계산 열 또는 계산 테이블에 없는 날짜/시간 값이 없는 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

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

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

`CALENDARAUTO(3)` 는 2010년 4월 1일부터 2012년 3월 31일 사이의 모든 날짜를 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATE

아티클 • 2023. 10. 20.

지정된 날짜를 datetime 형식으로 반환합니다.

구문

DAX

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

매개 변수

용도 정의

어

연도 연도를 나타내는 숫자입니다.

도

연도 인수의 값에는 1~4자리 숫자가 포함될 수 있습니다. 연도 인수는 컴퓨터에서 사용하는 날짜 시스템에 따라 해석됩니다.

1900년 3월 1일부터 날짜가 지원됩니다.

소수 자릿수가 있는 숫자를 입력하면 숫자가 반올림됩니다.

값이 9999보다 크거나 0보다 작으면(음수 값) 함수는 #VALUE! 오류를 반환합니다.

연도 값이 0에서 1899 사이이면 최종 값을 생성하기 위해 값이 1900에 추가됩니다. 아래 예제를 참조하세요. 참고: 원치 않는 결과를 방지하려면 가능하면 연도 인수에 4자리 숫자를 사용해야 합니다. 예를 들어 07을 사용하면 1907이 연도 값으로 반환됩니다.

개월 다음 규칙에 따라 월 또는 계산을 나타내는 숫자입니다.

월

음수 정수는 지원되지 않습니다. 유효한 값은 1-12입니다.

월이 1에서 12까지의 숫자인 경우 해당 연도의 월을 나타냅니다. 1은 1월을 나타내고, 2는 2월을 나타내며, 12일까지는 12월을 나타냅니다.

12보다 큰 정수로 입력하면 다음 계산이 발생합니다. 날짜는 연도에 월 값을 추가하여 계산됩니다. 예를 들어 DATE(2008, 18, 1)가 있는 경우 2009년 6월 1일과 동일한 날짜/시간 값을 반환합니다. 2008년 초에 18개월이 추가되어 2009년 6월의 값이 생성되기 때문입니다. 아래 예제를 참조하세요.

일 다음 규칙에 따라 일 또는 계산을 나타내는 숫자입니다.

용어 정의

음수 정수는 지원되지 않습니다. 유효한 값은 1-31입니다.

날짜가 1에서 지정된 달의 마지막 날까지의 숫자인 경우 해당 월의 일을 나타냅니다.

지정된 월의 마지막 날보다 큰 정수로 입력하면 다음 계산이 발생합니다. 날짜는 일값을 월별로 추가하여 계산됩니다. 예를 들어 수식 `DATE(2008, 3, 32)`에서 DATE 함수는 3월 초에 32일이 추가되어 4월 1일의 값을 생성하므로 2008년 4월 1일에 해당하는 datetime 값을 반환합니다.

일에서 소수 부분을 포함하는 경우 가장 가까운 정수 값으로 반올림됩니다.

반환 값

지정된 날짜(datetime)를 반환합니다.

설명

- DATE 함수는 입력된 정수를 인수로 사용하고 해당 날짜를 생성합니다. DATE 함수는 수식에서 연도, 월 및 요일을 제공하는 경우에 가장 유용합니다. 예를 들어 기본 데이터에는 YYYYMMDD와 같이 날짜로 인식되지 않는 형식의 날짜가 포함될 수 있습니다. DATE 함수를 다른 함수와 함께 사용하여 날짜를 날짜로 인식할 수 있는 숫자로 변환할 수 있습니다.
- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX 날짜 함수는 항상 날짜/시간 데이터 형식을 반환합니다. 그러나 원하는 경우 서식을 사용하여 날짜를 일련 번호로 표시할 수 있습니다.
- 날짜 및 날짜/시간을 형식 `dt"YYYY-MM-DD"` `dt"YYYY-MM-DDThh:mm:ss"`으로 리터럴로 지정할 수도 `dt"YYYY-MM-DD hh:mm:ss"` 있습니다. 리터럴로 지정하는 경우 식에서 DATE 함수를 사용할 필요가 없습니다. 자세한 내용은 [DAX 구문 | 날짜 및 시간입니다](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

단순 날짜

다음 수식은 2009년 7월 8일 날짜를 반환합니다.

DAX

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

년 전 1899

연도 인수에 입력하는 값이 0(0)에서 1899(포함) 사이인 경우 해당 값은 연도를 계산하기 위해 1900에 추가됩니다. 다음 수식은 1908년 1월 2일을 반환합니다($1900+08$)。

DAX

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

년 후 1899

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

DAX

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

개월

월이 12보다 크면 월은 지정된 연도의 첫 번째 달에 해당 월 수를 추가합니다. 다음 수식은 2009년 2월 2일 날짜를 반환합니다.

DAX

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

일

날짜가 지정한 월의 일 수보다 크면 해당 월의 첫째 날에 해당 일 수가 추가됩니다. 다음 수식은 2008년 2월 4일 날짜를 반환합니다.

DAX

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

참고 항목

[날짜 및 시간 함수](#)

[DAY 함수](#)

[TODAY 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATEDIFF

아티클 • 2023. 10. 20.

두 날짜 간의 간격 경계 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
Date1	스칼라 날짜/시간 값입니다.
Date2	스칼라 날짜/시간 값입니다.
간격	날짜를 비교할 때 사용할 간격입니다. 값은 다음 중 하나입니다. <ul style="list-style-type: none">- SECOND- MINUTE- HOUR- DAY- WEEK- MONTH- QUARTER- YEAR

반환 값

두 날짜 간에 간격 경계의 수입니다.

설명

Date2가 Date1보다 크면 양수 결과가 반환됩니다. Date1이 Date2보다 크면 음수 결과가 반환됩니다.

예시

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

다음 DAX 쿼리:

```
DAX

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

다음을 반환합니다.

Value1	Value2
Year	2
분기	9
Month	29
Week	130
요일	914

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Datevalue

아티클 • 2023. 10. 20.

텍스트 형식의 날짜를 datetime 형식의 날짜로 변환합니다.

구문

DAX

`DATEVALUE(date_text)`

매개 변수

용어	정의
date_text	날짜를 나타내는 텍스트입니다.

속성 값/반환 값

날짜/시간 형식의 날짜입니다.

설명

- 변환할 때 DATEVALUE는 모델의 로캘 및 날짜/시간 설정을 사용하여 날짜 값을 결정 합니다. 모델 날짜/시간 설정이 월/일/연도 형식으로 날짜를 나타내는 경우 문자열 "1/8/2009"는 2009년 1월 8일에 해당하는 날짜/시간 값으로 변환됩니다. 그러나 모델 날짜/시간 설정이 날짜/월/연도 형식으로 날짜를 나타내는 경우 동일한 문자열이 2009년 8월 1일에 해당하는 날짜/시간 값으로 변환됩니다.
- 모델의 로캘 및 날짜/시간 설정을 사용한 변환이 실패하면 DATEVALUE는 다른 날짜 형식을 사용하려고 시도합니다. 이 경우 일부 행은 하나의 형식을 사용하여 변환될 수 있고 다른 행은 다른 형식으로 변환됩니다. 예를 들어 "2018년 5월 4일"은 2018년 5월 4일로 변환할 수 있으며 "2018년 20월 4일"은 4월 20일로 변환할 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 데이터/시간 설정은 모델을 만들 때 애플리케이션 및 컴퓨터에 의해 처음에 결정됩니다.
- 날짜 및 날짜/시간을 형식 `dt"YYYY-MM-DD"` `dt"YYYY-MM-DDThh:mm:ss"`으로 리터럴로 지정할 수도 `dt"YYYY-MM-DD hh:mm:ss"` 있습니다. 리터럴로 지정된 경우 식에서 DATEVALUE 함수를 사용할 필요가 없습니다. 자세한 내용은 [DAX 구문 | 날짜 및 시간입니다](#).

예시

다음 예제에서는 모델 로캘 및 날짜 및 시간을 표시하는 방법에 대한 설정에 따라 다른 날짜/시간 값을 반환합니다.

- 날짜/시간 설정에서 날짜/시간 설정에서 이 예에서는 2009년 1월 8일에 해당하는 날짜/시간 값을 반환합니다.
- 날짜/시간 설정에서 월이 날짜보다 이전인 경우 이 예제에서는 2009년 8월 1일에 해당하는 datetime 값을 반환합니다.

DAX

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

참고 항목

[날짜 및 시간 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAY

아티클 • 2023. 12. 14.

월의 날짜(1에서 31까지)를 반환합니다.

구문

DAX

DAY(<date>)

매개 변수

[+] 테이블 확장

용어	정의
날짜	날짜/시간 형식의 날짜 또는 날짜의 텍스트 표현입니다.

반환 값

월의 일을 나타내는 정수입니다.

설명

- DAY 함수는 찾으려는 날짜의 인수로 사용됩니다. 날짜를 반환하는 식을 사용하거나 날짜/시간 형식으로 날짜를 입력하여 다른 날짜 함수를 사용하여 함수에 날짜를 제공할 수 있습니다. 날짜에 허용되는 문자열 형식 중 하나로 날짜를 입력할 수도 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공된 날짜 값의 표시 형식에 관계 없이 그레고리오 값이 됩니다. 예를 들어 제공된 날짜의 표시 형식이 Hijri인 경우 YEAR, MONTH 및 DAY 함수에 대해 반환되는 값은 해당하는 그레고리오 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 형식으로 날짜를 나타내는 경우 문자열 "1/8/2009"는 2009년 1월 8일에 해당하는 날짜/시간 값으로 해석되고 함수는 8을 반환합니다. 그

러나 현재 날짜/시간 설정이 날짜/월/연도 형식으로 날짜를 나타내는 경우 동일한 문자열은 2009년 8월 1일에 해당하는 날짜/시간 값으로 해석되고 함수는 1을 반환합니다.

예: 날짜 열에서 날짜 가져오기

다음 수식은 [Birthdate] 열의 날짜에서 날짜를 반환합니다.

DAX

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

예제 - 문자열 날짜에서 날짜 가져오기

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

DAX

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

예제 - 일 값을 조건으로 사용

다음 식은 각 판매 주문이 이루어진 날을 반환하고 주문이 월 10일에 배치된 경우 행을 홍보 판매 항목으로 플래그를 지정합니다.

DAX

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

관련 콘텐츠

[날짜 및 시간 함수](#)

[TODAY 함수](#)

[DATE 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Edate

아티클 • 2023. 10. 20.

시작 날짜 이전 또는 이후의 지정된 개월 수인 날짜를 반환합니다. EDATE를 사용하여 발행일과 같은 날짜에 해당하는 만기일 또는 기한을 계산합니다.

구문

DAX

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

매개 변수

용어	정의
start_date	시작 날짜를 나타내는 날짜/시간 또는 텍스트 형식의 날짜입니다.
월	start_date 이전 또는 이후의 개월 수를 나타내는 정수입니다.

반환 값

날짜(datetime)입니다.

설명

- 날짜를 순차적인 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜/시간 형식으로 작동합니다. 다른 형식으로 저장된 날짜는 암시적으로 변환됩니다.
- start_date **유효한 날짜가 아니면** EDATE에서 오류를 반환합니다. 첫 번째 인수로 제공하는 열 참조 또는 날짜가 날짜인지 확인합니다.
- 월**이 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 월/일/연도 형식의 날짜를 나타내는 경우 다음 문자열 "1/8/2009"는 2009년 1월 8일에 해당하는 날짜/시간 값으로 해석됩니다. 그러나 현재 날짜 시간 설정이 일/월/연도 형식의 날짜를 나타내는 경우 동일한 문자열은 2009년 8월 1일에 해당하는 날짜/시간 값으로 해석됩니다.

- 요청된 날짜가 해당 월의 마지막 날을 지난 경우 해당 월의 마지막 날짜가 반환됩니다. 예를 들어 EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) 반환 2009년 2월 28일; 시작 날짜 이후 1개월에 해당하는 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 [TransactionDate] 열에 저장된 주문 날짜 3개월 후의 날짜를 반환하는 예제입니다.

DAX

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

참고 항목

[EOMONTH 함수](#)

[날짜 및 시간 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

EOMONTH

아티클 • 2023. 10. 20.

지정된 월 수 이전 또는 이후의 월 마지막 날의 날짜/시간 형식으로 날짜를 반환합니다. EOMONTH를 사용하여 월의 마지막 날에 해당하는 만기일 또는 기한을 계산합니다.

구문

DAX

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

매개 변수

용어	정의
start_date	날짜/시간 형식의 시작 날짜 또는 날짜의 허용된 텍스트 표현입니다.
월	start_date 이전 또는 이후의 개월 수를 나타내는 숫자입니다. 참고: 정수가 아닌 숫자를 입력하면 숫자가 가장 가까운 정수로 반올림되거나 아래로 반올림됩니다.

반환 값

날짜(datetime)입니다.

설명

- 날짜를 순차적인 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜/시간 형식으로 작동합니다. EOMONTH 함수는 다음과 같은 제한 사항을 사용하여 다른 형식으로 날짜를 수락할 수 있습니다.
- start_date 유효한 날짜가 아니면 EOMONTH에서 오류를 반환합니다.
- start_date 날짜/시간 형식이 아닌 숫자 값인 경우 EOMONTH는 숫자를 날짜로 변환합니다. 예기치 않은 결과를 방지하려면 EOMONTH 함수를 사용하기 전에 숫자를 날짜/시간 형식으로 변환합니다.
- start_date 월이 잘못된 날짜를 생성하는 경우 EOMONTH는 오류를 반환합니다. 1900년 3월 1일 이전과 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 2009년 1월 8일에 해당하는 날짜/시간 값으로 해석됩니다. 그러나 현재 날짜 시간 설정이 일/월/연도 형식의 날짜를 나타내는 경우 동일한 문자열은 2009년 8월 1일에 해당하는 날짜/시간 값으로 해석됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 식은 월 인수가 2로 반올림되므로 2008 년 5월 31일 을 반환합니다.

DAX

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

참고 항목

[EDATE 함수](#)

[날짜 및 시간 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

시간

아티클 • 2023. 12. 14.

시간을 0(오전 12:00)에서 23(오후 11:00)으로 반환합니다.

구문

DAX

`HOUR(<datetime>)`

매개 변수

 테이블 확장

용어	정의
날짜/시간	날짜/시간 값(예: 16:48:00 또는 오후 4:48)입니다.

반환 값

0에서 23까지의 정수입니다.

설명

- HOUR 함수는 찾으려는 시간이 포함된 시간을 인수로 사용합니다. 날짜/시간 함수, datetime을 반환하는 식을 사용하거나 허용되는 시간 형식 중 하나로 값을 직접 입력하여 시간을 제공할 수 있습니다. 시간은 시간의 허용된 텍스트 표현으로 입력할 수도 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 로캘은 콜론(:) 시간 구분 기호로 사용하고 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 이해 합니다.

예 1

다음 예제에서는 Orders라는 테이블의 TransactionTime 열에서 시간을 반환합니다.

DAX

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

예제 2

다음 예제에서는 15를 반환합니다. 즉, 24시간 시계에서 오후 3시에 해당하는 시간을 의미합니다. 텍스트 값은 자동으로 구문 분석되고 날짜/시간 값으로 변환됩니다.

DAX

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

관련 콘텐츠

[날짜 및 시간 함수](#)

[MINUTE 함수](#)

[YEAR 함수](#)

[SECOND 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

분

아티클 • 2023. 10. 20.

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

구문

DAX

MINUTE(<datetime>)

매개 변수

용어	정의
날짜/시간	16:48:00 또는 4:48 PM과 같이 허용되는 시간 형식의 날짜/시간 값 또는 텍스트입니다.

반환 값

0에서 59까지의 정수입니다.

설명

- 날짜와 시간을 일련 숫자 형식으로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간에 날짜/시간 데이터 형식을 사용합니다. 날짜 및 시간을 저장하는 열을 참조하거나, 날짜/시간 함수를 사용하거나, 날짜와 시간을 반환하는 식을 사용하여 MINUTE 함수에 datetime 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 로캘은 콜론(:) 시간 구분 기호로 사용하고 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 올바르게 구문 분석됩니다. 로캘 설정을 확인하여 결과를 이해합니다.

예 1

다음 예제에서는 Orders 테이블의 TransactionTime 열에 저장된 값에서 분을 반환합니다.

DAX

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

예제 2

다음 예제에서는 오후 1시 45분에 있는 시간(분)인 45를 반환합니다.

DAX

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

참고 항목

[날짜 및 시간 함수](#)

[HOUR 함수](#)

[YEAR 함수](#)

[SECOND 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MONTH

아티클 • 2023. 12. 14.

월을 1(1월)에서 12월(12월)로 반환합니다.

구문

DAX

MONTH(<datetime>)

매개 변수

 테이블 확장

용어	정의
날짜	날짜/시간 또는 텍스트 형식의 날짜입니다.

반환 값

1에서 12까지의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 작업 시 날짜/시간 형식을 사용합니다. 허용되는 **datetime** 형식을 입력하거나, 날짜가 포함된 열에 대한 참조를 제공하거나, 날짜를 반환하는 식을 사용하여 MONTH 함수에 인수로 사용되는 날짜를 입력할 수 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공된 날짜 값의 표시 형식에 관계 없이 그레고리오 값이 됩니다. 예를 들어 제공된 날짜의 표시 형식이 Hijri인 경우 YEAR, MONTH 및 DAY 함수에 대해 반환되는 값은 해당하는 그레고리오 날짜와 연결된 값입니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 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 2009년 1월 8일에 해당하는 날짜/시간 값으로 해석되고 함수는 1의 결과를 생성합니다.

그러나 현재 날짜 시간 설정이 일/월/연도 형식의 날짜를 나타내는 경우 동일한 문자열은 2009년 8월 1일에 해당하는 날짜/시간 값으로 해석되고 함수는 8의 결과를 생성합니다.

- 날짜의 텍스트 표현을 datetime 값으로 올바르게 변환할 수 없는 경우 함수는 오류를 반환합니다.

예 1

다음 식은 날짜 인수의 월인 3월에 해당하는 정수인 3을 반환합니다.

DAX

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

예제 2

다음 식은 Orders 테이블의 TransactionDate 열에 있는 날짜로부터 월을 반환합니다.

DAX

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

관련 콘텐츠

[날짜 및 시간 함수](#)

[HOUR 함수](#)

[MINUTE 함수](#)

[YEAR 함수](#)

[SECOND 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NETWORKDAYS

아티클 • 2023. 10. 20.

두 날짜 사이의 전체 근무일 수(포함)를 반환합니다. 매개 변수는 주말 요일 및 일 수를 지정합니다. 주말 및 휴일로 지정된 날은 근무일로 간주되지 않습니다.

구문

DAX

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

매개 변수

용어	정의
start_date	시작 날짜를 나타내는 날짜입니다. 차이를 계산할 날짜입니다. start_date는 end_date 보다 빠르거나 같거나 늦을 수 있습니다.
end_date	종료 날짜를 나타내는 날짜입니다. 차이를 계산할 날짜입니다. start_date는 end_date 보다 빠르거나 같거나 늦을 수 있습니다.
weekend	주말이라 start_date와 end_date 사이의 전체 근무일 수에 포함되지 않는 요일을 나타냅니다. Weekend는 주말이 발생하는 시기를 지정하는 주말 번호입니다. Weekend 번호 값은 다음 주말 요일을 나타냅니다. 1 또는 생략: 토요일, 일요일 2: 일요일, 월요일 3: 월요일, 화요일 4: 화요일, 수요일 5: 수요일, 목요일, 6: 목요일, 금요일 7: 금요일, 토요일 11: 일요일만 12: 월요일만 13: 화요일만 14: 수요일만 15: 목요일만 16: 금요일만 17: 토요일만
휴일	근무일 달력에서 제외할 하나 이상의 날짜로 구성된 열 테이블입니다.

Return Value

전체 근무일의 정수 값입니다.

설명

- 이 DAX 함수는 NETWORKDAYS.INTL 및 NETWORKDAYS 함수와 비슷합니다.
- start_date와 end_date 모두 BLANK이면 출력 값도 BLANK입니다.
- start_date 또는 end_date가 BLANK이면 BLANK인 start_date 또는 end_date가 Date(1899, 12, 30)로 취급됩니다.
- 날짜는 DATE 함수를 사용하거나 다른 식의 결과로 지정해야 합니다. 예를 들어 DATE (2022, 5, 30)는 2022년 5월 30일을 지정합니다. 날짜는 리터럴((dt”2022-05-30”) 형식)으로 지정할 수도 있습니다. 날짜를 텍스트로 지정하지 마세요.

예시

다음 식은 다음과 같습니다.

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

HRESULT = NO_ERROR를

[값]
0

이 예제에서는 시작 날짜가 토요일이고 종료 날짜가 월요일이므로 0이 반환됩니다. Weekend 매개 변수는 주말을 토요일 및 일요일로 지정하므로 이들 요일은 근무일이 아닙니다. Holiday 매개 변수는 30일(종료 날짜)을 휴일로 표시하므로 근무일이 남아 있지 않습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

현재

아티클 • 2023. 12. 14.

현재 날짜 및 시간을 `datetime` 형식으로 반환합니다.

NOW 함수는 워크시트에 현재 날짜와 시간을 표시하거나 현재 날짜와 시간을 기준으로 값을 계산하고 워크시트를 열 때마다 해당 값을 업데이트해야 하는 경우에 유용합니다.

구문

DAX

`NOW()`

반환 값

날짜(`datetime`)입니다.

설명

- NOW 함수의 결과는 수식이 포함된 열을 새로 고칠 때만 변경됩니다. 지속적으로 업데이트되지 않습니다.
- Power BI 서비스에서 NOW 함수의 결과는 항상 UTC 표준 시간대에 있습니다.
- TODAY 함수는 동일한 날짜를 반환하지만 시간과 관련하여 정확하지는 않습니다. 반환되는 시간은 항상 오전 12:00:00이며 날짜만 업데이트됩니다.

예시

다음 예제에서는 현재 날짜와 시간 및 3.5일을 더한 값을 반환합니다.

DAX

= `NOW()`+3.5

관련 콘텐츠

UTCNOW 함수

TODAY 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

QUARTER

아티클 • 2023. 12. 14.

분기를 1(1월~3월)에서 4(10월~ 12월)로 반환합니다.

구문

DAX

QUARTER(<date>)

매개 변수

[+] 테이블 확장

용어	정의
날짜	날짜입니다.

반환 값

1에서 4까지의 정수입니다.

설명

입력 값이 BLANK이면 출력 값도 BLANK입니다.

예 1

다음 DAX 쿼리:

DAX

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

HRESULT = NO_ERROR를

[+] 테이블 확장

[값]
1
4

예제 2

다음 DAX 쿼리:

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

HRESULT = NO_ERROR를

 테이블 확장

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

관련 콘텐츠

[YEAR](#)

[MONTH](#)

[DAY](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SECOND

아티클 • 2023. 10. 20.

시간 값의 초를 0에서 59까지의 숫자로 반환합니다.

구문

DAX

`SECOND(<time>)`

매개 변수

용어 정의

time 날짜/시간 형식(예: 오후 16:48:23 또는 오후 4:48:47)입니다.

반환 값

0에서 59까지의 정수입니다.

설명

- 날짜와 시간을 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간을 사용할 때 날짜/시간 형식을 사용합니다. 원본 데이터가 이 형식이 아니면 DAX는 데이터를 암시적으로 변환합니다. 서식을 사용하여 날짜와 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"). 다른 식의 결과로 시간 값을 제공하거나 시간이 포함된 열에 대한 참조로 제공할 수도 있습니다.
- 13.60과 같은 다른 데이터 형식의 숫자 값을 제공하는 경우 같은 일련 번호로 해석되고 초 동안 값을 추출하기 전에 날짜/시간 데이터 형식으로 표시됩니다. 결과를 더 쉽게 이해할 수 있도록 SECOND 함수에서 사용하기 전에 날짜와 같은 숫자를 나타낼 수 있습니다. 예를 들어 숫자 값(예: 25.56)이 포함된 열에서 SECOND를 사용하는 경우 수식은 24를 반환합니다. 즉, 날짜로 서식이 지정된 경우 값 25.56은 1900년 1월 25일 오후 1:26:24와 동일하기 때문입니다.

- 시간 인수가 날짜 및 시간의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 대부분의 로캘은 콜론(:) 시간 구분 기호로 사용하고 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 이해합니다.

예 1

다음 수식은 Orders라는 테이블의 TransactionTime 열에 포함된 시간(초)을 반환합니다.

DAX

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

예제 2

다음 수식은 2008년 3월 3일 12:00:03 값으로 표시되는 시간(초)인 3을 반환합니다.

DAX

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

참고 항목

[날짜 및 시간 함수](#)

[시간](#)

[분](#)

[YEAR](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TIME

아티클 • 2023. 10. 20.

숫자로 지정된 시간, 분 및 초를 날짜/시간 형식의 시간으로 변환합니다.

구문

DAX

`TIME(hour, minute, second)`

매개 변수

용어	정의
시간	가져오기 모드: 시간을 나타내는 0에서 32767까지의 숫자입니다. 23보다 큰 값은 24로 나뉘고 remainder는 하루의 분수로 표시되는 시간 값으로 처리됩니다. 예를 들어 TIME(27,0,0) = TIME(3,0,0) = 오전 3:00:00 DirectQuery 모드: 시간을 나타내는 0에서 23까지의 숫자입니다.
분	가져오기 모드: 분을 나타내는 0에서 32767까지의 숫자입니다. 59분을 초과하는 모든 값은 시간 및 분으로 변환됩니다. 1440(24시간)을 초과하는 값은 날짜 부분을 변경하지 않습니다. 대신 1440으로 나뉘고 remainder는 하루의 분수로 표시되는 분 값으로 처리됩니다. 예를 들어 TIME(0,2190,0) = TIME(0,750,0) = TIME(12,30,0) = 오후 12:30:00 DirectQuery 모드: 분을 나타내는 0에서 59까지의 숫자입니다.
second	가져오기 모드: 두 번째를 나타내는 0에서 32767까지의 숫자입니다. 59보다 큰 값은 시간, 분 및 초로 변환됩니다. 예를 들어 TIME(0,0,2000) = TIME(0,33,20) = 오전 12:33:20 DirectQuery 모드: 초를 나타내는 0에서 59까지의 숫자입니다.

반환 값

00:00:00(오전 12:00:00)부터 23:59:59(오후 11:59:59)까지의 시간(날짜/시간)입니다.

설명

- 날짜와 시간을 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간 값 을 날짜/시간 형식으로 사용합니다. DAX 함수에서 날짜/시간 값을 사용하면 다른 형식의 숫자가 암시적으로 변환됩니다. 일련 번호를 사용해야 하는 경우 서식을 사용하여 숫자가 표시되는 방식을 변경할 수 있습니다.
- 시간 값은 날짜 값의 일부이며 일련 번호 시스템에서 10진수로 표시됩니다. 따라서 날짜/시간 값 12:00 PM은 하루의 절반이므로 0.5와 같습니다.
- TIME 함수에 인수를 다른 식의 결과로 직접 입력하는 값으로 제공하거나 숫자 값이 포함된 열에 대한 참조를 통해 제공할 수 있습니다.
- 날짜 및 날짜/시간을 형식 `dt"YYYY-MM-DD"` `dt"YYYY-MM-DDThh:mm:ss"` 으로 리터럴로 지정할 수도 `dt"YYYY-MM-DD hh:mm:ss"` 있습니다. 리터럴로 지정된 경우 식에서 TIME 함수를 사용할 필요가 없습니다. 자세한 내용은 [DAX 구문 | 날짜 및 시간입니다](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제는 모두 오전 3:00 시간을 반환합니다.

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

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

예제 2

다음 예제는 모두 오후 12시 30분 시간을 반환합니다.

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

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

예 3

다음 예제에서는 열의 값에 따라 시간을 만듭니다. `intHours intMinutes intSeconds`

DAX

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

참고 항목

[DATE](#)

날짜 및 시간 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TIMEVALUE

아티클 • 2023. 12. 14.

텍스트 형식의 시간을 날짜/시간 형식의 시간으로 변환합니다.

구문

DAX

`TIMEVALUE(time_text)`

매개 변수

 테이블 확장

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

반환 값

날짜(datetime)입니다.

설명

- 시간 값은 날짜 값의 일부이며 소수로 표시됩니다. 예를 들어 12:00 PM은 하루의 절반이므로 0.5로 표시됩니다.
- time_text 인수가 날짜 및 시간의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 모델의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 대부분의 로캘은 콜론(:) 시간 구분 기호로 사용하며 콜론을 시간 구분 기호로 사용하는 입력 텍스트는 올바르게 구문 분석됩니다. 로캘 설정을 검토하여 결과를 이해합니다.
- 날짜 및 날짜/시간을 형식 `dt"YYYY-MM-DD"` `dt"YYYY-MM-DDThh:mm:ss"` 으로 리터럴로 지정할 수도 `dt"YYYY-MM-DD hh:mm:ss"` 있습니다. 리터럴로 지정하면 식에서 TIMEVALUE 함수를 사용할 필요가 없습니다. 자세한 내용은 [DAX 구문 | 날짜 및 시간입니다.](#)

예시

DAX

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

관련 콘텐츠

[날짜 및 시간 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

오늘

아티클 • 2023. 12. 14.

현재 날짜를 반환합니다.

구문

DAX

`TODAY()`

반환 값

날짜(datetime)입니다.

설명

- TODAY 함수는 통합 문서를 여는 시기에 관계없이 현재 날짜를 워크시트에 표시해야 하는 경우에 유용합니다. 간격을 계산하는 데에도 유용합니다.
- TODAY 함수가 예상 날짜를 업데이트하지 않는 경우 열 또는 통합 문서가 새로 고쳐지는 시기를 제어하는 설정을 변경해야 할 수 있습니다.
- NOW 함수는 비슷하지만 정확한 시간을 반환하는 반면 TODAY는 모든 날짜의 시간 값 12:00:00 PM을 반환합니다.

예시

누군가가 1963년에 태어났다는 것을 알고 있다면, 다음 공식을 사용하여 올해 생일 기준으로 그 사람의 나이를 찾을 수 있습니다.

DAX

`= YEAR(TODAY())-1963`

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

관련 콘텐츠

날짜 및 시간 함수

Nwo

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UTCNOW

아티클 • 2023. 12. 14.

현재 UTC 날짜 및 시간을 반환합니다.

구문

DAX

```
UTCNOW()
```

반환 값

A(datetime).

설명

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

예시

다음을 수행합니다.

DAX

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

HRESULT = NO_ERROR를

 테이블 확장

[값]

2018년 2월 2일 오전 4:48:08

관련 콘텐츠

NOW 함수

UTCTODAY 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UTCTODAY

아티클 • 2023. 12. 14.

현재 UTC 날짜를 반환합니다.

구문

DAX

```
UTCTODAY()
```

반환 값

날짜입니다.

설명

- UTCTODAY는 모든 날짜에 대해 오후 12:00:00 시간 값을 반환합니다.
- UTCNOW 함수는 비슷하지만 정확한 시간과 날짜를 반환합니다.

예시

다음을 수행합니다.

DAX

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

HRESULT = NO_ERROR를

 테이블 확장

[값]

2/2/2018

관련 콘텐츠

NOW 함수

UTCNOW 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

WEEKDAY

아티클 • 2023. 10. 20.

날짜의 요일을 식별하는 1에서 7까지의 숫자를 반환합니다. 기본적으로 일 범위는 1(일요일)에서 7(토요일)까지입니다.

구문

DAX

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

매개 변수

용어	정의
날짜	날짜/시간 형식의 날짜입니다. 날짜는 DATE 함수를 사용하거나, 날짜를 생성하는 식을 사용하거나, 다른 수식의 결과로 입력해야 합니다.
return_type	반환 값을 결정하는 숫자입니다. 반환 유형: 1, 주가 일요일(1)에 시작되고 토요일(7)에 끝납니다. 번호가 1부터 7까지입니다. 반환 유형: 2, 주가 월요일(1)에 시작되어 일요일(7)에 끝납니다. 반환 유형: 3, 주 월요일 (0)에 시작하고 일요일 (6).numbered 1 ~ 7에 종료 됩니다.

반환 값

1에서 7까지의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜 및 시간을 날짜/시간 형식으로 사용합니다. 날짜를 일련 번호로 표시해야 하는 경우 Excel에서 서식 옵션을 사용할 수 있습니다.

- 날짜의 허용된 텍스트 표현으로 날짜를 입력할 수도 있지만 예기치 않은 결과를 방지하려면 먼저 텍스트 날짜를 날짜/시간 **형식으로** 변환하는 것이 가장 좋습니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜/시간 설정을 사용하여 텍스트 값을 이해합니다. 현재 날짜/시간 설정이 월/일/연도 형식으로 날짜를 나타내는 경우 문자열 "1/8/2009"는 2009년 1월 8일에 해당하는 날짜/시간 **값으로** 해석됩니다. 그러나 현재 날짜/시간 설정이 날짜/월/연도 형식으로 날짜를 나타내는 경우 동일한 문자열은 2009년 8월 1일에 해당하는 날짜/시간 **값으로** 해석됩니다.

예시

다음 예제에서는 [HireDate] 열에서 날짜를 가져오고, 1을 추가하고, 해당 날짜에 해당하는 평일을 표시합니다. **return_type** 인수가 생략되었으므로 기본 형식이 사용되며, 1은 일요일이고 7은 토요일입니다. 결과가 4이면 그 날은 수요일입니다.

DAX

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

참고 항목

[날짜 및 시간 함수](#)

[WEEKNUM 함수](#)

[YEARFRAC 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

WEEKNUM

아티클 • 2023. 10. 20.

return_type 값에 따라 지정된 날짜의 주 번호를 반환합니다. 주 번호는 주가 1년 이내에 숫자로 떨어지는 위치를 나타냅니다.

이 함수에는 다음 두 가지 시스템이 사용됩니다.

- **시스템 1** - 1월 1일을 포함하는 주는 해당 연도의 첫 번째 주이며 1주차로 지정됩니다.
- **시스템 2** - 한 해의 첫 번째 목요일을 포함하는 주는 해당 연도의 첫 번째 주이며 1주 차로 지정됩니다. 이 시스템은 일반적으로 유럽 주 번호 매기기 시스템으로 알려진 ISO 8601에 지정된 방법론입니다.

구문

DAX

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

매개 변수

용어	정의
날짜	날짜/시간 형식의 날짜입니다.
return_type	(선택 사항) 주가 시작되는 요일을 결정하는 숫자입니다. 기본값은 1입니다. 설명 부분을 참조하세요.

반환 값

정수입니다.

설명

- 기본적으로 WEEKNUM 함수는 1월 1일이 포함된 주가 해당 연도의 첫 번째 주로 간주되는 달력 규칙을 사용합니다. 그러나 유럽에서 널리 사용되는 ISO 8601 달력 표준은 첫 주를 새해에 가장 많은 일(4일 이상)이 떨어지는 것으로 정의합니다. 즉,

return_type 21 이외의 유효한 값인 경우 1월 첫째 주에 3일 이하가 있는 연도에 대해 WEEKNUM 함수는 ISO 8601 정의와 다른 주 번호를 반환합니다.

- 21을 제외한 **return_type**의 경우 일부 DirectQuery 데이터 원본에서 다음 유효 값을 지원하지 않을 수 있습니다.

return_type	주 시작	System
1 또는 생략됨	일요일	1
2	월요일	1
11	월요일	1
12	화요일	1
13	수요일	1
14	목요일	1
15	금요일	1
16	토요일	1
17	일요일	1
21	월요일	2

예 1

다음 예제에서는 2010년 2월 14일의 주 번호를 반환합니다. 이 계산에서는 월요일에 몇 주가 시작된다고 가정합니다.

DAX

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

예제 2

다음 예제에서는 Employees 테이블에서 HireDate 열에 저장된 날짜의 주 번호를 반환합니다. 이 계산에서는 일요일에 몇 주가 시작된다고 가정합니다.

DAX

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

참고 항목

[YEARFRAC 함수](#)

[WEEKDAY 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

YEAR

아티클 • 2023. 12. 14.

날짜 연도를 1900-9999 범위의 4자리 정수로 반환합니다.

구문

DAX

YEAR(<date>)

매개 변수

데이터 확장

용어 정의

날짜 찾을 연도를 포함하는 날짜/시간 또는 텍스트 형식의 날짜입니다.

반환 값

1900-9999 범위의 정수입니다.

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜/시간 데이터 형식을 사용하여 날짜 및 시간을 사용합니다.
- 날짜는 DATE 함수를 사용하거나 다른 수식 또는 함수의 결과로 입력해야 합니다. 날짜의 허용된 텍스트 표현(예: 2007년 3월 3일 또는 2003년 3월 3일)에 날짜를 입력할 수도 있습니다.
- YEAR, MONTH 및 DAY 함수에서 반환되는 값은 제공된 날짜 값의 표시 형식에 관계 없이 그레고리오 값이 됩니다. 예를 들어 제공된 날짜의 표시 형식이 Hijri 달력을 사용하는 경우 YEAR, MONTH 및 DAY 함수에 대해 반환되는 값은 해당하는 그레고리오 날짜와 연결된 값이 됩니다.
- 날짜 인수가 날짜의 텍스트 표현인 경우 함수는 변환을 수행하기 위해 클라이언트 컴퓨터의 로캘 및 날짜 시간 설정을 사용하여 텍스트 값을 이해합니다. 문자열 형식

이 현재 로캘 설정과 호환되지 않는 경우 오류가 발생할 수 있습니다. 예를 들어 로캘에서 날짜를 월/일/연도로 정의하고 날짜가 일/월/연도로 제공되면 2009년 1월 25일로 해석되지 않고 잘못된 날짜로 해석됩니다.

예시

다음 예제에서는 2007을 반환합니다.

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

예제 - 식의 결과로 날짜 지정

설명

다음 예제에서는 오늘 날짜의 연도를 반환합니다.

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

관련 콘텐츠

[날짜 및 시간 함수](#)

[HOUR 함수](#)

[MINUTE 함수](#)

[YEAR 함수](#)

[SECOND 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

YEARFRAC

아티클 • 2023. 12. 14.

두 날짜 사이의 전체 일 수로 표시되는 연도의 비율을 계산합니다. YEARFRAC 워크시트 함수를 사용하여 특정 용어에 할당할 연간 혜택 또는 의무의 비율을 식별합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
start_date	날짜/시간 형식의 시작 날짜입니다.
end_date	날짜/시간 형식의 종료 날짜입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 모든 인수는 정수로 잘립니다.
	Basis - Description
	0 - 미국(NASD) 30/360(기본값)
	1 - 실제/실제
	2 - 실제/360
	3 - 실제/365
	4 - 유럽 30/360

반환 값

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

설명

- 날짜를 일련 번호로 저장하는 Microsoft Excel과 달리 DAX는 날짜/시간 형식을 사용하여 날짜 및 시간을 사용합니다. 날짜를 일련 번호로 표시해야 하는 경우 Excel에서 서식 옵션을 사용할 수 있습니다.
- `start_date` 또는 `end_date` 유효한 날짜가 아니면 YEARFRAC에서 오류를 반환합니다.
- `basis < 0` 또는 `basis > 4`이면 YEARFRAC에서 오류를 반환합니다.

예 1

다음 예제에서는 두 열의 날짜 간 차이로 표시되는 연도의 비율을 반환하고 `ShippingDate` 다음 `TransactionDate` 을 수행합니다.

DAX

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

예제 2

다음 예제에서는 1월 1일과 3월 1일 날짜 간의 차이로 표시되는 연도의 비율을 반환합니다.

DAX

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

예기치 않은 결과를 가져오지 않으려면 가능하면 4자리 연도를 사용합니다. 연도가 잘리면 현재 연도가 가정됩니다. 날짜가 생략되거나 생략되면 해당 월의 첫 번째 날짜가 가정됩니다.

두 번째 인수인 `basis`도 생략되었습니다. 따라서 연도 분수는 미국(NASD) 30/360 표준에 따라 계산됩니다.

관련 콘텐츠

[날짜 및 시간 함수](#)

[WEEKNUM 함수](#)

YEARFRAC 함수

WEEKDAY 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Filter 함수

아티클 • 2024. 02. 22.

DAX의 필터 및 값 함수는 가장 복잡하고 강력하며 Excel 함수와 크게 다릅니다. 조회 함수는 데이터베이스와 같은 테이블 및 관계를 사용하여 작동합니다. 필터링 함수를 사용하면 데이터 컨텍스트를 조작하여 동적 계산을 만들 수 있습니다.

이 범주에서

 테이블 확장

함수	설명
ALL	적용되었을 수 있는 필터를 무시하고 테이블의 모든 행 또는 열의 모든 값을 반환합니다.
ALLCROSSFILTERED	테이블에 적용되는 모든 필터를 지웁니다.
ALLEXCEPT	지정한 열에 적용된 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다.
ALLNOBLANKROW	관계의 부모 테이블에서 빈 행을 제외한 모든 행 또는 빈 행을 제외한 열의 모든 고유 값을 반환하고 존재할 수 있는 컨텍스트 필터를 무시합니다.
ALLSELECTED	다른 모든 컨텍스트 필터 또는 명시적 필터를 유지하면서 현재 쿼리의 열 및 행에서 컨텍스트 필터를 제거합니다.
계산	수정된 필터 컨텍스트에서 식을 평가합니다.
CALCULATETABLE	수정된 필터 컨텍스트에서 테이블 식을 평가합니다.
이전	멘션 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.
초기	지정된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.
필터	다른 테이블 또는 식의 하위 집합을 나타내는 테이블을 반환합니다.
첫 번째	시각적 계산에만 사용됩니다. 축의 첫 번째 행에서 시각적 행렬의 값을 검색합니다.
INDEX	지정된 파티션 내에서 위치 매개 변수로 지정된 절대 위치 또는 지정된 축에 따라 정렬된 행을 반환합니다.
KEEPFILTERS	CALCULATE 또는 CALCULATETABLE 함수를 평가하는 동안 필터가 적용되는 방식을 수정합니다.
LAST	시각적 계산에만 사용됩니다. 축의 마지막 행에서 시각적 행렬의 값을 검색합니다.

함수	설명
	합니다.
LOOKUPVALUE	검색 조건에 지정된 모든 조건을 충족하는 행의 값을 반환합니다. 함수는 하나 이상의 검색 조건을 적용할 수 있습니다.
MATCHBY	창 함수에서 데이터를 일치시키고 현재 행을 식별하는 방법을 결정하는 데 사용되는 열을 정의합니다.
MOVINGAVERAGE	시각적 계산 데이터 표의 지정된 축을 따라 계산된 이동 평균을 반환합니다.
다음	시각적 계산에만 사용됩니다. 시각적 행렬에서 축의 다음 행에 있는 값을 검색합니다.
OFFSET	지정된 오프셋을 통해 동일한 테이블 내의 현재 행 앞이나 뒤 위치에 있는 단일 행을 반환합니다.
Orderby	각 창 함수의 파티션 내에서 정렬 순서를 결정하는 열을 정의합니다.
PARTITIONBY	창 함수 <의 관계> 매개 변수를 분할하는 데 사용되는 열을 정의합니다.
이전	시각적 계산에만 사용됩니다. 시각적 행렬에서 축의 이전 행에 있는 값을 검색합니다.
범위	현재 행을 기준으로 지정된 축 내의 행 간격을 반환합니다. WINDOW의 바로 가기입니다.
RANK	지정된 간격 내에 있는 행의 순위를 반환합니다.
REMOVEFILTERS	지정된 테이블 또는 열에서 필터를 지웁니다.
ROWNUMBER	지정된 간격 내에 있는 행의 고유 순위를 반환합니다.
Runningsum	시각적 계산 데이터 표의 지정된 축을 따라 계산된 실행 합계를 반환합니다.
SELECTEDVALUE	columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 값을 반환합니다. 그렇지 않으면 alternateResult를 반환합니다.
WINDOW	지정된 간격 내에 배치되는 여러 행을 반환합니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

ALL

아티클 • 2023. 10. 20.

적용되었을 수 있는 필터를 무시하고 테이블의 모든 행 또는 열의 모든 값을 반환합니다. 이 함수는 필터를 지우고 테이블의 모든 행에서 계산을 만드는 데 유용합니다.

구문

DAX

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

매개 변수

용어	정의
table	필터를 지울 테이블입니다.
column	필터를 지울 열입니다.

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

반환 값

필터가 제거된 테이블 또는 열입니다.

설명

- 이 함수는 그 자체로 사용되지 않지만 다른 계산이 수행되는 결과 집합을 변경하는데 사용할 수 있는 중간 함수 역할을 합니다.
- ALL() 함수를 포함하는 DAX 식의 일반적인 동작은 적용된 모든 필터가 무시된다는 것입니다. 그러나 특정 DAX 쿼리에 필요한 처리량을 줄이기 위해 필터링을 최적화하는 DAX 기술인 자동 존재로 인해 그렇지 않은 몇 가지 시나리오가 있습니다. 자동 존재 및 ALL()이 예기치 않은 결과를 제공하는 예제는 같은 테이블의 두 개 이상의 열(예: 슬라이서 사용)을 필터링할 때와 같은 테이블에 ALL()을 사용하는 측정값이 있는 경우입니다. 이 경우 자동 존재는 여러 필터를 하나로 병합하고 기존 값 조합에 대해서만 필터링합니다. 이 병합으로 인해 측정값은 기존 값 조합에 따라 계산되

고 결과는 예상대로 모든 값이 아닌 필터링된 값을 기반으로 합니다. 자동 존재 및 계산에 미치는 영향에 대한 자세한 내용은 Microsoft MVP Alberto Ferrari의 [DAX Auto-Exist](#) 이해 문서를 sql.bi.com 참조하세요.

- 다음 표에서는 다양한 시나리오에서 ALL 및 ALLEXCEPT 함수를 사용하는 방법을 설명합니다.

함수 및 사용	설명
ALL()	모든 위치에서 모든 필터를 제거합니다. ALL()은 필터를 지우는 데만 사용할 수 있지만 테이블을 반환하는 데는 사용할 수 없습니다.
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다. 실제로 ALL(Table)은 테이블의 모든 값을 반환하고, 그렇지 않으면 적용되었을 수 있는 컨텍스트에서 필터를 제거합니다. 이 함수는 여러 수준의 그룹화 작업을 하고 집계된 값과 총 값의 비율을 만드는 계산을 만들 때 유용합니다. 첫 번째 예제에서는 이 시나리오를 보여 줍니다.
ALL(Column[, Column[, ...]])	테이블의 지정된 열에서 모든 필터를 제거합니다. 테이블의 다른 열에 대한 다른 모든 필터는 계속 적용됩니다. 모든 열 인수는 동일한 테이블에서 와야 합니다. ALL(Column) 변형은 하나 이상의 특정 열에 대한 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터를 유지하려는 경우에 유용합니다. 두 번째 및 세 번째 예제에서는 이 시나리오를 보여 줍니다.
ALLEXCEPT(Table, Column1 [,Column2]...)	지정한 열에 적용되는 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다. 이는 테이블의 여러 열에서 필터를 제거하려는 경우에 편리한 바로 가기입니다.

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

예 1

총 판매액에 대한 범주 판매 비율 계산

피벗 테이블에서 현재 셀의 판매액을 모든 재판매인의 총 판매액으로 나눈다고 가정합니다. 피벗 테이블 사용자가 데이터를 필터링하거나 그룹화하는 방법에 관계없이 분모가 동일한지 확인하려면 ALL을 사용하여 올바른 총합계를 만드는 수식을 정의합니다.

다음 표에서는 코드 섹션에 표시된 수식을 사용하여 새 측정값인 **All Reseller Sales Ratio**를 만들 때의 결과를 보여 줍니다. 작동 방식을 확인하려면 피벗 테이블의 행 레이블 영역에 **CalendarYear** 필드를 추가하고 **ProductCategoryName** 필드를 열 레이블 영역에 추가합니다. 그런 다음, 측정값인 **모든 재판매인 판매 비율**을 피벗 테이블의 값 영역으로 끕니다. 결과를 백분율로 보려면 Excel의 서식 기능을 사용하여 측정값이 포함된 셀에 백분율 숫자 서식을 적용합니다.

Row Labels	Accessories	Bikes	Clothing	Components	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

DAX

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

수식은 다음과 같이 생성됩니다.

1. 숫자는 `SUMX(ResellerSales_USD,`
`ResellerSales_USD[SalesAmount_USD])` CalendarYear 및 ProductCategoryName에 컨
 텍스트 필터가 적용된 피벗 테이블의 현재 셀에 대한
`ResellerSales_USD[SalesAmount_USD]`의 값 합계입니다.
2. 분모의 경우 먼저 테이블을 지정하고 `ResellerSales_USD ALL` 함수를 사용하여 테이
 블의 모든 컨텍스트 필터를 제거합니다.
3. 그런 다음 `SUMX` 함수를 사용하여 `ResellerSales_USD[SalesAmount_USD]` 열의 값을
 합산합니다. 즉, 모든 재판매인 판매에 대한 `ResellerSales_USD[SalesAmount_USD]`
 의 합계를 얻습니다.

예제 2

현재 연도까지의 총 판매액에 대한 제품 판매 비율 계산

각 제품 범주(ProductCategoryName)에 대해 연도별 매출 비율을 보여 주는 테이블을 만
 들려는 경우를 가정합니다. ProductCategoryName의 각 값에 대해 매년 백분율을 얻으려
 면 해당 특정 연도 및 제품 범주의 판매 합계를 모든 연도의 동일한 제품 범주에 대한 판
 매 합계로 나누어야 합니다. 즉, ProductCategoryName에서 필터를 유지하지만 백분율의
 분모를 계산할 때 연도에 필터를 제거하려고 합니다.

다음 표에서는 코드 섹션에 표시된 수식을 사용하여 새 측정값인 Reseller Sales Year를 만들 때의 결과를 보여 줍니다. 작동 방식을 확인하려면 피벗 테이블의 행 레이블 영역에 CalendarYear 필드를 추가하고 ProductCategoryName 필드를 열 레이블 영역에 추가합니다. 결과를 백분율로 보려면 Excel의 서식 기능을 사용하여 측정값 인 Reseller Sales Year가 포함된 셀에 백분율 숫자 서식을 적용합니다.

행 레이블	Accessories	Bikes	Clothing	Components	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

DAX

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

수식은 다음과 같이 생성됩니다.

1. numerator `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])` 는 CalendarYear 및 ProductCategoryName 열에 컨텍스트 필터가 적용된 피벗 테이블의 현재 셀에 대한 ResellerSales_USD[SalesAmount_USD]의 값 합계입니다.
2. 분모의 경우 ALL(Column) 함수를 사용하여 CalendarYear에서 기존 필터를 제거합니다. 열 레이블에서 기존 컨텍스트 필터를 적용한 후 ResellerSales_USD 테이블의 다시 기본 행에 대한 합계를 계산합니다. 순 효과는 분모의 경우 합계가 선택한 ProductCategoryName(암시적 컨텍스트 필터)과 Year의 모든 값에 대해 계산된다는 것입니다.

예 3

연간 총 매출에 대한 제품 범주 기여도 계산

각 제품 범주에 대한 매출 비율을 연 단위로 표시하는 테이블을 만들려는 경우를 가정합니다. 특정 연도의 각 제품 범주에 대한 백분율을 얻으려면 해당 특정 제품 범주 (ProductCategoryName)의 판매 합계를 n년 단위로 계산한 다음 결과 값을 모든 제품 범

주에 대해 연도 n의 판매 합계로 나눕니다. 즉, 연도에 필터를 유지하지만 백분율의 분모를 계산할 때 ProductCategoryName에서 필터를 제거하려고 합니다.

다음 표에서는 코드 섹션에 표시된 수식을 사용하여 새 측정값인 Reseller Sales CategoryName을 만들 때의 결과를 보여 줍니다. 작동 방식을 확인하려면 피벗 테이블의 행 레이블 영역에 CalendarYear 필드를 추가하고 ProductCategoryName 필드를 열 레이블 영역에 추가합니다. 그런 다음 피벗 테이블의 값 영역에 새 측정값을 추가합니다. 결과를 백분율로 보려면 Excel의 서식 기능을 사용하여 새 측정값 인 Reseller Sales CategoryName이 포함된 셀에 백분율 숫자 서식을 적용합니다.

Row Labels	Accessories	Bikes	Clothing	Components	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

DAX

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

수식은 다음과 같이 생성됩니다.

- 숫자는 `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])` 피벗 테이블의 현재 셀에 대한 `ResellerSales_USD[SalesAmount_USD]`에 있는 값의 합계이며, 필드, `CalendarYear` 및 `ProductCategoryName`에 컨텍스트 필터가 적용됩니다.
- 분모의 경우 `ALL(Column)` 함수를 사용하여 `ProductCategoryName`에서 필터를 제거하고 행 레이블에서 기존 컨텍스트 필터를 적용한 후 `ResellerSales_USD` 테이블에서 다시 기본 행에 대한 합계를 계산합니다. 순 효과는 분모의 경우 선택한 연도(암시적 컨텍스트 필터)와 `ProductCategoryName`의 모든 값에 대해 합계가 계산된다는 것입니다.

참고 항목

필터 함수

ALL 함수

ALLEXCEPT 함수

FILTER 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ALLCROSSFILTERED

아티클 • 2023. 10. 20.

테이블에 적용되는 모든 필터를 지웁니다.

구문

DAX

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

매개 변수

용어	정의
table	필터를 지울 테이블입니다.

반환 값

해당 없음. 설명을 참조하세요.

설명

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

예시

DAX

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

```

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

```

반환

DimSalesReason[SalesReasonName]	[TotalQuantity1]	[TotalQuantity2]
데모 이벤트	60398	
잡지 광고	60398	
제조업체	60398	1818
프로모션 중	60398	7390
기타	60398	3653
가격	60398	47733
품질	60398	1551
검토	60398	1640
스폰서십	60398	
텔레비전 광고	60398	730

① 참고

FactInternetSales 테이블과 DimSalesReason 테이블 사이에는 직접 또는 간접 다 대다 관계가 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ALLEXCEPT

아티클 • 2023. 10. 20.

지정한 열에 적용된 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다.

구문

DAX

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

매개 변수

용어	정의
table	후속 인수에 지정된 열의 필터를 제외하고 모든 컨텍스트 필터가 제거되는 테이블입니다.
column	컨텍스트 필터를 유지해야 하는 열입니다.

ALLEXCEPT 함수에 대한 첫 번째 인수는 기본 테이블에 대한 참조여야 합니다. 모든 후속 인수는 기본 열에 대한 참조여야 합니다. ALLEXCEPT 함수에는 테이블 식이나 열 식을 사용할 수 없습니다.

반환 값

지정된 열의 필터를 제외한 모든 필터가 제거된 테이블입니다.

설명

- 이 함수는 그 자체로 사용되지 않지만 다른 계산이 수행되는 결과 집합을 변경하는데 사용할 수 있는 중간 함수 역할을 합니다.
- ALL 및 ALLEXCEPT는 다음과 같은 다양한 시나리오에서 사용할 수 있습니다.

함수 및 사용	설명
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다. 실제로 ALL(Table)은 테이블의 모든 값을 반환하고, 그렇지 않으면 적용되었을 수 있는 컨텍스트에서 필터를 제거합니다. 이 함수는 여러 수준의 그룹화 작업을

함수 및 사용	설명
	하고 집계된 값과 총 값의 비율을 만드는 계산을 만들 때 유용합니다.
ALL(Column[, Column[, ...]])	테이블의 지정된 열에서 모든 필터를 제거합니다. 테이블의 다른 열에 대한 다른 모든 필터는 계속 적용됩니다. 모든 열 인수는 동일한 테이블에서 와야 합니다. ALL(Column) 변형은 하나 이상의 특정 열에 대한 컨텍스트 필터를 제거하고 다른 모든 컨텍스트 필터를 유지하려는 경우에 유용합니다.
ALLEXCEPT(Table, Column1 [,Column2]...)	지정한 열에 적용되는 필터를 제외하고 테이블의 모든 컨텍스트 필터를 제거합니다. 이는 테이블의 여러 열에서 필터를 제거하려는 경우에 편리한 바로 가기입니다.

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

예시

다음 측정값 수식은 SalesAmount_USD 합계를 계산하고 ALLEXCEPT 함수를 사용하여 DateTime 테이블에서 필터가 CalendarYear 열에 적용된 경우를 제외하고 모든 컨텍스트 필터를 제거합니다.

DAX

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

수식은 ALLEXCEPT를 사용하므로 DateTime 테이블의 CalendarYear를 제외한 모든 열이 시각화를 조각화하는 데 사용될 때마다 수식은 모든 슬라이서 필터를 제거하여 SalesAmount_USD 합계와 같은 값을 제공합니다. 그러나 CalendarYear 열을 사용하여 시각화를 조각화하는 경우 결과는 다릅니다. CalendarYear는 ALLEXCEPT에 대한 인수로 지정되므로 연도에 데이터가 조각화되면 행 수준에서 연도에 필터가 적용됩니다.

참고 항목

[필터 함수](#)

[ALL 함수](#)

[FILTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ALLNOBLANKROW

아티클 • 2023. 10. 20.

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

구문

DAX

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

매개 변수

용어	정의
table	모든 컨텍스트 필터가 제거되는 테이블입니다.
column	모든 컨텍스트 필터가 제거되는 열입니다.

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

반환 값

전달된 매개 변수가 테이블이거나 값 열인 경우 전달된 매개 변수가 열인 테이블입니다.

설명

- ALLNOBLANKROW 함수는 부모 테이블이 부모 열에 일치하지 않는 값을 가진 자식 테이블에 하나 이상의 행이 있을 때 부모 테이블이 표시하는 빈 행만 필터링합니다. 자세한 설명은 아래 예제를 참조하세요.
- 다음 표에는 DAX에서 제공되는 ALL의 변형과 그 차이점이 요약되어 있습니다.

함수 및 사용	설명
ALL(열)	테이블의 지정된 열에서 모든 필터를 제거합니다. 테이블의 다른 모든 필터는 다른 열보다 계속 적용됩니다.
ALL(Table)	지정된 테이블에서 모든 필터를 제거합니다.

함수 및 사용	설명
ALLEXCEPT(Table,Col1,Col2...)	지정한 열을 제외한 테이블의 모든 컨텍스트 필터를 재정의 합니다.
ALLNOBLANK(table column)	관계의 부모 테이블에서 빈 행을 제외한 모든 행 또는 빈 행을 제외한 열의 모든 고유 값을 반환하고 존재할 수 있는 컨텍스트 필터를 무시합니다.

ALL(Table) 및 ALL(Column)을 사용하는 단계별 예제와 함께 ALL 함수의 작동 방식에 대한 일반적인 설명은 [ALL 함수를 참조하세요](#).

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

예시

샘플 데이터에서 ResellerSales_USD 테이블에는 값이 없으므로 통합 문서 내 관계의 부모 테이블과 관련될 수 없는 행이 하나 포함되어 있습니다. 빈 행 동작과 관련 없는 데이터에 대한 개수를 처리하는 방법을 볼 수 있도록 피벗 테이블에서 이 테이블을 사용합니다.

1단계: 관련 없는 데이터 확인

파워 피벗 창을 열고 ResellerSales_USD 테이블을 선택합니다. ProductKey 열에서 빈 값을 필터링합니다. 한 행이 다시 기본. 해당 행에서 SalesOrderLineNumber를 제외한 모든 열 값은 비어 있어야 합니다.

2단계: 피벗 테이블 만들기

새 피벗 테이블을 만든 다음 날짜/시간 열을 끊습니다. [Calendar Year], 행 레이블 창 다음 표에서는 예상 결과를 보여줍니다.

Row Labels
2005
2006
2007
2008
총합계

2008년과 총합계 사이의 빈 레이블을 확인합니다. 이 빈 레이블은 부모 테이블에 일치하는 값이 없는 자식 테이블의 값을 설명하기 위해 만들어진 특수 그룹인 알 수 없는 멤버를

나타냅니다. 이 예제에서는 datetime입니다. [Calendar Year] 열입니다.

피벗 테이블에 이 빈 레이블이 표시되면 날짜/시간 열과 관련된 일부 테이블에서 알 수 있습니다. [Calendar Year]에는 빈 값 또는 일치하지 않는 값이 있습니다. 부모 테이블은 빈 레이블을 표시하는 테이블이지만 일치하지 않는 행은 하나 이상의 자식 테이블에 있습니다.

이 빈 레이블 그룹에 추가되는 행은 부모 테이블의 값과 일치하지 않는 값(예: datetime 테이블에 없는 날짜) 또는 null 값으로, 날짜 값이 전혀 없음을 의미합니다. 이 예제에서는 자식 판매 테이블의 모든 열에 빈 값을 배치했습니다. 자식 테이블보다 부모 테이블에 값이 더 많으면 문제가 발생하지 않습니다.

3단계: ALL 및 ALLNOBLANK를 사용하여 행 개수 계산

datetime 테이블에 다음 두 측정값을 추가하여 테이블 행 수를 계산합니다. **datetime**의 Countrows ALLNOBLANK, Countrows ALL of datetime. 이러한 측정값을 정의하는 데 사용할 수 있는 수식은 다음과 같습니다.

DAX

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

빈 피벗 테이블에서 날짜/시간을 추가합니다. [Calendar Year] 열을 행 레이블에 추가한 다음 새로 만든 측정값을 추가합니다. 결과는 다음 표와 같습니다.

Row Labels	datetime의 Countrows ALLNOBLANK	Countrows ALL of datetime
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
총합계	1280	1281

결과는 테이블 행 수에서 1행의 차이를 보여 줍니다. 그러나 파워 피벗 창을 열고 datetime 테이블을 선택하면 여기에 멘션 특수한 빈 행이 알 수 없는 멤버이므로 테이블에서 빈 행을 찾을 수 없습니다.

4단계: 개수가 정확한지 확인

ALLNOBLANKROW가 실제로 빈 행을 계산하지 않고 부모 테이블의 특수 빈 행만 처리한다는 것을 증명하기 위해 ResellerSales_USD 테이블에 다음 두 측정값을 추가합니다.
ResellerSales_USD Countrows ALLNOBLANKROW, Countrows ALL of ResellerSales_USD.

새 피벗 테이블을 만들고 날짜/시간 열을 끊습니다. [Calendar Year], 행 레이블 창 이제 방금 만든 측정값을 추가합니다. 결과는 다음과 같습니다.

Row Labels	ResellerSales_USD 개수 ALLNOBLANKROW	ResellerSales_USD 모두 계산
2005	60856	60856
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856
총합계	60856	60856

이제 두 측정값의 결과가 동일합니다. ALLNOBLANKROW 함수는 테이블에서 실제로 빈 행을 계산하지 않고, 관계의 자식 테이블 중 하나 이상이 일치하지 않는 값이나 빈 값을 포함하는 경우 부모 테이블에서 생성된 특수한 경우인 빈 행만 처리하기 때문입니다.

참고 항목

[필터 함수](#)

[ALL 함수](#)

[FILTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ALLSELECTED

아티클 • 2023. 10. 20.

다른 모든 컨텍스트 필터 또는 명시적 필터를 유지하면서 현재 쿼리의 열 및 행에서 컨텍스트 필터를 제거합니다.

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

구문

DAX

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

매개 변수

용어	정의
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 이 매개 변수는 식일 수 없습니다. 이 매개 변수는 선택 사항입니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다. 이 매개 변수는 선택적 요소입니다.

반환 값

열 및 행 필터가 없는 쿼리의 컨텍스트입니다.

설명

- 인수가 하나 있는 경우 인수는 tableName 또는 columnName입니다. 둘 이상의 인수가 있는 경우 동일한 테이블의 열이어야 합니다.
- 이 함수는 쿼리 내에서 명시적으로 설정된 모든 필터를 유지하고 행 및 열 필터 이외의 모든 컨텍스트 필터를 유지하므로 ALL()과 다릅니다.

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

예시

다음 예제에서는 DAX 식을 사용하여 테이블 보고서에서 다양한 수준의 시각적 합계를 생성하는 방법을 보여 줍니다. 보고서에서 두 개의 이전 필터가 재판매인 판매 데이터에 적용되었습니다. 하나는 Sales Territory Group = *Europe*이고 다른 하나는 프로모션 유형 = *볼륨 할인*입니다. 필터가 적용되면 전체 보고서, 모든 연도 또는 모든 제품 범주에 대한 시각적 합계를 계산할 수 있습니다. 또한 그림상 모든 재판매인 판매의 총합계도 가져와 보고서의 모든 필터를 제거합니다. 다음 DAX 식을 평가하면 Visual Totals를 사용하여 테이블을 빌드하는 데 필요한 모든 정보가 포함된 테이블이 생성됩니다.

DAX

```

define
measure 'Reseller Sales'[Reseller Sales Amount]=sum('Reseller Sales'[Sales
Amount])
measure 'Reseller Sales'[Reseller Grand Total]=calculate(sum('Reseller
Sales'[Sales Amount]), ALL('Reseller Sales'))
measure 'Reseller Sales'[Reseller Visual Total]=calculate(sum('Reseller
Sales'[Sales Amount]), ALLSELECTED())
measure 'Reseller Sales'[Reseller Visual Total for All of Calendar
Year]=calculate(sum('Reseller Sales'[Sales Amount]),
ALLSELECTED('Date'[Calendar Year]))
measure 'Reseller Sales'[Reseller Visual Total for All of Product Category
Name]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Product
Category'[Product Category Name]))
evaluate
CalculateTable(
    //CT table expression
    summarize(
//summarize table expression
crossjoin(distinct('Product Category'[Product Category Name]),
distinct('Date'[Calendar Year]))
//First Group by expression
, 'Product Category'[Product Category Name]
//Second Group by expression
, 'Date'[Calendar Year]
//Summary expressions
, "Reseller Sales Amount", [Reseller Sales Amount]
, "Reseller Grand Total", [Reseller Grand Total]
, "Reseller Visual Total", [Reseller Visual Total]
, "Reseller Visual Total for All of Calendar Year", [Reseller Visual Total
for All of Calendar Year]
, "Reseller Visual Total for All of Product Category Name", [Reseller Visual
Total for All of Product Category Name]
)
//CT filters
, 'Sales Territory'[Sales Territory Group]="Europe", 'Promotion'[Promotion

```

```

Type]="Volume Discount"
)
order by [Product Category Name], [Calendar Year]

```

AdventureWorks DW 테이블 형식 모델에 대해 SQL Server Management Studio에서 위의 쿼리를 실행한 후 다음 결과를 가져옵니다.

[제품 범주 이름]	[역 년]	[재판매인 판매액]	[재판매인 총합 계]	[재판매인 Visual Total]	[Reseller Visual Total for All of Calendar Year]	[모든 제품 범주 이름에 대한 재판매인 시각적 합계]
Accessories	2000		80450596.9823	877006.7987	38786.018	
Accessories	2001		80450596.9823	877006.7987	38786.018	
Accessories	2002	625.7933	80450596.9823	877006.7987	38786.018	91495.3104
Accessories	2003	26037.3132	80450596.9823	877006.7987	38786.018	572927.0136
Accessories	2004	12122.9115	80450596.9823	877006.7987	38786.018	212584.4747
Accessories	2005		80450596.9823	877006.7987	38786.018	
Accessories	2006		80450596.9823	877006.7987	38786.018	
Bikes	2000		80450596.9823	877006.7987	689287.7939	
Bikes	2001		80450596.9823	877006.7987	689287.7939	
Bikes	2002	73778.938	80450596.9823	877006.7987	689287.7939	91495.3104
Bikes	2003	439771.4136	80450596.9823	877006.7987	689287.7939	572927.0136
Bikes	2004	175737.4423	80450596.9823	877006.7987	689287.7939	212584.4747
Bikes	2005		80450596.9823	877006.7987	689287.7939	
Bikes	2006		80450596.9823	877006.7987	689287.7939	
의류	2000		80450596.9823	877006.7987	95090.7757	
의류	2001		80450596.9823	877006.7987	95090.7757	
의류	2002	12132.4334	80450596.9823	877006.7987	95090.7757	91495.3104
의류	2003	58234.2214	80450596.9823	877006.7987	95090.7757	572927.0136
의류	2004	24724.1209	80450596.9823	877006.7987	95090.7757	212584.4747
의류	2005		80450596.9823	877006.7987	95090.7757	

[제품 범주 이름]	[연도]	[재판매인 판매액]	[재판매인 총합계]	[재판매인 Visual Total]	[Reseller Visual Total for All of Calendar Year]	[모든 제품 범주 이름에 대한 재판매인 시각적 합계]
의류	2006	80450596.9823	877006.7987	95090.7757		
구성 요소	2000	80450596.9823	877006.7987	53842.2111		
구성 요소	2001	80450596.9823	877006.7987	53842.2111		
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.

재판매인 판매액

연도 및 제품 범주에 대한 Reseller Sales의 실제 값입니다. 이 값은 보고서 가운데에 있는 연도와 케이터리의 교차점에 있는 셀에 나타납니다.

Reseller Visual Total for All of Calendar Year

모든 연도의 제품 범주에 대한 총 값입니다. 이 값은 지정된 제품 범주에 대한 열 또는 행의 끝과 보고서의 모든 연도에 걸쳐 표시됩니다.

모든 제품 범주 이름에 대한 재판매인 Visual Total

모든 제품 범주에서 1년 동안의 총 값입니다. 이 값은 지정된 연도의 열 또는 행 끝과 보고서의 모든 제품 범주에 표시됩니다.

재판매인 Visual Total

모든 연도 및 제품 범주의 총 값입니다. 이 값은 일반적으로 테이블의 오른쪽 아래 모서리에 나타납니다.

재판매인 총합계

필터가 적용되기 전의 모든 재판매인 판매의 총합계입니다. [Reseller Visual Total]의 차이점을 알 수 있습니다. 이 보고서에는 제품 범주 그룹에 하나씩, 다른 하나는 프로모션 유형에 있는 두 개의 필터가 포함되어 있습니다.

① 참고

식에 명시적 필터가 있는 경우 해당 필터도 식에 적용됩니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

CALCULATE

아티클 • 2024. 01. 18.

수정된 필터 컨텍스트에서 식을 평가합니다.

① 참고

CALCULATETABLE 함수도 있습니다. 테이블 자체를 반환하는 식에 적용된 필터 컨텍스트를 수정한다는 점을 제외하고 정확히 동일한 기능을 수행합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
식	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 같습니다.

- 부울 필터 식
- 테이블 필터 식
- 필터 수정 함수

필터가 여러 개 있는 경우 AND(&> 논리 연산자)를 사용하여 평가할 수 있습니다. 즉, 모든 조건이 TRUE이거나 OR(||) 논리 연산자를 통해 평가될 수 있습니다. 즉, 두 조건 중 하나가 true일 수 있습니다.

부울 필터 식

부울 식 필터는 TRUE 또는 FALSE로 평가되는 식입니다. 준수해야 하는 몇 가지 규칙이 있습니다.

- 단일 테이블의 열을 참조할 수 있습니다.
- 측정값을 참조할 수 없습니다.
- 중첩된 CALCULATE 함수를 사용할 수 없습니다.

Power BI Desktop의 2021년 9월 릴리스부터 다음이 적용됩니다.

- 집계 함수에 인수로 전달되지 않는 한 테이블을 검색하거나 반환하는 함수를 사용 할 수 없습니다.
- 스칼라 값을 반환하는 집계 함수를 포함할 수 있습니다. 예를 들어

DAX

```
Total sales on the last selected date =  
CALCULATE (  
    SUM ( Sales[Sales Amount] ),  
    'Sales'[OrderDateKey] = MAX ( 'Sales'[OrderDateKey] )  
)
```

테이블 필터 식

테이블 식 필터는 테이블 개체를 필터로 적용합니다. 모델 테이블에 대한 참조일 수 있지만 테이블 개체를 반환하는 함수일 가능성이 높습니다. FILTER 함수를 사용하여 부울 필터 식으로 정의할 수 없는 필터 조건을 포함하여 복잡한 필터 조건을 적용할 수 있습니다.

필터 한정자 함수

필터 한정자 함수를 사용하면 단순히 필터를 추가하는 것 이상을 수행할 수 있습니다. 필터 컨텍스트를 수정할 때 추가 제어를 제공합니다.

테이블 확장

함수	목적
REMOVEFILTERS	테이블의 하나 이상의 열 또는 단일 테이블의 모든 열에서 모든 필터 또는 필터를 제거합니다.
ALL ¹ , ALLEXCEPT, ALLNOBLANKROW	하나 이상의 열 또는 단일 테이블의 모든 열에서 필터를 제거합니다.
KEEPFILTERS	동일한 열에서 기존 필터를 제거하지 않고 필터를 추가합니다.

함수	목적
USERELATIONSHIP	관련 열 간에 비활성 관계를 설정합니다. 이 경우 활성 관계는 자동으로 비활성 상태가 됩니다.
크로스필터	필터 방향을 수정하거나(둘 다에서 단일로, 단일에서 둘 다로) 관계를 사용하지 않도록 설정합니다.

¹ ALL 함수와 해당 변형은 필터 한정자와 테이블 개체를 반환하는 함수로 작동합니다. REMOVEFILTERS 함수가 도구에서 지원되는 경우 필터를 제거하는 데 사용하는 것이 좋습니다.

반환 값

식의 결과인 값입니다.

설명

- 필터 식이 제공되면 CALCULATE 함수는 필터 컨텍스트를 수정하여 식을 평가합니다. 각 필터 식에 대해 필터 식이 KEEPFILTERS 함수에 래핑되지 않은 경우 두 가지 가능한 표준 결과가 있습니다.
 - 열(또는 테이블)이 필터 컨텍스트에 없는 경우 새 필터가 필터 컨텍스트에 추가되어 식을 평가합니다.
 - 열(또는 테이블)이 이미 필터 컨텍스트에 있는 경우 새 필터에서 기존 필터를 덮어쓰고 CALCULATE 식을 평가합니다.
- 필터 없이 사용되는 CALCULATE 함수는 특정 요구 사항을 달성합니다. 행 컨텍스트를 필터링 컨텍스트로 전환합니다. 모델 데이터를 요약하는 식(모델 측정값 아님)을 행 컨텍스트에서 평가해야 하는 경우 필요합니다. 이 시나리오는 계산 열 수식에서 발생하거나 반복기 함수의 식이 평가될 때 발생할 수 있습니다. 모델 측정값이 행 컨텍스트에서 사용되는 경우 컨텍스트 전환은 자동으로 수행됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

다음 Sales 테이블 측정값 정의는 수익 결과를 생성하지만 파란색이 있는 제품에 대해서만 생성됩니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

DAX

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

테이블 확장

범주	매출액	Blue Revenue
Accessories	\$1,272,057.89	\$165,406.62
Bikes	\$94,620,526.21	\$8,374,313.88
의류	\$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 수정된 필터 컨텍스트에서 Sales 테이블 Sales Amount 열의 합계를 평가합니다. 새 필터가 Product 테이블 색 열에 추가되거나 필터가 열에 이미 적용된 필터를 덮어씁니다.

다음 Sales 테이블 측정값 정의는 모든 판매 채널의 판매 대비 판매 비율을 생성합니다.

테이블 확장

채널	매출액	수익 % 총 채널
인터넷	\$29,358,677.22	26.74%
재판매인	\$80,450,596.98	73.26%
합계	\$109,809,274.20	100.00%

DAX

```
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 컨텍스트)의 합계를 수정된 필터 컨텍스트의 동일한 식으로 나누는 식입니다. 필터 한정자 함수인 REMOVEFILTERS 함수를 사용하여 필터 컨텍스트를 수정하는 CALCULATE 함수입니다. Sales Order 테이블 채널 열에서 필터를 제거합니다.

다음 Customer 테이블 계산 열 정의는 고객을 로열티 클래스로 분류합니다. 매우 간단한 시나리오입니다. 고객이 창출한 수익이 \$2500 미만이면 낮음으로 분류되고, 그렇지 않으면 높음으로 분류됩니다.

DAX

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

이 예제에서는 행 컨텍스트가 필터 컨텍스트로 변환됩니다. 컨텍스트 전환이라고 합니다. ALLEXCEPT 함수는 CustomerKey 열을 제외한 모든 Customer 테이블 열에서 필터를 제거합니다.

관련 콘텐츠

[필터 컨텍스트](#)

[행 컨텍스트](#)

[CALCULATETABLE 함수](#)

[필터 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CALCULATETABLE

아티클 • 2023. 10. 20.

수정된 필터 컨텍스트에서 테이블 식을 평가합니다.

① 참고

CALCULATE 함수도 있습니다. 스칼라 값을 반환하는 식에 적용된 필터 컨텍스트를 수정한다는 점을 제외하고 정확히 동일한 기능을 수행합니다.

구문

DAX

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

매개 변수

용어	정의
식	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 같습니다.

- 부울 필터 식
- 테이블 필터 식
- 필터 수정 함수

여러 필터가 있는 경우 AND 논리 연산자를 사용하여 평가됩니다. 즉, 모든 조건이 동시에 TRUE여야 합니다.

부울 필터 식

부울 식 필터는 TRUE 또는 FALSE로 평가되는 식입니다. 준수해야 하는 몇 가지 규칙이 있습니다.

- 단일 열만 참조할 수 있습니다.
- 측정값을 참조할 수 없습니다.
- 중첩된 CALCULATE 함수를 사용할 수 없습니다.

Power BI Desktop의 2021년 9월 릴리스부터 다음이 적용됩니다.

- 집계 함수에 인수로 전달되지 않는 한 테이블을 검색하거나 반환하는 함수를 사용 할 수 없습니다.
- 스칼라 값을 반환하는 집계 함수를 포함할 수 있습니다.

테이블 필터 식

테이블 식 필터는 테이블 개체를 필터로 적용합니다. 모델 테이블에 대한 참조일 수 있지만 테이블 개체를 반환하는 함수일 가능성이 높습니다. FILTER 함수를 사용하여 부울 필터 식으로 정의할 수 없는 필터 조건을 포함하여 복잡한 필터 조건을 적용할 수 있습니다.

필터 한정자 함수

필터 한정자 함수를 사용하면 단순히 필터를 추가하는 것 이상을 수행할 수 있습니다. 필터 컨텍스트를 수정할 때 추가 제어를 제공합니다.

함수	목적
REMOVEFILTERS	테이블의 하나 이상의 열 또는 단일 테이블의 모든 열에서 모든 필터 또는 필터를 제거합니다.
ALL ¹ , ALLEXCEPT, ALLNOBLANKROW	하나 이상의 열 또는 단일 테이블의 모든 열에서 필터를 제거합니다.
KEEPFILTERS	동일한 열에서 기존 필터를 제거하지 않고 필터를 추가합니다.
USERELATIONSHIP	관련 열 간에 비활성 관계를 설정합니다. 이 경우 활성 관계는 자동으로 비활성 상태가 됩니다.
크로스필터	필터 방향을 수정하거나(둘 다에서 단일로, 단일에서 둘 다로) 관계를 사용하지 않도록 설정합니다.

¹ ALL 함수와 해당 변형은 필터 한정자와 테이블 개체를 반환하는 함수로 작동합니다.

REMOVEFILTERS 함수가 도구에서 지원되는 경우 필터를 제거하는 데 사용하는 것이 좋습니다.

반환 값

값의 테이블입니다.

설명

- 필터 식이 제공되면 CALCULATETABLE 함수는 필터 컨텍스트를 수정하여 식을 평가합니다. 각 필터 식에 대해 필터 식이 KEEPFILTERS 함수에 래핑되지 않은 경우 두 가지 가능한 표준 결과가 있습니다.
 - 열(또는 테이블)이 필터 컨텍스트에 없는 경우 새 필터가 필터 컨텍스트에 추가되어 식을 평가합니다.
 - 열(또는 테이블)이 이미 필터 컨텍스트에 있는 경우 새 필터에서 기존 필터를 덮어쓰고 CALCULATETABLE 식을 평가합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 CALCULATETABLE 함수를 사용하여 2006년 인터넷 판매 합계를 가져옵니다. 이 값은 나중에 2006년의 모든 판매와 비교하여 인터넷 판매 비율을 계산하는 데 사용됩니다.

다음 수식은 다음과 같습니다.

DAX

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

그러면 다음 표가 표시됩니다.

Row Labels	인터넷 SalesAmount_USD	CalculateTable 2006 Internet Sales	Internet Sales to 2006 ratio
2005	\$2,627,031.40	\$5,681,440.58	0.46
2006	\$5,681,440.58	\$5,681,440.58	1.00

Row Labels	인터넷 SalesAmount_USD	CalculateTable 2006 Internet Sales	Internet Sales to 2006 ratio
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)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

이전

아티클 • 2023. 10. 20.

멘션 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.

EARLIER은 특정 값을 입력으로 사용하고 해당 입력을 기반으로 계산을 생성하려는 중첩된 계산에 유용합니다. Microsoft Excel에서는 현재 행의 컨텍스트 내에서만 이러한 계산을 수행할 수 있습니다. 그러나 DAX에서는 입력 값을 저장한 다음 전체 테이블의 데이터를 사용하여 계산할 수 있습니다.

EARLIER은 주로 계산 열의 컨텍스트에서 사용됩니다.

구문

DAX

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

매개 변수

용어	정의
column	열로 확인되는 열 또는 식입니다.
num	(선택 사항) 외부 평가 패스에 대한 양수입니다. 다음 평가 수준 아웃은 1로 표시됩니다. 2개 수준 아웃은 2로 표시됩니다. 생략된 기본값은 1입니다.

반환 값

열의 현재 행 값(외부 평가 패스 수)입니다.

예외

오류에 대한 설명

설명

- 테이블 검색을 시작하기 전에 행 컨텍스트가 있는 경우 EARLIER 이 성공합니다. 그렇지 않으면 오류를 반환합니다.

- EARLIER는 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 행이 10개 있는 경우 약 100개의 작업이 필요할 수 있습니다. 행이 100개이면 10,000개에 가까운 작업이 수행될 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

① 참고

실제로 VertiPaq 메모리 내 분석 엔진은 실제 계산 수를 줄이기 위해 최적화를 수행하지만 재귀를 포함하는 수식을 만들 때는 주의해야 합니다.

예시

EARLIER의 사용을 설명하기 위해 순위 값을 계산한 다음 다른 계산에서 해당 순위 값을 사용하는 시나리오를 빌드해야 합니다.

다음 예제는 이 간단한 테이블인 **ProductSubcategory**를 기반으로 하며 각 **ProductSubcategory**의 총 매출을 보여 줍니다.

순위 열을 포함한 최종 테이블이 여기에 표시됩니다.

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
18	밥 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	Caps	\$47,934.54	28
7	Chains	\$8,847.08	35
29	Cleaners	\$16,882.62	32
8	크랭크 세트	\$191,522.09	15
9	Derailleurs	\$64,965.33	22
30	Fenders	\$41,974.10	29
10	포크	\$74,727.66	21

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
20	장갑	\$228,353.58	12
4	핸들바	\$163,257.06	17
11	헤드세트	\$57,659.99	25
31	Helmets	\$451,192.31	9
32	Hydration Packs	\$96,893.78	20
21	Jerseys	\$699,429.78	7
33	조명		36
34	잠금	\$15,059.47	33
1	산악용 자전거	\$34,305,864.29	2
12	Mountain Frames	\$4,511,170.68	4
35	Panniers		36
13	Pedals	\$140,422.20	19
36	펌프	\$12,695.18	34
2	도로용 자전거	\$40,551,696.34	1
14	Road Frames	\$3,636,398.71	5
15	Saddles	\$52,526.47	26
22	반바지	\$385,707.80	10
23	양말	\$28,337.85	31
24	Tights	\$189,179.37	16
37	Tires and Tubes	\$224,832.81	13
3	Touring Bikes	\$13,334,864.18	3
16	Touring Frames	\$1,545,344.02	6
25	Vests	\$240,990.04	11
17	바퀴	\$648,240.04	8

순위 값 만들기

행에서 지정된 값에 대한 순위 값을 가져오는 한 가지 방법은 비교되는 값보다 크거나 작은 값이 있는 동일한 테이블의 행 수를 계산하는 것입니다. 이 기술은 테이블에서 가장 높은 값에 대해 빈

값이나 0 값을 반환하는 반면, 같은 값은 동일한 순위 값을 가지며 다음 값(같은 값 이후)은 연속되지 않은 순위 값을 갖습니다. 아래 샘플을 참조하세요.

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

DAX

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

다음 단계에서는 계산 방법을 자세히 설명합니다.

- EARLIER 함수는 **테이블의 현재 행에 대한 TotalSubcategorySales** 값을 가져옵니다. 이 경우 프로세스가 시작되므로 테이블의 첫 번째 행입니다.
- EARLIER([TotalSubcategorySales])은 외부 루프의 현재 행인 \$156,167.88로 평가됩니다.
- 이제 FILTER 함수는 모든 행의 값이 \$156,167.88(EARLIER의 현재 값)보다 큰 TotalSubcategorySales 값을 갖는 테이블을 반환합니다.
- COUNTROWS 함수는 필터링된 테이블의 행 수를 계산하고 해당 값을 현재 행의 새 계산 열에 1을 더한 값에 할당합니다. 상위 순위 값이 공백이 되지 않도록 하려면 1을 추가해야 합니다.
- 계산 열 수식은 다음 행으로 이동하고 1~4단계를 반복합니다. 이러한 단계는 테이블의 끝에 도달할 때까지 반복됩니다.

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

참고 항목

[EARLIEST 함수](#)

[필터 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

초기

아티클 • 2023. 12. 14.

지정된 열의 외부 평가 패스에서 지정된 열의 현재 값을 반환합니다.

구문

DAX

`EARLiest(<column>)`

매개 변수

 테이블 확장

용어	정의
column	열에 대한 참조입니다.

반환 값

필터가 제거된 열입니다.

설명

- EARLiest 함수는 EARLIER과 유사하지만 한 가지 추가 재귀 수준을 지정할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

현재 샘플 데이터는 이 시나리오를 지원하지 않습니다.

DAX

`= EARLiest(<column>)`

관련 콘텐츠

[EARLIER 함수](#)

[필터 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FILTER

아티클 • 2023. 10. 20.

다른 테이블 또는 식의 하위 집합을 나타내는 테이블을 반환합니다.

구문

DAX

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

매개 변수

용어 정의

table 필터링할 테이블입니다. 테이블은 테이블을 생성하는 식일 수도 있습니다.

필터 테이블의 각 행에 대해 계산할 부울 식입니다. 예를 들어 `[Amount] > 0` 또는 `[Region] = "France"` 와 같습니다.

반환 값

필터링된 행만 포함하는 테이블입니다.

설명

- FILTER를 사용하여 작업 중인 테이블의 행 수를 줄이고 계산에 특정 데이터만 사용 할 수 있습니다. FILTER는 독립적으로 사용되지 않지만 테이블을 인수로 필요로 하는 다른 함수에 포함된 함수로 사용됩니다.
- FILTER를 사용하는 경우 모범 사례는 FILTER를 필터 인수로 사용하지 않는 방법을 참조하세요.

DAX에서 COUNT 대신 COUNTROWS 사용

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

예시

다음 예제에서는 미국 판매를 필터링하는 측정값을 사용한 다음, 연도 및 제품 범주별로 조각화하여 미국 외부에서 인터넷 판매 보고서를 만듭니다. 이 측정값을 만들려면 Sales Territory를 사용하여 테이블인 Internet Sales USD를 필터링한 다음 SUMX 함수에서 필터링된 테이블을 사용합니다.

이 예제에서 식은 다음과 같습니다.

DAX

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

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

다음 표에서는 아래 코드 섹션에 제공되는 수식인 NON USA Internet Sales 측정값에 대한 개념 증명을 보여 줍니다. 이 표에서는 모든 인터넷 판매를 미국 이외의 인터넷 판매와 비교하여 계산에서 미국 판매를 제외하여 필터 식이 작동한다는 것을 보여 줍니다.

이 테이블을 다시 만들려면 SalesTerritoryCountry 필드를 보고서 또는 피벗 테이블의 행 레이블 영역에 추가합니다.

표 1. 미국 및 기타 모든 지역의 총 판매액 비교

Row Labels	Internet Sales	미국 이외의 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 측정값인 NON USA Internet Sales를 사용하여 피벗 테이블을 만들 때의 결과를 보여줍니다. 피벗 테이블의 행 레이블 영역에 CalendarYear 필드를 추가하고 ProductCategoryName 필드를 열 레이블 영역에 추가합니다.

표 2. 제품 범주별 미국 이외의 판매액 비교

Row Labels	Accessories	Bikes	Clothing	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

DAX

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

참고 항목

[필터 함수](#)

[ALL 함수](#)

[ALLEXCEPT 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

INDEX

아티클 • 2024. 02. 26.

지정된 파티션 내에서 위치 매개 변수로 지정된 절대 위치에 지정된 순서로 정렬된 행을 반환합니다. 현재 파티션을 단일 파티션으로 추론할 수 없는 경우 여러 행이 반환될 수 있습니다.

구문

DAX

```
INDEX(<position>[, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
배치	<p>데이터를 가져올 절대 위치(1부터 시작):</p> <ul style="list-style-type: none">- <위치>는 양수입니다. 1은 첫 번째 행이고, 2는 두 번째 행입니다. - <위치>는 음수입니다.-1은 마지막 행이고, -2는 두 번째 마지막 행입니다. <p><위치>가 경계를 벗어나거나 0이거나 BLANK()인 경우 <INDEX>는 빈 테이블을 반환합니다. 스칼라 값을 반환하는 DAX 식일 수 있습니다.</p>
관계	<p>(선택 사항) 출력이 반환되는 테이블 식입니다.</p> <p>지정한 경우 <partitionBy>의 모든 열은 <해당 열 또는 관련 테이블에서 나와야 합니다.</p> <p>생략된 경우:</p> <ul style="list-style-type: none">- <orderBy> 를 명시적으로 지정해야 합니다.- 모든 <orderBy> 및 <partitionBy> 식은 정규화된 열 이름이어야 하며 단일 테이블에서 가져옵니다.- <orderBy> 및 <partitionBy>의 모든 열에 대한 <ALLSELECTED()>로 기본값이 지정됩니다.
축	<p>(선택 사항) 시각적 세이프의 축입니다. 시각적 계산에서만 사용할 수 있으며 <관계를 대체합니다>.</p>
orderBy	<p>(선택 사항) 각 파티션의 정렬 방법을 정의하는 식을 포함하는 ORDERBY() 절입니다.</p> <p>생략된 경우:</p> <ul style="list-style-type: none">- <관계를> 명시적으로 지정해야 합니다.- <partitionBy>에 아직 지정되지 않은 관계의 모든 열<에 <따라 정렬하는 기본값입니다>.

용어	정의
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다. 현재 지원되는 유일한 값은 DEFAULT입니다. 여기서 숫자 값에 대한 동작은 빈 값이 0과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다.
partitionBy	(선택 사항) 관계가 <정의하는 열을 포함하는 PARTITIONBY() 절입니다. 생략 <하면 관계가> 단일 파티션으로 처리됩니다.
matchBy	(선택 사항) 데이터를 일치시키고 현재 행을 식별하는 방법을 정의하는 열을 포함하는 MATCHBY() 절입니다.
reset	(선택 사항) 시각적 계산에서만 사용할 수 있습니다. 계산이 다시 설정되는지, 시각적 세이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

절대 위치에 있는 행입니다.

설명

각 <partitionBy> 및 <matchBy> 열에는 다음 동작을 사용하여 작동할 "현재 파티션"을 정의하는 데 도움이 되는 해당 외부 값이 있어야 합니다.

- 정확히 하나의 해당 외부 열이 있는 경우 해당 값이 사용됩니다.
- 해당 외부 열이 없는 경우:
 - 인덱스는 먼저 해당하는 외부 열이 없는 모든 <partitionBy> 및 <matchBy> 열을 결정합니다.
 - INDEX의 부모 컨텍스트에서 이러한 열에 대한 기존 값의 모든 조합에 대해 INDEX 가 평가되고 행이 반환됩니다.
 - INDEX의 최종 출력은 이러한 행의 통합입니다.
- 해당하는 외부 열이 두 개 이상 있으면 오류가 반환됩니다.

<matchBy>가 있는 경우 <INDEX는 matchBy 및 <partitionBy>> 열을 사용하여 <행을 식별 하려고 합니다.

`matchBy`>가 없고 `orderBy` 및 `<partitionBy>` 내에 `<지정된 열이 관계의>` 모든 행<을 고유하게 식별할 수 없는 경우<:

- INDEX는 모든 행을 고유하게 식별하는 데 필요한 최소 개수의 추가 열을 찾으려고 합니다.
- 이러한 열을 찾을 수 있는 경우 INDEX는 이러한 새 열을 `orderBy`>에 <자동으로 추가하고 각 파티션은 이 새 OrderBy 열 집합을 사용하여 정렬됩니다.
- 이러한 열을 찾을 수 없으면 오류가 반환됩니다.

다음과 같은 경우 빈 테이블이 반환됩니다.

- `PartitionBy` 열의 해당 외부 값이 관계> 내에 <없습니다.
- 위치> 값은 <파티션 내에 없는 위치를 나타냅니다.

관계>와 <동일한 테이블에 <정의된 계산 열 내에서 INDEX를 사용하고 `orderBy`>를 생략하면 오류가 반환됩니다.

<초기화>는 시각적 계산에서만 사용할 수 있으며 `orderBy` 또는 `<partitionBy>`와 함께 <사용할 수 없습니다. <다시 설정>이 있으면 축>을 <지정할 수 있지만 <관계형>은 지정할 수 없습니다.

예제 1 - 계산 열

다음 DAX 쿼리:

```
DAX  
EVALUATE INDEX(1, ALL(DimDate[CalendarYear]))
```

다음 표를 반환합니다.

 테이블 확장

DimDate[CalendarYear]
2005

예제 2 - 계산 열

다음 DAX 쿼리:

```
DAX
```

```

EVALUATE
SUMMARIZECOLUMNS (
    FactInternetSales[ProductKey],
    DimDate[MonthNumberOfYear],
    FILTER (
        VALUES(FactInternetSales[ProductKey]),
        [ProductKey] < 222
    ),
    "CurrentSales", SUM(FactInternetSales[SalesAmount]),
    "LastMonthSales",
    CALCULATE (
        SUM(FactInternetSales[SalesAmount]),
        INDEX(-1, ORDERBY(DimDate[MonthNumberOfYear]))
    )
)
ORDER BY [ProductKey], [MonthNumberOfYear]

```

다음 표를 반환합니다.

테이블 확장

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
214	1	5423.45	8047.7
214	2	4968.58	8047.7
214	3	5598.4	8047.7
214	4	5073.55	8047.7
214	5	5248.5	8047.7
214	6	7487.86	8047.7
214	7	7382.89	8047.7
214	8	6543.13	8047.7
214	9	6788.06	8047.7
214	10	6858.04	8047.7
214	11	8607.54	8047.7
214	12	8047.7	8047.7
217	1	5353.47	7767.78
217	2	4268.78	7767.78
217	3	5773.35	7767.78

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
217	4	5738.36	7767.78
217	5	6158.24	7767.78
217	6	6998	7767.78
217	7	5563.41	7767.78
217	8	5913.31	7767.78
217	9	5913.31	7767.78
217	10	6823.05	7767.78
217	11	6683.09	7767.78
217	12	7767.78	7767.78

예제 3 - 시각적 계산

다음 시각적 계산 DAX 쿼리는 다음과 같습니다.

DAX
SalesComparedToBeginningOfYear = [SalesAmount] - CALCULATE(SUM([SalesAmount]), INDEX(1, ROWS, HIGHESTPARENT))
SalesComparedToBeginningOfQuarter = [SalesAmount] - CALCULATE(SUM([SalesAmount]), INDEX(1, , -1))

각 월에

대해 포함되도록 테이블을 향상시킵니다. - 총 판매 금액;

- 각 연도의 첫 번째 달의 차이;

및 각 분기의 첫 번째 달의 차이.

아래 스크린샷은 시각적 행렬과 첫 번째 시각적 계산식을 보여줍니다.

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesComparedToBeginningOfYear	SalesComparedToBeginningOfQuarter
2011	1	1	\$323,743.2056	0.00	0.00
		2	\$437,372.4476	113,629.24	113,629.24
		3	\$514,918.2284	191,175.02	191,175.02
		Total	\$1,276,033.8816	0.00	0.00
	2	4	\$523,950.0258	200,206.82	0.00
		5	\$539,469.0004	215,725.79	15,518.97
		6	\$647,524.2114	323,781.01	123,574.19

관련 콘텐츠

[OFFSET](#)

[Orderby](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

KEEPFILTERS

아티클 • 2023. 10. 20.

CALCULATE 또는 CALCULATETABLE 함수를 평가하는 동안 필터가 적용되는 방식을 수정합니다.

구문

DAX

KEEPFILTERS(<expression>)

매개 변수

용어	정의
식	임의의 식입니다.

반환 값

값의 테이블입니다.

설명

- CALCULATE 및 CALCULATETABLE 함수 컨텍스트 내에서 KEEPFILTERS를 사용하여 해당 함수의 표준 동작을 재정의합니다.
- 기본적으로 CALCULATE 같은 함수의 필터 인수는 식을 계산하기 위한 컨텍스트로 사용되며, 이에 따라 CALCULATE의 필터 인수가 동일한 열에 대한 기존 필터를 모두 대체합니다. CALCULATE에 대한 필터 인수의 영향을 받는 새 컨텍스트는 필터 인수의 일부로 멘션 열의 기존 필터에만 영향을 줍니다. CALCULATE 또는 기타 관련 함수의 인수에 멘션 열이 아닌 다른 열에 대한 필터가 다시 적용되고 기본.
- KEEPFILTERS 함수를 사용하면 이 동작을 수정할 수 있습니다. KEEPFILTERS를 사용하는 경우 현재 컨텍스트의 모든 기존 필터가 필터 인수의 열과 비교되고 이러한 인수의 교집합은 식을 평가하기 위한 컨텍스트로 사용됩니다. 한 열에 대한 순 효과는 CALCULATE에 사용되는 필터 인수와 KEEPFILTER 함수의 인수에 있는 필터 둘 다 인수 집합이 모두 적용된다는 것입니다. 즉, CALCULATE 필터는 현재 컨텍스트를 대체하는 반면 KEEPFILTERS는 현재 컨텍스트에 필터를 추가합니다.

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

예시

다음 예제에서는 CALCULATE 또는 CALCULATETABLE 수식의 일부로 KEEPFILTERS 함수를 사용하는 방법을 보여 주는 몇 가지 일반적인 시나리오를 안내합니다.

처음 세 식은 비교에 사용할 간단한 데이터를 가져옵니다.

- 워싱턴 주에 대한 인터넷 판매.
- 워싱턴 주와 오리건 주에 대한 인터넷 판매(두 주 모두 결합).
- 워싱턴 주와 브리티시 컬럼비아 주에 대한 인터넷 판매(두 지역 모두 결합).

네 번째 식은 워싱턴과 오리건의 인터넷 판매를 계산하고 워싱턴과 브리티시 컬럼비아에 대한 필터가 적용됩니다.

다음 식은 워싱턴 및 오리건의 Internet Sales를 계산하지만 KEEPFILTERS를 사용합니다. 워싱턴과 브리티시 컬럼비아에 대한 필터는 이전 컨텍스트의 일부입니다.

DAX

```
EVALUATE ROW(
    "$$ in WA"
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
                )
    , "$$ in WA and OR"
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
                || 'Geography'[State Province Code]="OR"
                )
    , "$$ in WA and BC"
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
                || 'Geography'[State Province Code]="BC"
                )
    , "$$ in WA and OR ??"
    , CALCULATE(
        CALCULATE('Internet Sales'[Internet Total Sales]
                  , 'Geography'[State Province Code]="WA"
                  || 'Geography'[State Province Code]="OR"
                  )
        , 'Geography'[State Province Code]="WA"
        || 'Geography'[State Province Code]="BC"
        )
    , "$$ in WA !!"
    , CALCULATE(
```

```

    CALCULATE('Internet Sales'[Internet Total Sales]
        , KEEPFILTERS('Geography'[State Province Code]="WA"
            || 'Geography'[State Province Code]="OR"
        )
    )
    , 'Geography'[State Province Code]="WA"
    || 'Geography'[State Province Code]="BC"
)
)

```

이 식이 샘플 데이터베이스 AdventureWorks DW에 대해 평가되면 다음 결과를 가져옵니다.

열	값
[\$\$ in WA]	\$ 2,467,248.34
[WA 및 OR의 \$\$]	\$ 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

① 참고

위의 결과는 교육용으로 단일 행 대신 표에 서식이 지정되었습니다.

먼저 식을 검사합니다. **[\$\$ in WA and OR ??]**. 외부 CALCULATE 식에 워싱턴 및 브리티시 컬럼비아에 대한 필터가 포함되어 있으므로 이 수식이 워싱턴과 오리건에서 판매 값을 반환하는 방법을 궁금해할 수 있습니다. 대답은 CALCULATE의 기본 동작이 'Geography'[State Province Code]의 외부 필터를 재정의하고 필터가 동일한 열에 적용되므로 자체 필터 인수를 대체한다는 것입니다.

다음으로, 식 **[WA !!의 \$\$]**을 검사합니다. 인수 필터에 오리건이 포함되고 외부 CALCULATE 식에 워싱턴과 브리티시 컬럼비아의 필터가 포함되어 있기 때문에 이 수식이 워싱턴의 판매 값을 어떻게 반환할 수 있는지 궁금할 수 있습니다. 정답은 KEEPFILTERS가 CALCULATE의 기본 동작을 수정하고 추가 필터를 추가한다는 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". 두 필터가 모두 동일한 열에 적용되므로 결과 필터 'Geography'[State Province Code]="WA" 는 식을 평가할 때 적용되는 필터입니다.

참고 항목

[필터 함수](#)

[CALCULATE 함수](#)

[CALCULATETABLE 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LOOKUPVALUE

아티클 • 2023. 10. 20.

하나 이상의 검색 조건에 지정된 모든 조건을 충족하는 행의 값을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
result_columnName	반환하려는 값이 포함된 기존 열의 이름입니다. 식일 수 없습니다.
search_columnName	기존 열의 이름입니다. result_columnName 같은 테이블이나 관련 테이블에 있을 수 있습니다. 식일 수 없습니다. 여러 쌍의 search_columnName 및 search_value 지정할 수 있습니다.
search_value	search_columnName 검색할 값입니다. 여러 쌍의 search_columnName 및 search_value 지정할 수 있습니다.
alternateResult	(선택 사항) result_columnName 컨텍스트가 0개 이상의 고유 값으로 필터링되었을 때 반환되는 값입니다. 지정하지 않으면 result_columnName 0 값으로 필터링되거나 result_columnName 컨텍스트에 둘 이상의 고유 값이 있는 경우 오류가 발생할 때 이 함수는 BLANK를 반환합니다.

반환 값

search_columnName 및 search_value 모든 쌍이 정확히 일치하는 행의 result_columnName 값입니다.

모든 검색 값을 충족하는 일치 항목이 없으면 BLANK 또는 alternateResult (지정된 경우) 가 반환됩니다. 즉, 일부 조건만 일치하는 경우 함수는 조회 값을 반환하지 않습니다.

여러 행이 검색 값과 일치하고 이러한 행에 대한 resultColumnName 값이 동일하면 해당 값이 반환됩니다. 그러나 resultColumnName 다른 값을 반환하는 경우 오류 또는 alternateResult(지정된 경우)가 반환됩니다.

설명

- 결과 열이 포함된 테이블과 검색 열이 포함된 테이블 간에 관계가 있는 경우 대부분의 경우 LOOKUPVALUE 대신 RELATED 함수를 사용하는 것이 더 효율적이며 더 나은 성능을 제공합니다.
- 여러 쌍의 searchColumnName 및 search_value 지정할 수 있습니다.
- 함수가 검색 테이블의 행을 반복하기 전에 search_value 및 alternateResult 매개 변수가 평가됩니다.
- LOOKUPVALUE에서 반환된 오류를 캡처하려면 ISERROR 또는 IFERROR 함수를 사용하지 마십시오. 단일 출력 값을 확인할 수 없을 때 함수에 대한 일부 입력으로 인해 오류가 발생하는 경우 alternateResult 매개 변수를 제공하는 것이 오류를 처리하는 가장 안정적이고 성능이 뛰어난 방법입니다.
- power Pivot 계산 열에 지정된 경우 alternateResult 매개 변수는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

이 예제에서 LOOKUPVALUE는 주문이 수행된 날 주문에 대해 지불하는 데 사용되는 통화의 평균 비율을 검색하는 데 사용됩니다.

```
DAX  
Exchange Rate =  
LOOKUPVALUE (  
    'Currency Rate'[Average Rate],  
    'Currency Rate'[CurrencyKey], [CurrencyKey],  
    'Currency Rate'[DateKey], [OrderDateKey]  
)
```

주문 날짜와 통화는 모두 올바른 날짜와 통화에 대한 평균 요금을 찾는 데 필요합니다. OrderDateKey 및 CurrencyKey는 통화 비율 테이블에서 평균 금리를 조회하는 데 사용되는 키입니다.

환율을 사용하여 다음을 사용하여 현지 통화로 판매액을 계산할 수 있습니다.

DAX

```
Sales Amount Local Currency = [Sales Amount] * [Exchange Rate]
```

예제 2

이 예제에서 Sales 테이블에 정의된 다음 계산 열은 LOOKUPVALUE 함수를 사용하여 Sales Order 테이블에서 채널 값을 반환합니다.

DAX

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

그러나 이 경우 Sales Order와 Sales 테이블 사이에 관계가 있으므로 RELATED 함수를 사용하는 것이 더 효율적입니다.

DAX

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

참고 항목

[RELATED 함수\(DAX\)](#)

[정보 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MATCHBY

아티클 • 2023. 12. 14.

창 함수에서 데이터를 일치시키고 현재 행을 식별하는 방법을 결정하는 데 사용되는 열을 정의합니다.

구문

DAX

```
MATCHBY ( [<matchBy_columnName>[ , matchBy_columnName [ , ...]]] )
```

매개 변수

 테이블 확장

용어	정의
matchBy_columnName	(선택 사항) 창 함수 관계의 현재 행을 식별하는 데 사용할 기준 열의 이름입니다<>. RELATED()는 관계와> 관련된 테이블의 열을 참조하는 <데 사용할 수도 있습니다.

반환 값

이 함수는 값을 반환하지 않습니다.

설명

이 함수는 창 함수 식 내에서만 사용할 수 있습니다.

예시

오프셋을 [참조하세요](#).

관련 콘텐츠

ORDERBY, PARTITIONBY 및 MATCHBY 함수 이해

INDEX

OFFSET

Orderby

PARTITIONBY

WINDOW

RANK

ROWNUMBER

피드백

이 페이지가 도움이 되었나요?

Yes

No

MOVINGAVERAGE

아티클 • 2024. 02. 21.

시각적 계산 데이터 표의 지정된 축을 따라 계산된 이동 평균을 반환합니다. 즉, 마지막 <windowSize> 행에 대해 계산된 지정된 열의 평균입니다.

구문

DAX

```
MOVINGAVERAGE ( <column>, <windowSize>[, <includeCurrent>][, <axis>][, <blanks>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
column	각 행에 대한 값을 제공하는 열입니다.
windowSize	계산에 포함할 행 수입니다. 상수 값이어야 합니다.
includeCurrent	(선택 사항) 범위에 현재 행을 포함할지 여부를 지정하는 논리 값입니다. 기본값은 True입니다.
축	(선택 사항) 축 참조로, 이동 평균을 계산할 방향입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다.
reset	(선택 사항) 계산이 다시 설정되는지, 시각적 세이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. -0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

스칼라 값으로, 현재 행의 이동 평균입니다.

설명

이 함수는 시각적 계산에서만 사용할 수 있습니다.

<includeCurrent>, <축>, <<공백> 및 <재설정> 매개 변수는 생략할 수 있습니다.

예 1

각 제품 범주 및 달력 월의 총 판매량을 요약하는 테이블이 지정된 경우 다음 DAX 쿼리는 지난 6개월 동안 해당 범주에 대한 총 판매액의 평균을 가진 열을 추가합니다.

DAX

```
AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)
```

아래 스크린샷은 시각적 행렬 및 시각적 계산 식을 보여줍니다.

		AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)							
CalendarYear	MonthNumberOfYear	EnglishProductCategoryName	Accessories	Bikes	Clothing	Total SalesAmount	AvgSalesLast6 Months		
2012	11			\$513,987.7576	499,021.53	\$513,987.7576	499,021.53		
	12			\$595,333.0034	512,488.63	\$595,333.0034	512,488.63		
	Total			\$5,836,345.8175	6,344,417.60	\$5,836,345.8175	6,344,417.60		
2013	1	\$10,545.97	10,545.97	\$766,754.1686	561,259.44	\$4,775.28	4,775.28		
	2	\$36,924.35	23,735.16	\$739,000.87	598,749.66	\$16,696.66	10,735.97		
	3	\$54,689.48	34,053.27	\$933,134.99	674,486.47	\$26,341.8	15,937.91		
	4	\$52,204.71	38,591.13	\$930,705.79	746,486.10	\$25,222.8	18,259.14		
	5	\$54,600.84	41,793.07	\$1,033,128.45	833,009.55	\$25,598.1	19,726.93		
	6	\$62,495.09	45,243.41	\$1,512,801.23	985,920.92	\$28,925.62	21,260.04		
	7	\$58,364.86	53,213.22	\$1,342,684.25	1,081,909.26	\$30,661.88	25,574.48		
	8	\$60,752.56	57,184.59	\$1,455,104.25	1,201,259.83	\$29,639.91	27,731.69		
	9	\$57,717.29	57,689.23	\$1,347,106.88	1,270,255.14	\$27,614.56	27,943.81		

예제 2

동일한 테이블을 감안할 때 다음 DAX 쿼리는 이전 12개월 동안 해당 범주에 대한 총 판매액의 평균을 가진 열을 추가합니다(현재 월을 포함하지 않음).

DAX

```
AvgSalesPrev12Months = MOVINGAVERAGE([SalesAmount], 12, FALSE, Rows, KEEP)
```

참고 항목

[INDEX](#)

[Orderby](#)

PARTITIONBY

Runningsum

WINDOW

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

OFFSET

아티클 • 2024. 02. 21.

지정된 오프셋을 통해 동일한 테이블 내의 현재 행 앞이나 뒤 위치에 있는 단일 행을 반환합니다. 현재 행을 단일 행으로 추론할 수 없는 경우 여러 행이 반환될 수 있습니다.

구문

DAX

```
OFFSET ( <delta>[, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
delta	데이터를 가져올 현재 행 앞(음수 값) 또는 이후(양수 값) 행의 수입니다. 스칼라 값을 반환하는 DAX 식일 수 있습니다.
관계	(선택 사항) 출력 행이 반환되는 테이블 식입니다. 지정한 경우 <code>partitionBy</code> 의 모든 열은 <해당 열 또는 관련 테이블에서 나와야 합니다. 생략된 경우: <ul style="list-style-type: none">- <code><orderBy></code> 를 명시적으로 지정해야 합니다.- 모든 <code><orderBy></code> 및 <code><partitionBy></code> 식은 정규화된 열 이름이어야 하며 단일 테이블에서 가져옵니다.- <code>orderBy</code> 및 <code>partitionBy</code>의 모든 열에 대한 <code><ALLSELECTED()</code>로 기본값이 지정<됩니다.>
축	(선택 사항) 시각적 셰이프의 축입니다. 시각적 계산에서만 사용할 수 있으며 <code>관계를></code> 대체합니다<.
orderBy	(선택 사항) 각 파티션의 정렬 방법을 정의하는 식을 포함하는 <code>ORDERBY()</code> 절입니다. 생략된 경우: <ul style="list-style-type: none">- <관계를> 명시적으로 지정해야 합니다.- <code>partitionBy</code>에 아직 지정되지 않은 관계의> 모든 열<에 <따라 정렬하는 기본값입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다. 현재 지원되는 유일한 값은 <code>DEFAULT</code> 입니다. 여기서 숫자 값에 대한 동작은 빈 값이 0

용어	정의
	과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다.
partitionBy	(선택 사항) 관계가 > 분할되는 방법을 <정의하는 열을 포함하는 PARTITIONBY() 절입니다. 생략 <하면 관계가> 단일 파티션으로 처리됩니다.
matchBy	(선택 사항) 데이터를 일치시키고 현재 행을 식별하는 방법을 정의하는 열을 포함하는 MATCHBY() 절입니다.
reset	(선택 사항) 시각적 계산에서만 사용할 수 있습니다. 계산이 다시 설정되는지, 시각적 셰이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

관계>로부터 <하나 이상의 행>

설명

DAX 테이블 함수에 의해 추가된 열을 제외하고, matchBy>가 없는 경우 <관계의> 각 열 <또는 matchBy 및 <partitionBy>의 <각 열은 matchBy>>가 있는 경우 <다음 동작으로 작동할 현재 행을 정의하는 데 도움이 되는 해당 외부 값이 있어야 합니다.

- 정확히 하나의 해당 외부 열이 있는 경우 해당 값이 사용됩니다.
- 해당 외부 열이 없으면 다음을 수행합니다.
 - OFFSET은 먼저 해당 외부 열이 없는 모든 열을 결정합니다.
 - OFFSET의 부모 컨텍스트에서 이러한 열에 대한 기준 값의 모든 조합에 대해 OFFSET이 평가되고 행이 반환됩니다.
 - OFFSET의 최종 출력은 이러한 행의 공용 구조체입니다.
- 해당하는 외부 열이 두 개 이상 있으면 오류가 반환됩니다.

DAX 테이블 함수에서 <관계의> 모든 <열을 추가한 경우 오류가 반환됩니다.

matchBy>가 있는 경우 <OFFSET는 matchBy 및 <partitionBy>> 열을 사용하여 <행을 식별하려고 합니다.

`matchBy`>가 없고 `orderBy` 및 `<partitionBy>` 내에 `<지정된 열이 관계의>` 모든 행<을 고유하게 식별할 수 없는 경우 <다음을 수행합니다.

- `OFFSET`은 모든 행을 고유하게 식별하는 데 필요한 최소 개수의 추가 열을 찾으려고 합니다.
- 이러한 열을 찾을 수 있는 경우 `OFFSET`는 이러한 새 열을 `orderBy`>에 <자동으로 추가하고 각 파티션은 이 새 `OrderBy` 열 집합을 사용하여 정렬됩니다.
- 이러한 열을 찾을 수 없으면 오류가 반환됩니다.

다음과 같은 경우 빈 테이블이 반환됩니다.

- `OrderBy` 또는 `PartitionBy` 열의 해당 외부 값은 관계> 내에 <없습니다.
- `델타>` 값으로 <인해 파티션 내에 없는 행으로 이동됩니다.

`OFFSET`이 관계>와 <동일한 테이블에 <정의된 계산 열 내에서 사용되고 `orderBy`>를 생략하면 오류가 반환됩니다.

<초기화>는 시각적 계산에서만 사용할 수 있으며 `orderBy` 또는 `<partitionBy>`와 함께 <사용할 수 없습니다. <다시 설정>이 있으면 축>을 <지정할 수 있지만 <관계형>은 지정할 수 없습니다.

예제 1 - 계산 열

다음 DAX 쿼리:

```
DAX

DEFINE
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProductCategory[EnglishProductName],
        DimDate[CalendarYear],
        "CurrentYearSales", SUM(FactInternetSales[SalesAmount])
    )
EVALUATE
    ADDCOLUMNS (
        vRelation,
        "PreviousYearSales",
        SELECTCOLUMNS(
            OFFSET (
                -1,
                vRelation,
                ORDERBY([CalendarYear]),
                PARTITIONBY([EnglishProductName])
            ),
            [CurrentYearSales]
        )
    )
```

)

각 제품 범주 및 연도의 총 판매액과 이전 연도의 해당 범주에 대한 총 판매량을 요약하는 테이블을 반환합니다.

예제 2 - 측정값

다음 DAX 쿼리:

DAX

```
DEFINE
MEASURE DimProduct[CurrentYearSales] = SUM(FactInternetSales[SalesAmount])
MEASURE DimProduct[PreviousYearSales] =
CALCULATE(SUM(FactInternetSales[SalesAmount]), OFFSET(-1, ,
ORDERBY(DimDate[CalendarYear])))
EVALUATE
SUMMARIZECOLUMNS (
    DimDate[CalendarYear],
    "CurrentYearSales", DimProduct[CurrentYearSales],
    "PreviousYearSales", DimProduct[PreviousYearSales]
)
```

측정값에 OFFSET()을 사용하여 각 연도의 총 판매액과 이전 연도의 총 매출을 요약하는 테이블을 반환합니다.

예제 3 - 계산 열

다음 DAX 쿼리:

DAX

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

```
)  
)
```

SalesOrderNumber 및 SalesOrderLineNumber에서 현재 판매가 식별되는 판매 금액의 내림차순으로 각 판매의 이전 판매 금액을 나타내는 열을 추가하여 FactInternetSales 테이블을 반환합니다. MATCHBY가 없으면 FactInternetSales 테이블에 키 열이 없으므로 쿼리에서 오류를 반환합니다.

예제 4 - 시각적 계산

다음 시각적 계산 DAX 쿼리:

DAX

```
SalesRelativeToPreviousMonth = [SalesAmount] - CALCULATE(SUM([SalesAmount]),  
OFFSET(-1, ROWS, HIGHESTPARENT))
```

각 월과 같은 연도 내의 이전 월별 총 판매액의 차이를 반환합니다.

아래 스크린샷은 시각적 행렬 및 시각적 계산식을 보여줍니다.

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRelativeToPreviousMonth
2011	1	1	\$323,743.2056	323,743.21
		2	\$437,372.4476	113,629.24
		3	\$514,918.2284	77,545.78
	2	Total	\$1,276,033.8816	1,276,033.88
		4	\$523,950.0258	9,031.80
		5	\$539,469.0004	15,518.97
		6	\$647,524.214	108,055.21

관련 콘텐츠

[INDEX](#)

[Orderby](#)

[PARTITIONBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

이 페이지가 도움이 되었나요?

 Yes

 No

Orderby

아티클 • 2023. 10. 20.

각 창 함수의 파티션 내에서 정렬 순서를 결정하는 식을 정의합니다.

구문

DAX

```
ORDERBY ( [<orderBy_expression>[, <order>[, <orderBy_expression>[, <order>]] ...]] )
```

매개 변수

용어	정의
orderBy_expression	(선택 사항) 각 창 함수의 파티션 내에서 데이터를 정렬하는 데 사용할 스칼라 식입니다.
order	(선택 사항) "<OrderDirection> [<BlankHandling>]" 형식의 두 부분으로 구성된 값입니다. <OrderDirection>은 orderBy_expression> 값(즉, 오름차순 또는 내림차순)을 정렬하는 방법을 지정합니다. 유효한 값은 다음과 같습니다. DESC. 대체 값: 0(zero)/ FALSE . orderBy_expression> 값의 내림차순으로 <정렬합니다. ASC. 대체 값: 1/ TRUE . orderBy_expression> 값의 오름차순으로 <정렬합니다. 순서>를 생략하면 <기본값입니다. <BlankHandling> 부분은 선택 사항입니다. 공백의 순서를 지정합니다. 유효한 값은 다음과 같습니다. BLANKS 기본값입니다. 기본값입니다. 숫자 값의 동작은 빈 값이 0과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다. 먼저 공백을 지정합니다. 공백은 오름차순 또는 내림차순 정렬 순서에 관계 없이 항상 시작 부분에 정렬됩니다. 공백이 마지막으로 표시됩니다. 공백은 오름차순 또는 내림차순 정렬 순서에 관계없이 항상 끝에 정렬됩니다.

반환 값

이 함수는 값을 반환하지 않습니다.

설명

이 함수는 창 함수 식 내에서만 사용할 수 있습니다.

예시

오프셋을 [참조하세요](#).

참고 항목

[ORDERBY](#), [PARTITIONBY](#) 및 [MATCHBY](#) 함수 이해

[INDEX](#)

[OFFSET](#)

[PARTITIONBY](#)

[MATCHBY](#)

[WINDOW](#)

[RANK](#)

[ROWNUMBER](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PARTITIONBY

아티클 • 2023. 12. 14.

창 함수 <의 관계> 매개 변수를 분할하는 데 사용되는 열을 정의합니다.

구문

DAX

```
PARTITIONBY ( [<partitionBy_columnName>[, partitionBy_columnName [, ...]]] )
```

매개 변수

 테이블 확장

용어	정의
partitionBy_columnName	(선택 사항) 창 함수의 관계를 > 분할하는 데 사용할 기준 열의 이름입니다<. RELATED()는 관계와 > 관련된 테이블의 열을 참조하는 <데 사용할 수도 있습니다.

반환 값

이 함수는 값을 반환하지 않습니다.

설명

이 함수는 창 함수 식 내에서만 사용할 수 있습니다.

예시

오프셋을 [참조하세요](#).

관련 콘텐츠

[ORDERBY, PARTITIONBY 및 MATCHBY 함수 이해](#)

[INDEX](#)

OFFSET

Orderby

MATCHBY

WINDOW

RANK

ROWNUMBER

피드백

이 페이지가 도움이 되었나요?

Yes

No

RANGE

아티클 • 2024. 02. 26.

현재 행을 기준으로 지정된 축 내의 행 간격을 반환합니다. 이 간격은 현재 행 앞의 마지막 <단계> 행 또는 현재 행 뒤의 첫 번째 <단계> 행으로 구성됩니다.

구문

DAX

```
RANGE ( <step>[, <includeCurrent>][, <axis>][, <blanks>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
step	범위에 포함할 현재 행 앞(음수 값) 또는 이후(양수 값)의 행 수입니다. 상수 값이어야 합니다. - 음수이면 창에 현재 행 앞의 마지막 단계 행이 포함됩니다. - 그렇지 않으면 창에 현재 행 뒤의 첫 번째 단계 행이 포함됩니다.
includeCurrent	(선택 사항) 범위에 현재 행을 포함할지 여부를 지정하는 논리 값입니다. 기본값은 True입니다.
축	(선택 사항) 축 참조, 간격을 만들 방향입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다.
reset	(선택 사항) 계산이 다시 설정되는지, 시각적 세이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

데이터 행의 간격입니다.

설명

이 함수는 시각적 계산에서만 사용할 수 있습니다.

<includeCurrent>, <축>, <<공백> 및 <재설정> 매개 변수는 생략할 수 있습니다.

예 1

각 제품 범주 및 월의 총 판매량을 요약하는 테이블이 지정된 경우 다음 DAX 쿼리는 지난 6개월 동안의 총 판매액이 포함된 열을 추가합니다.

DAX									
TotalSalesLast6Months = CALCULATE(SUM([SalesAmount]), RANGE(-5, Rows))									

아래 스크린샷은 시각적 행렬 및 시각적 계산 식을 보여줍니다.

X ✓ fx 1 TotalSalesLast6Months = CALCULATE(SUM([SalesAmount]), RANGE(-5, Rows))									
CalendarYear	EnglishProductName	CategoryName	Accessories	Bikes	Clothing	Total Sales	Total Sales		
	MonthNumber	Year	SalesAmount	TotalSalesLast6 Months	SalesAmount	TotalSalesLast6 Months	SalesAmount	TotalSalesLast6 Months	
2012	11			\$513,987.7576	2,994,129.18		\$513,987.7576	2,994,129.18	
	12			\$595,333.0034	3,074,931.78		\$595,333.0034	3,074,931.78	
	Total			\$5,836,345.8175	12,688,835.20		\$5,836,345.8175	12,688,835.20	
2013	1		\$10,545.97	10,545.97	\$766,754.1686	3,367,556.63	\$4,775.28	4,775.28	\$782,075.4186
	2		\$36,924.35	47,470.32	\$739,000.87	3,592,497.96	\$16,696.66	21,471.94	\$792,621.88
	3		\$54,689.48	102,159.80	\$933,134.99	4,046,918.83	\$26,341.8	47,813.74	\$1,014,166.27
	4		\$52,204.71	154,364.51	\$930,705.79	4,478,916.58	\$25,222.8	73,036.54	\$1,008,133.3
	5		\$54,600.84	208,965.35	\$1,033,128.45	4,998,057.27	\$25,598.1	98,634.64	\$1,113,327.39
	6		\$62,495.09	271,460.44	\$1,512,801.23	5,915,525.50	\$28,925.62	127,560.26	\$1,604,221.94
	7		\$58,364.86	319,279.33	\$1,342,684.25	6,491,455.58	\$30,661.88	153,446.86	\$1,431,710.99
	8		\$60,752.56	343,107.54	\$1,455,104.25	7,207,558.96	\$29,639.91	166,390.11	\$1,545,496.72
	9		\$57,717.29	346,135.35	\$1,347,106.88	7,621,530.85	\$27,614.56	167,662.87	\$1,432,438.73
									8,135,329.07

예제 2

동일한 테이블이 있는 경우 다음 DAX 쿼리는 현재 월을 포함하지 않고 다음 12개월 동안 총 판매액이 있는 열을 추가합니다.

DAX									
TotalSalesFollowingYear = CALCULATE(SUM([SalesAmount]), RANGE(12, FALSE, Rows, KEEP))									

참고 항목

INDEX

Orderby

PARTITIONBY

WINDOW

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RANK

아티클 • 2024. 02. 26.

지정된 순서에 따라 정렬된 지정된 파티션 내의 현재 컨텍스트에 대한 순위를 반환합니다. 일치 항목을 찾을 수 없으면 순위가 비어 있습니다.

구문

DAX

```
RANK ( [<ties>][, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
ties	(선택 사항) 둘 이상의 행이 연결된 경우 순위를 처리하는 방법을 정의합니다. 지정한 경우 지원되는 값은 DENSE 또는 SKIP입니다. 생략된 경우: <ul style="list-style-type: none">- 기본값은 SKIP입니다.
관계	(선택 사항) 출력 행이 반환되는 테이블 식입니다. 지정된 경우 orderBy 및 <partitionBy>의 <모든 열이 해당 열에서 와야 합니다.> 생략된 경우: <ul style="list-style-type: none">- <orderBy> 를 명시적으로 지정해야 합니다.- 모든 <orderBy> 및 <partitionBy> 열은 정규화되어야 하며 단일 테이블에서 가져옵니다.- orderBy 및 partitionBy 의 모든 열에 대한 <ALLSELECTED()로 기본값이 지정<됩니다.>
축	(선택 사항) 시각적 셰이프의 축입니다. 시각적 계산에서만 사용할 수 있으며 관계를 대체합니다<.
orderBy	(선택 사항) 각 파티션의 정렬 방법을 정의하는 열을 포함하는 ORDERBY() 절입니다. 생략된 경우: <ul style="list-style-type: none">- <관계를> 명시적으로 지정해야 합니다.- partitionBy에 아직 지정되지 않은 관계의> 모든 열<에 <따라 정렬하는 기본값입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 지원되는 값은 다음과 같습니다.

용어	정의
	<ul style="list-style-type: none"> 숫자 값에 대한 동작이 빈 값인 KEEP(기본값)은 0과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다. 첫째, 공백은 오름차순 또는 내림차순 정렬 순서에 관계없이 항상 시작 부분에 정렬됩니다. 마지막으로, 공백은 오름차순 또는 내림차순 정렬 순서에 관계없이 항상 끝에 정렬됩니다.
	<p><개별 식의> ORDERBY() 함수에서 공백 매개 변수와 공백을 모두 지정<하면 개별 orderBy 식의 공백>이 관련 orderBy 식에 우선 순위를 두고 공백이 지정되지 않은> <orderBy 식은 부모 Window 함수의 공백 매개 변수를 적용<합니다>.</p>
partitionBy	(선택 사항) 관계가> 분할되는 방법을 <정의하는 열을 포함하는 PARTITIONBY() 절입니다. 생략 <하면 관계가> 단일 파티션으로 처리됩니다.
matchBy	(선택 사항) 데이터를 일치시키고 현재 행을 식별하는 방법을 정의하는 열을 포함하는 MATCHBY() 절입니다.
reset	(선택 사항) 시각적 계산에서만 사용할 수 있습니다. 계산이 다시 설정되는지, 시각적 셰이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

현재 컨텍스트의 순위 번호입니다.

설명

- 각 <orderBy>, <partitionBy 및 <matchBy>> 열에는 다음 동작을 사용하여 작동할 현재 행을 정의하는 데 도움이 되는 해당 외부 값이 있어야 합니다.
 - 정확히 하나의 해당 외부 열이 있는 경우 해당 값이 사용됩니다.
 - 해당 외부 열이 없으면 다음을 수행합니다.
 - RANK는 먼저 해당 외부 열이 없는 모든 <orderBy>, <partitionBy> 및 <matchBy> 열을 결정합니다.
 - RANK 부모 컨텍스트에서 이러한 열에 대한 기준 값의 모든 조합에 대해 RANK가 평가되고 행이 반환됩니다.

- RANK의 최종 출력은 순위 번호입니다.
- matchBy>가 있는 경우 <RANK는 matchBy 및 <partitionBy>>의 <열을 사용하여 현재 행을 확인합니다.
- orderBy 및 <partitionBy> 내에 <지정된 열이 관계의> 모든 행<을 고유하게 식별 할 수 없는 경우 두 개 이상의 행의 순위가 같을 수 있으며 순위는 ties 매개 변수에 의해 결정됩니다.
- RANK는 총 행에 대해 빈 값을 반환합니다. 식을 철저히 테스트하는 것이 좋습니다.
- SUM이 SUMX와 비교하여 RANKX와 비교되지 않습니다.
- <초기화>는 시각적 계산에서만 사용할 수 있으며 orderBy 또는 <partitionBy>와 함께 <사용할 수 없습니다. <다시 설정>이 있으면 축>을 <지정할 수 있지만 <관계 형>은 지정할 수 없습니다.

예제 1 - 계산 열

다음 DAX 쿼리:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "Rank",
    RANK(
        DENSE,
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        LAST,
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName])))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]
asc
```

StateProvinceName 및 City에서 동일한 EnglishCountryRegionName을 사용하여 각 지리의 순위를 지정하는 테이블을 반환합니다. 빈 <orderBy> 열 값은 끝에 정렬됩니다.

예제 2 - 시각적 계산

다음 시각적 계산 DAX 쿼리는 다음과 같습니다.

DAX

```
SalesRankWithinYear = RANK(DENSE, ORDERBY([SalesAmount], DESC),  
PARTITIONBY([CalendarYear]))
```

```
SalesRankAllHistory = RANK(DENSE, ORDERBY([SalesAmount], DESC))
```

각 연도 내의 총 판매액과 전체 기록을 기준으로 매월 순위를 매기는 두 개의 열을 만듭니다.

아래 스크린샷은 시각적 행렬과 첫 번째 시각적 계산식을 보여줍니다.

X ✓ fx 1 SalesRankWithinYear = RANK(DENSE, ORDERBY([SalesAmount], DESC), PARTITIONBY([CalendarYear]))

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory
2011	1	1	\$323,743.2056	12.00	37.00
		2	\$437,372.4476	11.00	34.00
		3	\$514,918.2284	10.00	25.00
		Total	\$1,276,033.8816	4.00	11.00
	2	4	\$523,950.0258	9.00	24.00
		5	\$539,469.0004	8.00	23.00
		6	\$647,524.214	4.00	16.00
		Total	\$1,710,943.2402	3.00	7.00
	3	7	\$648,343.8132	3.00	15.00
		8	\$599,874.2732	7.00	20.00
		9	\$609,121.865	6.00	19.00
		Total	\$1,857,339.9514	2.00	6.00
	4	10	\$655,170.3294	2.00	14.00
		11	\$716,748.7178	1.00	13.00
		12	\$636,253.2642	5.00	17.00
		Total	\$2,008,172.3114	1.00	5.00
Total			\$6,852,489.3846	1.00	2.00

관련 콘텐츠

[INDEX](#)

[Orderby](#)

[PARTITIONBY](#)

[WINDOW](#)

[ROWNUMBER](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

REMOVEFILTERS

아티클 • 2023. 10. 20.

지정된 테이블 또는 열에서 필터를 지웁니다.

구문

DAX

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

매개 변수

용어	정의
table	필터를 지울 테이블입니다.
column	필터를 지울 열입니다.

반환 값

해당 없음. 설명을 참조하세요.

설명

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

예 1

DAX 쿼리

DAX

```
DEFINE  
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])  
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],  
CALCULATE([TotalSales],REMOVEFILTERS()))  
  
EVALUATE  
SUMMARIZECOLUMNS(  
ROLLUPADDISSUBTOTAL(DimProductCategory[EnglishProductName],
```

```

    "IsGrandTotal"),
    "TotalSales", [TotalSales],
    "%Sales", [%Sales]
)
ORDER BY
[IsGrandTotal] DESC, [TotalSales] DESC

```

반환

DimProductCategory[EnglishProductCategoryName]	[IsGrandTotal]	[TotalSales]	[%Sales]
Row1	True	29358677.2207	1
Bikes	False	28318144.6507	0.964557920570538
Accessories	False	700759.96	0.023868921434441
의류	False	339772.61	0.0115731579950215

예제 2

DAX 쿼리

```

DAX

DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategory
Name])))
EVALUATE
SUMMARIZECOLUMNS(
    DimProductCategory[EnglishProductCategoryName],
    DimProductSubcategory[EnglishProductSubcategoryName],
    "TotalSales", [TotalSales],
    "%Sales", [%Sales]
)
ORDER BY
    DimProductCategory[EnglishProductCategoryName] ASC,
    DimProductSubcategory[EnglishProductSubcategoryName] ASC

```

반환

DimProductCategory [EnglishProductCategoryName]	DimProductSubcategory [EnglishProductSubcategoryName]	[TotalSales]	[%Sales]
Accessories	자전거 랙	39360	0.05616759
Accessories	자전거 스탠드	39591	0.05649723
Accessories	물병 및 거치대	56798.19	0.08105228

DimProductCategory [EnglishProductCategoryName]	DimProductSubcategory [EnglishProductSubcategoryName]	[TotalSales]	[%Sales]
Accessories	Cleaners	7218.6	0.0103011
Accessories	Fenders	46619.58	0.06652717
Accessories	Helmets	225335.6	0.3215589
Accessories	Hydration Packs	40307.67	0.05751994
Accessories	Tires and Tubes	245529.32	0.35037578
Bikes	산악용 자전거	9952759.564	0.35146228
Bikes	도로용 자전거	14520584.04	0.51276608
Bikes	Touring Bikes	3844801.05	0.13577164
의류	Caps	19688.1	0.05794493
의류	장갑	35020.7	0.10307099
의류	Jerseys	172950.68	0.5090189
의류	반바지	71319.81	0.20990453
의류	양말	5106.32	0.01502864
의류	Vests	35687	0.10503201

피드백

이 페이지가 도움이 되었나요?

Yes

No

ROWNUMBER

아티클 • 2024. 02. 23.

지정된 순서에 따라 정렬된 지정된 파티션 내의 현재 컨텍스트에 대한 고유 순위를 반환합니다. 일치 항목을 찾을 수 없으면 rownumber가 비어 있습니다.

구문

DAX

```
ROWNUMBER ( [<relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>]  
[,<matchBy>][, <reset>] )
```

매개 변수

 테이블 확장

용어	정의
관계	(선택 사항) 출력 행이 반환되는 테이블 식입니다. 지정된 경우 orderBy 및 <partitionBy>의 <모든 열이 해당 열에서 와야 합니다. 생략된 경우: <ul style="list-style-type: none">- <orderBy> 를 명시적으로 지정해야 합니다.- 모든 <orderBy> 및 <partitionBy> 열은 정규화되어야 하며 단일 테이블에서 가져옵니다.- orderBy 및 partitionBy의 모든 열에 대한 <ALLSELECTED()로 기본값이 지정<됩니다.>
축	(선택 사항) 시각적 셰이프의 축입니다. 시각적 계산에서만 사용할 수 있으며 관계를 대체합니다.<
orderBy	(선택 사항) 각 파티션의 정렬 방법을 정의하는 열을 포함하는 ORDERBY() 절입니다. 생략된 경우: <ul style="list-style-type: none">- <관계를> 명시적으로 지정해야 합니다.- partitionBy에 아직 지정되지 않은 관계의> 모든 열<에 <따라 정렬하는 기본값입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 지원되는 값은 다음과 같습니다. <ul style="list-style-type: none">• DEFAULT(기본값)입니다. 여기서 숫자 값에 대한 동작은 빈 값이며 0과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다.• 첫째, 공백은 오름차순 또는 내림차순 정렬 순서에 관계없이 항상 시작 부분에 정렬됩니다.

용어	정의
	<ul style="list-style-type: none"> 마지막으로, 공백은 오름차순 또는 내림차순 정렬 순서에 관계없이 항상 끝에 정렬됩니다. <p><개별 식의> ORDERBY() 함수에서 공백 매개 변수와 공백을 모두 지정<하면 개별 orderBy 식의 공백>이 관련 orderBy 식에 우선 순위를 두고 공백이 지정되지 않은> <orderBy 식은 부모 Window 함수의 공백 매개 변수를 적용<합니다>.</p>
partitionBy	(선택 사항) 관계가> 분할되는 방법을 <정의하는 열을 포함하는 PARTITIONBY() 절입니다. 생략 <하면 관계가> 단일 파티션으로 처리됩니다.
matchBy	(선택 사항) 데이터를 일치시키고 현재 행을 식별하는 방법을 정의하는 열을 포함하는 MATCHBY() 절입니다.
reset	(선택 사항) 시각적 계산에서만 사용할 수 있습니다. 계산이 다시 설정되는지, 시각적 셰이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. <ul style="list-style-type: none"> - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

현재 컨텍스트의 행 번호입니다.

설명

각 <orderBy>, <partitionBy 및 <matchBy>> 열에는 다음 동작을 사용하여 작동할 현재 행을 정의하는 데 도움이 되는 해당 외부 값이 있어야 합니다.

- 정확히 하나의 해당 외부 열이 있는 경우 해당 값이 사용됩니다.
- 해당 외부 열이 없으면 다음을 수행합니다.
 - ROWNUMBER는 먼저 해당 외부 열이 없는 모든 <orderBy>, <partitionBy> 및 <matchBy> 열을 결정합니다.
 - ROWNUMBER 부모 컨텍스트에서 이러한 열에 대한 기존 값의 모든 조합에 대해 ROWNUMBER가 평가되고 행이 반환됩니다.
 - ROWNUMBER의 최종 출력은 이러한 행의 통합입니다.
- 해당하는 외부 열이 두 개 이상 있으면 오류가 반환됩니다.

`matchBy`>가 있는 경우 `<ROUNNUMBER>`는 `matchBy` 및 `<partitionBy>`>의 <열을 사용하여 현재 행을 압축합니다.

`orderBy` 및 `<partitionBy>`> 내에 <지정된 열이 관계의> 모든 행을 <고유하게 식별할 수 없는 경우 다음을 수행합니다.

- `ROUNNUMBER`는 모든 행을 고유하게 식별하는 데 필요한 최소 개수의 추가 열을 찾으려고 합니다.
- 이러한 열을 찾을 수 있는 경우 `ROUNNUMBER`는
 - 모든 행을 고유하게 식별하는 데 필요한 최소 개수의 추가 열을 찾습니다.
 - 이러한 새 열을 `orderBy` 절에 <자동으로 추가합니다.
 - 이 새 `orderBy` 열 집합을 사용하여 각 파티션을 정렬합니다.
- 이러한 열을 찾을 수 없고 함수가 런타임에 동률을 감지하면 오류가 반환됩니다.

<초기화>는 시각적 계산에서만 사용할 수 있으며 `orderBy` 또는 `<partitionBy>`>와 함께 <사용할 수 없습니다. <다시 설정>이 있으면 <지정할 수 있지만 <관계형>은 지정할 수 없습니다.

예제 1 - 계산 열

다음 DAX 쿼리:

```
DAX

EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "UniqueRank",
    ROUNNUMBER(
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName])))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City]
asc
```

`StateProvinceName` 및 `City`에서 동일한 `EnglishCountryRegionName`을 사용하여 각 지리의 순위를 고유하게 지정하는 테이블을 반환합니다.

예제 2 - 시각적 계산

다음 시각적 계산 DAX 쿼리는 다음과 같습니다.

DAX

```
SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC),  
PARTITIONBY([CalendarYear]))  
  
SalesRankAllHistory = ROWNUMBER(ORDERBY([SalesAmount], DESC))
```

각 연도 및 전체 기록에서 총 판매액별로 매월 고유하게 순위를 지정하는 두 개의 열을 만듭니다.

아래 스크린샷은 시각적 행렬과 첫 번째 시각적 계산식을 보여줍니다.

X ✓ fx 1 SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC), PARTITIONBY([CalendarYear]))

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory
2011	1	1	\$323,743.2056	12.00	37.00
		2	\$437,372.4476	11.00	34.00
		3	\$514,918.2284	10.00	25.00
		Total	\$1,276,033.8816	4.00	11.00
	2	4	\$523,950.0258	9.00	24.00
		5	\$539,469.0004	8.00	23.00
		6	\$647,524.214	4.00	16.00
		Total	\$1,710,943.2402	3.00	7.00
	3	7	\$648,343.8132	3.00	15.00
		8	\$599,874.2732	7.00	20.00
		9	\$609,121.865	6.00	19.00
		Total	\$1,857,339.9514	2.00	6.00
	4	10	\$655,170.3294	2.00	14.00
		11	\$716,748.7178	1.00	13.00
		12	\$636,253.2642	5.00	17.00
		Total	\$2,008,172.3114	1.00	5.00
Total			\$6,852,489.3846	1.00	2.00

관련 콘텐츠

[INDEX](#)

[Orderby](#)

[PARTITIONBY](#)

[WINDOW](#)

[RANK](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

Runningsum

아티클 • 2024. 02. 24.

시각적 계산 데이터 표의 지정된 축을 따라 계산된 실행 합계를 반환합니다. 즉, 현재 행까지 모든 행에 대해 계산된 지정된 열의 합계입니다.

구문

DAX

```
RUNNINGSUM ( <column>[ , <axis>][ , <blanks>][ , <reset>] )
```

매개 변수

 테이블 확장

용어	정의
column	각 행에 대한 값을 제공하는 열입니다.
축	(선택 사항) 축 참조로, 실행 합계를 계산할 방향입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다.
reset	(선택 사항) 계산이 다시 설정되는지, 시각적 세이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호 에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당 합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

스칼라 값으로, 현재 행까지의 실행 합계입니다.

설명

이 함수는 시각적 계산에서만 사용할 수 있습니다.

<축>, <공백> 및 <다시 설정> 매개 변수를 생략할 수 있습니다.

예시

각 제품 범주 및 연도의 총 판매량을 요약하는 테이블이 지정된 경우 다음 DAX 쿼리는 해당 범주의 총 판매액이 지정된 연도까지 열을 추가합니다.

DAX
SalesUpToDate = RUNNINGSUM([SalesAmount], Rows)

아래 스크린샷은 시각적 행렬 및 시각적 계산 식을 보여줍니다.

	EnglishProductCategoryName	Accessories	Bikes	Clothing	Total	
CalendarYear		SalesAmount	SalesUpToDate	SalesAmount	SalesUpToDate	SalesAmount
2011			\$6,852,489.3846	6,852,489.38		\$6,852,489.3846
2012			\$5,836,345.8175	12,688,835.20		\$5,836,345.8175
2013		\$647,604.83	647,604.83	\$15,085,083.2586	27,773,918.46	\$312,059.21
2014		\$53,155.13	700,759.96	\$544,226.19	28,318,144.65	\$27,713.4
Total		\$700,759.96	700,759.96	\$28,318,144.6507	28,318,144.65	\$339,772.61

참고 항목

[INDEX](#)

[MOVINGAVERAGE](#)

[Orderby](#)

[PARTITIONBY](#)

[WINDOW](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

SELECTEDVALUE

아티클 • 2023. 10. 20.

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 값을 반환합니다. 그렇지 않으면 alternateResult를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.
alternateResult	(선택 사항) columnName의 컨텍스트가 0개 이상의 고유 값으로 필터링되었을 때 반환되는 값입니다. 제공되지 않으면 기본값은 BLANK()입니다.

반환 값

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우의 값입니다. 그렇지 않으면 alternateResult입니다.

설명

- 에 해당하는 `SELECTEDVALUE(<columnName>, <alternateResult>)` 식은 .입니다
`IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>).`
- SELECTEDVALUE를 사용하는 경우 모범 사례에 대한 자세한 내용은 [DAX의 값 대신 SELECTEDVALUE 사용을 참조하세요.](#)
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

DAX

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

다음을 반환합니다.

DimProduct[색]	[Is Total]	[선택한 색]
검정	FALSE	검정
파랑	FALSE	파랑
회색	FALSE	회색
다색	FALSE	다색
해당 없음	FALSE	해당 없음
빨간색	FALSE	빨간색
은	FALSE	은
은색/검정색	FALSE	은색/검정색
백인	FALSE	흰색
노란색	FALSE	노란색
	TRUE	단일 선택 없음

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

WINDOW

아티클 • 2024. 02. 23.

지정된 간격 내에 배치되는 여러 행을 반환합니다.

구문

DAX

```
WINDOW ( from[, from_type], to[, to_type][, <relation> or <axis>][,
<orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

매개 변수

[+] 테이블 확장

용어	정의
보낸 사람	창이 시작되는 위치를 나타냅니다. 스칼라 값을 반환하는 DAX 식일 수 있습니다. 동작은 <code>from_type</code> 매개 변수 에 <따라 달라집니다. - <code>from_type</code> REL인 경우 <창의 첫 번째 행을 가져오기 위해 현재 행에서 돌아갈 행 수(음수 값) 또는 전달(양수 값)입니다. - <code>from_type</code> ABS이고 <원본>이 양수이면 <파티션 시작부터 창 시작 위치입니다. 인덱싱은 1부터 시작하는 것입니다. 예를 들어 1은 파티션의 시작 부분에서 창이 시작됩니다. <code>from></code> 이 음수이면 <파티션의 끝에서 창의 시작 위치입니다. -1은 파티션의 마지막 행을 의미합니다.
<code>from_type</code>	<code>from></code> 매개 변수의 동작을 <수정합니다>. 가능한 값은 ABS(절대) 및 REL(상대)입니다. 기본값은 REL입니다.
끝	<원본>과 동일<하지만 창의 끝을 나타냅니다. 마지막 행이 창에 포함됩니다.
<code>to_type</code>	<code>from_type</code> 동일<하지만 동작<을 수정합니다>.>
관계	(선택 사항) 출력 행이 반환되는 테이블 식입니다. 지정한 경우 <code>partitionBy</code> >의 모든 열은 <해당 열 또는 관련 테이블에서 나와야 합니다. 생략된 경우: - <orderBy> 를 명시적으로 지정해야 합니다. - 모든 <orderBy> 및 <partitionBy> 식은 정규화된 열 이름이어야 하며 단일 테이블에서 가져옵니다. - <code>orderBy</code> 및 <code>partitionBy</code> >의 모든 열에 대한 <ALLSELECTED()로 기본값이 지정<됩니다>.

용어	정의
축	(선택 사항) 시각적 셰이프의 축입니다. 시각적 계산에서만 사용할 수 있으며 관계를 > 대체합니다 <.
orderBy	(선택 사항) 각 파티션의 정렬 방법을 정의하는 식을 포함하는 ORDERBY() 절입니다. 생략된 경우: - <관계를> 명시적으로 지정해야 합니다. - partitionBy에 아직 지정되지 않은 관계의 > 모든 열<에 <따라 정렬하는 기본값입니다.
공백	(선택 사항) 정렬할 때 빈 값을 처리하는 방법을 정의하는 열거형입니다. 이 매개 변수는 나중에 사용하도록 예약되어 있습니다. 현재 지원되는 유일한 값은 DEFAULT입니다. 여기서 숫자 값에 대한 동작은 빈 값이 0과 음수 값 사이에 정렬됩니다. 문자열에 대한 동작은 빈 문자열을 포함하여 모든 문자열 앞에 빈 값이 정렬됩니다.
partitionBy	(선택 사항) 관계가 > 분할되는 방법을 <정의하는 열을 포함하는 PARTITIONBY() 절입니다. 생략 <하면 관계가> 단일 파티션으로 처리됩니다.
matchBy	(선택 사항) 데이터를 일치시키고 현재 행을 식별하는 방법을 정의하는 열을 포함하는 MATCHBY() 절입니다.
reset	(선택 사항) 시각적 계산에서만 사용할 수 있습니다. 계산이 다시 설정되는지, 시각적 셰이프의 열 계층 구조 수준을 나타내는지 나타냅니다. 허용되는 값은 NONE, LOWESTPARENT, HIGHESTPARENT 또는 정수입니다. 동작은 정수 기호에 따라 달라집니다. - 0이거나 생략하면 계산이 다시 설정되지 않습니다. NONE에 해당합니다. - 양수이면 정수는 곡물과 관계없이 가장 높은 열부터 시작하는 열을 식별합니다. HIGHESTPARENT는 1과 같습니다. - 음수이면 정수는 현재 곡물을 기준으로 가장 낮은 열부터 시작하는 열을 식별합니다. LOWESTPARENT는 -1과 같습니다.

반환 값

창의 모든 행입니다.

설명

DAX 테이블 함수에 의해 추가된 열을 제외하고 matchBy가 없는 경우 <관계의> 각 열 <또는 matchBy 및 <partitionBy>의 <각 열은 matchBy>>가 있는 경우 <작동할 현재 행을 정의하는 데 도움이 되는 해당 외부 값이 있어야 합니다. from_type 및 to_type 모두 값 ABS가 있는 경우 <다음이 partitionBy> 열에<만 적용됩니다.><>

- 정확히 하나의 해당 외부 열이 있는 경우 해당 값이 사용됩니다.
- 해당 외부 열이 없는 경우:

- WINDOW는 먼저 해당 외부 열이 없는 모든 열을 결정합니다.
- WINDOW의 부모 컨텍스트에서 이러한 열에 대한 기존 값의 모든 조합에 대해 WINDOW가 평가되고 해당 행이 반환됩니다.
- WINDOW 최종 출력은 이러한 행의 통합입니다.
- 해당하는 외부 열이 두 개 이상 있으면 오류가 반환됩니다.

DAX 테이블 함수에서 관계의 > 모든 <열을 추가한 경우 오류가 반환됩니다.

`matchBy`>가 있는 경우 <WINDOW는 `matchBy` 및 <`partitionBy`>> 열을 사용하여 <행을 식별하려고 합니다.

`matchBy`>가 없고 `orderBy` 및 <`partitionBy`>> 내에 <지정된 열이 관계의> 모든 행을 <고유하게 식별할 수 없는 경우 <다음을 수행합니다.

- WINDOW는 모든 행을 고유하게 식별하는 데 필요한 최소 개수의 추가 열을 찾으려고 합니다.
- 이러한 열을 찾을 수 있는 경우 WINDOW는 이러한 새 열을 `orderBy`>에 <자동으로 추가하고 각 파티션은 이 새 `orderBy` 열 집합을 사용하여 정렬됩니다.
- 이러한 열을 찾을 수 없으면 오류가 반환됩니다.

다음과 같은 경우 빈 테이블이 반환됩니다.

- `orderBy` 또는 <`partitionBy`>> 열의 <해당 외부 값이 관계> 내에 <없습니다.
- 전체 창이 파티션 외부에 있거나 창의 시작이 끝 이후입니다.

WINDOW가 관계와 >< 동일한 테이블에 <정의된 계산 열 내에서 사용되고 `orderBy`>를 생략하면 오류가 반환됩니다.

창의 시작이 첫 번째 행 앞에 있으면 첫 번째 행으로 설정됩니다. 마찬가지로 창의 끝이 파티션의 마지막 행 뒤인 경우 마지막 행으로 설정됩니다.

<초기화>는 시각적 계산에서만 사용할 수 있으며 `orderBy` 또는 <`partitionBy`>>와 함께 <사용할 수 없습니다. <다시 설정>이 있으면 <축>을 <지정할 수 있지만 <관계형>은 지정할 수 없습니다.

예제 1 - 측정값

다음 측정값은 다음과 같습니다.

DAX

```
3-day Average Price =
AVERAGEX(
    WINDOW(
        -2,REL,0,REL,
        SUMMARIZE(ALLSELECTED('Sales'), 'Date'[Date], 'Product'[Product]),
```

```

        ORDERBY('Date'[Date]),
        KEEP,
        PARTITIONBY('Product'[Product])
    ),
    CALCULATE(AVERAGE(Sales[Unit Price]))
)

```

각 제품의 3일 평균 단가를 반환합니다. 3일 기간은 제품에 판매가 있는 3일로 구성되며, 반드시 3일 연속 판매는 아닙니다.

예제 2 - 측정값

다음 측정값은 다음과 같습니다.

DAX

```

RunningSum =
SUMX (
    WINDOW (
        1, ABS, 0, REL,
        ALLSELECTED (
            'Date'[Fiscal Year],
            'Date'[Month Number Of Year]
        ),
        PARTITIONBY ('Date'[Fiscal Year])
    ),
    [Total Sales]
)

```

매 회계 연도에 대해 다시 시작되는 월별 연도 수별 총 판매액의 실행 합계를 반환합니다.

 테이블 확장

Year	연도의 월 수	매출액	Runningsum
FY2018	1	\$1,327,675	\$1,327,675
FY2018	2	\$3,936,463	\$5,264,138
FY2018	3	\$700,873	\$5,965,011
FY2018	4	\$1,519,275	\$7,484,286
FY2018	5	\$2,960,378	\$10,444,664
FY2018	6	\$1,487,671	\$11,932,336

Year	연도의 월 수	매출액	Runningsum
FY2018	7	\$1,423,357	\$13,355,693
FY2018	8	\$2,057,902	\$15,413,595
FY2018	9	\$2,523,948	\$17,937,543
FY2018	10	\$561,681	\$18,499,224
FY2018	11	\$4,764,920	\$23,264,145
FY2018	12	\$596,747	\$23,860,891
FY2019	1	\$1,847,692	\$1,847,692
FY2019	2	\$2,829,362	\$4,677,054
FY2019	3	\$2,092,434	\$6,769,488
FY2019	4	\$2,405,971	\$9,175,459
FY2019	5	\$3,459,444	\$12,634,903
FY2019	6	\$2,850,649	\$15,485,552
FY2019	7	\$2,939,691	\$18,425,243
FY2019	8	\$3,964,801	\$22,390,045
FY2019	9	\$3,287,606	\$25,677,650
FY2019	10	\$2,157,287	\$27,834,938
FY2019	11	\$3,611,092	\$31,446,030
FY2019	12	\$2,624,078	\$34,070,109
FY2020	1	\$3,235,187	\$3,235,187
FY2020	2	\$4,070,046	\$7,305,233
FY2020	3	\$4,429,833	\$11,735,066
FY2020	4	\$4,002,614	\$15,737,680
FY2020	5	\$5,265,797	\$21,003,477
FY2020	6	\$3,465,241	\$24,468,717
FY2020	7	\$3,513,064	\$27,981,781
FY2020	8	\$5,247,165	\$33,228,947

Year	연도의 월 수	매출액	Runningsum
FY2020	9	\$5,104,088	\$38,333,035
FY2020	10	\$3,542,150	\$41,875,184
FY2020	11	\$5,151,897	\$47,027,081
FY2020	12	\$4,851,194	\$51,878,275

예제 3 - 시각적 계산

다음 시각적 계산 DAX 쿼리:

DAX
TotalSalesRunningSumByYear = <code>SUMX(WINDOW(0, ABS, 0, REL, ROWS, HIGHESTPARENT), [SalesAmount])</code>

매년 계산된 월별 누적 총 판매량을 반환합니다. 동일한 결과와 함께 HIGHESTPARENT 대신 값 1과 -2를 사용할 수 있습니다.

아래 스크린샷은 시각적 행렬 및 시각적 계산식을 보여줍니다.

				TotalSalesRunningSumByYear		TotalSalesRunningSumByQuarter
CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	TotalSalesRunningSumByYear	TotalSalesRunningSumByQuarter	
2011	1	1	\$323,743.2056	323,743.21	323,743.21	
		2	\$437,372.4476	761,115.65	761,115.65	
		3	\$514,918.2284	1,276,033.88	1,276,033.88	
	2	Total	\$1,276,033.8816	1,276,033.88	1,276,033.88	
		4	\$523,950.0258	1,799,983.91	523,950.03	
		5	\$539,469.0004	2,339,452.91	1,063,419.03	
		6	\$647,524.214	2,986,977.12	1,710,943.24	

예제 4 - 시각적 계산

다음 시각적 계산 DAX 쿼리:

DAX
TotalSalesRunningSumByQuarter = <code>SUMX(WINDOW(0, ABS, 0, REL, , -1), [SalesAmount])</code>

각 분기에 따라 계산된 월별 누적 총 매출을 반환합니다.

관련 콘텐츠

[INDEX](#)

[MOVINGAVERAGE](#)

[OFFSET](#)

[Orderby](#)

[PARTITIONBY](#)

[범위](#)

[RANK](#)

[ROWNUMBERRUNNINGSUM](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

재무 기능

아티클 • 2023. 10. 20.

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

이 범주에서

함수	설명
ACCRINT	정기적인 이자를 지불하는 유가 증권의 누적 이자를 반환합니다.
ACCRINTM	만기 시 이자를 지급하는 유가 증권의 누적 이자를 반환합니다.
AMORDEGRC	각 회계 기간의 감가 상각액을 반환합니다. AMORLINC와 유사하게, 감가상각 계수를 제외하고 자산의 수명에 따라 적용됩니다.
AMORLINC	각 회계 기간의 감가 상각액을 반환합니다.
COUPDAYBS	쿠폰 기간의 시작부터 결산 날짜까지의 일 수를 반환합니다.
COUPDAYS	결산 날짜가 포함된 쿠폰 기간의 일 수를 반환합니다.
COUPDAYSNC	결산 날짜부터 다음 쿠폰 날짜까지의 일 수를 반환합니다.
COUPNCD	결산일 이후의 다음 쿠폰 날짜를 반환합니다.
COUPNUM	가장 가까운 전체 쿠폰 반올림된 결산일과 만기일 사이에 지불해야 하는 쿠폰 수를 반환합니다.
COUPPCD	결산일 이전 쿠폰 날짜를 반환합니다.
CUMIPMT	start_period end_period 사이의 대출에 대해 지불한 누적 이자를 반환합니다.
CUMPRINC	start_period end_period 사이의 대출에서 지불한 누적 원금을 반환합니다.
DB	고정 감소 잔액 방법을 사용하여 지정된 기간 동안 자산의 감가 상각액을 반환합니다.
Ddb	이중 감소 잔액 메서드 또는 지정한 다른 방법을 사용하여 지정된 기간 동안 자산의 감가 상각액을 반환합니다.
디스크	보안에 대한 할인율을 반환합니다.
DOLLARDE	정수 부분으로 표현된 달러 가격과 소수 부분(예: 1.02)을 10진수로 표현된 달러 가격으로 변환합니다.
DOLLARFR	정수 부분으로 표현된 달러 가격과 소수 부분(예: 1.02)을 10진수로 표현된 달러 가격으로 변환합니다.

함수	설명
기간	\$100의 가정된 파 값에 대한 마콜리 기간을 반환합니다.
효과	명목 연 이자율 및 연간 복합 기간 수를 감안할 때 유효한 연간 이자율을 반환합니다.
Fv	일정한 이자율을 기준으로 투자의 미래 가치를 계산합니다.
INTRATE	완전 투자 유가 증권의 이자율을 반환합니다.
IPMT	주기적이고 일정한 지급액 및 일정한 이자율을 기준으로 투자에 대해 지정된 기간의 이자 지급액을 반환합니다.
ISPMT	짝수 원금 상환으로 지정된 대출 기간(또는 투자)에 대해 지불하거나 받은 이자를 계산합니다.
MDURATION	액면가가 \$100인 보안에 대해 수정된 Macauley 기간을 반환합니다.
명목상	유효 금리와 연간 복합 기간 수를 감안할 때 명목 연 이자율을 반환합니다.
Nper	정기적, 일정한 지불 및 일정한 이자율을 기준으로 투자 기간 수를 반환합니다.
ODDFPRICE	첫 번째 기간이 홀수(단기 또는 장기)인 유가 증권의 액면가 \$100당 가격을 반환합니다.
ODDFYIELD	홀수(짧거나 긴) 첫 번째 마침표가 있는 보안의 수익률을 반환합니다.
ODDLPRICE	지난 쿠폰 동안 홀수(짧거나 긴) 유가 증권의 액면가 \$100당 가격을 반환합니다.
ODDLYIELD	홀수(짧거나 긴) 마지막 기간이 있는 보안의 수익률을 반환합니다.
PDURATION	투자가 지정된 값에 도달하는 데 필요한 기간 수를 반환합니다.
Pmt	일정한 상환액과 일정한 이자율을 기준으로 대출 상환액을 계산합니다.
PPMT	주기적이고 일정한 지급액 및 일정한 이자율을 기준으로 투자에 대해 지정된 기간 동안 원금에 대한 지급액을 반환합니다.
가격	정기 이자를 지급하는 유가 증권의 액면가 \$100당 가격을 반환합니다.
PRICEDISC	할인된 유가 증권의 액면가 \$100당 가격을 반환합니다.
PRICEMAT	만기 시 이자를 지불하는 유가 증권의 액면가 \$100당 가격을 반환합니다.
태양광 발전	일정한 이자율을 기준으로 대출 또는 투자의 현재 가치를 계산합니다.
속도	연금 기간당 이자율을 반환합니다.
받은	완전 투자 유가 증권의 만기 시 받은 금액을 반환합니다.
RRI	투자 증가에 해당하는 이자율을 반환합니다.

함수	설명
.sln	한 기간 동안 자산의 직선 감가 상각액을 반환합니다.
시드	지정된 기간 동안 자산의 연수 합계 감가 상각액을 반환합니다.
TBILLEQ	국채에 해당하는 채권 수익률을 반환합니다.
TBILLPRICE	국채의 액면가 \$100당 가격을 반환합니다.
트빌리엘드	국채 수익률을 반환합니다.
VDB	이중 감소 잔액 메서드 또는 지정한 다른 방법을 사용하여 부분 기간을 포함하여 지정한 기간 동안 자산의 감가 상각액을 반환합니다.
XIRR	반드시 주기적인 것은 아닌 현금 흐름 일정에 대한 내부 수익률을 반환합니다.
XNPV	반드시 주기적인 것은 아닌 현금 흐름 일정에 대한 현재 값을 반환합니다.
항복	정기적인 이자를 지불하는 유가 증권의 수익률을 반환합니다.
YIELDDISC	할인된 보안에 대한 연간 수익률을 반환합니다.
YIELDMAT	만기 시 이자를 지급하는 유가 증권의 연간 수익률을 반환합니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

ACCRINT

아티클 • 2023. 10. 20.

정기적인 이자를 지불하는 유가 증권의 누적 이자를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
이슈	보안의 발급 날짜입니다.
first_interest	유가 증권의 첫 번째 이자 날짜입니다.
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행 일 이후의 날짜입니다.
rate	보안의 연간 쿠폰 비율입니다.
par	보안의 파 값입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 수 있는 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.
calc_method	(선택 사항) 결산 날짜가 first_interest 날짜보다 늦을 때 누적된 총 이자를 계산하는 방법을 지정하는 논리 값입니다. calc_method 생략하면 TRUE로 간주됩니다. - calc_method TRUE로 평가되거나 생략된 경우 ACCRINT는 발생한 총 이자를 문제에서 해결로 반환합니다. - calc_method FALSE로 평가되면 ACCRINT는 first_interest 발생한 이자를 정산으로 반환합니다.

basis 매개 변수는 다음 값을 허용합니다.

기초	일 수 기준
0 또는 생략됨	미국(NASD) 30/360

기초	일 수 기준
1	실제/실제
2	실제/360
3	실제/365
4	유럽 30/360

Return Value

누적된 이자입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 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 맞는 준 쿠폰 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 및 settlement는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - 문제, first_interest 또는 결제가 유효한 날짜가 아닙니다.
 - 문제 \geq 해결.
 - 속도 ≤ 0 입니다.
 - $\text{par} \leq 0$.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.

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

예제

Data	설명
2007년 3월 1일	문제 날짜
2008년 8월 31일	첫 번째 이자 날짜
2008년 5월 1일	결산일
10%	쿠폰 비율
1000	파 가치
2	빈도는 반기(위 참조)입니다.
0	30/360 기준(위 참조)

예 1

다음 DAX 쿼리:

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

위에 지정된 조건이 있는 보안에 대해 발행에서 결제까지의 누적 이자를 반환합니다.

[값]
116.944444444444

예제 2

다음 DAX 쿼리:

```
DAX
EVALUATE
{
```

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

위에 지정된 조건이 있는 보안에 대해 first_interest 발생한 이자를 결제로 반환합니다.

[값]

66.9444444444445

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ACCRINTM

아티클 • 2023. 10. 20.

만기 시 이자를 지급하는 유가 증권의 누적 이자를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
이슈	보안의 발급 날짜입니다.
성숙	보안의 만기일입니다.
rate	보안의 연간 쿠폰 비율입니다.
par	보안의 파값입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

누적된 이자입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- ACCRINTM은 다음과 같이 계산됩니다.

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

여기서

- A = 월별 기준으로 계산된 누적 일수입니다. 만기 항목에 대한 관심의 경우 발행 일부터 만기일까지의 일 수가 사용됩니다.
- D = 연간 기준입니다.
- 문제 및 완성도는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - 문제 또는 만기가 유효한 날짜가 아닙니다.
 - 문제 \geq 성숙도입니다.
 - 속도 ≤ 0 입니다.
 - $\text{par} \leq 0$.
 - $\text{basis} < 0$ 또는 $\text{basis} > 4$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
2008년 4월 1일	문제 날짜
2008년 6월 15일	만기일
10%	백분율 쿠폰
1000	파 값
3	실제/365 기준(위 참조)

다음 DAX 쿼리:

DAX

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

위에 지정된 조건을 사용하여 보안에 대한 누적 이자를 반환합니다.

[값]

20.5479452054795

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

AMORDEGRC

아티클 • 2023. 10. 20.

각 회계 기간의 감가 상각액을 반환합니다. 이 함수는 프랑스 회계 시스템에 대해 제공됩니다. 회계 기간 중간에 자산을 구매하는 경우 비례 배분 감가 상각을 고려합니다. 이 함수는 자산의 수명에 따라 계산에 감가상각 계수가 적용된다는 점을 제외하고 AMORLINC와 유사합니다.

구문

DAX

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

매개 변수

용어	정의
비용	자산의 비용입니다.
date_purchased	자산 구매 날짜입니다.
first_period	첫 번째 기간의 끝 날짜입니다.
구조	자산 수명이 끝날 때의 회수 값입니다.
기간	기간입니다.
rate	감가상각률입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기초	날짜 시스템
0 또는 생략됨	360일(NASD 메서드)
1	실제
3	1년 365일
4	1년 360일(유럽 방법)

Return Value

각 회계 기간의 감가상각입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 이 함수는 자산의 수명 마지막 기간까지 또는 누적 감가상각 값이 자산의 비용에서 회수 값을 뺀 값보다 클 때까지 감가 상각액을 반환합니다.
- 감가상각 계수는 다음과 같습니다.

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

- 감가상각률은 마지막 기간 이전 기간 동안 50%로 증가하고 마지막 기간 동안 100%로 증가할 것입니다.
- 기간 및 기준은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - cost < 0.
 - first_period 또는 date_purchased 유효한 날짜가 아닙니다.
 - date_purchased > first_period.
 - salvage < 0 또는 salvage > cost.
 - period < 0.
 - 속도 ≤ 0입니다.
 - 자산의 수명은 0과 1, 1, 2, 2, 3 또는 4와 5 사이입니다.
 - basis는 0, 1, 3 또는 4 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
2400	비용
2008년 8월 19일	구매한 날짜
2008년 12월 31일	첫 번째 기간의 끝
300	인양 값
1	Period
15%	감가상각비율
1	실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건에 따라 첫 번째 마침표의 감가 상각액을 반환합니다.

[값]
776

피드백

이 페이지가 도움이 되었나요?

Yes

No

AMORLINC

아티클 • 2023. 10. 20.

각 회계 기간의 감가 상각액을 반환합니다. 이 함수는 프랑스 회계 시스템에 대해 제공됩니다. 회계 기간 중간에 자산을 구매하는 경우 비례 배분 감가 상각을 고려합니다.

구문

DAX

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

매개 변수

용어	정의
비용	자산의 비용입니다.
date_purchased	자산 구매 날짜입니다.
first_period	첫 번째 기간의 끝 날짜입니다.
구조	자산 수명이 끝날 때의 회수 값입니다.
기간	기간입니다.
rate	감가상각률입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

기초	날짜 시스템
0 또는 생략됨	360일(NASD 메서드)
1	실제
3	1년 365일
4	1년 360일(유럽 방법)

Return Value

각 회계 기간의 감가상각입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 기간 및 기준은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - cost < 0.
 - first_period 또는 date_purchased 유효한 날짜가 아닙니다.
 - date_purchased > first_period.
 - salvage < 0 또는 salvage > cost.
 - period < 0.
 - 속도 ≤ 0 입니다.
 - basis는 0, 1, 3 또는 4 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
2400	비용
2008년 8월 19일	구매한 날짜
2008년 12월 31일	첫 번째 기간의 끝
300	인양 값
1	Period
15%	감가상각비율
1	실제 기준(위 참조)

다음 DAX 쿼리:

DAX

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

위에 지정된 조건에 따라 첫 번째 마침표의 감가 상각액을 반환합니다.

[값]

360

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUPDAYBS

아티클 • 2023. 10. 20.

쿠폰 기간의 시작부터 결산 날짜까지의 일 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

쿠폰 기간의 시작부터 결산 날짜까지의 일 수입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년 후인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 11일	결산일
15-11월 11일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위의 조건을 가진 채권에 대해 쿠폰 기간의 시작부터 결산 날짜까지의 일 수를 반환합니다.

[값]

71

피드백

이 페이지가 도움이 되었나요?

Yes

No

COUPDAYS

아티클 • 2023. 10. 20.

결산 날짜가 포함된 쿠폰 기간의 일 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

결산 날짜가 포함된 쿠폰 기간의 일 수입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 2008년 1월 1일 이후 30년 후인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 11일	결산일
15-11월 11일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건의 채권에 대한 결산 날짜가 포함된 쿠폰 기간의 일 수를 반환합니다.

[값]

181

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUPDAYSNC

아티클 • 2023. 10. 20.

결산 날짜부터 다음 쿠폰 날짜까지의 일 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

결산일에서 다음 쿠폰 날짜까지의 일 수입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 11일	결산일
15-11월 11일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건의 채권에 대해 결산일에서 다음 쿠폰 날짜까지의 일 수를 반환합니다.

[값]

110

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COUPNCD

아티클 • 2023. 10. 20.

결산일 이후의 다음 쿠폰 날짜를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

다음 쿠폰 날짜는 결산 날짜 이후입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 2008년 1월 1일 이후 30년 후인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 11일	결산일
15-11월 11일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건이 있는 채권에 대해 결산일 이후의 다음 쿠폰 날짜를 반환합니다.

[값]

2011년 5월 15일 오전 12:00:00

피드백

이 페이지가 도움이 되었나요?

Yes

No

COUPNUM

아티클 • 2023. 10. 20.

가장 가까운 전체 쿠폰 반올림된 결산일과 만기일 사이에 지불해야 하는 쿠폰 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기준	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

결산일과 만기일 사이에 지불해야 하는 쿠폰 수입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 7일	결산일
15-11월 8일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

```
DAX
EVALUATE
{
```

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

```
}
```

위에 지정된 조건을 사용하여 채권에 대한 쿠폰 지불 횟수를 반환합니다.

[**값**]

4

피드백

이 페이지가 도움이 되었나요?

Yes

No

COUPPCD

아티클 • 2023. 10. 20.

결산일 이전 쿠폰 날짜를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

결산일 이전 쿠폰 날짜입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
1월 25일-1월 11일	결산일
15-11월 11일	만기일
2	반기 쿠폰(위 참조)
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에서 지정한 조건을 사용하여 채권의 결산일 이전 쿠폰 날짜를 반환합니다.

[값]

2010년 11월 15일 오전 12:00:00

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CUMIPMT

아티클 • 2023. 10. 20.

start_period end_period 사이의 대출에 대해 지불한 누적 이자를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	이자율입니다.
Nper	총 지불 기간 수입니다.
pv	현재 값입니다.
start_period	계산의 첫 번째 기간입니다. 1에서 end_period 사이여야 합니다(포함).
end_period	계산의 마지막 기간입니다. start_period nper(포함) 사이여야 합니다.
type	지불 시기입니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

Type	타이밍
0(영)	기간 종료 시 지급
1	기간 시작 시 지급

Return Value

지정된 기간 동안 지급된 누적 이자입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연 이자율이 10%인 4년 만기 대출을 매월 상환하는 경우 금리는 0.1/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.1, nper에 4를 사용합니다.
- start_period, end_period 및 형식은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - 속도 ≤ 0 입니다.
 - nper < 1 .
 - pv ≤ 0 .
 - start_period < 1 또는 start_period $>$ end_period.
 - end_period $<$ start_period 또는 end_period $>$ nper.
 - 형식은 0 또는 1 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
9%	연 이자율
30	대출 년
125000	현재 값

예 1

다음 DAX 쿼리:

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

매월 초에 지급한다고 가정하고 13~24년 지급 2년차에 지급된 총 이자를 반환합니다.

[값]
-11052.3395838718

예제 2

다음 DAX 쿼리:

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

월말에 지급한다고 가정하면 첫 달에 단일 지불로 지급된 이자를 반환합니다.

[값]

-937.5

피드백

이 페이지가 도움이 되었나요?

Yes

No

CUMPRINC

아티클 • 2023. 10. 20.

start_period end_period 사이의 대출에서 지불한 누적 원금을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	이자율입니다.
Nper	총 지불 기간 수입니다.
pv	현재 값입니다.
start_period	계산의 첫 번째 기간입니다. 1에서 end_period 사이여야 합니다(포함).
end_period	계산의 마지막 기간입니다. start_period nper(포함) 사이여야 합니다.
type	지불 시기입니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

Type	타이밍
0(영)	기간 종료 시 지급
1	기간 시작 시 지급

Return Value

지정된 기간에 지불한 누적 원금입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연 이자율이 10%인 4년 만기 대출을 매월 상환하는 경우 금리는 0.1/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.1, nper에 4를 사용합니다.
- start_period, end_period 및 형식은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - 속도 ≤ 0 입니다.
 - nper < 1 .
 - pv ≤ 0 .
 - start_period < 1 또는 start_period $>$ end_period.
 - end_period $<$ start_period 또는 end_period $>$ nper.
 - 형식은 0 또는 1 이외의 숫자입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
9%	연 이자율
30	기간(연도)
125000	현재 값

예 1

다음 DAX 쿼리:

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

지급 2년차에 지급된 총 원금(13~24세)을 매월 초에 지급하는 것으로 가정합니다.

[값]
-927.153472378062

예제 2

다음 DAX 쿼리:

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

월말에 지불이 이루어진 것으로 가정하여 첫 달에 단일 지불로 지급된 원금을 반환합니다.

[값]

-68.2782711809784

피드백

이 페이지가 도움이 되었나요?

Yes

No

DB

아티클 • 2023. 10. 20.

고정 감소 잔액 방법을 사용하여 지정된 기간 동안 자산의 감가 상각액을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
비용	초기 자산 비용입니다.
구조	감가상각 종료 시의 값입니다(자산의 회수 값이라고도 함). 이 값은 0일 수 있습니다.
생활	자산이 감가상각되는 기간의 수입니다(자산의 유용한 수명이라고도 함).
기간	감가상각을 계산할 기간입니다. 기간은 수명과 동일한 단위를 사용해야 합니다. 1에서 수명(포함) 사이여야 합니다.
개월	(선택 사항) 첫 해의 월 수입니다. 월을 생략하면 12로 간주됩니다.

Return Value

지정된 기간 동안의 감가상각입니다.

설명

- 고정 감소 잔액 메서드는 감가상각을 고정 금리로 계산합니다. DB는 다음 수식을 사용하여 기간 동안 감가 상각액을 계산합니다.

$$(\text{cost} - \text{이전 기간의 총 감가상각}) \times \text{rate}$$

여기서

$$\circ \text{ rate} = 1 - \left(\left(\frac{\text{salvage}}{\text{cost}} \right)^{\left(\frac{1}{\text{life}} \right)} \right), \text{ 반올림된 소수 자릿수 3자리}$$

- 첫 번째 기간과 마지막 기간에 대한 감가 상각은 특별한 경우입니다.
 - 첫 번째 기간 동안 DB는 다음 수식을 사용합니다.

$$\frac{\text{cost} \times \text{rate} \times \text{month}}{12}$$

- 마지막 기간 동안 DB는 다음 수식을 사용합니다.

$$\frac{(\text{cost} - \text{total depreciation from prior periods}) \times \text{rate} \times (12 - \text{month})}{12}$$

- 기간 및 월은 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

예 1

다음 DAX 쿼리:

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

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

[값]

166666.666666667

예제 2

다음은 수명 동안 다른 연도의 모든 자산의 총 감가 상각액을 계산합니다. 여기서 첫 해에는 7개월의 감가상각만 포함되며, 작년에는 5개월만 포함됩니다.

DAX

```
DEFINE
    VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])+1
    VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
    ADDCOLUMNS (
        DepreciationPeriods,
        "Current Period Total Depreciation",
        SUMX (
            FILTER (
                Asset,
                [Value] <= [LifetimeYears]+1
            ),
            DB([InitialCost], [SalvageValue], [LifetimeYears], [Value], 7)
        )
    )
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Ddb

아티클 • 2023. 10. 20.

이중 감소 잔액 메서드 또는 지정한 다른 방법을 사용하여 지정된 기간 동안 자산의 감가 상각액을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
비용	초기 자산 비용입니다.
구조	감가상각 종료 시의 값입니다(자산의 회수 값이라고도 함). 이 값은 0일 수 있습니다.
생활	자산이 감가상각되는 기간의 수입니다(자산의 유용한 수명이라고도 함).
기간	감가상각을 계산할 기간입니다. 기간은 수명과 동일한 단위를 사용해야 합니다. 1에서 수명(포함) 사이여야 합니다.
요소	(선택 사항) 잔액이 감소하는 비율입니다. 인수를 생략하면 2(이중 감소 잔액 방법)로 간주됩니다.

Return Value

지정된 기간 동안의 감가상각입니다.

설명

- 이중 감소 잔액 메서드는 감가상각을 가속 속도로 계산합니다. 감가상각은 첫 번째 기간에 가장 높고 연속적인 기간에서 감소합니다. DDB는 다음 수식을 사용하여 기간 동안 감가 상각액을 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 감소 잔액 방법을 사용하지 않으려면 인수를 변경합니다.
- 감가 상각액이 감가상각 계산보다 클 때 직선 감가 상각 방법으로 전환하려면 VDB 함수를 사용합니다.
- 기간은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - cost < 0.
 - salvage < 0.
 - life < 1.
 - period < 1 또는 period > life.
 - 요소 ≤ 0입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

예 1

다음 DAX 쿼리:

DAX

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

10년 후 가치가 \$0이라고 가정하여 5th 해의 자산 감가상각을 반환합니다. 이 계산에서는 1.5의 요소를 사용합니다.

[값]

78300.9375

예제 2

다음은 수명 동안 다른 연도의 모든 자산의 총 감가 상각액을 계산합니다. 이 계산에서는 기본 요소 2(이중 감소 잔액 방법)를 사용합니다.

DAX

```
DEFINE
    VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])
    VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
    ADDCOLUMNS (
        DepreciationPeriods,
        "Current Period Total Depreciation",
        SUMX (
            FILTER (
                Asset,
                [Value] <= [LifetimeYears]
            ),
            DDB([InitialCost], [SalvageValue], [LifetimeYears], [Value])
        )
    )
```

피드백

이 페이지가 도움이 되었나요?

구역

아티클 • 2023. 10. 20.

보안에 대한 할인율을 반환합니다.

구문

DAX

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

매개 변수

용어 정의

정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
pr	유가 증권의 액면가 \$100당 가격입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
기저	(선택 사항) 사용할 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

할인율입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2018년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2018년 1월 1일, 결산일은 2018년 7월 1일, 만기일은 발행일 2018년 1월 1일 이후 30년 후인 2048년 1월 1일입니다.
- DISC는 다음과 같이 계산됩니다.

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

여기서

- B = 연도 기준으로 1년 단위의 일 수입니다.
- DSM = settlement와 maturity 사이의 일 수입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - pr \leq 0.
 - 상환 \leq 0.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
07/01/2018	결산일

Data	설명
01/01/2048	만기일
97.975	가격
100	상환 값
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건이 있는 채권의 채권 할인율을 반환합니다.

[값]
0.000686384169121348

피드백

이 페이지가 도움이 되었나요?

 Yes  No

DOLLARDE

아티클 • 2023. 10. 20.

정수 부분으로 표현된 달러 가격과 소수 부분(예: 1.02)을 10진수로 표현된 달러 가격으로 변환합니다. 소수 달러 숫자는 때때로 보안 가격에 사용됩니다.

구문

DAX

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

매개 변수

용어	정의
fractional_dollar	정수 부분과 소수 부분으로 표현되는 숫자로, 소수 기호로 구분됩니다.
분수	분수의 분모에 사용할 정수입니다.

Return Value

fractional_dollar 10진수 값입니다.

설명

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

예시

다음 DAX 쿼리:

DAX

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

원래 소수 가격인 1.02의 10진수 가격인 1.125를 1과 2/16으로 반환합니다. 분수 값은 16 이므로 가격은 1/16의 정밀도를 가집니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DOLLARFR

아티클 • 2023. 10. 20.

10진수로 표현된 달러 가격을 정수 부분으로 표현된 달러 가격과 1.02와 같은 분수 부분으로 변환합니다. 소수 달러 숫자는 때때로 보안 가격에 사용됩니다.

구문

DAX

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

매개 변수

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

Return Value

정수 부분과 소수 부분으로 표현되는 decimal_dollar 소수 자릿수 값입니다.

설명

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

예시

다음 DAX 쿼리:

DAX

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

원래 10진수 가격인 1.125의 해당 분수 가격인 1.02 및 2/16으로 읽은 1.02를 반환합니다.
분수 값은 16이므로 가격은 1/16의 정밀도를 가집니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DURATION

아티클 • 2023. 10. 20.

추정 액면가 \$100에 대한 Macauley 기간을 반환합니다. 기간은 현금 흐름의 현재 가치의 가중 평균으로 정의되며, 수익률 변화에 대한 채권 가격의 응답의 척도로 사용됩니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
coupon	보안의 연간 쿠폰 비율입니다.
yld	유가 증권의 연간 수익률.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

마콜리 기간입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - coupon < 0.
 - yld < 0
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
07/01/2018	결산일
01/01/2048	만기일
8.0%	백분율 쿠폰
9.0%	백분율 수익률
2	빈도는 반기(위 참조)입니다.

Data	설명
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 채권에 대한 마콜리 기간을 반환합니다.

[값]
10.9191452815919

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

효과

아티클 • 2023. 10. 20.

명목 연 이자율 및 연간 복합 기간 수를 감안할 때 유효한 연간 이자율을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
nominal_rate	명목 이자율입니다.
nper	연간 복합 기간의 수입니다.

Return Value

유효 연 이자율입니다.

설명

- EFFECT는 다음과 같이 계산됩니다.

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

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

예시

Data	설명
5.25%	명목 이자율
4	연간 복합 기간 수

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 유효 이자율을 반환합니다.

[값]
0.0535426673707584

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FV

아티클 • 2023. 10. 20.

일정한 이자율을 기준으로 투자의 미래 가치를 계산합니다. 주기적, 일정한 지불 및/또는 단일 일괄 결제로 FV를 사용할 수 있습니다.

구문

DAX

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

매개 변수

용어	정의
rate	기간당 이자율입니다.
Nper	연금의 총 지급 기간 수입니다.
Pmt	각 기간 동안 결제한 금액입니다. 그것은 연금의 수명 동안 변경할 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료나 세금은 포함되지 않습니다.
pv	(선택 사항) 현재 값 또는 일련의 미래 지불이 지금 가치가 있는 일괄 합계 금액입니다. pv를 생략하면 BLANK로 간주됩니다.
type	(선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.	결제가 기한인 경우
0 또는 생략됨	마침표가 끝날 때
1	기간의 시작 부분에서

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

Return Value

투자의 미래 가치.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연간 이자가 12%인 4년 대출금을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 모든 인수에 대해 저축 예금과 같이 지불하는 현금은 음수로 표시됩니다. 배당금 검사 같이 받는 현금은 양수로 표시됩니다.
- 형식이 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nper < 1
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
6%	연 이자율
10	결제 수
200-	지불 금액
-500	현재 값
1	지불은 기간의 시작 부분에 기한입니다 (0은 지불이 기간의 끝에 기한임을 나타냅니다)

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 투자의 미래 가치를 반환합니다.

[값]
2581.40337406012

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

INTRATE

아티클 • 2023. 10. 20.

완전 투자 유가 증권의 이자율을 반환합니다.

구문

DAX

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

매개 변수

용어 정의

정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
투자	유가 증권에 투자한 금액입니다.
보너스	만기 시 받을 금액입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

이자율입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- INTRATE는 다음과 같이 계산됩니다.

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

여기서

- B = 연도 기준으로 1년 단위의 일 수입니다.
- DIM = 결산에서 만기까지의 일 수입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - 투자 ≤ 0 .
 - 상환 ≤ 0 .
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
2/15/2008	결산일
5/15/2008	만기일

Data	설명
\$1,000,000	투자
\$1,014,420	상환 값
2	실제/360 기준

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 채권의 할인율을 반환합니다.

[값]
0.05768

피드백

이 페이지가 도움이 되었나요?

Yes

No

IPMT

아티클 • 2023. 10. 20.

주기적이고 일정한 지급액 및 일정한 이자율을 기준으로 투자에 대해 지정된 기간의 이자 지급액을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	기간당 이자율입니다.
per	이자를 찾으려는 기간입니다. 1에서 nper(포함) 사이여야 합니다.
Nper	연금의 총 지급 기간 수입니다.
pv	현재 값 또는 일련의 미래 지불이 지금 가치가 있는 일괄 합계 금액입니다.
Fv	(선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 BLANK로 간주됩니다.
type	(선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.	결제가 기한인 경우
0 또는 생략됨	마침표가 끝날 때
1	기간의 시작 부분에서

Return Value

지정된 기간의 이자 지급입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연간 이자가 12%인 4년 대출금을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 모든 인수에 대해 저축 예금과 같이 지불하는 현금은 음수로 표시됩니다. 배당금 검사 같이 받는 현금은 양수로 표시됩니다.
- 형식이 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
10.00%	연간 이자
3	대출 년
\$8,000	대출의 현재 가치

예 1

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 대출의 첫 번째 달에 만기된 월별 이자를 반환합니다.

[값]

-66.6666666666667

예제 2

다음 DAX 쿼리:

DAX

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

위에 지정된 조건이 있는 대출에 대한 작년의 연간 이자를 반환하며, 여기서 매년 상환됩니다.

[값]

-292.447129909366

피드백

이 페이지가 도움이 되었나요?

Yes

No

ISPMT

아티클 • 2023. 10. 20.

짝수 원금 상환으로 지정된 대출 기간(또는 투자)에 대해 지불하거나 받은 이자를 계산합니다.

구문

DAX

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

매개 변수

용어	정의
rate	투자에 대한 이자율입니다.
per	이자를 찾으려는 기간입니다. 0과 nper-1(포함) 사이여야 합니다.
Nper	투자에 대한 총 지불 기간 수입니다.
pv	투자의 현재 가치입니다. 대출의 경우 pv는 대출 금액입니다.

Return Value

지정된 기간 동안 지급된 이자(또는 받은 이자)입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연 이자율 12%로 4년 만기 대출을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 모든 인수에 대해 저축 예금이나 기타 인출과 같이 지불하는 현금은 음수로 표시됩니다. 배당금 검사 및 기타 예금과 같이 받은 현금은 양수로 표시됩니다.
- ISPMT는 1이 아닌 0으로 시작하는 각 기간을 계산합니다.

- 대부분의 대출은 정기적 인 지불과 상환 일정을 사용합니다. IPMT 함수는 이러한 유형의 대출에 대해 지정된 기간 동안의 이자 지급을 반환합니다.
- 일부 대출은 원금 상환과 함께 상환 일정을 사용합니다. ISPMT 함수는 이러한 유형의 대출에 대해 지정된 기간 동안의 이자 지급을 반환합니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nper = 0.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
\$4,000	현재 값
4	기간 수
10%	비용

ISPMT를 사용하는 시기를 설명하기 위해 아래 상각 테이블은 위에 지정된 조건에 짹수 원금 상환 일정을 사용합니다. 각 기간의 이자 청구는 이전 기간의 미지급 잔액의 요율 시간과 같습니다. 그리고 각 기간의 지급액은 짹수 원금과 해당 기간의 이자와 동일합니다.

Period	원금 결제	이자 지급	총 지불액	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 쿼리:

```
DAX
DEFINE
  VAR NumPaymentPeriods = 4
  VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1)
EVALUATE
ADDCOLUMNS (
```

```
PaymentPeriods,  
"Interest Payment",  
ISPMT(0.1, [Value], NumPaymentPeriods, 4000)  
)
```

위에 지정된 짹수 원금 상환 일정 및 조건을 사용하여 각 기간 동안 지급된 이자를 반환합니다. 이 값은 수신되지 않은 이자가 지급됨을 나타내기 위해 음수입니다.

[값]	[이자 지급]
0	-400
1	300-
2	200-
3	100-

피드백

이 페이지가 도움이 되었나요?

Yes

No

MDURATION

아티클 • 2023. 10. 20.

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

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
coupon	보안의 연간 쿠폰 비율입니다.
yld	유가 증권의 연간 수익률.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

수정된 마콜리 기간입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- 수정된 기간은 다음과 같이 정의됩니다.

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

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

예시

Data	설명
1/1/2008	결산일
1/1/2016	만기일

Data	설명
8%	백분율 쿠폰
9%	백분율 수익률
2	반도는 반기(위 참조)입니다.
1	실제/실제 기준(위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 채권의 수정된 마콜리 기간을 반환합니다.

[값]
5.73566981391884

피드백

이 페이지가 도움이 되었나요?

 Yes  No

명목상

아티클 • 2023. 10. 20.

유효 금리와 연간 복합 기간 수를 감안할 때 명목 연 이자율을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
effect_rate	유효 이자율입니다.
nper	연간 복합 기간의 수입니다.

Return Value

명목 연 이자율입니다.

설명

- NOMINAL과 EFFECT 간의 관계는 다음 수식에 나와 있습니다.

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

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

예시

Data	설명
5.3543%	유효 이자율
4	연간 복합 기간 수

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 명목 이자율을 반환합니다.

[값]
0.052500319868356

피드백

이 페이지가 도움이 되었나요?

Yes

No

Nper

아티클 • 2023. 10. 20.

정기적, 일정한 지불 및 일정한 이자율을 기준으로 투자 기간 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	기간당 이자율입니다.
Pmt	각 기간 동안 결제한 금액입니다. 그것은 연금의 수명 동안 변경할 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료나 세금은 포함되지 않습니다.
pv	현재 값 또는 일련의 미래 지불이 지금 가치가 있는 일괄 합계 금액입니다.
Fv	(선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 BLANK로 간주됩니다.
type	(선택 사항) 숫자 0 또는 1이며 결제 기한을 나타냅니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.	결제가 기한인 경우
0 또는 생략됨	마침표가 끝날 때
1	기간의 시작 부분에서

Return Value

투자 기간 수입니다.

설명

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

예시

Data	설명
12%	연 이자율
100-	각 기간마다 결제한 금액
1000-	현재 값
10000	미래 값
1	지불은 기간의 시작 부분에 기한이 있습니다 (위 참조)

다음 DAX 쿼리:

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

위에 지정된 조건에 설명된 투자 기간 수를 반환합니다.

[값]
59.6738656742946

피드백

이 페이지가 도움이 되었나요?

Yes

No

ODDFPRICE

아티클 • 2023. 10. 20.

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

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
이슈	보안의 발급 날짜입니다.
first_쿠폰	보안의 첫 번째 쿠폰 날짜입니다.
rate	유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- ODDFPRICE는 다음과 같이 계산됩니다.

홀수 짧은 첫 번째 쿠폰:

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

여기서

- A = 쿠폰 기간의 시작부터 결산 날짜(누적 일)까지의 일 수입니다.
- DSC = 결제 날짜부터 다음 쿠폰 날짜까지의 일 수입니다.
- DFC = 홀수 첫 번째 쿠폰 시작부터 첫 번째 쿠폰 날짜까지의 일 수입니다.
- E = 쿠폰 기간의 일 수입니다.
- N = 결산 날짜와 상환 날짜 사이에 지불해야 하는 쿠폰 수입니다. (이 숫자에 분수가 포함되어 있으면 다음 정수로 올 수 있습니다.)

홀수 긴 첫 번째 쿠폰:

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

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 = 결제 날짜부터 다음 쿠폰 날짜까지의 일 수입니다.
- E = 쿠폰 기간의 일 수입니다.
- N = 첫 번째 실제 쿠폰 날짜와 상환 날짜 사이에 지불해야 하는 쿠폰 수입니다. (이 숫자에 분수가 포함되어 있으면 다음 정수로 올 수 있습니다.)
- NC = 홀수 기간에 맞는 준 쿠폰 기간의 수입니다. (이 숫자에 분수가 포함되어 있으면 다음 정수로 올 수 있습니다.)
- NL_i = 홀수 기간 내 전체 i^{th} 또는 마지막 준이표 기간의 정상 길이(일수)입니다.
- N_q = 결산일과 첫 이표 사이 전체 준이표 기간의 수입니다.

- settlement, maturity, issue 및 first_쿠폰 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, issue 또는 first_쿠폰 유효한 날짜가 아닙니다.
 - maturity > first_coupon > settlement > issue 조건이 충족되지 않습니다.
 - rate < 0.
 - yld < 0.
 - 상환 ≤ 0.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	인수 설명
11/11/2008	결산일
3/1/2021	만기일
10/15/2008	문제 날짜
3/1/2009	첫 번째 쿠폰 날짜
7.85%	백분율 쿠폰
6.25%	백분율 수익률
\$100.00	재점화 값
2	빈도는 반기입니다.
1	실제/실제 기준

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    ODDFPRICE(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15), DATE(2009,3,1), 0.0785, 0.0625, 100.00, 2, 1)
}
```

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

[값]

113.597717474079

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ODDFYIELD

아티클 • 2023. 10. 20.

홀수(짧거나 긴) 첫 번째 마침표가 있는 보안의 수익률을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
이슈	보안의 발급 날짜입니다.
first_쿠폰	보안의 첫 번째 쿠폰 날짜입니다.
rate	유가 증권의 이자율입니다.
pr	보안 가격입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

기초	일 수 기준
2	실제/360
3	실제/365
4	유럽 30/360

Return Value

유가 증권의 수익률입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- ODDFYIELD는 반복 메서드를 사용하여 계산됩니다. ODDFPRICE 함수에 사용되는 수식을 기반으로 Newton 메서드를 사용합니다. 수익률은 지정된 수익률의 예상 가격이 가격에 근접할 때까지 100번의 반복을 통해 변경됩니다. ODDFYIELD에서 사용하는 수식은 ODDFPRICE를 참조하세요.
- settlement, maturity, issue 및 first_쿠폰 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, issue 또는 first_쿠폰 유효한 날짜가 아닙니다.
 - maturity > first_coupon > settlement > issue 조건이 충족되지 않습니다.
 - rate < 0.
 - pr ≤ 0.
 - 상환 ≤ 0.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	인수 설명
2008년 11월 11일	결산일
2021년 3월 1일	만기일
2008년 10월 15일	문제 날짜
2009년 3월 1일	첫 번째 쿠폰 날짜
5.75%	백분율 쿠폰
84.50	가격
100	재점화 값
2	빈도는 반기입니다.
0	30/360 기준

다음 DAX 쿼리:

DAX
EVALUATE { ODDFYIELD(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15), DATE(2009,3,1), 0.0575, 84.50, 100, 2, 0) }

위에 지정된 용어를 사용하여 홀수(짧거나 긴) 첫 번째 마침표가 있는 보안의 수익률을 반환합니다.

[값]
0.0772455415972989

피드백

이 페이지가 도움이 되었나요?

Yes

No

ODDLPRICE

아티클 • 2023. 10. 20.

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

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행 일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
last_interest	보안의 마지막 쿠폰 날짜입니다.
rate	유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

기초	일 수 기준
3	실제/365
4	유럽 30/360

Return Value

액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement, maturity 및 last_interest 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 last_interest 유효한 날짜가 아닙니다.
 - maturity > settlement > last_interest 조건이 충족되지 않습니다.
 - rate < 0.
 - yld < 0.
 - 상환 ≤ 0.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

Data	인수 설명
2008년 2월 7일	결산일
2008년 6월 15일	만기일
2007년 10월 15일	마지막 이자 날짜
3.75%	백분율 쿠폰
4.05%	백분율 수익률
\$100	재점화 값
2	빈도는 반기입니다.
0	30/360 기준

DAX

```
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.

[값]

99.8782860147213

피드백

이 페이지가 도움이 되었나요?

Yes

No

ODDLYIELD

아티클 • 2023. 10. 20.

홀수(짧거나 긴) 마지막 기간이 있는 보안의 수익률을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
last_interest	보안의 마지막 쿠폰 날짜입니다.
rate	유가 증권의 이자율입니다.
pr	보안 가격입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

유가 증권의 수익률입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- ODDLYIELD는 다음과 같이 계산됩니다.

$$\text{ODDLYIELD} = \left[\frac{\left(\text{redemption} + \left(\left(\sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right) \right) - \left(\text{par} + \left(\left(\sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right) \right)}{\text{par} + \left(\left(\sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right)} \right] \times \left[\frac{\text{frequency}}{\left(\sum_{i=1}^{\text{NC}} \frac{\text{DSC}_i}{\text{NL}_i} \right)} \right]$$

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 = 홀수에 맞는 준 쿠폰 기간의 수입니다. 이 숫자에 분수가 포함되어 있으면 다음 정수로 높아질 것입니다.
- $NL_i = i^{th}$ 의 날짜의 정상 길이 또는 홀수 이표 기간 내 마지막 준이표 기간입니다.
- settlement, maturity, last_interest 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity, last_interest 유효한 날짜가 아닙니다.
 - maturity > settlement > last_interest 조건이 충족되지 않습니다.
 - rate < 0.
 - pr ≤ 0.
 - 상환 ≤ 0.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

Data	인수 설명
4/20/2008	결산일
6/15/2008	만기일
12/24/2007	마지막 이자 날짜
3.75%	백분율 쿠폰
\$99.875	가격
\$100	상환 값
2	반도는 반기입니다.
0	30/360 기준

DAX
<pre>EVALUATE { ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375, 99.875, 100, 2, 0) }</pre>

위에서 지정한 용어를 사용하여 홀수(긴 기간 부족)가 있는 보안의 수익률을 반환합니다.

[값]
0.0451922356291692

피드백

이 페이지가 도움이 되었나요?

PDURATION

아티클 • 2023. 10. 20.

투자가 지정된 값에 도달하는 데 필요한 기간 수를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	기간당 이자율입니다.
pv	투자의 현재 가치입니다.
Fv	투자의 원하는 미래 가치.

Return Value

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.
 - 속도 ≤ 0 입니다.
 - $\text{pv} \leq 0$.
 - $\text{fv} \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 DAX 쿼리:

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

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

[값]

3.85986616262266

예제 2

다음 DAX 쿼리:

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

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

[값]

87.6054764193714

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Pmt

아티클 • 2023. 10. 20.

일정한 상환액과 일정한 이자율을 기준으로 대출 상환액을 계산합니다.

구문

DAX

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

매개 변수

용어 정의

rate 대출 이자율입니다.

Nper 대출에 대한 총 상환 횟수입니다.

pv 현재 값 또는 향후 일련의 지불 가치가 있는 총 금액입니다. 보안 주체라고도 합니다.

Fv (선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 BLANK로 간주됩니다.

type (선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.

결제가 기한인 경우

0 또는 생략됨

마침표가 끝날 때

1

기간의 시작 부분에서

참고: PMT의 인수에 대한 자세한 설명은 PV 함수를 참조하세요.

Return Value

단일 대출 상환 금액입니다.

설명

- PMT가 반환하는 지불에는 원금과 이자가 포함되지만 세금, 예약 지불 또는 대출과 관련된 수수료는 포함되지 않습니다.
- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연 이자율 12%로 4년 만기 대출을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 형식이 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nper < 1

팁: 대출 기간 동안 지급된 총 금액을 찾으려면 반환된 PMT 값을 nper로 곱합니다.

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

예제

예 1

Data	설명
8%	연 이자율
10	지급 월 수
\$10,000	대출 금액

다음 DAX 쿼리:

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

위에 지정된 조건을 가진 대출에 대해 월초에 지급된 월별 지불 금액을 반환합니다.

[값]
-1030.16432717797

참고: 1030.16432717797은 기간당 지불입니다. 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%	연 이자율
18	지급 기간(년)
\$50,000	대출 금액

다음 DAX 쿼리:

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

[값]
-129.081160867991

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

피드백

이 페이지가 도움이 되었나요?

Yes

No

PPMT

아티클 • 2023. 10. 20.

주기적이고 일정한 지급액 및 일정한 이자율을 기준으로 투자에 대해 지정된 기간 동안 원금에 대한 지급액을 반환합니다.

구문

DAX

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

매개 변수

용어	정의
rate	대출 이자율입니다.
per	마침표 지정 1에서 nper(포함) 사이여야 합니다.
Nper	연금의 총 지급 기간 수입니다.
PV	현재 값 - 일련의 미래 지불이 지금 가치가 있는 총 금액입니다.
Fv	(선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 BLANK로 간주됩니다.
Type	(선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

Type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.	결제가 기한인 경우
0 또는 생략됨	마침표가 끝날 때
1	기간의 시작 부분에서

참고: PPMT의 인수에 대한 자세한 설명은 PV 함수를 참조하세요.

Return Value

지정된 기간 동안 원금에 대한 지불입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연 이자율 12%로 4년 만기 대출을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 형식이 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

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

다음 DAX 쿼리:

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

위에 지정된 조건을 사용하여 대출에 대한 첫 달에 이루어진 원금 상환액을 반환합니다.

[값]
-75.6231860083663

예제 2

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

다음 DAX 쿼리:

DAX
<pre>EVALUATE { PPMT(0.08, 10, 10, 200000.00) }</pre>

위에 지정된 조건을 사용하여 대출에 대한 10년차의 원금 상환액을 반환합니다.

[값]
-27598.0534624214

피드백

이 페이지가 도움이 되었나요?

Yes

No

가격

아티클 • 2023. 10. 20.

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

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
rate	보안의 연간 쿠폰 비율입니다.
yld	유가 증권의 연간 수익률.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일이므로 1899년 12월 30일입니다.

- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- basis 및 frequency는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - rate < 0.
 - yld < 0.
 - 상환 \leq 0.
 - frequency는 1, 2 또는 4 이외의 숫자입니다.
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

중요:

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

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

- N = 1(N은 결산 날짜와 상환 날짜 사이에 지불해야 하는 쿠폰 수)인 경우 PRICE는 다음과 같이 계산됩니다.

$$DSR = E - A$$

$$T1 = 100 \times \frac{\text{rate}}{\text{frequency}} + \text{redemption}$$

$$T2 = \frac{\text{yld}}{\text{frequency}} \times \frac{DSR}{E} + 1$$

$$T3 = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{A}{E}$$

$$PRICE = \frac{T1}{T2} - T3$$

여기서

- DSC = 결제 날짜부터 다음 쿠폰 날짜까지의 일 수입니다.
- E = 결산 날짜가 속하는 쿠폰 기간의 일 수입니다.
- A = 쿠폰 기간의 시작부터 결산 날짜까지의 일 수입니다.

예시

Data	인수 설명
2/15/2008	결산일

Data	인수 설명
11/15/2017	만기일
5.75%	반기 쿠폰 백분율
6.50%	백분율 수익률
\$100	상환 값
2	반도는 반기입니다.
0	30/360 기준

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    PRICE(DATE(2008,2,15), DATE(2017,11,15), 0.0575, 0.065, 100, 2, 0)
}
```

위에서 지정한 조건을 사용하여 채권의 채권 가격을 반환합니다.

[값]
94.6343616213221

피드백

이 페이지가 도움이 되었나요?

 Yes  No

PRICEDISC

아티클 • 2023. 10. 20.

할인된 유가 증권의 액면가 \$100당 가격을 반환합니다.

구문

DAX

```
PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

매개 변수

용어 정의	
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
할인	보안의 할인율입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2018년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2018년 1월 1일, 결산일은 2018년 7월 1일, 만기일은 발행일 2018년 1월 1일 이후 30년 후인 2048년 1월 1일입니다.
- PRICEDISC는 다음과 같이 계산됩니다.

$$\text{PRICEDISC} = \text{redemption} - \text{discount} \times \text{redemption} \times \frac{\text{DSM}}{\text{B}}$$

여기서

- B = 연도 단위에 따라 연도의 일 수입니다.
- DSM = 결산부터 만기까지의 일 수입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - 할인 ≤ 0 .
 - 상환 ≤ 0 .
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	인수 설명
2/16/2008	결산일
3/1/2008	만기일

Data	인수 설명
5.25%	할인율
\$100	상환 값
2	실제/360 기준

다음 DAX 쿼리:

DAX
<pre>EVALUATE { PRICEDISC(DATE(2008,2,16), DATE(2008,3,1), 0.0525, 100, 2) }</pre>

위의 조건이 적용되는 채권의 액면가 \$100당 채권 가격을 반환합니다.

[값]
99.7958333333333

피드백

이 페이지가 도움이 되었나요?

Yes

No

PRICEMAT

아티클 • 2023. 10. 20.

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

구문

DAX

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
이슈	보안의 발급 날짜입니다.
rate	발행일 기준 유가 증권의 이자율입니다.
yld	유가 증권의 연간 수익률.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- PRICEMAT는 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 = 연도 단위에 따라 연도의 일 수입니다.
- DSM = 결산부터 만기까지의 일 수입니다.
- DIM = 문제부터 만기까지의 일 수입니다.
- A = 문제부터 결제까지의 일 수입니다.
- settlement, maturity 및 issue는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 issue는 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

Data	설명
2/15/2008	결산일
4/13/2008	만기일
11/11/2007	문제 날짜
6.10%	반기 쿠폰 백분율
6.10%	백분율 수익률
0	30/360 기준

DAX

```
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.

[값]

99.9844988755569

피드백

이 페이지가 도움이 되었나요?

Yes

No

PV

아티클 • 2023. 10. 20.

일정한 이자율을 기준으로 대출 또는 투자의 현재 가치를 계산합니다. 주기적이고 일정한 지불(예: 모기지 또는 기타 대출) 및/또는 투자 목표인 미래 가치와 함께 PV를 사용할 수 있습니다.

구문

DAX

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

매개 변수

용어	정의
rate	기간당 이자율입니다. 예를 들어 연 10%의 이자율로 자동차 대출을 받고 매월 상환하는 경우 월별 이자율은 0.1/12 또는 0.0083입니다. 0.1/12 또는 0.0083을 수식에 속도로 입력합니다.
Nper	연금의 총 지급 기간 수입니다. 예를 들어 4년 차 대출을 받고 매월 상환하는 경우 대출 기간은 4*12(또는 48)입니다. nper에 대한 수식에 48을 입력합니다.
Pmt	연금의 수명 동안 변경할 수 없는 각 기간을 지불했다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료나 세금은 포함되지 않습니다. 예를 들어 \$10,000에 대한 월 상환액은 4년 자동차 대출, 12퍼센트 이율의 경우 \$263.33입니다. 수식에 -263.33을 pmt로 입력합니다.
Fv	(선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 BLANK로 간주됩니다. 예를 들어 18년의 특별 프로젝트 비용을 지불하기 위해 \$50,000를 저축하려는 경우 \$50,000가 미래 가치입니다. 그런 다음 금리로 보수적 인 추측을하고 매달 저축해야 할 양을 결정할 수 있습니다.
type	(선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.	결제가 기한인 경우
0 또는 생략됨	마침표가 끝날 때
1	기간의 시작 부분에서

Return Value

대출 또는 투자의 현재 가치입니다.

설명

- 속도 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연간 이자가 12%인 4년 대출금을 매월 상환하는 경우 금리는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 금리에 0.12, nper에 4를 사용합니다.
- 연금에 적용되는 함수는 다음과 같습니다.
 - CUMIPMT
 - CUMPRINC
 - FV
 - IPMT
 - Pmt
 - PPMT
 - PV
 - 속도
 - XIRR
 - XNPV
- 연금은 연속 기간 동안 일련의 일정한 현금 지불입니다. 예를 들어 자동차 대출이나 모기지는 연금입니다. 자세한 내용은 각 연금 함수에 대한 설명을 참조하세요.
- 연금 기능에서 저축 예금과 같이 지불하는 현금은 음수로 표시됩니다. 배당금 검사 같이 받은 현금은 양수로 표시됩니다. 예를 들어 은행에 \$1,000가 예치될 때 예금자에게는 인수 -1000으로 표시되고 은행에는 인수 1000으로 표시됩니다.
- 한 가지 재정적 논쟁은 다른 인수의 관점에서 해결됩니다.
 - rate가 0이 아니면 다음을 수행합니다.

$$pv \times (1 + rate)^{nper} + pmt(1 + rate \times type) \times \left(\frac{(1 + rate)^{nper} - 1}{rate} \right) + fv = 0$$

- 속도가 0이면 다음을 수행합니다.

$$(pmt \times nper) + pv + fv = 0$$

- 형식이 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.

- nper < 1 또는 blank
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
\$500.00	매달 말에 보험 연금에서 지불 한 돈.
8%	지급된 돈으로 얻은 이자율입니다.
20	몇 년 동안 돈이 지급됩니다.

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    PV(0.08/12, 12*20, 500.00, 0, 0)
}
```

위에 지정된 용어를 사용하여 연금의 현재 값을 반환합니다.

[값]
-59777.1458511878

피드백

이 페이지가 도움이 되었나요?

Yes

No

속도

아티클 • 2023. 10. 20.

연금 기간당 이자율을 반환합니다. RATE는 반복을 통해 계산되며 솔루션이 0개 이상 있을 수 있습니다. RATE의 연속 결과가 20회 반복 후 0.0000001 이내로 수렴되지 않으면 오류가 반환됩니다.

구문

DAX

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]]])
```

매개 변수

용어 정의

Nper 연금의 총 지급 기간 수입니다.

Pmt 지불은 각 기간을 만든 연금의 수명 동안 변경할 수 없습니다. 일반적으로 pmt에는 원금과 이자가 포함되지만 다른 수수료나 세금은 포함되지 않습니다.

Pv 현재 값 - 일련의 미래 지불이 지금 가치가 있는 총 금액입니다.

Fv (선택 사항) 미래 가치 또는 마지막 지불 후 달성하려는 현금 잔액입니다. fv를 생략하면 0으로 간주됩니다(예: 대출의 미래 값은 0).

Type (선택 사항) 지불 기한을 나타내는 숫자 0 또는 1입니다. 형식을 생략하면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

추측 (선택 사항) 속도에 대한 당신의 추측.

- 생략하면 10%로 간주됩니다.
- RATE가 수렴되지 않는 경우 추측을 위해 다른 값을 시도합니다. 추측이 0에서 1 사이이면 일반적으로 RATE가 수렴됩니다.

Type 매개 변수는 다음 값을 허용합니다.

형식을 다음과 같게 설정합니다.

결제가 기한인 경우

0 또는 생략됨

마침표가 끝날 때

1

기간의 시작 부분에서

Return Value

기간당 이자율입니다.

설명

- 추측 및 nper를 지정하는 데 사용하는 단위에 대해 일관성이 있는지 확인합니다. 연간 이자가 12%인 4년 대출금을 매월 상환하는 경우 추측에는 0.12/12, nper의 경우 4*12를 사용합니다. 동일한 대출에 대해 연간 상환을 하는 경우 추측에는 0.12, nper에는 4를 사용합니다.
- 형식이 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - nper ≤ 0 .
 - RATE는 20번의 반복 후 0.0000001 이내로 수렴되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
4	대출 년
200-	월별 결제
8000	대출 금액

예 1

다음 DAX 쿼리:

```
DAX  
EVALUATE  
{  
    RATE(4*12, -200, 8000)  
}
```

위에 지정된 조건을 사용하여 대출의 월별 이자율을 반환합니다.

[값]

0.00770147248820137

예제 2

다음 DAX 쿼리:

DAX

```
EVALUATE
{
    RATE(4*12, -200, 8000) * 12
}
```

위에 지정된 조건을 사용하여 대출의 연간 이자율을 반환합니다.

[값]

0.0924176698584164

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

배송 완료

아티클 • 2023. 10. 20.

완전 투자 유가 증권의 만기 시 받은 금액을 반환합니다.

구문

DAX

```
RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
투자	유가 증권에 투자한 금액입니다.
할인	보안의 할인율입니다.
기저	(선택 사항) 사용할 수 있는 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

만기 시 받은 금액입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- RECEIVED는 다음과 같이 계산됩니다.

$$\text{RECEIVED} = \frac{\text{investment}}{1 - (\text{discount} \times \frac{\text{DIM}}{\text{B}})}$$

여기서

- B = 연도 기준으로 1년 단위의 일 수입니다.
- DIM = 문제부터 만기까지의 일 수입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - 투자 ≤ 0 .
 - 할인 ≤ 0 .
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

Data	설명
2월 15일-2월 8일	결산(문제) 날짜
5월 15일-5월 8일	만기 일
\$1,000,000.00	투자
5.75%	할인율
2	실제/360 기준

DAX

```
EVALUATE
{
    RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

위에 지정된 조건의 채권에 대해 만기 시 받을 총 금액을 반환합니다.

[값]

1014584.6544071

피드백

이 페이지가 도움이 되었나요?

Yes

No

RRI

아티클 • 2023. 10. 20.

투자 증가에 해당하는 이자율을 반환합니다.

구문

DAX

`RRI(<nper>, <pv>, <fv>)`

매개 변수

용어	정의
Nper	투자 기간 수입니다.
Pv	투자의 현재 가치입니다.
Fv	투자의 미래 가치.

Return Value

동등한 이자율입니다.

설명

- RRI는 다음 수식을 사용하여 계산된 nper(기간 수), Pv(현재 값) 및 Fv(미래 값)가 지정된 이자율을 반환합니다.

$$\left(\frac{FV}{PV} \right)^{\left(\frac{1}{nper} \right)} - 1$$

- 다음과 같은 경우 오류가 반환됩니다.
 - nper ≤ 0.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
\$10,000	현재 값
\$21,000	미래 값
4	투자한 연도

다음 DAX 쿼리:

```
DAX  
EVALUATE  
{  
    RRI(4*12, 10000, 21000)  
}
```

위에 지정된 조건을 사용하여 투자 증가에 해당하는 이자율을 반환합니다.

[값]
0.0155771057566627

피드백

이 페이지가 도움이 되었나요?

Yes

No

SLN

아티클 • 2023. 10. 20.

한 기간 동안 자산의 직선 감가 상각액을 반환합니다.

구문

DAX

`SLN(<cost>, <salvage>, <life>)`

매개 변수

용어	정의
비용	초기 자산 비용입니다.
구조	감가상각 종료 시의 값입니다(자산의 회수 값이라고도 함).
생활	자산이 감가상각되는 기간의 수입니다(자산의 유용한 수명이라고도 함).

Return Value

한 기간 동안의 직선 감가 상각입니다.

설명

- 다음과 같은 경우 오류가 반환됩니다.
 $life = 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
\$30,000	비용
\$7,500	인양 값

Data	설명
10	수년간의 유용한 생활

다음 DAX 쿼리:

DAX
EVALUATE { SLN(30000, 7500, 10) }

위에 지정된 조건을 사용하여 연간 감가상각 수당을 반환합니다.

[값]
2250

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SYD

아티클 • 2023. 10. 20.

지정된 기간 동안 자산의 연수 합계 감가 상각액을 반환합니다.

구문

DAX

```
SYD(<cost>, <salvage>, <life>, <per>)
```

매개 변수

용어 정의

비용 초기 자산 비용입니다.

구조 감가상각 종료 시의 값입니다(자산의 회수 값이라고도 함).

생활 자산이 감가상각되는 기간의 수입니다(자산의 유용한 수명이라고도 함).

per 기간입니다. Life와 동일한 단위를 사용해야 합니다. 1에서 수명(포함) 사이여야 합니다.

Return Value

지정된 기간의 연도 합계 감가상각입니다.

설명

- SYD는 다음과 같이 계산됩니다.

$$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life}) \times (\text{life} + 1)}$$

- 다음과 같은 경우 오류가 반환됩니다.
 - life < 1.
 - per < 1 또는 per > life.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
\$30,000.00	초기 비용
\$7,500.00	인양 값
10	수명(년)

예 1

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 1)
}
```

위에서 지정한 조건에 따라 첫 해에 대한 자산의 연도 합계 감가 상각 허용액을 반환합니다.

[값]
4090.90909090909

예제 2

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 10)
}
```

위에 지정된 조건에 따라 10번째(최종) 연도에 대한 자산의 연도 합계 감가상각 수당을 반환합니다.

[값]

409.090909090909

피드백

이 페이지가 도움이 되었나요?

Yes

No

TBILLEQ

아티클 • 2023. 10. 20.

국채에 해당하는 채권 수익률을 반환합니다.

구문

DAX

TBILLEQ(<settlement>, <maturity>, <discount>)

매개 변수

용어	정의
정착	재무부 법안의 합의 날짜입니다. 유가 증권 결산일은 국채가 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	국채의 만기일입니다. 만기일은 국채가 만료되는 날짜입니다.
할인	국채의 할인율입니다.

Return Value

국채의 채권 등가 수익률.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- TBILLEQ는 다음과 같이 계산됩니다.

$$TBILLEQ = \frac{365 \times \text{discount}}{360 - (\text{discount} \times \text{DSM})}$$

여기서

- DSM은 연간 360일 기준으로 계산된 결산과 만기 사이의 일 수입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - 만기 또는 만기 \geq 결산 후 1년이 넘습니다.
 - 할인 ≤ 0 .
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
3/31/2008	결산일
6/1/2008	만기일
9.14%	할인율

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    TBILLEQ(DATE(2008,3,31), DATE(2008,6,1), 0.0914)
}
```

위에 지정된 조건을 사용하여 국채에 해당하는 채권 수익률을 반환합니다.

[값]
0.094151493565943

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TBILLPRICE

아티클 • 2023. 10. 20.

국채의 액면가 \$100당 가격을 반환합니다.

구문

DAX

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

매개 변수

용어	정의
정착	재무부 법안의 합의 날짜입니다. 유가 증권 결산일은 국채가 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	국채의 만기일입니다. 만기일은 국채가 만료되는 날짜입니다.
할인	국채의 할인율입니다.

Return Value

국채의 액면가 \$100당 가격입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- TBILLPRICE는 다음과 같이 계산됩니다.

$$\text{TBILLPRICE} = 100 \times \left(1 - \frac{\text{discount} \times \text{DSM}}{360}\right)$$

여기서

- DSM = 결산일부터 만기까지의 일 수이며, 결산일 이후 1년이 넘는 만기일은 제외됩니다.
- settlement 및 maturity는 정수로 잘립니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - 만기 또는 만기 \geq 결산 후 1년이 넘습니다.
 - 할인 ≤ 0 .
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
3/31/2008	결산일
6/1/2008	만기일
9.0%	할인율

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    TBILLPRICE(DATE(2008,3,31), DATE(2008,6,1), 0.09)
}
```

위에 지정된 조건을 사용하여 국채의 액면가 \$100당 가격을 반환합니다.

[값]
98.45

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

트빌리엘드

아티클 • 2023. 10. 20.

국채 수익률을 반환합니다.

구문

DAX

TBILLYIELD(<settlement>, <maturity>, <pr>)

매개 변수

용어	정의
정착	재무부 법안의 합의 날짜입니다. 유가 증권 결산일은 국채가 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	국채의 만기일입니다. 만기일은 국채가 만료되는 날짜입니다.
pr	국채의 액면가 \$100당 가격입니다.

Return Value

국채 수익률.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- TBILLYIELD는 다음과 같이 계산됩니다.

$$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

여기서

- DSM = 결산일부터 만기까지의 일 수이며, 결산일 이후 1년이 넘는 만기일은 제외됩니다.
- settlement 및 maturity는 정수로 잘립니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - 만기 또는 만기 \geq 결산 후 1년이 넘습니다.
 - $pr \leq 0$.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리:

Data	설명
3/31/2008	결산일
6/1/2008	만기일
\$98.45	액면가 \$100당 가격

```
DAX
EVALUATE
{
    TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45)
}
```

위에 지정된 조건을 사용하여 국채 수익률을 반환합니다.

[값]
0.0914169629253426

피드백

이 페이지가 도움이 되었나요?

Yes

No

VDB

아티클 • 2023. 10. 20.

이중 감소 잔액 메서드 또는 지정한 다른 방법을 사용하여 부분 기간을 포함하여 지정한 기간 동안 자산의 감가 상각액을 반환합니다. VDB는 가변 감소 잔액을 의미합니다.

구문

DAX

```
VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[, <no_switch>]])
```

매개 변수

용어	정의
비용	초기 자산 비용입니다.
구조	감가상각 종료 시의 값입니다(자산의 회수 값이라고도 함). 이 값은 0일 수 있습니다.
생활	자산이 감가상각되는 기간의 수입니다(자산의 유용한 수명이라고도 함).
start_period	감가 상각을 계산할 시작 기간입니다. Start_period 생명과 동일한 단위를 사용해야 합니다. 1에서 수명(포함) 사이여야 합니다.
end_period	감가 상각을 계산할 종료 기간입니다. End_period 수명과 동일한 단위를 사용해야 합니다. start_period 수명(포함) 사이에 있어야 합니다.
요소	(선택 사항) 잔액이 감소하는 비율입니다. 인수를 생략하면 2(이중 감소 잔액 방법)로 간주됩니다. 이중 감소 잔액 방법을 사용하지 않으려면 인수를 변경합니다. 이중 감소 잔액 방법에 대한 설명은 DDB를 참조하세요.
no_switch	(선택 사항) 감가 상각액이 감가상각 계산보다 클 때 직선 감가 상각으로 전환할지 여부를 지정하는 논리 값입니다. 생략하면 FALSE로 간주됩니다. - no_switch TRUE로 평가되면 감가 상각액이 감가상각 계산보다 큰 경우에도 VDB는 직선 감가 상각으로 전환되지 않습니다. - no_switch FALSE로 평가되거나 생략되면 감가 상각액이 감가상각 계산보다 크면 VDB가 직선 감가 상각으로 전환됩니다.

Return Value

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 TRUE 또는 FALSE로 평가되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

Data	설명
2400	초기 비용
300	인양 값
10	수명(년)

예 1

다음 DAX 쿼리:

```
DAX  
EVALUATE  
{  
    VDB(2400, 300, 10*365, 0, 1)  
}
```

2의 인수를 사용하여 자산의 첫 날의 감가 상각액을 반환합니다.

[값]

1.31506849315068

예제 2

다음 DAX 쿼리:

```
DAX  
  
EVALUATE  
{  
    VDB(2400, 300, 10*12, 6, 18, 3)  
}
```

6^{th} 월과 18^{th} 월 사이의 자산의 감가 상각액을 반환합니다. 이 계산에서는 3단계를 사용합니다.

[값]

540.185558199698

예 3

다음 DAX 쿼리:

```
DAX  
  
EVALUATE  
{  
    VDB(2400, 300, 10, 0, 0.875, 1.5)  
}
```

세법이 감소 잔액의 150% 감가상각으로 제한한다고 가정하면 소유한 첫 회계 연도에 자산의 감가 상각액을 반환합니다. 자산은 회계 연도의 1 분기 중간에 구입됩니다.

[값]

315

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

XIRR

아티클 • 2023. 12. 14.

반드시 주기적인 것은 아닌 현금 흐름 일정에 대한 내부 수익률을 반환합니다.

구문

DAX

```
XIRR(<table>, <values>, <dates>, [, <guess>[, <alternateResult>]])
```

매개 변수

[+] 테이블 확장

용어	정의
table	값 및 날짜 식을 계산해야 하는 테이블입니다.
values	테이블의 각 행에 대한 현금 흐름 값을 반환하는 식입니다.
날짜	테이블의 각 행에 대한 현금 흐름 날짜를 반환하는 식입니다.
추측	(선택 사항) 내부 수익률에 대한 초기 추측입니다. 생략하면 0.1의 기본 추측이 사용됩니다.
alternateResult	(선택 사항) 솔루션을 확인할 수 없는 경우 오류 대신 반환되는 값입니다.

반환 값

지정된 입력에 대한 내부 반환 속도입니다. 계산에서 유효한 결과를 반환하지 못하면 alternateResult로 지정된 오류 또는 값이 반환됩니다.

설명

- 값은 다음 함수를 충족하는 속도로 계산됩니다.

$$0 = \sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

여기서

- P_j 는 j^{th} 번째 지급
- d_j 는 j^{th} 번째 지급일
- d_1 은 첫 번째 지급일
- 일련의 현금 흐름 값에는 하나 이상의 양수와 1개의 음수가 포함되어야 합니다.
- ISERROR 또는 IFERROR 함수를 사용하여 XIRR에서 반환된 오류를 캡처하지 않습니다. 함수에 대한 일부 입력으로 인해 솔루션 오류가 발생하지 않을 수 있는 경우 alternateResult 매개 변수를 제공하는 것이 오류를 처리하는 가장 안정적이고 성능이 뛰어난 방법입니다.
- alternateResult 매개 변수를 사용하는 방법에 대한 자세한 내용을 알아보려면 이 [비디오](#)를 확인하세요.
- 초기 지불의 절대값이 작으면 계산에서 유효한 결과를 반환하지 못할 수 있습니다.
- 지불에 0 값을 포함하지 않습니다. 최종 결과에는 영향을 주지 않으며 초기 결제로 0을 사용하면 항상 XIRR() 계산에 실패합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 수식은 CashFlows 테이블의 내부 수익률을 계산합니다.

DAX

```
= XIRR( CashFlows, [Payment], [Date] )
```

 테이블 확장

날짜	지불
1/1/2014	10000-
3/1/2014	2750
10/30/2014	4250
2/15/2015	3250
4/1/2015	2750

수익률 = 37.49%

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

XNPV

아티클 • 2023. 10. 20.

반드시 주기적인 것은 아닌 현금 흐름 일정에 대한 현재 값을 반환합니다.

구문

DAX

```
XNPV(<table>, <values>, <dates>, <rate>)
```

매개 변수

용어	정의
table	값 및 날짜 식을 계산해야 하는 테이블입니다.
values	테이블의 각 행에 대한 현금 흐름 값을 반환하는 식입니다.
날짜	테이블의 각 행에 대한 현금 흐름 날짜를 반환하는 식입니다.
rate	테이블의 각 행에 대한 현금 흐름에 적용할 할인율입니다.

반환 값

순 현재 값을입니다.

설명

- 값은 다음 합계로 계산됩니다.

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

여기서

- P_j 는 j^{th} 번째 지급
- d_j 는 j^{th} 번째 지급일
- d_1 은 첫 번째 지급일

- 일련의 현금 흐름 값에는 하나 이상의 양수와 1개의 음수가 포함되어야 합니다.

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

예시

다음은 CashFlows 테이블의 현재 값을 계산합니다.

DAX	
= XNPV (CashFlows, [Payment], [Date], 0.09)	
날짜	지불
1/1/2014	10000-
3/1/2014	2750
10/30/2014	4250
2/15/2015	3250
4/1/2015	2750

현재 값 = 2086.65

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

항복

아티클 • 2023. 10. 20.

정기적인 이자를 지불하는 유가 증권의 수익률을 반환합니다. YIELD를 사용하여 채권 수익률을 계산합니다.

구문

DAX

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

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
rate	보안의 연간 쿠폰 비율입니다.
pr	유가 증권의 액면가 \$100당 가격입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
frequency	연간 쿠폰 지급 수입니다. 연간 지불의 경우 빈도 = 1; 반기, 빈도 = 2; 분기별로, frequency = 4입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

기초	일 수 기준
4	유럽 30/360

Return Value

유가 증권의 수익률입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- 상환까지 쿠폰 기간이 하나 이하인 경우 YIELD는 다음과 같이 계산됩니다.

$$\text{YIELD} = \frac{\text{redemption} - (\text{par} + (\text{A}/\text{E}) \times (\text{rate}/\text{frequency}))}{(\text{par} + (\text{A}/\text{E}) \times (\text{rate}/\text{frequency})) \times \text{frequency}} \times \text{DSR}$$

여기서

- A = 쿠폰 기간의 시작부터 결산 날짜(누적 일)까지의 일 수입니다.
- DSR = 결산 날짜부터 상환 날짜까지의 일 수입니다.
- E = 쿠폰 기간의 일 수입니다.
- 상환까지 쿠폰 기간이 두 개 이상 있는 경우 YIELD는 100회 반복을 통해 계산됩니다. 해상도는 함수 PRICE에 사용되는 수식에 따라 Newton 메서드를 사용합니다. 수익률이 가격에 근접한 경우 예상 가격이 될 때까지 수익률이 변경됩니다.
- settlement 및 maturity는 정수로 잘립니다.
- frequency 및 basis는 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement ≥ maturity.

- $rate < 0$.
- $pr \leq 0$.
- 상환 ≤ 0 .
- frequency는 1, 2 또는 4 이외의 숫자입니다.
- basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
2월 15일-2월 8일	결산일
15-11월 16일	만기일
5.75%	백분율 쿠폰
95.04287	가격
\$100	상환 값
2	빈도는 반기(위 참조)입니다.
0	30/360 기준(위 참조)

다음 DAX 쿼리:

```
DAX
EVALUATE
{
    YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

위에 지정된 조건을 사용하여 채권의 수익률을 반환합니다.

[값]
0.0650000068807314

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

YIELDDISC

아티클 • 2023. 10. 20.

할인된 보안에 대한 연간 수익률을 반환합니다.

구문

DAX

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

매개 변수

용어 정의	
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
pr	유가 증권의 액면가 \$100당 가격입니다.
보너스	유가 증권의 액면가 \$100당 상환액입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

연간 수익률입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement 및 maturity는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement 또는 maturity는 유효한 날짜가 아닙니다.
 - settlement \geq maturity.
 - $pr \leq 0$.
 - 상환 ≤ 0 .
 - basis < 0 또는 basis > 4.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	-
2월 16일-2월 8일	결산일
3월 1일	만기일
99.795	가격
\$100	상환 값
2	실제/360 기준

다음 DAX 쿼리:

DAX

```
EVALUATE
{
    YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2)
}
```

위에 지정된 조건에 따라 유가 증권의 연간 수익률을 반환합니다.

[값]

0.0528225719868583

피드백

이 페이지가 도움이 되었나요?

Yes

No

YIELDMAT

아티클 • 2023. 10. 20.

만기 시 이자를 지급하는 유가 증권의 연간 수익률을 반환합니다.

구문

DAX

```
YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

매개 변수

용어	정의
정착	보안의 합의 날짜입니다. 보안 결산 날짜는 유가 증권이 구매자에게 거래되는 발행일 이후의 날짜입니다.
성숙	보안의 만기일입니다. 만기일은 보안이 만료되는 날짜입니다.
이슈	보안의 발급 날짜입니다.
rate	발행일 기준 유가 증권의 이자율입니다.
pr	유가 증권의 액면가 \$100당 가격입니다.
기저	(선택 사항) 사용할 일 수 기준의 유형입니다. 기준이 생략되면 0으로 간주됩니다. 허용되는 값은 이 표 아래에 나열되어 있습니다.

basis 매개 변수는 다음 값을 허용합니다.

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

Return Value

연간 수익률입니다.

설명

- 날짜는 순차적인 일련 번호로 저장되므로 계산에 사용할 수 있습니다. DAX에서는 1899년 12월 30일이 0일이고 2008년 1월 1일은 39448년 12월 30일 이후 39,448일 이므로 1899년 12월 30일입니다.
- 결산일은 구매자가 채권과 같은 쿠폰 구매하는 날짜입니다. 만기일은 쿠폰 만료되는 날짜입니다. 예를 들어 2008년 1월 1일에 30년 만기 채권이 발행되고 6개월 후에 구매자가 매입한다고 가정해 보겠습니다. 발행일은 2008년 1월 1일, 결산일은 2008년 7월 1일, 만기일은 발행일 2008년 1월 1일 이후 30년인 2038년 1월 1일입니다.
- settlement, maturity 및 issue는 정수로 잘립니다.
- 은 가장 가까운 정수로 반올림됩니다.
- 다음과 같은 경우 오류가 반환됩니다.
 - settlement, maturity 또는 issue는 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Data	설명
15-3월 8일	결산일
11월 3일-11월 8일	만기일
11월 8일-11월 7일	문제 날짜
6.25%	반기 쿠폰 백분율
100.0123	가격
0	30/360 기준(위 참조)

다음 DAX 쿼리:

DAX

```
EVALUATE
{
    YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625,
    100.0123, 0)
}
```

위에 지정된 용어를 사용하여 보안에 대한 수익률을 반환합니다.

[값]

0.0609543336915387

피드백

이 페이지가 도움이 되었나요?

Yes

No

정보 함수

아티클 • 2023. 10. 20.

DAX 정보 함수는 인수로 제공되는 셀 또는 행을 살펴보고 값이 예상 형식과 일치하는지 여부를 알려줍니다. 예를 들어 ISERROR 함수는 참조하는 값에 오류가 포함되어 있는 경우 TRUE를 반환합니다.

이 범주에서

함수	설명
COLUMNSTATISTICS	모델의 모든 테이블의 모든 열에 대한 통계 테이블을 반환합니다.
포함	참조된 모든 열에 대한 값이 있거나 해당 열에 포함된 경우 true를 반환합니다. 그렇지 않으면 함수가 false를 반환합니다.
CONTAINSROW	값 행이 있거나 테이블에 포함된 경우 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.
CONTAINSSTRING	한 문자열에 다른 문자열이 포함되어 있는지 여부를 나타내는 TRUE 또는 FALSE를 반환합니다.
CONTAINSSTRINGEXACT	한 문자열에 다른 문자열이 포함되어 있는지 여부를 나타내는 TRUE 또는 FALSE를 반환합니다.
Customdata	연결 문자열 CustomData 속성의 내용을 반환합니다.
HASONEFILTER	columnName에서 직접 필터링된 값의 수가 1이면 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.
HASONEVALUE	columnName의 컨텍스트가 하나의 고유 값으로만 필터링 된 경우 TRUE를 반환합니다. 그렇지 않으면 FALSE입니다.
ISAFTER	Start At 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 충족하는 행에 대해 true를 반환하는 부울 함수입니다.
ISBLANK	값이 비어 있는지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.
ISCROSSFILTERED	같은 테이블이나 관련 테이블의 columnName 또는 다른 열을 필터링할 때 TRUE를 반환합니다.
Isempty	테이블이 비어 있는지 확인합니다.
ISERROR	값이 오류인지 확인하고 TRUE 또는 FALSE를 반환합니다.

함수	설명
ISEVEN	숫자가 짝수이면 TRUE를 반환하고, 숫자가 홀수이면 FALSE를 반환합니다.
ISFILTERED	columnName을 직접 필터링할 때 TRUE를 반환합니다.
ISINSCOPE	지정된 열이 수준 계층 구조의 수준인 경우 true를 반환합니다.
ISLOGICAL	값이 논리 값인지(TRUE 또는 FALSE)인지 확인하고 TRUE 또는 FALSE를 반환합니다.
ISNONTEXT	값이 텍스트가 아닌지(빈 셀이 텍스트가 아님) TRUE 또는 FALSE를 반환하는지 확인합니다.
Isnumber	값이 숫자인지 확인하고 TRUE 또는 FALSE를 반환합니다.
ISODD	숫자가 홀수이면 TRUE를 반환하고, 숫자가 짝수이면 FALSE를 반환합니다.
ISONORAFTER	Start At 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 총족하는 행에 대해 true를 반환하는 부울 함수입니다.
ISSELECTEDMEASURE	계산 항목에 대한 식에서 컨텍스트에 있는 측정값을 결정하는 데 사용되는 측정값은 측정값 목록에 지정된 측정값 중 하나입니다.
ISSUBTOTAL	인수로 지정된 열에 대한 부분합 값이 행에 포함된 경우 True를 반환하는 SUMMARIZE 식에 다른 열을 만들고, 그렇지 않으면 False를 반환합니다.
ISTEXT	값이 텍스트인지 확인하고 TRUE 또는 FALSE를 반환합니다.
비시각적	SUMMARIZECOLUMNS 식의 값 필터를 시각적 개체가 아닌 것으로 표시합니다.
SELECTEDMEASURE	계산 항목에 대한 식에서 컨텍스트에 있는 측정값을 참조하는 데 사용됩니다.
SELECTEDMEASUREFORMATSTRING	계산 항목에 대한 식에서 컨텍스트에 있는 측정값의 형식 문자열을 검색하는 데 사용됩니다.
SELECTEDMEASURENAME	계산 항목에 대한 식에서 이름으로 컨텍스트에 있는 측정값을 결정하는 데 사용됩니다.
USERCULTURE	현재 사용자의 로캘을 반환합니다.
USERNAME	연결 시 시스템에 지정된 자격 증명에서 do기본 이름 및 사용자 이름을 반환합니다.
USEROBJECTID	현재 사용자의 개체 ID 또는 SID를 반환합니다.

합수	설명
USERPRINCIPALNAME	사용자 계정 이름을 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COLUMNSTATISTICS

아티클 • 2023. 10. 20.

모델의 모든 테이블의 모든 열에 대한 통계 테이블을 반환합니다.

구문

DAX

COLUMNSTATISTICS ()

매개 변수

이 함수는 매개 변수를 사용하지 않습니다.

반환 값

통계 테이블입니다. 이 테이블의 각 행은 모델의 다른 열을 나타냅니다. 테이블 열에는 다음과이 포함됩니다.

- **테이블 이름:** 현재 열의 테이블입니다.
- **열 이름:** 현재 열의 이름입니다.
- **최소:** 현재 열에 있는 최소값입니다.
- **Max:** 현재 열에 있는 최대값입니다.
- **카디널리티:** 현재 열에 있는 고유 값의 수입니다.
- **최대 길이:** 현재 열에 있는 가장 긴 문자열의 길이입니다(문자열 열에만 적용됨).

설명

- 오류 상태의 열과 쿼리 범위 계산 테이블의 열은 결과 테이블에 나타나지 않습니다.
- 필터 컨텍스트의 필터가 COLUMNSTATISTICS()에 적용되면 오류가 반환됩니다.

예시

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

다음 DAX 쿼리:

DAX

```
DEFINE
    TABLE FilteredProduct =
        FILTER (
            Product,
            [Color] == "Blue"
        )
    COLUMN Customer[Location] = [State-Province] & " " & [Country-Region]

EVALUATE
COLUMNSTATISTICS ()
```

모델의 모든 테이블의 모든 열에 대한 통계가 있는 테이블을 반환합니다. 이 테이블에는 쿼리 범위 계산 열인 Customer[Location]에 대한 통계도 포함되어 있습니다. 그러나 테이블에는 쿼리 범위 계산 테이블 FilteredProduct의 열이 포함되지 않습니다.

Table Name	Column Name	Min	Max	Cardinality	Max Length
Customer	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			18485	
Customer	CustomerKey	-1	29483	18485	
Customer	Customer ID	[Not Applicable]	AW00029483	18485	16
Customer	Customer	[Not Applicable]	Zoe Watson	18401	26
Customer	City	[Not Applicable]	York	270	21
Customer	State-Province	[Not Applicable]	Yveline	54	19
Customer	Country-Region	[Not Applicable]	United States	7	16
Customer	Postal Code	[Not Applicable]	YO15	324	16
Customer	Location	[Not Applicable]	Yveline France	54	33
Date	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			1461	
Date	DateKey	20170701	20210630	1461	
Date	Date	7/1/2017 12:00:00 AM	6/30/2021 12:00:00 AM	1461	
Date	Fiscal Year	FY2018	FY2021	4	6
Date	Fiscal Quarter	FY2018 Q1	FY2021 Q4	16	9
Date	Month	2017 Aug	2021 May	48	8
Date	MonthKey	201707	202106	48	
Date	Full Date	2017 Aug, 01	2021 May, 31	1461	12
Sales Territory	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			11	
Sales Territory	SalesTerritoryKey	1	11	11	

참고 항목

[필터 컨텍스트](#)

[CALCULATETABLE 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CONTAINS

아티클 • 2023. 10. 20.

참조된 모든 열에 대한 값이 있거나 해당 열에 포함된 경우 true를 반환합니다. 그렇지 않으면 함수가 false를 반환합니다.

구문

DAX

```
CONTAINS(<table>, <columnName>, <value>[ , <columnName>, <value>]...)
```

매개 변수

용어	정의
table	데이터 테이블을 반환하는 DAX 식입니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.
value	columnName에서 검색할 단일 스칼라 값을 반환하는 DAX 식입니다. 인수 목록에 전달되기 전에 식을 정확히 한 번 계산해야 합니다.

반환 값

지정된 각 값을 해당 columnName에서 찾을 수 있거나 해당 열에 포함된 경우 TRUE 값을입니다. 그렇지 않으면 함수는 FALSE를 반환합니다.

설명

- columnName 인수와 같은 쌍으로 제공되어야 합니다. 그렇지 않으면 오류가 반환됩니다.
- columnName은 지정된 테이블 또는 테이블과 관련된 테이블에 속해야 합니다.
- columnName이 관련 테이블의 열을 참조하는 경우 정규화되어야 합니다. 그렇지 않으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 제품 214 및 고객 11185의 인터넷 판매가 동시에 발생했는지 여부를 알려주는 측정값을 만듭니다.

DAX

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

CONTAINSROW 함수

아티클 • 2023. 12. 14.

모든 열에 지정된 값이 있는 행이 하나 이상 있을 경우 TRUE를 반환합니다.

구문

DAX

```
CONTAINSROW(<Table>, <Value> [, <Value> [, ...] ] )
```

매개 변수

[\[\]](#) 테이블 확장

용어	정의
테이블	테스트할 테이블입니다.
값	스칼라 값을 반환하는 유효한 DAX 식입니다.

반환 값

TRUE 또는 FALSE입니다.

설명

- 구문을 제외하고 IN 연산자와 CONTAINSROW 함수는 기능적으로 동일합니다.

DAX

```
<scalarExpr> IN <tableExpr>
( <scalarExpr1>, <scalarExpr2>, ... ) IN <tableExpr>
```

- scalarExprN의 수는 tableExpr의 열 수와 일치해야 합니다.
- NOT IN은 DAX의 연산자가 아닙니다. IN 연산자의 논리적 부정을 수행하려면 전체 식 앞에 NOT을 배치합니다. 예를 들어 NOT [Color] IN { "Red", "Yellow", "Blue" }.

- = 연산자와 달리 IN 연산자와 CONTAINSROW 함수는 엄격한 비교를 수행합니다. 예를 들어 BLANK 값은 0과 일치하지 않습니다.

예제

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

예 1

다음 DAX 쿼리:

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    ( [Color] )
        IN {
            "Red",
            "Yellow",
            "Blue"
        }
)
ORDER BY [Color]
```

및

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
        [Color]
    )
)
ORDER BY [Color]
```

단일 열이 있는 다음 테이블을 반환합니다.

[색]
파랑
빨간색
노란색

예제 2

해당하는 DAX 쿼리는 다음과 같습니다.

DAX
<pre>EVALUATE FILTER (ALL (Product[Color]), NOT [Color] IN { "Red", "Yellow", "Blue" }) ORDER BY [Color]</pre>

및

DAX
<pre>EVALUATE FILTER (ALL (Product[Color]), NOT CONTAINSROW ({ "Red", "Yellow", "Blue" }, [Color])) ORDER BY [Color]</pre>

단일 열이 있는 다음 테이블을 반환합니다.

[색]
검정
회색
다색
해당 없음
은
Silver\Black
백인

관련 콘텐츠

[IN 연산자](#)

[DAX 쿼리](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CONTAINSSTRING

아티클 • 2023. 10. 20.

한 문자열에 다른 문자열이 포함되어 있는지 여부를 나타내는 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
CONTAINSSTRING(<within_text>, <find_text>)
```

매개 변수

용어	정의
within_text	find_text 검색할 텍스트입니다.
find_text	찾을 텍스트입니다.

반환 값

true이면 find_text within_text; 하위 문자열입니다. 그렇지 않으면 FALSE입니다.

설명

- CONTAINSSTRING은 대/소문자를 구분하지 않습니다.
- 사용할 수 있으며 * 카드 문자를 사용할 수 있습니다?. 야생 카드 문자를 이스케이프하는 데 사용합니다~.

예시

DAX 쿼리

DAX

```
EVALUATE  
ROW(  
    "Case 1", CONTAINSSTRING("abcd", "bc"),
```

```
    "Case 2", CONTAINSSTRING("abcd", "BC"),
    "Case 3", CONTAINSSTRING("abcd", "a*d"),
    "Case 4", CONTAINSSTRING("abcd", "ef")
)
```

반환

[사례 1]	[사례 2]	[사례 3]	[사례 4]
TRUE	TRUE	TRUE	FALSE

피드백

이 페이지가 도움이 되었나요?

Yes

No

CONTAINSSTRINGEXACT

아티클 • 2023. 10. 20.

한 문자열에 다른 문자열이 포함되어 있는지 여부를 나타내는 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

매개 변수

용어	정의
within_text	find_text 검색할 텍스트입니다.
find_text	찾을 텍스트입니다.

반환 값

true이면 find_text within_text; 하위 문자열입니다. 그렇지 않으면 FALSE입니다.

설명

CONTAINSSTRINGEXACT는 대/소문자를 구분합니다.

예시

DAX 쿼리

DAX

```
EVALUATE  
ROW(  
    "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),  
    "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),  
    "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),  
    "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
```

)

반환

[사례 1]	[사례 2]	[사례 3]	[사례 4]
TRUE	FALSE	거짓	FALSE

피드백

이 페이지가 도움이 되었나요?

Yes

No

CUSTOMDATA

아티클 • 2023. 10. 20.

연결 문자열 CustomData 속성의 내용을 반환합니다.

구문

DAX

```
CUSTOMDATA()
```

반환 값

연결 문자열 CustomData 속성의 콘텐츠입니다.

CustomData 속성이 연결 시 정의되지 않은 경우 비어 있습니다.

설명

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

예시

다음 DAX 수식은 CustomData 속성이 "OK"로 설정되었는지 확인합니다.

DAX

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No  
custom data in connection string property or unexpected value")
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

HASONEFILTER

아티클 • 2023. 10. 20.

columnName에서 직접 필터링된 값의 수가 1이면 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.

구문

DAX

HASONEFILTER(<columnName>)

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.

반환 값

columnName에서 직접 필터링된 값의 수가 1이면 TRUE이고, 그렇지 않으면 FALSE를 반환합니다.

설명

- 이 함수는 HASONEVALUE()가 교차 필터를 기반으로 작동하는 반면 HASONEFILTER()는 직접 필터에서 작동한다는 차이가 있는 HASONEVALUE()와 비슷합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

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.

DAX

```
=  
IF(HASONEFILTER(ResellerSales_USD[ProductKey]), FILTERS(ResellerSales_USD[Pro  
ductKey]), BLANK())
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

HASONEVALUE

아티클 • 2023. 10. 20.

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 TRUE를 반환합니다. 그렇지 않으면 FALSE입니다.

구문

HTML

```
HASONEVALUE(<columnName>)
```

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.

반환 값

columnName의 컨텍스트가 하나의 고유 값으로만 필터링된 경우 TRUE입니다. 그렇지 않으면 FALSE입니다.

설명

- HASONEVALUE()에 해당하는 식은 `COUNTROWS(VIEWVALUES(<columnName>)) = 1`.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 측정값 수식은 미리 정의된 시나리오에 대한 백분율을 예측하기 위해 컨텍스트가 하나의 값으로 분할되는지 확인합니다. 이 경우 Reseller Sales를 2007년의 판매와 비교하려는 경우 컨텍스트가 1년 단위로 필터링되는지 알아야 합니다. 또한 비교가 의미가 없는 경우 BLANK를 반환하려고 합니다.

DAX

```
=  
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])/  
CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=  
2007),BLANK())
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISAFTER

아티클 • 2023. 12. 14.

'Start At' 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 충족하는 행에 대해 true를 반환하는 부울 함수입니다.

정렬 순서에 따라 첫 번째 매개 변수는 두 번째 매개 변수와 비교됩니다. 정렬 순서가 오름차순인 경우 수행할 비교는 두 번째 매개 변수보다 큰 첫 번째 매개 변수입니다. 정렬 순서가 내림차순인 경우 수행할 비교는 첫 번째 매개 변수보다 작은 두 번째 매개 변수입니다.

구문

DAX

```
ISAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

매개 변수

 테이블 확장

용어	정의
스칼라 식	열 참조 또는 정수 또는 문자열 값과 같은 스칼라 값을 반환하는 식입니다. 일반적으로 첫 번째 매개 변수는 열 참조이고 두 번째 매개 변수는 스칼라 값입니다.
정렬 순서	(선택 사항) 열이 정렬되는 순서입니다. ASC(오름차순) 또는 내림차순(DEC)일 수 있습니다. 기본적으로 정렬 순서는 오름차순입니다.

반환 값

true 또는 false.

설명

이 함수는 ISONORAFTER와 유사합니다. 차이점은 ISAFTER가 필터 값 이후에 엄격하게 정렬된 값에 대해 true를 반환하고, 여기서 ISONORAFTER는 필터 값에 대해 정렬된 값에 대해 true를 반환합니다.

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

예시

테이블 이름: '정보'

[+] 테이블 확장

국가/지역	State(상태)	Count	합계
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900
미국	CA	5	500
미국	WA	10	900

다음 식은 다음과 같습니다.

DAX

```
FILTER (
    Info,
    ISAFTER (
        Info[Country], "IND", ASC,
        Info[State], "MH", ASC )
)
```

HRESULT = NO_ERROR를

[+] 테이블 확장

Country	State(상태)	Count	합계
IND	Wb	10	900
미국	CA	5	500
미국	WA	10	900

관련 콘텐츠

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISBLANK

아티클 • 2023. 12. 14.

값이 비어 있는지 여부를 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
ISBLANK(<value>)
```

매개 변수

 테이블 확장

용어	정의
value	테스트할 값 또는 식입니다.

반환 값

값이 비어 있으면 TRUE의 부울 값입니다. 그렇지 않으면 FALSE입니다.

설명

BLANKS 를 사용할 때 모범 사례에 대해 자세히 알아보려면 DAX에서 BLANK를 값으로 변환하지 마십시오.

예시

이 수식은 전년 대비 매출 증가 또는 감소 비율을 계산합니다. 이 예제에서는 IF 함수를 사용하여 오류 0으로 나누기를 방지하기 위해 이전 연도 판매의 값을 검사.

DAX

```
//Sales to Previous Year Ratio  
  
= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])  
    , BLANK()  
    , ( 'CalculatedMeasures'[Total Sales] -
```

```
'CalculatedMeasures'[PreviousYearTotalSales] )  
/ 'CalculatedMeasures'[PreviousYearTotalSales])
```

결과

[+] 테이블 확장

Row Labels	총 판매 수	전년도 총 판매액	전년 대비 판매액 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		

관련 콘텐츠

정보 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISCROSSFILTERED

아티클 • 2023. 10. 20.

지정한 테이블 또는 열이 교차 필터링될 경우 TRUE를 반환합니다.

구문

DAX

`ISCROSSFILTERED(<TableNameOrColumnName>)`

매개 변수

용어	정의
TableNameOrColumnName	기준 테이블 또는 열의 이름입니다. 식일 수 없습니다.

반환 값

ColumnName 또는 *TableName*의 열이 교차 필터링될 경우 TRUE를 반환합니다. 그렇지 않으면 FALSE를 반환합니다.

설명

- 필터가 *ColumnName*, *TableName*의 열 또는 관련 테이블의 열에 적용되면 열 또는 테이블이 교차 필터링된다고 말합니다.
- 필터가 *ColumnName* 또는 *TableName*의 열에 적용되면 열 또는 테이블이 직접 필터링된다고 말합니다. 그러므로 *ColumnName* 또는 *TableName*의 열이 필터링되면 [ISFILTERED](#) 함수도 TRUE를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[ISFILTERED](#) 함수

[FILTERS](#) 함수

HASONEFILTER 함수

HASONEVALUE 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISEMPTY

아티클 • 2023. 10. 20.

테이블이 비어 있는지 확인합니다.

구문

DAX

`ISEMPTY(<table_expression>)`

매개 변수

용어	정의
table_expression	테이블을 반환하는 테이블 참조 또는 DAX 식입니다.

반환 값

테이블이 비어 있으면 True이고(행이 없는 경우), 그렇지 않으면 False입니다.

설명

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

예시

아래 표의 경우 'Info'입니다.

국가/지역	주	군	합계
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900
미국	CA	5	500

국가/지역	주	군	합계
미국	WA	10	900

DAX

```
EVALUATE  
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info,  
[County]>25))))
```

반환 값: FALSE

피드백

이 페이지가 도움이 되었나요?

Yes

No

ISERROR

아티클 • 2023. 12. 14.

값이 오류인지 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

DAX

`ISERROR(<value>)`

매개 변수

 테이블 확장

용어	정의
value	테스트할 값입니다.

반환 값

값이 오류인 경우 TRUE의 부울 값입니다. 그렇지 않으면 FALSE입니다.

설명

- ISERROR를 사용하는 경우 모범 사례는 오류 함수의 적절한 사용을 참조 [하세요](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 총 인터넷 판매와 총 재판매인 판매의 비율을 계산합니다. ISERROR 함수는 0으로 나누기 등의 오류를 검사 데 사용됩니다. 빈 값이 반환되는 오류가 있으면 비율이 반환됩니다.

DAX

```
= IF( ISERROR(
    SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
    )
, BLANK()
, SUM('ResellerSales_USD'[SalesAmount_USD])
/SUM('InternetSales_USD'[SalesAmount_USD])
)
```

관련 콘텐츠

[정보 함수](#)

[IFERROR 함수](#)

[IF 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISEVEN

아티클 • 2023. 10. 20.

숫자가 짹수이면 TRUE를 반환하고, 숫자가 홀수이면 FALSE를 반환합니다.

구문

DAX

`ISEVEN(number)`

매개 변수

용어 정의

번호 테스트할 값입니다. number가 정수가 아니면 잘립니다.

반환 값

숫자가 짹수이면 TRUE를 반환하고, 숫자가 홀수이면 FALSE를 반환합니다.

설명

- number가 숫자가 아니면 ISEVEN은 #VALUE 반환합니다. 오류 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

ISFILTERED

아티클 • 2023. 10. 20.

지정한 테이블 또는 열이 직접 필터링될 경우 TRUE를 반환합니다.

구문

DAX

`ISFILTERED(<TableNameOrColumnName>)`

매개 변수

용어	정의
TableNameOrColumnName	기준 테이블 또는 열의 이름입니다. 식일 수 없습니다.

반환 값

ColumnName 또는 *TableName*의 열이 직접 필터링될 경우 TRUE를 반환합니다. 그렇지 않으면 FALSE를 반환합니다.

설명

- 필터가 *ColumnName* 또는 *TableName*의 열에 적용되면 열 또는 테이블이 직접 필터링된다고 말합니다.
- 필터가 *ColumnName*, *TableName*의 열 또는 관련 테이블의 열에 적용되면 열 또는 테이블이 교차 필터링된다고 말합니다. 그러므로 *ColumnName*, *TableName*의 열 또는 관련 테이블의 열이 필터링되면 [ISCROSSFILTERED](#) 함수도 TRUE를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

참고 항목

[ISCROSSFILTERED](#) 함수

[FILTERS](#) 함수

HASONEFILTER 함수

HASONEVALUE 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISINSCOPE

아티클 • 2023. 12. 14.

지정된 열이 수준 계층 구조의 수준인 경우 true를 반환합니다.

구문

DAX
ISINSCOPE(<columnName>)

매개 변수

 테이블 확장

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.

반환 값

지정된 열이 수준 계층 구조의 수준인 경우 TRUE입니다.

설명

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

예시

DAX
DEFINE MEASURE FactInternetSales[% of Parent] = SWITCH (TRUE(), ISINSCOPE(DimProduct[Subcategory]), DIVIDE(SUM(FactInternetSales[Sales Amount]), CALCULATE(SUM(FactInternetSales[Sales Amount]), ALLSELECTED(DimProduct[Subcategory]))) , ISINSCOPE(DimProduct[Category]),

```

DIVIDE(
    SUM(FactInternetSales[Sales Amount]),
    CALCULATE(
        SUM(FactInternetSales[Sales Amount]),
        ALLSELECTED(DimProduct[Category]))
),
1
) * 100
EVALUATE
SUMMARIZECOLUMNS
(
ROLLUPADDISSUBTOTAL
(
    DimProduct[Category], "Category Subtotal",
    DimProduct[Subcategory], "Subcategory Subtotal"
),
TREATAS(
    {"Bike Racks", "Bike Stands", "Mountain Bikes", "Road Bikes", "Touring
Bikes"}, DimProduct[Subcategory]),
"Sales", SUM(FactInternetSales[Sales Amount]),
"% of Parent", [% of Parent]
)
ORDER BY
[Category Subtotal] DESC, [Category],
[Subcategory Subtotal] DESC, [Subcategory]

```

반환

 테이블 확장

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	Touring Bikes	FALSE	FALSE	3,844,801.05	13.58

관련 콘텐츠

SUMMARIZECOLUMNS 함수

CALCULATE 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISLOGICAL

아티클 • 2023. 12. 14.

값이 논리 값인지(TRUE 또는 FALSE)인지 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
ISLOGICAL(<value>)
```

매개 변수

 테이블 확장

용어	정의
value	테스트할 값입니다.

반환 값

값이 논리 값이면 TRUE이고, FALSE 또는 FALSE 이외의 값이 있으면 FALSE입니다.

설명

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

예시

다음 세 가지 샘플은 ISLOGICAL의 동작을 보여줍니다.

DAX

```
//RETURNS: Is Boolean type or Logical  
= IF(ISLOGICAL(true), "Is Boolean type or Logical", "Is different type")  
  
//RETURNS: Is Boolean type or Logical  
= IF(ISLOGICAL(false), "Is Boolean type or Logical", "Is different type")
```

```
//RETURNS: Is different type  
= IF(ISLOGICAL(25), "Is Boolean type or Logical", "Is different type")
```

관련 콘텐츠

정보 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISNONTEXT

아티클 • 2023. 12. 14.

값이 텍스트가 아닌지(빈 셀이 텍스트가 아님) TRUE 또는 FALSE를 반환하는지 확인합니다.

구문

DAX

`ISNONTEXT(<value>)`

매개 변수

[+] 테이블 확장

용어	정의
value	검사 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 ISNONTEXT 함수의 동작을 보여 줍니다.

DAX

```
//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")
```

관련 콘텐츠

정보 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISNUMBER

아티클 • 2023. 12. 14.

값이 숫자인지 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
ISNUMBER(<value>)
```

매개 변수

 테이블 확장

용어	정의
value	테스트할 값입니다.

반환 값

값이 숫자이면 TRUE이고, 그렇지 않으면 FALSE입니다.

설명

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

예시

다음 세 가지 샘플은 ISNUMBER의 동작을 보여줍니다.

DAX

```
//RETURNS: Is number  
= IF(ISNUMBER(0), "Is number", "Is Not number")  
  
//RETURNS: Is number  
= IF(ISNUMBER(3.1E-1), "Is number", "Is Not number")
```

```
//RETURNS: Is Not number  
= IF(ISNUMBER("123"), "Is number", "Is Not number")
```

관련 콘텐츠

[정보 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISODD

아티클 • 2023. 10. 20.

숫자가 홀수이면 TRUE를 반환하고, 숫자가 짝수이면 FALSE를 반환합니다.

구문

DAX

`ISODD(number)`

매개 변수

용어 정의

번호 테스트할 값입니다. number가 정수가 아니면 잘립니다.

반환 값

숫자가 홀수이면 TRUE를 반환하고, 숫자가 짝수이면 FALSE를 반환합니다.

설명

- number가 숫자가 아니면 ISODD는 #VALUE 반환합니다. 오류 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

ISONORAFTER

아티클 • 2023. 12. 14.

Start At 절의 동작을 에뮬레이트하고 모든 조건 매개 변수를 충족하는 행에 대해 true를 반환하는 부울 함수입니다.

정렬 순서에 따라 첫 번째 매개 변수는 두 번째 매개 변수와 비교됩니다. 정렬 순서가 오름차순인 경우 수행할 비교는 두 번째 매개 변수보다 큰 첫 번째 매개 변수입니다. 정렬 순서가 내림차순인 경우 수행할 비교는 첫 번째 매개 변수보다 작은 두 번째 매개 변수입니다.

구문

DAX

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]]...)
```

매개 변수

 테이블 확장

용어	정의
스칼라 식	열 참조 또는 정수 또는 문자열 값과 같은 스칼라 값을 반환하는 식입니다. 일반적으로 첫 번째 매개 변수는 열 참조이고 두 번째 매개 변수는 스칼라 값입니다.
정렬 순서	(선택 사항) 열이 정렬되는 순서입니다. 오름차순(ASC)이거나 내림차순(DESC)일 수 있습니다. 기본적으로 정렬 순서는 오름차순입니다.

반환 값

true 또는 false.

설명

이 함수는 ISAFTER와 유사합니다. 차이점은 ISONORAFTER가 필터 값에 정렬된 값에 대해 true를 반환하고, 여기서 ISAFTER는 필터 값 이후에 엄격하게 정렬된 값에 대해 true를 반환합니다.

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

예시

Info라는 이름의 다음 테이블을 사용한다고 가정합니다.

[+] 테이블 확장

국가/지역	State(상태)	Count	합계
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900
미국	CA	5	500
미국	WA	10	900

다음 식은 다음과 같습니다.

DAX

```
FILTER (
    Info,
    ISFILTERED (
        Info[Country], "IND", ASC,
        Info[State], "MH", ASC
    )
)
```

HRESULT = NO_ERROR를

[+] 테이블 확장

국가/지역	State(상태)	Count	합계
IND	MH	25	1000
IND	Wb	10	900
미국	CA	5	500
미국	WA	10	900

관련 콘텐츠

ISAFTER

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISSELECTEDMEASURE

아티클 • 2023. 10. 20.

계산 항목에 대한 식에서 컨텍스트에 있는 측정값을 결정하는 데 사용되는 측정값은 측정값 목록에 지정된 측정값 중 하나입니다.

구문

DAX

```
ISSELECTEDMEASURE( M1, M2, ... )
```

매개 변수

용어	정의
M1, M2, ...	측정값 목록입니다.

반환 값

현재 컨텍스트에 있는 측정값이 매개 변수 목록에 지정된 측정값 중 하나인지 여부를 나타내는 부울입니다.

설명

- 계산 항목에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 계산 항목 식은 현재 측정값이 매개 변수 목록에 지정된 측정값 중 하나인 경우 검사. 측정값의 이름을 바꾸면 수식 수정에 식의 이름 변경 내용이 반영됩니다.

DAX

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
```

```
SELECTEDMEASURE (),  
DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )  
)
```

참고 항목

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISSUBTOTAL

아티클 • 2023. 10. 20.

인수로 지정된 열에 대한 부분합 값이 행에 포함된 경우 True를 반환하는 SUMMARIZE 식에 다른 열을 만들고, 그렇지 않으면 False를 반환합니다.

구문

DAX

```
ISSUBTOTAL(<columnName>)
```

SUMMARIZE를 사용하여

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[,  
ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...]])[, <name>,  
{<expression>|ISSUBTOTAL(<columnName>)}]...)
```

매개 변수

용어	정의
columnName	SUMMARIZE 함수 테이블의 열 이름 또는 테이블과 관련된 테이블의 열 이름입니다.

반환 값

인수로 지정된 열에 대한 부분합 값이 행에 포함되어 있으면 True 값이고, 그렇지 않으면 False를 반환합니다.

설명

- 이 함수는 SUMMARIZE [함수의](#) 식에서만 사용할 수 있습니다.
- 이 함수 앞에 부울 열의 이름이 있어야 합니다.

예시

요약을 참조하세요.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISTEXT

아티클 • 2023. 12. 14.

값이 텍스트인지 확인하고 TRUE 또는 FALSE를 반환합니다.

구문

DAX

```
ISTEXT(<value>)
```

매개 변수

 테이블 확장

용어	정의
value	검사 값입니다.

반환 값

값이 텍스트이면 TRUE이고, 그렇지 않으면 FALSE입니다.

설명

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

예시

다음 예제에서는 ISTEXT 함수의 동작을 보여 줍니다.

DAX

```
//RETURNS: Is Text  
= IF(ISTEXT("text"), "Is Text", "Is Non-Text")  
  
//RETURNS: Is Text  
= IF(ISTEXT(""), "Is Text", "Is Non-Text")
```

```
//RETURNS: Is Non-Text  
= IF(ISTEXT(1), "Is Text", "Is Non-Text")  
  
//RETURNS: Is Non-Text  
= IF(ISTEXT(BLANK()), "Is Text", "Is Non-Text")
```

관련 콘텐츠

정보 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NONVISUAL

아티클 • 2023. 10. 20.

SUMMARIZECOLUMNS 식의 값 필터를 [시각적 개체가 아닌 것으로 표시합니다](#). 이 함수는 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.

구문

DAX

`NONVISUAL(<expression>)`

매개 변수

용어	정의
식	테이블이 아닌 단일 값을 반환하는 DAX 식입니다.

반환 값

값의 테이블입니다.

설명

- SUMMARIZECOLUMNS의 값 필터를 측정값에 영향을 주지 않고 그룹별 열에만 적용되도록 표시합니다.
- 이 함수는 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.
SUMMARIZECOLUMNS 함수의 [filterTable](#) 인수 또는 [ROLLUPADDISSTOTAL](#) 또는 [ROLLUPISSUBTOTAL](#) 함수의 [groupLevelFilter](#) 인수로 사용됩니다.

예시

SUMMARIZECOLUMNS를 참조 [하세요](#).

피드백

이 페이지가 도움이 되었나요?

Yes

No

SELECTEDMEASURE

아티클 • 2023. 12. 14.

계산 항목 또는 동적 형식 문자열에 대한 식에서 컨텍스트에 있는 측정값을 참조하는 데 사용됩니다.

구문

DAX

```
SELECTEDMEASURE()
```

매개 변수

없음

반환 값

계산 항목 또는 형식 문자열이 계산될 때 현재 컨텍스트에 있는 측정값에 대한 참조입니다.

설명

- 계산 항목 또는 형식 문자열에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 계산 항목 식은 측정값이 컨텍스트에 있는 모든 항목에 대한 연도별 날짜를 계산합니다.

DAX

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

다음 식을 사용하여 값이 수백, 수천 또는 수백만인지에 따라 측정값의 형식 문자열을 동적으로 조정할 수 있습니다.

DAX

```
SWITCH(  
    TRUE(),  
    SELECTEDMEASURE() < 1000, "$#,##0", //Values less than 1000 have  
    no text after them  
    SELECTEDMEASURE() < 1000000, "$#,##0,.0 K", //Values between 1000 and  
    1000000 are formatted as #.## K  
    "$#,##0,,.0 M" //Values greater than 1000000  
    are formatted as #.## M  
)
```

관련 콘텐츠

[SELECTEDMEASURENAME](#)
[ISSELECTEDMEASURE](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SELECTEDMEASUREFORMATSTRING

아티클 • 2023. 10. 20.

계산 항목에 대한 식에서 컨텍스트에 있는 측정값의 형식 문자열을 검색하는 데 사용됩니다.

구문

DAX

```
SELECTEDMEASUREFORMATSTRING()
```

매개 변수

없음

반환 값

계산 항목이 계산될 때 현재 컨텍스트에 있는 측정값의 형식 문자열을 포함하는 문자열입니다.

설명

- 이 함수는 계산 그룹의 계산 항목에 대한 식에서만 참조할 수 있습니다. 계산 항목의 **Format String Expression** 속성에서 사용하도록 설계되었습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 식은 계산 항목에 대한 Format String Expression 속성에 의해 계산됩니다. 필터 컨텍스트에 단일 통화가 있는 경우 DimCurrency[FormatString] 열에서 형식 문자열이 검색됩니다. 그렇지 않으면 컨텍스트에서 측정값의 형식 문자열이 사용됩니다.

DAX

```
SELECTEDVALUE(DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING())
```

참고 항목

SELECTEDMEASURE

ISSELECTEDMEASURE

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SELECTEDMEASURENAME

아티클 • 2023. 12. 14.

계산 항목에 대한 식에서 이름으로 컨텍스트에 있는 측정값을 결정하는 데 사용됩니다.

구문

DAX

```
SELECTEDMEASURENAME()
```

매개 변수

없음

반환 값

계산 항목이 계산될 때 현재 컨텍스트에 있는 측정값의 이름을 포함하는 문자열 값입니다.

설명

- 계산 항목에 대한 식에서만 참조할 수 있습니다.
- 이 함수는 종종 계산 그룹을 작성할 때 디버깅 목적으로 사용됩니다.

예시

다음 계산 항목 식은 현재 측정값이 Expense Ratio이고 조건부로 계산 논리를 적용하는지 검사. 검사 문자열 비교를 기반으로 하므로 수식 수정이 적용되지 않으며 자동으로 반영되는 개체 이름 바꾸기에는 도움이 되지 않습니다. 수식 수정에 도움이 되는 유사한 비교는 ISSELECTEDMEASURE 함수를 대신 참조하세요.

DAX

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
```

```
DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )  
)
```

관련 콘텐츠

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

USERCULTURE

아티클 • 2023. 10. 20.

운영 체제, 브라우저 설정 또는 Power BI 서비스에 따라 결정되는 현재 사용자의 로캘(언어 코드-국가 코드)을 반환합니다.

참고: 이 함수는 현재 용량 단위 Power BI Premium, 사용자 단위 Power BI Premium, Power BI Embedded에서만 지원됩니다.

구문

DAX

`USERCULTURE()`

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

문자열로 로캘을 지정합니다.

설명

- Power BI 서비스 로캘은 **설정>언어>설정**에 의해 결정됩니다. 기본값은 사용자의 브라우저 언어 설정에 의해 결정됩니다.
- 계산 테이블 및 계산 열 식에서 사용되는 경우 테이블이 DirectQuery 또는 가져오기 모드인지에 따라 결과가 다를 수 있습니다. DirectQuery 모드에서 결과는 Power BI 서비스 언어 설정에 지정된 언어(로캘)에 따라 결정됩니다. 언어 설정의 기본값 로캘은 사용자의 브라우저 언어 설정에 따라 결정됩니다. 즉, 동일한 계산 테이블 또는 열이 각 사용자의 브라우저 언어 설정에 따라 다른 결과를 반환할 수 있습니다. 가져오기 모드에서 결과는 새로 고침 중에 정적으로 결정되며 쿼리 시간에는 달라지지 않습니다. 예약 또는 대화형과 같은 관리되는 새로 고침의 경우 로캘은 사용자의 브라우저 언어 설정을 기반으로 하지 않고 고정 로캘을 대신 사용합니다. 그러나 XMLA 엔드포인트를 사용하여 사용자 지정 로캘을 지정하여 고정 로캘을 재정의할 수 있습니다.

- Power BI의 필드 매개 변수 기능과 결합하면 USERCULTURE를 사용하여 동일한 모델 내에서 측정값 및 RLS(행 수준 보안) 개체 식에 사용될 때 동적 시각화 제목 및 캡션 안정적으로 변환할 수 있습니다. 그러나 쿼리 및 라이브 연결 보고서 측정값과 같이 모델 외부에서 호출된 USERCULTURE를 포함하는 식은 올바르게 번역된 제목 및 캡션 의존해서는 안 됩니다.
- USERCULTURE는 측정값, RLS(행 수준 보안) 및 계산 항목과 같은 모델 내에서 호출된 개체 식에 사용되는 경우 올바른 사용자 로캘을 반환합니다. 그러나 쿼리 및 라이브 연결 보고서 측정값과 같이 모델 외부의 식에서 사용되는 경우 올바른 사용자 로캘을 반환하지 않을 수 있습니다.
- 라이브 연결 보고서에서 USERCULTURE는 보고서 측정값 식에서 호출할 때 올바른 사용자 로캘을 반환하지 않을 수 있습니다.

예시

다음 식의 경우,

DAX

```
FORMAT(TODAY(), "dddd", USERCULTURE())
```

현재 사용자의 언어 설정에 따라 USERCULTURE가 현재 요일을 반환합니다. 예를 들면 다음과 같습니다.

로캘	서식이 지정된 평일
de-DE	Dienstag
en-US	Tuesday
es-ES_tradnl	martes
eu-ES	asteartea
lt-IT	martedì
nl-NL	dinsdag
pl-PL	wtorek
ro-RO	marți
ru-RU	вторник
uk-UA	вівторок

참고 항목

Power BI Desktop의 식 기반 제목

USERNAME

USERPRINCIPALNAME

USEROBJECTID

피드백

이 페이지가 도움이 되었나요?

Yes

No

USERNAME

아티클 • 2023. 10. 20.

연결 시 시스템에 지정된 자격 증명에서 기본 이름 및 사용자 이름을 반환합니다.

구문

DAX

```
USERNAME()
```

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

연결 시 시스템에 제공된 자격 증명의 사용자 이름

예시

다음 수식은 사용자 로그인이 UsersTable의 일부인지 확인합니다.

DAX

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

USEROBJECTID

아티클 • 2023. 10. 18.

Microsoft Entra ID 또는 SID(보안 식별자)에서 현재 사용자의 개체 ID를 반환합니다.

구문

DAX

```
USEROBJECTID()
```

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

Power BI 또는 Azure Analysis Services 모델 또는 SQL Server Analysis Services 모델의 SID에 대한 Microsoft Entra ID의 현재 사용자 개체 ID입니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

USERPRINCIPALNAME

아티클 • 2023. 10. 20.

사용자 계정 이름을 반환합니다.

구문

DAX

USERPRINCIPALNAME()

매개 변수

이 식에는 매개 변수가 없습니다.

반환 값

연결 시 userprincipalname입니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

논리 함수

아티클 • 2023. 10. 20.

논리 함수는 식에 따라 동작하여 식의 값 또는 집합에 대한 정보를 반환합니다. 예를 들어 IF 함수를 사용하여 식의 결과를 검사 조건부 결과를 만들 수 있습니다.

이 범주에서

함수	설명
및	두 인수가 모두 TRUE인지 확인하고 두 인수가 모두 TRUE이면 TRUE를 반환합니다.
BITAND	두 숫자의 비트 'AND'를 반환합니다.
BITLSHIFT	지정된 비트 수만큼 왼쪽으로 이동한 숫자를 반환합니다.
BITOR	두 숫자의 비트 'OR'를 반환합니다.
BITRSHIFT	지정된 비트 수만큼 오른쪽으로 이동한 숫자를 반환합니다.
BITXOR	두 숫자의 비트 'XOR'를 반환합니다.
합체	BLANK로 계산되지 않는 첫 번째 식을 반환합니다.
FALSE	논리값 FALSE를 반환합니다.
경우	조건을 확인하고 TRUE이면 한 값을 반환하고, 그렇지 않으면 두 번째 값을 반환합니다.
경우. 열망	조건을 확인하고 TRUE이면 한 값을 반환하고, 그렇지 않으면 두 번째 값을 반환합니다. 조건 식에 관계없이 항상 분기 식을 실행하는 즉시 실행 계획을 사용합니다.
IFERROR	식이 오류를 반환하는 경우 식을 계산하고 지정된 값을 반환합니다.
다음이 아 님	FALSE를 TRUE로 변경하거나 TRUE를 FALSE로 변경합니다.
OR	TRUE를 반환하기 위해 인수 중 하나가 TRUE인지 여부를 확인합니다.
스위치	값 목록에 대해 식을 평가하고 가능한 여러 결과 식 중 하나를 반환합니다.
TRUE	논리값 TRUE를 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

AND

아티클 • 2024. 01. 18.

두 인수가 모두 TRUE인지 확인하고 두 인수가 모두 TRUE이면 TRUE를 반환합니다. 그렇지 않은 경우 false를 반환합니다.

구문

DAX

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

매개 변수

[+] 테이블 확장

용어	정의
logical_1, logical_2	테스트할 논리 값입니다.

반환 값

테스트하는 값의 조합에 따라 true 또는 false를 반환합니다.

설명

DAX의 AND 함수는 두 개의 인수만 허용합니다. 여러 식에 대해 AND 연산을 수행해야 하는 경우 일련의 계산을 만들거나 AND 연산자(&>)를 사용하여 더 간단한 식으로 모든 계산을 조인할 수 있습니다.

예 1

다음 수식은 AND 함수의 구문을 보여줍니다.

DAX

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

인수로 전달된 두 조건 모두 AND 함수에 true이므로 수식은 "All True"를 반환합니다.

예제 2

다음 샘플에서는 중첩된 수식과 함께 AND 함수를 사용하여 두 개의 계산 집합을 동시에 비교합니다. 각 제품 범주에 대해 수식은 인터넷 채널의 현재 연도 판매액과 전년도 판매액이 같은 기간의 재판매인 채널보다 큰지 여부를 결정합니다. 두 조건이 모두 true이면 각 범주에 대해 수식이 "Internet hit" 값을 반환합니다.

DAX

```
= IF( AND( SUM('InternetSales_USD'[SalesAmount_USD])  
        >SUM('ResellerSales_USD'[SalesAmount_USD])  
        , CALCULATE(SUM('InternetSales_USD'[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey]))  
        >CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey]))  
        )  
    , "Internet Hit"  
    , ""  
)
```

반환

 테이블 확장

Row Labels	2005	2006	2007	2008	-	총합계
빕 쇼츠						
자전거 랙						
자전거 스탠드					인터넷 적중	
물병 및 거치대					인터넷 적중	
바텀 브라켓						
브레이크						
Caps						
Chains						
Cleaners						
크랭크 세트						
Derailleurs						

Row Labels	2005	2006	2007	2008	-	총합계
Fenders				인터넷 적중		
포크						
장갑						
핸들바						
헤드세트						
Helmets						
Hydration Packs						
Jerseys						
조명						
잠금						
산악용 자전거						
Mountain Frames						
Panniers						
Pedals						
펌프						
도로용 자전거						
Road Frames						
Saddles						
반바지						
양말						
Tights						
Tires and Tubes				인터넷 적중		
Touring Bikes						
Touring Frames						
Vests						
바퀴						

Row Labels	2005	2006	2007	2008	-	총합계
총합계						

관련 콘텐츠

논리 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

BITAND

아티클 • 2023. 12. 14.

두 숫자의 비트 AND를 반환합니다.

구문

DAX

```
BITAND(<number>, <number>)
```

매개 변수

 테이블 확장

용어	정의
숫자	숫자를 반환하는 스칼라 식입니다. 정수가 아니면 잘립니다.

반환 값

두 숫자의 비트 AND입니다.

설명

- 이 함수는 양수와 음수를 모두 지원합니다.

예시

다음 DAX 쿼리:

```
DAX
```

```
EVALUATE { BITAND(13, 11) }
```

9를 반환합니다.

관련 콘텐츠

BITLSHIFT

BITRSHIFT

BITOR

BITXOR

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

BITLSHIFT

아티클 • 2023. 10. 20.

지정된 비트 수만큼 왼쪽으로 이동한 숫자를 반환합니다.

구문

DAX

`BITLSHIFT(<Number>, <Shift_Amount>)`

매개 변수

용어	정의
숫자	정수 식을 반환하는 DAX 식입니다.
Shift_Amount	정수 식을 반환하는 DAX 식입니다.

반환 값

정수 값입니다.

설명

- DAX 비트시프트 함수를 사용하기 전에 정수의 비트시프트 작업 및 오버플로/언더플로의 특성을 이해해야 합니다.
- Shift_Amount 음수이면 반대 방향으로 이동합니다.
- Shift_Amount 절대값이 64보다 크면 오류가 없지만 오버플로/언더플로가 발생합니다.
- Number에는 제한이 없지만 결과가 오버플로/언더플로될 수 있습니다.

예제

예 1

다음 DAX 쿼리:

DAX

```
EVALUATE  
{ BITLSHIFT(2, 3) }
```

16을 반환합니다.

예제 2

다음 DAX 쿼리:

DAX

```
EVALUATE  
{ BITLSHIFT(128, -1) }
```

64를 반환합니다.

예 3

다음 DAX 쿼리:

DAX

```
Define  
Measure Sales[LeftShift] = BITLSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
Sales[Amount],  
"LEFTSHIFT",  
[LeftShift]  
)
```

Shifts는 각 판매액을 3비트로 남겨두고 비트 이동 판매액을 반환합니다.

참고 항목

[BITRSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

BITOR

아티클 • 2023. 12. 14.

두 숫자의 비트 OR을 반환합니다.

구문

DAX

```
BITOR(<number>, <number>)
```

매개 변수

 테이블 확장

용어	정의
숫자	숫자를 반환하는 스칼라 식입니다. 정수가 아니면 잘립니다.

반환 값

두 숫자의 비트 OR입니다.

설명

- 이 함수는 양수와 음수를 모두 지원합니다.

예시

다음 DAX 쿼리:

```
DAX
```

```
EVALUATE
```

```
{ BITOR(9, 10) }
```

11을 반환합니다.

관련 콘텐츠

[BITAND](#)

[BITXOR](#)

[BITLSHIFT](#)

[BITRSHIFT](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

BITRSHIFT

아티클 • 2023. 10. 20.

지정된 비트 수만큼 오른쪽으로 이동한 숫자를 반환합니다.

구문

DAX

BITRSHIFT(<Number>, <Shift_Amount>)

매개 변수

용어	정의
숫자	정수 식을 반환하는 DAX 식입니다.
Shift_Amount	정수 식을 반환하는 DAX 식입니다.

반환 값

정수 값입니다.

설명

- DAX 비트시프트 함수를 사용하기 전에 정수의 비트시프트 작업 및 오버플로/언더플로의 특성을 이해해야 합니다.
- Shift_Amount 음수이면 반대 방향으로 이동합니다.
- Shift_Amount 절대값이 64보다 크면 오류가 없지만 오버플로/언더플로가 발생합니다.
- Number에는 제한이 없지만 결과가 오버플로/언더플로될 수 있습니다.

예제

예 1

다음 DAX 쿼리:

DAX

```
EVALUATE  
{ BITRSHIFT(16, 3) }
```

2를 반환합니다.

예제 2

다음 DAX 쿼리:

DAX

```
EVALUATE  
{ BITRSHIFT(1024, -3) }
```

8192를 반환합니다.

예 3

다음 DAX 쿼리:

DAX

```
Define  
    Measure Sales[RightShift] = BITRSHIFT(SELECTEDVALUE(Sales[Amount]), 3)  
  
EVALUATE  
SUMMARIZECOLUMNS(  
    Sales[Amount],  
    "RIGHTSHIFT",  
    [RightShift]  
)
```

각 판매액을 3비트로 바로 이동하고 비트 이동 판매액을 반환합니다.

참고 항목

[BITLSHIFT](#)

[BITAND](#)

[BITOR](#)

[BITXOR](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

BITXOR

아티클 • 2023. 12. 14.

두 숫자의 비트 XOR를 반환합니다.

구문

DAX

```
BITXOR(<number>, <number>)
```

매개 변수

 테이블 확장

용어	정의
숫자	숫자를 반환하는 스칼라 식입니다. 정수가 아니면 잘립니다.

반환 값

두 숫자의 비트 XOR입니다.

설명

- 이 함수는 양수와 음수를 모두 지원합니다.

예시

다음 DAX 쿼리:

```
DAX
```

```
EVALUATE { BITXOR(9, 10) }
```

3을 반환합니다.

관련 콘텐츠

BITOR

BITAND

BITLSHIFT

BITRSHIFT

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COALESCE

아티클 • 2023. 10. 20.

BLANK로 계산되지 않는 첫 번째 식을 반환합니다. 모든 식이 BLANK로 평가되면 BLANK가 반환됩니다.

구문

DAX

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

매개 변수

용어	정의
식	스칼라 식을 반환하는 DAX 식입니다.

반환 값

식 중 하나에서 오는 스칼라 값이거나 모든 식이 BLANK로 계산되는 경우 BLANK입니다.

설명

입력 식은 서로 다른 데이터 형식일 수 있습니다.

예 1

다음 DAX 쿼리:

DAX

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

BLANK로 계산되지 않는 첫 번째 식인 를 반환 10 합니다.

예제 2

다음 DAX 식은 다음과 같습니다.

DAX

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

FactInternetSales 테이블의 SalesAmount 열에 있는 모든 값의 합계를 반환합니다 0. 총 판매액의 BLANK 값을 .로 변환하는 0 데 사용할 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FALSE

아티클 • 2023. 12. 14.

논리값 FALSE를 반환합니다.

구문

DAX

```
FALSE()
```

반환 값

항상 FALSE입니다.

설명

FALSE라는 단어도 논리값 FALSE로 해석됩니다.

예시

수식은 'InternetSales_USD'[SalesAmount_USD] 열의 값이 200000보다 작거나 같으면 논리값 FALSE를 반환합니다.

DAX

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

다음 표에서는 예제 수식이 행 레이블의 'ProductCategory'[ProductName]과 열 레이블의 'DateTime'[CalendarYear]과 함께 사용되는 경우의 결과를 보여 줍니다.

 테이블 확장

Row Labels	2005	2006	2007	2008	-	총합계
Accessories	FALSE	FALSE	TRUE	TRUE	거짓	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	거짓	TRUE
의류	FALSE	거짓	거짓	거짓	FALSE	TRUE

Row Labels	2005	2006	2007	2008	-	총합계
구성 요소	FALSE	거짓	거짓	거짓	거짓	거짓
	거짓	거짓	거짓	거짓	거짓	FALSE
총합계	TRUE	TRUE	TRUE	TRUE	거짓	TRUE

관련 콘텐츠

[TRUE 함수](#)

[NOT 함수](#)

[IF 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

IF

아티클 • 2023. 12. 14.

조건을 확인하고 TRUE이면 한 값을 반환하고, 그렇지 않으면 두 번째 값을 반환합니다.

구문

DAX

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

매개 변수

 테이블 확장

용어	정의
logical_test	TRUE 또는 FALSE로 평가할 수 있는 모든 값 또는 식입니다.
value_if_true	논리 테스트가 TRUE이면 반환되는 값입니다.
value_if_false	(선택 사항) 논리 테스트가 FALSE이면 반환되는 값입니다. 생략하면 BLANK가 반환됩니다.

반환 값

value_if_true, value_if_false 또는 BLANK입니다.

설명

- IF 함수는 value_if_true 및 value_if_false 서로 다른 데이터 형식인 경우 변형 데이터 형식을 반환할 수 있지만 value_if_true 및 value_if_false 모두 숫자 데이터 형식인 경우 함수는 단일 데이터 형식을 반환하려고 시도합니다. 후자의 경우 IF 함수는 두 값을 모두 수용하도록 데이터 형식을 암시적으로 변환합니다.

예를 들어 수식 `IF(<condition>, TRUE(), 0)`은 TRUE 또는 0을 반환하지만 value_if_false 정수 데이터 형식인 경우에도 수식 `IF(<condition>, 1.0, 0)`은 소수 값만 반환합니다. 암시적 데이터 형식 변환에 대한 자세한 내용은 데이터 형식을 참조 [하세요](#).

- 조건 식에 관계없이 분기 식을 실행하려면 IF를 사용합니다. [대신 EAGER](#).

예제

다음 Product 테이블 계산 열 정의는 IF 함수를 다양한 방법으로 사용하여 정가에 따라 각 제품을 분류합니다.

첫 번째 예제에서는 List Price 열 값이 500보다 작은지 테스트합니다. 이 조건이 true이면 Low 값이 반환됩니다. value_if_false 값이 없으므로 BLANK가 반환됩니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#) 을 참조하세요.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low"
)
```

두 번째 예제에서는 동일한 테스트를 사용하지만 이번에는 value_if_false 값을 포함합니다. 따라서 수식은 각 제품을 낮음 또는 높음으로 분류합니다.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    "High"
)
```

세 번째 예제에서는 동일한 테스트를 사용하지만 이번에는 IF 함수를 중첩하여 추가 테스트를 수행합니다. 따라서 수식은 각 제품을 낮음, 보통 또는 높음으로 분류합니다.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    IF(
        'Product'[List Price] < 1500,
        "Medium",
        "High"
    )
)
```

```
)  
)
```

💡 팁

여러 IF 함수를 중첩해야 하는 경우 SWITCH **함수가** 더 나은 옵션이 될 수 있습니다. 이 함수는 두 개 이상의 가능한 값을 반환하는 식을 작성하는 보다 세련된 방법을 제공합니다.

관련 콘텐츠

[경우. EAGER 함수](#)

[SWITCH 함수\(DAX\)](#)

[논리 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

IF.EAGER

아티클 • 2023. 10. 20.

조건을 확인하고 TRUE이면 한 값을 반환하고, 그렇지 않으면 두 번째 값을 반환합니다.
조건 식에 관계없이 항상 분기 식을 실행하는 즉시 실행 계획을 사용합니다.

구문

DAX

```
IF.EAGER(<logical_test>, <value_if_true>[, <value_if_false>])
```

매개 변수

용어	정의
logical_test	TRUE 또는 FALSE로 평가할 수 있는 모든 값 또는 식입니다.
value_if_true	논리 테스트가 TRUE이면 반환되는 값입니다.
value_if_false	(선택 사항) 논리 테스트가 FALSE이면 반환되는 값입니다. 생략하면 BLANK가 반환됩니다.

반환 값

value_if_true, value_if_false 또는 BLANK입니다.

설명

- IF입니다. VALUE_IF_TRUE 및 value_if_false 서로 다른 데이터 형식인 경우 EAGER 함수는 변형 데이터 형식을 반환할 수 있지만 value_if_true 및 value_if_false 모두 숫자 데이터 형식인 경우 함수는 단일 데이터 형식을 반환하려고 시도합니다. 후자의 경우 IF입니다. EAGER 함수는 두 값을 모두 수용하도록 데이터 형식을 암시적으로 변환합니다.

예를 들어 수식 `IF.EAGER(<condition>, TRUE(), 0)` 은 TRUE 또는 0을 반환하지만 value_if_false 정수 데이터 형식인 경우에도 수식 `IF.EAGER(<condition>, 1.0, 0)` 은 소수 값만 반환합니다. 암시적 데이터 형식 변환에 대한 자세한 내용은 데이터 형식을 참조 [하세요](#).

- 경우. EAGER는 IF 함수와 동일한 기능 동작을 가지고 있지만 실행 계획의 차이로 인해 성능이 다를 수 있습니다. `IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>)` 에는 다음 DAX 식과 동일한 실행 계획이 있습니다.

DAX

```
VAR _value_if_true = <value_if_true>
VAR _value_if_false = <value_if_false>
RETURN
IF (<logical_test>, _value_if_true, _value_if_false)
```

참고: 두 분기 식은 조건 식에 관계없이 평가됩니다.

예제

IF 예제를 참조 [하세요](#).

참고 항목

[IF 함수](#)

[논리 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

IFERROR

아티클 • 2023. 12. 14.

식이 오류를 반환하면 식을 계산하고 지정된 값을 반환합니다. 그렇지 않으면 식 자체의 값을 반환합니다.

구문

DAX

```
IFERROR(value, value_if_error)
```

매개 변수

[+] 테이블 확장

용어	정의
value	모든 값 또는 식입니다.
value_if_error	모든 값 또는 식입니다.

반환 값

값과 동일한 형식의 스칼라

설명

- IFERROR 함수를 사용하여 식의 오류를 트래핑하고 처리할 수 있습니다.
- 값 또는 value_if_error 빈 셀인 경우 IFERROR는 빈 문자열 값("")으로 처리합니다.
- IFERROR 함수는 IF 함수를 기반으로 하며 동일한 오류 메시지를 사용하지만 인수는 적습니다. IFERROR 함수와 IF 함수 간의 관계는 다음과 같습니다.

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

A 및 B에 대해 반환되는 값은 동일한 데이터 형식이어야 합니다. 따라서 값에 사용되는 열 또는 식과 value_if_error 대해 반환된 값은 동일한 데이터 형식이어야 합니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.
- IFERROR를 사용하는 경우 모범 사례는 오류 함수의 적절한 사용을 참조 [하세요](#).

예시

다음 예제에서는 식 $25/0$ 이 오류로 평가되면 9999를 반환합니다. 식이 오류 이외의 값을 반환하면 해당 값이 호출 식에 전달됩니다.

DAX

```
= IFERROR(25/0,9999)
```

관련 콘텐츠

[논리 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NOT

아티클 • 2023. 12. 14.

FALSE를 TRUE로 변경하거나 TRUE를 FALSE로 변경합니다.

구문

DAX

```
NOT(<logical>)
```

매개 변수

 테이블 확장

용어	정의
논리	TRUE 또는 FALSE로 평가할 수 있는 값 또는 식입니다.

반환 값

TRUE 또는 FALSE입니다.

예시

다음 예제에서는 IF 함수를 설명하기 위해 만든 계산 열에서 값을 검색합니다. 이 예제에서 계산 열의 이름은 기본 이름 **인 Calculated Column1**을 사용하여 지정되었으며 다음 수식을 포함합니다. = IF([Orders]<300,"true","false")

수식은 [Orders] 열의 값을 검사 주문 수가 300 미만이면 "true"를 반환합니다.

이제 새 계산 열인 **계산 열2**를 만들고 다음 수식을 입력합니다.

DAX

```
= NOT([CalculatedColumn1])
```

계산 열1의 각 행에 대해 "true" 및 "false" 값은 논리 값 TRUE 또는 FALSE로 해석되고 NOT 함수는 해당 값의 논리적 반대값을 반환합니다.

관련 콘텐츠

TRUE 함수

FALSE 함수

IF 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

또는

아티클 • 2023. 10. 20.

TRUE를 반환하기 위해 인수 중 하나가 TRUE인지 여부를 확인합니다. 두 인수가 모두 FALSE이면 함수가 FALSE를 반환합니다.

구문

DAX

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

매개 변수

용어	정의
logical_1, logical_2	테스트할 논리 값입니다.

반환 값

부울 값입니다. 두 인수 중 어느 것이 TRUE이면 값이 TRUE입니다. 두 인수가 모두 FALSE이면 값이 FALSE입니다.

설명

- DAX의 **OR 함수는** 두 개의 인수만 허용합니다. 여러 식에 대해 OR 연산을 수행해야 하는 경우 일련의 계산을 만들거나 OR 연산자(||)를 사용하여 더 간단한 식으로 모든 계산을 조인할 수 있습니다.
- 함수는 첫 번째 TRUE 인수까지 인수를 계산한 다음 TRUE를 반환합니다.

예시

다음 예제에서는 OR 함수를 사용하여 우수 둉그라미에 속한 영업 사원을 가져오는 방법을 보여줍니다. 2007년 투어링 바이크 판매 또는 2007년 200만 달러 이상의 매출을 달성한 사람들을 인정합니다.

DAX

```

IF( OR( CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
'ProductSubcategory'[ProductSubcategoryName]="Touring Bikes") > 1000000
      , CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
'DateTime'[CalendarYear]=2007) > 2500000
      )
    , "Circle of Excellence"
    , ""
)

```

반환

Row Labels	2005	2006	2007	2008	-	총합계
아바스, Syed E						
Alberts, Amy E						
안스만 울프, 파멜라 O						
브라이스, 마이클 G	우수 원					
캠벨, 데이비드 R						
카슨, 질리언	우수 원					
Ito, Shu K						
Jiang, Stephen Y						
Mensa-Annan, Tete A						
Mitchell, Linda C	우수 원					
박재 B	우수 원					
Reiter, Tsvi Michael						
사라이바, 호세 에드발도	우수 원					
Tsoflias, Lynn N						
Valdez, Rachel B						
바르가스, 개럿 R						
Varkey Chudukatil, Ranjit R						우수 원
총합계	우수 원					

참고 항목

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

스위치

아티클 • 2023. 10. 20.

값 목록에 대해 식을 평가하고 가능한 여러 결과 식 중 하나를 반환합니다. 이 함수는 중첩된 [IF](#) 문이 여러 개 없는 것을 방지하는 데 사용할 수 있습니다.

구문

DAX

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

매개 변수

용어	정의
식	식을 여러 번 계산할 단일 스칼라 값을 반환하는 DAX 식입니다(각 행/컨텍스트에 대해).
value	식의 결과와 일치시킬 상수 값입니다.
result	식의 결과가 해당 값과 일치하는지 평가할 스칼라 식입니다.
else	식의 결과가 값 인수와 일치하지 않는 경우 계산할 스칼라 식입니다.

반환 값

값과 일치하는 항목이 있으면 해당 결과의 스칼라 값이 반환됩니다. 값과 일치하는 항목이 없으면 다른 값이 반환됩니다. 일치하는 값이 없고 다른 값이 지정되지 않으면 BLANK가 반환됩니다.

설명

- 계산할 식은 상수 값 또는 식일 수 있습니다. 이 함수의 일반적인 용도는 첫 번째 매개 변수를 TRUE로 설정하는 것입니다. 아래 예제를 참조하세요.
- 모든 결과 식과 다른 식은 동일한 데이터 형식이어야 합니다.
- 조건의 순서가 중요합니다. 한 값이 일치하면 해당 결과가 반환되고 다른 후속 값은 평가되지 않습니다. 평가할 가장 제한적인 값이 덜 제한적인 값보다 앞에 지정되어 있는지 확인합니다. 아래 예제를 참조하세요.

예제

SWITCH의 일반적인 용도는 식을 상수 값과 비교하는 것입니다. 다음 예제에서는 월 이름의 계산 열을 만듭니다.

DAX

```
= SWITCH (
    [Month Number Of Year],
    1, "January",
    2, "February",
    3, "March",
    4, "April",
    5, "May",
    6, "June",
    7, "July",
    8, "August",
    9, "September",
    10, "October",
    11, "November",
    12, "December",
    "Unknown month number"
)
```

SWITCH의 또 다른 일반적인 용도는 중첩된 여러 IF 문을 바꾸는 것입니다. 이 작업은 다음 예제와 같이 식을 TRUE로 설정하여 수행됩니다. 이 예제에서는 재고 부족의 잠재적 위험을 식별하기 위해 제품에 대한 재고 수준 다시 정렬 및 금고0을 비교합니다.

DAX

```
= SWITCH (
    TRUE,
    [Reorder Point] > [Safety Stock Level], "Good: Safety stock level exceeded",
    [Reorder Point] = [Safety Stock Level], "Minimal: Safety stock level met",
    [Reorder Point] < [Safety Stock Level], "At risk: Safety stock level not met",
    ISBLANK ( [Reorder Point] ), "Incomplete: Reorder point not set",
    ISBLANK ( [Safety Stock Level] ), "Incomplete: Safety stock level not set",
    "Unknown"
)
```

값의 순서가 중요합니다. 다음 예제에서는 첫 번째 값이 두 번째 값보다 덜 제한적이므로 두 번째 결과가 반환되지 않습니다. 이 예제의 결과는 항상 "A" 또는 "C"이지만 "B"는 없습니다.

DAX

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

다음 문은 결과 인수의 데이터 형식이 다르므로 오류를 반환합니다. 모든 결과 및 기타 인수의 데이터 형식은 동일해야 합니다.

DAX

```
= SWITCH (
    [Class],
    "L", "Large",
    "H", 0.1
)
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TRUE

아티클 • 2023. 12. 14.

논리값 TRUE를 반환합니다.

구문

DAX

```
TRUE()
```

반환 값

항상 TRUE입니다.

설명

TRUE라는 단어도 논리값 TRUE로 해석됩니다.

예시

'InternetSales_USD'[SalesAmount_USD] 열의 값이 200000보다 큰 경우 수식은 논리값 TRUE를 반환합니다.

DAX

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) > 200000, TRUE(), false())
```

다음 표에서는 행 레이블의 'ProductCategory'[ProductName]과 열 레이블의 'DateTime'[CalendarYear]이 있는 보고서에서 예제 수식이 사용되는 경우의 결과를 보여 줍니다.

[\[+\] 테이블 확장](#)

Row Labels	2005	2006	2007	2008	-	총합계
Accessories	FALSE	FALSE	TRUE	TRUE	거짓	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	거짓	TRUE

Row Labels	2005	2006	2007	2008	-	총합계
의류	FALSE	거짓	거짓	거짓	FALSE	TRUE
구성 요소	FALSE	거짓	거짓	거짓	거짓	거짓
	거짓	거짓	거짓	거짓	거짓	FALSE
총합계	TRUE	TRUE	TRUE	TRUE	거짓	TRUE

관련 콘텐츠

FALSE

다음이 아님

경우

피드백

이 페이지가 도움이 되었나요?

Yes

No

수학 및 삼각 함수

아티클 • 2023. 10. 20.

DAX(데이터 분석 식)의 수학 함수는 Excel 수학 및 삼각 함수와 매우 유사합니다. 이 섹션에서는 DAX에서 제공하는 수학 함수를 나열합니다.

이 범주에서

함수	설명
아 bs	숫자의 절대값을 반환합니다.
ACOS	숫자의 아크코사인 또는 역코사인을 반환합니다.
ACOSH	숫자의 역 하이퍼볼릭 코사인을 반환합니다.
ACOT	숫자의 아크코탄젠트 또는 역코탄젠트를 반환합니다.
ACOTH	숫자의 역 하이퍼볼릭 코탄젠트를 반환합니다.
Asin	숫자의 아크사인 또는 역 사인을 반환합니다.
ASINH	숫자의 역 하이퍼볼릭 사인을 반환합니다.
Atan	숫자의 아크탄젠트 또는 역 탄젠트를 반환합니다.
ATANH	숫자의 역 하이퍼볼릭 탄젠트를 반환합니다.
천장	숫자를 가장 가까운 정수 또는 가장 가까운 중요 배수로 올립합니다.
변환	식을 다른 데이터 형식으로 변환합니다.
Cos	지정된 각도의 코사인을 반환합니다.
COSH	숫자의 하이퍼볼릭 코사인을 반환합니다.
침대	라디안으로 지정된 각도의 코탄젠트를 반환합니다.
COTH	쌍곡선 각도의 쌍곡 코탄젠트를 반환합니다.
통화	인수를 평가하고 결과를 통화 데이터 형식으로 반환합니다.
DEGREES	라디안을 도 단위로 변환합니다.
분할	나누기 작업을 수행하고 0으로 나누기에서 대체 결과 또는 BLANK()를 반환합니다.
EVEN	가장 가까운 짹수로 반올림된 숫자를 반환합니다.

함수	설명
EXP	지정된 숫자의 힘으로 e를 반환합니다.
사실	지정된 숫자로 끝나는 계열 $1*2*3*...*$ 에 해당하는 숫자의 계수를 반환합니다.
바닥	숫자를 0으로 내림하여 가장 가까운 중요도 배수로 내림합니다.
Gcd	두 개 이상의 정수 중 가장 큰 공통 수수를 반환합니다.
INT	숫자를 가장 가까운 정수로 내림합니다.
Iso. 천장	숫자를 가장 가까운 정수 또는 가장 가까운 중요 배수로 올립합니다.
Lcm	가장 일반적인 정수 배수를 반환합니다.
Ln	숫자의 자연 로그를 반환합니다.
로그	지정한 기준으로 숫자의 로그를 반환합니다.
LOG10	숫자의 base-10 로그를 반환합니다.
모드	숫자를 수로 나눈 후 remainder를 반환합니다. 결과에는 항상 수수와 동일한 기호가 있습니다.
MROUND	원하는 배수로 반올림된 숫자를 반환합니다.
이상한	가장 가까운 홀수로 반올림된 숫자를 반환합니다.
Pi	15자리까지 정확한 Pi 값인 3.14159265358979를 반환합니다.
전원	숫자의 결과를 출력으로 반환합니다.
몫	나누기를 수행하고 나누기 결과의 정수 부분만 반환합니다.
라디안	각도를 라디안으로 변환합니다.
랜드	0보다 크거나 같고 1보다 작은 난수를 균등하게 분산하여 반환합니다.
RANDBETWEEN	지정한 두 숫자 사이의 범위에서 난수를 반환합니다.
라운드	숫자를 지정된 숫자 수로 반올림합니다.
라운드다운	숫자를 0으로 내림합니다.
검거	숫자를 0(0)에서 위로 반올림합니다.
서명	숫자의 부호, 계산 결과 또는 열의 값을 결정합니다.
죄	지정된 각도의 사인을 반환합니다.
Sinh	숫자의 하이퍼볼릭 사인을 반환합니다.
Sqrt	숫자의 제곱근을 반환합니다.

함수	설명
SQRTPI	(number * pi)의 제곱근을 반환합니다.
tan	지정된 각도의 탄젠트를 반환합니다.
Tanh	숫자의 쌍곡 탄젠트를 반환합니다.
TRUNC	숫자의 소수 또는 소수 부분을 제거하여 숫자를 정수로 자른다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

아 bs

아티클 • 2023. 12. 14.

숫자의 절대값을 반환합니다.

구문

DAX

ABS(<number>)

매개 변수

 테이블 확장

용어	정의
번호	절대값을 원하는 숫자입니다.

반환 값

10진수입니다.

설명

숫자의 절대값은 부호가 없는 10진수(정수 또는 10진수)입니다. ABS 함수를 사용하여 양 수가 필요한 함수에 중첩된 경우 음수가 아닌 숫자만 식에서 반환되도록 할 수 있습니다.

예시

다음 예제에서는 새 계산 열 인 DealerMarkup에서 사용할 수 있는 정가와 딜러 가격 간의 차이의 절대값을 반환합니다.

DAX

= **ABS([DealerPrice]-[ListPrice])**

관련 콘텐츠

수학 및 삼각 함수

SIGN 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ACOS

아티클 • 2023. 10. 20.

숫자의 아크코사인 또는 역코사인을 반환합니다. 아크코사인은 코사인이 숫자인 각도입니다. 반환된 각도는 0에서 pi까지의 범위에서 라디안으로 지정됩니다.

구문

DAX

ACOS(number)

매개 변수

용어	정의
숫자	원하는 각도의 코사인이며 -1에서 1까지여야 합니다.

반환 값

숫자의 아크코사인 또는 역코사인을 반환합니다.

설명

결과를 라디안에서 도로 변환하려면 180/PI()를 곱하거나 DEGREES 함수를 사용합니다.

예시

수식	설명	결과
= ACOS(-0.5)	아크코사인 -0.5 라디안, 2*pi/3.	2.094395102
= ACOS(-0.5)*180/PI()	-0.5도의 아크코사인	120

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ACOSH

아티클 • 2023. 10. 20.

숫자의 역 하이퍼볼릭 코사인을 반환합니다. 숫자는 1보다 크거나 같아야 합니다. 역 하이퍼볼릭 코사인은 하이퍼볼릭 코사인이 숫자인 값이므로 ACOSH(COSH(number))는 숫자와 같습니다.

구문

DAX

`ACOSH(number)`

매개 변수

용어	정의
번호	1보다 크거나 같은 실수입니다.

반환 값

숫자의 역 하이퍼볼릭 코사인을 반환합니다.

설명

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

예시

수식	설명	결과
= ACOSH(1)	역 하이퍼볼릭 코사인 1.	0
= ACOSH(10)	역 하이퍼볼릭 코사인 10.	2.993228

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ACOT

아티클 • 2023. 10. 20.

아크코탄젠트의 주체 값 또는 숫자의 역코탄젠트를 반환합니다.

구문

DAX

ACOT(number)

매개 변수

용어	정의
숫자	원하는 각도의 코사인입니다. 실수여야 합니다.

반환 값

단일 10진수 값입니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

ACOTH

아티클 • 2023. 10. 20.

숫자의 역 하이퍼볼릭 코탄젠트를 반환합니다.

구문

DAX

ACOTH(number)

매개 변수

용어	정의
숫자	Number의 절대값은 1보다 커야 합니다.

반환 값

단일 10진수 값입니다.

설명

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

피드백

이 페이지가 도움이 되었나요?

Yes

No

Asin

아티클 • 2023. 10. 20.

숫자의 아크사인 또는 역 사인을 반환합니다. 아크사인은 사인이 숫자인 각도입니다. 반환된 각도는 $-\pi/2$ 에서 $\pi/2$ 범위의 라디안으로 지정됩니다.

구문

DAX

ASIN(number)

매개 변수

용어	정의
번호	원하는 각도의 사인이며 -1에서 1까지여야 합니다.

반환 값

숫자의 아크사인 또는 역 사인을 반환합니다.

설명

아크사인을 도 단위로 표현하려면 결과를 $180/\text{PI}()$ 로 곱하거나 DEGREES 함수를 사용합니다.

예시

수식	설명	결과
= ASIN(-0.5)	라디안 -0.5의 아크사인, $-\pi/6$	-0.523598776
= ASIN(-0.5)*180/PI()	-0.5도의 아크사인	30-
= DEGREES(ASIN(-0.5))	-0.5도의 아크사인	30-

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ASINH

아티클 • 2023. 10. 20.

숫자의 역 하이퍼볼릭 사인을 반환합니다. 역 하이퍼볼릭 사인은 하이퍼볼릭 사인이 숫자 $\ln(\sqrt{e^x + 1} - \sqrt{e^x - 1})$ 으로 ASINH(SINH(number))는 숫자와 같습니다.

구문

DAX

ASINH(number)

매개 변수

용어	정의
번호	모든 실수입니다.

반환 값

숫자의 역 하이퍼볼릭 사인을 반환합니다.

설명

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

예시

수식	설명	결과
= ASINH(-2.5)	-2.5의 역 하이퍼볼릭 사인	-1.647231146
= ASINH(10)	역 하이퍼볼릭 사인 10	2.99822295

피드백

이 페이지가 도움이 되었나요?

Yes

No

Atan

아티클 • 2023. 10. 20.

숫자의 아크탄젠트 또는 역 탄젠트를 반환합니다. 아크탄젠트는 탄젠트가 숫자인 각도입니다. 반환된 각도는 -pi/2에서 pi/2 범위의 라디안으로 지정됩니다.

구문

DAX

ATAN(number)

매개 변수

용어	정의
번호	원하는 각도의 탄젠트입니다.

반환 값

숫자의 역 하이퍼볼릭 탄젠트를 반환합니다.

설명

아크탄젠트를 도 단위로 표현하려면 결과를 180/PI()로 곱하거나 DEGREES 함수를 사용합니다.

예시

수식	설명	결과
= ATAN(1)	아크탄젠트 1인 라디안, pi/4	0.785398163
= ATAN(1)*180/PI()	아크탄젠트 1도	45

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ATANH

아티클 • 2023. 12. 14.

숫자의 역 하이퍼볼릭 탄젠트를 반환합니다. 숫자는 -1에서 1 사이여야 합니다(-1과 1 제외). 역 하이퍼볼릭 탄젠트는 하이퍼볼릭 탄젠트 수가 숫자인 값이므로 ATANH(TANH(number))는 숫자와 같습니다.

구문

DAX

`ATANH(number)`

매개 변수

 테이블 확장

용어	정의
번호	1에서 -1 사이의 모든 실수입니다.

반환 값

숫자의 역 하이퍼볼릭 탄젠트를 반환합니다.

설명

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

예시

 테이블 확장

수식	Description	결과
= ATANH(0.76159416)	0.76159416의 역 하이퍼볼릭 탄젠트	1.00000001
= ATANH(-0.1)		-0.100335348

수식	Description	결과

관련 콘텐츠

ATAN 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

천장

아티클 • 2023. 10. 20.

숫자를 가장 가까운 정수 또는 가장 가까운 중요 배수로 올림합니다.

구문

DAX

`CEILING(<number>, <significance>)`

매개 변수

용어	정의
번호	반올림할 숫자 또는 숫자가 포함된 열에 대한 참조입니다.
중요성	반올림하려는 중요도의 배수입니다. 예를 들어 가장 가까운 정수로 반올림하려면 1을 입력합니다.

반환 값

지정된 대로 반올림된 숫자입니다.

설명

- DAX에는 다음과 같은 차이점이 있는 두 개의 CEILING 함수가 있습니다.
 - CEILING 함수는 Excel에서 CEILING 함수의 동작을 에뮬레이트합니다.
 - ISO입니다. CEILING 함수는 최대값을 결정하기 위해 ISO 정의 동작을 따릅니다.
- 두 함수는 양수에 대해 동일한 값을 반환하지만 음수에 대해서는 다른 값을 반환합니다. SIGNIFICANCE의 양수 배수를 사용하는 경우 CEILING 및 ISO입니다. CEILING는 음수를 위쪽으로 반올림합니다(양수 무한대쪽으로). 음수의 중요도를 사용하는 경우 CEILING는 음수를 음수 무한대로 반올림하고 ISO는 음수로 반올림합니다. CEILING는 음수를 위쪽으로 반올림합니다(양수 무한대로).
- 반환 형식은 일반적으로 다음과 같은 예외를 제외하고 중요한 인수와 동일한 형식입니다.
 - 숫자 인수 형식이 통화인 경우 반환 형식은 통화입니다.

- significance 인수 형식이 부울이면 반환 형식은 정수입니다.
- significance 인수 형식이 숫자가 아니면 반환 형식이 실수입니다.

예 1

다음 수식은 4.45를 반환합니다. 이는 가격 책정에서 더 작은 단위를 사용하지 않으려면 유용할 수 있습니다. 기존 제품의 가격이 \$4.42인 경우 CEILING을 사용하여 가격을 5센트의 가장 가까운 단위로 반올림할 수 있습니다.

DAX

```
= CEILING(4.42, 0.05)
```

예제 2

다음 수식은 이전 예제와 비슷한 결과를 반환하지만 ProductPrice 열에 저장된 숫자 값을 사용합니다.

DAX

```
= CEILING([ProductPrice], 0.05)
```

참고 항목

[수학 및 삼각 함수](#)

[FLOOR 함수](#)

[Iso. CEILING 함수](#)

[ROUNDUP 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CONVERT

아티클 • 2023. 10. 20.

식을 다른 데이터 형식으로 변환합니다.

구문

DAX

```
CONVERT(<Expression>, <Datatype>)
```

매개 변수

용어	정의
식	유효한 식입니다.
Datatype	INTEGER(정수), DOUBLE(10진수), STRING(텍스트), BOOLEAN(True/False), CURRENCY(고정 10진수), DATETIME(날짜, 시간 등)을 포함하는 열거형입니다.

반환 값

<Expression> 값을 <Datatype>으로 변환해서 반환합니다.

설명

- 값을 지정된 데이터 형식으로 변환할 수 없는 경우 함수는 오류를 반환합니다.
- DAX 계산 열은 단일 데이터 형식이어야 합니다. 정수 열에 대한 MEDIAN 및 MEDIANX 함수는 정수 또는 double의 혼합 데이터 형식을 반환하므로 다음 계산 열 식은 결과적으로 `MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned])` 오류를 반환합니다.
- 혼합 데이터 형식을 방지하려면 식이 항상 이중 데이터 형식을 반환하도록 변경합니다. 예를 들면 다음과 같습니다.

```
MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned],  
DOUBLE)).
```
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

DAX 쿼리

DAX

```
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

반환

[값]

2

피드백

이 페이지가 도움이 되었나요?

Yes

No

Cos

아티클 • 2023. 10. 20.

지정된 각도의 코사인을 반환합니다.

구문

DAX

`COS(number)`

매개 변수

용어 정의

번호 필수입니다. 코사인을 원하는 라디안 단위의 각도입니다.

반환 값

지정된 각도의 코사인을 반환합니다.

설명

각도가 도인 경우 각도를 PI()/180으로 곱하거나 RADIANS 함수를 사용하여 각도를 라디안으로 변환합니다.

예시

수식	설명	결과
= COS(1.047)	코사인 1.047 라디안	0.5001711
= COS(60*PI()/180)	60도의 코사인	0.5
= COS(RADIANS(60))	60도의 코사인	0.5

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COSH

아티클 • 2023. 10. 20.

숫자의 하이퍼볼릭 코사인을 반환합니다.

구문

DAX

`COSH(number)`

매개 변수

용어 정의

번호 필수입니다. 하이퍼볼릭 코사인을 찾으려는 실수입니다.

반환 값

숫자의 하이퍼볼릭 코사인입니다.

설명

- 하이퍼볼릭 코사인의 수식은 다음과 같습니다.

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

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

예시

수식	설명	결과
= COSH(4)	하이퍼볼릭 코사인 4	27.308233
= COSH(EXP(1))	자연 로그의 베이스의 하이퍼볼릭 코사인입니다.	7.6101251

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

침대

아티클 • 2023. 10. 20.

라디안으로 지정된 각도의 코탄젠트를 반환합니다.

구문

DAX

COT (<number>)

매개 변수

용어	정의
번호	코탄젠트를 사용할 라디안 단위의 각도입니다.

반환 값

지정된 각도의 코탄젠트입니다.

설명

- 숫자의 절대값은 2^{27} 미만이어야 하며 0일 수 없습니다.
- number가 제약 조건을 벗어나면 오류가 반환됩니다.
- 숫자가 숫자가 아닌 값이면 오류가 반환됩니다.

예시

다음 DAX 쿼리는

DAX

EVALUATE { COT(30) }

반환

[값]

-0.156119952161659

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COTH

아티클 • 2023. 10. 20.

쌍곡선 각도의 쌍곡 코탄젠트를 반환합니다.

구문

DAX

COTH (<number>)

매개 변수

용어 정의

번호 쌍곡 코탄젠트를 원하는 라디안 단위의 쌍곡선 각도입니다.

반환 값

지정된 각도의 쌍곡 코탄젠트입니다.

설명

- 쌍곡 코탄젠트는 일반 (원형) 코탄젠트의 아날로그입니다.
- 숫자의 절대값은 2^{27} 미만이어야 하며 0일 수 없습니다.
- number가 제약 조건을 벗어나면 오류가 반환됩니다.
- 숫자가 숫자가 아닌 값이면 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는

DAX

```
EVALUATE { COTH(2) }
```

반환

[값]

1.03731472072755

피드백

이 페이지가 도움이 되었나요?

Yes

No

통화

아티클 • 2023. 10. 20.

인수를 평가하고 결과를 통화 데이터 형식으로 반환합니다.

구문

DAX

CURRENCY(<value>)

매개 변수

용어 정의

value 다른 모든 작업 전에 식을 정확히 한 번 계산할 단일 스칼라 값을 반환하는 DAX 식입니다.

반환 값

평가되고 통화 형식 값으로 반환되는 식의 값입니다.

설명

- CURRENCY 함수는 5번째 유효 소수점(값)을 반올림하여 4번째 10진수를 반환합니다. 5번째 유효 소수점이 5보다 크거나 같으면 반올림이 발생합니다. 예를 들어 값이 3.666666666666이면 통화 변환 시 \$3.6667가 반환됩니다. 하지만 값이 3.0123456789이면 통화 변환 시 \$3.0123가 반환됩니다.
- 식의 데이터 형식이 `True`/`False`인 경우 `CURRENCY(<True/False>)`는 `True` 값의 경우 `1.0000`을 반환하고 `False` 값의 경우 `0.0000`을 반환합니다.
- 식의 데이터 형식이 `Text`인 경우 `CURRENCY(<Text>)`는 텍스트를 숫자로 변환하려고 시도합니다. 변환에 성공하면 숫자가 통화로 변환되고, 그렇지 않으면 오류가 반환됩니다.
- 식의 데이터 형식이 `DateTime`이면 `CURRENCY(<DateTime>)`는 `datetime` 값을 숫자로 변환하고 해당 숫자를 통화로 변환합니다. `DateTime` 값에는 지정된 날짜와 1900-03-01 사이의 일 수를 나타내는 정수 부분과 일의 비율을 나타내는 분수(여기

서 12시간 또는 정오는 0.5일임)가 있습니다. 식 값이 적절한 DateTime 값이 아니면 오류가 반환됩니다.

예시

숫자 1234.56을 통화 데이터 형식으로 변환합니다.

DAX

```
= CURRENCY(1234.56)
```

값 1234.56000을 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DEGREES

아티클 • 2023. 10. 20.

라디안을 도 단위로 변환합니다.

구문

DAX

DEGREES(angle)

매개 변수

용어 정의

각도 필수입니다. 변환하려는 라디안 단위의 각도입니다.

예시

수식	설명	결과
= DEGREES(PI())	pi 라디안의 정도	180

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

분할

아티클 • 2023. 12. 14.

나누기 작업을 수행하고 0으로 나누기에서 대체 결과 또는 BLANK()를 반환합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
분자	나눌 배당금 또는 숫자입니다.
분모	나눌 수 있는 수입니다.
alternateresult	(선택 사항) 0으로 나누면 오류가 발생할 때 반환되는 값입니다. 제공되지 않으면 기본값은 BLANK()입니다.

반환 값

- 10진수입니다.

설명

- 0으로 나누는 대체 결과는 상수여야 합니다.
- DIVIDE를 사용하는 경우 모범 사례는 DAX에서 DIVIDE 함수와 나누기 연산자(/)를 참조하세요.

예시

다음 예제에서는 2.5를 반환합니다.

DAX

```
= DIVIDE(5,2)
```

예 1

다음 예제에서는 BLANK를 반환합니다.

```
DAX
```

```
= DIVIDE(5,0)
```

예제 2

다음 예제에서는 1을 반환합니다.

```
DAX
```

```
= DIVIDE(5,0,1)
```

관련 콘텐츠

[QUOTIENT 함수](#)

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

EVEN

아티클 • 2023. 10. 20.

가장 가까운 짹수로 반올림된 숫자를 반환합니다. 이 함수는 두 가지로 제공되는 항목을 처리하는 데 사용할 수 있습니다. 예를 들어 압축 상자는 하나 또는 두 개의 항목의 행을 허용합니다. 가장 가까운 두 개까지 반올림된 항목 수가 상자의 용량과 일치하면 상자가 가득 찬습니다.

구문

DAX

EVEN(number)

매개 변수

용어	정의
번호	반올림할 값입니다.

반환 값

가장 가까운 짹수로 반올림된 숫자를 반환합니다.

설명

- number가 숫자가 아니면 EVEN는 #VALUE 반환합니다. 오류 값입니다.
- 숫자의 부호에 관계없이 0에서 벗어나면 값이 반올림됩니다. number가 짹수이면 반올림이 발생하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

수식	설명	결과
= EVEN(1.5)	1.5를 가장 가까운 짹수로 반올림합니다.	2

수식	설명	결과
EVEN(3)	가장 가까운 짝수로 반올림합니다.	
= EVEN(2)	2를 가장 가까운 짝수로 반올림합니다.	2
= EVEN(-1)	-1을 가장 가까운 짝수로 반올림합니다.	-2

피드백

이 페이지가 도움이 되었나요?

Yes

No

EXP

아티클 • 2023. 12. 14.

지정된 숫자의 힘으로 e를 반환합니다. 상수 e는 자연 로그의 기준인 2.71828182845904와 같습니다.

구문

DAX

`EXP(<number>)`

매개 변수

[+] 테이블 확장

용어 정의
정의

번호 기본 e에 적용되는 지수입니다. 상수 e는 자연 로그의 기준인 2.71828182845904와 같습니다.

반환 값

10진수입니다.

예외

설명

- EXP는 지정된 수의 자연 로그인 LN의 역수입니다.
- e 이외의 밑의 힘을 계산하려면 지수 연산자(^)를 사용합니다. 자세한 내용은 [DAX 연산자 참조를 참조하세요](#).

예시

다음 수식은 열 [Power]에 포함된 숫자의 능력으로 e를 계산합니다.

DAX

= EXP([Power])

관련 콘텐츠

[수학 및 삼각 함수](#)

[LN 함수](#)

[EXP 함수](#)

[LOG 함수](#)

[LOG 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FACT

아티클 • 2023. 12. 14.

지정된 숫자로 끝나는 계열 1*2*3*...*에 해당하는 숫자의 계수를 반환합니다.

구문

DAX

FACT(<number>)

매개 변수

 테이블 확장

용어	정의
번호	인수를 계산할 음수가 아닌 숫자입니다.

반환 값

10진수입니다.

설명

- 숫자가 정수가 아니면 잘리고 오류가 반환됩니다. 결과가 너무 크면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 수식은 열 **[Values]**에 있는 일련의 정수에 대한 팩터를 반환합니다.

DAX

= **FACT**([Values])

다음 표에서는 예상 결과를 보여줍니다.

 테이블 확장

값	결과
0	1
1	6
2	2
3	6
4	24
5	120
170	7.257415615308E+306

관련 콘텐츠

[수학 및 삼각 함수](#)

[TRUNC 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

바닥

아티클 • 2023. 12. 14.

숫자를 0으로 내림하여 가장 가까운 중요도 배수로 내림합니다.

구문

DAX

`FLOOR(<number>, <significance>)`

매개 변수

 테이블 확장

용어	정의
번호	반올림할 숫자 값입니다.
중요성	반올림할 배수입니다. 인수번호 와 significance 는 모두 양수이거나 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.
- 숫자의 부호에 관계없이 0에서 벗어나면 값이 반올림됩니다. 숫자가 의미의 정확한 배수인 경우 반올림이 발생하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 수식에서는 InternetSales 테이블의 [Total Product Cost] 열에 있는 값을 사용하고 1의 가장 가까운 배수로 내림합니다.

DAX

```
= FLOOR(InternetSales[Total Product Cost], .1)
```

다음 표에서는 일부 샘플 값의 예상 결과를 보여 줍니다.

 테이블 확장

값	예상 결과
10.8423	10.8
8.0373	8
2.9733	2.9

관련 콘텐츠

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Gcd

아티클 • 2023. 10. 20.

두 개 이상의 정수 중 가장 큰 공통 수수를 반환합니다. 가장 큰 공통 제수는 number1과 number2를 re기본der 없이 나누는 가장 큰 정수입니다.

구문

DAX

GCD(number1, [number2], ...)

매개 변수

용어	정의
number1,	Number1은 필수이며, 후속 숫자는 선택 사항입니다. 1~255개 값입니다. 정수
number2, ...	가 아닌 값이 있으면 잘립니다.

반환 값

두 개 이상의 정수 중 가장 큰 공통 수입니다.

설명

- 숫자가 아닌 인수가 있으면 GCD에서는 #VALUE! 오류 값입니다.
- 인수가 0보다 작으면 GCD에서는 #NUM! 오류 값입니다.
- 값이 균등하게 나눕니다.
- 소수는 그 자체만 있고 다른 하나는 제수로만 있습니다.
- GCD에 대한 매개 변수가 $>= 2^{53}$ 이면 GCD는 #NUM 반환합니다. 오류 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

수식	설명	결과
= GCD(5, 2)	5와 2의 가장 큰 공통 수입니다.	1
= GCD(24, 36)	24 및 36의 가장 큰 공통 수수입니다.	12
= GCD(7, 1)	7과 1의 가장 큰 공통 수수입니다.	1

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

INT

아티클 • 2023. 12. 14.

숫자를 가장 가까운 정수로 내림합니다.

구문

DAX

`INT(<number>)`

매개 변수

[+] 테이블 확장

용어	정의
번호	정수로 반올림할 숫자입니다.

반환 값

정수입니다.

설명

TRUNC와 INT는 둘 다 정수 반환과 비슷합니다. TRUNC는 숫자의 소수 부분을 제거합니다. INT는 숫자의 소수 부분 값을 기준으로 숫자를 가장 가까운 정수로 내림합니다. INT 및 TRUNC는 음수를 사용하는 경우에만 다릅니다. -4를 반환하지만 `INT(-4.3)` -5가 더 낮은 숫자이므로 -5를 반환합니다 `TRUNC(-4.3)`.

예시

다음 식은 값을 1로 반올림합니다. ROUND 함수를 사용하는 경우 결과는 2입니다.

DAX

= `INT(1.5)`

관련 콘텐츠

[수학 및 삼각 함수](#)

[ROUND 함수](#)

[ROUNDUP 함수](#)

[ROUNDDOWN 함수](#)

[MROUND 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ISO.CEILING

아티클 • 2023. 10. 20.

숫자를 가장 가까운 정수 또는 가장 가까운 중요 배수로 올립합니다.

구문

DAX

ISO.CEILING(<number>[, <significance>])

매개 변수

용어 정의

번호

반올림할 숫자 또는 숫자가 포함된 열에 대한 참조입니다.

증명
요성
(선택 사항) 반올림하려는 중요도의 배수입니다. 예를 들어 가장 가까운 정수로 반올림하려면 1을 입력합니다. significance 단위를 지정하지 않으면 숫자가 가장 가까운 정수로 반올림됩니다.

반환 값

숫자 인수와 같은 형식의 숫자로, 지정된 대로 반올림됩니다.

설명

DAX에는 다음과 같은 차이점이 있는 두 개의 CEILING 함수가 있습니다.

- CEILING 함수는 Excel에서 CEILING 함수의 동작을 에뮬레이트합니다.
- ISO입니다. CEILING 함수는 최대값을 결정하기 위해 ISO 정의 동작을 따릅니다.

두 함수는 양수에 대해 동일한 값을 반환하지만 음수에 대해서는 다른 값을 반환합니다. SIGNIFICANCE의 양수 배수를 사용하는 경우 CEILING 및 ISO입니다. CEILING는 음수를 위쪽으로 반올림합니다(양수 무한대쪽으로). 음수의 중요도를 사용하는 경우 CEILING는 음수를 음수 무한대로 반올림하고 ISO는 음수로 반올림합니다. CEILING는 음수를 위쪽으로 반올림합니다(양수 무한대로).

결과 형식은 일반적으로 다음과 같은 예외를 제외하고 인수로 사용되는 significance의 형식과 동일합니다.

- 첫 번째 인수가 통화 형식이면 결과는 통화 형식이 됩니다.
- 선택적 인수가 포함되지 않은 경우 결과는 정수 형식입니다.
- significance 인수가 부울 형식이면 결과는 정수 형식입니다.
- significance 인수가 숫자가 아닌 형식이면 결과는 실제 형식입니다.

예: 양수

다음 수식은 4.45를 반환합니다. 이는 가격 책정에서 더 작은 단위를 사용하지 않으려면 유용할 수 있습니다. 기존 제품의 가격이 \$4.42인 경우 ISO를 사용할 수 있습니다. 5 센트의 가장 가까운 단위로 가격을 반올림하는 CEILING.

DAX

```
= ISO.CEILING(4.42,0.05)
```

예: 음수

다음 수식은 ISO 최대값 -4.40을 반환합니다.

DAX

```
= ISO.CEILING(-4.42,0.05)
```

참고 항목

[수학 및 삼각 함수](#)

[FLOOR 함수](#)

[CEILING 함수](#)

[ROUNDUP 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LCM

아티클 • 2023. 10. 20.

가장 일반적인 정수 배수를 반환합니다. 가장 일반적인 배수는 모든 정수 인수 number1, number2 등의 배수인 가장 작은 양의 정수입니다. LCM을 사용하여 분모가 다른 분수를 추가합니다.

구문

DAX

`LCM(number1, [number2], ...)`

매개 변수

용어	정의
number1, number2,...	Number1은 필수이며, 후속 숫자는 선택 사항입니다. 1~255개 값으로, 가장 일반적인 배수가 가장 적습니다. 값이 정수가 아니면 잘립니다.

반환 값

가장 일반적인 정수 배수를 반환합니다.

설명

- 숫자가 아닌 인수가 있으면 LCM은 #VALUE 반환합니다. 오류 값입니다.
- 인수가 0보다 작으면 LCM은 #NUM 반환합니다. 오류 값입니다.
- $LCM(a,b) >= 2^{53}$ 이면 LCM은 #NUM! 오류 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

수식	설명	결과
= LCM(5, 2)	5와 2의 최소 공통 배수입니다.	10
= LCM(24, 36)	24 및 36의 최소 공통 배수입니다.	72

피드백

이 페이지가 도움이 되었나요?

Yes

No

Ln

아티클 • 2023. 12. 14.

숫자의 자연 로그를 반환합니다. 자연 로그는 상수 e(2.71828182845904)를 기반으로 합니다.

구문

DAX

`LN(<number>)`

매개 변수

[] 테이블 확장

용어	정의
번호	자연 로그를 사용할 양수입니다.

반환 값

10진수입니다.

설명

LN은 EXP 함수의 역입니다.

예시

다음 예제에서는 열 `[Values]`에 있는 숫자의 자연 로그를 반환합니다.

DAX

= `LN([Values])`

관련 콘텐츠

수학 및 삼각 함수

EXP 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LOG

아티클 • 2023. 12. 14.

지정한 기준으로 숫자의 로그를 반환합니다.

구문

DAX

`LOG(<number>,<base>)`

매개 변수

 테이블 확장

용어	정의
번호	로그를 사용할 양수입니다.
base	로그의 밑입니다. 생략하면 기본은 10입니다.

반환 값

10진수입니다.

설명

값이 너무 커서 표시할 수 없는 경우 오류가 발생할 수 있습니다.

LOG10 함수는 유사하지만 항상 공통 로그를 반환합니다. 즉, base 10의 로그입니다.

예시

다음 수식은 동일한 결과 2를 반환합니다.

DAX

```
= LOG(100,10)  
= LOG(100)
```

```
= LOG10(100)
```

관련 콘텐츠

[수학 및 삼각 함수](#)

[EXP 함수](#)

[LOG 함수](#)

[LOG10 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LOG10

아티클 • 2023. 12. 14.

숫자의 base-10 로그를 반환합니다.

구문

DAX

`LOG10(<number>)`

매개 변수

 테이블 확장

용어	정의
번호	base-10 로그를 사용할 양수입니다.

반환 값

10진수입니다.

설명

LOG 함수를 사용하면 base 10을 사용하는 대신 로그의 밑을 변경할 수 있습니다.

예시

다음 수식은 동일한 결과인 2를 반환합니다.

DAX

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

관련 콘텐츠

수학 및 삼각 함수

EXP 함수

LOG 함수

LOG10 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MOD

아티클 • 2023. 12. 14.

숫자를 수로 나눈 후 re기본der를 반환합니다. 결과에는 항상 수수와 동일한 기호가 있습니다.

구문

DAX

```
MOD(<number>, <divisor>)
```

매개 변수

[+] 테이블 확장

용어	정의
번호	나누기를 수행한 후 re기본der를 찾으려는 숫자입니다.
divisor	나눌 숫자입니다.

반환 값

정수입니다.

설명

- 수위가 0이면 MOD에서 오류를 반환합니다. 0으로 나눌 수 없습니다.
- MOD 함수는 INT 함수의 관점에서 표현할 수 있습니다. $MOD(n, d) = n - d * INT(n/d)$

예 1

다음 수식은 2로 나눈 3의 re기본der인 1을 반환합니다.

DAX

```
= MOD(3,2)
```

예제 2

다음 수식은 -1을 반환하고, 3의 remainder를 2로 나눕니다. 부호는 항상 수수의 부호와 동일합니다.

DAX

```
= MOD(-3, -2)
```

관련 콘텐츠

[수학 및 삼각 함수](#)

[ROUND 함수](#)

[ROUNDUP 함수](#)

[ROUNDDOWN 함수](#)

[MROUND 함수](#)

[INT 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MROUND

아티클 • 2023. 10. 20.

원하는 배수로 반올림된 숫자를 반환합니다.

구문

DAX

MROUND(<number>, <multiple>)

매개 변수

용어	정의
번호	반올림할 숫자입니다.
여러 가지	숫자를 반올림하려는 중요도의 배수입니다.

반환 값

10진수입니다.

설명

MROUND는 지정된 배수로 나눈 숫자의 remainder가 다중 값의 절반보다 크거나 같으면 0에서 멀리 반올림합니다.

예: 소수 자릿수

다음 식은 1.3을 .2의 가장 가까운 배수로 반올림합니다. 예상 결과는 1.4입니다.

DAX

= MROUND(1.3,0.2)

예: 음수

다음 식은 -10을 -3의 가장 가까운 배수로 반올림합니다. 예상 결과는 -9입니다.

DAX

```
= MROUND(-10,-3)
```

예: 오류

다음 식은 숫자가 다른 기호를 가지므로 오류를 반환합니다.

DAX

```
= MROUND(5,-2)
```

참고 항목

[수학 및 삼각 함수](#)

[ROUND 함수](#)

[ROUNDUP 함수](#)

[ROUNDDOWN 함수](#)

[MROUND 함수](#)

[INT 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

이상한

아티클 • 2023. 10. 20.

가장 가까운 홀수로 반올림된 숫자를 반환합니다.

구문

DAX

`ODD(number)`

매개 변수

용어	정의
번호	필수입니다. 반올림할 값입니다.

반환 값

가장 가까운 홀수로 반올림된 숫자를 반환합니다.

설명

- number가 숫자가 아니면 ODD는 #VALUE! 오류 값입니다.
- 숫자의 부호에 관계없이 0에서 벗어나면 값이 반올림됩니다. number가 홀수이면 반올림이 발생하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

수식	설명	결과
= ODD(1.5)	1.5를 가장 가까운 홀수로 반올림합니다.	3
= ODD(3)	3을 가장 가까운 홀수로 반올림합니다.	3

수식	설명	결과
= ODD(2)	2를 가장 가까운 홀수로 반올림합니다.	3
= ODD(-1)	-1을 가장 가까운 홀수로 반올림합니다.	-1
= ODD(-2)	-2를 0에서 가장 가까운 홀수로 반올림합니다.	-3

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PI

아티클 • 2023. 12. 14.

15자리까지 정확한 Pi 값인 3.14159265358979를 반환합니다.

구문

DAX

`PI()`

반환 값

Pi 값이 3.14159265358979인 10진수로, 15자리로 정확합니다.

설명

Pi는 수학 상수입니다. DAX에서 Pi는 Excel과 동일한 15자리까지 정확한 실수로 표시됩니다.

예시

다음 수식은 열 `[Radius]`의 반지름이 지정된 원의 영역을 계산합니다.

DAX

`= PI()*([Radius]*2)`

관련 콘텐츠

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

전원

아티클 • 2023. 12. 14.

숫자의 결과를 출력으로 반환합니다.

구문

DAX

`POWER(<number>, <power>)`

매개 변수

[+] 테이블 확장

용어	정의
번호	실수일 수 있는 기본 번호입니다.
power	밑 수를 거듭 제곱하기 위한 지수입니다.

반환 값

10진수입니다.

예시

다음 예제에서는 25를 반환합니다.

DAX

`= POWER(5,2)`

관련 콘텐츠

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

QUOTIENT

아티클 • 2023. 12. 14.

나누기를 수행하고 나누기 결과의 정수 부분만 반환합니다. 나누기의 remainder를 카드 디스하려는 경우 이 함수를 사용합니다.

구문

DAX

`QUOTIENT(<numerator>, <denominator>)`

매개 변수

[+] 테이블 확장

용어	정의
분자	나눌 배당금 또는 숫자입니다.
분모	나눌 수 또는 숫자입니다.

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 포함된 경우 값의 전체 열에 대해 오류가 반환됩니다.

예시

다음 수식은 동일한 결과 2를 반환합니다.

DAX

```
= QUOTIENT(5,2)
```

DAX

```
= QUOTIENT(10/2,2)
```

관련 콘텐츠

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RADIANS

아티클 • 2023. 10. 20.

각도를 라디안으로 변환합니다.

구문

DAX

RADIANS(angle)

매개 변수

용어	정의
각도	필수입니다. 변환할 각도입니다.

예시

수식	설명	결과
= RADIANS(270)	라디안 270도(4.712389 또는 $3\pi/2$ 라디안)	4.712389

피드백

이 페이지가 도움이 되었나요?

Yes

No

RAND

아티클 • 2023. 10. 20.

0보다 크거나 같고 1보다 작은 난수를 균등하게 분산하여 반환합니다. 이 함수를 포함하는 셀이 다시 계산될 때마다 반환되는 숫자가 변경됩니다.

구문

DAX

RAND()

반환 값

10진수입니다.

설명

- 다시 계산은 모델이 수동 **또는 자동** 다시 계산 모드로 설정되었는지 여부, 데이터가 새로 고쳐졌는지 여부 등 다양한 요인에 따라 달라집니다.
- 고정 값이 없는 RAND 및 기타 휘발성 함수가 항상 다시 계산되는 것은 아닙니다. 예를 들어 쿼리 또는 필터링을 실행해도 이러한 함수가 다시 평가되지는 않습니다. 그러나 전체 열을 다시 계산하면 이러한 함수에 대한 결과가 다시 계산됩니다. 이러한 상황에는 외부 데이터 원본에서 새로 고침 또는 이러한 함수가 포함된 수식을 다시 평가하는 데이터의 수동 편집이 포함됩니다.
- 함수가 측정값 정의에 사용되는 경우 RAND는 항상 다시 계산됩니다.
- RAND 함수는 0으로 나누기 등의 오류를 방지하기 위해 0의 결과를 반환할 수 없습니다.

예제

다른 두 숫자 사이에 난수를 생성하려면 다음을 사용합니다.

DAX

= **RAND()*(b-a)+a**

0보다 크고 1보다 작은 난수를 생성하려면 다음을 수행합니다.

DAX

```
= RAND()
```

0보다 크고 100보다 작은 난수를 생성하려면

DAX

```
= RAND()*100
```

0보다 크고 100보다 작은 난수를 생성하려면

DAX

```
INT(RAND()*100)
```

참고 항목

[수학 및 삼각 함수](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RANDBETWEEN

아티클 • 2023. 12. 14.

지정한 두 숫자 사이의 범위에서 난수를 반환합니다.

구문

DAX

RANDBETWEEN(<bottom>, <top>)

매개 변수

 테이블 확장

용어	정의
맨 아래	함수가 반환할 가장 작은 정수입니다.
상단	함수가 반환할 가장 큰 정수입니다.

반환 값

정수입니다.

설명

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

예시

다음 수식은 1에서 10 사이의 난수를 반환합니다.

DAX

= **RANDBETWEEN(1,10)**

관련 콘텐츠

수학 및 삼각 함수

통계 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROUND

아티클 • 2023. 10. 20.

숫자를 지정된 숫자 수로 반올림합니다.

구문

DAX

```
ROUND(<number>, <num_digits>)
```

매개 변수

용어	정의
번호	반올림할 숫자입니다.
num_digits	반올림할 자릿수입니다. 음수 값은 소수점 왼쪽에 숫자를 반올림합니다. 가장 가까운 정수로 반올림되는 0의 값입니다.

반환 값

10진수입니다.

설명

- num_digits **0보다 크면** 숫자가 지정된 소수 자릿수로 반올림됩니다.
- num_digits **0이면** 가장 가까운 정수로 반올림됩니다.
- num_digits **0보다 작으면** 소수점 왼쪽으로 반올림됩니다.
- 관련된 함수
 - 항상 반올림하려면(0에서 멀리) ROUNDUP 함수를 사용합니다.
 - 항상 0으로 반올림하려면 ROUNDDOWN 함수를 사용합니다.
 - 숫자를 특정 배수(예: 0.5의 가장 가까운 배수로 반올림)로 반올림하려면 MROUND 함수를 사용합니다.
 - TRUNC 및 INT 함수를 사용하여 숫자의 정수 부분을 가져옵니다.

예 1

다음 수식은 2.15를 10진수 위치로 반올림합니다. 예상 결과는 2.2입니다.

DAX

```
= ROUND(2.15,1)
```

예제 2

다음 수식은 21.5를 소수점 왼쪽의 소수점 10진수 위치로 반올림합니다. 예상 결과는 20입니다.

DAX

```
= ROUND(21.5,-1)
```

참고 항목

[수학 및 삼각 함수](#)

[라운드](#)

[라운드다운](#)

[MROUND](#)

[INT](#)

[TRUNC](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

라운드다운

아티클 • 2023. 12. 14.

숫자를 0으로 내림합니다.

구문

DAX

`ROUNDDOWN(<number>, <num_digits>)`

매개 변수

 테이블 확장

용어	정의
번호	반올림하려는 실수입니다.
num_digits	반올림할 자릿수입니다. 음수는 소수점 왼쪽으로 반올림합니다. 0에서 가장 가까운 정수로 0입니다.

반환 값

10진수입니다.

설명

- num_digits 0보다 크면 숫자 값이 지정된 소수 자릿수로 반올림됩니다.
- num_digits 0이면 숫자 값이 가장 가까운 정수로 반올림됩니다.
- num_digits 0보다 작으면 숫자 값이 소수점 왼쪽으로 반올림됩니다.
- ROUNDDOWN은 항상 숫자를 반올림한다는 점을 제외하고 ROUND처럼 동작합니다. INT 함수도 반올림하지만 INT를 사용하면 결과가 항상 정수인 반면 ROUNDDOWN을 사용하면 결과의 전체 자릿수를 제어할 수 있습니다.

예 1

다음 예제에서는 3.14159를 소수점 이하의 3자리로 내림합니다. 예상 결과는 3.141입니다.

DAX

```
= ROUNDOWN(3.14159,3)
```

예제 2

다음 예제에서는 31415.92654 값을 소수점 왼쪽에 있는 20진수 위치로 내림합니다. 예상 결과는 31400입니다.

DAX

```
= ROUNDOWN(31415.92654, -2)
```

관련 콘텐츠

[수학 및 삼각 함수](#)

[라운드](#)

[검거](#)

[라운드다운](#)

[MROUND](#)

[INT](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROUNDUP

아티클 • 2023. 12. 14.

숫자를 0(0)에서 위로 반올림합니다.

구문

DAX

`ROUNDUP(<number>, <num_digits>)`

매개 변수

 테이블 확장

용어	정의
번호	반올림하려는 실수입니다.
num_digits	반올림할 자릿수입니다. 소수점의 왼쪽으로 반올림하는 num_digits 음수 값입니다. num_digits 0이거나 생략되면 숫자가 가장 가까운 정수로 반올림됩니다.

반환 값

10진수입니다.

설명

- num_digits 0보다 크면 숫자가 지정된 소수 자릿수로 반올림됩니다.
- num_digits 0이면 숫자가 가장 가까운 정수로 반올림됩니다.
- num_digits 0보다 작으면 숫자가 소수점 왼쪽으로 반올림됩니다.
- ROUNDUP은 항상 숫자를 반올림한다는 점을 제외하고 ROUND처럼 동작합니다.

예시

다음 수식은 Pi를 소수점 이하 4자리로 반올림합니다. 예상 결과는 3.1416입니다.

DAX

```
= ROUNDUP(PI(),4)
```

예: 두 번째 인수로 10진수

다음 수식은 1.3을 0.2의 가장 가까운 배수로 반올림합니다. 예상 결과는 2입니다.

DAX

```
= ROUNDUP(1.3,0.2)
```

예: 음수를 두 번째 인수로

다음 수식은 다음 표에 표시된 예상 결과와 함께 열 FreightCost의 값을 반올림합니다.

DAX

```
= ROUNDUP([Values],-1)
```

num_digits 0보다 작으면 10진수 기호의 왼쪽에 있는 위치 수가 지정한 값만큼 증가합니다.

 테이블 확장

FreightCost	예상 결과
13.25	20
2.45	10
25.56	30
1.34	10
345.01	350

관련 콘텐츠

[수학 및 삼각 함수](#)

[라운드](#)

[라운드다운](#)

MROUND

INT

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

서명

아티클 • 2023. 12. 14.

숫자의 부호, 계산 결과 또는 열의 값을 결정합니다. 이 함수는 숫자가 양수이면 1을 반환하고, 숫자가 0이면 0을 반환하고, 숫자가 음수이면 -1을 반환합니다.

구문

DAX

SIGN(<number>)

매개 변수

[+] 테이블 확장

용어	정의
번호	모든 실수, 숫자가 포함된 열 또는 숫자로 계산되는 식입니다.

반환 값

정수입니다. 가능한 반환 값은 1, 0 및 -1입니다.

[+] 테이블 확장

반환 값	설명
1	숫자가 양수입니다.
0	숫자가 0입니다.
-1	숫자가 음수입니다.

예시

다음 수식은 판매 가격을 뺀 비용을 계산하는 식 결과의 부호를 반환합니다.

DAX

```
= SIGN( ([Sale Price] - [Cost]) )
```

관련 콘텐츠

수학 및 삼각 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SIN

아티클 • 2023. 10. 20.

지정된 각도의 사인을 반환합니다.

구문

DAX

SIN(number)

매개 변수

용어 정의

번호 필수입니다. 사인을 원하는 라디안 단위의 각도입니다.

반환 값

지정된 각도의 사인을 반환합니다.

설명

인수가 도 단위인 경우 PI()/180을 곱하거나 RADIANS 함수를 사용하여 라디안으로 변환합니다.

예시

수식	설명	결과
= SIN(PI())	파이 라디안 사인 (0, 대략).	0.0
= SIN(PI()/2)	pi/2 라디안 사인.	1.0
= SIN(30*PI()/180)	30도의 사인.	0.5
= SIN(RADIANS(30))	30도의 사인.	0.5

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Sinh

아티클 • 2023. 10. 20.

숫자의 하이퍼볼릭 사인을 반환합니다.

구문

DAX

SINH(number)

매개 변수

용어	정의
번호	필수입니다. 모든 실수입니다.

반환 값

숫자의 하이퍼볼릭 사인을 반환합니다.

설명

- 하이퍼볼릭 사인의 수식은 다음과 같습니다.

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

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

예시

1.03초 미만의 결과를 얻을 확률입니다.

DAX

= 2.868*SINH(0.0342*1.03)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Sqrt

아티클 • 2023. 12. 14.

숫자의 제곱근을 반환합니다.

구문

DAX

SQRT(<number>)

매개 변수

[+] 테이블 확장

용어 정의

번호 제곱근을 사용할 숫자, 숫자가 들어 있는 열 또는 숫자로 계산되는 식입니다.

반환 값

10진수입니다.

설명

숫자가 음수이면 SQRT 함수는 오류를 반환합니다.

예시

다음 수식은

DAX

= **SQRT**(25)

관련 콘텐츠

[수학 및 삼각 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SQRTPI

아티클 • 2023. 10. 20.

(number * pi)의 제곱근을 반환합니다.

구문

DAX

SQRTPI(number)

매개 변수

용어	정의
번호	필수입니다. pi를 곱한 숫자입니다.

반환 값

(number * pi)의 제곱근을 반환합니다.

예시

수식	설명	결과
= SQRTPI(1)	pi의 제곱근입니다.	1.772454
= SQRTPI(2)	2 * pi의 제곱근.	2.506628

피드백

이 페이지가 도움이 되었나요?

Yes

No

TAN

아티클 • 2023. 10. 20.

지정된 각도의 탄젠트를 반환합니다.

구문

DAX

TAN(number)

매개 변수

용어 정의

번호 필수입니다. 탄젠트를 사용할 라디안 단위의 각도입니다.

반환 값

지정된 각도의 탄젠트를 반환합니다.

설명

인수가 도 단위인 경우 PI()/180을 곱하거나 RADIANS 함수를 사용하여 라디안으로 변환합니다.

예시

수식	설명	결과
= TAN(0.785)	탄젠트 0.785 라디안(0.99920)	0.99920
= TAN(45*PI()/180)	45도 탄젠트(1)	1
= TAN(RADIANS(45))	45도 탄젠트(1)	1

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Tanh

아티클 • 2023. 10. 20.

숫자의 쌍곡 탄젠트를 반환합니다.

구문

DAX

TANH(number)

매개 변수

용어	정의
번호	필수입니다. 모든 실수입니다.

반환 값

숫자의 쌍곡 탄젠트를 반환합니다.

설명

- 하이퍼볼릭 탄젠트 수식은 다음과 같습니다.

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

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

예시

수식	설명	결과
= TANH(-2)	-2의 쌍곡 탄젠트(-0.96403)	-0.964028
= TANH(0)	하이퍼볼릭 탄젠트 0(0)	0
= TANH(0.5)	0.5의 쌍곡 탄젠트(0.462117)	0.462117

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TRUNC

아티클 • 2023. 12. 14.

숫자의 소수 또는 소수 부분을 제거하여 숫자를 정수로 자른다.

구문

DAX

`TRUNC(<number>,<num_digits>)`

매개 변수

 테이블 확장

용어	정의
번호	자르려는 숫자입니다.
num_digits	잘림의 전체 자릿수를 지정하는 숫자입니다. 생략하면 0(0)입니다.

반환 값

정수입니다.

설명

TRUNC와 INT는 둘 다 정수 반환과 비슷합니다. TRUNC는 숫자의 소수 부분을 제거합니다. INT는 숫자의 소수 부분 값을 기준으로 숫자를 가장 가까운 정수로 내립니다. INT 및 TRUNC는 음수를 사용하는 경우에만 다릅니다. `TRUNC(-4.3)` -4를 반환하지만 `INT(-4.3)` -5가 더 작은 숫자이므로 -5를 반환합니다.

예 1

다음 수식은 pi의 정수 부분인 3을 반환합니다.

DAX

```
= TRUNC(PI())
```

예제 2

다음 수식은 -8.9의 정수 부분인 -8을 반환합니다.

DAX

```
= TRUNC(-8.9)
```

관련 콘텐츠

[수학 및 삼각 함수](#)

[라운드](#)

[검거](#)

[라운드다운](#)

[MROUND](#)

[INT](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

기타 함수

아티클 • 2023. 10. 20.

이러한 함수는 범주에서 정의할 수 없는 고유한 작업을 수행합니다.

이 범주에서

함수	설명
빈	공백을 반환합니다.
오류	오류 메시지와 함께 오류가 발생합니다.
EVALUATEANDLOG	첫 번째 인수의 값을 반환하고 DAX 평가 로그 프로파일러 이벤트에 기록합니다.
TOCSV	테이블을 CSV 형식의 문자열로 반환합니다.
TOJSON	테이블을 JSON 형식의 문자열로 반환합니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

BLANK

아티클 • 2023. 12. 14.

공백을 반환합니다.

구문

DAX

`BLANK()`

반환 값

공백입니다.

설명

- 공백은 null과 동일하지 않습니다. DAX는 데이터베이스 null과 Excel의 빈 셀 모두에 공백을 사용합니다.
- 일부 DAX 함수는 빈 셀을 Microsoft Excel과 약간 다르게 처리합니다. 공백 및 빈 문자열("")은 항상 동일하지는 않지만 일부 작업에서는 이러한 문자열을 처리할 수 있습니다.

예시

다음 예제에서는 수식에서 공백으로 작업하는 방법을 보여 줍니다. 수식은 재판매인과 인터넷 채널 간의 판매 비율을 계산합니다. 그러나 비율을 계산하기 전에 분모는 0 값에 대해 검사 합니다. 분모가 0이면 빈 값이 반환됩니다. 그렇지 않으면 비율이 계산됩니다.

DAX

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,  
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_USD]) )
```

이 수식을 사용하여 테이블 시각화를 만들 때 예상되는 결과가 표에 표시됩니다.

Row Labels	Accessories	Bikes	Clothing	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

원본 데이터 원본에서 BLANK 함수에서 평가한 열에 텍스트, 빈 문자열 또는 null이 포함되어 있을 수 있습니다. 원래 데이터 원본이 SQL Server 데이터베이스인 경우 null 및 빈 문자열은 서로 다른 종류의 데이터입니다. 그러나 이 작업의 경우 암시적 형식 캐스트가 수행되고 DAX는 이를 동일하게 처리합니다.

관련 콘텐츠

[텍스트 함수](#)

[ISBLANK 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

오류

아티클 • 2023. 10. 20.

오류 메시지와 함께 오류가 발생합니다.

구문

DAX

```
ERROR(<text>)
```

매개 변수

용어	정의
text	오류 메시지가 포함된 텍스트 문자열입니다.

반환 값

None

설명

- ERROR 함수는 스칼라 값이 필요한 모든 위치에 DAX 식에 배치할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 DAX 쿼리:

DAX

```
DEFINE  
MEASURE DimProduct[Measure] =  
    IF(  
        SELECTEDVALUE(DimProduct[Color]) = "Red",  
        ERROR("red color encountered"),  
        SELECTEDVALUE(DimProduct[Color])
```

```
)  
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])  
ORDER BY [Color]
```

실패하고 "적색이 발생했습니다"가 포함된 오류 메시지가 발생합니다.

예제 2

다음 DAX 쿼리:

DAX

```
DEFINE  
MEASURE DimProduct[Measure] =  
    IF(  
        SELECTEDVALUE(DimProduct[Color]) = "Magenta",  
        ERROR("magenta color encountered"),  
        SELECTEDVALUE(DimProduct[Color])  
    )  
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])  
ORDER BY [Color]
```

다음 표를 반환합니다.

DimProduct[색]	[측정값]
검정	검정
파랑	파랑
회색	회색
다색	다색
해당 없음	해당 없음
빨간색	빨간색
은	은
Silver\Black	Silver\Black
백인	흰색
노란색	노란색

Magenta는 제품 색 중 하나가 아니므로 ERROR 함수가 실행되지 않습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

EVALUATEANDLOG

아티클 • 2023. 10. 20.

첫 번째 인수의 값을 반환하고 DAX 평가 로그 프로파일러 이벤트에 기록합니다. 이 함수는 Power BI Desktop에서만 완벽하게 작동합니다. 다른 환경에서는 간단한 통과 함수 역할을 합니다.

구문

DAX

```
EVALUATEANDLOG(<Value>, [Label], [MaxRows])
```

매개 변수

용어	정의
값	계산 및 기록할 스칼라 식 또는 테이블 식입니다.
레이블	(선택 사항) 함수 호출의 인스턴스를 쉽게 식별하는 데 사용할 수 있는 DAX 평가 로그 이벤트의 json 텍스트와 레이블 열 모두에 포함된 상수 문자열입니다.
MaxRows	(선택 사항) 첫 번째 인수가 테이블 식인 경우 DAX 평가 로그 이벤트의 json 텍스트에 있는 최대 행 수입니다. 기본값은 10입니다.

반환 값

첫 번째 인수의 값입니다.

DAX 평가 로그 프로파일러 이벤트에 기록된 JSON 구조에는 다음이 포함됩니다.

- "expression"은 첫 번째 인수의 텍스트 버전입니다.
- 식에 지정된 경우 "label"은 Label 매개 변수입니다.
- "inputs"는 첫 번째 인수의 값에 영향을 주는 평가 컨텍스트의 열 목록입니다.
- "outputs"는 첫 번째 인수가 스칼라 식이고 첫 번째 인수가 테이블 식인 경우 출력 열 목록인 경우 단일 열 [Value]의 목록입니다.
- "data"는 첫 번째 인수가 스칼라 식인 경우 입력 값 및 출력 값의 목록이며, 첫 번째 인수가 테이블 식인 경우 입력 값 및 해당 출력 행의 목록입니다.
- "rowCount"는 첫 번째 인수가 테이블 식인 경우의 행 수입니다. json 출력의 행 수가 MaxRows 매개 변수에 의해 잘리더라도 rowCount는 잘리지 않은 행의 실제 수입니다.

설명

- SQL Server Profiler 및 오픈 소스 DAX 디버그 출력 도구를 사용하여 추적 이벤트를 캡처할 수 있습니다.
- 이 함수는 DAX 식의 거의 모든 하위 식과 함께 사용할 수 있으며 전체 식은 여전히 유효합니다.
- 첫 번째 인수가 단일 쿼리에서 여러 번 평가되면 함수는 입력 값과 해당 출력 값을 모두 포함하는 단일 DAX 평가 로그 이벤트를 생성합니다.
- 레이블 매개 변수를 지정하면 해당 값은 DAX 평가 로그 이벤트의 json 출력과 Label 열 모두에서 반환됩니다.
- 첫 번째 인수가 테이블 식인 경우 DAX 평가 로그 이벤트에 상위 MaxRows 행만 표시됩니다.
- 경우에 따라 이 함수는 최적화로 인해 실행되지 않습니다.
- DAX 평가 로그 이벤트가 100만 자보다 큰 경우 올바른 json 구조를 유지하기 위해 잘립니다.

예 1

다음 DAX 쿼리:

```
DAX

evaluate
SUMMARIZE(
    EVALUATEANDLOG(FILTER(Sales, [ProductKey] = 528)),
    Sales[SalesTerritoryKey],
    "sum",
    sum(Sales[Sales Amount])
)
```

다음 DAX 평가 로그 이벤트를 반환합니다.

```
JSON

{
    "expression": "FILTER(Sales, [ProductKey] = 528)",
    "inputs": [],
    "outputs": ["'Sales'[SalesOrderLineKey]", "'Sales'[ResellerKey]",
    "'Sales'[CustomerKey]", "'Sales'[ProductKey]", "'Sales'[OrderDateKey]",
    "'Sales'[DueDateKey]", "'Sales'[ShipDateKey]", "'Sales'[SalesTerritoryKey]",
    "'Sales'[Order Quantity]", "'Sales'[Unit Price]", "'Sales'[Extended
```

```

        "Amount]", "'Sales'[Product Standard Cost]", "'Sales'[Total Product Cost]",
        "'Sales'[Sales Amount]", "'Sales'[Unit Price Discount Pct]"],
        "data": [
            {
                "input": [],
                "rowCount": 3095,
                "output": [
                    [52174001, -1, 23785, 528, 20190707, 20190717, 20190714, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [52173001, -1, 26278, 528, 20190707, 20190717, 20190714, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [52082001, -1, 23831, 528, 20190705, 20190715, 20190712, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [52054002, -1, 11207, 528, 20190704, 20190714, 20190711, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [52036001, -1, 25337, 528, 20190704, 20190714, 20190711, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [51939002, -1, 23670, 528, 20190702, 20190712, 20190709, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [51911002, -1, 11746, 528, 20190701, 20190711, 20190708, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [51379003, -1, 13745, 528, 20190612, 20190622, 20190619, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [51264002, -1, 11282, 528, 20190605, 20190615, 20190612, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                    [51184003, -1, 11263, 528, 20190531, 20190610, 20190607, 1,
                     1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0]
                ]
            }
        ]
    }
}

```

예제 2

스칼라 인수 및 다양한 특성이 있는 다음 DAX 쿼리:

DAX

```

evaluate
SELECTCOLUMNS(
    TOPN(5, Customer),
    [Customer],
    "Customer",
    EVALUATEANDLOG([Customer] & ", " & [Country-Region], "customerLog")
)

```

다음 DAX 평가 로그 이벤트를 반환합니다.

JSON

```
{  
    "expression": "[Customer] & \\", \\ & [Country-Region]",  
    "label": "customerLog",  
    "inputs": ["'Customer'[Customer]", "'Customer'[Country-Region]"],  
    "data": [  
        {  
            "input": ["Russell Xie", "United States"],  
            "output": "Russell Xie, United States"  
        },  
        {  
            "input": ["Savannah Baker", "United States"],  
            "output": "Savannah Baker, United States"  
        },  
        {  
            "input": ["Maurice Tang", "United States"],  
            "output": "Maurice Tang, United States"  
        },  
        {  
            "input": ["Emily Wood", "United States"],  
            "output": "Emily Wood, United States"  
        },  
        {  
            "input": ["Meghan Hernandez", "United States"],  
            "output": "Meghan Hernandez, United States"  
        }  
    ]  
}
```

참고 항목

[TOCSV](#)

[TOJSON](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

TOCSV

아티클 • 2023. 12. 14.

테이블을 CSV 형식의 문자열로 반환합니다.

구문

DAX

```
TOCSV(<Table>, [MaxRows], [Delimiter], [IncludeHeaders])
```

매개 변수

 테이블 확장

용어	정의
테이블	CSV로 변환할 테이블입니다.
MaxRows	(선택 사항) 변환할 최대 행 수입니다. 기본값은 10개 행입니다.
구분 기호	(선택 사항) 열 구분 기호입니다. 기본값은 쉼표 ","입니다.
IncludeHeaders	(선택 사항) 열 이름의 머리글을 첫 번째 행으로 지정합니다. 기본값은 True입니다.

반환 값

테이블의 CSV 표현이 있는 문자열입니다.

예시

다음 DAX 쿼리:

DAX

```
EVALUATE  
{TOCSV(DimSalesTerritory)}
```

HRESULT = NO_ERROR를

```
'DimSalesTerritory'[SalesTerritoryKey], 'DimSalesTerritory'[SalesTerritoryAlternateKey], 'DimSalesTerritory'[SalesTerritoryRegion], 'DimSalesTerritory'[SalesTerritoryCountry], 'DimSalesTerritory'[SalesTerritoryGroup]  
1,1,Northwest,United States,North America  
2,2,Northeast,United States,North America  
3,3,Central,United States,North America  
4,4,Southwest,United States,North America  
5,5,Southeast,United States,North America  
6,6,Canada,Canada,North America  
7,7,France,France,Europe  
8,8,Germany,Germany,Europe  
9,9,Australia,Australia,Pacific  
10,10,United Kingdom,United Kingdom,Europe
```

관련 콘텐츠

[TOJSON](#)

[EVALUATEANDLOG](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

TOJSON

아티클 • 2023. 12. 14.

JSON 형식을 사용하여 테이블을 문자열로 반환합니다.

구문

DAX

```
TOJSON(<Table>, [MaxRows])
```

매개 변수

 테이블 확장

용어	정의
테이블	JSON으로 변환할 테이블입니다.
MaxRows	(선택 사항) 변환할 최대 행 수입니다. 기본값은 10개 행입니다.

반환 값

테이블의 JSON 표현이 있는 문자열입니다. 이 표현은 열 이름을 "header"로, 행 개수를 "rowCount"로, 값을 "data"로 포함합니다.

예시

다음 DAX 쿼리:

DAX

```
EVALUATE  
{TOJSON(DimSalesTerritory)}
```

HRESULT = NO_ERROR를

JSON

```
{  
  "header": ["'DimSalesTerritory'[SalesTerritoryKey]",  
             "'DimSalesTerritory'[SalesTerritoryAlternateKey]",  
             "'DimSalesTerritory'[SalesTerritoryRegion]",  
             "'DimSalesTerritory'[SalesTerritoryCountry]",  
             "'DimSalesTerritory'[SalesTerritoryGroup]"],<br>  
  "rowCount": 11,  
  "data": [  
    [1, 1, "Northwest", "United States", "North America"],  
    [2, 2, "Northeast", "United States", "North America"],  
    [3, 3, "Central", "United States", "North America"],  
    [4, 4, "Southwest", "United States", "North America"],  
    [5, 5, "Southeast", "United States", "North America"],  
    [6, 6, "Canada", "Canada", "North America"],  
    [7, 7, "France", "France", "Europe"],  
    [8, 8, "Germany", "Germany", "Europe"],  
    [9, 9, "Australia", "Australia", "Pacific"],  
    [10, 10, "United Kingdom", "United Kingdom", "Europe"]  
  ]  
}
```

관련 콘텐츠

[TOCSV](#)

[EVALUATEANDLOG](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

부모 및 자식 함수

아티클 • 2023. 10. 20.

이러한 함수는 부모/자식 계층으로 표시되는 데이터를 관리합니다. 자세한 내용은 [DAX의 부모-자식 계층 구조에 대한 함수 이해](#)를 참조하세요.

이 범주에서

함수	설명
PATH	현재 식별자의 모든 부모 식별자를 사용하여 구분된 텍스트 문자열을 반환합니다.
PATHCONTAINS	지정된 항목이 지정된 경로 내에 있으면 <i>TRUE</i> 를 반환합니다.
PATHITEM	PATH 함수를 계산한 결과 문자열에서 지정된 위치에 있는 항목을 반환합니다.
PATHITEMREVERSE	PATH 함수를 계산한 결과 문자열에서 지정된 위치에 있는 항목을 반환합니다.
PATHLENGTH	<i>self</i> 를 포함하여 지정된 PATH 결과의 지정된 항목에 대한 부모 수를 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX에서 부모-자식 계층 구조에 대한 함수 이해

아티클 • 2023. 10. 20.

DAX는 사용자가 모델에서 부모-자식 계층 구조로 표시되는 데이터를 관리하는 데 도움이 되는 5가지 함수를 제공합니다. 이 함수를 사용하면 사용자가 행에 있는 부모의 전체 계보를 가져올 수 있으며, 현재 행 위의 부모 n-수준인 상위 부모에 대한 계보가 있는 수준은 몇 개입니까? 현재 행 계층 구조의 맨 위에서 n-하위 항목이며 특정 부모가 현재 행 계층의 부모인가요?

DAX의 부모-자식 함수

다음 표에는 모든 함수 예제에서 사용되는 EmployeeKey 및 ParentEmployeeKey 열의 부모-자식 계층 구조가 포함되어 있습니다.

EmployeeKey	ParentEmployeeKey
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

위의 표에서 직원 112에는 부모가 정의되어 있지 않으며, 직원 14에는 관리자로 직원 112가 있고(ParentEmployeeKey), 직원 3에는 관리자로 직원 14, 직원 11, 13 및 162에는 직원 3이 관리자로 있는 것을 볼 수 있습니다. 위의 내용은 직원 112에게 그 이상의 관리자가 없으며 여기에 표시된 모든 직원의 최고 관리자임을 이해하는 데 도움이 됩니다. 또한 직원 3은 직원 14에게 보고하고 직원 11, 13, 162는 3으로 보고합니다.

다음 표에서는 사용 가능한 함수, 함수에 대한 간략한 설명 및 위에 표시된 동일한 데이터에 대한 함수의 예를 보여 줍니다.

PATH 함수 - 현재 행에 대한 모든 부모 식별자가 있는 구분된 텍스트를 반환합니다. 가장 오래된 텍스트나 맨 위부터 현재 행까지입니다.

EmployeeKey	ParentEmployeeKey	Path
112		112
14	112	112 14
3	14	112 14 3
11	3	112 14 3 11
13	3	112 14 3 13
162	3	112 14 3 162
117	162	112 14 3 162 117
221	162	112 14 3 162 221
81	162	112 14 3 162 81

PATHLENGTH 함수 - 지정된 PATH()의 수준 수를 반환하며, 현재 수준에서 가장 오래된 부모 수준 또는 최상위 상위 수준까지 시작합니다. 다음 예제 열에서 PathLength는 "`= PATHLENGTH([Path])`"으로 정의됩니다. 이 예제에는 이 함수의 작동 방식을 이해하는 데 도움이 되는 Path() 예제의 모든 데이터가 포함됩니다.

EmployeeKey	ParentEmployeeKey	Path	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 - PATH()에서 지정된 위치에 있는 항목을 결과와 같이 왼쪽에서 오른쪽으로 계산하여 반환합니다. 다음 예제 열 PathItem - 왼쪽에서 4번째는 "`= PATHITEM([Path],`

4)로 정의됩니다. 이 예제에서는 Path() 예제에서 동일한 샘플 데이터를 사용하여 왼쪽의 Path 문자열에서 네 번째 위치에 있는 EmployeKey를 반환합니다.

EmployeeKey	ParentEmployeeKey	Path	PathItem - 왼쪽에서 4번째
112		112	
14	112	112 14	
3	14	112 14 3	
11	3	112 14 3 11	11
13	3	112 14 3 13	13
162	3	112 14 3 162	162
117	162	112 14 3 162 117	162
221	162	112 14 3 162 221	162
81	162	112 14 3 162 81	162

PATHITEMREVERSE 함수 - 함수 결과와 같은 PATH()의 위치에 있는 항목을 반환하며 오른쪽에서 왼쪽으로 뒤로 계산합니다.

다음 예제 열 PathItemReverse - 오른쪽의 3번째 열은 "= PATHITEMREVERSE([Path], 3)" 으로 정의됩니다. 이 예제에서는 Path() 예제에서 동일한 샘플 데이터를 사용하여 오른쪽의 Path 문자열에서 세 번째 위치에 있는 EmployeKey를 반환합니다.

EmployeeKey	ParentEmployeeKey	Path	PathItemReverse - 오른쪽에서 3번째
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 항목이 지정된 경로 내에 있으면 **TRUE**를 반환합니다. 다음 예제 열 PathContains에서 직원 162는 "= PATHCONTAINS([Path], "162")"으로 정의됩니다. 이 예제에서는 지정된 경로에 직원 162가 포함된 경우 **TRUE**를 반환합니다. 이 예제에서 위의 Path() 예제의 결과를 사용합니다.

EmployeeKey	ParentEmployeeKey	Path	PathContains - 직원 162
112		112	FALSE
14	112	112 14	FALSE
3	14	112 14 3	FALSE
11	3	112 14 3 11	FALSE
13	3	112 14 3 13	FALSE
162	3	112 14 3 162	TRUE
117	162	112 14 3 162 117	TRUE

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PATH

아티클 • 2023. 10. 20.

현재 식별자의 모든 부모에 대한 식별자가 있는 구분된 텍스트 문자열을 반환합니다. 이 문자열은 가장 오래된 항목부터 시작하여 현재까지 계속됩니다.

구문

DAX

```
PATH(<ID_columnName>, <parent_columnName>)
```

매개 변수

용어	정의
ID_columnName	테이블의 행에 대한 고유 식별자를 포함하는 기존 열의 이름입니다. 식일 수 없습니다. ID_columnName 값의 데이터 형식은 텍스트 또는 정수여야 하며 parentColumnName 참조되는 열과 동일한 데이터 형식이어야 합니다.
parentColumnName	현재 행의 부모에 대한 고유 식별자를 포함하는 기존 열의 이름입니다. 식일 수 없습니다. parentColumnName 데이터 형식의 값 데이터 형식은 텍스트 또는 정수여야 하며 ID_columnName 값과 동일한 데이터 형식이어야 합니다.

반환 값

현재 식별자에 대한 모든 부모의 식별자를 포함하는 구분된 텍스트 문자열입니다.

설명

- 이 함수는 현재 행 값과 관련된 항목을 반환하기 위해 일종의 내부 계층 구조가 있는 테이블에서 사용됩니다. 예를 들어 직원, 직원의 관리자 및 관리자가 포함된 Employees 테이블에서 직원을 관리자에 연결하는 경로를 반환할 수 있습니다.
- 경로는 부모-자식 관계의 단일 수준으로 제한되지 않습니다. 지정된 시작 행에서 여러 수준 위로 올라간 관련 행을 반환할 수 있습니다.
 - 오름차순을 구분하는 데 사용되는 구분 기호는 세로 막대 '|'입니다.

- ID_columnName 및 parent_columnName 같은 동일한 데이터 형식, 텍스트 또는 정수여야 합니다.
 - parent_columnName 같은 ID_columnName 있어야 합니다. 즉, 자식 수준에 값이 없으면 부모를 조회할 수 없습니다.
 - parent_columnName BLANK이면 PATH()는 ID_columnName 값을 반환합니다. 즉, 직원의 관리자를 찾으지만 parent_columnName 열에 데이터가 없는 경우 PATH 함수는 직원 ID만 반환합니다.
 - ID_columnName 중복이 있고 해당 중복 항목에 대해 parent_columnName 동일한 경우 PATH()는 공통 parent_columnName 값을 반환합니다. 그러나 중복 항목에 대해 parent_columnName 값이 다른 경우 PATH()는 오류를 반환합니다. 즉, 동일한 직원 ID에 대한 목록이 두 개 있고 관리자 ID가 같으면 PATH 함수는 해당 관리자의 ID를 반환합니다. 그러나 관리자 ID가 다른 두 개의 동일한 직원 ID가 있는 경우 PATH 함수는 오류를 반환합니다.
 - ID_columnName BLANK이면 PATH()는 BLANK를 반환합니다.
 - ID_columnName 세로 막대 '|'가 포함된 경우 PATH()는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 각 직원의 모든 관리자를 나열하는 계산 열을 만듭니다.

DAX

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PATHCONTAINS

아티클 • 2023. 10. 20.

지정된 항목이 지정된 경로 내에 있으면 *TRUE*를 반환합니다.

구문

DAX

```
PATHCONTAINS(<path>, <item>)
```

매개 변수

용어	정의
경로	PATH 함수를 평가한 결과로 만들어진 문자열입니다.
item	경로 결과에서 찾을 텍스트 식입니다.

반환 값

경로에 항목이 있으면 *TRUE* 값이고, 그렇지 않으면 *FALSE*입니다.

설명

- 항목이 정수이면 텍스트로 변환되고 함수가 평가됩니다. 변환에 실패하면 함수는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 관리자 ID를 사용하고 직원 집합을 검사 계산 열을 만듭니다. 관리자 ID가 PATH 함수에서 반환된 관리자 목록 중 하나이면 PATHCONTAINS 함수는 *true*를 반환합니다. 그렇지 않으면 *false*를 반환합니다.

DAX

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
"23")
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PATHITEM

아티클 • 2023. 10. 20.

PATH 함수를 계산한 결과 문자열에서 지정된 위치에 있는 항목을 반환합니다. 위치는 왼쪽에서 오른쪽으로 계산됩니다.

구문

DAX

`PATHITEM(<path>, <position>[, <type>])`

매개 변수

용어	정의
경로	PATH 함수 결과 형식의 텍스트 문자열입니다.
배치	반환할 항목의 위치가 있는 정수 식입니다.
type	(선택 사항) 결과의 데이터 형식을 정의하는 열거형입니다.

형식 열거형

열거형	대체 열거형	설명
TEXT	0	결과는 데이터 형식 텍스트와 함께 반환됩니다. 않습니다(기본값).
INTEGER	1	결과는 정수로 반환됩니다.

반환 값

식별자 목록의 지정된 위치에 있는 PATH 함수에서 반환된 식별자입니다. PATH 함수에서 반환된 항목은 가장 먼 값에서 현재 항목으로 정렬됩니다.

설명

- 이 함수는 PATH 함수에서 반환된 계층 구조에서 특정 수준을 반환하는 데 사용할 수 있습니다. 예를 들어 모든 직원의 건너뛰기 수준 관리자만 반환할 수 있습니다.

- 1보다 작거나 경로의 요소 수보다 큰 위치에 대한 숫자를 지정하면 PATHITEM 함수는 BLANK를 반환합니다.
- 형식이 유효한 열거형 요소가 아니면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 현재 직원의 세 번째 계층 관리자를 반환합니다. 부모 계층 구조가 있는 문자열을 현재 직원에게 반환하는 PATH 함수에 대한 입력으로 직원 및 관리자 ID를 사용합니다. 해당 문자열에서 PATHITEM은 세 번째 항목을 정수로 반환합니다.

DAX

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PATHITEMREVERSE

아티클 • 2023. 10. 20.

PATH 함수를 계산한 결과 문자열에서 지정된 위치에 있는 항목을 반환합니다. 위치는 오른쪽에서 왼쪽으로 뒤로 계산됩니다.

구문

DAX

`PATHITEMREVERSE(<path>, <position>[, <type>])`

매개 변수

용어 정의

경로 PATH 함수의 계산으로 인한 텍스트 문자열입니다.

배치 반환할 항목의 위치가 있는 정수 식입니다. 위치는 오른쪽에서 왼쪽으로 뒤로 계산됩니다.

type (선택 사항) 결과의 데이터 형식을 정의하는 열거형입니다.

형식 열거형

열거형	대체 열거형	설명
TEXT	0	결과는 데이터 형식 텍스트와 함께 반환됩니다. 않습니다(기본값).
INTEGER	1	결과는 정수로 반환됩니다.

반환 값

지정된 경로의 n 위치 오름차순으로, 현재에서 가장 오래된 경로로 계산합니다.

설명

- 이 함수는 PATH 함수에서 생성된 계층 구조에서 개별 항목을 가져오는 데 사용할 수 있습니다.

- 이 함수는 계층의 표준 순서를 반대로 변경하여 가장 가까운 항목이 먼저 나열되도록 합니다. 예를 들어 PATH 함수가 계층 구조에서 직원 위의 관리자 목록을 반환하는 경우 위치 1에 직원의 ID가 포함되어 있으므로 PATHITEMREVERSE 함수는 직원의 직계 관리자를 2위 위치로 반환합니다.
- position에 지정된 숫자가 1(일)보다 작거나 path의 요소 수보다 큰 경우 PATHITEM 함수에서 BLANK를 반환합니다.
- 형식이 유효한 열거형 요소가 아니면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 직원 ID 열을 PATH 함수에 대한 입력으로 사용하고 반환되는 조부모 요소 목록을 반대로 바꿉니다. 지정된 위치는 3이고 반환 형식은 1입니다. 따라서 PATHITEMREVERSE 함수는 관리자를 나타내는 정수가 직원으로부터 두 수준 위로 올라옵니다.

DAX

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),  
3, 1)
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PATHLENGTH

아티클 • 2023. 10. 20.

self를 포함하여 지정된 PATH 결과의 지정된 항목에 대한 부모 수를 반환합니다.

구문

DAX

`PATHLENGTH(<path>)`

매개 변수

용어	정의
경로	PATH 함수의 계산으로 인한 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 직원 ID를 PATH 함수에 대한 입력으로 사용하고 계층 구조의 해당 직원 위에 있는 관리자 목록을 반환합니다. PATHLENGTH 함수는 해당 결과를 가져오고 시작한 직원을 포함하여 다양한 수준의 직원과 관리자를 계산합니다.

DAX

`= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))`

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

관계 함수

아티클 • 2023. 10. 20.

이 범주의 함수는 테이블 간의 관계를 관리하고 활용하기 위한 것입니다.

이 범주에서

함수	설명
크로스필터	두 열 사이에 있는 관계에 대한 계산에 사용할 교차 필터링 방향을 지정합니다.
관련	다른 테이블에서 관련 값을 반환합니다.
RELATEDTABLE	지정된 필터에 의해 수정된 컨텍스트에서 테이블 식을 평가합니다.
USERELATIONSHIP	특정 계산에 사용할 관계를 columnName1과 columnName2 사이에 있는 관계로 지정합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CROSSFILTER

아티클 • 2023. 10. 20.

두 열 사이에 있는 관계에 대한 계산에 사용할 교차 필터링 방향을 지정합니다.

구문

DAX

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

매개 변수

용어	정의
columnName1	일반적으로 사용할 관계의 여러 측면을 나타내는 표준 DAX 구문과 정규화된 기존 열의 이름입니다. 인수가 역순으로 지정되면 함수는 인수를 사용하기 전에 인수를 교환합니다. 이 인수는 식일 수 없습니다.
columnName2	일반적으로 사용할 관계의 한쪽 또는 조회 쪽을 나타내는 표준 DAX 구문과 정규화된 기존 열의 이름입니다. 인수가 역순으로 지정되면 함수는 인수를 사용하기 전에 인수를 교환합니다. 이 인수는 식일 수 없습니다.
방향	<p>사용할 교차 필터 방향입니다. 다음 중 하나여야 합니다.</p> <p>없음 - 이 관계에 따라 교차 필터링이 발생하지 않습니다.</p> <p>둘 다 - 양쪽 필터가 다른 쪽을 필터링합니다.</p> <p>OneWay - 관계 필터의 한쪽 또는 조회 쪽에서 다른 쪽을 필터링합니다. 이 옵션은 일대일 관계와 함께 사용할 수 없습니다. 조회 쪽이 명확하지 않으므로 다대다 관계에서 이 옵션을 사용하지 마세요. 대신 OneWay_LeftFiltersRight 또는 OneWay_RightFiltersLeft 사용합니다.</p> <p>OneWay_LeftFiltersRight - <columnName1> 측에 있는 필터는 <columnName2> 측을 필터링합니다. 이 옵션은 일대일 또는 다대일 관계와 함께 사용할 수 없습니다.</p> <p>OneWay_RightFiltersLeft - of <columnName2> 측에 있는 필터는 <columnName1> 측을 필터링합니다. 이 옵션은 일대일 또는 다대일 관계와 함께 사용할 수 없습니다.</p>

반환 값

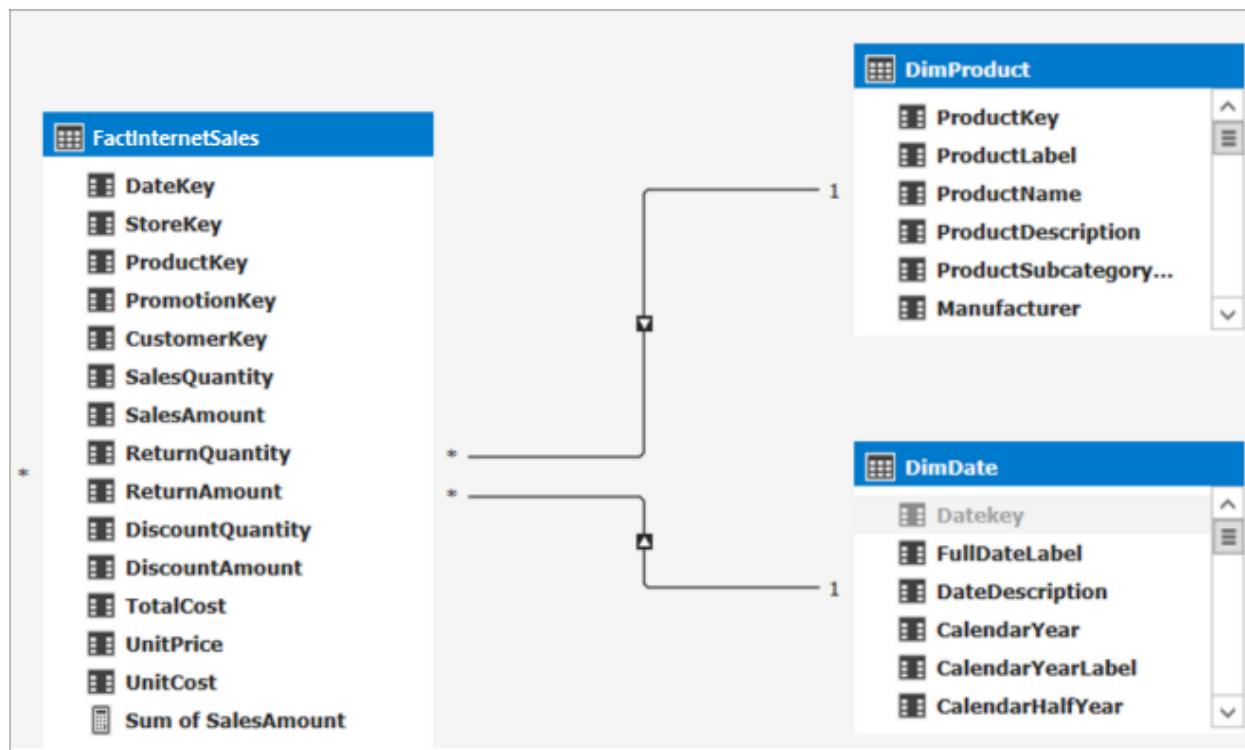
함수는 값을 반환하지 않습니다. 함수는 쿼리 기간 동안 표시된 관계의 교차 필터링 방향만 설정합니다.

설명

- 1:1 관계의 경우 한 방향과 양방향 간에 차이가 없습니다.
- CROSSFILTER는 필터를 인수로 사용하는 함수(예: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD 및 TOTALYTD 함수)에서만 사용할 수 있습니다.
- CROSSFILTER는 모델의 기존 관계를 사용하여 끝점 열로 관계를 식별합니다.
- CROSSFILTER에서 관계의 교차 필터링 설정은 중요하지 않습니다. 즉, 관계가 필터로 설정되었는지 아니면 모델의 양방향이 함수의 사용량에 영향을 주지 않는지 여부입니다. CROSSFILTER는 기존 교차 필터링 설정을 재정의합니다.
- 인수로 명명된 열이 관계의 일부가 아니거나 인수가 다른 관계에 속하는 경우 오류가 반환됩니다.
- CALCULATE 식이 중첩되고 두 개 이상의 CALCULATE 식에 CROSSFILTER 함수가 포함된 경우 가장 안쪽의 CROSSFILTER는 충돌 또는 모호성 발생 시 우선하는 식입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 모델 다이어그램에서 DimProduct와 DimDate는 FactInternetSales와 단일 방향 관계를 갖습니다.



기본적으로 연도별로 판매된 제품 수는 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.

- 관계에 대해 양방향 교차 필터링을 챕니다. 이렇게 하면 이러한 두 테이블 간의 모든 데이터에 대해 필터가 작동하는 방식이 변경됩니다.
- CROSSFILTER 함수를 사용하여 이 측정값에 대한 관계 작동 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 같습니다.

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

측정값 식에서 CROSSFILTER 함수를 사용하면 예상된 결과를 얻을 수 있습니다.

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

피드백

이 페이지가 도움이 되었나요?

Yes

No

RELATED

아티클 • 2023. 10. 20.

다른 테이블에서 관련 값을 반환합니다.

구문

DAX

`RELATED(<column>)`

매개 변수

용어	정의
column	검색할 값이 들어 있는 열입니다.

반환 값

현재 행과 관련된 단일 값입니다.

설명

- RELATED 함수를 사용하려면 현재 테이블과 관련 정보가 있는 테이블 사이에 관계가 있어야 합니다. 원하는 데이터가 포함된 열을 지정하고 함수는 기존 다대일 관계를 따라 관련 테이블의 지정된 열에서 값을 가져옵니다. 관계가 없으면 관계를 만들어야 합니다.
- RELATED 함수는 조회를 수행할 때 적용되었을 수 있는 필터에 관계없이 지정된 테이블의 모든 값을 검사합니다.
- RELATED 함수에는 행 컨텍스트가 필요합니다. 따라서 현재 행 컨텍스트가 명확하지 않은 계산 열 식이나 테이블 검색 함수를 사용하는 식의 중첩 함수로만 사용할 수 있습니다. SUMX와 같은 테이블 검색 함수는 현재 행 값의 값을 가져오고 다른 테이블에서 해당 값의 인스턴스를 검색합니다.
- RELATED 함수는 제한된 관계에서 [열을](#) 가져오는 데 사용할 수 없습니다.

예시

다음 예제에서는 미국 이외의 Internet Sales 측정값을 만들어 미국 판매를 제외하는 판매 보고서를 생성합니다. 측정값을 만들려면 salesTerritory 테이블의 미국 속하는 모든 판매를 제외하려면 InternetSales_USD 테이블을 필터링해야 합니다. 미국 SalesTerritory 테이블에 5번 표시됩니다. 각 지역에 대해 한 번씩 표시됩니다. 북서, 북동부, 중부, 남서부 및 남동부.

측정값을 만들기 위해 Internet Sales를 필터링하는 첫 번째 방법은 다음과 같은 필터 식을 추가하는 것입니다.

DAX

```
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 직관에 어긋나고 입력 오류가 발생하기 쉬우며, 나중에 기존 지역이 분할된 경우에는 작동하지 않을 수 있습니다.

더 나은 방법은 InternetSales_USD SalesTerritory 간의 기존 관계를 사용하고 국가가 미국 달라야 한다고 명시적으로 명시하는 것입니다. 이렇게 하려면 다음과 같은 필터 식을 만듭니다.

DAX

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

이 식은 RELATED 함수를 사용하여 InternetSales_USD 테이블의 키 열 SalesTerritoryKey 값부터 시작하여 SalesTerritory 테이블의 국가 값을 조회합니다. 조회 결과는 필터 함수에서 InternetSales_USD 행이 필터링되는지 여부를 확인하는 데 사용됩니다.

① 참고

예제가 작동하지 않으면 테이블 간에 관계를 만들어야 할 수 있습니다.

DAX

```
= SUMX(FILTER( 'InternetSales_USD'  
, RELATED('SalesTerritory'[SalesTerritoryCountry]))
```

```

        <>"United States"
    )
,'InternetSales_USD'[SalesAmount_USD])

```

다음 표에서는 측정값의 필터 식인 비 USA Internet Sales가 의도한 대로 작동한다는 것을 증명하기 위해 각 지역의 합계만 보여 줍니다.

Row Labels	Internet Sales	Non USA Internet Sales
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

다음은 보고서 테이블 시각적 개체에서 이 측정값을 사용한 경우 얻을 수 있는 항목을 보여줍니다.

Row Labels	Accessories	Bikes	Clothing	Grand Total
2005		\$1,526,481.95		\$1,526,481.95
2006		\$3,554,744.04		\$3,554,744.04
2007	\$156,480.18	\$5,640,106.05	\$70,142.77	\$5,866,729.00
2008	\$228,159.45	\$5,386,558.19	\$102,675.04	\$5,717,392.68
총합계	\$384,639.63	\$16,107,890.23	\$172,817.81	\$16,665,347.67

참고 항목

[RELATEDTABLE](#)

필터 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RELATEDTABLE

아티클 • 2023. 12. 14.

지정된 필터에 의해 수정된 컨텍스트에서 테이블 식을 평가합니다.

구문

DAX

`RELATEDTABLE(<tableName>)`

매개 변수

• 테이블 확장

용어	정의
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 식일 수 없습니다.

반환 값

값의 테이블입니다.

설명

- RELATEDTABLE 함수는 데이터가 필터링되는 컨텍스트를 변경하고 사용자가 지정한 새 컨텍스트에서 식을 계산합니다.
- 이 함수는 논리 식이 없는 CALCULATETABLE 함수의 바로 가기입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 RELATEDTABLE 함수를 사용하여 Product Category 테이블에서 Internet Sales를 사용하여 계산 열을 만듭니다.

DAX

```
= SUMX( RELATEDTABLE('InternetSales_USD')
    , [SalesAmount_USD])
```

다음 표에서는 결과를 보여줍니다.

 테이블 확장

Product Category Key	Product Category AlternateKey	Product Category Name	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

관련 콘텐츠

[CALCULATETABLE](#)

[필터 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

USERELATIONSHIP

아티클 • 2023. 10. 20.

특정 계산에 사용할 관계를 columnName1과 columnName2 사이에 있는 관계로 지정합니다.

구문

DAX

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

매개 변수

용어	정의
columnName1	일반적으로 사용할 관계의 여러 측면을 나타내는 표준 DAX 구문과 정규화된 기존 열의 이름입니다. 인수가 역순으로 지정되면 함수는 인수를 사용하기 전에 인수를 교환합니다. 이 인수는 식일 수 없습니다.
columnName2	일반적으로 사용할 관계의 한쪽 또는 조회 쪽을 나타내는 표준 DAX 구문과 정규화된 기존 열의 이름입니다. 인수가 역순으로 지정되면 함수는 인수를 사용하기 전에 인수를 교환합니다. 이 인수는 식일 수 없습니다.

반환 값

함수는 값을 반환하지 않습니다. 함수는 계산 기간 동안 지정된 관계만 사용하도록 설정합니다.

설명

- USERELATIONSHIP은 필터를 인수로 사용하는 함수(예: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCYEAR, TOTALMTD, TOTALQTD 및 TOTALYTD 함수)에서만 사용할 수 있습니다.
- 측정값이 포함된 테이블에 대해 행 수준 보안이 정의된 경우 USERELATIONSHIP을 사용할 수 없습니다. 예를 들어 `CALCULATE(SUM([SalesAmount]),`

```
USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))
```

DimCustomer에 대해 행 수준 보안이 정의된 경우 오류를 반환합니다.

- USERELATIONSHIP은 모델의 기존 관계를 사용하여 해당 끝 지점 열로 관계를 식별 합니다.
- USERELATIONSHIP에서 관계의 상태 중요하지 않습니다. 즉, 관계가 활성 상태인지 여부에 관계없이 함수 사용에 영향을 주지 않습니다. 관계가 비활성인 경우에도 모델에 있을 수 있지만 함수 인수에 멘션 않은 다른 활성 관계를 사용하고 재정의합니다.
- 인수로 명명된 열이 관계의 일부가 아니거나 인수가 다른 관계에 속하는 경우 오류가 반환됩니다.
- 계산에서 테이블 A를 테이블 B에 조인하는 데 여러 관계가 필요한 경우 각 관계는 다른 USERELATIONSHIP 함수에 표시되어야 합니다.
- CALCULATE 식이 중첩되고 두 개 이상의 CALCULATE 식에 USERELATIONSHIP 함수가 포함된 경우 가장 안쪽의 USERELATIONSHIP이 충돌 또는 모호성 발생 시 우선하는 식입니다.
- 최대 10개의 USERELATIONSHIP 함수를 중첩할 수 있지만 식의 중첩 수준은 이보다 더 깊을 수 있습니다. 즉, 샘플 식 `=CALCULATE(CALCULATE(CALCULATE(<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.
- 일대일 관계의 경우 USERELATIONSHIP은 관계를 한 방향으로만 활성화합니다. 특히 필터는 *columnName2*의 테이블에서 *columnName1*의 테이블로만 흐름이 진행됩니다. 양방향 교차 필터링을 사용하고자 하는 경우 동일한 계산에서 방향이 반대인 USERELATIONSHIP 2개를 사용할 수 있습니다. 예: `CALCULATE(..., USERELATIONSHIP(T1[K], T2[K]), USERELATIONSHIP(T2[K], T1[K]))`.

예시

다음 샘플에서는 InternetSales와 DateTime 테이블 간의 기본 활성 관계를 재정의하는 방법을 보여 줍니다. 기본 관계는 OrderDate 열, InternetSales 테이블 및 DateTime 테이블의 Date 열 사이에 존재합니다.

인터넷 판매 합계를 계산하고 기존 OrderDate 대신 ShippingDate로 조각화할 수 있도록 하려면 다음 식을 사용하여 측정값인 [ShippingDate의 InternetSales]를 만듭니다.

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

InternetSales[ShipmentDate]와 DateTime[Date] 간의 관계는 존재해야 하며 활성 관계가 아니어야 합니다. 또한 InternetSales[OrderDate]와 DateTime[Date] 간의 관계가 존재해야 하며 활성 관계여야 합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

통계 함수

아티클 • 2023. 10. 20.

통계 함수는 표준 편차 및 순열 수와 같은 통계 분포 및 확률과 관련된 값을 계산합니다.

이 범주에서

함수	설명
Beta. Dist	베타 분포를 반환합니다.
Beta. Inv	베타 누적 확률 밀도 함수(BETA)의 역함수 값을 반환합니다. DIST).
CHISQ. Dist	카이 제곱 분포를 반환합니다.
CHISQ. Dist. Rt	카이 제곱 분포의 오른쪽 꼬리 확률을 반환합니다.
CHISQ. Inv	카이 제곱 분포의 왼쪽 꼬리 확률의 역방향 확률을 반환합니다.
CHISQ. Inv. Rt	카이 제곱 분포의 오른쪽 꼬리 확률의 역방향 확률을 반환합니다.
결합(COMBIN)	지정된 수의 항목에 대한 조합 수를 반환합니다.
COMBINA	지정된 수의 항목에 대한 조합 수(반복 포함)를 반환합니다.
자신감. 규범	신뢰 구간은 값의 범위입니다.
자신감. T	Student의 t 분포를 사용하여 모집단 평균에 대한 신뢰 구간을 반환합니다.
EXPON. Dist	지수 분포를 반환합니다.
GEOMEAN	열에 있는 숫자의 기하 평균을 반환합니다.
GEOMEANX	테이블의 각 행에 대해 계산된 식의 기하 평균을 반환합니다.
LINEST	최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산합니다.
LINESTX	최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산합니다. 테이블의 각 행에 대해 계산된 식의 데이터 결과입니다.
중간	열의 숫자 중앙값을 반환합니다.
MEDIANX	테이블의 각 행에 대해 계산되는 식의 중앙값 수를 반환합니다.
규범. Dist	지정된 평균 및 표준 편차에 대한 정규 분포를 반환합니다.
규범. Inv	지정된 평균 및 표준 편차에 대한 일반 누적 분포의 역함수입니다.
규범. S.DIST	표준 정규 분포를 반환합니다(평균이 0이고 표준 편차가 1인 경우).

함수	설명
규범. S.INV	표준 정규 누적 분포의 역함수 값을 반환합니다.
백분위. Exc	범위에서 값의 k번째 백분위수(k는 0..1 범위에서 제외)를 반환합니다.
백분위. Inc	범위에서 값의 k번째 백분위수(k는 0..1 범위 포함)를 반환합니다.
PERCENTILEX. Exc	테이블의 각 행에 대해 계산되는 식의 백분위수 수를 반환합니다.
PERCENTILEX. Inc	테이블의 각 행에 대해 계산되는 식의 백분위수 수를 반환합니다.
PERMUT	숫자 개체에서 선택할 수 있는 지정된 수의 개체에 대한 순열 수를 반환합니다.
포아송. Dist	포아송 분포를 반환합니다.
순위. Eq	숫자 목록에서 숫자의 순위를 반환합니다.
RANKX	테이블 인수의 각 행에 대한 숫자 목록의 숫자 순위를 반환합니다.
SAMPLE	지정한 테이블에서 N행의 샘플을 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.
T.DIST	Student의 왼쪽 단측 t 분포를 반환합니다.
T.DIST.2T	두 꼬리 학생 t 분포를 반환합니다.
T.DIST. Rt	오른쪽 꼬리 학생 t 분포를 반환합니다.
T.INV	Student t-distribution의 왼쪽 역방향을 반환합니다.
T.INV.2t	Student t-distribution의 2측 역방향을 반환합니다.
Var. P	전체 모집단의 분산을 반환합니다.
Var. S	샘플 모집단의 분산을 반환합니다.
VARX. P	전체 모집단의 분산을 반환합니다.
VARX. S	샘플 모집단의 분산을 반환합니다.

이 페이지가 도움이 되었나요?

 Yes

 No

Beta. Dist

아티클 • 2023. 10. 20.

베타 분포를 반환합니다. 베타 분포는 일반적으로 사람들이 텔레비전을 시청하는 날의 분수와 같이 샘플에서 무언가의 비율의 변화를 연구하는 데 사용됩니다.

구문

DAX

```
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

매개 변수

용어	정의
x	함수를 평가할 A와 B 사이의 값입니다.
알파	분포의 매개 변수입니다.
베타	분포의 매개 변수입니다.
A	선택 사항. x 간격의 하한입니다.
b	선택 사항. x 간격의 상한입니다.

반환 값

베타 분포를 반환합니다.

설명

- 숫자가 아닌 인수가 있으면 BETA입니다. DIST는 #VALUE 반환합니다! 오류 값입니다.
- 정수가 아닌 인수가 있으면 반올림됩니다.
- 알파 ≤ 0 또는 베타 ≤ 0이면 BETA입니다. DIST는 #NUM 반환합니다! 오류 값입니다.
- x < A, x > B 또는 A = B, BETA인 경우 DIST는 #NUM 반환합니다! 오류 값입니다.

- A 및 B에 대한 값을 생략하면 BETA입니다. DIST는 표준 누적 베타 분포를 사용하므로 A = 0 및 B = 1입니다.
 - 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.
-

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Beta. Inv

아티클 • 2023. 10. 20.

베타 누적 확률 밀도 함수(BETA)의 역함수 값을 반환합니다. DIST).

probability = BETA인 경우 DIST(x,... TRUE) 다음 베타. INV(probability,...) = x. 베타 배포는 예상 완료 시간 및 가변성을 고려하여 프로젝트 계획에서 가능한 완료 시간을 모델링하는 데 사용할 수 있습니다.

구문

DAX

BETA.INV(probability,alpha,beta,[A],[B])

매개 변수

용어	정의
확률	베타 분포와 관련된 확률입니다.
알파	분포의 매개 변수입니다.
베타	분포 매개 변수입니다.
A	선택 사항. x 간격의 하한입니다.
b	선택 사항. x 간격의 상한입니다.

반환 값

베타 누적 확률 밀도 함수(BETA)의 역함수 값을 반환합니다. DIST).

설명

- 숫자가 아닌 인수가 있으면 BETA입니다. INV는 #VALUE 반환합니다! 오류 값입니다.
- 정수가 아닌 인수가 있으면 반올림됩니다.
- 알파 ≤ 0 또는 베타 ≤ 0이면 BETA입니다. INV는 #NUM 반환합니다! 오류 값입니다.
- 확률 ≤ 0 또는 확률 > 1이면 BETA입니다. INV는 #NUM 반환합니다! 오류 값입니다.

- A 및 B에 대한 값을 생략하면 BETA입니다. INV는 표준 누적 베타 분포를 사용하므로 A = 0 및 B = 1입니다.
 - 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.
-

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CHISQ. Dist

아티클 • 2023. 10. 20.

카이 제곱 분포를 반환합니다.

카이 제곱 분포는 일반적으로 사람들이 텔레비전을 보는 데 소비하는 일의 분수와 같은 샘플에 걸쳐 무언가의 비율의 변화를 연구하는 데 사용됩니다.

구문

DAX

CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)

매개 변수

용어	정의
x	분포를 평가할 값입니다.
Deg_freedom	자유도의 수입니다.
cumulative	함수의 형식을 결정하는 논리 값입니다. 누적이 TRUE이면 CHISQ입니다. DIST는 누적 분포 함수를 반환합니다. FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

카이 제곱 분포입니다.

설명

- x 또는 deg_freedom 숫자가 아니면 오류가 반환됩니다.
- deg_freedom 정수가 아니면 반올림됩니다.
- x < 0인 경우 오류가 반환됩니다.
- deg_freedom < 1 또는 deg_freedom > 10^10인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는

DAX

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

반환

[값]

0.632120558828558

피드백

이 페이지가 도움이 되었나요?

Yes

No

CHISQ. Dist. Rt

아티클 • 2023. 10. 20.

카이 제곱 분포의 오른쪽 꼬리 확률을 반환합니다.

카이 제곱 분포는 카이 제곱 테스트와 연결됩니다. 카이 제곱 테스트를 사용하여 관찰된 값과 예상 값을 비교합니다. 예를 들어, 유전 실험은 다음 세대의 식물이 특정 색 집합을 나타낼 것이라고 가설할 수 있습니다. 관찰된 결과를 예상 결과와 비교하여 원래 가설이 유효한지 여부를 결정할 수 있습니다.

구문

DAX

`CHISQ.DIST.RT(<x>, <deg_freedom>)`

매개 변수

용어	정의
x	분포를 평가할 값입니다.
Deg_freedom	자유도의 수입니다.

반환 값

카이 제곱 분포의 오른쪽 꼬리 확률입니다.

설명

- x 또는 deg_freedom 숫자가 아니면 오류가 반환됩니다.
- deg_freedom 정수가 아니면 반올림됩니다.
- x < 0인 경우 오류가 반환됩니다.
- deg_freedom < 1 또는 deg_freedom > 10^10인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는

DAX

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

반환

[값]

0.84914503608461

피드백

이 페이지가 도움이 되었나요?

Yes

No

CHISQ. Inv

아티클 • 2023. 10. 20.

카이 제곱 분포의 왼쪽 꼬리 확률의 역방향 확률을 반환합니다.

카이 제곱 분포는 일반적으로 사람들이 텔레비전을 보는 데 소비하는 일의 분수와 같은 샘플에 걸쳐 무언가의 비율의 변화를 연구하는 데 사용됩니다.

구문

DAX

CHISQ.INV(probability,deg_freedom)

매개 변수

용어	정의
확률	카이 제곱 분포와 관련된 확률입니다.
Deg_freedom	자유도의 수입니다.

반환 값

카이 제곱 분포의 왼쪽 꼬리 확률의 역방향 확률을 반환합니다.

설명

- 인수가 숫자가 아니면 CHISQ입니다. INV는 #VALUE 반환합니다! 오류 값입니다.
- probability < 0 또는 probability > 1이면 CHISQ.INV는 #NUM! 오류 값입니다.
- deg_freedom 정수가 아니면 반올림됩니다.
- deg_freedom < 0 또는 deg_freedom > 10^10이면 CHISQ.INV는 #NUM! 오류 값입니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

수식	설명	결과
= CHISQ. INV(0.93,1)	자유도 1도를 사용하여 0.93에 대한 카이 제곱 분포의 왼쪽 꼬리 확률과 반대입니다.	5.318520074
= CHISQ. INV(0.6,2)	자유도 2도를 사용하여 0.6에 대한 카이 제곱 분포의 왼쪽 꼬리 확률과 반대입니다.	1.832581464

피드백

이 페이지가 도움이 되었나요?

Yes

No

CHISQ. Inv. Rt

아티클 • 2023. 10. 20.

카이 제곱 분포의 오른쪽 꼬리 확률의 역방향 확률을 반환합니다.

probability = CHISQ인 경우 Dist. RT(x, \dots), then CHISQ. Inv. RT(probability, ...) = x . 이 함수를 사용하여 원래 가설이 유효한지 여부를 결정하기 위해 관찰된 결과를 예상 결과와 비교합니다.

구문

DAX

CHISQ.INV.RT(probability,deg_freedom)

매개 변수

용어	정의
확률	카이 제곱 분포와 관련된 확률입니다.
Deg_freedom	자유도의 수입니다.

반환 값

카이 제곱 분포의 오른쪽 꼬리 확률의 역방향 확률을 반환합니다.

설명

- 두 인수 중 하나가 숫자가 아니면 CHISQ입니다. Inv. RT는 #VALUE 반환합니다! 오류 값입니다.
- probability < 0 또는 probability > 1이면 CHISQ입니다. Inv. RT는 #NUM 반환합니다! 오류 값입니다.
- deg_freedom 정수가 아니면 반올림됩니다.
- deg_freedom < 1이면 CHISQ입니다. Inv. RT는 #NUM 반환합니다! 오류 값입니다.

- 확률 값이 지정되면 CHISQ입니다. Inv. RT는 CHISQ와 같은 x 값을 찾습니다. Dist. RT(x , deg_freedom) = probability. 따라서 CHISQ의 정밀도입니다. Inv. RT는 CHISQ의 정밀도에 따라 달라집니다. DIST.RT. CHISQ. Inv. RT는 반복 검색 기술을 사용합니다. 64회 반복 후 검색이 수렴되지 않은 경우 함수는 #N/A 오류 값을 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

결합(COMBIN)

아티클 • 2023. 10. 20.

지정된 수의 항목에 대한 조합 수를 반환합니다. COMBIN을 사용하여 지정된 수의 항목에 대해 가능한 총 그룹 수를 결정합니다.

구문

DAX

`COMBIN(number, number_chosen)`

매개 변수

용어	정의
번호	항목의 수입니다.
number_chosen	각 조합의 항목 수입니다.

반환 값

지정된 수의 항목에 대한 조합 수를 반환합니다.

설명

- 숫자 인수는 정수로 잘립니다.
- 두 인수 중 하나가 숫자가 아니면 COMBIN에서는 #VALUE! 오류 값입니다.
- number < 0, number_chosen < 0 또는 number < number_chosen 경우 COMBIN은 #NUM 반환합니다. 오류 값입니다.
- 조합은 내부 순서에 관계없이 항목의 집합 또는 하위 집합입니다. 조합은 내부 순서가 중요한 순열과 다릅니다.
- 조합 수는 다음과 같습니다. 여기서 number = n 및 number_chosen = k입니다.

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

Where

$$P_{k,n} = \frac{n!}{(n - k)!}$$

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

예시

수식	설명	결과
= COMBIN(8,2)	8명의 후보자로 구성될 수 있는 가능한 2인 팀.	28

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COMBINA

아티클 • 2023. 10. 20.

지정된 수의 항목에 대한 조합 수(반복 포함)를 반환합니다.

구문

DAX

`COMBINA(number, number_chosen)`

매개 변수

용어	정의
번호	0보다 크거나 같고 Number_chosen 보다 크거나 같아야 합니다. 정수가 아닌 값은 잘립니다.
number_chosen	0보다 크거나 같아야 합니다. 정수가 아닌 값은 잘립니다.

반환 값

지정된 수의 항목에 대한 조합 수(반복 포함)를 반환합니다.

설명

- 두 인수의 값이 제약 조건을 벗어나면 COMBINA는 #NUM 반환합니다. 오류 값입니다.
- 두 인수 중 하나가 숫자가 아닌 값이면 COMBINA는 #VALUE 반환합니다. 오류 값입니다.
- 다음 수식이 사용됩니다. 여기서 N 는 Number이고 M 는 Number_chosen.

$$\binom{N + M - 1}{N - 1}$$

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

예시

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

피드백

이 페이지가 도움이 되었나요?

Yes

No

자신감. 규범

아티클 • 2023. 10. 20.

신뢰 구간은 값의 범위입니다. 샘플 평균 x 는 이 범위의 중심에 있고 범위는 $x \pm \text{CONFIDENCE.NORM}$ 입니다. 예를 들어 x 가 매일을 통해 주문된 제품에 대한 배달 시간의 샘플 평균인 경우 x 는 신뢰도를 \pm . NORM은 모집단의 범위입니다. 모집단 평균인 μ_0 의 경우, 이 범위에서 표본 평균을 x 보다 μ_0 에서 더 많이 얻을 확률은 알파보다 크며, 모집단 평균인 μ_0 의 경우 이 범위가 아닌 경우 x 보다 μ_0 이상에서 표본 평균을 얻을 확률이 알파보다 작습니다. 즉, 모집단 평균이 μ_0 이라는 가설의 유의 수준 알파에서 x , standard_dev 및 크기를 사용하여 두 꼬리 테스트를 구성한다고 가정합니다. 그런 다음 μ_0 이 신뢰 구간에 있으면 해당 가설을 거부하지 않으며 μ_0 이 신뢰 구간에 없는 경우 해당 가설을 거부합니다. 신뢰 구간을 사용하면 다음 패키지가 신뢰 구간에 있는 배달 시간이 소요될 확률 1 - 알파가 있음을 유추할 수 없습니다.

구문

DAX

`CONFIDENCE.NORM(alpha,standard_dev,size)`

매개 변수

용어	정의
알파	신뢰도 수준을 계산하는 데 사용되는 중요도 수준입니다. 신뢰 수준이 $100*(1 - alpha)\%$ 와 같거나, 즉 0.05의 알파는 95% 신뢰 수준을 나타냅니다.
standard_dev	데이터 범위에 대한 모집단 표준 편차이며 알려진 것으로 간주됩니다.
standard_dev, 크기	샘플 크기입니다.

반환 값

값 범위

설명

- 숫자가 아닌 인수가 있으면 CONFIDENCE입니다. NORM은 #VALUE 반환합니다! 오류 값입니다.
- 알파 ≤ 0 또는 알파 ≥ 1 이면 CONFIDENCE입니다. NORM은 #NUM 반환합니다! 오류 값입니다.
- standard_dev ≤ 0 이면 신뢰도입니다. NORM은 #NUM 반환합니다! 오류 값입니다.
- 크기가 정수가 아니면 반올림됩니다.
- 크기 $<$ 1이면 CONFIDENCE입니다. NORM은 #NUM 반환합니다! 오류 값입니다.
- 알파가 0.05라고 가정하는 경우 표준 표준 곡선($1 - \alpha$) 또는 95% 아래의 영역을 계산해야 합니다. 이 값은 ± 1.96 입니다. 따라서 신뢰 구간은 다음과 같습니다.

$$\bar{x} \pm 1.96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

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

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

자신감. T

아티클 • 2023. 10. 20.

Student의 t 분포를 사용하여 모집단 평균에 대한 신뢰 구간을 반환합니다.

구문

DAX

`CONFIDENCE.T(alpha,standard_dev,size)`

매개 변수

용어	정의
알파	신뢰도 수준을 계산하는 데 사용되는 중요도 수준입니다. 신뢰 수준이 $100*(1 - \alpha)$ %와 같거나, 즉 0.05의 알파는 95% 신뢰 수준을 나타냅니다.
standard_dev	데이터 범위에 대한 모집단 표준 편차이며 알려진 것으로 간주됩니다.
size	샘플 크기입니다.

반환 값

Student의 t 분포를 사용하여 모집단 평균에 대한 신뢰 구간을 반환합니다.

설명

- 숫자가 아닌 인수가 있으면 CONFIDENCE입니다. T는 #VALUE 반환합니다! 오류 값입니다.
- 알파 ≤ 0 또는 알파 ≥ 1 이면 CONFIDENCE입니다. T는 #NUM 반환합니다! 오류 값입니다.
- standard_dev ≤ 0 이면 신뢰도입니다. T는 #NUM 반환합니다! 오류 값입니다.
- 크기가 정수가 아니면 반올림됩니다.
- 크기가 1이면 CONFIDENCE입니다. T는 #DIV/0을 반환합니다. 오류 값입니다.

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

예시

수식	설명	결과
= 신뢰도입니다. $T(0.05, 1, 50)$	표본 크기 50을 기준으로 모집단 평균에 대한 신뢰 구간이며, 5% 중요도 수준과 표준 편차는 1입니다. 이는 Student의 t-distribution을 기반으로 합니다.	0.284196855

피드백

이 페이지가 도움이 되었나요?

Yes

No

EXPON. Dist

아티클 • 2023. 10. 20.

지수 분포를 반환합니다. EXPON을 사용합니다. DIST는 자동화된 은행 출납자가 현금을 전달하는 데 걸리는 시간과 같은 이벤트 간의 시간을 모델링합니다. 예를 들어 EXPON을 사용할 수 있습니다. 프로세스에 최대 1분이 소요되는 확률을 결정하는 DIST입니다.

구문

DAX

`EXPON.DIST(x,lambda,cumulative)`

매개 변수

용어	정의
x	필수입니다. 함수의 값입니다.
람다	필수입니다. 매개 변수 값입니다.
cumulative	필수입니다. 제공할 지수 함수의 형태를 나타내는 논리 값입니다. 누적이 TRUE이면 EXPON입니다. DIST는 누적 분포 함수를 반환합니다. FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

지수 분포를 반환합니다.

설명

- x 또는 람다가 숫자가 아니면 EXPON입니다. DIST는 #VALUE 반환합니다! 오류 값입니다.
- x 또는 람다가 정수가 아니면 반올림됩니다.
- x < 0이면 EXPON입니다. DIST는 #NUM 반환합니다! 오류 값입니다.
- 람다 ≤ 0이면 EXPON입니다. DIST는 #NUM 반환합니다! 오류 값입니다.
- 확률 밀도 함수에 대한 수식은 다음과 같습니다.

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

- 누적 분포 함수의 수식은 다음과 같습니다.

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

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

피드백

이 페이지가 도움이 되었나요?

Yes

No

GEOMEAN

아티클 • 2023. 12. 14.

열에 있는 숫자의 기하 평균을 반환합니다.

테이블의 각 행에 대해 계산된 식의 기하 평균을 반환하려면 [GEOMEANX 함수를 사용합니다.](#)

구문

DAX

GEOMEAN(<column>)

매개 변수

 테이블 확장

용어	정의
column	기하학적 평균을 계산할 숫자가 들어 있는 열입니다.

반환 값

10진수입니다.

설명

- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- GEOMEAN(Table[Column])은 GEOMEANX(Table, Table[Column])
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 Investment 테이블에서 Return 열의 기하 평균을 계산합니다.

DAX

= GEOMEAN(Investment[Return])

관련 콘텐츠

[GEOMEANX 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

GEOMEANX

아티클 • 2023. 12. 14.

테이블의 각 행에 대해 계산된 식의 기하 평균을 반환합니다.

열에 있는 숫자의 기하 평균을 반환하려면 GEOMEAN 함수를 사용합니다.

구문

DAX

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

매개 변수

 테이블 확장

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- GEOMEANX 함수는 테이블 또는 테이블을 반환하는 식을 첫 번째 인수로 사용합니다. 두 번째 인수는 기하학적 평균을 계산하려는 숫자 또는 열로 계산되는 식을 포함하는 열입니다.
- 열의 숫자만 계산됩니다. 공백, 논리 값 및 텍스트는 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 Investments 테이블에서 ReturnPct 열의 기하 평균을 계산합니다.

DAX

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

관련 콘텐츠

[GEOMEAN 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LINEST

아티클 • 2023. 10. 20.

최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산한 다음, 선을 설명하는 테이블을 반환합니다. 선의 수식은 $y = \text{Slope1} * x_1 + \text{Slope2} * x_2 + \dots + \text{Intercept}$ 형식입니다.

구문

DAX

```
LINEST ( <columnY>, <columnX>[ , ...][ , <const> ] )
```

매개 변수

용어	정의
columnY	알려진 y 값의 열입니다. 스칼라 형식이 있어야 합니다.
columnX	알려진 x 값의 열입니다. 스칼라 형식이 있어야 합니다. 하나 이상을 제공해야 합니다.
const	(선택 사항) 상수 인터셉트를 0으로 강제 적용할지 여부를 지정하는 상수 TRUE/FALSE 값입니다. TRUE를 지정하거나 생략하면 인터셉트 값이 정상적으로 계산됩니다. FALSE이면 Intercept 값이 0으로 설정됩니다.

반환 값

줄을 설명하는 단일 행 테이블과 추가 통계입니다. 사용 가능한 열은 다음과 같습니다.

- **Slope1, Slope2, ..., SlopeN:** 각 x-값에 해당하는 계수입니다.
- **절편:** 절편 값;
- **StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN:** 계수 Slope1, Slope2, ..., SlopeN;
- **StandardErrorIntercept:** 상수 인터셉트의 표준 오류 값입니다.
- **CoefficientOfDetermination:** 결정 계수(r^2)입니다. 예상 값과 실제 y-값 및 값의 범위를 0에서 1까지 비교합니다. 값이 높을수록 샘플의 상관 관계가 높아질 수 있습니다.
- **StandardError:** y 예상치에 대한 표준 오류입니다.
- **FStatistic:** F 통계 또는 F 관찰 값입니다. F 통계를 사용하여 종속 변수와 독립 변수 간의 관찰된 관계가 우연히 발생하는지 여부를 확인합니다.
- **DegreesOfFreedom:** 자유도입니다. 이 값을 사용하여 통계 테이블에서 F-Critical 값을 찾고 모델에 대한 신뢰도 수준을 결정할 수 있습니다.
- **RegressionSumOfSquares:** 제곱의 회귀 합계입니다.
- **ResidualSumOfSquares:** 정사각형의 잔여 합계입니다.

설명

<columnY> 와 <columnX>는 모두 동일한 테이블에 속해야 합니다.

예 1

다음 DAX 쿼리:

```
DAX  
  
EVALUATE LINEST(  
    'FactInternetSales'[SalesAmount],  
    'FactInternetSales'[TotalProductCost]  
)
```

열이 10개인 단일 행 테이블을 반환합니다.

Slope1	가로채기	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
1.67703250456677	6.34550460373026	0.000448675725548806	0.279131821917317	0.995695557281456
StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
60.9171030357485	13970688.6139993	60396	51843736761.658	224123120.339218

- Slope1 및 Intercept: 계산된 선형 모델의 계수입니다.
- StandardErrorSlope1 및 StandardErrorIntercept: 위의 계수에 대한 표준 오류 값입니다.
- CoefficientOfDetermination, StandardError, FStatistic, DegreesOfFreedom, RegressionSumOfSquares 및 ResidualSumOfSquares: 모델에 대한 회귀 통계입니다.

지정된 인터넷 판매의 경우 이 모델은 다음 수식으로 판매 금액을 예측합니다.

```
SalesAmount = Slope1 * TotalProductCost + Intercept
```

예제 2

다음 DAX 쿼리:

```
DAX  
  
EVALUATE LINEST(  
    'DimCustomer'[TotalSalesAmount],  
    'DimCustomer'[YearlyIncome],  
    'DimCustomer'[TotalChildren],  
    'DimCustomer'[BirthDate]  
)
```

열이 14개인 단일 행 테이블을 반환합니다.

- Slope1
- Slope2
- Slope3
- 가로채기
- StandardErrorSlope1

- StandardErrorSlope2
- StandardErrorSlope3
- StandardErrorIntercept
- CoefficientOfDetermination
- StandardError
- FStatistic
- DegreesOfFreedom
- RegressionSumOfSquares
- ResidualSumOfSquares

지정된 고객의 경우 이 모델은 다음 수식으로 총 판매량을 예측합니다(생년월일은 자동으로 숫자로 변환 됨).

```
TotalSalesAmount = Slope1 * YearlyIncome + Slope2 * TotalChildren + Slope3 * BirthDate +
Intercept
```

참고 항목

[LINESTX](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LINESTX

아티클 • 2023. 10. 20.

최소 제곱 메서드를 사용하여 지정된 데이터에 가장 적합한 직선을 계산한 다음, 선을 설명하는 테이블을 반환합니다. 테이블의 각 행에 대해 계산된 식의 데이터 결과입니다. 선의 수식은 $y = \text{Slope1} * x_1 + \text{Slope2} * x_2 + \dots + \text{Intercept}$ 형식입니다.

구문

DAX

```
LINESTX ( <table>, <expressionY>, <expressionX>[ , ...][ , <const>] )
```

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
expressionY	알려진 y 값을 얻기 위해 테이블의 각 행에 대해 계산할 식입니다. 스칼라 형식이 있어야 합니다.
expressionX	테이블의 각 행에 대해 계산할 식으로 알려진 x 값을 가져옵니다. 스칼라 형식이 있어야 합니다. 하나 이상을 제공해야 합니다.
const	(선택 사항) 상수 인터셉트를 0으로 강제 적용할지 여부를 지정하는 상수 TRUE/FALSE 값입니다. TRUE를 지정하거나 생략하면 인터셉트 값이 정상적으로 계산됩니다. FALSE이면 Intercept 값이 0으로 설정됩니다.

반환 값

줄을 설명하는 단일 행 테이블과 추가 통계입니다. 사용 가능한 열은 다음과 같습니다.

- Slope1, Slope2, ..., SlopeN:** 각 x-값에 해당하는 계수입니다.
- 절편:** 절편 값;
- StandardErrorSlope1, StandardErrorSlope2, ..., StandardErrorSlopeN:** 계수 Slope1, Slope2, ..., SlopeN;
- StandardErrorIntercept:** 상수 인터셉트의 표준 오류 값입니다.
- CoefficientOfDetermination:** 결정 계수(r^2)입니다. 예상 값과 실제 y-값 및 값의 범위를 0에서 1까지 비교합니다. 값이 높을수록 샘플의 상관 관계가 높아질 수 있습니다.
- StandardError:** y 예상치에 대한 표준 오류입니다.
- FStatistic:** F 통계 또는 F 관찰 값입니다. F 통계를 사용하여 종속 변수와 독립 변수 간의 관찰된 관계가 우연히 발생하는지 여부를 확인합니다.

- **DegreesOfFreedom**: 자유도입니다. 이 값을 사용하여 통계 테이블에서 F-Critical 값을 찾고 모델에 대한 신뢰도 수준을 결정할 수 있습니다.
- **RegressionSumOfSquares**: 제곱의 회귀 합계입니다.
- **ResidualSumOfSquares**: 정사각형의 잔여 합계입니다.

예 1

다음 DAX 쿼리:

```
DAX
DEFINE VAR TotalSalesByRegion = SUMMARIZECOLUMNS(
    'Sales Territory'[Sales Territory Key],
    'Sales Territory'[Population],
    "Total Sales", SUM(Sales[Sales Amount]))
)
EVALUATE LINESTX(
    'TotalSalesByRegion',
    [Total Sales],
    [Population]
)
```

열이 10개인 단일 행 테이블을 반환합니다.

Slope1	가로채기	StandardErrorSlope1	StandardErrorIntercept	CoefficientOfDetermination
6.42271517588	-410592.76216	0.24959467764561	307826.343996223	0.973535860750193

StandardError	FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
630758.1747292	662.165707642	18	263446517001130	7161405749781.07

- **Slope1** 및 **Intercept**: 계산된 선형 모델의 계수입니다.
- **StandardErrorSlope1** 및 **StandardErrorIntercept**: 위의 계수에 대한 표준 오류 값입니다.
- **CoefficientOfDetermination**, **StandardError**, **FStatistic**, **DegreesOfFreedom**, **RegressionSumOfSquares** 및 **ResidualSumOfSquares**: 모델에 대한 회귀 통계입니다.

지정된 판매 지역의 경우 이 모델은 다음 수식을 통해 총 매출을 예측합니다.

```
Total Sales = Slope1 * Population + Intercept
```

예제 2

다음 DAX 쿼리:

```
DAX
```

```

DEFINE VAR TotalSalesByCustomer = SUMMARIZECOLUMNS(
    'Customer'[Customer ID],
    'Customer'[Age],
    'Customer'[NumOfChildren],
    "Total Sales", SUM(Sales[Sales Amount])
)
EVALUATE LINESTX(
    'TotalSalesByCustomer',
    [Total Sales],
    [Age],
    [NumOfChildren]
)

```

열이 12개인 단일 행 테이블을 반환합니다.

Slope1	Slope2	가로채기	StandardErrorSlope1
69.0435458093763	33.005949841721	-871.118539339539	0.872588875481658
StandardErrorSlope2	StandardErrorIntercept	CoefficientOfDetermination	StandardError
6.21158863903435	26.726292527427	0.984892920482022	68.5715034014342
FStatistic	DegreesOfFreedom	RegressionSumOfSquares	ResidualSumOfSquares
3161.91535144391	97	29734974.9782379	456098.954637092

지정된 고객의 경우 이 모델은 다음 수식을 통해 총 매출을 예측합니다.

```
Total Sales = Slope1 * Age + Slope2 * NumOfChildren + Intercept
```

참고 항목

[LINEST](#)

[통계 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

중간

아티클 • 2023. 12. 14.

열의 숫자 중앙값을 반환합니다.

테이블의 각 행에 대해 계산된 표현의 중앙값을 반환하려면 MEDIANX 함수를 사용합니다.

구문

DAX

`MEDIAN(<column>)`

매개 변수

 테이블 확장

용어	정의
column	중앙값을 계산할 숫자가 들어 있는 열입니다.

반환 값

10진수입니다.

설명

- 열의 숫자만 계산됩니다. 공백은 무시됩니다. 논리 값, 날짜 및 텍스트는 지원되지 않습니다.
- MEDIAN(Table[Column])은 MEDIANX(Table, Table[Column])와 동일합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 Customers라는 테이블에서 Age라는 열의 중앙값을 계산합니다.

DAX

```
= MEDIAN( Customers[Age] )
```

관련 콘텐츠

[MEDIANX 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MEDIANX

아티클 • 2023. 12. 14.

테이블의 각 행에 대해 계산되는 식의 중앙값 수를 반환합니다.

열에 있는 숫자의 중앙값을 반환하려면 MEDIAN 함수를 사용합니다.

구문

DAX

MEDIANX(<table>, <expression>)

매개 변수

 테이블 확장

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.

반환 값

10진수입니다.

설명

- MEDIANX 함수는 테이블 또는 테이블을 반환하는 식을 첫 번째 인수로 사용합니다. 두 번째 인수는 중앙값을 계산할 숫자가 들어 있는 열이거나 열로 계산되는 식입니다.
- 열의 숫자만 계산됩니다.
- 논리 값과 텍스트는 무시됩니다.
- MEDIANX는 공백을 무시하지 않습니다. 그러나 MEDIAN는 공백을 무시합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음은 미국에 거주하는 고객의 평균 연령을 계산합니다.

DAX

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country] = "USA" ) ),  
Customers[Age] )
```

관련 콘텐츠

[MEDIAN 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

규범. Dist

아티클 • 2023. 12. 14.

지정된 평균 및 표준 편차에 대한 정규 분포를 반환합니다.

구문

DAX

```
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

매개 변수

 테이블 확장

용어	정의
X	배포할 값입니다.
평균	분포의 산술 평균입니다.
Standard_dev	분포의 표준 편차입니다.
누적*	함수의 형식을 결정하는 논리 값입니다. 누적이 TRUE이면 NORM입니다. DIST는 누적 분포 함수를 반환합니다. 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

DAX

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

반환

 테이블 확장

[값]

0.908788780274132

관련 콘텐츠

[규범. S.DIST 함수](#)

[규범. INV 함수](#)

[규범. S.INV](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

규범. Inv

아티클 • 2023. 12. 14.

지정된 평균 및 표준 편차에 대한 일반 누적 분포의 역함수입니다.

구문

DAX

```
NORM.INV(Probability, Mean, Standard_dev)
```

매개 변수

 테이블 확장

용어	정의
확률	정규 분포에 해당하는 확률입니다.
평균	분포의 산술 평균입니다.
Standard_dev	분포의 표준 편차입니다.

반환 값

지정된 평균 및 표준 편차에 대한 정규 누적 분포의 역함수 분포를 반환합니다.

설명

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

예시

DAX

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

반환

[값]

42.00000200956628780274132

관련 콘텐츠

[규범. S.INV](#)

[규범. S.DIST 함수](#)

[규범. DIST 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

규범. S.DIST

아티클 • 2023. 12. 14.

표준 정규 분포를 반환합니다(평균이 0이고 표준 편차가 1인 경우).

구문

DAX

NORM.S.DIST(Z, Cumulative)

매개 변수

 테이블 확장

용어
정의

Z 배포할 값입니다.

누적
누적은 함수의 형태를 결정하는 논리 값입니다. 누적이 TRUE이면 NORM입니다. S.DIST는 누적 분포 함수를 반환합니다. FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

표준 정규 분포(평균이 0이고 표준 편차가 1인 경우)

설명

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

예시

DAX

EVALUATE { NORM.S.DIST(1.333333, TRUE) }

반환

[값]

0.908788725604095

관련 콘텐츠

[규범. INV 함수](#)

[규범. DIST 함수](#)

[규범. S.INV](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

규범. S.INV

아티클 • 2023. 12. 14.

표준 정규 누적 분포의 역함수 값을 반환합니다. 분포의 평균은 0이고 표준 편자는 1입니다.

구문

DAX

NORM.S.INV(Probability)

매개 변수

[+] 테이블 확장

용어	정의
확률	정규 분포에 해당하는 확률입니다.

반환 값

표준 정규 누적 분포의 역함수입니다. 분포의 평균은 0이고 표준 편자는 1입니다.

설명

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

예시

DAX

```
EVALUATE { NORM.S.INV(0.908789) }
```

반환

[값]

1.33333467304411

관련 콘텐츠

[규범. Inv](#)

[규범. S.DIST 함수](#)

[규범. DIST 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

백분위. EXC

아티클 • 2023. 12. 14.

범위에서 값의 k번째 백분위수(k는 0..1 범위에서 제외)를 반환합니다.

테이블의 각 행에 대해 계산된 식의 백분위수 수를 반환하려면 PERCENTILEX를 사용합니다. [EXC 함수](#)입니다.

구문

DAX

`PERCENTILE.EXC(<column>, <k>)`

매개 변수

 테이블 확장

용어	정의
column	상대 순위를 정의하는 값이 들어 있는 열입니다.
k	0..1 범위에서 제외된 백분위수 값입니다.

반환 값

범위 내 값의 k번째 백분위수입니다. 여기서 k는 0..1 범위에서 제외됩니다.

설명

- 열이 비어 있으면 BLANK()가 반환됩니다.
- k가 0이거나 비어 있으면 $1/(n+1)$ 의 백분위수 순위는 가장 작은 값을 반환합니다. 0이면 범위를 벗어나고 오류가 반환됩니다.
- k가 숫자가 아니거나 0에서 1 범위 밖에 있으면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 백분위수입니다. EXC는 보간하여 k번째 백분위수의 값을 확인합니다.

- 백분위. EXC는 지정된 백분위수 값이 배열의 두 값 사이에 있을 때 보간됩니다. 지정된 k 백분위수에 대해 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

[PERCENTILEX. Exc](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

백분위. Inc

아티클 • 2023. 12. 14.

범위에서 값의 k번째 백분위수(k는 0..1 범위 포함)를 반환합니다.

테이블의 각 행에 대해 계산된 식의 백분위수 수를 반환하려면 PERCENTILEX를 사용합니다. INC.

구문

DAX

PERCENTILE.INC(<column>, <k>)

매개 변수

 테이블 확장

용어	정의
column	상대 순위를 정의하는 값이 들어 있는 열입니다.
k	0..1 범위의 백분위수 값(포함)입니다.

반환 값

범위 내 값의 k번째 백분위수입니다. 여기서 k는 0..1 범위(포함)입니다.

설명

- 열이 비어 있으면 BLANK()가 반환됩니다.
- k가 0이거나 비어 있으면 $1/(n+1)$ 의 백분위수 순위는 가장 작은 값을 반환합니다. 0 이면 범위를 벗어나고 오류가 반환됩니다.
- k가 숫자가 아니거나 0에서 1 범위 밖에 있으면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 백분위수입니다. INC는 보간하여 k번째 백분위수 값을 결정합니다.

- 백분위. INC는 지정된 백분위수 값이 배열의 두 값 사이에 있을 때 보간됩니다. 지정된 k 백분위수에 대해 보간할 수 없으면 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

[PERCENTILEX. Inc](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PERCENTILEX. EXC

아티클 • 2023. 10. 20.

테이블의 각 행에 대해 계산되는 식의 백분위수 수를 반환합니다.

열에 있는 숫자의 백분위수를 반환하려면 PERCENTILE을 사용합니다 . [EXC 함수](#)입니다.

구문

DAX

`PERCENTILEX.EXC(<table>, <expression>, k)`

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
k	0에서 1까지만 사용할 수 있는 범위의 원하는 백분위수 값입니다.

반환 값

테이블의 각 행에 대해 계산되는 식의 백분위수입니다.

설명

- k가 0이거나 비어 있으면 $1/(n+1)$ 의 백분위수 순위는 가장 작은 값을 반환합니다. 0 이면 범위를 벗어나고 오류가 반환됩니다.
- k가 숫자가 아니거나 0에서 1 범위 밖에 있으면 오류가 반환됩니다.
- k가 $1/(n + 1)$ 의 배수가 아니면 PERCENTILEX입니다. EXC는 보간하여 k번째 백분위 수의 값을 확인합니다.
- PERCENTILEX. EXC는 지정된 백분위수 값이 배열의 두 값 사이에 있을 때 보간됩니다. 지정된 k 백분위수에 대해 보간할 수 없으면 오류가 반환됩니다.

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

참고 항목

[백분위. Exc](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PERCENTILEX. Inc

아티클 • 2023. 10. 20.

테이블의 각 행에 대해 계산되는 식의 백분위수 수를 반환합니다.

열에 있는 숫자의 백분위수를 반환하려면 PERCENTILE을 사용합니다. INC.

구문

DAX

PERCENTILEX.INC(<table>, <expression>;, k)

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
k	범위 0에서 1까지의 원하는 백분위수 값입니다.

반환 값

테이블의 각 행에 대해 계산되는 식의 백분위수입니다.

설명

- k가 0이거나 비어 있으면 $1/(n - 1)$ 의 백분위수 순위는 가장 작은 값을 반환합니다. 0 이면 범위를 벗어나고 오류가 반환됩니다.
- k가 숫자가 아니거나 0에서 1 범위 밖에 있으면 오류가 반환됩니다.
- k가 $1/(n - 1)$ 의 배수가 아니면 PERCENTILEX입니다. EXC는 보간하여 k번째 백분위수의 값을 확인합니다.
- PERCENTILEX. INC는 지정된 백분위수 값이 배열의 두 값 사이에 있을 때 보간됩니다. 지정된 k 백분위수에 대해 보간할 수 없으면 오류가 반환됩니다.

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

참고 항목

[백분위. Inc](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PERMUT

아티클 • 2023. 10. 20.

숫자 개체에서 선택할 수 있는 지정된 수의 개체에 대한 순열 수를 반환합니다. 순열은 내부 순서가 중요한 개체 또는 이벤트의 집합 또는 하위 집합입니다. 순열은 내부 순서가 중요하지 않은 조합과 다릅니다. 복권 스타일 확률 계산에 이 함수를 사용합니다.

구문

DAX

PERMUT(number, number_chosen)

매개 변수

용어	정의
번호	필수입니다. 개체 수를 설명하는 정수입니다.
number_chosen	필수입니다. 각 순열의 개체 수를 설명하는 정수입니다.

반환 값

숫자 개체에서 선택할 수 있는 지정된 수의 개체에 대한 순열 수를 반환합니다.

설명

- 두 인수는 모두 정수로 잘립니다.
- number 또는 number_chosen 숫자가 아니면 PERMUT에서는 #VALUE! 오류 값입니다.
- number ≤ 0이거나 number_chosen < 0이면 PERMUT에서 #NUM 반환합니다. 오류 값입니다.
- number < 가 number_chosen 경우 PERMUT는 #NUM 반환합니다. 오류 값입니다.
- 순열 수에 대한 수식은 다음과 같습니다.

$$P_{k,n} = \frac{n!}{(n - k)!}$$

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

예시

다음 수식에서 2가 선택된 3개의 개체 그룹에 대해 순열이 가능합니다.

DAX

```
= PERMUT(3,2)
```

결과

6

피드백

이 페이지가 도움이 되었나요?

Yes

No

포아송. Dist

아티클 • 2023. 10. 20.

포아송 분포를 반환합니다. 포아송 분포의 일반적인 적용은 1분 만에 유료 광장에 도착하는 자동차 수와 같은 특정 시간에 대한 이벤트 수를 예측하는 것입니다.

구문

DAX

POISSON.DIST(x,mean,cumulative)

매개 변수

용어	정의
x	필수입니다. 이벤트 수입니다.
평균	필수입니다. 예상된 숫자 값입니다.
cumulative	필수입니다. 반환된 확률 분포의 형태를 결정하는 논리값입니다. 누적이 TRUE이면 POISSON입니다. DIST는 발생하는 임의의 이벤트 수가 0에서 x 포함 사이일 수 있는 누적 포아송 확률을 반환합니다. 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 아니면 POISSON입니다. DIST는 #VALUE 반환합니다! 오류 값입니다.
- x < 0이면 POISSON입니다. DIST는 #NUM 반환합니다! 오류 값입니다.
- 평균 < 0인 경우 POISSON입니다. DIST는 #NUM 반환합니다! 오류 값입니다.
- 포아송. DIST는 다음과 같이 계산됩니다.

- 누적 = FALSE의 경우:

$$\text{POISSON} = \frac{e^{-\lambda} \lambda^x}{x!}$$

- 누적 = TRUE의 경우:

$$\text{CUMPOISSON} = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^x}{k!}$$

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

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RANK.EQ

아티클 • 2023. 10. 20.

숫자 목록에서 숫자의 순위를 반환합니다.

구문

DAX

```
RANK.EQ(<value>, <columnName>[, <order>])
```

매개 변수

용어	정의
value	순위를 찾을 단일 스칼라 값을 반환하는 DAX 식입니다. 식은 함수가 평가되기 전에 정확히 한 번 평가되어야 하며 인수 목록에 값이 전달됩니다.
columnName	순위를 결정할 기준 열의 이름입니다. ADDCOLUMNS, ROW 또는 SUMMARIZE 함수를 사용하여 만든 식이나 열일 수 없습니다.
order	(선택 사항) 숫자를 낮음에서 높음 또는 높음에서 낮음으로 순위를 지정하는 방법을 지정하는 값입니다.

order 값

value	대체 값	설명
0(영)	FALSE	columnName의 내림차순 순위입니다. 값이 columnName에서 가장 높은 숫자와 같으면 RANK입니다. EQ는 1입니다.
1	TRUE	columnName의 오름차순 순위입니다. 값이 columnName에서 가장 낮은 숫자와 같으면 RANK입니다. EQ는 1입니다.

반환 값

columnName의 숫자 중 값의 순위를 나타내는 숫자입니다.

설명

- *columnName* 은 ADDCOLUMNS, ROW 또는 SUMMARIZE 함수를 사용하여 만든 열을 참조할 수 없습니다. 나
- 값이 *columnName*에 없거나 값이 비어 있으면 RANK입니다. EQ는 빈 값을 반환합니다.
- 값의 중복 값은 동일한 순위 값을 받습니다. 할당된 다음 순위 값은 순위 값과 중복 값의 수입니다. 예를 들어 5(5) 값이 순위가 11인 경우 다음 값은 16($11 + 5$)의 순위를 받습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 예제에서는 동일한 열의 모든 숫자에 대해 InternetSales_USD 테이블에서 SalesAmount_USD 값의 순위를 지정하는 계산 열을 만듭니다.

DAX

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD],  
InternetSales_USD[SalesAmount_USD])
```

예제 2

다음 예제에서는 지정된 샘플에 대해 값의 하위 집합의 순위를 지정합니다. 특정 국가별 시험에서 성적을 가진 지역 학생 테이블이 있고 해당 국가별 시험에서 전체 점수 집합을 가지고 있다고 가정합니다. 다음 계산 열은 각 지역 학생에 대한 국가 순위를 제공합니다.

DAX

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

RANKX

아티클 • 2023. 10. 20.

테이블 인수의 각 행에 대한 숫자 목록의 숫자 순위를 반환합니다.

구문

DAX

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

매개 변수

table

식이 계산되는 데이터 테이블을 반환하는 DAX 식입니다.

expression

단일 스칼라 값을 반환하는 DAX 식입니다. 식은 순위에 대해 가능한 모든 값을 생성하기 위해 테이블의 각 행에 대해 계산됩니다. 식이 BLANK로 계산되는 경우 함수 동작을 이해하려면 설명 섹션을 참조하세요.

value

(선택 사항) 순위를 찾을 단일 스칼라 값을 반환하는 DAX 식입니다. 식에서 값을 찾을 수 없는 경우 함수의 동작을 이해하려면 설명 섹션을 참조하세요.

값 매개 변수를 생략하면 현재 행의 식 값이 대신 사용됩니다.

order

(선택 사항) 값의 순위를 낮음에서 높음 또는 높음으로 지정하는 방법을 지정하는 값입니다.

value	대체 값	설명
0(영)	FALSE	식 값의 내림차순 순위입니다. 값이 식에서 가장 높은 숫자와 같으면 RANKX는 1을 반환합니다. 순서 매개 변수를 생략하면 기본값입니다.
1	TRUE	식의 오름차순으로 순위를 지정합니다. 값이 식에서 가장 낮은 숫자와 같으면 RANKX는 1을 반환합니다.

관계

(선택 사항) 관계가 있을 때 순위를 결정하는 방법을 정의하는 열거형입니다.

열거 형	설명
Skip	다음 순위 값은 동률의 순위 값과 연결된 값의 수입니다. 예를 들어 5(5) 값이 순위가 11인 경우 다음 값은 16(11 + 5)의 순위를 받습니다. 이 같은 ties 매개 변수를 생략할 때 기본값입니다.
밀도	다음 순위 값은 동률 이후의 다음 순위 값입니다. 예를 들어 5(5) 값이 11의 순위와 연결된 경우 다음 값은 12의 순위를 받습니다.

반환 값

테이블 번호의 모든 행에 대해 계산된 식의 가능한 모든 값 중에서 값의 순위 번호입니다.

설명

- 식 또는 값이 BLANK로 계산되면 숫자를 생성하는 모든 식에 대해 0으로 처리되거나 모든 텍스트 식에 대해 빈 텍스트로 처리됩니다.
- 값이 식의 가능한 값 중 일부가 아닌 경우 RANKX는 식의 값에 값을 일시적으로 추가하고 RANKX를 다시 평가하여 적절한 값 순위를 결정합니다.
- 인수 목록에 빈 쉼표(,)를 배치하여 선택적 인수를 건너뛸 수 있습니다(예: RANKX(Inventory, [InventoryCost],,, "Dense")
- 10진수 데이터 형식이 있는 값 간의 같음 관련 비교($=$, $<$, $>$, \geq 및 \leq) 시 RANKX 함수를 사용하면 예기치 않은 결과가 반환될 수 있습니다. 10진수 데이터 형식이 있는 값은 IEEE 표준 754 부동 소수점으로 저장되며, 본질적으로 정확도에 한계가 있기 때문에 잘못된 결과가 발생할 수 있습니다. 예기치 못한 결과를 방지하려면 데이터 형식을 고정 10진수로 변경하거나 ROUND 함수를 사용해 강제 반올림하세요.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

Products 테이블의 다음 계산 열은 인터넷 채널의 각 제품에 대한 판매 순위를 계산합니다.

DAX

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SAMPLE

아티클 • 2023. 10. 20.

지정한 테이블에서 N행의 샘플을 반환합니다.

구문

DAX

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[,<orderBy_expression>, [<order>]]...])
```

매개 변수

용어	정의
n_value	반환할 행 수입니다. 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다. 정수가 아닌 값(또는 식)을 입력하면 결과가 정수로 캐스팅됩니다.
table	'n' 샘플 행을 추출할 위치에서 데이터 테이블을 반환하는 DAX 식입니다.
orderBy_expression	테이블의 각 행에 대해 결과 값이 계산되는 스칼라 DAX 식입니다.
order	(선택 사항) orderBy_expression 값(오름차순 또는 내림차순: 0)을 정렬하는 방법을 지정하는 값으로, order_by 값의 내림차순으로 정렬합니다. 1, order_by 오름차순으로 순위가 지정됩니다.

반환 값

n_value 0 이하인 경우 테이블의 N행 샘플 또는 빈 테이블로 구성된 테이블입니다. OrderBy 인수가 제공되면 샘플은 안정적이고 결정적이며 첫 번째 행, 마지막 행 및 균등하게 분산된 행을 반환합니다. 순서 지정이 지정되지 않은 경우 샘플은 임의이고 안정적이지 않으며 결정적이지 않습니다.

설명

- n_value 0 이하이면 SAMPLE은 빈 테이블을 반환합니다.
- 샘플에서 중복 값을 방지하려면 두 번째 인수로 제공된 테이블을 정렬에 사용되는 열로 그룹화해야 합니다.

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

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

STDEV.S

아티클 • 2023. 10. 20.

샘플 모집단의 표준 편차를 반환합니다.

구문

DAX

`STDEV.S(<ColumnName>)`

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다.

반환 값

샘플 모집단의 표준 편차를 나타내는 숫자입니다.

예외

설명

- Stdev. S는 열이 모집단의 샘플을 참조한다고 가정합니다. 데이터가 전체 모집단을 나타내는 경우 STDEV.P를 사용하여 표준 편차를 계산합니다.
- Stdev. S는 다음 수식을 사용합니다.

$$\sqrt{\frac{\sum(x - \bar{x})^2}{n-1}}$$

여기서 x 은 표본 모집단의 평균 \bar{x} 값이고 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.

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

예시

다음 예제에서는 테이블 InternetSales_USD 샘플 채우기인 경우 열의 표준 편차를 계산하는 측정값의 수식(SalesAmount_USD)을 보여 줍니다.

DAX

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

STDEV.P

아티클 • 2023. 10. 20.

전체 모집단의 표준 편차를 반환합니다.

구문

DAX

`STDEV.P(<ColumnName>)`

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다.

반환 값

전체 모집단의 표준 편차를 나타내는 숫자입니다.

설명

- Stdev. P에서는 열이 전체 모집단을 참조한다고 가정합니다. 데이터가 모집단의 샘플을 나타내는 경우 STDEV.S를 사용하여 표준 편차를 계산합니다.
- Stdev. P는 다음 수식을 사용합니다.

$$\sqrt{\frac{\sum(x - \bar{x})^2}{n}}$$

여기서 x 은 전체 모집단의 평균 \bar{x} 값이고 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 테이블 InternetSales_USD 전체 모집단일 때 열 SalesAmount_USD 표준 편차를 계산하는 측정값의 수식을 보여 줍니다.

DAX

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

STDEVX. S

아티클 • 2023. 10. 20.

샘플 모집단의 표준 편차를 반환합니다.

구문

DAX

`STDEVX.S(<table>, <expression>)`

매개 변수

용어 정의

table 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

식 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

샘플 모집단의 표준 편차가 있는 숫자입니다.

예외

설명

- STDEVX. S는 테이블의 각 행에 대한 식을 평가하고 해당 테이블이 모집단의 샘플을 참조한다고 가정하여 식의 표준 편차를 반환합니다. 테이블이 전체 모집단을 나타내는 경우 STDEVX.P를 사용하여 표준 편차를 계산합니다.
- STDEVX. S는 다음 수식을 사용합니다.

$$\sqrt{\sum(x - \bar{x})^2 / (n-1)}$$

여기서 x 은 전체 모집단의 평균 \bar{x} 값이고 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.

- columnName에 빠져 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 Product 테이블에서 수식을 사용할 때 샘플 모집단에 대한 제품당 단가의 표준 편차를 예측하는 계산 열의 수식을 보여 줍니다.

DAX

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

STDEVX. P

아티클 • 2023. 10. 20.

전체 모집단의 표준 편차를 반환합니다.

구문

DAX

`STDEVX.P(<table>, <expression>)`

매개 변수

용어 정의

table 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

식 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 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. 테이블의 데이터가 모집단의 샘플을 나타내는 경우 STDEVX를 사용하여 표준 편차를 계산해야 합니다. 대신 S.
- STDEVX. P는 다음 수식을 사용합니다.

$$\sqrt{\sum(x - \bar{x})^2/n}$$

여기서 x 은 전체 모집단의 평균 \bar{x} 값이고 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 Product 테이블에서 수식을 사용할 때 제품당 단가의 표준 편차를 계산하는 계산 열의 수식을 보여 줍니다.

DAX

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

T.DIST

아티클 • 2023. 12. 14.

Student의 왼쪽 단축 t 분포를 반환합니다.

구문

DAX

```
T.DIST(X,Deg_freedom,Cumulative)
```

매개 변수

[+] 테이블 확장

용어	정의
X	분포를 평가할 숫자 값입니다.
Deg_freedom	자유도의 수를 나타내는 정수입니다.
누적	함수의 형식을 결정하는 논리 값입니다. 누적이 TRUE이면 T.DIST는 누적 분포 함수를 반환합니다. FALSE이면 확률 밀도 함수를 반환합니다.

반환 값

학생의 왼쪽 꼬리 t 분포입니다.

설명

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

예시

DAX

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

반환

[테이블 확장]

[값]

0.994695326367377

관련 콘텐츠

[T.DIST.2T](#)

[T.DIST. Rt](#)

[T.INV](#)

[T.INV.2t](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

T.DIST.2T

아티클 • 2023. 12. 14.

두 꼬리 학생 t 분포를 반환합니다.

구문

DAX

```
T.DIST.2T(X,Deg_freedom)
```

매개 변수

 테이블 확장

용어	정의
X	분포를 평가할 숫자 값입니다.
Deg_freedom	자유도의 수를 나타내는 정수입니다.

반환 값

두 꼬리 학생의 t-분포입니다.

설명

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

예시

DAX

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

반환

[값]

0.054644929975921

관련 콘텐츠

[T.DIST](#)

[T.DIST. Rt](#)

[T.INV](#)

[T.INV.2t](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

T.DIST. Rt

아티클 • 2023. 12. 14.

오른쪽 꼬리 학생 t 분포를 반환합니다.

구문

DAX

```
T.DIST.RT(X,Deg_freedom)
```

매개 변수

 테이블 확장

용어	정의
X	분포를 평가할 숫자 값입니다.
Deg_freedom	자유도의 수를 나타내는 정수입니다.

반환 값

오른쪽 꼬리 학생의 t-분포입니다.

설명

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

예시

DAX

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

반환

[값]

0.0273224649879605

관련 콘텐츠

[T.DIST](#)

[T.DIST.2T](#)

[T.INV](#)

[T.INV.2t](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

T.INV

아티클 • 2023. 12. 14.

Student t-distribution의 왼쪽 역방향을 반환합니다.

구문

DAX

`T.INV(Probability, Deg_freedom)`

매개 변수

[+] 테이블 확장

용어	정의
확률	Student의 t 분포와 관련된 확률입니다.
Deg_freedom	분포의 특징을 지정할 자유도의 수입니다.

반환 값

학생 t 분포의 좌측 역방향입니다.

설명

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

예시

DAX

`EVALUATE { T.INV(0.75, 2) }`

반환

[값]

0.816496580927726

관련 콘텐츠

[T.INV.2T](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST. Rt](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

T.INV.2T

아티클 • 2023. 12. 14.

Student t-distribution의 2측 역방향을 반환합니다.

구문

DAX

```
T.INV.2T(Probability, Deg_freedom)
```

매개 변수

 테이블 확장

용어	정의
확률	Student의 t 분포와 관련된 확률입니다.
Deg_freedom	분포의 특징을 지정할 자유도의 수입니다.

반환 값

학생 t 분포의 두 꼬리 역입니다.

설명

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

예시

DAX

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

반환

[값]

0.606533075825759

관련 콘텐츠

[T.INV](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST. Rt](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Var. S

아티클 • 2023. 10. 20.

샘플 모집단의 분산을 반환합니다.

구문

DAX

VAR.S(<columnName>)

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다.

반환 값

샘플 모집단의 분산이 있는 숫자입니다.

설명

- Var. S는 열이 모집단의 샘플을 참조한다고 가정합니다. 데이터가 전체 모집단을 나타내는 경우 VAR.P를 사용하여 분산을 계산합니다.
- Var. S는 다음 수식을 사용합니다.

$$\sum(x - \bar{x})^2 / (n-1)$$

여기서 x 는 샘플 모집단의 평균 \bar{x} 값입니다.

및 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 샘플 채우기에 대한 InternetSales_USD SalesAmount_USD 열의 분산을 계산하는 측정값의 수식을 보여 줍니다.

DAX

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

VAR.P

아티클 • 2023. 10. 20.

전체 모집단의 분산을 반환합니다.

구문

DAX

VAR.P(<columnName>)

매개 변수

용어	정의
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다.

반환 값

전체 모집단의 분산이 있는 숫자입니다.

설명

- Var. P에서는 열이 전체 모집단을 참조한다고 가정합니다. 데이터가 모집단의 샘플을 나타내는 경우 VAR.S를 사용하여 분산을 계산합니다.
- Var. P는 다음 수식을 사용합니다.

$$\sum(x - \bar{x})^2/n$$

여기서 x 는 전체 모집단의 평균 \bar{x} 값입니다.

및 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 빈 행이 2개 미만인 경우 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 전체 모집단에 대해 InternetSales_USD 테이블에서 SalesAmount_USD 열의 분산을 예측하는 측정값의 수식을 보여 줍니다.

DAX

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

VARX. S

아티클 • 2023. 10. 20.

샘플 모집단의 분산을 반환합니다.

구문

DAX

VARX.S(<table>, <expression>)

매개 변수

용어 정의

table 데이터 테이블을 반환하는 DAX 식입니다.

식 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

샘플 모집단의 분산을 나타내는 숫자입니다.

설명

- VARX. S는 테이블의 각 행에 대한 식을 평가하고 식의 분산을 반환합니다. 즉, 테이블이 모집단의 샘플을 참조한다는 가정하에 반환합니다. 테이블이 전체 모집단을 나타내는 경우 VARX.P를 사용하여 분산을 계산해야 합니다.
- Var. S는 다음 수식을 사용합니다.

$$\sum(x - \bar{x})^2/(n-1)$$

여기서 x 는 샘플 모집단의 평균 \bar{x} 값입니다.

및 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.

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

예시

다음 예제에서는 Product 테이블에서 수식을 사용할 때 샘플 모집단의 제품당 단가 차이를 예측하는 계산 열의 수식을 보여 줍니다.

DAX

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

VARX.P

아티클 • 2023. 10. 20.

전체 모집단의 분산을 반환합니다.

구문

DAX

VARX.P(<table>, <expression>)

매개 변수

용어 정의

table 데이터 테이블을 반환하는 DAX 식입니다.

식 단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

전체 모집단의 분산이 있는 숫자입니다.

설명

- VARX.P는 <table>의 각 행에 대해 <expression>을 계산하고 <table>이 전체 모집단을 참조하는 것으로 간주하여 <expression>의 분산을 반환합니다. 테이블>이 모집단의 샘플을 나타내는 경우 <VARX.S>를 사용하여 분산을 계산합니다.
- VARX.P는 다음 수식을 사용합니다.

$$\sum(x - \bar{x})^2/n$$

여기서 x 는 전체 모집단의 평균 \bar{x} 값입니다.

및 n 은 모집단 크기입니다.

- 빈 행은 columnName에서 필터링되며 계산에서 고려되지 않습니다.
- columnName에 비어 있지 않은 행이 2개 미만인 경우 오류가 반환됩니다.

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

예시

다음 예제에서는 Product 테이블에서 수식을 사용할 때 제품당 단가의 분산을 계산하는 계산 열의 수식을 보여 줍니다.

DAX

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

테이블 조작 함수

아티클 • 2023. 10. 20.

이러한 함수는 테이블을 반환하거나 기존 테이블을 조작합니다.

이 범주에서

함수	설명
ADDCOLUMNS	지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.
ADDMISSINGITEMS	여러 열의 항목 조합이 아직 없는 경우 테이블에 추가합니다.
CROSSJOIN	인수에 있는 모든 테이블의 모든 행에 대한 Cartesian 곱이 들어 있는 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 평가하고 데이터를 반환합니다.
DISTINCT 열	지정된 열의 고유 값이 들어 있는 1열 테이블을 반환합니다.
DISTINCT 테이블	다른 테이블 또는 식에서 중복 행을 제거하여 테이블을 반환합니다.
EXCEPT	다른 테이블에 나타나지 않는 한 테이블의 행을 반환합니다.
필터	columnName에 필터로 직접 적용된 값 테이블을 반환합니다.
생성	table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블을 반환합니다.
GENERATEALL	table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블을 반환합니다.
GENERATESERIES	산술 계열의 값을 포함하는 단일 열 테이블을 반환합니다.
Groupby	SUMMARIZE 함수와 마찬가지로 GROUPBY는 추가하는 확장 열에 대해 암시적 CALCULATE를 수행하지 않습니다.
무시	BLANK/NULL 계산에서 특정 식을 생략하여 SUMMARIZECOLUMNS를 수정합니다.
INTERSECT	중복을 유지하면서 두 테이블의 행 교집합을 반환합니다.
NATURALINNERJOIN	다른 테이블을 사용하여 테이블의 내부 조인을 수행합니다.

함수	설명
NATURALLEFTOUTERJOIN	RightTable을 사용하여 LeftTable의 조인을 수행합니다.
롤업	groupBy_columnName 매개 변수로 정의된 열의 결과에 롤업 행을 추가하여 SUMMARIZE의 동작을 수정합니다.
ROLLUPADDISSTOTAL	groupBy_columnName 열을 기반으로 롤업/부분합 행을 결과에 추가하여 SUMMARIZECOLUMNNS의 동작을 수정합니다.
ROLLUPISSSTOTAL	롤업 그룹을 ADDMISSINGITEMS 식 내에서 ROLLUPADDISSTOTAL에 의해 추가된 열과 쌍을 이룹니다.
ROLLUPGROUP	groupBy_columnName 매개 변수로 정의된 열의 결과에 롤업 행을 추가하여 SUMMARIZE 및 SUMMARIZECOLUMNNS의 동작을 수정합니다.
ROW	각 열에 지정된 식에서 생성된 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.
요약	그룹 집합에 대해 요청된 합계에 대한 요약 테이블을 반환합니다.
SUMMARIZECOLUMNNS	그룹 집합에 대한 요약 테이블을 반환합니다.
테이블 생성자	하나 이상의 열 테이블을 반환합니다.
TOPN	지정된 테이블의 상위 N개 행을 반환합니다.
TREATAS	테이블 식의 결과를 관련 없는 테이블의 열에 필터로 적용합니다.
UNION	테이블 쌍에서 공용 구조체(조인) 테이블을 만듭니다.
VALUES	지정한 테이블 또는 열의 고유 값을 포함하는 하나의 열 테이블을 반환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ADDCOLUMNS

아티클 • 2023. 10. 20.

지정된 테이블 또는 테이블 식에 계산 열을 추가합니다.

구문

DAX

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

매개 변수

용어	정의
table	데이터 테이블을 반환하는 DAX 식입니다.
name	큰따옴표로 묶인 열에 지정된 이름입니다.
식	테이블의 각 행에 대해 계산되는 스칼라 식을 반환하는 DAX 식입니다.

반환 값

모든 원래 열과 추가된 열이 있는 테이블입니다.

설명

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

예시

다음 예제에서는 재판매인 채널 및 인터넷 판매의 총 판매액 값을 포함하는 확장 버전의 Product Category 테이블을 반환합니다.

DAX

```
ADDCOLUMNS(ProductCategory,
            , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
            , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

다음 표에서는 테이블을 수신할 것으로 예상되는 함수가 수신하는 데이터의 미리 보기입니다.

ProductCategory[ProductName]	ProductCategory[ProductCategoryAlternateKey]	ProductCategory[ProductCategoryKey]	[인터넷 판매]	[재판 판매]
Bikes	1	1	25107749.77	6308
구성 요소	2	2		1120
의류	3	3	306157.5829	1669
Accessories	4	4	640920.1338	5343

피드백

이 페이지가 도움이 되었나요?

 Yes  No

ADDMISSINGITEMS

아티클 • 2023. 10. 20.

SUMMARIZECOLUMNS에서 반환된 테이블에 빈 값이 있는 행을 추가합니다.

구문

DAX

```
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.
table	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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

SUMMARIZECOLUMNS를 사용하여

SUMMARIZECOLUMNS에서 반환된 테이블에는 값이 있는 행만 포함됩니다.

ADDMISSINGITEMS 식 내에서 SUMMARIZECOLUMNS 식을 래핑하면 값이 없는 행도 반환됩니다.

예시

ADDMISSINGITEMS가 없으면 다음 쿼리가 수행됩니다.

DAX

```
SUMMARIZECOLUMNS(  
    'Sales'[CustomerId],  
    "Total Qty", SUM ( Sales[TotalQty] )  
)
```

반환

고객 ID	TotalQty
A	5
b	3
C	3
E	2

ADDMISSINGITEMS를 사용하여 다음 쿼리를 수행합니다.

DAX

```
EVALUATE  
ADDMISSINGITEMS (  
    'Sales'[CustomerId],  
    SUMMARIZECOLUMNS(  
        'Sales'[CustomerId],  
        "Total Qty", SUM ( Sales[TotalQty] )  
,  
        'Sales'[CustomerId]  
)
```

반환

고객 ID	TotalQty
A	5
b	3
C	3
D	

고객 ID	TotalQty
E	2
F	

피드백

이 페이지가 도움이 되었나요?

Yes

No

CROSSJOIN

아티클 • 2023. 10. 20.

인수에 있는 모든 테이블의 모든 행에 대한 Cartesian 곱이 들어 있는 테이블을 반환합니다. 새 테이블의 열은 모든 인수 테이블의 모든 열입니다.

구문

DAX

`CROSSJOIN(<table>, <table>[, <table>]...)`

매개 변수

용어	정의
table	데이터 테이블을 반환하는 DAX 식

반환 값

인수에 있는 모든 테이블의 모든 행에 대한 Cartesian 곱이 들어 있는 테이블입니다.

설명

- 테이블 인수의 열 이름은 모두 모든 테이블에서 달라야 하며 오류가 반환됩니다.
- CROSSJOIN()이 반환하는 총 행 수는 인수의 모든 테이블에서 행 수의 곱과 같습니다. 또한 결과 테이블의 총 열 수는 모든 테이블의 열 수에 대한 합계입니다. 예를 들어 TableA에 rA 행과 cA 열이 있고 TableB에 rB 행과 cB 열이 있고 TableC에 rC 행과 cC 열이 있는 경우 결과 테이블에 $rA \times rB \times rC$ 행과 $cA + cB + cC$ 열이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 색과 편지지라는 두 테이블에 CROSSJOIN()을 적용한 결과를 보여 줍니다.

색 표에는 색과 패턴이 포함되어 있습니다.

색상	패턴
빨간색	가로 줄무늬
녹색	세로 줄무늬
파랑	크로스해칭

표 편지지에는 글꼴과 프레젠테이션이 포함되어 있습니다.

글꼴	프레젠테이션
리프	양각
sans-serif	새겨진

교차 조인을 생성하는 식은 다음과 같습니다.

DAX
CROSSJOIN(Colors, Stationery)

테이블 식이 필요한 곳마다 위의 식을 사용하는 경우 식의 결과는 다음과 같습니다.

색상	패턴	글꼴	프레젠테이션
빨간색	가로 줄무늬	리프	양각
녹색	세로 줄무늬	리프	양각
파랑	크로스해칭	리프	양각
빨간색	가로 줄무늬	sans-serif	새겨진
녹색	세로 줄무늬	sans-serif	새겨진
파랑	크로스해칭	sans-serif	새겨진

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CURRENTGROUP

아티클 • 2023. 10. 20.

GROUPBY 결과의 현재 행에 속하는 GROUPBY 식의 테이블 인수에서 행 집합을 반환합니다.

구문

DAX

CURRENTGROUP ()

매개 변수

없음

반환 값

groupBy_columnName 인수의 값 그룹에 해당하는 GROUPBY 함수의 테이블 인수에 있는 행입니다.

설명

- 이 함수는 GROUPBY 식 내에서만 사용할 수 있습니다.
- 이 함수는 인수를 사용하지 않으면 AVERAGEX, COUNTAX, COUNTX, GEOMEANX, MAXX, MINX, PRODUCTX, STDEVX 중 하나에 대한 첫 번째 인수로만 지원됩니다. S, STDEVX. P, SUMX, VARX. S, VARX. P.

예시

GROUPBY를 참조하세요.

피드백

이 페이지가 도움이 되었나요?

Yes

No

Datatable

아티클 • 2023. 10. 20.

인라인 데이터 값 집합을 선언하는 메커니즘을 제공합니다.

구문

DAX

```
DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}}...)
```

매개 변수

용어	정의
ColumnName	테이블을 반환하는 DAX 식입니다.
DataType	INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME을 포함하는 열거형
값	배열 배열을 제공하기 위해 중첩된 1차원 배열 상수에 Excel 구문을 사용하는 단일 인수입니다. 이 인수는 테이블에 있을 데이터 값 집합을 나타냅니다. 예를 들면 다음과 같습니다. { {row1}, {row2}의 값, {row3}의 값 등 } 여기서 {row1}은 쉼표로 구분된 상수 식 집합입니다. 즉, 상수의 조합으로 DATE, TIME 및 BLANK를 비롯한 몇 가지 기본 함수와 함께 DATE와 TIME 및 단항 빼기 연산자 사이의 더하기 연산자를 결합하여 음수 값을 표현할 수 있습니다. 유효한 값은 모두 3, -5, BLANK(), "2009-04-15 02:45:21"입니다. 값은 직접 실행 식 외부의 항목을 참조할 수 없으며 열, 테이블, 관계 또는 기타 항목을 참조할 수 없습니다. 누락된 값은 BLANK()와 동일하게 처리됩니다. 예를 들어 {1,2,BLANK(),4}와 {1,2,,4}는 같습니다.

반환 값

인라인 값 집합을 선언하는 테이블입니다.

설명

- DATATABLE 과 달리 테이블 생성자는 모든 스칼라 식을 입력 값으로 허용합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

DAX

```
= DataTable("Name", STRING,
            "Region", STRING
            ,{
                {"User1", "East"},  

                {"User2", "East"},  

                {"User3", "West"},  

                {"User4", "West"},  

                {"User4", "East"}  

            }  

        )
```

피드백

이 페이지가 도움이 되었나요?

Yes

No

DETAILROWS

아티클 • 2023. 10. 20.

측정값에 대해 정의된 세부 정보 행 식을 평가하고 데이터를 반환합니다.

구문

DAX

DETAILROWS([Measure])

매개 변수

용어	정의
측정값	측정값의 이름입니다.

반환 값

세부 정보 행 식에서 반환된 데이터가 있는 테이블입니다. 세부 정보 행 식이 정의되지 않은 경우 측정값이 포함된 테이블의 데이터가 반환됩니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

DISTINCT(열)

아티클 • 2023. 10. 20.

지정된 열의 고유 값이 들어 있는 1열 테이블을 반환합니다. 즉, 중복 값이 제거되고 고유한 값만 반환됩니다.

① 참고

이 함수는 워크시트의 셀이나 열에 값을 반환하는 데 사용할 수 없습니다. 대신 DISTINCT 함수를 수식 내에 중첩하여 다른 함수에 전달한 다음 다른 작업에 대해 계산, 합계 또는 사용할 수 있는 고유 값 목록을 가져옵니다.

구문

DAX

DISTINCT(<column>)

매개 변수

용어	정의
column	고유 값을 반환할 열입니다. 또는 열을 반환하는 식입니다.

반환 값

고유한 값의 열입니다.

설명

- DISTINCT의 결과는 현재 필터 컨텍스트의 영향을 받습니다. 예를 들어 다음 예제의 수식을 사용하여 측정값을 만드는 경우 특정 지역 또는 기간만 표시하도록 테이블을 필터링할 때마다 결과가 변경됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련된 함수

다른 테이블 또는 식에서 중복 행을 제거하여 테이블을 반환하는 DISTINCT 함수 DISTINCT(테이블)의 또 다른 버전이 있습니다.

VALUES 함수는 DISTINCT와 비슷합니다. 고유 값 목록을 반환하는 데 사용할 수도 있으며 일반적으로 DISTINCT와 정확히 동일한 결과를 반환합니다. 그러나 일부 컨텍스트에서 VALUES는 하나의 추가 특수 값을 반환합니다. 자세한 내용은 VALUES 함수를 참조 [하세요](#).

예시

다음 수식은 인터넷 채널을 통해 주문을 생성한 고유 고객의 수를 계산합니다. 다음 표에서는 수식이 보고서에 추가되면 가능한 결과를 보여 줍니다.

DAX

```
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

DISTINCT에서 반환하는 값 목록을 열에 직접 붙여 넣을 수 없습니다. 대신 목록을 사용하여 값을 계산, 필터링 또는 집계하는 다른 함수에 DISTINCT 함수의 결과를 전달합니다. 예제를 가능한 한 간단하게 만들기 위해 여기에서 고유 값의 테이블이 COUNTROWS 함수에 전달되었습니다.

Row Labels	Accessories	Bikes	Clothing	총합계
2005		1013		1013
2006		2677		2677
2007	6792	4875	2867	9309
2008	9435	5451	4196	11377
총합계	15114	9132	6852	18484

또한 결과는 가산적이지 않습니다. 즉, 2007년의 총 고유 고객 수는 그 해의 액세서리, 자전거 및 의류의 고유 고객의 합계가 아닙니다. 그 이유는 고객이 여러 그룹으로 계산될 수 있기 때문입니다.

참고 항목

필터 함수

FILTER 함수

RELATED 함수

VALUES 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DISTINCT(테이블)

아티클 • 2023. 12. 14.

다른 테이블 또는 식에서 중복 행을 제거하여 테이블을 반환합니다.

구문

DAX

```
DISTINCT(<table>)
```

매개 변수

 테이블 확장

용어 정의

table 고유한 행을 반환할 테이블입니다. 테이블은 테이블을 생성하는 식일 수도 있습니다.

반환 값

고유 행만 포함하는 테이블입니다.

관련된 함수

열 이름을 입력 매개 변수로 사용하는 DISTINCT 함수 [DISTINCT\(열\)](#)의 또 다른 버전이 있습니다.

예시

다음 쿼리:

DAX

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

테이블을 반환합니다.

[Value1]	[Value2]
1	A
2	B

관련 콘텐츠

필터 함수

DISTINCT(열)

FILTER 함수

RELATED 함수

VALUES 함수

피드백

이 페이지가 도움이 되었나요?

Yes

No

EXCEPT

아티클 • 2023. 10. 20.

식에서 두 번째 테이블에 나타나지 않는 첫 번째 테이블의 행을 반환합니다.

구문

DAX

```
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 테이블입니다.

설명

- 두 테이블에 행이 모두 표시되면 결과 집합에 행과 해당 행의 중복 항목이 없습니다. 행이 table_expression1만 나타나면 결과 집합에 행과 해당 중복 항목이 표시됩니다.
- 열 이름은 table_expression1 열 이름과 일치합니다.
- 반환된 테이블에는 두 번째 테이블의 열 계보에 관계없이 table_expression1 열을 기반으로 계보가 있습니다. 예를 들어 첫 번째 table_expression 첫 번째 열에 모델의 기본 열 C1에 대한 계보가 있는 경우 Except은 두 번째 table_expression 첫 번째 열의 값 가용성에 따라 행을 줄이고 기본 열 C1의 계보를 그대로 유지합니다.
- 두 테이블의 열 수는 같아야 합니다.
- 열은 위치 지정에 따라 비교되고 데이터 비교는 형식 강제 변환 없이 비교됩니다.
- 반환되는 행 집합은 두 식의 순서에 따라 달라집니다.
- 반환된 테이블에는 table_expression1 관련된 테이블의 열이 포함되지 않습니다.

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

예시

상태1

주
A
B
B
B
C
D
D

상태2

주
b
C
D
D
D
E
E
E

Except(States1, States2)

주
A

Except(States2, States1)

주

E

E

E

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

필터

아티클 • 2023. 10. 20.

columnName에 필터로 직접 적용되는 값을 반환합니다.

구문

DAX

```
FILTERS(<columnName>)
```

매개 변수

용어	설명
columnName	표준 DAX 구문을 사용하는 기존 열의 이름입니다. 식일 수 없습니다.

반환 값

columnName에 필터로 직접 적용되는 값을입니다.

설명

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

예시

다음 예제에서는 열에 있는 직접 필터 수를 확인하는 방법을 보여 줍니다.

DAX

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

이 예제에서는 식이 평가되는 컨텍스트에 적용된 ResellerSales_USD[ProductKey]에 대한 직접 필터 수를 알 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

GENERATE

아티클 • 2023. 10. 20.

table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블을 반환합니다.

구문

DAX

```
GENERATE(<table1>, <table2>)
```

매개 변수

용어	정의
table1	테이블을 반환하는 DAX 식입니다.
table2	테이블을 반환하는 DAX 식입니다.

반환 값

table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블

설명

- table1의 현재 행에 대한 table2 평가에서 빈 테이블을 반환하는 경우 결과 테이블에 table1의 현재 행이 포함되지 않습니다. 이는 table1의 현재 행이 결과에 포함되고 table2에 해당하는 열에 해당 행에 대한 null 값이 있는 GENERATEALL()과 다릅니다.
- table1과 table2의 모든 열 이름은 달라야 하며 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 다음 표와 같이 Resellers 채널에 대한 지역 및 제품 범주별 판매 요약 테이블을 원합니다.

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductName]	[재판매인 판매]
유럽	Accessories	\$ 142,227.27
유럽	Bikes	\$ 9,970,200.44
유럽	의류	\$ 365,847.63
유럽	구성 요소	\$ 2,214,440.19
북아메리카	Accessories	\$ 379,305.15
북아메리카	Bikes	\$ 52,403,796.85
북아메리카	의류	\$ 1,281,193.26
북아메리카	구성 요소	\$ 8,882,848.05
태평양	Accessories	\$ 12,769.57
태평양	Bikes	\$ 710,677.75
태평양	의류	\$ 22,902.38
태평양	구성 요소	\$ 108,549.71

다음 수식은 위의 표를 생성합니다.

DAX

```
GENERATE(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
    ,SUMMARIZE(ProductCategory
    , [ProductName]
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
    ResellerSales_USD[SalesAmount_USD])
    )
)
```

1. 첫 번째 SUMMARIZE 문 `SUMMARIZE(SalesTerritory,`

`SalesTerritory[SalesTerritoryGroup])` 은 아래와 같이 각 행이 영역 그룹인 지역 그

룹 테이블을 생성합니다.

SalesTerritory[SalesTerritoryGroup]
북아메리카
유럽
태평양
해당 없음

2. 두 번째 SUMMARIZE 문 `SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))` 은 아래와 같이 각 그룹에 대한 재판매인 판매가 포함된 제품 범주 그룹의 테이블을 생성합니다.

ProductCategory[ProductCategoryName]	[재판매인 판매]
Bikes	\$ 63,084,675.04
구성 요소	\$ 11,205,837.96
의류	\$ 1,669,943.27
Accessories	\$ 534,301.99

3. 그러나 위의 테이블을 가져와서 지역 그룹 테이블의 각 행 컨텍스트에서 평가하면 각 지역에 대해 서로 다른 결과를 얻을 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

GENERATEALL

아티클 • 2023. 10. 20.

table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블을 반환합니다.

구문

DAX

`GENERATEALL(<table1>, <table2>)`

매개 변수

용어	정의
table1	테이블을 반환하는 DAX 식입니다.
table2	테이블을 반환하는 DAX 식입니다.

반환 값

table1의 각 행과 table1의 현재 행 컨텍스트에서 table2를 평가한 결과 테이블 사이에 카티전 제품이 있는 테이블

설명

- table1의 현재 행에 대한 table2 평가에서 빈 테이블이 반환되면 table1의 현재 행이 결과에 포함되고 table2에 해당하는 열에는 해당 행에 대한 null 값이 포함됩니다. 이는 table1의 현재 행이 결과에 포함되지 않는 GENERATE()와 다릅니다.
- table1과 table2의 모든 열 이름은 달라야 하며 오류가 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서 사용자는 다음 표와 같이 재판매인 채널의 지역 및 제품 범주별 판매 요약 테이블을 원합니다.

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductName]	[재판매인 판매]
유럽	Accessories	\$ 142,227.27
유럽	Bikes	\$ 9,970,200.44
유럽	의류	\$ 365,847.63
유럽	구성 요소	\$ 2,214,440.19
해당 없음	Accessories	
해당 없음	Bikes	
해당 없음	의류	
해당 없음	구성 요소	
북아메리카	Accessories	\$ 379,305.15
북아메리카	Bikes	\$ 52,403,796.85
북아메리카	의류	\$ 1,281,193.26
북아메리카	구성 요소	\$ 8,882,848.05
태평양	Accessories	\$ 12,769.57
태평양	Bikes	\$ 710,677.75
태평양	의류	\$ 22,902.38
태평양	구성 요소	\$ 108,549.71

다음 수식은 위의 표를 생성합니다.

DAX

```
GENERATEALL(
    SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
    ,SUMMARIZE(ProductCategory
    , [ProductName]
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
```

```
ResellerSales_USD[SalesAmount_USD])  
)  
)
```

- 첫 번째 SUMMARIZE는 아래 나열된 것과 같이 각 행이 지역 그룹인 지역 그룹 테이블을 생성합니다.

SalesTerritory[SalesTerritoryGroup]
북아메리카
유럽
태평양
해당 없음

- 두 번째 SUMMARIZE는 아래와 같이 각 그룹에 대한 재판매인 판매가 포함된 제품 범주 그룹의 테이블을 생성합니다.

ProductCategory[ProductCategoryName]	[재판매인 판매]
Bikes	\$ 63,084,675.04
구성 요소	\$ 11,205,837.96
의류	\$ 1,669,943.27
Accessories	\$ 534,301.99

- 그러나 위의 테이블을 가져와서 지역 그룹 테이블의 각 행 컨텍스트에서 테이블을 평가할 때 각 지역에 대해 서로 다른 결과를 얻습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

GENERATESERIES

아티클 • 2023. 10. 20.

산술 계열의 값이 들어 있는 단일 열 테이블을 반환합니다. 즉, 각 값이 상수 수량으로 앞의 값과 다른 값 시퀀스를 반환합니다. 반환된 열의 이름은 Value입니다.

구문

DAX

```
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 증분 값입니다. 제공되지 않으면 기본값은 1입니다.

반환 값

산술 계열의 값을 포함하는 단일 열 테이블입니다. 열의 이름은 Value입니다.

설명

- endValue가 startValue보다 작으면 빈 테이블이 반환됩니다.
- incrementValue는 양수 값이어야 합니다.
- 시퀀스는 endValue보다 작거나 같은 마지막 값에서 중지됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예 1

다음 DAX 쿼리:

DAX

```
EVALUATE GENERATESERIES(1, 5)
```

단일 열이 있는 다음 테이블을 반환합니다.

[값]

1

2

3

4

5

예제 2

다음 DAX 쿼리:

DAX

```
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)
```

단일 열이 있는 다음 테이블을 반환합니다.

[값]

1.2

1.6

2

2.4

예 3

다음 DAX 쿼리:

DAX

```
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

단일 열이 있는 다음 테이블을 반환합니다.

[값]
10
10.5
11
11.5
12

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

Groupby

아티클 • 2023. 10. 20.

GROUPBY 함수는 SUMMARIZE 함수와 [유사합니다](#). 그러나 GROUPBY는 추가하는 확장 열에 대해 암시적 [CALCULATE](#) 를 수행하지 않습니다. GROUPBY는 새 함수인 [CURRENTGROUP](#)을 추가하는 확장 열의 집계 함수 내에서 사용할 수 있도록 허용합니다. GROUPBY는 단일 테이블 검색에서 여러 집계를 수행하는 데 사용됩니다.

구문

DAX

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, ...]]] [, <name>, <expression> [, <name>, <expression> [, ...]]])
```

매개 변수

용어	정의
table	데이터 테이블을 반환하는 DAX 식입니다.
groupBy_columnName	데이터를 그룹화할 테이블(또는 관련 테이블)에 있는 기존 열의 이름입니다. 이 매개 변수는 식일 수 없습니다.
name	큰따옴표로 묶인 GroupBy 열 목록에 추가되는 새 열에 지정된 이름입니다.
식	첫 번째 인수가 CURRENTGROUP()인 X 집계 함수 중 하나입니다. 지원되는 X 집계 함수의 전체 목록은 아래 CURRENTGROUP 섹션을 참조하세요.

반환 값

groupBy_columnName 인수에 대해 선택한 열과 이름 인수로 지정된 확장 열을 가진 테이블입니다.

설명

- GROUPBY 함수는 다음을 수행합니다.
 1. 지정된 테이블(및 "일대일" 방향의 모든 관련 테이블)로 시작합니다.

2. 모든 GroupBy 열을 사용하여 그룹화 만듭니다(1단계의 테이블에 있어야 합니다.).

3. 각 그룹은 결과의 한 행이지만 원래 테이블의 행 집합을 나타냅니다.

4. 각 그룹에 대해 추가되는 확장 열을 평가합니다. SUMMARIZE 함수와 달리 암시적 CALCULATE는 수행되지 않으며 그룹이 필터 컨텍스트에 배치되지 않습니다.

- 이름을 정의하는 각 열에는 해당 식이 있어야 합니다. 그렇지 않으면 오류가 반환됩니다. 첫 번째 인수인 이름은 결과에서 열의 이름을 정의합니다. 두 번째 인수인 식은 해당 열의 각 행에 대한 값을 얻기 위해 수행되는 계산을 정의합니다.
- groupBy_columnName 테이블이나 관련 테이블에 있어야 합니다.
- 각 이름은 큰따옴표로 묶어야 합니다.
- 함수는 하나 이상의 groupBy_columnName 열 값으로 선택한 행 집합을 요약 행 집합으로 그룹화합니다. 각 그룹에 대해 하나의 행이 반환됩니다.
- GROUPBY는 주로 DAX 테이블 식의 중간 결과에 대한 집계를 수행하는데 사용됩니다. 모델의 실제 테이블에 대한 효율적인 집계를 위해 SUMMARIZECOLUMN 또는 SUMMARIZE 함수를 사용하는 것이 좋습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 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 테이블 인수에서 행 집합을 반환합니다. CURRENTGROUP 함수는 [인수를 사용하지 않으며 AVERAGEX, COUNTAX, COUNTX, GEOMEANX, MAXX, MINX, PRODUCTX, STDEVX](#) 중 하나에 대한 첫 번째 인수로만 지원됩니다. [S, STDEVX. P, SUMX, VARX, S, VARX. P.](#)

예시

다음 예제에서는 먼저 SUMMARIZECOLUMN 함수를 사용하여 [실제 테이블에 대해 국가 및 제품 범주별로 그룹화된 총 매출을 계산합니다](#). 그런 다음 GROUPBY 함수를 사용하여 첫 번째 단계의 중간 결과를 검사하여 제품 범주에서 각 국가의 최대 매출을 찾습니다.

DAX

```
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](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

IGNORE

아티클 • 2023. 10. 20.

BLANK/NULL 계산에서 특정 식을 생략하여 SUMMARIZECOLUMNNS 함수의 동작을 수정합니다. IGNORE를 사용하지 않는 모든 식이 BLANK/NULL을 반환하는 행은 IGNORE를 사용하는 식이 BLANK/NULL로 평가되는지 여부에 관계없이 제외됩니다. 이 함수는 SUMMARIZECOLUMNNS 식 내에서만 사용할 수 있습니다.

구문

DAX

```
IGNORE(<expression>)
```

SUMMARIZECOLUMNNS를 사용하여

DAX

```
SUMMARIZECOLUMNNS(<groupBy_columnName>[ , < groupBy_columnName >]... ,  
[<filterTable>]...[ , <name>, IGNORE(...)]...)
```

매개 변수

용어	정의
식	테이블이 아닌 단일 값을 반환하는 DAX 식입니다.

반환 값

함수가 값을 반환하지 않습니다.

설명

IGNORE는 SUMMARIZECOLUMNNS에 대한 [식 인수로만 사용할 수 있습니다.](#)

예시

SUMMARIZECOLUMNNS를 참조 [하세요](#).

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

INTERSECT

아티클 • 2023. 10. 20.

중복을 유지하면서 두 테이블의 행 교집합을 반환합니다.

구문

DAX

```
INTERSECT(<table_expression1>, <table_expression2>)
```

매개 변수

용어	정의
Table_expression	테이블을 반환하는 DAX 식입니다.

반환 값

table_expression2 있는 table_expression1 모든 행이 들어 있는 테이블입니다.

예외

설명

- 교차는 정류되지 않습니다. 일반적으로 Intersect(T1, T2)는 Intersect(T2, T1)와 다른 결과 집합을 갖습니다.
- 중복 행은 유지됩니다. 행이 table_expression1 table_expression2 표시되면 table_expression_1 모든 중복 항목이 결과 집합에 포함됩니다.
- 열 이름은 table_expression1 열 이름과 일치합니다.
- 반환된 테이블에는 두 번째 테이블의 열 계보에 관계없이 table_expression1 열을 기반으로 계보가 있습니다. 예를 들어 첫 번째 table_expression 첫 번째 열에 모델의 기본 열 C1에 대한 계보가 있는 경우 교차는 두 번째 table_expression 첫 번째 열의 교차에 따라 행을 줄이고 기본 열 C1의 계보를 그대로 유지합니다.
- 열은 위치 지정에 따라 비교되고 데이터 비교는 형식 강제 변환 없이 비교됩니다.

- 반환된 테이블에는 table_expression1 관련된 테이블의 열이 포함되지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

상태1

주
A
A
B
B
B
C
D
D

상태2

주
b
C
D
D
D
E

Intersect(States1, States2)

주
B

주

B

B

C

D

D

Intersect(States2, States1)

주

b

C

D

D

D

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NATURALINNERJOIN

아티클 • 2023. 12. 14.

다른 테이블을 사용하여 테이블의 내부 조인을 수행합니다.

구문

DAX

`NATURALINNERJOIN(<LeftTable>, <RightTable>)`

매개 변수

 테이블 확장

용어	정의
LeftTable	조인의 왼쪽에 있는 테이블을 정의하는 테이블 식입니다.
RightTable	조인의 오른쪽에 있는 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 있습니다.

설명

- 두 테이블의 공통 열(이름별)에서 조인됩니다. 두 테이블에 공통 열 이름이 없으면 오류가 반환됩니다.
- 결과에 대한 정렬 순서 보장은 없습니다.
- 조인되는 열은 두 테이블 모두에서 동일한 데이터 형식이어야 합니다.
- 동일한 원본 테이블의 열(계보가 동일)만 조인됩니다. 예를 들어 WebSales와 StoreSales 간에 다대일 관계가 있는 Products[ProductID], WebSales[ProductdID], ProductID 열을 기반으로 하는 Products 테이블, WebSales 및 StoreSales 테이블이 [ProductID]에 조인됩니다.

- 조인하는 동안 엄격한 비교 의미 체계가 사용됩니다. 형식 강제 변환이 없습니다. 예를 들어 1은 1.0과 같지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

[NATURALLEFTOUTERJOIN](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NATURALLEFTOUTERJOIN

아티클 • 2023. 12. 14.

왼쪽 우선 외부 조인 의미 체계를 사용하여 RightTable과 LeftTable의 조인을 수행합니다.

구문

DAX

`NATURALLEFTOUTERJOIN(<LeftTable>, <RightTable>)`

매개 변수

[+] 테이블 확장

용어	정의
LeftTable	조인의 왼쪽에 있는 테이블을 정의하는 테이블 식입니다.
RightTable	조인의 오른쪽에 있는 테이블을 정의하는 테이블 식입니다.

반환 값

지정된 공통 열의 값이 LeftTable에도 있는 RightTable의 행만 포함하는 테이블입니다. 반환되는 테이블에는 왼쪽 테이블의 공통 열과 두 테이블의 다른 열이 있습니다.

설명

- 두 테이블의 공통 열(이름별)에서 조인됩니다. 두 테이블에 공통 열 이름이 없으면 오류가 반환됩니다.
- 결과에 대한 정렬 순서 보장은 없습니다.
- 조인되는 열은 두 테이블 모두에서 동일한 데이터 형식이어야 합니다.
- 동일한 원본 테이블의 열(계보가 동일)만 조인됩니다. 예를 들어 WebSales와 StoreSales 간에 다대일 관계가 있는 Products[ProductID], WebSales[ProductdID], ProductID 열을 기반으로 하는 Products 테이블, WebSales 및 StoreSales 테이블이 [ProductID]에 조인됩니다.

- 조인하는 동안 엄격한 비교 의미 체계가 사용됩니다. 형식 강제 변환이 없습니다. 예를 들어 1은 1.0과 같지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

[NATURALINNERJOIN](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROLLUP

아티클 • 2023. 10. 20.

groupBy_columnName 매개 변수로 정의된 열의 결과에 롤업 행을 추가하여 SUMMARIZE 함수의 동작을 수정합니다. 이 함수는 SUMMARIZE 식 내에서만 사용할 수 있습니다.

구문

DAX

```
ROLLUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

SUMMARIZE를 사용하여

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[ , <groupBy_columnName>]...[ ,  
ROLLUP(<groupBy_columnName>[ ,< groupBy_columnName>...]])[, <name>,  
<expression>]...)
```

매개 변수

용어	정의
groupBy_columnName	해당 열에 있는 값을 기반으로 요약 그룹을 만드는데 사용할 기준 열 또는 ROLLUPGROUP 함수의 정규화된 이름입니다. 이 매개 변수는 식일 수 없습니다.

반환 값

이 함수는 값을 반환하지 않습니다. 부분합할 열 집합만 지정합니다.

설명

이 함수는 SUMMARIZE 식 내에서만 사용할 수 있습니다.

예시

요약을 참조하세요.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROLLUPADDISSUBTOTAL

아티클 • 2023. 10. 20.

groupBy_columnName 열을 기반으로 롤업/부분합 행을 결과에 추가하여 SUMMARIZECOLUMNNS [함수의 동작](#)을 수정합니다. 이 함수는 SUMMARIZECOLUMNNS [식 내에서](#)만 사용할 수 있습니다.

구문

DAX

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

매개 변수

용어	정의
grandtotalFilter	(선택 사항) 총합계 수준에 적용할 필터입니다.
groupBy_columnName	해당 열에 있는 값을 기반으로 요약 그룹을 만드는데 사용되는 기준 열의 이름입니다. 식일 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

없음

예시

SUMMARIZECOLUMNNS를 참조 [하세요](#).

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROLLUPGROUP

아티클 • 2023. 10. 20.

groupBy_columnName 매개 변수로 정의된 열의 결과에 롤업 행을 추가하여 [SUMMARIZE](#) 및 [SUMMARIZECOLUMNS](#) 함수의 동작을 수정합니다. 이 함수는 [SUMMARIZE](#) 또는 [SUMMARIZECOLUMNS](#) 식 내에서만 사용할 수 있습니다.

구문

DAX

```
ROLLUPGROUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

매개 변수

용어	정의
groupBy_columnName	해당 열에 있는 값을 기반으로 요약 그룹을 만드는데 사용할 기준 열 또는 ROLLUPGROUP 함수의 정규화된 이름입니다. 이 매개 변수는 식일 수 없습니다.

반환 값

이 함수는 값을 반환하지 않습니다. ROLLUP 또는 [ROLLUPADDISSUBTOTAL](#)로 부분합하는 동안 단일 그룹으로 처리할 열 집합을 표시합니다.

설명

ROLLUPGROUP은 ROLLUP, ROLLUPADDISSUBTOTAL 또는 [ROLLUPISSUBTOTAL](#)에 대한 groupBy_columnName 인수로만 사용할 수 있습니다.

예시

[SUMMARIZE](#) 및 [SUMMARIZECOLUMNS](#)를 참조하세요.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROLLUPISSUBTOTAL

아티클 • 2023. 10. 20.

ROLLUPADDISSTOTAL에서 추가한 [열과 룰업 그룹을 페어링합니다](#). 이 함수는 ADDMISSINGITEMS 식 내에서만 사용할 수 있습니다.

구문

DAX

```
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	해당 열에 있는 값을 기반으로 요약 그룹을 만드는 데 사용되는 기준 열의 이름입니다. 식일 수 없습니다.
isSubtotal_columnName	ISSUBTOTAL 열의 이름입니다. 열의 값은 ISSUBTOTAL 함수를 사용하여 계산됩니다.
groupLevelFilter	(선택 사항) 현재 수준에 적용할 필터입니다.

반환 값

None

설명

이 함수는 ADDMISSINGITEMS 식 내에서만 사용할 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ROW 함수

아티클 • 2023. 10. 20.

각 열에 지정된 식에서 생성된 값이 포함된 단일 행이 있는 테이블을 반환합니다.

구문

DAX

```
ROW(<name>, <expression>[ ,<name>, <expression>]...)
```

매개 변수

용어	정의
name	큰따옴표로 묶인 열에 지정된 이름입니다.
식	채울 단일 스칼라 값을 반환하는 DAX 식입니다. <i>name</i> .

반환 값

단일 행 테이블

설명

- 인수는 항상 이름과 식 쌍으로 제공되어야 합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 인터넷 및 재판매인 채널의 총 판매액이 있는 단일 행 테이블을 반환합니다.

DAX

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),
    "Resellers Total Sales (USD)",
```

```
SUM(ResellerSales_USD[SalesAmount_USD]))
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SELECTCOLUMNS

아티클 • 2023. 10. 20.

테이블에서 선택한 열과 DAX 식에서 지정한 새 열이 있는 테이블을 반환합니다.

구문

DAX

```
SELECTCOLUMNS(<Table>, [<Name>], <Expression>, <Name>], ...)
```

매개 변수

용어	정의
테이블	테이블을 반환하는 DAX 식입니다.
이름	큰따옴표로 묶인 열에 지정된 이름입니다.
식	열 참조, 정수 또는 문자열 값과 같은 스칼라 값을 반환하는 식입니다.

반환 값

첫 번째 인수로 지정된 테이블과 행 수가 같은 테이블입니다. 반환된 테이블에는 <Name>, <Expression> 인수의 각 쌍에 해당하는 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 Customer 테이블을 사용한다고 가정합니다.

Country	주	개수	합계
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900
미국	CA	5	500
미국	WA	10	900

DAX

```
SELECTCOLUMNS(Customer, "Country, State", [Country]&", "&[State])
```

반환

국가, 주
IND, JK
IND, MH
IND, WB
미국, CA
미국, WA

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SUBSTITUTEWITHINDEX

아티클 • 2023. 10. 20.

인수로 제공된 두 테이블의 왼쪽 세미조인을 나타내는 테이블을 반환합니다. 세미조인은 공통 열 이름 및 공통 데이터 형식에 따라 결정되는 공통 열을 사용하여 수행됩니다. 조인되는 열은 정수 형식이고 인덱스를 포함하는 반환된 테이블의 단일 열로 바뀝니다. 인덱스는 지정된 정렬 순서가 지정된 경우 오른쪽 조인 테이블에 대한 참조입니다.

제공된 왼쪽/첫 번째 테이블에 없는 제공된 오른쪽/초 테이블의 열은 반환된 테이블에 포함되지 않으며 조인에 사용되지 않습니다.

인덱스는 0부터 시작하여 제공된 오른쪽/초 조인 테이블의 각 추가 행에 대해 하나씩 증가합니다. 인덱스는 오른쪽/초 조인 테이블에 지정된 정렬 순서를 기반으로 합니다.

구문

DAX

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>][, <orderBy_expression>, [<order>]]...])
```

매개 변수

용어	정의
table	세 번째 인수(indexColumnsTable)로 지정된 테이블을 사용하여 왼쪽 세미조인을 수행하여 필터링할 테이블입니다. 이 테이블은 왼쪽 세미조인의 왼쪽에 있는 테이블이므로 반환된 테이블에는 두 테이블의 모든 공통 열이 반환된 테이블의 단일 인덱스 열로 대체된다는 점을 제외하고 이 테이블과 동일한 열이 포함됩니다.
indexColumnName	이 함수에 대한 인수로 제공된 두 테이블의 모든 공통 열을 대체하는 인덱스 열의 이름을 지정하는 문자열입니다.
indexColumnsTable	왼쪽 세미조인의 두 번째 테이블입니다. 왼쪽 세미조인의 오른쪽에 있는 테이블입니다. 이 테이블에 있는 값만 함수에서 반환됩니다. 또한 이 테이블의 열(열 이름 기준)은 이 함수에서 반환된 테이블의 단일 인덱스 열로 바뀝니다.
orderBy_expression	결과 값이 올바른 인덱스 값을 생성하기 위해 indexColumnsTable 테이블의 원하는 정렬 순서를 지정하는 데 사용되는 DAX 식입니다. indexColumnsTable 테이블에 지정된 정렬 순서는 테이블의 각 행의 인덱스를 정의하며, 이 인덱스는 이 함수에 대한 첫 번째 인수로 제공된 테이블에 표

용어	정의
	시될 때 indexColumnsTable의 값 조합을 나타내기 위해 반환된 테이블에서 사용됩니다.
order	<p>(선택 사항) orderBy_expression 값을 오름차순 또는 내림차순으로 정렬하는 방법을 지정하는 값입니다.</p> <p>값: Desc. 대체 값: 0(zero)/FALSE. orderBy_expression 값의 내림차순으로 정렬합니다. 순서 매개 변수를 생략하면 기본값입니다.</p> <p>값: ASC. 대체 값: 1/TRUE. orderBy_expression 오름차순으로 순위를 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

피드백

이 페이지가 도움이 되었나요?

Yes

No

SUMMARIZE

아티클 • 2023. 10. 20.

그룹 집합에 대해 요청된 합계에 대한 요약 테이블을 반환합니다.

구문

DAX

```
SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

매개 변수

용어	정의
table	데이터 테이블을 반환하는 DAX 식입니다.
groupBy_ColumnName	(선택 사항) 해당 열에 있는 값을 기반으로 요약 그룹을 만드는 데 사용되는 기존 열의 정규화된 이름입니다. 이 매개 변수는 식일 수 없습니다.
name	큰따옴표로 묶인 요약 열 또는 요약 열에 지정된 이름입니다.
식	단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.

반환 값

groupBy_columnName 인수에 대해 선택한 열과 이름 인수로 디자인된 요약된 열이 있는 테이블입니다.

설명

- 이름을 정의하는 각 열에는 해당 식이 있어야 합니다. 그렇지 않으면 오류가 반환됩니다. 첫 번째 인수인 이름은 결과에서 열의 이름을 정의합니다. 두 번째 인수인 식은 해당 열의 각 행에 대한 값을 얻기 위해 수행되는 계산을 정의합니다.
- groupBy_columnName 테이블이나 테이블과 관련된 테이블에 있어야 합니다.
- 각 이름은 큰따옴표로 묶어야 합니다.
- 함수는 하나 이상의 groupBy_columnName 열 값으로 선택한 행 집합을 요약 행 집합으로 그룹화합니다. 각 그룹에 대해 하나의 행이 반환됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 역년 및 제품 범주 이름을 중심으로 그룹화된 재판매인 판매 요약을 반환합니다. 이 결과 테이블에서는 연도 및 제품 범주별로 재판매인 판매를 분석할 수 있습니다.

DAX

```
SUMMARIZE(ResellerSales_USD  
    , DateTime[CalendarYear]  
    , ProductCategory[ProductCategoryName]  
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])  
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])  
)
```

다음 표에서는 테이블을 수신할 것으로 예상되는 함수가 수신하는 데이터의 미리 보기입니다.

DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액(USD)]	[할인 금액(USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	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	Accessories	153299.924	865.5945
2005	Accessories	18594.4782	4.293
2006	Accessories	86612.7463	1061.4872
2007	Accessories	275794.8403	4756.6546

ROLLUP을 사용하여

ROLLUP 구문을 추가하면 groupBy_columnName 열의 결과에 롤업 행을 추가하여 SUMMARIZE 함수의 동작을 수정합니다. ROLLUP은 SUMMARIZE 식 내에서만 사용할 수 있습니다.

예시

다음 예제에서는 SUMMARIZE 함수 호출의 Group-By 열에 롤업 행을 추가합니다.

DAX

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductName])
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

다음 표를 반환합니다.

DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액(USD)]	[할인 금액(USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	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

DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액(USD)]	[할인 금액(USD)]
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	Accessories	153299.924	865.5945
2005	Accessories	18594.4782	4.293
2006	Accessories	86612.7463	1061.4872
2007	Accessories	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 부분합을 방지할 수](#) 있습니다. ROLLUPGROUP은 ROLLUP, ROLLUPADDISSUBTOTAL 또는 ROLLUPISSUBTOTAL 식 내에서만 사용할 수 있습니다.

예시

다음 예제에서는 모든 범주를 사용하여 각 연도의 부분합이 없는 모든 연도 및 범주의 총합계만 보여줍니다.

DAX

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductName]))
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

다음 표를 반환합니다.

DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액(USD)]	[할인 금액(USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	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

DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액(USD)]	[할인 금액(USD)]
2007	의류	815853.2868	12489.3835
2008	Accessories	153299.924	865.5945
2005	Accessories	18594.4782	4.293
2006	Accessories	86612.7463	1061.4872
2007	Accessories	275794.8403	4756.6546
		76494758.25	527507.9262

ISSUBTOTAL 사용

ISSUBTOTAL을 사용하면 ROWTOTAL에 인수로 지정된 열에 대한 부분합 값이 행에 포함되어 있으면 `True`를 반환하는 SUMMARIZE 식에 다른 열을 만들 수 있습니다. 그렇지 않으면 `False`를 반환합니다. `ISSUBTOTAL`은 SUMMARIZE 식 내에서만 사용할 수 있습니다.

예시

다음 샘플에서는 지정된 SUMMARIZE 함수 호출의 각 ROLLUP 열에 대해 ISSUBTOTAL 열을 생성합니다.

DAX
<pre>SUMMARIZE(ResellerSales_USD , ROLLUP(DateTime[CalendarYear], ProductCategory[ProductName]) , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD]) , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount]) , "Is Sub Total for DateTime[CalendarYear]", ISSUBTOTAL(DateTime[CalendarYear]) , "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductName]))</pre>

다음 표를 반환합니다.

[DateTimeCalendarYear 의 하위 합계입니다.]	[ProductCategoryName 의 하위 합계입니다.]	DateTime[CalendarYear]	ProductCategory[ProductName]	[판매액 (USD)]	[할인 금액 (USD)]
FALSE	거짓				
거짓	FALSE	2008	Bikes	12968255.42	36167.6592
FALSE	FALSE	2005	Bikes	6958251.043	4231.1621
FALSE	FALSE	2006	Bikes	18901351.08	178175.8399
FALSE	FALSE	2007	Bikes	24256817.5	276065.992
FALSE	FALSE	2008	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	Accessories	153299.924	865.5945
FALSE	FALSE	2005	Accessories	18594.4782	4.293
FALSE	FALSE	2006	Accessories	86612.7463	1061.4872

[DateTimeCalendarYear 의 하위 합계입니다.]	[ProductCategoryName 의 하위 합계입니다.]	DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[판매액 (USD)]	[할인 금액 (USD)]
FALSE	FALSE	2007	Accessories	275794.8403	4756.6546
FALSE	TRUE				
거짓	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

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SUMMARIZECOLUMNS

아티클 • 2023. 12. 22.

그룹 집합에 대한 요약 테이블을 반환합니다.

구문

DAX

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, <groupBy_columnName>]..., [<filterTable>]...[, <name>, <expression>]...)
```

매개 변수

 테이블 확장

용어	정의
groupBy_columnName	반환된 테이블에 고유 값이 포함된 기본 테이블에 대한 정규화된 열 참조 (Table[Column])입니다. 각 groupBy_columnName 열은 다음에 지정된 열과 교차 조인(서로 다른 테이블) 또는 자동 존재(동일한 테이블)입니다.
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	테이블이 아닌 단일 값을 반환하는 DAX 식입니다.

반환 값

지정된 그룹화에 따라 제공된 열의 값 조합이 포함된 테이블입니다. 제공된 식 중 하나 이상이 비어있지 않은 값을 반환하는 행만 반환되는 테이블에 포함됩니다. 모든 식이 행에 대해 BLANK/NULL로 평가되는 경우 해당 행은 반환되는 테이블에 포함되지 않습니다.

설명

- 이 함수는 결과에 대한 정렬 순서를 보장하지 않습니다.

- `groupBy_columnName` 매개 변수에서 열을 두 번 이상 지정할 수 없습니다. 예를 들어 다음 수식이 잘못되었습니다.

```
SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
```

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

필터 컨텍스트

다음과 같은 쿼리를 고려해 보세요.

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

이 쿼리에서 측정값이 없으면 `groupBy` 열에는 `FILTER` 식의 열(예: `Customer` 테이블)이 포함되지 않습니다. `groupBy` 열에는 필터가 적용되지 않습니다. `Sales Territory` 및 `Customer` 테이블은 재판매인 판매 팩트 테이블을 통해 간접적으로 관련될 수 있습니다. 직접 관련이 없으므로 필터 식은 no-op이며 `groupBy` 열은 영향을 받지 않습니다.

그러나 이 쿼리를 사용하면 다음을 수행합니다.

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer'[First Name] = "Alicia")
)
```

`groupBy` 열에는 필터의 영향을 받는 열이 포함되어 있으며 해당 필터는 `groupBy` 결과에 적용됩니다.

IGNORE 포함

`IGNORE` 구문을 사용하여 BLANK/NULL 계산에서 특정 식을 생략하여 `SUMMARIZECOLUMNS` 함수의 동작을 수정할 수 있습니다. `IGNORE`를 사용하지 않는 모든 식이 BLANK/NULL을 반환하는 행은 `IGNORE`를 사용하는 식이 BLANK/NULL로 평가되는지 여부에 관계없이 제외됩니다. `IGNORE`는 `SUMMARIZECOLUMNS` 식 내에서만 사용할 수 있습니다.

예시

DAX

```
SUMMARIZECOLUMNS(  
    Sales[CustomerId], "Total Qty",  
    IGNORE( SUM( Sales[Qty] ) ),  
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )  
)
```

그러면 Sales[CustomerId] 열이 룰업되어 지정된 그룹화의 모든 고객에 대한 부분합이 생성됩니다. IGNORE가 없으면 결과는 다음과 같습니다.

[+] 테이블 확장

고객 ID	TotalQty	BlankIfTotalQtyIsNot3
A	5	
b	3	3
C	3	3

IGNORE를 사용하여

[+] 테이블 확장

고객 ID	TotalQty	BlankIfTotalQtyIsNot3
B	3	3
C	3	3

모든 식이 무시됨,

DAX

```
SUMMARIZECOLUMNS(  
    Sales[CustomerId], "Blank",  
    IGNORE( BLANK() ), "BlankIfTotalQtyIsNot5",  
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )  
)
```

두 식이 일부 행에 대해 공백으로 반환되더라도 공백을 반환하는 서명되지 않은 식이 있으므로 포함됩니다.

고객 ID	Blank	BlankIfTotalQtyIsNot5
A	5	
b		
C		

NONVISUAL을 사용하는 경우

NONVISUAL [함수는](#) SUMMARIZECOLUMNES 함수의 값 필터를 측정값에 영향을 주지 않고 groupBy 열에만 적용하는 것으로 표시합니다. [NONVISUAL](#) 은 SUMMARIZECOLUMNES 식 내에서만 사용할 수 있습니다.

예시

DAX

```

DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNES
(
    DimDate[CalendarYear],
    NONVISUAL(TREATAS({2007, 2008}, DimDate[CalendarYear])),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales],
    ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]

```

[Visual Total Sales]가 모든 연도의 합계인 결과를 반환합니다.

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 함수가 없는 동일한 쿼리는 다음과 같습니다.

DAX

```

DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    TREATAS({2007, 2008}, DimDate[CalendarYear]),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales],
    ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]

```

[Visual Total Sales]가 선택한 2년 동안의 합계인 결과를 반환합니다.

[] 테이블 확장

DimDate[CalendarYear]	[Sales]	[Visual Total Sales]
2007	9,791,060.30	19,561,960.04
2008	9,770,899.74	19,561,960.04

ROLLUPADDISSTOTAL 사용

ROLLUPADDISSTOTAL [구문을 추가](#)하면 groupBy_columnName 열을 기반으로 룰업/부분합 행을 결과에 추가하여 SUMMARIZECOLUMNS 함수의 동작을 수정합니다.

[ROLLUPADDISSTOTAL](#)은 SUMMARIZECOLUMNS 식 내에서만 사용할 수 있습니다.

단일 부분합을 사용하는 예제

DAX

```

DEFINE
VAR vCategoryFilter =
    TREATAS({"Accessories", "Clothing"}, Product[Category])
VAR vSubcategoryFilter =
    TREATAS({"Bike Racks", "Mountain Bikes"}, Product[Subcategory])
EVALUATE
SUMMARIZECOLUMNS
(
    ROLLUPADDISSTOTAL
    (
        Product[Category], "IsCategorySubtotal", vCategoryFilter,
        Product[Subcategory], "IsSubcategorySubtotal", vSubcategoryFilter
    ),
    "Total Qty", SUM(Sales[Qty])
)

```

```

)
ORDER BY
[IsCategorySubtotal] DESC, [Category],
[IsSubcategorySubtotal] DESC, [Subcategory]

```

다음 표를 반환합니다.

 테이블 확장

범주	하위 범주	IsCategorySubtotal	IsSubcategorySubtotal	총 Qty
		True	True	60398
Accessories		False	True	36092
Accessories	자전거 랙	False	False	328
Bikes	산악용 자전거	False	False	4970
의류		False	True	9101

여러 부분합이 있는 예제

DAX

```

SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSTOTAL ( Sales[CustomerId],
"IsCustomerSubtotal" ),
    ROLLUPADDISSTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM(
Sales[Qty] )
)

```

판매는 주별, 고객별, 날짜별, 1에 대한 부분합으로 그룹화됩니다. 주별 판매액(날짜 2 기준)입니다. 주별 판매액(고객 3 기준) 고객 및 날짜 모두에 롤업되어 주별 판매로 이어집니다.

다음 표를 반환합니다.

 테이블 확장

CustomerID	IsCustomerSubtotal	State(상태)	총 Qty	날짜	IsDateSubtotal
A	FALSE	WA	5	2014/7/10	
B	FALSE	WA	1	2014/7/10	
B	FALSE	WA	2	7/11/2014	

CustomerID	IsCustomerSubtotal	State(상태)	총 Qty	날짜	IsDateSubtotal
C	FALSE	또는	2	2014/7/10	
C	FALSE	또는	1	7/11/2014	
	TRUE	WA	6	2014/7/10	
	TRUE	WA	2	7/11/2014	
	TRUE	또는	2	2014/7/10	
	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 ROLLUPGROUP을 ROLLUPADDISSTOTAL과 함께 사용하여 포함할 요약 그룹/세분성(부분합)을 지정하여 반환되는 부분합 행 수를 줄일 수 있습니다. **ROLLUPGROUP**은 SUMMARIZECOLUMNS 또는 SUMMARIZE 식 내에서만 사용할 수 있습니다.

여러 부분합이 있는 예제

DAX

```
SUMMARIZECOLUMNS(
    ROLLUPADDISSTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSTOTAL( ROLLUPGROUP(Regions[City], Regions[State]),
    "IsCityStateSubtotal"), "Total Qty", SUM( Sales[Qty] )
)
```

여전히 도시 및 주별로 그룹화되지만 부분합을 보고할 때 함께 앱연되면 다음 표가 반환됩니다.

[+] 테이블 확장

State(상태)	고객 ID	IsCustomerSubtotal	총 Qty	City	IsCityStateSubtotal
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

관련 콘텐츠

[요약](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

테이블 생성자

아티클 • 2023. 10. 20.

하나 이상의 열 테이블을 반환합니다.

구문

DAX

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ),  
... }
```

매개 변수

용어	정의
scalarExprN	스칼라 값을 반환하는 DAX 식입니다.

반환 값

하나 이상의 열 테이블입니다. 열이 하나만 있는 경우 열의 이름은 Value입니다. 열이 N개 있는 경우(여기서 $N > 1$) 열 이름은 왼쪽에서 오른쪽으로 Value1, Value2, ..., ValueN이 됩니다.

설명

- 첫 번째 구문은 단일 열의 테이블을 반환합니다. 두 번째 구문은 하나 이상의 열 테이블을 반환합니다.
- 스칼라 식의 수는 모든 행에 대해 동일해야 합니다.
- 열 값의 데이터 형식이 행이 다르면 모든 값이 공통 데이터 형식으로 변환됩니다.

예 1

다음 DAX 쿼리:

DAX

```
EVALUATE { 1, 2, 3 }
```

및

DAX

```
EVALUATE { (1), (2), (3) }
```

단일 열의 다음 테이블을 반환합니다.

[값]
1
2
3

예제 2

다음 DAX 쿼리:

DAX

```
EVALUATE
{
    (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),
    (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),
    (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")
}
```

반환

[Value1]	[Value2]	[Value3]	[Value4]
1.5	2017/1/1	199.99	A
2.5	1/2/2017	249.99	b
3.5	1/3/2017	299.99	C

예 3

다음 DAX 쿼리:

DAX

```
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

문자열 데이터 형식의 단일 열에 대한 다음 테이블을 반환합니다.

[값]
1
2017/1/1
TRUE
A

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TOPN

아티클 • 2023. 10. 20.

지정된 테이블의 상위 N개 행을 반환합니다.

구문

DAX

```
TOPN(<N_Value>, <Table>, <OrderBy_Expression>, [<Order>[,<OrderBy_Expression>, [<Order>]]...])
```

매개 변수

매개 변수	정의
N_Value	반환할 행 수입니다. 스칼라 값을 반환하는 DAX 식입니다. 이 식은 행/컨텍스트별로 여러 번 계산됩니다. 반환되는 행 수가 <i>n_value</i> 보다 클 수 있는 경우를 자세히 알아보려면 설명 섹션을 참조하세요.
테이블	상위 'n'개 행을 추출할 데이터의 테이블을 반환하는 DAX 식입니다. 빈 테이블이 반환되는 경우를 자세히 알아보려면 설명 섹션을 참조하세요.
OrderBy_Expression	(선택 사항) 결과 값을 사용하여 테이블을 정렬하고 테이블의 각 행에 대해 계산되는 DAX 식입니다.
순서	(선택 사항) <i>OrderBy_Expression</i> 값을 정렬하는 방법을 지정하는 값입니다. <ul style="list-style-type: none">- 0(영) 또는 FALSE. <i>Order</i> 값을 내림차순으로 정렬합니다. <i>Order</i> 매개 변수가 생략된 경우 기본값입니다.- 1 또는 TRUE. <i>OrderBy</i>의 오름차순으로 순위를 매깁니다.

반환 값

*Table*의 N개 행으로 구성된 테이블이거나 *N_Value*가 0(영) 이하인 경우 빈 테이블입니다. 행이 특정 순서로 정렬되지 않습니다.

설명

- 테이블의 N번째 행에서 *Order_By* 값의 동률이 있는 경우 모든 동률 행이 반환됩니다. 그런 다음 N번째 행에 연결이 있는 경우 함수는 n개 이상의 행을 반환할 수 있습니다.
- N_Value가 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 측정값 수식은 판매 금액별로 판매된 상위 10개 제품을 반환합니다.

DAX

```
= SUMX(
    TOPN(
        10,
        SUMMARIZE(
            InternetSales,
            InternetSales[ProductKey],
            "TotalSales", SUM(InternetSales[SalesAmount])
        ),
        [TotalSales], DESC
    ),
    [TotalSales]
)
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TREATAS

아티클 • 2023. 10. 20.

테이블 식의 결과를 관련 없는 테이블의 열에 필터로 적용합니다.

구문

DAX

```
TREATAS(table_expression, <column>[ , <column>[ , <column>[,...]]] )
```

매개 변수

용어	정의
table_expression	테이블이 생성되는 식입니다.
column	하나 이상의 기준 열. 식일 수 없습니다.

반환 값

table_expression 열에 있는 모든 행이 들어 있는 테이블입니다.

설명

- 지정한 열 수는 테이블 식의 열 수와 일치해야 하며 순서가 같아야 합니다.
- 테이블 식에 반환된 값이 열에 없으면 무시됩니다. 예를 들어 TREATAS({"Red", "Green", "Yellow"}, DimProduct[Color])는 DimProduct[Color] 열에 대한 필터를 "빨강", "녹색" 및 "노란색"의 세 가지 값으로 설정합니다. DimProduct[Color]에 "Yellow"가 없으면 유효 필터 값은 "Red" 및 "Green"입니다.
- 테이블 간에 관계가 없는 경우에 가장 적합합니다. 관련된 테이블 간에 여러 관계가 있는 경우 USERELATIONSHIP을 대신 사용하는 것이 좋습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서 모델에는 관련이 없는 두 개의 제품 테이블이 포함되어 있습니다. 사용자가 DimProduct1[ProductCategory]에 필터를 적용하여 자전거, 좌석, 타이어, 동일한 필터, 자전거, 좌석, 타이어를 선택하는 경우 DimProduct2[ProductCategory]에 적용됩니다.

DAX

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

참고 항목

[INTERSECT](#)

[필터](#)

[USERELATIONSHIP](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UNION

아티클 • 2023. 10. 20.

테이블 쌍에서 공용 구조체(조인) 테이블을 만듭니다.

구문

DAX

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

매개 변수

용어	정의
table_expression	테이블을 반환하는 DAX 식입니다.

반환 값

두 테이블 식의 모든 행을 포함하는 테이블입니다.

설명

- 두 테이블의 열 수는 같아야 합니다.
- 열은 해당 테이블의 위치별로 결합됩니다.
- 반환 테이블의 열 이름은 table_expression1 열 이름과 일치합니다.
- 중복 행은 유지됩니다.
- 반환된 테이블에는 가능한 경우 계보가 있습니다. 예를 들어 각 table_expression 첫 번째 열에 모델의 동일한 기본 열 C1에 대한 계보가 있는 경우 UNION 결과의 첫 번째 열에는 C1에 대한 계보가 있습니다. 그러나 결합된 열에 다른 기본 열에 대한 계보가 있거나 확장 열이 있는 경우 UNION의 결과 열에는 계보가 없습니다.
- 데이터 형식이 다른 경우 결과 데이터 형식은 데이터 형식 강제 변환에 대한 규칙에 따라 결정됩니다.
- 반환된 테이블에는 관련 테이블의 열이 포함되지 않습니다.

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

예시

다음 식은 USAInventory 테이블과 INDInventory 테이블을 단일 테이블로 결합하여 공용 구조체를 만듭니다.

DAX
<code>UNION(UsaInventory, IndInventory)</code>

USAInventory

국가/지역	주	개수	합계
미국	CA	5	500
미국	WA	10	900

INDInventory

국가/지역	주	개수	합계
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900

반환 테이블,

국가/지역	주	개수	합계
미국	CA	5	500
미국	WA	10	900
IND	2K	20	800
IND	MH	25	1000
IND	Wb	10	900

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

VALUES

아티클 • 2023. 10. 20.

입력 매개 변수가 열 이름인 경우 지정된 열의 고유 값이 포함된 하나의 열 테이블을 반환합니다. 중복 값이 제거되고 고유한 값만 반환됩니다. BLANK 값을 추가할 수 있습니다. 입력 매개 변수가 테이블 이름인 경우 지정된 테이블의 행을 반환합니다. 중복 행은 유지됩니다. BLANK 행을 추가할 수 있습니다.

① 참고

이 함수는 워크시트의 셀이나 열에 값을 반환하는 데 사용할 수 없습니다. 대신 수식에 중첩된 중간 함수로 사용하여 다른 값을 필터링하거나 합계를 계산하는 데 사용할 수 있는 고유 값 목록을 가져옵니다.

구문

DAX

`VALUES(<TableNameOrColumnName>)`

매개 변수

용어	정의
TableName 또는 ColumnName	고유한 값을 반환할 열 또는 행을 반환할 테이블입니다.

반환 값

입력 매개 변수가 열 이름인 경우 단일 열 테이블입니다. 입력 매개 변수가 테이블 이름인 경우 동일한 열의 테이블이 반환됩니다.

설명

- 필터링된 컨텍스트에서 VALUES 함수를 사용하는 경우 VALUES에서 반환된 고유 값은 필터의 영향을 받습니다. 예를 들어 지역별로 필터링하고 City 값 목록을 반환하는 경우 목록에는 필터에서 허용하는 지역의 도시만 포함됩니다. 기존 필터에 관계

없이 모든 도시를 반환하려면 ALL 함수를 사용하여 테이블에서 필터를 제거해야 합니다. 두 번째 예제에서는 VALUES와 함께 ALL을 사용하는 방법을 보여 줍니다.

- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.
- VALUES를 사용하는 경우 모범 사례는 VALUES 대신 SELECTEDVALUE 사용을 참조하세요.

관련된 함수

대부분의 시나리오에서 인수가 열 이름인 경우 VALUES 함수의 결과는 DISTINCT 함수의 결과와 동일합니다. 두 함수 모두 중복 항목을 제거하고 지정된 열에서 가능한 값 목록을 반환합니다. 그러나 VALUES 함수는 빈 값을 반환할 수도 있습니다. 이 빈 값은 관련 테이블에서 고유 값을 조회하지만 관계에 사용된 값이 한 테이블에서 누락된 경우에 유용합니다. 데이터베이스 용어에서 이를 참조 무결성 위반이라고 합니다. 데이터에서 이러한 불일치는 하나의 테이블이 업데이트되고 관련 테이블이 업데이트되지 않을 때 발생할 수 있습니다.

인수가 테이블 이름인 경우 VALUES 함수의 결과는 참조 무결성 위반이 있는 경우 지정된 테이블의 모든 행과 빈 행을 반환합니다. DISTINCT 함수는 중복 행을 제거하고 지정된 테이블에서 고유한 행을 반환합니다.

① 참고

DISTINCT 함수는 열 이름 또는 유효한 테이블 식을 인수로 사용할 수 있지만 VALUES 함수는 열 이름 또는 테이블 이름만 인수로 허용합니다.

다음 표에서는 참조 무결성이 유지되지 않을 때 두 개의 관련 테이블에서 발생할 수 있는 데이터 간의 불일치를 요약합니다.

MyOrders 테이블	MySales 테이블
6월 1일	6월 1일 판매
6월 2일	6월 2일 판매
(주문 날짜가 입력되지 않음)	6월 3일 판매

DISTINCT 함수를 사용하여 날짜 목록을 반환하는 경우 두 날짜만 반환됩니다. 그러나 VALUES 함수를 사용하는 경우 함수는 두 날짜와 빈 멤버를 더한 값을 반환합니다. 또한 MyOrders 테이블에 일치하는 날짜가 없는 MySales 테이블의 모든 행은 이 알 수 없는 멤버와 "일치"됩니다.

예시

다음 수식은 고유한 청구서(판매 주문)의 수를 계산하고 제품 범주 이름을 포함하는 보고서에서 사용될 때 다음 결과를 생성합니다.

DAX

```
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

반환

Row Labels	송장 개수 계산
Accessories	18,208
Bikes	15,205
의류	7,461
총합계	27,659

참고 항목

[FILTER 함수](#)

[COUNTROWS 함수](#)

[필터 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

텍스트 함수

아티클 • 2023. 10. 20.

DAX(데이터 분석 식)에는 Excel의 문자열 함수 라이브러리를 기반으로 하는 텍스트 함수 집합이 포함되어 있지만 테이블 형식 모델의 테이블 및 열에서 작동하도록 수정되었습니다. 이 섹션에서는 DAX 언어로 사용할 수 있는 텍스트 함수에 대해 설명합니다.

이 범주에서

함수	설명
COMBINEVALUES	두 개 이상의 텍스트 문자열을 하나의 텍스트 문자열에 조인합니다.
연결	두 텍스트 문자열을 하나의 텍스트 문자열에 조인합니다.
CONCATENATEX	테이블의 각 행에 대해 계산된 식의 결과를 연결합니다.
정확한	두 텍스트 문자열을 비교하고 정확히 같으면 TRUE를 반환하고, 그렇지 않으면 FALSE를 반환합니다.
찾을	다른 텍스트 문자열 내에서 한 텍스트 문자열의 시작 위치를 반환합니다.
고정	숫자를 지정된 소수 자릿수로 반올림하고 결과를 텍스트로 반환합니다.
FORMAT	지정된 형식에 따라 값을 텍스트로 변환합니다.
LEFT	텍스트 문자열의 시작 부분부터 지정된 문자 수를 반환합니다.
LEN	텍스트 문자열의 문자 수를 반환합니다.
낮은	텍스트 문자열의 모든 문자를 소문자로 변환합니다.
중간	시작 위치와 길이가 지정된 텍스트 문자열의 중간에서 문자 문자열을 반환합니다.
REPLACE	REPLACE는 지정한 문자 수에 따라 텍스트 문자열의 일부를 다른 텍스트 문자열로 바꿉니다.
REPT	지정된 횟수만큼 텍스트를 반복합니다.
RIGHT	RIGHT는 지정한 문자 수에 따라 텍스트 문자열의 마지막 문자 또는 문자를 반환합니다.
검색	왼쪽에서 오른쪽으로 읽는 특정 문자 또는 텍스트 문자열을 처음 찾은 문자의 수를 반환합니다.
대체	기존 텍스트를 텍스트 문자열의 새 텍스트로 대체합니다.

함수	설명
TRIM	단어 사이의 단일 공백을 제외하고 텍스트에서 모든 공백을 제거합니다.
UNICHAR	숫자 값에서 참조하는 유니코드 문자를 반환합니다.
UNICODE	텍스트 문자열의 첫 번째 문자에 해당하는 숫자 코드를 반환합니다.
상단	텍스트 문자열을 모든 대문자로 변환합니다.
값	숫자를 나타내는 텍스트 문자열을 숫자로 변환합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

COMBINEVALUES

아티클 • 2023. 10. 20.

두 개 이상의 텍스트 문자열을 하나의 텍스트 문자열에 조인합니다. 이 함수의 주요 목적은 DirectQuery 모델에서 다중 열 관계를 지원하는 것입니다. 자세한 내용은 [설명](#)을 참조하세요.

구문

DAX

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

매개 변수

용어	정의
delimiter	연결 중에 사용할 구분 기호입니다. 상수 값이어야 합니다.
식	값이 단일 텍스트 문자열에 조인되는 DAX 식입니다.

반환 값

연결된 문자열입니다.

설명

- COMBINEVALUES 함수는 입력 값이 다를 때 출력 문자열도 다르다고 가정하지만 유효성을 검사하지는 않습니다. 이 가정에 따라 COMBINEVALUES를 사용하여 두 DirectQuery 테이블의 여러 열을 조인하는 관계를 구축하기 위해 계산 열을 만드는 경우 쿼리 시 최적화된 조인 조건이 생성됩니다. 예를 들어 Table1(Column1, Column2)과 Table2(Column1, Column2) 간에 관계를 만들려는 경우 다음과 같이 각 테이블에 하나씩 두 개의 계산 열을 만들 수 있습니다.

DAX

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

및

DAX

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

그런 다음, 사이에 `Table1[CalcColumn]` `Table2[CalcColumn]` 관계를 만듭니다. 리터럴 그대로 해당 SQL 연산자 및 함수로 변환되는 다른 DAX 함수 및 연산자와 달리 위의 관계는 다음과 같이 SQL 조인 조건자를 생성합니다.

DAX

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

및

DAX

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND  
Table2.Column2 IS NULL)
```

- 조인 조건자는 복잡한 SQL 연산자 및 함수를 포함하는 것보다 훨씬 더 나은 쿼리 성능을 제공할 수 있습니다.
- COMBINEVALUES 함수는 사용자가 적절한 구분 기호를 선택하여 입력 값의 고유한 조합이 고유한 출력 문자열을 생성하도록 하지만 가정이 참인지는 확인하지 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 가져오는 경우 함께 조인되지만 두 테이블이 모두 동일한 DirectQuery 원본에 있는 경우 두 행이 함께 조인되지 않습니다.

예시

다음 DAX 쿼리:

DAX

```
EVALUATE  
DISTINCT (
```

```
SELECTCOLUMNS ( Date, "Month", COMBINEVALUES ( ", ", [MonthName],  
[CalendarYear] ) )
```

다음 단일 열 테이블을 반환합니다.

[월]
2020년 1월
2020년 2월
2020년 3월
2020년 4월
2020년 5월
2020년 6월
2020년 7월
2020년 8월
2020년 9월
2020년 10월
2020년 11월
2020년 12월
2021년 1월
2021년 1월
2021년 2월
2021년 3월
2021년 4월
2021년 5월
2021년 6월
2021년 7월
2021년 8월
2021년 9월
2021년 10월

[월]

2021년 11월

2021년 12월

피드백

이 페이지가 도움이 되었나요?

Yes

No

CONCATENATE

아티클 • 2023. 10. 20.

두 텍스트 문자열을 하나의 텍스트 문자열에 조인합니다.

구문

DAX

`CONCATENATE(<text1>, <text2>)`

매개 변수

용어 정의

`text1` 단일 텍스트 문자열로 조인할 첫 번째 텍스트 문자열입니다. 이 문자열은 텍스트 또는 숫자를 포함할 수 있습니다. 열 참조를 사용할 수도 있습니다.

`Text2` 단일 텍스트 문자열로 조인할 두 번째 텍스트 문자열입니다. 이 문자열은 텍스트 또는 숫자를 포함할 수 있습니다. 열 참조를 사용할 수도 있습니다.

반환 값

연결된 문자열입니다.

설명

- CONCATENATE 함수는 두 개의 텍스트 문자열을 하나의 텍스트 문자열로 조인합니다. 조인된 항목은 텍스트로 표시된 텍스트, 숫자, 부울 값이거나 이러한 항목의 조합일 수 있습니다. 열에 적절한 값이 포함된 경우 열 참조를 사용할 수도 있습니다.
- DAX의 CONCATENATE 함수는 두 개의 인수만 허용하는 반면 Excel 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 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예: 리터럴 연결

샘플 수식은 인수로 제공하는 두 문자열 값을 결합하여 새 문자열 값을 만듭니다.

DAX

```
= CONCATENATE("Hello ", "World")
```

예제: 열에서 문자열 연결

샘플 수식은 전화 번호부에 나열된 고객의 전체 이름을 반환합니다. 중첩 함수를 두 번째 인수로 사용하는 방법을 확인합니다. 인수로 사용하려는 값이 세 개 이상일 때 여러 문자열을 연결하는 한 가지 방법입니다.

DAX

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

예제: 열에서 문자열의 조건부 연결

샘플 수식은 이름, 중간 이니셜 및 성의 조합으로 전체 고객 이름을 사용하여 Customer 테이블에 새 계산 열을 만듭니다. 중간 이름이 없으면 성은 이름 바로 뒤로 옵니다. 중간 이름이 있는 경우 중간 이름의 첫 글자만 사용되고 초기 문자 뒤에 마침표가 있습니다.

DAX

```
= CONCATENATE([FirstName]&" ", CONCATENATE(IF(LEN([MiddleName])>1,  
LEFT([MiddleName],1)&" ", ""), [LastName]))
```

이 수식은 중첩된 CONCATENATE 및 IF 함수를 앤퍼샌드(&) 연산자와 함께 사용하여 세 문자열 값을 조건부로 연결하고 공백을 구분 기호로 추가합니다.

예제: 서로 다른 데이터 형식의 열 연결

다음 예제에서는 데이터 형식이 다른 열의 값을 연결 하는 방법을 보여 줍니다. 연결하려는 값이 숫자인 경우 값은 암시적으로 텍스트로 변환됩니다. 두 값이 모두 숫자이면 두 값이 모두 텍스트로 캐스팅되고 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

DAX

```
= CONCATENATE('Products'[Product abbreviation], 'Products'[Product number])
```

DAX의 CONCATENATE 함수는 두 개의 인수만 허용하는 반면 Excel CONCATENATE 함수는 최대 255개의 인수를 허용합니다. 인수를 더 추가해야 하는 경우 앤퍼샌드(&) 연산자를 사용할 수 있습니다. 예를 들어 다음 수식은 MTN-40 및 MTN-42 결과를 생성합니다.

DAX

```
= [Product abbreviation] & "-" & [Product number]
```

참고 항목

[CONCATENATEX](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CONCATENATEX

아티클 • 2023. 10. 20.

테이블의 각 행에 대해 계산된 식의 결과를 연결합니다.

구문

DAX

```
CONCATENATEX(<table>, <expression>[, <delimiter> [, <orderBy_expression> [, <order>]]...])
```

매개 변수

용어	정의
table	식이 계산될 행이 포함된 테이블입니다.
식	테이블의 각 행에 대해 계산할 식입니다.
delimiter	(선택 사항) 연결 중에 사용할 구분 기호입니다.
orderBy_expression	(선택 사항) 결과 값이 출력 문자열의 연결된 값을 정렬하는 데 사용되는 DAX 식입니다. 테이블의 각 행에 대해 평가됩니다.
order	(선택 사항) orderBy_expression 값을 오름차순 또는 내림차순으로 정렬하는 방법을 지정하는 값입니다.

선택적 order 매개 변수는 다음 값을 허용합니다.

값	대체 값	설명
0(영)	FALSE, DESC	orderBy_expression 값의 내림차순으로 정렬합니다. 순서 매개 변수를 생략할 때 기본값입니다.
1	TRUE, ASC	orderBy_expression 값의 오름차순으로 정렬합니다.

반환 값

연결된 문자열입니다.

설명

- 이 함수는 테이블을 반환하는 식 또는 첫 번째 인수로 사용합니다. 두 번째 인수는 연결할 값을 포함하는 열 또는 값을 반환하는 식입니다.
- orderBy_expression 지정하지 않는 한 연결된 값이 특정 순서로 정렬될 필요는 없습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

직원 테이블

FirstName	LastName
Alan	브루
Michael	Blythe

다음 수식은 다음과 같습니다.

```
DAX
= CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

HRESULT = NO_ERROR를
"앨런 브루어, 마이클 브라이스"

피드백

이 페이지가 도움이 되었나요?

Yes

No

정확한

아티클 • 2023. 12. 14.

두 텍스트 문자열을 비교하고 정확히 같으면 TRUE를, 아니면 FALSE를 반환합니다. EXACT는 대/소문자를 구분하지만 서식 차이는 무시합니다. EXACT는 대/소문자를 구분합니다.

구문

DAX

`EXACT(<text1>,<text2>)`

매개 변수

[+] 테이블 확장

용어	정의
text1	텍스트가 포함된 첫 번째 텍스트 문자열 또는 열입니다.
Text2	텍스트를 포함하는 두 번째 텍스트 문자열 또는 열입니다.

반환 값

True 또는 False (부울)

예시

Product 테이블의 계산된 열에 사용되는 다음 수식은 현재 행에 대한 Product 값과 비교하여 현재 행에 대한 Model 값을 확인하고 같으면 True를 반환하고, 다르면 False를 반환합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

DAX

`= EXACT([Product], [Model])`

관련 콘텐츠

텍스트 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

찾을

아티클 • 2023. 10. 20.

다른 텍스트 문자열 내에서 한 텍스트 문자열의 시작 위치를 반환합니다. FIND는 대/소문자를 구분합니다.

구문

DAX

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

매개 변수

용어	정의
find_text	찾을 텍스트입니다. 큰따옴표(빈 텍스트)를 사용하여 within_text 첫 번째 문자와 일치합니다.
within_text	찾으려는 텍스트가 포함된 텍스트입니다.
start_num	(선택 사항) 검색을 시작할 문자입니다. 생략 하면 start_num = 1입니다. within_text 첫 번째 문자는 문자 번호 1입니다.
NotFoundValue	(선택 사항이지만 강력하게 권장됨) 작업에서 일치하는 substring을 찾을 수 없는 경우에 반환되는 값으로, 일반적으로 0, -1 또는 BLANK()입니다. 지정하지 않으면 오류가 반환됩니다.

반환 값

찾으려는 텍스트 문자열의 시작점을 보여 주는 숫자입니다.

설명

- Microsoft Excel에는 SBCS(싱글바이트 문자 집합) 및 DBCS(더블바이트 문자 집합) 언어를 수용하는 여러 버전의 FIND 함수가 있지만 DAX는 유니코드를 사용하고 각 문자를 같은 방식으로 계산합니다. 따라서 문자 형식에 따라 다른 버전을 사용할 필요가 없습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

- FIND는 와일드 카드 지원하지 않습니다. wild카드를 사용하려면 SEARCH를 사용합니다.

예시

다음 DAX 쿼리는 재판매인 이름이 포함된 문자열에서 "Bike"라는 첫 번째 글자의 위치를 찾습니다. 찾을 수 없으면 Blank가 반환됩니다.

FIND는 대/소문자를 구분합니다. 이 예제에서 <find_text> 인수에 "bike"가 사용된 경우 결과가 반환되지 않습니다. 대/소문자를 구분하지 않으려면 [SEARCH](#)를 사용합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

```
DAX

EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of Bike", FIND ( "Bike", 'Reseller'[Reseller], 1, BLANK ())
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
    Reseller", "Warehouse" }
)
```

반환

재판매인	비즈니스 유형	Bike 위치
Volume Bike Sellers	창고	8
Mass Market Bikes	부가 가치 재판매인	13
Twin Cycles	부가 가치 재판매인	
Rich Department Store	창고	
Rental Gallery	Specialty Bike Shop	
Budget Toy Store	창고	
Global Sports Outlet	창고	
Online Bike Catalog	창고	8
Helmets and Cycles	부가 가치 재판매인	

재판매인	비즈니스 유형	Bike 위치
Jumbo Bikes	Specialty Bike Shop	7

참고 항목

검색

텍스트 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FIXED

아티클 • 2023. 12. 14.

숫자를 지정된 소수 자릿수로 반올림하고 결과를 텍스트로 반환합니다. 결과가 쉼표의 사용 여부에 관계없이 반환되도록 지정할 수 있습니다.

구문

DAX

`FIXED(<number>, <decimals>, <no_commas>)`

매개 변수

[+] 테이블 확장

용어	정의
번호	반올림하여 텍스트로 변환할 숫자 또는 숫자가 포함된 열입니다.
소수	(선택 사항) 소수점 오른쪽에 있는 자릿수입니다. 생략하면 2입니다.
no_commas	(선택 사항) 논리 값: 1이면 반환된 텍스트에 쉼표가 표시되지 않습니다. 0을 지정하거나 생략하면 반환된 텍스트에 쉼표가 표시됩니다.

반환 값

텍스트로 표시되는 숫자입니다.

설명

- decimals **매개 변수**에 사용되는 값이 음수이면 숫자가 소수점 왼쪽으로 반올림됩니다.
- 10진수를 생략하면 2로 간주됩니다.
- no_commas 0이거나 생략된 경우 반환된 텍스트에는 평소와 같이 쉼표가 포함됩니다.
- 명령을 사용하여 숫자가 포함된 셀의 서식을 지정하고 FIXED 함수를 사용하여 숫자를 직접 서식을 지정하는 경우의 주요 차이점은 FIXED에서 결과를 텍스트로 변환한

다는 점입니다. 서식 메뉴의 명령으로 서식이 지정된 숫자는 여전히 숫자입니다.

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

예시

계산 열에 사용된 다음 수식은 Product[List Price]의 현재 행에 대한 숫자 값을 가져오고 소수 자릿수가 2이고 쉼표가 없는 텍스트로 반환합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#) 을 참조하세요.

DAX

```
= FIXED([List Price],2,1)
```

관련 콘텐츠

[천장](#)

[바닥](#)

[Iso. 천장](#)

[MROUND](#)

[라운드](#)

[라운드다운](#)

[검거](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FORMAT

아티클 • 2023. 10. 20.

지정된 형식에 따라 값을 텍스트로 변환합니다.

구문

DAX

```
FORMAT(<value>, <format_string>[, <locale_name>])
```

매개 변수

용어	정의
value	단일 값으로 계산되는 값 또는 식입니다.
format_string	서식 템플릿이 있는 문자열입니다.
locale_name	(선택 사항) 함수에서 사용할 로캘의 이름입니다. 가능한 값은 Windows API 함수 LocaleNameToLCID() 에서 허용하는 문자열입니다.

반환 값

format_string 정의된 대로 서식이 지정된 값을 포함하는 문자열입니다.

① 참고

값이 BLANK이면 함수는 빈 문자열을 반환합니다.

format_string BLANK이면 값 형식이 "일반 번호" 또는 "일반 날짜" 형식으로 지정됩니다(값 데이터 형식에 따라).

설명

- 미리 정의된 형식 문자열은 결과의 서식을 지정할 때 모델 문화권 속성을 사용합니다. 기본적으로 모델 문화권 속성은 컴퓨터의 사용자 로캘에 따라 설정됩니다. 새 Power BI Desktop 모델의 경우 모델 culture 속성은 옵션 > 국가별 설정 > 모델 언어

에서 변경할 수 있습니다. Analysis Services의 경우 모델 문화권은 인스턴스에서 처음 정의한 언어 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 결과 또는 오류가 발생할 수 있습니다. 예를 들어 "백분율"의 약어인 "p"는 지원되지 않습니다. 미리 정의된 형식 문자열 목록에 포함되지 않은 FORMAT 함수에 대한 인수로 제공하는 문자열은 사용자 지정 형식 문자열의 일부 또는 문자열 리터럴로 처리됩니다.
- FORMAT을 사용하여 로캘을 지정하는 방법에 대한 자세한 내용은 관련 [동영상](#) 을 참조하세요.
- FORMAT을 사용하면 측정값 결과가 텍스트 데이터 형식으로 변경됩니다. 측정값 결과가 원래 숫자 데이터 형식인 경우 FORMAT을 사용하면 값 섹션에 차트와 같은 숫자 데이터 형식이 필요한 시각적 개체에서 측정값을 사용할 수 없습니다. Power BI에서 측정값에 동적 형식 문자열을 사용하여 측정값의 숫자 데이터 형식과 기본 조건부 서식 문자열을 지정할 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제

형식 문자열

DAX

```
= FORMAT( 12345.67, "General Number")
= FORMAT( 12345.67, "Currency")
= FORMAT( 12345.67, "Fixed")
= FORMAT( 12345.67, "Standard")
= FORMAT( 12345.67, "Percent")
= FORMAT( 12345.67, "Scientific")
```

HRESULT = NO_ERROR를

12345.67 "일반 번호"는 서식이 없는 숫자를 표시합니다.

\$12,345.67 "통화"는 통화 로캘 서식이 지정된 숫자를 표시합니다. 이 샘플에서는 기본 미국 통화 서식을 보여 드립니다.

12345.67 "고정"은 소수 구분 기호의 왼쪽에 적어도 한 자리, 소수 구분 기호 오른쪽에 두 자리 이상을 표시합니다.

12,345.67 "표준"은 소수 구분 기호의 왼쪽에 적어도 한 자리, 소수 구분 기호 오른쪽에 두 자리 이상을 표시하며 천 개의 구분 기호를 포함합니다. 이 샘플에서는 기본 미국 숫자 서식을 보여줍니다.

1,234,567.00% "백분율"은 숫자의 오른쪽에 서식을 지정하고 백분율 기호를 단일 공백으로 구분하여 백분율(100을 곱한 값)으로 표시합니다.

1.23E+04 "Scientific"은 10진수가 두 자리인 과학적 표기법으로 숫자를 표시합니다.

선택적 locale_name 있는 날짜/시간

DAX

```
= FORMAT(dt"2020-12-15T12:30:59", BLANK(), "en-US")
= FORMAT(dt"2020-12-15T12:30:59", BLANK(), "en-GB")
= FORMAT(dt"2020-12-15T12:30:59", "mm/dd/yyyy", "en-GB")
```

HRESULT = NO_ERROR를

2020년 12월 15일 오후 12:30:59 여기서 월은 12시간 형식입니다.

2020년 15월 12일 12:30:59 전월과 시간은 24시간 형식입니다.

2020년 12월 15일 12:30:59 이전 달과 시간은 24시간 형식입니다. 로캘이 아닌 종속 형식 문자열이 지정되었으므로 로캘이 적용되지 않고 로캘이 아닌 형식이 반환됩니다.

미리 정의된 숫자 형식

format_string 인수에 다음과 같이 미리 정의된 숫자 형식을 지정할 수 있습니다.

서식	설명
"General"	천 단위 구분 기호가 없는 숫자를 표시합니다.
"Number"	
"Currency"	1,000개의 구분 기호가 있는 숫자를 표시합니다(해당하는 경우). 는 소수 구분 기호 오른쪽에 두 숫자를 표시합니다. 출력은 시스템 로캘 설정에 따릅니다.
"Fixed"	소수 구분 기호 왼쪽으로 최소 한 자릿수, 오른쪽으로 두 자릿수를 표시합니다.
"Standard"	1000개의 구분 기호가 있는 숫자를 표시합니다. 왼쪽에는 한 자리 이상, 소수 구분 기호의 오른쪽에는 두 자리가 있습니다.

서식	설명
"Percent"	오른쪽에 바로 백분율 기호(%)가 추가된 100을 곱한 숫자를 표시합니다. 은 항상 소수 구분 기호 오른쪽에 두 자릿수를 표시합니다.
"Scientific"	표준 과학적 표기법을 사용하여 두 개의 유효 숫자를 제공합니다.
"Yes/No"	숫자가 0이면 아니요를 표시합니다. 그렇지 않으면 예를 표시합니다.
"True/False"	숫자가 0이면 False를 표시합니다. 그렇지 않으면 True를 표시합니다.
"On/Off"	숫자가 0이면 끄기, 그렇지 않으면 On을 표시합니다.

사용자 지정 숫자 형식

숫자에 대한 사용자 지정 형식 식은 세미콜론으로 구분된 1~3개의 섹션을 가질 수 있습니다. 형식 문자열 인수에 명명된 숫자 형식 중 하나가 포함된 경우 하나의 섹션만 허용됩니다.

사용하는 경우	결과
섹션 하나만	서식 식은 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.

DAX

```
"#,##0; (#,##0)"
```

세미콜론 사이에 아무 것도 포함하지 않으면 누락된 섹션은 양수 값의 형식을 사용하여 정의됩니다. 예를 들어 다음 형식은 첫 번째 섹션의 형식을 사용하여 양수 및 음수 값을 표시하고 값이 0이면 "Zero"를 표시합니다.

DAX

```
"#,##0"
```

세미콜론 사이에 아무 것도 포함하지 않으면 양수 값의 형식을 사용하여 누락된 섹션이 표시됩니다.

사용자 지정 숫자 서식 문자

`format_string` 인수에는 다음과 같은 사용자 지정 숫자 서식 문자를 지정할 수 있습니다.

문자	설명
None	형식을 지정하지 않고 숫자를 표시합니다.
(0)	숫자 자리 표시자. 숫자 또는 0을 표시합니다. 식에 서식 문자열에 0이 표시되는 위치에 숫자가 있으면 표시합니다. 그렇지 않으면 해당 위치에 0을 표시합니다. 숫자의 숫자가 형식 식의 0(10진수 양쪽)보다 적은 경우 선행 또는 후행 0을 표시합니다. 숫자의 소수 구분 기호 오른쪽에 있는 숫자가 서식 식의 소수 구분 기호 오른쪽에 있는 0보다 많은 경우 숫자를 0만큼 소수 자릿수로 반올림합니다. 숫자의 소수 구분 기호 왼쪽에 소수 구분 기호 왼쪽에 0보다 더 많은 숫자가 있는 경우 수정하지 않고 여분의 숫자를 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 0자리 자리 표시자처럼 작동합니다. 단, 숫자의 자릿수가 형식 식의 소수 구분 기호 양쪽에 있는 #문자보다 같거나 적은 경우 선행 및 후행 0이 표시되지 않는다는 점을 제외하면 이 기호는 0자리 자리 표시자처럼 작동합니다.
(.)	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. 소수 자릿수로 표시된 선행 0을 표시하려면 소수 구분 기호 왼쪽의 첫 번째 자리 표시자로 0을 사용합니다. 형식이 지정된 출력에서 소수 자리 표시자로 사용되는 실제 문자는 시스템에서 인식하는 숫자 형식에 따라 다릅니다.
(%)	백분율 자리 표시자. 식에 100을 곱합니다. 백분율 문자(%)는 서식 문자열에 표시되는 위치에 삽입됩니다.
(,)	천 단위 구분 기호입니다. 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 끊인 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,,”를 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 시간 구분 기호를 나타내는 데 다른 문자를 사용할 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

문자	설명
	지정된 출력에서 날짜 구분 기호로 사용되는 실제 문자는 시스템 설정에 따라 결정됩니다.
(E- E+ e- e+)	지수 형식. 형식 속에서 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 문자 중 하나 이외의 문자를 표시하려면 앞에 백슬래시(\)를 추가하거나 큰따옴표(" ")로 묶습니다.
\()	형식 문자열에서 다음 문자를 표시합니다. 특별한 의미가 있는 문자를 리터럴 문자로 표시하려면 앞에 백슬래시(\)를 줍니다. 백슬래시 자체는 표시되지 않습니다. 백슬래시를 사용하는 것은 다음 문자를 큰따옴표로 묶는 것과 같습니다. 백슬래시 자체를 표시하려면 백슬래시를 두 개(\ \) 사용하세요. 리터럴 문자로 표시할 수 없는 문자의 예로는 날짜 서식 및 시간 서식 문자(a, c, d, h, m, n, p, q, s, t, w, y, /, 및 :), 숫자 서식 문자(#, 0, %, E, e, 쉼표 및 마침표) 및 문자열 서식 문자(@, &, , <> 및 !)가 있습니다.
("ABC")	문자열을 큰따옴표(" ")로 묶어 표시합니다.

미리 정의된 날짜/시간 형식

format_string 인수에 다음과 같은 미리 정의된 날짜/시간 형식을 지정할 수 있습니다. 이 외의 형식을 사용하는 경우 사용자 지정 날짜/시간 형식으로 해석됩니다.

서식	설명
"General Date"	날짜 및/또는 시간을 표시합니다. 예를 들어 2008년 3월 12일 오전 11:07:31입니다. 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 수요일입니다.
"Short Date"	현재 문화권의 짧은 날짜 형식을 사용하여 날짜를 표시합니다. 예를 들어 2008년 3월 12일입니다.
"Long Time" 또는 "Medium Time"	현재 문화권의 긴 시간 형식을 사용하여 시간을 표시합니다. 일반적으로 시간, 분, 초를 포함합니다. 예를 들어 오전 11:07:31입니다.
"Short Time"	시간을 12시간 형식으로 표시합니다. 예를 들어 오전 11:07입니다.

사용자 지정 날짜/시간 형식

다음 형식 문자를 `format_string` 지정하여 사용자 지정 날짜/시간 형식을 만들 수 있습니다.

문자	설명
(:)	시간 구분 기호. 일부 로캘에서는 시간 구분 기호를 나타내는 데 다른 문자를 사용할 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.
(\)	백슬래시. 다음 문자를 리터럴 문자로 표시합니다. 따라서 서식 문자로 해석되지 않습니다.
("")	큰따옴표입니다. 큰따옴표 안에 묶인 텍스트가 표시됩니다. 따라서 서식 문자로 해석되지 않습니다.
c	날짜를 해당 순서대로 <code>ddddd</code> 표시하고 시간을 표시 <code>ttttt</code> 합니다. 날짜 일련 번호에 소수 부분이 없는 경우 날짜 정보만 표시합니다. 정수 부분이 없는 경우 시간 정보만 표시합니다.
d	선행 0(1-31)이 없는 숫자로 하루를 표시합니다.
dd	일 수를 앞에 0(01-31)으로 표시합니다.
ddd	요일을 약어(일-토)로 표시합니다. 지역화됨.
dddd	하루를 전체 이름으로 표시합니다(일요일-토요일). 지역화됨.
ddddd	날짜를 시스템의 짧은 날짜 형식 설정에 따라 서식이 지정된 전체 날짜(일, 월 및 연도 포함)로 표시합니다. 기본 간단한 날짜 형식은 <code>mm/dd/yyyy</code> 입니다.
ddyyyy	시스템에서 인식하는 긴 날짜 설정에 따라 날짜 일련 번호를 전체 날짜(일, 월 및 연도 포함)로 표시합니다. 기본 긴 날짜 형식은 <code>dddd, mmmm d, yyyy</code> 입니다.
w	요일을 숫자로 표시합니다(일요일의 경우 1부터 토요일의 경우 7까지).
ww	연도의 주를 숫자로 표시합니다(1-54).
분	월을 선행 0(1-12)이 없는 숫자로 표시합니다. <code>m</code> 바로 뒤에 <code>h</code> 오거나 <code>hh</code> 월이 아닌 분이 표시됩니다.
mm	월을 앞에 0(01-12)이 있는 숫자로 표시합니다. <code>mm</code> 바로 뒤에 <code>h</code> 오거나 <code>hh</code> 월이 아닌 분이 표시됩니다.
mmm	월을 약어(Jan-Dec)로 표시합니다. 지역화됨.

문자	설명
mmmm	월을 전체 월 이름으로 표시합니다(1월-12월). 지역화됨.
q	연도의 분기를 숫자(1-4)로 표시합니다.
y	연도의 날을 숫자로 표시합니다(1-366).
yy	연도를 2자리 숫자(00-99)로 표시합니다.
yyyy	연도를 4자리 숫자(100-9999)로 표시합니다.
시간	앞에 오는 0(0-23)이 없는 숫자로 시간을 표시합니다.
hh	시간을 앞에 0(00-23)이 있는 숫자로 표시합니다.
n	분을 앞에 오는 0(0-59)이 없는 숫자로 표시합니다.
nn	분을 앞에 오는 0(00-59)이 있는 숫자로 표시합니다.
s	앞에 오는 0(0-59)이 없는 숫자로 두 번째 값을 표시합니다.
ss	두 번째 숫자를 앞에 0(00-59)이 있는 숫자로 표시합니다.
ttttt	시간을 시스템에서 인식하는 시간 형식으로 정의된 시간 구분 기호를 사용하여 서식이 지정된 전체 시간(시간, 분 및 초 포함)으로 표시합니다. 앞에 오는 0 옵션이 선택되고 시간이 오전 10:00 이전인 경우 앞에 0이 표시됩니다. 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	12시간 시계를 사용하고 정오 이전의 시간에 소문자 AM을 표시합니다. 는 정오부터 오후 11시 59분까지의 시간에 소문자 PM을 표시합니다.
A/P	12시간 시계를 사용하고 정오 이전의 시간에 대문자 A를 표시합니다. 는 정오부터 오후 11시 59분까지의 시간에 대문자 P를 표시합니다.
a/p	12시간 시계를 사용하고 정오 이전의 시간에 소문자 A를 표시합니다. 정오부터 오후 11시 59분까지의 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 24시간 시계로 설정된 경우 문자열은 일반적으로 빈 문자열로 설정됩니다.

날짜/시간 서식은 현재 사용자 로캘을 사용하여 문자열의 서식을 지정합니다. 예를 들어 2020년 6월 25일 날짜를 고려합니다. 형식 문자열 "m/d/yyyy"를 사용하여 형식이 지정되면 다음과 같습니다.

- 사용자 로캘은 미국(en-US)의 미국: "2020년 6월 25일"

- 사용자 로캘은 독일(de-DE): "6.25.2020"

사용자 지정 날짜/시간 형식 예제

다음 예제에서는 2020년 6월 25일 목요일 오후 1:23:45에 날짜/시간을 사용합니다. 독일(de-DE)은 24시간 시스템을 사용합니다. AM/PM에 해당하는 항목은 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"	목요일	권장 사항
"ddd"	목요일	Donnerstag
"dddd"	06/25/2020	25.06.2020
"ddddd"	2020년 6월 25일 목요일	도너태그, 25세. Juni 2020
"w"	5	5
"ww"	26	26
"m"	6	6
"mm"	06	06
"mmm"	6월	6월
"mmmm"	6월	Juni
"q"	2	2
"y"	177	177
"yy"	20	20
"yyyy"	2020	2020
""Year"" yyyy"	2020년	2020년
"yyyy \Qq"	2020년 2분기	2020년 2분기
"dd/mm/yyyy"	25/06/2020	25.06.2020
"mm/dd/yyyy"	06/25/2020	06.25.2020

형식	결과(en-US)	결과(de-DE)
"h:nn:ss"	13:23:45	13:23:45
"h:nn:ss AMPM"	오후 1:23:45	1:23:45
"hh:nn:ss"	13:23:45	13:23:45
"hh:nn:ss AMPM"	오후 01:23:45	01:23:45
"tttt"	13:23:45	13:23:45
"tttt AMPM"	오후 13:23:45	13:23:45
"mm/dd/yyyy hh:nn:ss AMPM"	2020년 6월 25일 오후 01:23:45	6.25.2020 01:23:45

피드백

이 페이지가 도움이 되었나요?

Yes

No

LEFT

아티클 • 2023. 12. 14.

텍스트 문자열의 시작 부분부터 지정된 문자 수를 반환합니다.

구문

DAX

```
LEFT(<text>, <num_chars>)
```

매개 변수

 테이블 확장

용어	정의
text	추출할 문자를 포함하는 텍스트 문자열 또는 텍스트가 포함된 열에 대한 참조입니다.
num_chars	(선택 사항) LEFT에서 추출할 문자 수입니다. 생략하면 1입니다.

반환 값

텍스트 문자열입니다.

설명

- Microsoft Excel에는 싱글바이트 및 더블바이트 문자 언어의 텍스트 작업을 위한 다양한 함수가 포함되어 있지만 DAX는 유니코드에서 작동하며 모든 문자를 동일한 길이로 저장합니다. 따라서 단일 함수로 충분합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 예제에서는 [ResellerName] 열에 있는 회사 이름의 처음 다섯 문자와 [GeographyKey] 열에 있는 지리적 코드의 처음 다섯 글자를 반환하고 연결하여 식별자를 만듭니다.

DAX

```
= CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))
```

`num_chars` 인수가 사용 가능한 문자 수보다 큰 숫자인 경우 함수는 사용 가능한 최대 문자를 반환하고 오류를 발생시키지 않습니다. 예를 들어 [GeographyKey] 열에는 1, 12 및 311과 같은 숫자가 포함됩니다. 따라서 결과에는 가변 길이도 있습니다.

관련 콘텐츠

[텍스트 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LEN

아티클 • 2023. 10. 20.

텍스트 문자열의 문자 수를 반환합니다.

구문

DAX

`LEN(<text>)`

매개 변수

용어 정의

text 찾으려는 길이가 있는 텍스트 또는 텍스트가 포함된 열입니다. 공백은 문자로 계산됩니다.

반환 값

텍스트 문자열의 문자 수를 나타내는 정수입니다.

설명

- Microsoft Excel에는 싱글바이트 및 더블바이트 문자 언어를 사용하는 기능이 있지만 DAX는 유니코드를 사용하고 길이가 같은 모든 문자를 저장합니다.
- LEN은 기본 언어 설정에 관계없이 항상 각 문자를 1로 계산합니다.
- 날짜 또는 부울과 같은 텍스트가 아닌 값이 포함된 열에 LEN을 사용하는 경우 함수는 현재 열 형식을 사용하여 값을 텍스트로 암시적으로 캐스팅합니다.

예시

다음 수식은 [AddressLine1] 및 [AddressLine2] 열의 주소 길이를 합산합니다.

DAX

= `LEN([AddressLine1])+LEN([AddressLine2])`

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LOWER

아티클 • 2023. 12. 14.

텍스트 문자열의 모든 문자를 소문자로 변환합니다.

구문

DAX

`LOWER(<text>)`

매개 변수

 테이블 확장

용어 정의

text 소문자로 변환할 텍스트 또는 텍스트가 포함된 열에 대한 참조입니다.

반환 값

소문자로 된 텍스트입니다.

설명

문자가 아닌 문자는 변경되지 않습니다. 예를 들어 수식 `= LOWER("123ABC")` 은 123abc를 반환합니다.

예시

다음 수식은 [ProductCode] 열의 각 행을 가져오고 값을 모든 소문자로 변환합니다. 열의 숫자는 영향을 받지 않습니다.

DAX

`= LOWER('New Products'[ProductCode])`

관련 콘텐츠

텍스트 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

중간

아티클 • 2023. 10. 20.

시작 위치와 길이가 지정된 텍스트 문자열의 중간에서 문자 문자열을 반환합니다.

구문

DAX

```
MID(<text>, <start_num>, <num_chars>)
```

매개 변수

용어	정의
text	문자를 추출할 텍스트 문자열 또는 텍스트가 포함된 열입니다.
start_num	추출할 첫 번째 문자의 위치입니다. 위치는 1에서 시작합니다.
num_chars	반환할 문자 수입니다.

반환 값

지정된 길이의 텍스트 문자열입니다.

설명

Microsoft Excel에는 싱글바이트 및 더블바이트 문자 언어를 사용하는 기능이 다르지만 DAX는 유니코드를 사용하고 길이가 같은 모든 문자를 저장합니다.

예제

다음 식은

DAX

```
MID("abcde",2,3))
```

"bcd"를 반환합니다.

다음 식은

DAX

```
MID('Reseller'[ResellerName],1,5)
```

와 동일한 결과를 `LEFT([ResellerName],5)` 반환합니다. 두 식 모두 처음 5개의 열 `[ResellerName]` 문자를 반환합니다.

참고 항목

[텍스트 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

REPLACE

아티클 • 2023. 10. 20.

REPLACE는 지정한 문자 수에 따라 텍스트 문자열의 일부를 다른 텍스트 문자열로 바꿉니다.

구문

DAX

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

매개 변수

용어	정의
old_text	바꿀 문자가 포함된 텍스트 문자열 또는 텍스트가 포함된 열에 대한 참조입니다.
start_num	old_text new_text 바꿀 문자의 위치입니다.
num_chars	바꿀 문자 수입니다. 경고: num_chars 인수가 공백이거나 공백으로 계산되는 열을 참조하는 경우 new_text 문자열은 문자를 대체하지 않고 start_num 위치에 삽입됩니다. 이는 Excel에서와 동일한 동작입니다.
new_text	old_text 지정된 문자의 대체 텍스트입니다.

반환 값

텍스트 문자열입니다.

설명

- Microsoft Excel에는 싱글바이트 및 더블바이트 문자 언어에 사용할 수 있는 함수가 다르지만 DAX는 유니코드를 사용하므로 모든 문자를 동일한 길이로 저장합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 수식은 [ProductCode] 열에 있는 제품 코드의 처음 두 문자를 새 두 글자 코드인 OB로 바꾸는 새 계산 열을 만듭니다.

DAX

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

참고 항목

[텍스트 함수](#)

[SUBSTITUTE 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

REPT

아티클 • 2023. 12. 14.

지정된 횟수만큼 텍스트를 반복합니다. REPT를 사용하여 텍스트 문자열의 여러 인스턴스로 셀을 채웁니다.

구문

DAX

```
REPT(<text>, <num_times>)
```

매개 변수

 테이블 확장

용어	정의
text	반복하려는 텍스트입니다.
num_times	텍스트를 반복할 횟수를 지정하는 양수입니다.

반환 값

변경 내용을 포함하는 문자열입니다.

설명

- number_times 0이면 REPT는 공백을 반환합니다.
- number_times 정수가 아니면 잘립니다.
- REPT 함수의 결과는 32,767자를 초과할 수 없거나 REPT에서 오류를 반환합니다.

예: 리터럴 문자열 반복

다음 예제에서는 문자열 85를 세 번 반복하여 반환합니다.

DAX

```
= REPT("85",3)
```

예: 열 값 반복

다음 예제에서는 열 [MyText]의 문자열을 [MyNumber] 열의 횟수에 대해 반복하여 반환합니다. 수식이 전체 열에 대해 확장되므로 결과 문자열은 각 행의 텍스트 및 숫자 값에 따라 달라집니다.

DAX

```
= REPT([MyText],[MyNumber])
```

 테이블 확장

MyText	MyNumber	CalculatedColumn1
Text	2	TextText
숫자	0	
85	3	858585

관련 콘텐츠

[텍스트 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

RIGHT

아티클 • 2023. 12. 14.

RIGHT는 지정한 문자 수에 따라 텍스트 문자열의 마지막 문자 또는 문자를 반환합니다.

구문

DAX

```
RIGHT(<text>, <num_chars>)
```

매개 변수

 테이블 확장

용어	정의
text	추출할 문자가 포함된 텍스트 문자열 또는 텍스트가 포함된 열에 대한 참조입니다.
num_chars	(선택 사항) RIGHT에서 추출할 문자 수입니다. 은 생략되고 1입니다. 숫자가 포함된 열에 대한 참조를 사용할 수도 있습니다.

열 참조에 텍스트가 없으면 암시적으로 텍스트로 캐스팅됩니다.

반환 값

지정된 오른쪽 문자가 들어 있는 텍스트 문자열입니다.

설명

- RIGHT는 기본 언어 설정에 관계없이 항상 싱글 바이트 또는 더블 바이트 등 각 문자를 1로 계산합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예: 고정 문자 수 반환

다음 수식은 New Products 테이블에 있는 제품 코드의 마지막 두 숫자를 반환합니다.

DAX

```
= RIGHT('New Products'[ProductCode],2)
```

예: 열 참조를 사용하여 문자 수 지정

다음 수식은 MyCount 열의 수에 따라 새 제품 테이블의 제품 코드에서 가변 자릿수를 반환합니다. 열, MyCount에 값이 없거나 값이 비어 있으면 RIGHT도 공백을 반환합니다.

DAX

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

관련 콘텐츠

[텍스트 함수](#)

[LEFT](#)

[중간](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

검색...

아티클 • 2023. 10. 20.

왼쪽에서 오른쪽으로 읽는 특정 문자 또는 텍스트 문자열을 처음 찾은 문자의 수를 반환합니다. 검색은 대/소문자를 구분하지 않으며 악센트를 구분합니다.

구문

DAX

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

매개 변수

용어	정의
find_text	찾으려는 텍스트입니다. <code>find_text</code> 에 물음표(?), 별표(*) 등의 와일드카드 문자를 사용할 수 있습니다. 물음표는 모든 단일 문자와 일치합니다. 별표는 모든 문자 시퀀스와 일치합니다. 실제 물음표나 별표를 찾으려면 문자 앞에 타일(~)을 입력합니다.
within_text	<code>find_text</code> 검색하려는 텍스트 또는 텍스트가 포함된 열입니다.
start_num	(선택 사항) 검색을 시작할 <code>within_text</code> 문자 위치입니다. 생략하면 1입니다.
NotFoundValue	(선택 사항이지만 강력하게 권장됨) 작업에서 일치하는 substring을 찾을 수 없는 경우에 반환되는 값으로, 일반적으로 0, -1 또는 BLANK()입니다. 지정하지 않으면 오류가 반환됩니다.

반환 값

두 번째 텍스트 문자열의 첫 번째 문자에서 첫 번째 텍스트 문자열의 시작 위치 수입니다.

설명

- 검색 함수는 대/소문자를 구분하지 않습니다. "N"을 검색하면 'N' 또는 'n'이 처음 검색됩니다.
- 검색 함수는 악센트를 구분합니다. "á"를 검색하면 'á'가 처음 발생하지만 'a', 'à' 또는 대문자 버전 'A', 'Á'가 발생하지 않습니다.

- SEARCH 함수를 사용하여 다른 텍스트 문자열 내에서 문자 또는 텍스트 문자열의 위치를 확인한 다음 MID 함수를 사용하여 텍스트를 반환하거나 REPLACE 함수를 사용하여 텍스트를 변경할 수 있습니다.
- **within_text find_text** 찾을 수 없으면 수식이 오류를 반환합니다. 이 동작은 부분 문자열을 찾을 수 없는 경우 #VALUE 반환하는 Excel과 같습니다. **within_text Null**은 이 컨텍스트에서 빈 문자열로 해석됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는 재판매인 이름이 포함된 문자열에서 “cycle”이라는 첫 번째 글자의 위치를 찾습니다. 찾을 수 없으면 Blank가 반환됩니다.

SEARCH는 대/소문자를 구분하지 않습니다. 이 예에서는 <find_text> 인수에 “cycle” 또는 “Cycle”이 사용되는 경우 두 경우 모두에 대해 결과가 반환됩니다. 대/소문자를 구분하려면 [FIND](#)를 사용합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

DAX

```
EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of cycle", SEARCH ( "cycle", 'Reseller'[Reseller], 1,
        BLANK () )
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
    Reseller", "Warehouse" }
)
```

반환

재판매인	비즈니스 유형	cycle 위치
Volume Bike Sellers	창고	
Mass Market Bikes	부가 가치 재판매인	
Twin Cycles	부가 가치 재판매인	6
Rich Department Store	창고	

재판매인 Rental Gallery	비즈니스 유형 Specialty Bike Shop	cycle 위치
Budget Toy Store	창고	
Global Sports Outlet	창고	
Online Bike Catalog	창고	
Helmets and Cycles	부가 가치 재판매인	13
Jumbo Bikes	Specialty Bike Shop	

참고 항목

찾을

REPLACE

텍스트 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SUBSTITUTE

아티클 • 2023. 12. 14.

기존 텍스트를 텍스트 문자열의 새 텍스트로 대체합니다.

구문

DAX

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

매개 변수

 테이블 확장

용어	정의
text	문자를 대체하려는 텍스트 또는 텍스트가 포함된 열에 대한 참조입니다.
old_text	바꿀 기존 텍스트입니다.
new_text	old_text 바꿀 텍스트입니다.
instance_num	(선택 사항) 바꿀 old_text 발생합니다. 생략하면 모든 old_text 인스턴스가 대체됩니다.

반환 값

텍스트 문자열입니다.

설명

- 텍스트 문자열의 특정 텍스트를 바꾸려면 SUBSTITUTE 함수를 사용합니다. 텍스트 문자열의 특정 위치에서 발생하는 가변 길이의 텍스트를 바꾸려면 REPLACE 함수를 사용합니다.
- SUBSTITUTE 함수는 대/소문자를 구분합니다. 대/소문자를 텍스트와 old_text 일치하지 않으면 SUBSTITUTE에서 텍스트를 대체하지 않습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예: 문자열 내 대체

다음 수식은 열에서 발생하는 모든 위치에서 새 제품 코드 NW를 이전 제품 코드 PA로 대체하는 [Product Code] 열의 복사본을 만듭니다.

DAX

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

관련 콘텐츠

[텍스트 함수](#)

[REPLACE](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TRIM

아티클 • 2024. 01. 18.

단어 사이의 단일 공백을 제외하고 텍스트에서 모든 공백을 제거합니다.

구문

DAX

TRIM(<text>)

매개 변수

[] 테이블 확장

용어	정의
text	공백을 제거할 텍스트 또는 텍스트가 포함된 열입니다.

반환 값

공백이 제거된 문자열입니다.

설명

- 간격이 불규칙할 수 있는 다른 애플리케이션에서 받은 텍스트에 TRIM을 사용합니다.
- TRIM 함수는 원래 텍스트에서 7비트 ASCII 공백 문자(값 32)를 트리밍하도록 설계되었습니다. 유니코드 문자 집합에는 소수점 값이 160인 줄 바꿈하지 않는 공백 문자라는 추가 공백 문자가 있습니다. 이 문자는 일반적으로 웹 페이지에서 HTML 엔터 티 ()로 사용됩니다. TRIM 함수는 그 자체로 이 깨지지 않는 공백 문자를 제거하지 않습니다. 텍스트에서 공백 문자를 모두 자르는 방법의 예는 텍스트에서 공백 제거 및 인쇄되지 않는 문자를 참조하세요.

예시

다음 수식은 후행 공백이 없는 새 문자열을 만듭니다.

DAX

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

수식을 만들 때 수식을 입력한 것처럼 수식이 행을 통해 전파되므로 각 수식에 원래 문자열이 표시되고 결과가 표시되지 않습니다. 그러나 수식이 평가되면 문자열이 잘립니다.

수식이 다음과 같이 이전 수식에서 만든 계산 열의 길이를 검사 올바른 결과를 생성하는지 확인할 수 있습니다.

DAX

```
= LEN([Calculated Column 1])
```

관련 콘텐츠

[텍스트 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UNICHAR

아티클 • 2023. 10. 20.

숫자 값에서 참조하는 유니코드 문자를 반환합니다.

구문

DAX

`UNICHAR(number)`

매개 변수

용어	정의
번호	문자를 나타내는 유니코드 번호입니다.

반환 값

유니코드 번호로 표현되는 문자입니다.

설명

- XML 문자가 유효하지 않으면 UNICHAR에서 오류를 반환합니다.
- 유니코드 번호가 부분 서로게이트이고 데이터 형식이 유효하지 않으면 UNICHAR에서 오류를 반환합니다.
- 숫자가 허용 범위를 벗어나는 숫자 값이면 UNICHAR에서 오류를 반환합니다.
- number가 0이면 유니차르가 오류를 반환합니다.
- 반환되는 유니코드 문자는 문자 문자열(예: UTF-8 또는 UTF-16 코드)일 수 있습니다.

예시

다음 예제에서는 유니코드 번호 66(대문자 A)로 표시되는 문자를 반환합니다.

DAX

```
= UNICHAR(65)
```

다음 예제에서는 유니코드 번호 32(공백 문자)로 표시되는 문자를 반환합니다.

```
DAX
```

```
= UNICHAR(32)
```

다음 예제는 유니코드 숫자 9733(★ 문자)이 나타내는 문자를 반환합니다.

```
DAX
```

```
= UNICHAR(9733)
```

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UNICODE

아티클 • 2023. 10. 20.

텍스트의 첫 번째 문자에 해당하는 숫자(코드 포인트)를 반환합니다.

구문

DAX

```
UNICODE( <Text> )
```

매개 변수

용어	정의
Text	텍스트는 유니코드 값을 원하는 문자입니다.

반환 값

텍스트 문자열의 첫 번째 문자에 대한 숫자 코드입니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

UPPER

아티클 • 2023. 12. 14.

텍스트 문자열을 모든 대문자로 변환합니다.

구문

DAX

`UPPER (<text>)`

매개 변수

 테이블 확장

용어 정의

text 대문자로 변환할 텍스트 또는 텍스트가 포함된 열에 대한 참조입니다.

반환 값

대문자로 된 동일한 텍스트입니다.

예시

다음 수식은 열 [ProductCode]의 문자열을 모든 대문자로 변환합니다. 알파벳이 아닌 문자는 영향을 받지 않습니다.

DAX

`= UPPER(['New Products'[Product Code])`

관련 콘텐츠

[텍스트 함수](#)

[LOWER 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

값

아티클 • 2023. 10. 20.

숫자를 나타내는 텍스트 문자열을 숫자로 변환합니다.

구문

DAX

VALUE(<text>)

매개 변수

용어	정의
text	변환할 텍스트입니다.

반환 값

10진수 데이터 형식의 변환된 숫자입니다.

설명

- 텍스트 **매개 변수**로 전달된 값은 사용 중인 애플리케이션 또는 서비스에서 인식되는 상수, 숫자, 날짜 또는 시간 형식 중 어느 것일 수 있습니다. 텍스트가 이러한 형식 중 하나가 아니면 오류가 반환됩니다.
- 엔진이 필요에 따라 텍스트를 숫자로 암시적으로 변환하기 때문에 일반적으로 수식에서 VALUE 함수를 사용할 필요가 없습니다.
- 열 참조를 사용할 수도 있습니다. 예를 들어 혼합 숫자 형식을 포함하는 열이 있는 경우 VALUE를 사용하여 모든 값을 단일 숫자 데이터 형식으로 변환할 수 있습니다. 그러나 숫자와 텍스트가 혼합된 열에 VALUE 함수를 사용하는 경우 모든 행의 모든 값을 숫자로 변환할 수 없으므로 전체 열에 오류 플래그가 지정됩니다.

예시

다음 수식은 형식화된 문자열 "3"을 숫자 값 3으로 변환합니다.

DAX

```
= VALUE("3")
```

참고 항목

[텍스트 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

시간 인텔리전스 함수

아티클 • 2023. 10. 20.

DAX(데이터 분석 식)에는 일, 월, 분기 및 연도를 포함하여 기간을 사용하여 데이터를 조작한 다음 해당 기간에 대한 계산을 빌드하고 비교할 수 있는 시간 인텔리전스 함수가 포함되어 있습니다. 시간 인텔리전스 함수를 사용하기 전에 날짜 열이 포함된 테이블 중 하나를 날짜 테이블로 표시해야 합니다.

이 범주에서

함수	설명
CLOSINGBALANCEMONTH	현재 컨텍스트에서 월의 마지막 날짜에 있는 식을 평가합니다.
CLOSINGBALANCEQUARTER	현재 컨텍스트에서 분기의 마지막 날짜에 있는 식을 평가합니다.
CLOSINGBALANCEYEAR	현재 컨텍스트에서 연도의 마지막 날짜에 있는 식을 평가합니다.
DATEADD	현재 컨텍스트의 날짜에서 지정된 간격 수만큼 정방향 또는 뒤로 이동한 날짜 열이 포함된 테이블을 반환합니다.
DATESBETWEEN	지정된 시작 날짜로 시작하고 지정된 종료 날짜까지 계속되는 날짜 열이 포함된 테이블을 반환합니다.
DATESINPERIOD	지정된 시작 날짜로 시작하고 지정된 날짜 간격의 수와 형식에 대해 계속되는 날짜 열이 포함된 테이블을 반환합니다.
DATESMTD	현재 컨텍스트에서 현재까지의 월 날짜 열이 들어 있는 테이블을 반환합니다.
DATESQTD	현재 컨텍스트에서 현재까지의 분기 날짜 열이 들어 있는 테이블을 반환합니다.
DATESYTD	현재 컨텍스트에서 현재까지 연도의 날짜 열이 들어 있는 테이블을 반환합니다.
ENDOFMONTH	지정된 날짜 열의 현재 컨텍스트에서 월의 마지막 날짜를 반환합니다.
ENDOFQUARTER	지정된 날짜 열의 현재 컨텍스트에서 분기의 마지막 날짜를 반환합니다.
ENDOFYEAR	지정된 날짜 열의 현재 컨텍스트에서 연도의 마지막 날짜를 반환합니다.
FIRSTDATE	지정된 날짜 열에 대한 현재 컨텍스트의 첫 번째 날짜를 반환합니다.

함수	설명
FIRSTNONBLANK	식이 비어 있지 않은 현재 컨텍스트로 필터링된 열의 첫 번째 값을 반환합니다.
LASTDATE	지정된 날짜 열에 대한 현재 컨텍스트의 마지막 날짜를 반환합니다.
LASTNONBLANK	식이 비어 있지 않은 현재 컨텍스트로 필터링된 열의 마지막 값을 반환합니다.
다음 날	현재 컨텍스트의 날짜 열에 지정된 첫 번째 날짜를 기준으로 다음 날의 모든 날짜 열이 포함된 테이블을 반환합니다.
NEXTMONTH	현재 컨텍스트의 날짜 열에 있는 첫 번째 날짜를 기준으로 다음 달의 모든 날짜 열이 포함된 테이블을 반환합니다.
NEXTQUARTER	현재 컨텍스트에서 날짜 열에 지정된 첫 번째 날짜를 기준으로 다음 분기의 모든 날짜 열이 포함된 테이블을 반환합니다.
NEXTYEAR	현재 컨텍스트에서 날짜 열의 첫 번째 날짜를 기준으로 다음 연도의 모든 날짜 열이 들어 있는 테이블을 반환합니다.
OPENINGBALANCEMONTH	현재 컨텍스트에서 월의 첫 번째 날짜에 식을 평가합니다.
OPENINGBALANCEQUARTER	현재 컨텍스트에서 분기의 첫 번째 날짜에 식을 평가합니다.
OPENINGBALANCEYEAR	현재 컨텍스트에서 연도의 첫 번째 날짜에 식을 평가합니다.
PARALLELPERIOD	현재 컨텍스트에서 지정된 날짜 열의 날짜와 평행한 기간을 나타내는 날짜 열이 들어 있는 테이블을 반환하며, 날짜가 시간에서 앞으로 또는 시간을 거슬러 올라가는 간격의 개수를 이동했습니다.
이전 날짜	현재 컨텍스트에서 날짜 열의 첫 번째 날짜 이전 날짜를 나타내는 모든 날짜의 열이 들어 있는 테이블을 반환합니다.
PREVIOUSMONTH	현재 컨텍스트에서 날짜 열의 첫 번째 날짜를 기준으로 이전 달의 모든 날짜 열이 들어 있는 테이블을 반환합니다.
PREVIOUSQUARTER	현재 컨텍스트에서 날짜 열의 첫 번째 날짜를 기준으로 이전 분기의 모든 날짜 열이 들어 있는 테이블을 반환합니다.
PREVIOUSYEAR	현재 컨텍스트에서 날짜 열의 마지막 날짜가 지정된 경우 이전 연도의 모든 날짜 열이 포함된 테이블을 반환합니다.
SAMEPERIODLASTYEAR	현재 컨텍스트에서 지정된 날짜 열의 날짜에서 1년 뒤로 이동한 날짜 열이 들어 있는 테이블을 반환합니다.
STARTOFMONTH	지정된 날짜 열의 현재 컨텍스트에서 월의 첫 번째 날짜를 반환합니다.

함수	설명
STARTOFCQUARTER	지정된 날짜 열의 현재 컨텍스트에서 분기의 첫 번째 날짜를 반환합니다.
STARTOFCYEAR	지정된 날짜 열의 현재 컨텍스트에서 연도의 첫 번째 날짜를 반환합니다.
TOTALMTD	현재 컨텍스트에서 현재까지의 월 식 값을 평가합니다.
TOTALQTD	현재 컨텍스트에서 현재까지 분기의 날짜에 대한 식 값을 평가합니다.
TOTALYTD	현재 컨텍스트에서 식의 연간 값을 평가합니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CLOSINGBALANCEMONTH

아티클 • 2023. 12. 14.

현재 컨텍스트에서 월의 마지막 날짜에 있는 식을 평가합니다.

구문

DAX

`CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])`

매개 변수

 테이블 확장

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 월의 마지막 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

① 참고

부울 식에 대한 제약 조건은 CALCULATE 함수에 설명되어 있습니다.

① 참고

필터 식에는 CALCULATE 함수에 설명된 제한 사항이 있습니다.

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

예시

다음 샘플 수식은 제품 인벤토리의 '월말 인벤토리 값'을 계산하는 측정값을 만듭니다.

DAX

```
=  
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product  
Inventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEQUARTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

CLOSINGBALANCEQUARTER

아티클 • 2023. 12. 14.

현재 컨텍스트에서 분기의 마지막 날짜에 있는 식을 평가합니다.

구문

DAX

`CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])`

매개 변수

 테이블 확장

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 분기의 마지막 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

① 참고

부울 식에 대한 제약 조건은 CALCULATE 함수에 설명되어 있습니다.

① 참고

필터 식에는 CALCULATE 함수에 설명된 제한 사항이 있습니다.

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

예시

다음 샘플 수식은 제품 인벤토리의 '쿼터 엔드 인벤토리 값'을 계산하는 측정값을 만듭니다.

DAX

```
=  
CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEMONTH 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

CLOSINGBALANCEYEAR

아티클 • 2023. 12. 14.

현재 컨텍스트에서 연도의 마지막 날짜에 있는 식을 평가합니다.

구문

DAX

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

매개 변수

 테이블 확장

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) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

현재 컨텍스트에서 연도의 마지막 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- **year_end_date** 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- **date 인수는** 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.

- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

① 참고

부울 식에 대한 제약 조건은 CALCULATE 함수에 설명되어 있습니다.

① 참고

필터 식에는 CALCULATE 함수에 설명된 제한 사항이 있습니다.

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

예시

다음 샘플 수식은 제품 인벤토리의 'Year End Inventory Value'를 계산하는 측정값을 만듭니다.

DAX

```
=  
CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI  
nventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEYEAR 함수](#)

[CLOSINGBALANCEQUARTER 함수](#)

[CLOSINGBALANCEMONTH 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATEADD

아티클 • 2023. 10. 20.

현재 컨텍스트의 날짜에서 지정된 간격 수만큼 정방향 또는 뒤로 이동한 날짜 열이 포함된 테이블을 반환합니다.

구문

DAX

`DATEADD(<dates>, <number_of_intervals>, <interval>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
number_of_intervals	날짜에 추가하거나 뺄 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. 간격 값은 다음 <code>year</code> <code>quarter</code> <code>month</code> 중 하나일 수 있습니다. <code>day</code>

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

`date` 인수는 다음 중 어느 것이든 될 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

ⓘ 참고

부울 식에 대한 제약 조건은 `CALCULATE` 함수 항목에 설명되어 있습니다.

- `number_of_intervals` 지정한 숫자가 양수이면 날짜의 날짜가 시간 단위로 이동되고, 숫자가 음수이면 날짜의 날짜가 시간을 거슬러 올라갑니다.
- **간격** 매개 변수는 문자열 집합이 아닌 열거형이므로 값을 따옴표로 묶어서는 안 됩니다. 또한 다음 값을 사용할 때는 다음 `month quarter day` 값 `year`의 전체 철자를 입력해야 합니다.
- 결과 테이블에는 날짜 열에 **있는 날짜** 만 포함됩니다.
- 현재 컨텍스트의 날짜가 연속 간격을 형성하지 않으면 함수는 오류를 반환합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예제 - 날짜 집합 이동

다음 수식은 현재 컨텍스트에서 날짜 1년 전의 날짜를 계산합니다.

DAX

```
= DATEADD(DateTime[DateKey], -1, year)
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATESBETWEEN

아티클 • 2023. 10. 20.

지정된 시작 날짜로 시작하고 지정된 종료 날짜까지 계속되는 날짜 열이 포함된 테이블을 반환합니다.

이 함수는 CALCULATE [함수에 필터로 전달하는 데](#) 적합합니다. 식을 사용자 지정 날짜 범위별로 필터링하는 데 사용합니다.

① 참고

일, 월, 분기 또는 연도와 같은 표준 날짜 간격을 사용하는 경우 더 적합한 **DATESINPERIOD** 함수를 사용하는 것이 좋습니다.

구문

DAX

DATESBETWEEN(<Dates>, <StartDate>, <EndDate>)

매개 변수

용어	정의
날짜	날짜 열입니다.
StartDate	날짜 식입니다.
EndDate	날짜 식입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 가장 일반적인 사용 사례에서 **Dates**는 표시된 날짜 테이블의 날짜 열에 대한 참조입니다.

- **StartDate**가 BLANK이면 **StartDate**는 **Dates** 열에서 가장 이른 값이 됩니다.
- **EndDate**가 BLANK이면 **EndDate**는 **Dates** 열에서 가장 늦은 값이 됩니다.
- **StartDate** 및 **EndDate**로 사용되는 날짜는 포함됩니다. 따라서 예를 들어 **StartDate** 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 **StartDate** 값이 2016년 7월 1일이면 반환되는 테이블은 2017년 7월 1일부터 시작합니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 Sales 테이블 측정값 정의는 DATESBETWEEN 함수를 사용하여 LTD(Life-to-Date) 계산을 생성합니다. 수명-날짜는 시간 시작 이후 시간에 따른 측정값의 누적을 나타냅니다.

수식은 MAX 함수를 사용합니다. 이 함수는 필터 컨텍스트에 있는 최신 날짜를 반환합니다. 따라서 DATESBETWEEN 함수는 가장 이른 날짜부터 보고되는 최신 날짜까지의 날짜 테이블을 반환합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#)을 참조하세요.

DAX

```
Customers_LTD =
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    )
)
```

Date 테이블에 저장된 가장 빠른 날짜는 2017년 7월 1일입니다. 따라서 보고서가 2020년 6월까지 측정값을 필터링하면 DATESBETWEEN 함수는 2017년 7월 1일부터 2020년 6월 30일까지 날짜 범위를 반환합니다.

참고 항목

- DAX(시간 인텔리전스 함수)
 - 날짜 및 시간 함수(DAX)
 - DATESINPERIOD 함수(DAX)
-

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATESINPERIOD

아티클 • 2023. 10. 20.

지정된 시작 날짜로 시작하고 지정된 날짜 간격의 수와 형식에 대해 계속되는 날짜 열이 포함된 테이블을 반환합니다.

이 함수는 CALCULATE [함수에 필터로 전달하는 데](#) 적합합니다. 식을 사용하여 일, 월, 분기 또는 연도와 같은 표준 날짜 간격을 기준으로 식을 필터링합니다.

구문

DAX

```
DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)
```

매개 변수

용어	정의
날짜	날짜 열입니다.
start_date	날짜 식입니다.
number_of_intervals	날짜를 추가하거나 뺄 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. 간격 값은 다음 DAY MONTH QUARTER 중 하나일 수 있습니다. YEAR

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 가장 일반적인 사용 사례에서 **날짜**는 표시된 날짜 테이블의 날짜 열에 대한 참조입니다.
- number_of_intervals 지정한 숫자가 양수이면 날짜가 정시에 앞으로 이동되고, 숫자가 음수이면 날짜가 시간에서 뒤로 이동합니다.

- **interval** 매개 변수는 열거형입니다. 유효한 값은 **DAY**, **MONTH**, **QUARTER** 및 **YEAR**입니다. 열거형이므로 값이 문자열로 전달되지 않습니다. 따라서 따옴표 안에 묶지 마십시오.
- 반환된 테이블에는 날짜 열에 **저장된 날짜** 만 포함될 수 있습니다. 예를 들어 날짜 **열이** 2017년 7월 1일부터 시작되고 **start_date** 값이 2016년 7월 1일인 경우 반환된 테이블은 2017년 7월 1일부터 시작됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 Sales 테이블 측정값 정의는 DATESINPERIOD 함수를 사용하여 이전 연도(PY)의 수익을 계산합니다.

수식은 MAX **함수를** 사용합니다. 이 함수는 필터 컨텍스트에 있는 최신 날짜를 반환합니다. 따라서 DATESINPERIOD 함수는 지난 해의 최신 날짜부터 시작하는 날짜 테이블을 반환합니다.

이 문서의 예는 샘플 Adventure Works DW 2020 Power BI Desktop 모델과 함께 사용할 수 있습니다. 모델을 가져오려면 [DAX 샘플 모델](#) 을 참조하세요.

DAX

```
Revenue PY =
CALCULATE(
    SUM(Sales[Sales Amount]),
    DATESINPERIOD(
        'Date'[Date],
        MAX('Date'[Date]),
        -1,
        YEAR
    )
)
```

보고서는 2020년 6월로 필터링됩니다. MAX 함수는 2020년 6월 30일을 반환합니다. 그런 다음 DATESINPERIOD 함수는 2019년 7월 1일부터 2020년 6월 30일까지 날짜 범위를 반환합니다. 지난 해인 2020년 6월 30일부터 날짜 값의 1년입니다.

참고 항목

[DAX\(시간 인텔리전스 함수\)](#)
[날짜 및 시간 함수\(DAX\)](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DATESMTD

아티클 • 2023. 12. 14.

현재 컨텍스트에서 현재까지의 월 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

`DATESMTD(<dates>)`

매개 변수

 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

date 인수는 다음 중 어느 것이든 될 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

ⓘ 참고

부울 식에 대한 제약 조건은 CALCULATE 함수 항목에 설명되어 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '월-날짜 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESMTD(DateTime[DateKey]))
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATESYTD 함수](#)

[DATESQTD 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

DATESQTD

아티클 • 2023. 12. 14.

현재 컨텍스트에서 현재까지의 분기 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

`DATESQTD(<dates>)`

매개 변수

 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

date 인수는 다음 중 어느 것이든 될 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

ⓘ 참고

부울 식에 대한 제약 조건은 CALCULATE 함수 항목에 설명되어 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '분기별 실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESQTD(DateTime[DateKey]))
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATESYTD 함수](#)

[DATESMTD 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

DATESYTD

아티클 • 2023. 10. 20.

현재 컨텍스트에서 현재까지 연도의 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

```
DATESYTD(<dates> [,<year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

date 인수는 다음 중 어느 것이든 될 수 있습니다.

- 날짜/시간 열에 대한 참조입니다.
- 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
- 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

① 참고

부울 식에 대한 제약 조건은 CALCULATE 함수 항목에 설명되어 있습니다.

- year_end_date 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다. 로캘에 따라 형

식은 "m-dd" 또는 "dd-m"와 같을 수 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
DATESYTD(DateTime[DateKey]))
```

다음 샘플 수식은 날짜 형식에 미국 로캘을 사용하여 인터넷 판매의 '회계 연도 실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(  
    SUM(InternetSales_USD[SalesAmount_USD]),  
    DATESYTD(DateTime[DateKey],  
        "6-30"  
    )  
)
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATESMTD 함수](#)

[DATESQTD 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ENDOFMONTH

아티클 • 2023. 12. 14.

지정된 날짜 열의 현재 컨텍스트에서 월의 마지막 날짜를 반환합니다.

구문

DAX

`ENDOFMONTH(<dates>)`

매개 변수

 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다.](#)
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대해 월말을 반환하는 측정값을 만듭니다.

DAX

= [ENDOFMONTH](#)(DateTime[DateKey])

관련 콘텐츠

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFYEAR 함수](#)

[ENDOFQUARTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ENDOFQUARTER

아티클 • 2023. 10. 20.

지정된 날짜 열의 현재 컨텍스트에서 분기의 마지막 날짜를 반환합니다.

구문

DAX

`ENDOFQUARTER(<dates>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대한 분기의 끝을 반환하는 측정값을 만듭니다.

DAX

```
= ENDOFQUARTER(DateTime[DateKey])
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFYEAR 함수](#)

[ENDOFMONTH 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ENDOFTIME

아티클 • 2023. 10. 20.

지정된 날짜 열의 현재 컨텍스트에서 연도의 마지막 날짜를 반환합니다.

구문

DAX

```
ENDOFTIME(<dates> [,<year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- year_end_date 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대해 6월 30일에 끝나는 회계 연도의 끝을 반환하는 측정값을 만듭니다.

DAX

```
= ENDOFYEAR(DateTime[DateKey], "06/30/2004")
```

참고 항목

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[ENDOFMONTH 함수](#)

[ENDOFQUARTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

FIRSTDATE

아티클 • 2023. 12. 14.

지정된 날짜 열에 대한 현재 컨텍스트의 첫 번째 날짜를 반환합니다.

구문

DAX

`FIRSTDATE(<dates>)`

매개 변수

[] 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 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 인수에 테이블이 필요한 모든 함수의 인수로 사용할 수 있습니다. 또한 날짜 값이 필요할 때마다 반환된 값을 사용할 수 있습니다.

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

예시

다음 샘플 수식은 현재 컨텍스트에 대한 인터넷 판매 채널에서 판매가 이루어진 첫 번째 날짜를 가져오는 측정값을 만듭니다.

DAX

```
= FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

관련 콘텐츠

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[LASTDATE 함수](#)

[FIRSTNONBLANK 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FIRSTNONBLANK

아티클 • 2023. 12. 14.

식이 비어 있지 않은 현재 컨텍스트로 필터링된 열의 첫 번째 값을 반환합니다.

구문

DAX

`FIRSTNONBLANK(<column>, <expression>)`

매개 변수

 테이블 확장

용어	정의
column	열 식입니다.
식	열의 각 값에 대해 공백으로 계산되는 식입니다.

반환 값

계산된 첫 번째 값이 있는 단일 열과 단일 행이 포함된 테이블입니다.

설명

- 열 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 [CALCULATE 함수 항목](#)에 설명되어 있습니다.
- 이 함수는 일반적으로 식이 비어 있지 않은 열의 첫 번째 값을 반환하는 데 사용됩니다. 예를 들어 제품의 판매가 있었던 마지막 값을 가져올 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

LASTNONBLANK 함수
통계 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

FIRSTNONBLANKVALUE

아티클 • 2023. 10. 20.

열의 정렬된 값으로 필터링된 식을 평가하고 비어 있지 않은 식의 첫 번째 값을 반환합니다.

구문

DAX

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

매개 변수

용어	정의
column	단일 열 테이블을 반환하는 열 또는 식입니다.
식	<열>의 값마다 공백에 대해 평가되는 식입니다.

반환 값

<열>의 정렬된 값에 해당하는 <식>의 첫 번째 비어 있지 않은 값입니다.

설명

- 열 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 이 함수는 <식> 평가를 위해 <열>이 필터 컨텍스트에 추가된다는 점에서 FIRSTNONBLANK와는 다릅니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는

DAX

```
EVALUATE  
SUMMARIZECOLUMNS(  
    DimProduct[Class],  
    "FNBV",  
    FIRSTNONBLANKVALUE(  
        DimDate[Date],  
        SUM(FactInternetSales[SalesAmount])  
    )  
)
```

반환

DimProduct[클래스]	[FNBV]
L	699.0982
H	13778.24
M	1000.4375
	533.83

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LASTDATE

아티클 • 2023. 12. 14.

지정된 날짜 열에 대한 현재 컨텍스트의 마지막 날짜를 반환합니다.

구문

DAX

`LASTDATE(<dates>)`

매개 변수

 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다.](#)
- 현재 컨텍스트가 단일 날짜인 경우 FIRSTDATE 및 LASTDATE 함수에서 반환되는 날짜는 동일합니다.
- 기술적으로 Return 값은 단일 열과 단일 값을 포함하는 테이블입니다. 따라서 이 함수는 인수에 테이블이 필요한 모든 함수의 인수로 사용할 수 있습니다. 또한 날짜 값이 필요할 때마다 반환된 값을 사용할 수 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매 채널에서 판매가 이루어진 현재 컨텍스트에 대한 마지막 날짜를 가져오는 측정값을 만듭니다.

DAX

```
= LASTDATE('InternetSales_USD'[SaleDateKey])
```

관련 콘텐츠

[날짜 및 시간 함수](#)

[시간 인텔리전스 함수](#)

[FIRSTDATE 함수](#)

[LASTNONBLANK 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LASTNONBLANK

아티클 • 2023. 12. 14.

식이 비어 있지 않은 현재 컨텍스트로 필터링된 열의 마지막 값을 반환합니다.

구문

DAX

`LASTNONBLANK(<column>, <expression>)`

매개 변수

 테이블 확장

용어	정의
column	열 식입니다.
식	열의 각 값에 대해 공백으로 계산되는 식입니다.

반환 값

계산된 마지막 값이 있는 단일 열과 단일 행이 포함된 테이블입니다.

설명

- 열 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
 - 단일 열 테이블을 정의하는 부울 식
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 이 함수는 일반적으로 식이 비어 있지 않은 열의 마지막 값을 반환하는 데 사용됩니다. 예를 들어 제품의 판매가 있었던 마지막 값을 가져올 수 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

관련 콘텐츠

FIRSTNONBLANK 함수

통계 함수

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

LASTNONBLANKVALUE

아티클 • 2023. 10. 20.

열의 정렬된 값으로 필터링된 식을 평가하고 비어 있지 않은 식의 마지막 값을 반환합니다.

구문

DAX

```
LASTNONBLANKVALUE(<column>, <expression>)
```

매개 변수

용어	정의
column	단일 열 테이블을 반환하는 열 또는 식입니다.
식	<열>의 값마다 공백에 대해 평가되는 식입니다.

반환 값

<열>의 정렬된 값에 해당하는 <식>의 마지막 비어 있지 않은 값입니다.

설명

- 열 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 열에 대한 참조입니다.
 - 단일 열이 있는 테이블입니다.
- 이 함수는 <식> 평가를 위해 <열>이 필터 컨텍스트에 추가된다는 점에서 LASTNONBLANK라는 다릅니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 DAX 쿼리는

DAX

```
EVALUATE  
SUMMARIZECOLUMNS(  
    DimProduct[Class],  
    "LNBV",  
    LASTNONBLANKVALUE(  
        DimDate[Date],  
        SUM(FactInternetSales[SalesAmount]))  
    )  
)
```

반환

DimProduct[클래스]	[LNBV]
L	132.44
H	137.6
M	84.97
	2288.6

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NEXTDAY

아티클 • 2023. 12. 14.

현재 컨텍스트의 날짜 열에 지정된 첫 번째 날짜를 기준으로 다음 날의 모든 날짜 **열이 포함된 테이블을 반환합니다.**

구문

DAX

NEXTDAY(<dates>)

매개 변수

[+] 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 다음 날부터 첫 번째 날짜까지의 모든 날짜를 반환합니다. 예를 들어 date 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하는 경우 이 함수는 2009년 6월 11일과 동일한 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다.](#)
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '다음 날 판매'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTDAY('DateTime'[DateKey]))
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTQUARTER 함수](#)

[NEXTMONTH 함수](#)

[NEXTYEAR 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NEXTMONTH

아티클 • 2023. 12. 14.

현재 컨텍스트의 날짜 열에 있는 첫 번째 날짜를 기준으로 다음 달의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

DAX

NEXTMONTH(<dates>)

매개 변수

[+] 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 다음 날부터 첫 번째 날짜까지의 모든 날짜를 반환합니다. 예를 들어 date 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하는 경우 이 함수는 2009년 7월의 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다.](#)
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '다음 달 판매'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTMONTH('DateTime'[DateKey]))
```

관련 콘텐츠

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTDAY 함수](#)

[NEXTQUARTER 함수](#)

[NEXTYEAR 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NEXTQUARTER

아티클 • 2023. 10. 20.

현재 컨텍스트에서 날짜 열에 지정된 첫 번째 날짜를 기준으로 다음 분기의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

DAX

`NEXTQUARTER(<dates>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 첫 번째 날짜를 기준으로 다음 분기의 모든 날짜를 반환합니다. 예를 들어 날짜 열의 첫 번째 날짜가 2009년 6월 10일을 참조하는 경우 이 함수는 2009년 7월부터 9월까지의 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '다음 분기 매출'을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTQUARTER('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTDAY 함수](#)

[NEXTMONTH 함수](#)

[NEXTYEAR 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

NEXTYEAR

아티클 • 2023. 10. 20.

현재 컨텍스트에서 날짜 열의 첫 번째 날짜를 기준으로 다음 연도의 모든 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

```
NEXTYEAR(<dates>[,<year_end_date>])
```

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 열의 첫 번째 날짜를 기준으로 다음 해의 모든 날짜를 반환합니다. 예를 들어 날짜 열의 첫 번째 날짜가 2007년을 참조하는 경우 이 함수는 2008년의 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- year_end_date 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '내년 매출'을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
NEXTYEAR('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[NEXTDAY 함수](#)

[NEXTQUARTER 함수](#)

[NEXTMONTH 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

OPENINGBALANCEMONTH

아티클 • 2023. 12. 14.

현재 컨텍스트에서 월의 첫 번째 날짜에 식을 평가합니다.

구문

DAX

`OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])`

매개 변수

 테이블 확장

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 월의 첫 번째 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 필터 식에는 항목인 [CALCULATE 함수](#)에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 제품 인벤토리의 '월 시작 인벤토리 값'을 계산하는 측정값을 만듭니다.

DAX

=

```
OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product  
Inventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[OPENINGBALANCEYEAR 함수](#)

[OPENINGBALANCEQUARTER 함수](#)

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEMONTH 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

OPENINGBALANCEQUARTER

아티클 • 2023. 12. 14.

현재 컨텍스트에서 분기의 첫 번째 날짜에 식을 평가합니다.

구문

DAX

`OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])`

매개 변수

 테이블 확장

용어	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter*	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

현재 컨텍스트에서 분기의 첫 번째 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 필터 식에는 항목인 [CALCULATE 함수](#)에 설명된 제한 사항이 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 제품 인벤토리의 '분기 시작 인벤토리 값'을 계산하는 측정값을 만듭니다.

DAX

=

```
OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[OPENINGBALANCEYEAR 함수](#)

[OPENINGBALANCEMONTH 함수](#)

[시간 인텔리전스 함수](#)

[CLOSINGBALANCEQUARTER 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

OPENINGBALANCEYEAR

아티클 • 2023. 12. 14.

현재 컨텍스트에서 연도의 첫 번째 날짜에 식을 평가합니다.

구문

DAX

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

매개 변수

 테이블 확장

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) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

현재 컨텍스트에서 연도의 첫 번째 날짜에 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목에 설명되어 있습니다.
- 필터 식에는 항목인 CALCULATE 함수에 설명된 제한 사항이 있습니다.

- `year_end_date` 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 제품 인벤토리의 '연도 시작 인벤토리 값'을 계산하는 측정값을 만듭니다.

DAX

```
=  
OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI  
nventory[UnitsBalance]),DateTime[DateKey])
```

관련 콘텐츠

[OPENINGBALANCEQUARTER](#) 함수

[OPENINGBALANCEMONTH](#) 함수

[시간 인텔리전스](#) 함수

[CLOSINGBALANCEYEAR](#) 함수

피드백

이 페이지가 도움이 되었나요?

Yes

No

PARALLELPERIOD

아티클 • 2023. 10. 20.

현재 컨텍스트에서 지정된 날짜 열의 날짜와 평행한 기간을 나타내는 날짜 열이 들어 있는 테이블을 반환하며, 날짜가 시간에서 앞으로 또는 시간을 거슬러 올라가는 간격의 개수를 이동했습니다.

구문

DAX

`PARALLELPERIOD(<dates>, <number_of_intervals>, <interval>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
number_of_intervals	날짜에 추가하거나 뺄 간격 수를 지정하는 정수입니다.
interval	날짜를 이동할 간격입니다. 간격 값은 다음 <code>year</code> <code>quarter</code> <code>month</code> 중 하나일 수 있습니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 날짜로 지정된 열의 현재 날짜 집합을 사용하고, 첫 번째 날짜와 마지막 날짜를 지정된 간격 수로 이동한 다음, 이동된 두 날짜 사이의 모든 연속 날짜를 반환합니다. 간격이 월, 분기 또는 연도의 부분 범위인 경우 결과의 부분 개월도 채워져 전체 간격을 완료합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.

- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다.](#)
- number_of_intervals 지정한 숫자가 양수이면 날짜의 날짜가 시간 단위로 이동되고, 숫자가 음수이면 날짜의 날짜가 시간을 거슬러 올라갑니다.
- **간격** 매개 변수는 문자열 집합이 아닌 열거형이므로 값을 따옴표로 묶어서는 안 됩니다. 또한 값 : `year`, `quarter` 를 `month` 사용할 때 전체 철자가 지정되어야 합니다.
- 결과 테이블에는 기본 테이블 열의 값에 표시되는 날짜만 포함됩니다.
- PARALLELPERIOD 함수는 DATEADD가 반환하는 부분 마침표 대신 지정된 세분성 수준에서 항상 전체 기간을 반환한다는 점을 제외하고 DATEADD 함수와 유사합니다. 예를 들어 6월 10일부터 같은 해 6월 21일로 끝나는 날짜가 선택되어 있고 해당 선택 영역을 한 달 앞으로 이동하려는 경우 PARALLELPERIOD 함수는 다음 달(7월 1일부터 7월 31일까지)의 모든 날짜를 반환합니다. 그러나 DATEADD를 대신 사용하는 경우 결과에는 7월 10일부터 7월 21일까지의 날짜만 포함됩니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 이전 연도 매출을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PARALLELPERIOD(DateTime[DateKey], -1, year))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[DATEADD 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PREVIOUSDAY

아티클 • 2023. 10. 20.

현재 컨텍스트에서 날짜 열의 첫 번째 날짜 이전 날짜를 나타내는 모든 날짜의 열이 들어 있는 테이블을 반환합니다.

구문

DAX

PREVIOUSDAY(<dates>)

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 첫 번째 날짜를 확인한 다음, 첫 번째 날짜 이전 날짜에 해당하는 모든 날짜를 반환합니다. 예를 들어 date 인수의 첫 번째 날짜가 2009년 6월 10일을 참조하는 경우 이 함수는 2009년 6월 9일과 동일한 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 함수 항목 [에 설명되어 있습니다](#).
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '전날 판매'를 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSDAY('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSMONTH 함수](#)

[PREVIOUSQUARTER 함수](#)

[PREVIOUSYEAR 함수](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PREVIOUSMONTH

아티클 • 2023. 10. 20.

현재 컨텍스트에서 **Dates** 열의 첫 번째 날짜를 기준으로 이전 달의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

DAX

PREVIOUSMONTH(<Dates>)

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력으로 사용된 열의 첫 번째 날짜를 사용하여 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 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '이전 달 판매액'을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSMONTH('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[이전 날짜](#)

[PREVIOUSQUARTER](#)

[PREVIOUSYEAR](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PREVIOUSQUARTER

아티클 • 2023. 10. 20.

현재 컨텍스트에서 날짜 열의 첫 번째 날짜를 기준으로 이전 분기의 모든 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

PREVIOUSQUARTER(<dates>)

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열을 포함하는 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.
- **date** 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 인터넷 판매의 '이전 분기 매출'을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSQUARTER('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSMONTH](#)

[이전 날짜](#)

[PREVIOUSYEAR](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

PREVIOUSYEAR

아티클 • 2023. 10. 20.

현재 컨텍스트에서 날짜 열의 마지막 날짜가 지정된 경우 이전 연도의 모든 날짜 열이 포함된 테이블을 반환합니다.

구문

DAX

`PREVIOUSYEAR(<dates>[,<year_end_date>])`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
year_end_date	(선택 사항) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

날짜 값의 단일 열을 포함하는 테이블입니다.

설명

- 이 함수는 입력 매개 변수의 최신 날짜가 지정된 경우 이전 연도의 모든 날짜를 반환합니다. 예를 들어 date 인수의 최신 날짜가 2009년을 참조하는 경우 이 함수는 지정된 year_end_date 2008년의 모든 날짜를 반환합니다.
- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- year_end_date 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 무시됩니다.

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

예시

다음 샘플 수식은 인터넷 판매의 이전 연도 매출을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),  
PREVIOUSYEAR('DateTime'[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSMONTH](#)

[이전 날짜](#)

[PREVIOUSQUARTER](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

SAMEPERIODLASTYEAR

아티클 • 2023. 10. 20.

현재 컨텍스트에서 지정된 날짜 열의 날짜에서 1년 뒤로 이동한 날짜 열이 들어 있는 테이블을 반환합니다.

구문

DAX

`SAMEPERIODLASTYEAR(<dates>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

날짜 값의 단일 열 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 반환되는 날짜는 이 동일한 수식에서 반환된 날짜와 동일합니다. `DATEADD(dates, -1, year)`
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 재판매인 판매의 이전 연도 매출을 계산하는 측정값을 만듭니다.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),  
SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

참고 항목

[시간 인텔리전스 함수](#)

[날짜 및 시간 함수](#)

[PREVIOUSYEAR](#)

[PARALLELPERIOD](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

STARTOFTMONTH

아티클 • 2023. 12. 14.

지정된 날짜 열의 현재 컨텍스트에서 월의 첫 번째 날짜를 반환합니다.

구문

DAX

`STARTOFTMONTH(<dates>)`

매개 변수

 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대한 월의 시작을 반환하는 측정값을 만듭니다.

DAX

= STARTOFMONTH(DateTime[DateKey])

관련 콘텐츠

날짜 및 시간 함수

시간 인텔리전스 함수

STARTOFTYEAR

STARTOFQUARTER

피드백

이 페이지가 도움이 되었나요?

Yes

No

STARTOFQUARTER

아티클 • 2023. 12. 14.

지정된 날짜 열의 현재 컨텍스트에서 분기의 첫 번째 날짜를 반환합니다.

구문

DAX

`STARTOFQUARTER(<dates>)`

매개 변수

[+] 테이블 확장

용어	정의
날짜	날짜가 포함된 열입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대해 분기의 시작을 반환하는 측정값을 만듭니다.

DAX

= STARTOFCQUARTER(DateTime[DateKey])

관련 콘텐츠

날짜 및 시간 함수

시간 인텔리전스 함수

STARTOFCYEAR

STARTOFCMONTH

피드백

이 페이지가 도움이 되었나요?

Yes

No

STARTOFTYEAR

아티클 • 2023. 10. 20.

지정된 날짜 열의 현재 컨텍스트에서 연도의 첫 번째 날짜를 반환합니다.

구문

DAX

`STARTOFTYEAR(<dates>)`

매개 변수

용어	정의
날짜	날짜가 포함된 열입니다.
YearEndDate	(선택 사항) 연도 종료 날짜 값입니다.

반환 값

단일 열과 날짜 값이 있는 단일 행이 포함된 테이블입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 이 함수는 계산 열 또는 RLS(행 수준 보안) 규칙에서 사용되는 경우 DirectQuery 모드에서 사용할 수 없습니다.

예시

다음 샘플 수식은 현재 컨텍스트에 대해 연초의 시작을 반환하는 측정값을 만듭니다.

DAX

= STARTOFTIME(DateTime[DateKey])

참고 항목

날짜 및 시간 함수

시간 인텔리전스 함수

STARTOFTIME

STARTOFMONTH

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TOTALMTD

아티클 • 2023. 12. 14.

현재 컨텍스트에서 현재까지의 월 식 값을 평가합니다.

구문

DAX

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

매개 변수

 테이블 확장

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

날짜의 날짜가 지정된 경우 현재 월-날짜의 날짜에 대해 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 필터 식에는 CALCULATE 항목에 설명된 제한 사항이 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '월 실행 합계' 또는 '월 실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

관련 콘텐츠

[ALL](#)

[계산](#)

[TOTALYTD](#)

[TOTALQTD](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TOTALQTD

아티클 • 2023. 12. 14.

현재 컨텍스트에서 현재까지 분기의 날짜에 대한 식 값을 평가합니다.

구문

DAX

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

매개 변수

 테이블 확장

매개 변수	정의
식	스칼라 값을 반환하는 식입니다.
날짜	날짜가 포함된 열입니다.
filter	(선택 사항) 현재 컨텍스트에 적용할 필터를 지정하는 식입니다.

반환 값

날짜의 날짜가 지정된 현재 분기의 모든 날짜에 대해 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 필터 식에는 CALCULATE 항목에 설명된 제한 사항이 있습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '분기 실행 합계' 또는 '분기 실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

관련 콘텐츠

[ALL](#)

[계산](#)

[TOTALYTD](#)

[TOTALMTD](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

TOTALYTD

아티클 • 2023. 12. 14.

현재 컨텍스트에서 식의 연간 값을 평가합니다.

구문

DAX

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

매개 변수

[\[+\] 테이블 확장](#)

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) 연말 날짜를 정의하는 날짜가 있는 리터럴 문자열입니다. 기본값은 12 월 31일입니다.

반환 값

현재 연도별 날짜에 대해 계산된 식을 나타내는 스칼라 값입니다.

설명

- date 인수는 다음 중 어느 것이든 될 수 있습니다.
 - 날짜/시간 열에 대한 참조입니다.
 - 날짜/시간 값의 단일 열을 반환하는 테이블 식입니다.
 - 날짜/시간 값의 단일 열 테이블을 정의하는 부울 식입니다.
- 부울 식에 대한 제약 조건은 CALCULATE 항목에 설명되어 있습니다.
- 필터 식에는 CALCULATE 항목에 설명된 제한 사항이 있습니다.

- `year_end_date` 매개 변수는 통합 문서를 만든 클라이언트의 로캘과 동일한 로캘에 있는 날짜의 문자열 리터럴입니다. 날짜의 연도 부분은 필요하지 않으며 무시됩니다. 예를 들어 다음 수식은 EN-US 로캘 통합 문서에서 6/30의 (회계) `year_end_date` 지정합니다.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey],
ALL('DateTime'), "6/30")
```

이 예제에서 `year_end_date` "6/30", "Jun 30", "6월 30일" 또는 월/일로 확인되는 문자열로 지정할 수 있습니다. 그러나 문자열이 날짜로 확인되도록 "월/일"(표시된 대로)을 사용하여 `year_end_date` 지정하는 것이 좋습니다.

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

예시

다음 샘플 수식은 인터넷 판매의 '연도 실행 합계' 또는 '연도 실행 합계'를 계산하는 측정값을 만듭니다.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

관련 콘텐츠

[ALL](#)
[계산](#)
[DATESYTD](#)
[TOTALMTD](#)
[TOTALQTD](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

명세서

아티클 • 2023. 10. 20.

이 범주에서

문	설명
정의	(키워드) 하나 이상의 EVALUATE 문에 적용할 수 있는 하나 이상의 엔터티 정의를 소개합니다.
평가	(키워드) DAX 쿼리를 실행하는 데 필요한 테이블 식이 포함된 문을 도입합니다.
MEASURE	(키워드) 쿼리에서 하나 이상의 EVALUATE 문에 사용할 수 있는 측정값 정의를 도입합니다.
ORDER BY	(키워드) EVALUATE 문에서 반환하는 쿼리 결과의 정렬 순서를 정의하는 문을 도입합니다.
START AT	(키워드) ORDER BY 문의 쿼리 결과가 반환되는 시작 값을 정의하는 문을 도입합니다.
VAR	(키워드) 식의 결과를 명명된 변수로 저장한 다음 다른 측정값 식에 인수로 전달할 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DEFINE

아티클 • 2023. 12. 14.

DAX 쿼리의 하나 이상의 EVALUATE 문에 적용할 수 있는 하나 이상의 엔터티 정의가 있는 문을 도입합니다.

구문

```
DAX

[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
)
+
]

(EVALUATE <table expression>) +
```

매개 변수

[\[\] 테이블 확장](#)

용어	정의
Entity	MEASURE, VAR, TABLE ¹ 또는 COLUMN ¹ .
name	측정값, var, 테이블 또는 열 정의의 이름입니다. 식일 수 없습니다. 이름은 중복되어도 문제가 없습니다. 이름은 쿼리 기간 동안만 존재합니다.
식	테이블 또는 스칼라 값을 반환하는 모든 DAX 식입니다. 식은 정의된 엔터티를 사용할 수 있습니다. 스칼라 식을 테이블 식으로 변환해야 하는 경우 중괄호 {}를 사용하여 테이블 생성자 내부에 식을 넣거나 ROW() 함수를 사용하여 단일 행 테이블을 반환합니다.

[1]주의: 쿼리 범위가 지정된 TABLE 및 COLUMN 정의는 내부적으로만 사용됩니다. 쿼리에 대한 TABLE 및 COLUMN 식은 구문 오류 없이 정의할 수 있지만 런타임 오류가 발생할 수 있으므로 권장되지 않습니다.

설명

- DAX 쿼리에 EVALUATE 문은 여러 개 있을 수 있지만 DEFINE 문은 하나만 있을 수 있습니다. DEFINE 문의 정의는 쿼리의 모든 EVALUATE 문에 적용할 수 있습니다.
- DEFINE 문에는 하나 이상의 정의가 필요합니다.
- 쿼리에 대한 측정값 정의는 동일한 이름의 모델 측정값을 재정의합니다.
- VAR 이름에는 고유한 제한이 있습니다. 자세한 내용은 [VAR - 매개 변수](#)를 참조하세요.
- DEFINE 문을 사용하는 방법에 대한 자세한 내용은 [DAX 쿼리](#)를 참조하세요.

관련 콘텐츠

[평가](#)

[VAR](#)

[MEASURE](#)

[DAX 쿼리](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

EVALUATE

아티클 • 2023. 12. 14.

DAX 쿼리에 필요한 테이블 식을 포함하는 문을 도입합니다.

구문

DAX

EVALUATE <table>

매개 변수

 테이블 확장

용어	정의
table	테이블 식

반환 값

테이블 식의 결과입니다.

설명

- DAX 쿼리는 여러 EVALUATE 문을 포함할 수 있습니다.
- EVALUATE 문을 사용하는 방법에 대한 자세한 내용은 [DAX 쿼리](#)를 참조하세요.

예시

DAX

EVALUATE
'Internet Sales'

Internet Sales 테이블의 모든 행과 열을 테이블로 반환합니다.

관련 콘텐츠

ORDER BY

START AT

정의

VAR

DAX 쿼리

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

MEASURE

아티클 • 2023. 10. 20.

DAX 쿼리의 DEFINE 문에 측정값 정의를 도입합니다.

구문

DAX

```
[DEFINE
(
    MEASURE <table name>[<measure name>] = <scalar expression>
) +
]

(EVALUATE <table expression>) +
```

매개 변수

용어	정의
테이블 이름	측정값을 포함하는 테이블의 이름입니다.
측정값 이름	측정값의 이름입니다. 식일 수 없습니다. 이름은 중복되어도 문제가 없습니다. 이름은 쿼리 기간 동안만 존재합니다.
스칼라식	스칼라 값을 반환하는 DAX 식입니다.

반환 값

측정값 식의 계산된 결과입니다.

설명

- 쿼리에 대한 측정값 정의는 쿼리 기간 동안 동일한 이름의 모델 측정값을 재정의합니다. 모델 측정값에는 영향을 미치지 않습니다.
- 측정값 식은 동일한 쿼리의 다른 식과 함께 사용할 수 있습니다.
- MEASURE 문을 사용하는 방법에 대한 자세한 내용은 [DAX 쿼리](#)를 참조하세요.

참고 항목

정의

평가

VAR

DAX 쿼리

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

ORDER BY

아티클 • 2023. 12. 14.

DAX 쿼리에서 EVALUATE 문이 반환하는 쿼리 결과의 정렬 순서를 정의하는 문을 도입합니다.

구문

DAX

```
[ORDER BY {<expression>} [{ASC | DESC}]}[, ...]]
```

매개 변수

 테이블 확장

용어	정의
식	단일 스칼라 값을 반환하는 DAX 식입니다.
ASC	(기본값) 오름차순 정렬 순서입니다.
DESC	내림차순 정렬 순서입니다.

반환 값

ASC(오름차순) 또는 내림차순(DESC) 순서의 EVALUATE 문의 결과입니다.

설명

ORDER BY 문을 사용하는 방법에 대한 자세한 내용은 [DAX 쿼리](#)를 참조하세요.

관련 콘텐츠

[START AT](#)

[평가](#)

[VAR](#)

[정의](#)

[DAX 쿼리](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

START AT

아티클 • 2023. 12. 14.

DAX 쿼리의 EVALUATE 문에서 ORDER BY 절의 쿼리 결과가 반환되는 시작 값을 정의하는 문을 도입합니다.

구문

DAX

```
[START AT {<value>|<parameter>} [, ...]]
```

매개 변수

 테이블 확장

용어	정의
value	상수 값입니다. 식일 수 없습니다.
parameter	문자 접두사로 된 XMLA 문의 매개 변수 이름입니다 @ .

설명

- START AT 인수는 ORDER BY 문의 열과 일대일 대응을 갖습니다. ORDER BY 문과 같이 START AT 문에 여러 인수가 사용될 수 있지만, ORDER BY 절보다 많을 수 없습니다. START AT 문의 첫 번째 인수는 ORDER BY 열의 열 1에서 시작 값을 정의합니다. START AT 문의 두 번째 인수는 열 1의 첫 번째 값을 총족하는 행 내에서 ORDER BY 열의 열 2에 있는 시작 값을 정의합니다.
- START AT 문을 사용하는 방법에 대한 자세한 내용은 DAX 쿼리를 참조하세요.

관련 콘텐츠

[ORDER BY](#)

[평가](#)

[VAR](#)

[정의](#)

[DAX 쿼리](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

VAR

아티클 • 2023. 10. 20.

식의 결과를 명명된 변수로 저장한 다음 다른 측정값 식에 인수로 전달할 수 있습니다. 일단 변수 식에 대한 결과 값이 계산되면 변수가 다른 식에서 참조되더라도 해당 값은 변경되지 않습니다.

구문

DAX

```
VAR <name> = <expression>
```

매개 변수

용어 정의

name 변수(식별자)의 이름입니다.

구분 기호는 지원되지 않습니다. 예를 들어 'varName' 또는 [varName]에서 오류가 발생합니다.

지원되는 문자 집합: a-z, A-Z, 0-9.

0-9는 첫 번째 문자로 유효하지 않습니다.

_ (이중 밑줄)는 식별자 이름의 접두사로 허용됩니다.

다른 특수 문자는 지원되지 않습니다.

예약된 키워드(keyword) 허용되지 않습니다.

기존 테이블의 이름은 허용되지 않습니다.

빈 공간은 허용되지 않습니다.

식 스칼라 또는 테이블 값을 반환하는 DAX 식입니다.

반환 값

식 인수의 결과를 포함하는 명명된 변수입니다.

설명

- VAR에 인수로 전달된 식에는 다른 VAR 선언이 포함될 수 있습니다.
- 변수를 참조하는 경우:
 - 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.
- TableName[ColumnName] 구문을 통해 테이블 변수의 열을 참조할 수 없습니다.
- VAR을 사용하는 경우 모범 사례는 변수를 사용하여 DAX 수식을 개선하는 방법을 참조 [하세요](#).
- DAX 쿼리 내에서 VAR을 사용하는 방법에 대한 자세한 내용은 [DAX 쿼리](#)를 참조하세요.

예시

변수를 사용하지 않고 전년 대비 증가율을 계산하려면 세 개의 별도 측정값을 만들 수 있습니다. 이 첫 번째 측정값은 판매액 합계를 계산합니다.

DAX

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

두 번째 측정값은 이전 연도의 판매액을 계산합니다.

DAX

```
SalesAmount PreviousYear =
CALCULATE([Sum of SalesAmount],
SAMEPERIODLASTYEAR(Calendar[Date]))
)
```

그런 다음 다른 두 측정값을 결합하여 증가율을 계산하는 세 번째 측정값을 만들 수 있습니다. SalesAmount의 합계 측정값은 두 곳에서 사용됩니다. 먼저 판매가 있는지 확인한 다음 다시 백분율을 계산합니다.

DAX

```
Sum of SalesAmount YoY%: =
IF([Sum of SalesAmount] ,
DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of SalesAmount])
)
```

변수를 사용하여 동일한 결과를 계산하는 단일 측정값을 만들 수 있습니다.

DAX

```
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 얻을 수 있지만 더 읽기 쉬운 방식으로 얻을 수 있습니다. 식의 결과가 변수에 저장되기 때문에 변수를 사용할 때마다 다시 계산할 필요가 없으므로 측정값의 성능이 크게 향상될 수 있습니다.

참고 항목

[변수를 사용하여 DAX 수식 개선](#)

[DAX 쿼리](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX 용어집

아티클 • 2023. 10. 20.

분석 쿼리

Power BI 시각적 개체는 분석 쿼리를 사용하여 *데이터 모델*을 *쿼리*합니다. 분석 쿼리는 필터, 그룹화 및 요약이라는 세 가지 개별 단계를 사용하여 잠재적으로 큰 데이터 볼륨 및 모델 복잡성을 줄이기 위해 노력합니다. 분석 쿼리는 필드가 보고서 시각적 개체의 웹에 할당되면 자동으로 생성됩니다. 보고서 작성자는 필드 이름을 바꾸거나 요약 기술을 수정하거나 요약을 사용하지 않도록 설정하여 그룹화를 수행하여 필드 할당의 동작을 제어할 수 있습니다. 보고서 디자인 타임에 필터를 보고서, 보고서 페이지 또는 시각적 개체에 추가할 수 있습니다. 읽기용 보기에서 필터는 필터 **창에서** 또는 슬라이서 및 기타 시각적 개체와의 상호 작용(교차 필터링)을 통해 수정할 수 있습니다.

BLANK

DAX는 값이 없는 것을 BLANK로 정의합니다. SQL NULL과 동일하지만 정확히 동일하게 동작하지는 않습니다. Excel과 더 밀접하게 정렬되며 빈 셀을 정의하는 방법입니다. BLANK는 다른 작업과 결합될 때 0 또는 빈 문자열로 평가됩니다. 예를 들어 $\text{BLANK} + 20 = 20$ 입니다. 항상 대문자 사용; 복수는 소문자 "s"가 있는 BLANK입니다.

계산 열

DAX 수식을 작성하여 테이블 형식 모델에 열을 추가하는 데 사용되는 모델 계산입니다. 수식은 스칼라 값을 반환해야 하며 테이블의 각 행에 대해 계산됩니다. 계산 열을 Import 또는 DirectQuery 스토리지 모드 테이블에 추가할 수 있습니다.

계산된 측정값

테이블 형식 모델링에는 계산 측정값과 같은 개념이 없습니다. 대신 측정값을 사용합니다. 계산이라는 단어가 계산된 테이블 및 계산 열을 설명하기 위해 사용됩니다. 파워 쿼리에서 시작된 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 스토리지 가져오기 모드를 사용하는

테이블이 생성됩니다.

계산

하나 이상의 입력을 하나 이상의 결과로 변환하는 의도적인 프로세스입니다. 테이블 형식 데이터 모델에서 계산은 모델 개체일 수 있습니다. 계산 테이블, 계산 열 또는 측정값 중 하나

컨텍스트

DAX 수식이 평가되는 환경에 대해 설명합니다. 컨텍스트에는 행 컨텍스트와 필터 컨텍스트의 두 가지 유형이 있습니다. 행 컨텍스트는 "현재 행"을 나타내며 테이블 반복기에서 사용되는 계산 열 수식 및 식을 평가하는 데 사용됩니다. 필터 컨텍스트는 측정값을 평가하는 데 사용되며 모델 관계에 의해 전파된 모델 열 및 필터에 직접 적용되는 필터를 나타냅니다.

DAX

DAX(데이터 분석 식) 언어는 Sql Server Analysis Services의 Excel, Power BI, Azure Analysis Services 및 테이블 형식 모델링의 파워 피벗에 대한 수식 언어입니다. DAX를 사용하여 데이터 모델 계산을 추가하고 RLS(행 수준 보안) 규칙을 정의할 수도 있습니다.

동적 보안

보고서 사용자의 ID를 사용하여 RLS(행 수준 보안) 규칙을 적용하는 경우 규칙은 USERNAME 또는 USERPRINCIPALNAME 함수로 수행할 수 있는 사용자의 계정 이름을 사용하여 모델 테이블을 필터링합니다. [행 수준 보안](#)을 참조하세요.

식

평가되고 결과를 반환하는 DAX 논리의 단위입니다. 식은 하위 식이 할당된 경우 변수를 선언할 수 있으며 최종 식을 출력하는 RETURN 문을 포함해야 합니다. 식은 모델 개체(테이블, 열 또는 측정값), 함수, 연산자 또는 상수로 구성됩니다.

필드

필드 **창**에 표시되는 데이터 모델 리소스입니다. 필드는 보고서 필터 및 시각적 개체를 구성하는 데 사용됩니다. 필드는 모델 열, 계층 수준 및 측정값으로 구성됩니다.

수식

모델 계산을 정의하는 데 사용되는 하나 이상의 DAX 식입니다. 내부 식을 하위 식이라고 합니다. 복수는 수식입니다.

함수

DAX 함수에는 매개 변수 전달을 허용하는 인수가 있습니다. 수식은 여러 함수 호출을 사용할 수 있으며, 다른 함수 내에서 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 합계, 평균, 최소, 최대, 개수(고유), 개수, 표준 편차, 분산 또는 중앙값을 포함하여 가장 큰 요약 범위를 지원합니다. 다른 데이터 형식의 열도 요약할 수 있습니다. **텍스트** 열은 첫 번째(사전순), 마지막(사전순), 개수(고유) 또는 개수를 사용하여 요약할 수 있습니다. **날짜** 열은 가장 이른 날짜, 최신, 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 계산에서 데이터를 요약하는 방법을 유연하게 제어할 수 있습니다.

MDX

MDX(다차원 식) 언어는 SQL Server Analysis Services 다차원 모델(큐브라고도 함)에 대한 수식 언어입니다. MDX는 테이블 형식 모델을 쿼리하는 데 사용할 수 있지만 암시적 측정값을 정의할 수는 없습니다. 모델에 이미 정의된 측정값만 쿼리할 수 있습니다.

측정값

요약을 달성하는 계산입니다. 측정값은 [암시적](#)이거나 [명시적](#)입니다. 명시적 측정값은 DAX 수식을 작성하여 테이블 형식 데이터 모델에 추가된 계산입니다. 측정값 수식은 스칼라 값을 반환해야 합니다. 필드 창에서 명시적 측정값은 계산기 아이콘으로 표시됩니다. Excel에서 분석을 사용할 때와 마찬가지로 MDX(다차원 식)를 사용하여 모델을 쿼리할 때 명시적 측정값이 필요합니다. 명시적 측정값은 일반적으로 측정값이라고만 불립니다.

측정값 그룹

하나 이상의 측정값을 포함하고 계층 구조 또는 표시되는 열이 없는 모델 테이블입니다. 필드 창에서 각 측정값 그룹은 다중 계산기 아이콘으로 표시됩니다. 측정값 그룹은 필드 창의 맨 위에 함께 나열되고 이름별로 사전순으로 정렬됩니다.

모델 계산

테이블 형식 데이터 모델에 계산 테이블, 계산 열 또는 측정값을 추가하는 데 사용되는 명명된 수식입니다. 해당 구조는 <NAME> = <FORMULA>입니다. 대부분의 계산은 Power BI Desktop의 데이터 모델러에 의해 추가되지만 측정값을 라이브 연결 보고서에 추가할 수도 있습니다. 보고서 측정값을 참조하세요.

빠른 측정값

일반적으로 정의된 측정값에 대한 DAX 수식을 작성할 필요가 없는 Power BI Desktop의 기능입니다. 빠른 측정에는 범주별 평균, 순위 및 기준선의 차이가 포함됩니다.

보고서 측정값

보고서 수준 측정이라고도 합니다. DAX 수식을 작성하지만 Power BI 모델 또는 Analysis Services 테이블 형식 모델에 대한 연결에 대해서만 Power BI Desktop의 라이브 연결 보고서에 추가됩니다.

행 수준 보안

RLS라고도 합니다. 특정 사용자의 데이터 하위 집합에 대한 액세스를 제한하는 디자인 기술입니다. 테이블 형식 모델에서는 모델 역할을 만들어 달성합니다. 역할에는 테이블 행을 필터링하는 DAX 식인 규칙이 있습니다.

스칼라

DAX에서 스칼라는 단일 값입니다. 스칼라는 10진수, 정수, DateTime, String, Currency, Boolean 등의 모든 데이터 형식일 수 있습니다. 스칼라 값은 여러 값에서 계산된 식의 결과일 수 있습니다. 예를 들어 MAX()와 같은 집계 함수는 평가할 값 집합에서 단일 최대값을 반환합니다.

요약

열 값에 적용된 작업입니다. 측정값을 참조하세요.

시간 인텔리전스

시간 인텔리전스는 YTD(연도별)와 같은 시간에 따른 계산과 관련이 있습니다.

시간 인텔리전스 함수

DAX에는 많은 시간 인텔리전스 함수가 포함되어 있습니다. 인텔리전스 함수가 날짜 필터에 대한 필터 컨텍스트를 수정하여 결과를 얻을 때마다 함수 예: TOTALYTD 및 SAMEPERIODLASTYEAR.

값

시각화할 데이터입니다.

가상 매개 변수

슬라이서를 통해 사용자 입력을 수락하는 기능을 제공하는 Power BI Desktop 기능입니다. 각 매개 변수는 단일 열 계산 테이블과 단일 선택한 값을 반환하는 측정값을 만듭니다. 측정값은 모델 계산에 사용하여 사용자의 입력에 응답할 수 있습니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX 연산자

아티클 • 2023. 10. 20.

DAX(데이터 분석 식) 언어는 연산자를 사용하여 값을 비교하거나 산술 계산을 수행하거나 문자열을 사용하는 식을 만듭니다.

연산자 유형

산술, 비교, 텍스트 연결 및 논리의 네 가지 계산 연산자 유형이 있습니다.

산술 연산자

더하기, 빼기 또는 곱하기와 같은 기본 수학 연산을 수행하려면 숫자 결합; 숫자 결과를 생성하고 다음 산술 연산자를 사용합니다.

산술 연산자	의미	예시
+ (더하기 기호)	더하기	3+3
- (빼기 기호)	빼기 또는 기호	3-1-1
*(별표)	곱하기	3*3
/(슬래시)	나누기	3/3
^ (caret)	지수	16^4

① 참고

더하기 기호는 이진 연산자 및 단항 연산자로 모두 작동할 수 있습니다. 이진 연산자는 연산자의 양쪽에 숫자가 필요하며 더하기 작업을 수행합니다. 이진 연산자의 양 쪽에 있는 DAX 수식에서 값을 사용하는 경우 DAX는 값이 숫자가 아닌 경우 숫자 데이터 형식으로 캐스팅하려고 시도합니다. 반면 단항 연산자는 모든 형식의 인수에 적용할 수 있습니다. 더하기 기호는 형식이나 값에 영향을 주지 않으며 단순히 무시되는 반면, 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 TRUE 또는 FALSE의 논리적 값입니다.

비교 연산자	의미	예시
=	다음과 같음	[지역] = "USA"
==	엄격한 같음	[Region] == "USA"
>	보다 큼	[판매 날짜] > "2009년 1월"
<	보다 작음	[판매 날짜] < "2009년 1월 1일"
>=	크거나 같음	[금액] >= 20000
<=	보다 작거나 같음	[금액] <= 100
<>	같지 않음	[지역] <> "USA"

==를 제외한 모든 비교 연산자는 BLANK를 숫자 0, 빈 문자열 "", DATE(1899, 12, 30) 또는 FALSE로 처리합니다. 따라서 [Column] 값이 0이거나 BLANK이면 [Column] = 0이 true입니다. 반면 [Column] == 0은 [Column] 값이 0인 경우에만 true입니다.

텍스트 연결 연산자

앰퍼샌드(&)를 사용하여 두 개 이상의 텍스트 문자열을 조인하거나 연결하여 단일 텍스트를 생성합니다.

텍스트 연산자	의미	예시
&(앰퍼샌드)	연속 텍스트 값을 생성하기 위한 두 값 커넥트 또는 연결	[지역] & ", " & [City]

논리 연산자

논리 연산자(&&) 및 (||)를 사용하여 식을 결합하여 단일 결과를 생성합니다.

텍스트 연산자	의미	예제
&& (이 중 앰퍼 샌드)	각각 부울 결과가 있는 두 식 사이에 AND 조건을 만듭니다. 두 식이 모두 TRUE를 반환할 경우 식 조합에서도 TRUE가 반환되고 그렇지 않으면 조합이 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 비교되는 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 순서를 결정하는 방법과 순서를 변경하여 원하는 결과를 얻는 방법을 이해하는 것이 중요합니다.

계산 순서

식은 연산자와 값을 특정 순서로 평가합니다. 모든 식은 항상 등호(=)로 시작합니다. 등호는 성공한 문자가 식을 구성한다는 것을 나타냅니다.

등호 다음에 계산할 요소(피연산자)는 계산 연산자로 구분됩니다. 식은 항상 왼쪽에서 오른쪽으로 읽지만 요소가 그룹화되는 순서는 괄호를 사용하여 어느 정도 제어할 수 있습니다.

연산자 우선 순위

여러 연산자를 단일 수식으로 결합하면 다음 표에 따라 연산이 정렬됩니다. 연산자의 우선 순위 값이 같으면 왼쪽에서 오른쪽으로 정렬됩니다. 예를 들어 식에 곱하기 연산자와 나누기 연산자가 모두 포함된 경우 식에 나타나는 순서대로 왼쪽에서 오른쪽으로 계산됩니다.

연산자	설명
^	지수
-	로그인(-1에서와 같이)
* 및 /	곱하기 및 나누기
+ 및 -	더하기 및 빼기
&	두 개의 텍스트 문자열을 커넥트(연결)
=, ==, <, >, <=, >=, <>, IN	비교
NOT	NOT(단항 연산자)

괄호를 사용하여 계산 순서 제어

계산 순서를 변경하려면 먼저 계산해야 하는 수식의 해당 부분을 괄호로 묶어야 합니다. 예를 들어 다음 수식은 곱하기 전에 곱하기 전에 계산되므로 11을 생성합니다. 수식은 2를 3으로 곱한 다음 결과에 5를 추가합니다.

```
=5+2*3
```

반면 괄호를 사용하여 구문을 변경하는 경우 순서가 변경되어 5와 2가 함께 추가되고 결과는 3을 곱하여 21을 생성합니다.

```
DAX
```

```
=(5+2)*3
```

다음 예제에서는 수식의 첫 번째 부분 주위에 괄호를 적용하여 계산에서 식을 $(3 + 0.25)$ 먼저 계산한 다음 결과를 식의 결과로 나눕니다($3 - 0.25$).

```
DAX
```

```
=(3 + 0.25)/(3 - 0.25)
```

다음 예제에서는 연산자의 우선 순위 규칙에 따라 지수 연산자가 먼저 적용된 다음 기호 연산자가 적용됩니다. 이 식의 결과는 -4입니다.

```
DAX
```

```
--2^2
```

기호 연산자가 숫자 값에 먼저 적용되도록 하려면 다음 예제와 같이 괄호를 사용하여 연산자를 제어할 수 있습니다. 이 식의 결과는 4입니다.

```
DAX
```

```
= (-2)^2
```

호환성

DAX는 Microsoft Excel과 마찬가지로 다양한 데이터 형식을 쉽게 처리하고 비교합니다. 그러나 기본 계산 엔진은 SQL Server Analysis Services를 기반으로 하며 날짜 및 시간 유형에 대한 다양한 지원을 포함하여 관계형 데이터 저장소의 추가 고급 기능을 제공합니다. 따라서 경우에 따라 계산 결과 또는 함수 동작이 Excel에서와 동일하지 않을 수 있습니다. 또한 DAX는 Excel보다 더 많은 데이터 형식을 지원합니다. 이 섹션에서는 주요 차이점에 대해 설명합니다.

피연산자의 데이터 형식 강제 변환

일반적으로 연산자의 왼쪽과 오른쪽에 있는 두 피연산자는 동일한 데이터 형식이어야 합니다. 그러나 데이터 형식이 다른 경우 DAX는 이를 공통 데이터 형식으로 변환하여 경우에 따라 연산자를 적용합니다.

1. 두 피연산자는 가능한 가장 큰 공통 데이터 형식으로 변환됩니다.
2. 가능한 경우 연산자가 적용됩니다.

예를 들어 결합하려는 두 개의 숫자가 있다고 가정해 보겠습니다. `= [Price] * .20`와 같은 수식의 숫자 하나와 결과에는 소수 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 형식으로 실제 숫자로 변환합니다. 그런 다음 DAX는 곱셈을 적용합니다.

데이터 형식 조합에 따라 비교 작업에 형식 강제 변환을 적용할 수 없습니다. DAX에서 지원하는 데이터 형식의 전체 목록은 Power BI Desktop의 테이블 형식 모델 [및 데이터 형식에서 지원되는 데이터 형식을 참조하세요](#).

정수, 실수, 통화, 날짜/시간 및 공백은 비교를 위해 숫자로 간주됩니다. 비교를 수행할 때 빈 값이 0으로 계산됩니다. 비교 작업에는 다음과 같은 데이터 형식 조합이 지원됩니다.

왼쪽 데이터 형식	오른쪽 데이터 형식
숫자	숫자
Boolean	Boolean
문자열	문자열

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	Excel에서 사용되는 데이터 형식
숫자(I8, R8)	숫자(R8)
문자열	문자열
부울	Boolean
DateTime	변형
통화	통화

우선 순위 차이

DAX 수식에서 작업의 우선 순위는 기본적으로 Microsoft Excel에서 사용하는 것과 동일 하지만 백분율과 같은 일부 Excel 연산자는 지원되지 않습니다. 또한 범위는 지원되지 않습니다.

따라서 Excel에서 수식을 복사하여 붙여 넣을 때마다 수식의 일부 연산자 또는 요소가 유효하지 않을 수 있으므로 수식을 신중하게 검토해야 합니다. 작업이 수행되는 순서에 대해 의심의 여지가 있는 경우 괄호를 사용하여 작업 순서를 제어하고 결과에 대한 모호성을 제거하는 것이 좋습니다.

참고 항목

[DAX 구문](#)

[DAX 매개 변수 이름 지정](#)

피드백

이 페이지가 도움이 되었나요?

Yes

No

DAX 쿼리

아티클 • 2023. 10. 20.

Power BI 및 Excel과 같은 보고 클라이언트는 보고서에 필드가 배치되거나 필터가 적용될 때마다 DAX 쿼리를 실행합니다. SSMS([SQL Server Management Studio](#)), [Power BI Report Builder](#) 및 [DAX Studio](#) 와 같은 오픈 소스 도구를 사용하여 고유한 DAX 쿼리를 만들고 실행할 수 있습니다. DAX 쿼리는 도구 내에서 바로 결과를 테이블로 반환하므로 DAX 수식을 빠르게 만들고 테스트할 수 있습니다.

쿼리에 대해 알아보기 전에 DAX 기본 사항을 확실하게 이해하는 것이 중요합니다. 아직 없는 경우 [DAX 개요](#)를 검사.

키워드

DAX 쿼리에는 하나의 필수 키워드, EVALUATE로 구성되는 간단한 구문과 ORDER BY, START AT, DEFINE, MEASURE, VAR, TABLE 및 COLUMN과 같은 몇 가지 선택적 키워드로 구성된 구문이 있습니다. 각 키워드는 쿼리 기간 동안 사용되는 문을 정의합니다.

EVALUATE(필수)

가장 기본적인 수준에서 DAX 쿼리는 테이블 식을 포함하는 EVALUATE 문입니다. 하나 이상의 EVALUATE 문이 필요하지만 하나의 쿼리에 여러 EVALUATE 문이 포함될 수 있습니다.

EVALUATE 구문

DAX
EVALUATE <table>

EVALUATE 매개 변수

용어	정의
table	테이블 식입니다.

EVALUATE 예제

DAX

```
EVALUATE  
    'Internet Sales'
```

Internet Sales 테이블의 모든 행과 열을 테이블로 반환합니다.

The screenshot shows the Microsoft Power BI Data Editor interface. The query bar at the top contains the DAX code: `EVALUATE
 'Internet Sales'
)`. The results pane below displays a table with 10 columns and approximately 100 rows of data from the Internet Sales table. The columns are labeled: Internet Sales[ProductKey], Internet Sales[CustomerKey], Internet Sales[PromotionKey], Internet Sales[OrderKey], Internet Sales[SalesAmount], Internet Sales[SalesAmount], Internet Sales[SalesAmount], Internet Sales[SalesAmount], Internet Sales[SalesAmount], and Internet Sales[SalesAmount]. The data includes various product keys, customer keys, promotion keys, and sales amounts. A status bar at the bottom indicates "Query executed successfully." and provides connection information: "asazure://westus.asazure.wi...", "adventureworks", and "00:00:08".

ORDER BY(선택 사항)

선택적 ORDER BY 키워드(keyword) 쿼리 결과를 정렬하는 데 사용되는 하나 이상의 식을 정의합니다. 결과의 각 행에 대해 평가할 수 있는 식은 유효합니다.

ORDER BY 구문

DAX

```
EVALUATE <table>  
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]]
```

ORDER BY 매개 변수

용어	정의
식	단일 스칼라 값을 반환하는 DAX 식입니다.
ASC	(기본값) 오름차순 정렬 순서입니다.
DESC	내림차순 정렬 순서입니다.

ORDER BY 예

DAX

```
EVALUATE  
    'Internet Sales'  
  
ORDER BY  
    'Internet Sales'[Order Date]
```

주문 날짜의 오름차순으로 정렬된 인터넷 판매 테이블의 모든 행 및 열을 테이블로 반환합니다.

The screenshot shows the Microsoft Power BI Data Editor interface. The top navigation bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The ribbon below has tabs for New Query, DAX, XLSX, and DCF. The main area displays a DAX query:

```
EVALUATE  
    'Internet Sales'  
)  
ORDER BY  
    'Internet Sales'[Order Date]
```

Below the query, the Results tab is selected, showing a table with the following data:

Internet Sales[SalesID]	Internet Sales[TerritoryID]	Internet Sales[Region]	Internet Sales[CountryRegionCode]	Internet Sales[Currency]	Internet Sales[Order Date]	Internet Sales[Due Date]	Internet Sales[Ship Date]	Internet Sales[Status]
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
78.27	286.2616	89.4568			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1406.97€
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	1487.83€
9.0982	55.9279	17.4775			12/29/2010 12:00:00 AM	1/10/2011 12:00:00 AM	1/5/2011 12:00:00 AM	285.951€
99.99	271.9992	84.9998			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1487.83€
74.99	269.9992	84.3748			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1476.89€
78.27	286.2616	89.4568			12/30/2010 12:00:00 AM	1/11/2011 12:00:00 AM	1/6/2011 12:00:00 AM	1406.97€

At the bottom, a message indicates "Query executed successfully." and shows connection details: "asazure://westus.asazure.wi..." and "adventureworks". The status bar shows "Ready", "Ln 1", "Col 1", and "INS".

START AT(선택 사항)

선택적 START AT 키워드(keyword) ORDER BY 절 내에 사용됩니다. 쿼리 결과가 시작되는 값을 정의합니다.

START AT 구문

DAX

```
EVALUATE <table>  
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]  
[START AT {<value>|<parameter>} [, ...]]]
```

START AT 매개 변수

용어	정의
value	상수 값입니다. 식일 수 없습니다.
parameter	문자 접두사로 된 XMLA 문의 매개 변수 이름입니다 @ .

START AT 설명

START AT 인수에는 ORDER BY 절의 열과 일대일 대응이 있습니다. START AT 절에는 ORDER BY 절에 있는 만큼 많은 인수가 있을 수 있지만 더 이상 있을 수는 없습니다. START AT의 첫 번째 인수는 ORDER BY 열의 열 1에서 시작 값을 정의합니다. START AT의 두 번째 인수는 열 1의 첫 번째 값을 충족하는 행 내 ORDER BY 열의 열 2에서 시작 값을 정의합니다.

START AT 예제

```
DAX

EVALUATE
    'Internet Sales'

ORDER BY
    'Internet Sales'[Sales Order Number]
    START AT "SO7000"
```

SO7000에서 시작하여 판매 주문 번호의 오름차순으로 정렬된 인터넷 판매 테이블의 모든 행 및 열을 반환합니다.

The screenshot shows the Microsoft Data Explorer interface. The query window contains the following DAX code:

```
EVALUATE
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Sales Order Number]
    START AT "SO7000"
```

The results pane displays a table with the following data:

Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[Pr...]	Internet Sales[C...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[S...]	Internet Sales[R...]	Internet Sales[O...]
604	28153	1	19	6	SO70000	1	1	1
538	28153	1	19	6	SO70000	2	1	1
225	28153	1	19	6	SO70000	3	1	1
222	21797	1	100	1	SO70001	2	1	1
384	21797	1	100	1	SO70001	1	1	1

At the bottom of the interface, a message bar indicates: "Query executed successfully." and the URL "asazure://westus.asazure.wi...".

DEFINE(선택 사항)

선택적 **DEFINE** 키워드는 쿼리 기간 동안만 존재하는 하나 이상의 계산된 엔터티 정의를 도입합니다. 정의는 EVALUATE 문 앞에 있으며 쿼리의 모든 EVALUATE 문에 대해 유효합니다. 정의는 변수, 측정값, 테이블¹ 및 열¹일 수 있습니다. 정의는 현재 정의 앞이나 후에 나타나는 다른 정의를 참조할 수 있습니다. DEFINE 키워드가 쿼리에 포함된 경우 하나 이상의 정의가 필요합니다.

DEFINE 구문

```
DAX

[DEFINE
(
    (MEASURE <table name>[<measure name>] = <scalar expression>) |
    (VAR <var name> = <table or scalar expression>) |
    (TABLE <table name> = <table expression>) |
    (COLUMN <table name>[<column name>] = <scalar expression>) |
) +
]

(EVALUATE <table expression>) +
```

DEFINE 매개 변수

용어	정의
Entity	MEASURE, VAR, TABLE ¹ 또는 COLUMN ¹ .
name	측정값, var, 테이블 또는 열 정의의 이름입니다. 식일 수 없습니다. 이름은 중복되어도 문제가 없습니다. 이름은 쿼리 기간 동안만 존재합니다.
식	테이블 또는 스칼라 값을 반환하는 모든 DAX 식입니다. 식은 정의된 엔터티를 사용할 수 있습니다. 스칼라 식을 테이블 식으로 변환해야 하는 경우 중괄호 {}를 사용하여 테이블 생성자 내부에 식을 넣거나 <code>ROW()</code> 함수를 사용하여 단일 행 테이블을 반환합니다.

[1]**주의:** 쿼리 범위가 지정된 TABLE 및 COLUMN 정의는 내부적으로만 사용됩니다. 쿼리에 대한 TABLE 및 COLUMN 식은 구문 오류 없이 정의할 수 있지만 런타임 오류가 발생할 수 있으므로 권장되지 않습니다.

DEFINE 설명

- DAX 쿼리에 EVALUATE 문은 여러 개 있을 수 있지만 DEFINE 문은 하나만 있을 수 있습니다. DEFINE 문의 정의는 쿼리의 모든 EVALUATE 문에 적용할 수 있습니다.
- DEFINE 문에는 하나 이상의 정의가 필요합니다.

- 쿼리에 대한 측정값 정의는 동일한 이름의 모델 측정값을 재정의하지만 쿼리 내에서만 사용됩니다. 모델 측정값에는 영향을 미치지 않습니다.
- VAR 이름에는 고유한 제한이 있습니다. 자세한 내용은 [VAR - 매개 변수](#)를 참조하세요.

DEFINE 예제

DAX

```

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 2014년의 계산된 총 판매액과 2013년과 2014년 계산 총 매출을 테이블로 반환합니다. DEFINE 문의 측정값인 Internet Total Sales는 Total Sales 및 Combined Years Total Sales 식 모두에 사용됩니다.

The screenshot shows the Microsoft Power BI Data Editor interface. The top menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The Query ribbon tab is selected. The main area has tabs for New Query, DAX, XMLA, and DAXA. The Query pane displays the following DAX code:

```
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 shows a table with three columns: Date[Calendar Y..., [Total Sales], and [Combined Years...]. The data is as follows:

Date[Calendar Y...	[Total Sales]	[Combined Years...
2013	16351550.34	16397245.06
2014	45694.72	16397245.06

At the bottom, status indicators show 'Query executed successfully.' and connection details: asazure://westus.asazure.wi... | adventureworks | 00:00:01.

DAX 쿼리의 매개 변수

잘 정의된 DAX 쿼리 문을 매개 변수화한 다음 매개 변수 값의 변경 내용으로 반복해서 사용할 수 있습니다.

XMLA(Execute 메서드) 메서드에는 매개 변수를 정의하고 값을 할당할 수 있는 XMLA(Parameters 요소) 컬렉션 요소가 있습니다. 컬렉션 내에서 각 XMLA(Parameter 요소) 요소는 매개 변수의 이름과 해당 매개 변수의 값을 정의합니다.

매개 변수 이름 앞에 문자를 추가하여 XMLA 매개 변수를 참조합니다 @ . 값이 허용되는 구문의 모든 위치에서 값을 매개 변수 호출로 바꿀 수 있습니다. 모든 XMLA 매개 변수는 텍스트로 입력됩니다.

① 중요

매개 변수 섹션에 정의되고 STATEMENT> 요소에 사용되지 않는 매개 변수는 <XMLA에서 오류 응답을 생성합니다. Parameters 요소에 사용되지 않고 정의되지 않은 매개 변수>는 <XMLA에서 오류 응답을 생성합니다.

참고 항목

[DAX 문](#)

[SUMMARIZECOLUMNS](#)

[TREATAS](#)

[필터](#)

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX 매개 변수 명명 규칙

아티클 • 2023. 10. 20.

매개 변수 이름은 함수의 사용 및 이해를 용이하게 하기 위해 DAX 참조로 표준화됩니다.

매개 변수 이름

용어	정의
식	단일 스칼라 값을 반환하는 DAX 식입니다. 여기서 식은 각 행/컨텍스트에 대해 여러 번 계산됩니다.
value	다른 모든 작업 전에 식을 정확히 한 번 계산할 단일 스칼라 값을 반환하는 DAX 식입니다.
table	데이터 테이블을 반환하는 DAX 식입니다.
tableName	표준 DAX 구문을 사용하는 기존 테이블의 이름입니다. 식일 수 없습니다.
columnName	표준 DAX 구문을 사용하는 기존 열의 이름(일반적으로 정규화)입니다. 식일 수 없습니다.
name	새 개체의 이름을 제공하는 데 사용할 문자열 상수입니다.
order	정렬 순서를 결정하는 데 사용되는 열거형입니다.
ties	동률 값의 처리를 결정하는 데 사용되는 열거형입니다.
type	PathItem 및 PathItemReverse의 데이터 형식을 확인하는 데 사용되는 열거형입니다.

매개 변수 이름 접두사 또는 접두사만 사용

용어	정의
접두사	매개 변수 이름은 인수를 사용하는 방법과 매개 변수의 모호한 읽기를 방지하는 방법을 설명하는 접두사로 한정될 수 있습니다. 예시: Result_ColumnName - LOOKUPVALUE() 함수의 결과 값을 가져오는 데 사용되는 기존 열을 참조합니다. Search_ColumnName - LOOKUPVALUE() 함수에서 값을 검색하는 데 사용되는 기존 열을 나타냅니다.

용어 정의

생략 접두사에서 매개 변수를 설명할 수 있을 만큼 명확하면 매개 변수 이름이 생략됩니다.

예를 들어 다음 구문 DATE(Year_Value, Month_Value, Day_Value)를 갖는 대신 사용자가 DATE(Year, Month, Day)를 읽는 것이 더 명확합니다. 접미사 값을 세 번 반복해도 함수를 더 잘 이해할 수 없으며 읽기가 불필요하게 복잡해집니다.

그러나 접두사 매개 변수가 Year_columnName 경우 매개 변수 이름과 접두사는 사용자가 매개 변수에 기존 Years 열에 대한 참조가 필요하다는 것을 이해하도록 유지됩니다.

피드백

이 페이지가 도움이 되었나요?

 Yes

 No

DAX 구문

아티클 • 2023. 10. 20.

이 문서에서는 DAX 수식 식 언어에 대한 구문 및 요구 사항을 설명합니다.

구문 요구 사항

DAX 수식은 항상 등호(=)로 시작합니다. 등호 기호 후에는 스칼라로 계산되는 식이나 스칼라로 변환할 수 있는 식을 제공할 수 있습니다. 여기에는 다음이 포함됩니다.

- 스칼라 상수 또는 스칼라 연산자(+,-,*,/,>=,...,&&, ...)를 사용하는 식
- 열 또는 테이블에 대한 참조입니다. DAX 언어는 항상 테이블과 열을 함수에 대한 입력으로 사용하며 배열이나 임의의 값 집합은 사용하지 않습니다.
- 식의 일부로 제공되는 연산자, 상수 및 값입니다.
- 함수 및 해당 필수 인수의 결과입니다. 일부 DAX 함수는 스칼라 대신 테이블을 반환하며 테이블을 평가하고 스칼라를 반환하는 함수에 래핑되어야 합니다. 테이블이 단일 열, 단일 행 테이블이 아니면 스칼라 값으로 처리됩니다.

대부분의 DAX 함수에는 테이블, 열, 식 및 값을 포함할 수 있는 하나 이상의 인수가 필요합니다. 그러나 PI와 같은 일부 함수에는 인수가 필요하지 않지만 항상 null 인수를 나타내려면 괄호가 필요합니다. 예를 들어 PI가 아닌 PI()를 항상 입력해야 합니다. 다른 함수 내에 함수를 중첩할 수도 있습니다.

- 식. 식에는 열에 대한 연산자, 상수 또는 참조 중 어느 것이나 모두 포함될 수 있습니다.

예를 들어 다음은 모두 유효한 수식입니다.

수식	결과
= 3	3
= "Sales"	Sales
= 'Sales'[Amount]	Sales 테이블 내에서 이 수식을 사용하면 현재 행에 대한 Sales 테이블의 Amount 열 값이 표시됩니다.
= (0.03 *[금액])	현재 테이블의 Amount 열에 있는 값의 3%입니다.
=0.03 * [금액]	이 수식을 사용하여 백분율을 계산할 수 있지만 표에 서식을 적용하지 않으면 결과가 백분율로 표시되지 않습니다.

수식	결과
= PI()	상수 pi의 값입니다.

수식은 사용되는 방식에 따라 다르게 동작할 수 있습니다. 수식에서 사용하는 데이터가 계산에 사용될 수 있는 다른 데이터와 어떻게 관련되어 있는지 항상 알고 있어야 합니다.

명명 요건

데이터 모델에는 종종 여러 테이블이 포함됩니다. 테이블과 해당 열은 메모리 내 분석 엔진(VertiPaq)에 저장된 데이터베이스로 구성됩니다. 해당 데이터베이스 내에서 모든 테이블에는 고유한 이름이 있어야 합니다. 열 이름도 각 테이블 내에서 고유해야 합니다. 모든 개체 이름은 대/소문자를 구분하지 않습니다. 예를 들어 SALES와 Sales 이름은 동일한 테이블을 나타냅니다.

기존 데이터 모델에 추가하는 각 열과 측정값은 특정 테이블에 속해야 합니다. 테이블 내에서 계산 열을 만들 때 암시적으로 열이 포함된 테이블을 지정하거나 측정값을 만들고 측정값 정의를 저장할 테이블의 이름을 지정할 때 명시적으로 지정합니다.

테이블 또는 열을 함수에 대한 입력으로 사용하는 경우 일반적으로 열 이름을 한정 해야 합니다. 열의 정규화된 이름은 테이블 이름 뒤에 대괄호로 된 열 이름(예: '미국 판매'[Products])입니다. 다음 컨텍스트에서 열을 참조할 때는 항상 정규화된 이름이 필요합니다.

- 함수에 대한 인수인 VALUES
- 함수, ALL 또는 ALLEXCEPT에 대한 인수로
- 함수에 대한 필터 인수에서 CALCULATE 또는 CALCULATETABLE
- FUNCTION에 대한 인수로 RELATEDTABLE
- 임의의 시간 인텔리전스 함수에 대한 인수로

정규화되지 않은 열 이름은 대괄호로 묶인 열의 이름입니다(예: [Sales Amount]). 예를 들어 현재 테이블의 동일한 행에서 스칼라 값을 참조하는 경우 정규화되지 않은 열 이름을 사용할 수 있습니다.

테이블 이름에 공백, 예약된 키워드(keyword) 또는 허용되지 않는 문자가 포함된 경우 테이블 이름을 작은따옴표로 묶어야 합니다. 또한 로캘이 문자 집합을 지원하는지 여부에 관계없이 이름에 ANSI 영숫자 문자 범위 외부의 문자가 포함된 경우 테이블 이름을 따옴표로 묶어야 합니다. 예를 들어 'Талиа'와 같이 키릴 자모 문자로 작성된 테이블 이름이 포함된 통합 문서를 여는 경우 공백이 없더라도 테이블 이름을 따옴표로 묶어야 합니다.

① 참고

열의 정규화된 이름을 더 쉽게 입력하려면 수식 편집기에서 자동 완성 기능을 사용합니다.

테이블

- 현재 테이블과 다른 테이블에서 열이 있을 때마다 테이블 이름이 필요합니다. 테이블 이름은 데이터베이스 내에서 고유해야 합니다.
- 공백, 기타 특수 문자 또는 영어가 아닌 영숫자가 포함된 경우 테이블 이름은 작은따옴표로 묶어야 합니다.

측정값 그룹

- 측정값 이름은 항상 대괄호 안에 있어야 합니다.
- 측정값 이름에는 공백이 포함될 수 있습니다.
- 각 측정값 이름은 모델 내에서 고유해야 합니다. 따라서 테이블 이름은 기존 측정값을 참조할 때 측정값 이름 앞에 선택 사항입니다. 그러나 측정값을 만들 때는 항상 측정값 정의가 저장될 테이블을 지정해야 합니다.

열

열 이름은 테이블의 컨텍스트에서 고유해야 합니다. 그러나 여러 테이블에 이름이 같은 열이 있을 수 있습니다(명확성은 테이블 이름과 함께 제공됨).

일반적으로 열은 이름 충돌을 해결하거나 열 이름을 정규화해야 하는 특정 함수를 사용하는 경우를 제외하고 해당 열이 속한 기본 테이블을 참조하지 않고 참조할 수 있습니다.

예약 키워드

테이블에 사용하는 이름이 Analysis Services 예약 키워드(keyword) 동일한 경우 오류가 발생하고 테이블 이름을 바꿔야 합니다. 그러나 개체 이름이 대괄호(열의 경우) 또는 따옴표(테이블의 경우)로 묶인 경우 개체 이름에 키워드(keyword) 사용할 수 있습니다.

① 참고

따옴표는 애플리케이션에 따라 여러 다른 문자로 나타낼 수 있습니다. 외부 문서 또는 웹 페이지에서 수식을 붙여넣는 경우 따옴표를 열고 닫는 데 사용되는 문자의

ASCII 코드를 검사 동일한지 확인합니다. 그렇지 않으면 DAX가 기호를 따옴표로 인식할 수 없어 참조가 유효하지 않을 수 있습니다.

특수 문자

다음 문자 및 문자 형식은 테이블, 열 또는 측정값의 이름에서 유효하지 않습니다.

- 선행 또는 후행 공백; 공백이 이름 구분 기호, 대괄호 또는 단일 아포스트로피로 묶이지 않는 한
- 컨트롤 문자
- 개체 이름에 유효하지 않은 문자는 다음과 같습니다.

.,:^*!&%\$!+=()[]{}<>

개체 이름의 예

다음 표에서는 일부 개체 이름의 예를 보여 줍니다.

개체 유형	예제	설명
테이블 이름	Sales	테이블 이름에 공백이나 기타 특수 문자가 없으면 이름을 따옴표로 묶을 필요가 없습니다.
테이블 이름	'캐나다 판매'	이름에 공백, 탭 또는 기타 특수 문자가 포함된 경우 이름을 작은따옴표로 묶습니다.
정규화된 열 이름	판매액[금액]	테이블 이름은 열 이름 앞에 있고 열 이름은 대괄호로 묶습니다.
정규화된 측정값 이름	판매액[이익]	테이블 이름은 측정값 이름 앞에 있으며 측정값 이름은 대괄호로 묶습니다. 특정 컨텍스트에서는 정규화된 이름이 항상 필요합니다.
정규화되지 않은 열 이름	[금액]	정규화되지 않은 이름은 대괄호로 된 열 이름입니다. 정규화되지 않은 이름을 사용할 수 있는 컨텍스트에는 동일한 테이블 내의 계산 열 또는 동일한 테이블을 검색하는 집계 함수에 수식이 포함됩니다.
공백이 있는 테이블의 정규화된 열	'Canada Sales'[Qty]	테이블 이름은 공백을 포함하므로 작은따옴표로 묶어야 합니다.

기타 제한 사항

각 함수에 필요한 구문과 수행할 수 있는 작업 유형은 함수에 따라 크게 달라집니다. 그러나 일반적으로 다음 규칙은 모든 수식 및 식에 적용됩니다.

- DAX 수식 및 식은 테이블에 개별 값을 수정하거나 삽입할 수 없습니다.
- DAX를 사용하여 계산된 행을 만들 수 없습니다. 계산 열 및 측정값만 만들 수 있습니다.
- 계산 열을 정의할 때 함수를 모든 수준에 중첩할 수 있습니다.
- DAX에는 테이블을 반환하는 여러 함수가 있습니다. 일반적으로 이러한 함수에서 반환하는 값을 다른 함수에 대한 입력으로 사용하며, 테이블은 입력으로 필요합니다.

DAX 연산자 및 상수

다음 표에서는 DAX에서 지원하는 연산자를 나열합니다. 개별 연산자의 구문에 대한 자세한 내용은 [DAX 연산자를 참조하세요](#).

연산자 유형	기호 및 사용
괄호 연산자	() 인수의 우선 순위 및 그룹화
산술 연산자	+ (더하기) - (빼기)/ sign) * (곱하기) / (나누기) ^(지수)
비교 연산자	= (같음) >(보다 큼) <(보다 작음) >=(보다 크거나 같음) <=(보다 작거나 같음) <>(같지 않음)
텍스트 연결 연산자	&(연결)

연산자 유형	기호 및 사용
논리 연산자	&&(및)
	(또는)

DAX 수식에서 사용하는 열 또는 값의 데이터 형식을 캐스팅, 변환 또는 지정할 필요가 없습니다. DAX 수식에서 데이터를 사용하는 경우 DAX는 참조된 열의 데이터 형식과 입력한 값의 데이터 형식을 자동으로 식별하고 지정된 작업을 완료하는 데 필요한 경우 암시적 변환을 수행합니다.

예를 들어 날짜 값에 숫자를 추가하려고 하면 엔진은 함수 컨텍스트에서 작업을 해석하고 숫자를 공통 데이터 형식으로 변환한 다음 원하는 형식인 날짜로 결과를 표시합니다.

그러나 성공적으로 변환할 수 있는 값에는 몇 가지 제한 사항이 있습니다. 값 또는 열에 현재 작업과 호환되지 않는 데이터 형식이 있는 경우 DAX는 오류를 반환합니다. 또한 DAX는 가져온 기존 데이터의 데이터 형식을 데이터 모델로 명시적으로 변경, 변환 또는 캐스팅할 수 있는 함수를 제공하지 않습니다.

① 중요

DAX는 변형 데이터 형식의 사용을 지원하지 않습니다. 따라서 데이터를 데이터 모델로 로드하거나 가져올 때 각 열의 데이터는 일반적으로 일관된 데이터 형식이 될 것으로 예상됩니다.

일부 함수는 문자열을 포함한 스칼라 값을 반환하는 반면, 다른 함수는 정수와 실수, 날짜 및 시간 모두와 함께 작동합니다. 각 함수에 필요한 데이터 형식은 DAX 함수 섹션 [에 설명되어 있습니다](#).

여러 열과 여러 행의 데이터를 포함하는 테이블을 함수의 인수로 사용할 수 있습니다. 일부 함수는 메모리에 저장되고 다른 함수에 대한 인수로 사용할 수 있는 테이블을 반환합니다.

날짜 및 시간

DAX는 Microsoft SQL Server에서 사용하는 날짜/시간 데이터 형식을 사용하여 날짜 및 시간 값을 저장합니다. Datetime 형식은 부동 소수점 숫자를 사용합니다. 여기서 날짜 값은 1899년 12월 30일 이후의 일 수를 나타내는 정수 부분에 해당합니다. 시간 값은 시간, 분 및 초가 하루의 소수 자릿수로 표시되는 날짜 값의 소수 부분에 해당합니다. DAX 날짜 및 시간 함수는 인수를 datetime 데이터 형식으로 암시적으로 변환합니다.

① 참고

DAX에서 지원하는 정확한 최대 DateTime 값은 9999년 12월 31일 00:00:00입니다.

날짜 및 시간 리터럴

Power BI Desktop의 2021년 8월 버전부터 DAX 날짜 및 날짜/시간 값을 형식 `dt"YYYY-MM-DD"` `dt"YYYY-MM-DDThh:mm:ss"` `dt"YYYY-MM-DD hh:mm:ss"` 또는 리터럴로 지정할 수 있습니다. 리터럴로 지정된 경우 식에서 `DATE`, `TIME`, `DATEVALUE`, `TIMEVALUE` 함수를 사용할 필요가 없습니다.

예를 들어 다음 식은 DATE 및 TIME 함수를 사용하여 OrderDate를 필터링합니다.

DAX

```
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > (DATE(2015,1,9) + TIME(2,30,0)) && [OrderDate] <  
    (DATE(2015,12,31) + TIME(11,59,59))  
)
```

동일한 필터 식을 리터럴로 지정할 수 있습니다.

DAX

```
EVALUATE  
FILTER (  
    FactInternetSales,  
    [OrderDate] > dt"2015-1-9T02:30:00" && [OrderDate] < dt"2015-12-  
31T11:59:59"  
)
```

① 참고

DAX 날짜 및 날짜/시간 형식 리터럴 형식은 Excel의 모든 버전의 Power BI Desktop, Analysis Services 및 Power Pivot에서 지원되지 않습니다. 새 DAX 및 업데이트된 DAX 기능은 일반적으로 Power BI Desktop에서 처음 도입된 다음 나중에 Excel의 Analysis Services 및 파워 피벗에 포함됩니다.

이 페이지가 도움이 되었나요?

 Yes

 No