



CẨM NANG HƯỚNG DẪN SỬ DỤNG DAX VỚI POWER BI

TUNG PHAM

ICT24H GROUP | HO CHI MINH CITY

LỜI GIỚI THIỆU

Chào các bạn,

Ý tưởng về việc viết cẩm nang này xuất phát trong quá trình đào tạo cũng như triển khai các hệ thống báo cáo BI tại các doanh nghiệp. DAX được xem như công cụ đầy sức mạnh nhưng cũng được ghi nhận là món khó nhằn nhất khi vận dụng vào việc xây dựng các báo cáo. Theo quan điểm cá nhân của tôi, DAX chính là trái tim của Power BI cũng như Excel cũng không còn là Excel nữa nếu mất đi những bộ hàm tính toán đầy sức mạnh của mình.

Từ đấy, tôi quyết định viết cẩm nang này theo hướng mô tả từng nhóm hàm, cách thức hoạt động, giá trị trả về, ví dụ thực tế,... theo cách trực quan nhất để bất cứ ai cũng có thể tham khảo được.

Những thông tin được thu thập và biên soạn trong cẩm nang này thật ra đã được phổ biến trên thế giới bằng Anh ngữ, thông qua các công cụ tìm kiếm trên mạng Internet. Tuy nhiên, trên cơ sở những mục tiêu nhắm đến khi biên soạn nên công việc của tôi là cố gắng hệ thống các nguồn tư liệu tham khảo sao cho chúng có thể trở nên dễ dùng nhất đối với người đọc, cũng như trình bày chúng theo cách dễ hiểu, dễ tiếp cận nhất để bất cứ ai quan tâm đến DAX đều có thể sử dụng được mà không đòi hỏi phải có một nền tảng kiến thức sâu rộng về BI.

Đây là một quá trình khó khăn và phức tạp - nhất là đối với người chuyên về kỹ thuật như tôi. Vì thế, tôi rất mong nhận được sự cảm thông của bạn đọc gần xa về những sai sót nếu có trong quá trình biên soạn, cũng như chân thành đón nhận và lắng nghe mọi ý kiến đóng góp xây dựng để phiên bản kế tiếp của cuốn cẩm nang này được hoàn thiện hơn.

Chúng tôi hy vọng cẩm nang này sẽ giúp ích các bạn trong suốt quá trình tiếp cận làm quen cũng như khai thác sử dụng DAX trong công tác phân tích, xây dựng báo cáo tại doanh nghiệp.

Trân trọng,

TỔNG QUAN VỀ DAX TRONG POWER BI

Khi làm quen với Power BI, các bạn sẽ nghe nói nhiều về các hàm DAX, việc nắm vững DAX sẽ giúp bạn xây dựng & kiểm soát rất tốt các trường dữ liệu trước khi tiến hành tạo các báo cáo.

Vậy DAX là gì? Dùng làm gì & dùng như thế nào? Có sự khác biệt gì với hàm trong Excel hay các hàm MDX, các câu lệnh T-SQL trong SQL,...

Từ các câu hỏi đấy, dataMAKER tổng hợp lại thành những câu trả lời ngắn gọn sau. Hy vọng sẽ giúp các bạn có cái nhìn tổng quan về bộ hàm hữu ích này.

1. DAX là gì?

Cơ bản DAX là một tập hợp các hàm, toán tử và hằng số có thể được sử dụng trong công thức hoặc biểu thức để tính và trả về một hoặc nhiều giá trị. Hiểu theo cách đơn giản hơn, DAX giúp tạo thông tin mới từ dữ liệu đã có trong data model của bạn.

2. Tại sao DAX lại quan trọng đến như vậy?

Bộ hàm DAX rất đa dạng, từ những hàm tính toán cơ bản đến những hàm hỗ trợ khả năng phân tích giá trị theo thời gian, hỗ trợ tạo dựng các mối liên kết giữa các trường thông tin ở các bảng dữ liệu khác nhau sẽ giúp bạn giải những bài toán đa chiều phức tạp mà đôi khi việc xử lý thông qua các tính năng trên giao diện không thể thực hiện được hay phải sử dụng đến ngôn ngữ M phức tạp không phù hợp cho người dùng không phải là chuyên gia IT.

Khi bạn biết dùng DAX linh hoạt sẽ giúp ích rất nhiều trong việc bóc tách dữ liệu, tối ưu hiệu suất hệ thống – nên dùng DAX cho measure hay Calculated Column.

Tóm lại, nếu đã dùng Power BI thì bạn phải học – hiểu về DAX.

Lợi thế rất lớn cho các bạn đã thành thục các hàm trong Excel, đây là cầu nối rất tốt để có thể tiếp cận DAX.

3. Vậy DAX trong Power BI có khác gì với các hàm trong Excel?

Thư viện hàm DAX dựa trên thư viện hàm Excel, nhưng những thư viện này có nhiều điểm khác biệt. Phần này tóm tắt những điểm khác biệt và tương đồng giữa các hàm Excel và các hàm DAX.

- Nhiều hàm DAX có cùng tên và cùng hành vi chung với các hàm Excel nhưng đã được chỉnh sửa để nhận những kiểu dữ liệu đầu vào khác, và trong một số trường hợp, có thể trả về kiểu dữ liệu khác. Nhìn chung, bạn sẽ không dùng được hàm DAX trong công thức Excel hoặc không dùng được công thức Excel trong Power Pivot nếu không thực hiện một vài chỉnh sửa.
- Các hàm DAX không bao giờ lấy tham chiếu ô hoặc phạm vi làm tham chiếu, mà thay vào đó các hàm DAX lấy cột hoặc bảng làm tham chiếu.
- Hàm ngày và thời gian trong DAX trả về kiểu dữ liệu datetime. Ngược lại, các hàm ngày và thời gian trong Excel trả về một số nguyên biểu thị cho ngày dưới dạng số sê-ri.
- Nhiều hàm DAX mới trả về một bảng chứa các giá trị hoặc thực hiện các phép tính dựa trên dữ liệu đầu vào là một bảng chứa các giá trị. Ngược lại, Excel không có hàm nào trả về bảng, nhưng một số hàm có thể hoạt động với các mảng. Khả năng dễ dàng tham chiếu đến các cột và bảng hoàn tất là một tính năng mới trong Power Pivot.
- DAX cung cấp các hàm tra cứu mới, tương tự như các hàm tra cứu mảng và véc-tơ trong Excel. Tuy nhiên, các hàm DAX đòi hỏi phải thiết lập mối quan hệ giữa các bảng.
- Dữ liệu trong một cột được dự kiến là sẽ luôn có cùng một kiểu dữ liệu. Nếu dữ liệu không cùng một kiểu, thì DAX sẽ thay đổi toàn bộ cột thành kiểu dữ liệu có thể chứa đựng tốt nhất tất cả các giá trị.

4. Muốn tìm hiểu DAX tôi phải bắt đầu từ đâu?

Để viết 1 hàm DAX bạn cần tìm hiểu về ý nghĩa, cú pháp, cách khai báo toán tử trong cú pháp để cho ra kết quả với ngữ cảnh. Ở bài này dataMAKER sẽ không đi theo lối mòn liệt kê tất cả các hàm, cú pháp một cách cứng nhắc. Cách tiếp cận này chỉ làm cho bạn cảm thấy sự khô khan, cứng nhắc trong cách viết hàm DAX. Trong những bài sau, dataMAKER sẽ đi theo hướng viết DAX trong một ngữ cảnh thực tế để từ đó bạn dễ hình dung và áp dụng vào data model của mình.

Ngoài ra, các bạn có thể tham khảo thêm các bài mô tả chi tiết cách thức vận hành dựa trên những ví dụ cụ thể do dataMAKER trích lọc để khai thác tốt hơn.

Trong tài liệu này, dataMAKER sẽ tập trung mô tả các hàm thường dùng trước và sẽ mở rộng về sau cho các hàm DAX khác. Các bạn lưu ý cập nhật để có phiên bản mới nhất.

Nội dung

LỜI GIỚI THIỆU	2
TỔNG QUAN VỀ DAX TRONG POWER BI	3
1. DAX là gì?	3
2. Tại sao DAX lại quan trọng đến như vậy?	3
3. Vậy DAX trong Power BI có khác gì với các hàm trong Excel?	3
4. Muốn tìm hiểu DAX tôi phải bắt đầu từ đâu?.....	4
TỔNG HỢP BỘ HÀM DAX	8
1. Nhóm hàm toán học (Math & Trig Functions)	8
SUMX()	8
RAND() & RANDBETWEEN()	8
DIVIDE()	9
2. Nhóm hàm thống kê (Statistical Functions)	10
COUNT ()	11
COUNTA()	11
COUNTBLANK()	12
COUNTROWS()	12
COUNTX()	12
DISTINCTCOUNT()	13
CROSSJOIN()	13
MIN()	14
MINX()	14
MAX()	15
MAXX()	15
RANKX()	16
AVERAGE()	17
AVERAGEX()	17
ADDCOLUMNS()	18
SUMMARIZE()	19
3. Nhóm hàm Thời gian (Date and Time Functions)	20
DATE()	20
DATEVALUE()	20
TIMEVALUE()	21

TODAY()	21
WEEKNUM()	21
EDATE()	22
EOMONTH()	22
DAY()	23
MONTH()	23
YEAR()	24
DATEADD()	24
DATESBETWEEN()	25
DATESINPERIOD()	26
DATESMTD()	27
DATESMTD()	27
DATESQTD()	28
DATESYTD()	28
ENDOFMONTH()	29
ENDOFQUARTER()	29
ENDOFYEAR()	30
FIRSTDAY()	30
FIRSTNONBLANK()	31
LASTDAY()	31
LASTNONBLANK()	32
NEXTDAY()	32
NEXTMONTH()	33
NEXTQUARTER()	33
NEXTYEAR()	34
PARALLELPERIOD()	34
PREVIOUSDAY()	35
PREVIOUSMONTH()	36
PREVIOUSQUARTER()	36
PREVIOUSYEAR()	37
SAMEPERIODLASTYEAR()	37
STARTOFMONTH()	38
STARTOFQUARTER()	38

STARTOFYEAR()	38
4. Nhóm hàm Bộ lọc (Filter Functions)	40
CALCULATE()	40
ALL()	41
ALLEXCEPT()	41
ALLNOBLANKROW()	42
FILTER()	43
DISTINCT()	43
EARLIER()	44
HASONEFILTER()	44
HASONEVALUE()	45
VALUES ()	46
RELATED()	46
5. Nhóm hàm Thông tin (Information Functions)	47
CONTAINS()	47
ISBLANK()	47
ISERROR()	48
ISNUMBER()	48
ISTEXT()	49
LOOKUPVALUE()	50
6. Nhóm hàm Logic (Logical Functions)	51
AND()	51
OR()	51
IF()	52
SWITCH()	53
NOT()	54
IFERROR()	54
7. Nhóm hàm Text (Text Functions)	56
BLANK()	56
FIND()	56
CONCATENATE()	57

TỔNG HỢP BỘ HÀM DAX

1. Nhóm hàm toán học (Math & Trig Functions)

Mô tả: Các hàm toán học trong DAX tương tự như các hàm toán học và lượng giác của Excel. Tuy nhiên, có một số khác biệt trong các kiểu dữ liệu số được các hàm DAX sử dụng.

SUMX()

Mục đích: Tính tổng từng dòng trên bảng tham chiếu dựa vào phương thức tính toán được khai báo.

Cú pháp: **SUMX(<table>, <expression>)**

Tham số	Mô tả
table	khai báo thông tin bảng chứa dữ liệu cần tính toán
expression	khai báo các phương thức tính toán, có thể kết hợp các hàm filter

Giá trị trả về: giá trị thập phân

VD: thực hiện tính toán Tổng tiền dựa trên 2 thông tin: **Số lượng** & **Giá bán** trong bảng Banhang:

**SUMX('Banhang',
'Banhang'[Số lượng] * 'Banhang'[Giá bán])**

RAND() & RANDBETWEEN()

Mục đích đều dùng để tạo một giá trị ngẫu nhiên, thường dùng trong trường hợp cần giả lập giá trị để thử nghiệm các tính toán. Vì 2 hàm này có mục đích sử dụng như nhau & không khác biệt nhau nhiều về cú pháp nên dataMAKER sẽ gom chung như sau:

RAND()

Mục đích: dùng để tạo giá trị ngẫu nhiên trong khoảng giá trị cho phép: ≥ 0 && < 1 .

Cú pháp: **RAND()**

Kết quả trả về: số thập phân.

VD: **RAND()**

-> Kết quả trả về số sẽ là số ngẫu nhiên trong dãy số từ 0 ->1.

RANDBETWEEN()

Mục đích: dùng tạo giá trị ngẫu nhiên trong khoảng giá trị được người dùng nhập liệu.

Cú pháp: **RANDBETWEEN(value 1, value 2)**

Kết quả trả về: dạng số.

VD: **RANDBETWEEN(1,100)**

-> Kết quả trả về số sẽ là số ngẫu nhiên trong dãy số từ 1 -> 100.

Khả năng mở rộng:

Hai hàm trên có thể dùng kết hợp với các công thức tính toán hay các hàm khác, thậm chí bạn có thể kết hợp 2 hàm này với nhau.

DIVIDE()

Mục đích: thực hiện phép chia

Cú pháp: **DIVIDE(<numerator>,<denominator>[,<alternateresult>])**

Tham số	Mô tả
numerator	Khai báo tử số của hàm
denominator	Khai báo mẫu số cần chia
alternateresult	Giá trị trả về trong trường hợp chia cho giá trị 0, nếu không khai báo giá trị trả về mặc định là BLANK()

Kết quả trả về: số thập phân

VD: **DIVIDE(5,2)**

-> Kết quả trả về: 2.5

ROUND()

Mục đích: hàm thực hiện làm tròn giá trị.

Cú pháp: **ROUND(<number>, <num_digits>)**

Tham số	Mô tả
number	Khai báo số bạn muốn làm tròn thực hiện việc làm tròn giá trị.
num_digits	Được linh hoạt khai báo tùy theo giá trị mong muốn như sau: <ul style="list-style-type: none">– Giá trị > 0: làm tròn giá trị thập phân– Giá trị = 0: làm tròn đến giá trị gần nhất– Giá trị < 0: làm tròn theo giá trị đầu tiên

Kết quả trả về: số đã được làm tròn theo tham số num_digits.

VD1: Trường hợp giá trị tham số truyền vào > 0:

ROUND(29.09, 1)

-> Kết quả trả về: 29.10

VD2: Trường hợp giá trị tham số truyền vào = 0

ROUND(29.09, 0)

-> Kết quả trả về: 29.00

VD3: Trường hợp giá trị tham số truyền vào < 0

ROUND(29.09, -1)

-> Kết quả trả về: 30.00

2. Nhóm hàm thống kê (Statistical Functions)

Mô tả: nhóm hàm này thực hiện các kết hợp. Ngoài việc tạo tổng và số trung bình, hoặc tìm các giá trị tối thiểu và tối đa, bạn cũng có thể lọc cột trước khi tổng hợp hoặc tạo các tập hợp dựa trên các bảng có liên quan.

COUNT()

Mục đích: đếm số ô trong cột chứa giá trị số.

Cú pháp: **COUNT(<column>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị số cần thực hiện đếm.

Giá trị trả về: trả về số dòng đếm được với kiểu định dạng số nguyên.

Ví dụ: Đếm giá trị số trong cột Ngày bán:

COUNT('Sale Datamaker'[Ngày bán])

COUNTA()

Mục đích: đếm số lượng ô trong cột không trống. Không chỉ đếm các hàng chứa giá trị số, mà còn bao gồm văn bản, ngày tháng và các giá trị logic.

Cú pháp: **COUNTA(<column>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị số cần thực hiện đếm.

Giá trị trả về: trả về số dòng đếm được với kiểu định dạng số nguyên.

Ví dụ: Đếm số sản phẩm trong cột Sản phẩm – không đếm trường rỗng.

COUNTA('Sale Datamaker'[San pham])

COUNTBLANK()

Mục đích: đếm số lượng ô trống trong cột chỉ định.

Cú pháp: **COUNTBLANK(<column>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị số cần thực hiện đếm.

Giá trị trả về: trả về số dòng đếm được với kiểu định dạng số nguyên. Nếu không có dòng nào thỏa điều kiện sẽ trả về giá trị rỗng.

Ví dụ: Đếm số sản phẩm trong cột Sản phẩm – đếm trường rỗng.

COUNTBLANK('Sale Datamaker'[San pham])

COUNTROWS()

Mục đích: đếm số lượng dòng có giá trị trong cột chỉ định.

Cú pháp: **COUNTROWS(<table>)**

Tham số	Mô tả
table	Khai báo bảng chứa dòng có giá trị số cần thực hiện đếm.

Giá trị trả về: trả về số dòng đếm được với kiểu định dạng số nguyên.

Ví dụ: Đếm số sản phẩm trong bảng SALE:

COUNTROWS('Sale Datamaker')

COUNTX()

Mục đích: đếm số lượng dòng có giá trị trong cột chỉ định.

Cú pháp: **COUNTX(<table>, <expression>)**

Tham số	Mô tả
table	Khai báo bảng chứa dòng có giá trị số cần thực hiện đếm.
expression	Khai báo biểu thức trả về nhóm giá trị có chứa giá trị cần đếm.

Giá trị trả về: trả về số dòng đếm được.

Ví dụ: Đếm tất cả các dòng sản phẩm có định nghĩa giá bán:

COUNTX('Sale Datamaker', [Giá bán])

DISTINCTCOUNT()

Mục đích: đếm số lượng ô có giá trị trong cột chỉ định, chỉ đếm cột có giá trị khác nhau.

Cú pháp: **DISTINCTCOUNT(<column>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị số cần thực hiện đếm.

Giá trị trả về: trả về số dòng đếm các cột có giá trị duy nhất.

Ví dụ: Đếm số sản phẩm trong cột Sản phẩm – không đếm sản phẩm cùng tên.

DISTINCTCOUNT('Sale Datamaker'[San pham])

CROSSJOIN()

Mục đích: tạo bảng mới có giá trị từ các cột trong các bảng chỉ định.

Cú pháp: **CROSSJOIN(<table>,<table>[,<table>]...)**

Tham số	Mô tả
table	Khai báo các bảng dùng tạo bảng mới

Giá trị trả về: trả về bảng có giá trị từ các bảng chỉ định.

Ví dụ: Tôi tạo bảng mới từ 2 bảng Loại hình áp dụng (SMB, Enterprise) và bảng Tên sản phẩm (Office 365..)

CROSSJOIN(Business_Type, Product_Name)

MIN()

Mục đích: trả về giá trị nhỏ nhất trong cột, không áp dụng cho giá trị logic hoặc văn bản.

Cú pháp: **MIN(<column>)**

Tham số	Mô tả
column	Khai báo cột muốn tìm giá trị nhỏ nhất.

Giá trị trả về: trả về giá trị nhỏ nhất trong cột chỉ định.

Ví dụ: tôi thực hiện tìm giá trị doanh thu thấp nhất:

MIN('Sale Datamaker'[Doanh thu])

MINX()

Mục đích: trả về giá trị số nhỏ nhất có kết quả từ việc đánh giá một biểu thức cho mỗi dòng của một bảng.

Cú pháp: **MINX(<table>, <expression>)**

Tham số	Mô tả
---------	-------

table	Khai báo bảng muốn tìm giá trị nhỏ nhất.
expression	Thông tin biểu thức cần thực hiện kiểm tra trên từng dòng của bảng

Giá trị trả về: trả về dòng có giá trị nhỏ nhất.

Ví dụ: tôi thực hiện tìm giá trị lợi nhuận thấp nhất trong năm 2017:

MINX(

FILTER('Sale Datamaker'[Nam] =2017),[Lợi nhuận])

MAX()

Mục đích: trả về giá trị lớn nhất trong cột, không áp dụng cho giá trị logic hoặc văn bản.

Cú pháp: **MAX (<column>)**

Tham số	Mô tả
column	Khai báo cột muốn tìm giá trị lớn nhất.

Giá trị trả về: trả về giá trị lớn nhất trong cột chỉ định.

Ví dụ: tôi thực hiện tìm giá trị doanh thu lớn nhất:

MAX('Sale Datamaker'[Doanh thu])

MAXX()

Mục đích: trả về giá trị số nhỏ lớn có kết quả từ việc đánh giá một biểu thức cho mỗi dòng của một bảng.

Cú pháp: **MAXX(<table>, <expression>)**

Tham số	Mô tả
table	Khai báo bảng muốn tìm giá trị lớn nhất.
expression	Thông tin biểu thức cần thực hiện kiểm tra trên từng dòng của bảng

Giá trị trả về: trả về dòng có giá trị lớn nhất.

Ví dụ: tôi thực hiện tìm giá trị lợi nhuận lớn nhất trong năm 2017:

**MAXX(
 FILTER('Sale Datamaker'[Nam] =2017),[Loi nhuan])**

RANKX()

Mục đích: trả về thứ tự xếp hạng của một số trong danh sách các số cho mỗi dòng trong bảng.

Cú pháp: **RANKX(<table>, <expression>[,<value>[,<order>[,<ties>]]])**

Tham số	Mô tả
table	Bất kỳ biểu thức DAX nào trả về một bảng dữ liệu mà biểu thức được đánh giá.
expression	Bất kỳ biểu thức DAX nào trả về một giá trị vô hướng. Biểu thức được đánh giá cho mỗi hàng của bảng, để tạo ra tất cả các giá trị có thể có để xếp hạng
value	Bất kỳ biểu thức DAX nào trả về một giá trị vô hướng duy nhất có thứ hạng được tìm thấy. Khi tham số giá trị bị bỏ qua, giá trị của biểu thức tại dòng hiện tại được sử dụng thay thế.
order	Giá trị chỉ định cách xếp hạng giá trị, từ thấp đến cao hoặc cao đến thấp.

Giá trị trả về: danh sách thứ tự xếp hạng.

Ví dụ: tôi muốn xếp hạng khách hàng theo lợi nhuận bán hàng theo thứ tự từ cao đến thấp:

RANKX(

ALL('Sale Datamaker'[Khach hang], [Loi nhuan], ,Desc)

AVERAGE()

Mục đích: trả về giá trị trung bình (trung bình số học) của tất cả các số trong một cột.

Cú pháp: **AVERAGE(<column>)]**

Tham số	Mô tả
column	Cột chứa các số mà bạn muốn tính trung bình.

Giá trị trả về: giá trị trung bình các số trong cột chỉ định.

Ví dụ: tôi muốn tính giá trị trung bình doanh thu bán hàng:

AVERAGE('Sale Datamaker'[Doanh thu])

AVERAGEX()

Mục đích: tính trung bình (trung bình số học) của một tập hợp các biểu thức được đánh giá trên một bảng.

Cú pháp: **AVERAGEX(<table>], <expression>)**

Tham số	Mô tả
table	Khai báo bảng cần thực hiện tính trung bình
expression	Khai báo biểu thức cần đánh giá trên bảng.

Giá trị trả về: giá trị trung bình theo kiểu thập phân.

Ví dụ: tôi muốn tính giá trị trung bình của Giá bán với Thuế áp dụng theo từng mặt hàng:

```
AVERAGEX('Sale Datamaker',  
          'Sale Datamaker'[Gia ban] + 'Sale Datamaker'[VAT])
```

ADDCOLUMNS()

Mục đích: thêm cột vào bảng được chỉ định, cột thêm vào có thể là giá trị tham chiếu từ cột khác hoặc một biểu thức tính toán tạo giá trị theo dòng.

Cú pháp: **ADDCOLUMNS(<table>,<name>,<expression>[,<name>,<expression>]...)**

Tham số	Mô tả
table	Khai báo bảng cần thêm cột.
name	Khai báo tên cột mới.
expression	Khai báo biểu thức tính toán tạo giá trị theo từng dòng.

Giá trị trả về: cột mới được thêm vào bảng chỉ định.

Ví dụ: tôi muốn thêm cột giá gốc và thuế vào bảng Sale Datamaker:

```
ADDCOLUMNS('Sale Datamaker',  
            "Giá gốc",  
            SUMX(  
                RELATEDTABLE('BangGia Datamaker'), 'BangGia Datamaker'[Gia goc])  
            "Thuế",  
            SUMX(  
                RELATEDTABLE('BangGia Datamaker'), 'BangGia Datamaker'[VAT])
```

SUMMARIZE()

Mục đích: Trả về bảng tóm tắt cho tổng số được yêu cầu trên một nhóm các nhóm.

Cú pháp: **SUMMARIZE(<table>, <groupBy_columnName>[,<groupBy_columnName>]...
[,<name>,<expression>]...)**

Tham số	Mô tả
table	Khai báo bảng cần thêm cột.
groupBy_columnName	Tên đủ điều kiện của cột hiện tại được sử dụng để tạo nhóm tóm tắt dựa trên các giá trị được tìm thấy trong đó. Tham số này không thể là một biểu thức.
name	Tên được đặt cho cột mới, kèm theo dấu ngoặc kép.
expression	Bất kỳ biểu thức DAX nào trả về một giá trị vô hướng duy nhất, trong đó biểu thức được đánh giá nhiều lần (theo mỗi dòng / ngữ cảnh).

Giá trị trả về: bảng có các cột được chọn cho các đối số *groupBy_columnName* và các cột tóm tắt được tạo bởi các đối số tên.

Ví dụ: tôi muốn tạo bảng tóm tắt doanh thu bán hàng và giá trị giảm giá bán hàng được nhóm lại theo năm bán hàng và theo nhóm sản phẩm thực tế bán:

```
SUMMARIZE('Thongke Datamaker',
    'Date Datamaker[Nam],
    "Product Datamaker[Nhom san pham],
    "Doanh thu", SUMX(
        RELATEDTABLE('Sale Datamaker'), 'Sale Datamaker'[Doanh thu])
    "Giá giảm", SUMX(
        RELATEDTABLE('Sale Datamaker'), 'Sale Datamaker'[Giá giảm]))
```

3. Nhóm hàm Thời gian (Date and Time Functions)

Mô tả: Nhóm hàm thời gian khá phức tạp và được sử dụng nhiều nhất trong phân tích dữ liệu. Nhiều hàm ngày tháng và thời gian trong DAX rất giống với hàm ngày tháng và thời gian của Excel. Tuy nhiên, các hàm DAX sử dụng kiểu dữ liệu datetime và có thể lấy các giá trị từ một cột làm đối số. DAX cũng bao gồm một tập các chức năng thông minh phân tích về thời gian cho phép bạn thao tác dữ liệu bằng các khoảng thời gian, bao gồm ngày, tháng, quý và năm, sau đó xây dựng và so sánh các phép tính trong các chu kỳ thời gian.

Trong nhóm này, tôi sẽ đồng thời gom chung các hàm về Thời gian cơ bản (Date & Time) cùng nhóm hàm đặc biệt (Time Intelligence).

DATE()

Mục đích: tạo ra ngày dựa trên tham số năm/tháng/ngày được truyền vào theo định dạng datetime chuẩn.

Cú pháp: **DATE(<year>, <month>, <day>)**

Giá trị trả về: trả về ngày cụ thể.

Ví dụ: Tôi muốn tạo ngày 29/09/1988

DATE(1988,9,29)

DATEVALUE()

Mục đích: chuyển định dạng date chuẩn cho cột có giá trị văn bản.

Cú pháp: **DATEVALUE(date_text)**

Giá trị trả về: trả về ngày với định dạng date chuẩn.

Ví dụ: Tôi có cột thời gian (Date) được Power BI nhận diện theo định dạng text khi import, thực hiện chuyển đổi định dạng như sau:

DATEVALUE('Date Datamaker')

TIMEVALUE()

Mục đích: chuyển định dạng time chuẩn cho cột có giá trị văn bản.

Cú pháp: **TIMEVALUE(time_text)**

Giá trị trả về: trả về thời gian với định dạng time chuẩn.

Ví dụ: Tôi muốn trả về đúng định dạng time chuẩn cho chuỗi sau:

TIMEVALUE("12:49:55")

TODAY()

Mục đích: lấy giá trị ngày hiện hành.

Cú pháp: **TODAY()**

Giá trị trả về: trả về ngày hiện hành.

Ví dụ: Tôi muốn biết tuổi của người có năm sinh 1980 ở thời điểm hiện tại:

YEAR(TODAY()) - 1980

WEEKNUM()

Mục đích: lấy thông tin số tuần dựa trên thông tin ngày đưa vào.

Cú pháp: **WEEKNUM(<date>, <return_type>)**

Tham số	Mô tả
date	Khai báo ngày cần lấy thông tin số tuần
return_type	Có 2 kiểu sau: 1: ngày đầu tiên của tuần làm việc là Chủ Nhật 2: ngày đầu tiên của tuần làm việc là Thứ Hai

Giá trị trả về: số tuần thực tế của ngày chỉ định.

Ví dụ: Tôi muốn biết số tuần theo các ngày bán hàng:

WEEKNUM('Sale Datamaker'[Ngày bán])

EDATE()

Mục đích: trả về ngày với số tháng được chỉ định trước. Sử dụng EDATE để tính ngày đáo hạn hoặc ngày đến hạn rơi vào cùng ngày trong tháng chỉ định.

Cú pháp: **EDATE(<start_date>, <months>)**

Tham số	Mô tả
start_date	Khai báo ngày chỉ định cần tham chiếu
months	Khai báo số tháng cần xác định

Giá trị trả về: giá trị theo định dạng ngày.

Ví dụ: Tôi có ngày bán hàng như sau: 01/01/2017 -> 12/31/2017, vậy hàm sau sẽ trả về khoảng thời gian sau: 06/01/2017 -> 31/05/2018

EDATE('Sale Datamaker'[Ngày bán],5)

EOMONTH()

Mục đích: trả về ngày theo định dạng ngày giờ của ngày cuối cùng của tháng được chỉ định. Sử dụng EOMONTH để tính ngày đáo hạn hoặc ngày đến hạn rơi vào ngày cuối cùng của tháng.

Cú pháp: **EOMONTH(<start_date>, <months>)**

Tham số	Mô tả
---------	-------

start_date	Khai báo ngày cần cần tham chiếu
months	Khai báo số tháng cần xác định

Giá trị trả về: giá trị ngày cuối tháng theo định dạng ngày.

Ví dụ: Tôi có ngày bán hàng như sau: 01/01/2017 -> 12/31/2017, vậy hàm sau sẽ chỉ trả về ngày cuối tháng trong từng tháng: 04/30/2017 -> 31/05/2018

EOMONTH('Sale Datamaker'[Ngày ban],3)

DAY()

Mục đích: trả về giá trị ngày của ngày chỉ định, giá trị trả về trong khoảng thời gian từ ngày 01 đến 31. 12.

Cú pháp: **DAY(<date>)**

Tham số	Mô tả
date	Khai báo ngày cần lấy giá trị năm

Giá trị trả về: giá trị là số nguyên trong khoảng thời gian từ ngày 01 đến 31.

Ví dụ: Hàm sau trả về ngày 07:

DAY("05,07,2018")

MONTH()

Mục đích: trả về giá trị tháng của ngày chỉ định, giá trị trả về trong khoảng thời gian từ tháng 01 đến tháng 12.

Cú pháp: **MONTH(<date>)**

Tham số	Mô tả
date	Khai báo ngày cần lấy giá trị năm

Giá trị trả về: giá trị là số nguyên trong khoảng thời gian từ tháng 01 đến tháng 12

Ví dụ: Hàm sau trả về tháng 05:

MONTH("05,07,2018")

YEAR()

Mục đích: trả về giá trị năm của ngày chỉ định, giá trị trả về trong khoảng thời gian từ năm 1900 đến 9999.

Cú pháp: **YEAR(<date>)**

Tham số	Mô tả
date	Khai báo ngày cần lấy giá trị năm

Giá trị trả về: giá trị là số nguyên trong khoảng thời gian từ 1900 đến 9999.

Ví dụ: Hàm sau trả về năm 2018:

YEAR("05,07,2018")

DATEADD()

Mục đích: trả về một bảng có chứa giá trị theo định dạng ngày chuẩn, giá trị sẽ tăng hoặc giảm dựa vào tham số **number_of_intervals** được đưa vào theo ngữ thực tế.

Cú pháp: **DATEADD(<dates>, <number_of_intervals>, <interval>)**

Tham số	Mô tả
---------	-------

date	Khai báo cột chứa giá trị ngày
number_of_intervals	Khai báo số nguyên xác định số lượng khoảng thời gian để cộng vào hoặc trừ đi
interval	Khoảng thời gian muốn thay đổi ngày tháng, hỗ trợ theo: year, quarter, month, day

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Hàm sau trả về ngày/tháng/năm của năm trước ngày trong ngữ cảnh hiện tại:

DATEADD('Sale Datamaker'[Ngày bán], -1, YEAR)

DATESBETWEEN()

Mục đích: Trả về bảng chứa cột ngày bắt đầu bằng **start_date** và tiếp tục cho đến **end_date**.

Cú pháp: **DATESBETWEEN(<dates>, <start_date>, <end_date>)**

Tham số	Mô tả
dates	Khai báo cột ngày cần tham chiếu
start_date	Khai báo biểu thức cho ngày bắt đầu
end_date	Khai báo biểu thức cho ngày kết thúc

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi thực hiện tính doanh số bán hàng trong 3 tháng đầu năm 2018:

CALCULATE(

SUM('Sale Datamaker'[Doanh thu]),

DATESBETWEEN('Sale Datamaker'[Ngày bán],

**DATE(2018,1,1),
DATE(2018,3,31)))**

DATESINPERIOD()

Mục đích: trả về một bảng có chứa giá trị theo định dạng ngày chuẩn có ngày bắt đầu bằng **start_date** giá trị sẽ tăng hoặc giảm dựa vào tham số **number_of_intervals** được đưa vào theo ngữ thực tế.

Cú pháp: **DATESINPERIOD(<dates>,<start_date>,<number_of_intervals>,<interval>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày
start_date	Khai báo biểu thức cho ngày bắt đầu
number_of_intervals	Khai báo số nguyên xác định số lượng khoảng thời gian để cộng vào hoặc trừ đi
interval	Khoảng thời gian muốn thay đổi ngày tháng, hỗ trợ theo: year, quarter, month, day

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Ngày chốt kỳ kế toán của công ty là 31/5/2018, tôi muốn tính doanh số bán hàng của 31 ngày trong tháng cuối kỳ:

```
CALCULATE(  
    SUM('Sale Datamaker'[Doanh thu]),  
    DATESINPERIOD('Sale Datamaker'[Ngày ban],  
        DATE(2018,5,31),  
        -31,  
        DAY))
```

DATESMTD()

Mục đích: Trả về một bảng có chứa một cột của các ngày cho tháng đến nay.

Cú pháp: **DATESMTD(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi muốn tính lũy kế doanh thu trong tháng 5:

```
CALCULATE(
    SUM('Sale Datamaker'[Doanh thu]),
    DATESMTD(
        FILTER(
            MONTH('Sale Datamaker'[Ngày ban]) = 5)))
```

DATESMTD()

Mục đích: Trả về một bảng có chứa một cột của các ngày cho tháng đến nay.

Cú pháp: **DATESMTD(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi muốn tính lũy kế doanh thu trong tháng 5:

```
CALCULATE(
    SUM('Sale Datamaker'[Doanh thu]),
```

**DATESMTD(
 FILTER(
 MONTH('Sale Datamaker'[Ngày ban]) = 5)))**

DATESQTD()

Mục đích: trả về một bảng có chứa một cột của các ngày cho quý đến nay.

Cú pháp: **DATESMTD(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi muốn tính lũy kế doanh thu theo quý:

**CALCULATE(
 SUM('Sale Datamaker'[Doanh thu]),
 DATESMTD('Sale Datamaker'[Ngày ban]))**

DATESYTD()

Mục đích: trả về một bảng có chứa một cột của các ngày cho năm đến nay.

Cú pháp: **DATESYTD(<dates>, <year_end_date>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày
year_end_date	Chuỗi ký tự có ngày xác định year_end_date . Mặc định là ngày 31 tháng 12 (giá trị không bắt buộc khai báo)

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi muốn tính lũy kế doanh thu theo năm:

CALCULATE(

SUM('Sale Datamaker'[Doanh thu]),

DATESYTD('Sale Datamaker'[Ngày ban])

ENDOFMONTH(

Mục đích: trả về ngày theo định dạng ngày giờ của ngày cuối cùng của tháng được chỉ định.

Cú pháp: **ENDOFMONTH**(<dates>)

Tham số	Mô tả
date	Khai báo ngày cần cần tham chiếu

Giá trị trả về: giá trị ngày cuối tháng theo định dạng ngày.

Ví dụ: Tôi có ngày bán hàng như sau: 01/01/2017 -> 12/31/2017, vậy hàm sau sẽ chỉ trả về ngày cuối tháng trong từng tháng: 01/31/2017 -> 12/31/2017:

ENDOFMONTH('Sale Datamaker'[Ngày ban])

ENDOFQUARTER(

Mục đích: trả về ngày theo định dạng ngày giờ của ngày cuối cùng của quý được chỉ định.

Cú pháp: **ENDOFQUARTER**(<dates>)

Tham số	Mô tả
date	Khai báo ngày cần cần tham chiếu

Giá trị trả về: giá trị ngày cuối quý theo định dạng ngày.

Ví dụ: Tôi thực hiện hàm trả về ngày cuối của từng quý trong năm của ngày bán hàng:

ENDOFQUARTER('Sale Datamaker'[Ngày bán])

ENDOFYEAR()

Mục đích: trả về ngày theo định dạng ngày giờ của ngày cuối cùng của năm được chỉ định.

Cú pháp: **ENDOFYEAR(<dates>)**

Tham số	Mô tả
date	Khai báo ngày cần cần tham chiếu
year_end_date	Chuỗi ký tự có ngày xác định year_end_date . Mặc định là ngày 31 tháng 12 (giá trị không bắt buộc khai báo)

Ví dụ: Tôi thực hiện hàm trả về ngày cuối của từng năm của ngày bán hàng:

ENDOFYEAR('Sale Datamaker'[Ngày bán])

FIRSTDAY()

Mục đích: trả về ngày đầu tiên trong ngữ cảnh hiện tại cho cột ngày được chỉ định.

Cú pháp: **FIRSTDAY(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn và một dòng có giá trị ngày tháng.

Ví dụ: Tôi thực hiện hàm trả về ngày đầu tiên của cột ngày bán (giá trị theo từng dòng):

FIRSTDAY('Sale Datamaker'[Ngày bán])

FIRSTNONBLANK()

Mục đích: trả về giá trị đầu tiên trong cột được kiểm tra không có giá trị rỗng thông qua biểu thức đi cùng.

Cú pháp: **FIRSTNONBLANK(<column>, <expression>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị ngày
Expression	Biểu thức kiểm tra giá trị rỗng

Giá trị trả về: một bảng có chứa một cột và một hàng duy nhất với giá trị đầu tiên được trả về.

Ví dụ: Tôi muốn liệt kê những sản phẩm có doanh thu bán hàng tốt nhất, hàm sẽ được viết như sau (có kết hợp TOPN hỗ trợ thống kê):

```
FIRSTNONBLANK(
    TOPN(
        1,
        VALUES('Product DataMaker'[san pham]),
        [Doanh thu]),
    1)
```

LASTDAY()

Mục đích: trả về ngày cuối trong ngữ cảnh hiện tại cho cột ngày được chỉ định.

Cú pháp: **LASTDAY(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn và một dòng có giá trị ngày tháng.

Ví dụ: Tôi thực hiện hàm trả về ngày cuối của cột ngày bán (giá trị theo từng dòng):

LASTDAY('Sale Datamaker'[Ngày bán])

LASTNONBLANK()

Mục đích: trả về giá trị cuối trong cột được kiểm tra không có giá trị rỗng thông qua biểu thức đi cùng.

Cú pháp: **LASTNONBLANK(<column>, <expression>)**

Tham số	Mô tả
column	Khai báo cột chứa giá trị ngày
Expression	Biểu thức kiểm tra giá trị rỗng

Giá trị trả về: một bảng có chứa một cột và một hàng duy nhất với giá trị cuối được trả về.

Ví dụ: Tôi muốn lấy giá trị doanh thu bán hàng sau cùng có kết hợp kiểm tra giá trị rỗng của ngày phát sinh giao dịch bán hàng:

```
CALCULATE(  
    SUM('Sale DataMaker'[doanh thu]),  
    LASTNONBLANK('Dates dataMaker'[date],  
        CALCULATE(  
            SUM('Sale DataMaker'[doanh thu])))
```

NEXTDAY()

Mục đích: trả về một bảng có chứa một cột của tất cả các ngày kế tiếp, dựa trên ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **NEXTDAY(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về ngày kế tiếp của ngày trong cột Ngày bán:

NEXTDAY('Sale DataMaker'[Ngày bán])

NEXTMONTH()

Mục đích: trả về một bảng có chứa một cột của tất cả các tháng kế tiếp, dựa trên ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **NEXTMONTH(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về tháng kế tiếp của tháng trong cột Ngày bán:

NEXTMONTH('Sale DataMaker'[Ngày bán])

NEXTQUARTER()

Mục đích: trả về một bảng có chứa một cột của tất cả các quý kế tiếp, dựa trên ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **NEXTQUARTER(<dates>)**

Tham số	Mô tả
---------	-------

date	Khai báo cột chứa giá trị ngày
------	--------------------------------

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về quý kế tiếp của quý trong Ngày bán:

NEXTQUARTER('Sale DataMaker'[Ngày bán])

NEXTYEAR()

Mục đích: trả về một bảng có chứa một cột của tất cả các năm kế tiếp, dựa trên ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **NEXTYEAR(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về năm kế tiếp của năm trong cột Ngày bán:

NEXTYEAR 'Sale DataMaker'[Ngày bán]

PARALLELPERIOD()

Mục đích: trả về một bảng có chứa giá trị theo định dạng ngày chuẩn, giá trị sẽ tăng hoặc giảm dựa vào tham số **number_of_intervals** được đưa vào theo ngữ thực tế.

Cú pháp: **PARALLELPERIOD(<dates>, <number_of_intervals>, <interval>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

number_of_intervals	Khai báo số nguyên xác định số lượng khoảng thời gian để cộng vào hoặc trừ đi
interval	Khoảng thời gian muốn thay đổi ngày tháng, hỗ trợ theo: year, quarter, month.

Giá trị trả về: một bảng có chứa một cột các giá trị ngày tháng.

Ví dụ: Tôi thực hiện tính tổng doanh thu bán hàng của năm trước dựa trên năm trong ngữ cảnh hiện tại:

CALCULATE(

SUM('Sale Datamaker'[Doanh thu],

PARALLELPERIOD('Sale Datamaker'[Ngày bán], -1, YEAR)

PREVIOUSDAY()

Mục đích: trả về một bảng có chứa một cột của tất cả các ngày trước ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **PREVIOUSDAY(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về ngày trước của ngày trong cột Ngày bán:

PREVIOUSDAY('Sale DataMaker'[Ngày bán])

PREVIOUSMONTH()

Mục đích: trả về một bảng có chứa một cột của tháng trước ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **PREVIOUSMONTH(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về tháng trước của tháng trong cột Ngày bán:

PREVIOUSMONTH('Sale DataMaker'[Ngày bán])

PREVIOUSQUARTER()

Mục đích: trả về một bảng có chứa một cột của quý trước ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **PREVIOUSQUARTER(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về quý trước của quý trong cột Ngày bán:

PREVIOUSQUARTER('Sale DataMaker'[Ngày bán])

PREVIOUSYEAR()

Mục đích: trả về một bảng có chứa một cột của năm trước ngày được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **PREVIOUSYEAR(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về năm trước của năm trong cột Ngày bán:

PREVIOUSYEAR('Sale DataMaker'[Ngày bán])

SAMEPERIODLASTYEAR()

Mục đích: trả về một bảng có chứa cột giá trị cùng kỳ của năm trước của năm được chỉ định trong cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **SAMPERIODLASTYEAR(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: Tôi thực hiện tính tổng doanh thu bán hàng của năm trước dựa trên năm trong ngữ cảnh hiện tại (tính toán doanh số theo doanh thu cùng kỳ với năm hiện tại):

CALCULATE(

SUM('Sale Datamaker'[Doanh thu'],

SAMPERIODLASTYEAR('Sale Datamaker'[Ngày bán])

STARTOFMONTH()

Mục đích: trả về ngày đầu tiên của tháng dựa trên cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **STARTOFMONTH(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về ngày đầu tiên của tháng trong cột Ngày bán:

STARTOFMONTH('Sale DataMaker'[Ngày bán])

STARTOFQUARTER()

Mục đích: trả về ngày đầu tiên của quý dựa trên cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **STARTOFQUARTER(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về ngày đầu tiên của quý trong cột Ngày bán:

STARTOFQUARTER ('Sale DataMaker'[Ngày bán])

STARTOFYEAR()

Mục đích: trả về ngày đầu tiên của năm dựa trên cột ngày tháng trong ngữ cảnh hiện tại.

Cú pháp: **STARTOFYEAR(<dates>)**

Tham số	Mô tả
date	Khai báo cột chứa giá trị ngày

Giá trị trả về: một bảng có chứa một cột đơn có giá trị ngày tháng.

Ví dụ: hàm sau trả về ngày đầu tiên của năm trong cột Ngày bán:

STARTOFYEAR('Sale DataMaker'[Ngày bán])

dataMAKER.VN

4. Nhóm hàm Bộ lọc (Filter Functions)

Mô tả: Nhóm hàm bộ lọc trong DAX bao gồm một số hàm phức tạp và mạnh mẽ nhất, có sự khác biệt rất nhiều so với các hàm Excel mà chúng ta quen dùng. Việc tra cứu thông qua các mối quan hệ giữa các bảng dữ liệu, từ đây cho phép bạn tạo ra các ngữ cảnh tra cứu linh động.

CALCULATE()

Mục đích: là loại hàm trong DAX thường dùng để lọc giá trị theo 1 biểu thức tính toán nào đó.

Cú pháp: **CALCULATE(<expression>, <filter1>, <filter2>,...)**

Tham số	Mô tả
Expression	Khai báo biểu thức tính toán
filter1, filter2	Khai báo các điều kiện lọc dữ liệu cho biểu thức tính toán

Giá trị trả về: trả về kết quả là giá trị từ biểu thức tính toán

Ví dụ: Tôi muốn lấy tổng doanh thu trong năm 2018 của 1 sản phẩm nào đó từ các thông số Doanh thu, Năm, Sản phẩm trong bảng Sale Datamaker:

```
CALCULATE(
    SUM('Sale Datamaker'[Doanh thu]),
    'Sale Datamaker'[Năm]=2018,
    'Sale Datamaker'[Sản phẩm]="TV")
```

ALL()

Mục đích: Trả về toàn bộ các dòng trong bảng hay toàn bộ giá trị trong cột chỉ định, bỏ qua thiết lập của các bộ lọc khác. Nói cách khác hàm này dùng trong ngữ cảnh bạn muốn tính toán toàn bộ giá trị mà không cần quan tâm đến việc giá trị có áp dụng bộ lọc nào trước đó.

Cú pháp: **ALL({<table> [<column>[, <column>[, <column>[,...]]]})**

Tham số	Mô tả
table	Bảng muốn bỏ qua các thiết lập lọc trước đó
column	Cột muốn bỏ qua các thiết lập lọc trước đó

Giá trị trả về: bảng hay cột đã loại bỏ bộ lọc.

Ví dụ: Tôi muốn lấy tổng doanh thu của tất cả sản phẩm:

```
CALCULATE(
    SUM('Sale Datamaker'[Doanh thu]),
    ALL('Sale Datamaker'[Sản phẩm]))
```

ALLEXCEPT()

Mục đích: Trả về toàn bộ các dòng trong bảng hay toàn bộ giá trị trong cột, bỏ qua thiết lập của các bộ lọc khác ngoại trừ bộ lọc được áp dụng cho những cột được chỉ định.

Cú pháp: **ALLEXCEPT({<table> [<column>[, <column>[, <column>[,...]]]})**

Tham số	Mô tả
table	Bảng muốn bỏ qua các thiết lập lọc trước đó, bao gồm cả những cột muốn giữ lại bộ lọc

column	Cột muốn giữ lại các thiết lập lọc trước đó
--------	---

Giá trị trả về: bảng hay cột đã loại bỏ bộ lọc ngoại trừ những cột được khai báo trong hàm.

Ví dụ: Tôi muốn lấy tổng doanh thu của tất cả sản phẩm nhưng muốn giữ lại những thiết lập lọc đã áp dụng cho năm bán hàng:

CALCULATE(

SUM('Sale Datamaker'[Doanh thu]),

ALLEXCEPT('Sale Datamaker'[Năm]))

ALLNOBLANKROW()

Mục đích: Trả về toàn bộ các dòng trong bảng không có giá trị rỗng (Blank)

Cú pháp: **ALLNOBLANKROW({<table> [<column>[, <column>[, <column>[,...]]]})**

Tham số	Mô tả
table	Bảng muốn thực hiện kiểm tra rỗng
column	Cột muốn thực hiện kiểm tra rỗng

Giá trị trả về: bảng hay cột đã loại bỏ giá trị rỗng.

Ví dụ: Tôi muốn thực hiện đếm số dòng của bảng Bán hàng với điều kiện chỉ đếm những dòng có khai báo thông tin Ngày bán:

COUNTROWS(

ALLNOBLANKROW('Sale Datamaker'[Ngày bán]))

FILTER()

Mục đích: Trả về bảng đại diện cho tập hợp con của một bảng hoặc biểu thức khác.

Cú pháp: **FILTER(<table>, <filter>)**

Tham số	Mô tả
table	Bảng muốn thực hiện lọc
filter	Điều kiện lọc

Giá trị trả về: bảng chứa các dòng thỏa điều kiện lọc.

Ví dụ: Tôi muốn lấy tổng doanh thu của tất cả sản phẩm có giá bán > 3,000,000 VND:

SUMX(

```
FILTER('Sale Datamaker',  
      'Sale Datamaker'[Gia ban] > 3,000,000),  
      'Sale Datamaker'[Doanh thu])
```

DISTINCT()

Mục đích: trả về bảng có cột chứa các giá trị duy nhất từ cột được chỉ định. Nói cách khác, các giá trị trùng lặp sẽ bị xóa và chỉ các giá trị duy nhất được trả về.

Cú pháp: **DISTINCT(<column>)**

Tham số	Mô tả
column	Khai báo cột cần lấy giá trị duy nhất.

Giá trị trả về: cột chứa giá trị duy nhất.

Ví dụ: Tôi muốn biết có bao nhiêu khách hàng đã giao dịch:

COUNTROWS(

DISTINCT('Sale Datamaker'[Khach hang]))

EARLIER()

Mục đích: trả về giá trị hiện tại của cột được chỉ định.

Cú pháp: **EARLIER(<column>, <number>)**

Tham số	Mô tả
column	Khai báo cột cần lấy giá trị duy nhất.
number	Số lần đánh giá bên ngoài.

Giá trị trả về: Giá trị hiện tại của dòng từ cột, tại số lần đánh giá bên ngoài.

Ví dụ: Tôi muốn thực hiện công thức xếp hạng khách hàng theo lợi nhuận bán hàng đạt được:

COUNTROWS(

```
FILTER('Sale Datamaker'[Khach hang],  
EARLIER('Sale Datamaker'[Loi nhuan] < 'Sale Datamaker'[Loi nhuan]))  
+ 1
```

HASONEFILTER()

Mục đích: trả về TRUE khi số lượng giá trị được lọc trực tiếp trên **columnName** là một, ngược lại trả về FALSE.

Cú pháp: **HASONEFILTER(<columnName>)**

Tham số	Mô tả
columnName	Khai báo cột cần kiểm tra.

Giá trị trả về: trả về TRUE khi số lượng giá trị được lọc trực tiếp trên **columnName** là một, ngược lại trả về FALSE.

Ví dụ: Tôi muốn kiểm tra có giá trị nào được lọc trực tiếp, nếu có thì trả về giá trị đó ngược lại trả về BLANK().

IF(

```
HASONEFILTER('Product Datamaker'[Product ID],  
FILTERS('Product Datamaker'[Product ID],  
BLANK()))
```

HASONEVALUE()

Mục đích: trả về TRUE khi số lượng giá trị được lọc trên **columnName** là một, ngược lại trả về FALSE.

Hàm này cũng tương tự như HASONEFILTER() nhưng áp dụng cho các trường hợp lọc phức tạp (cross-filters)

Cú pháp: **HASONEVALUE(<columnName>)**

Tham số

Mô tả

columnName

Khai báo cột cần kiểm tra.

Giá trị trả về: trả về TRUE khi số lượng giá trị được lọc trên **columnName** là một, ngược lại trả về FALSE.

Ví dụ: Tôi muốn kiểm tra cột sau có giá trị không, nếu có thì trả về giá trị đó ngược lại trả về BLANK().

IF(

```
HASONEVALUE('Product Datamaker'[Product ID],  
VALUES('Product Datamaker'[Product ID],  
BLANK()))
```

VALUES ()

Mục đích: trả về bảng có cột chứa các giá trị duy nhất từ cột được chỉ định. Nói cách khác, các giá trị trùng lặp sẽ bị xóa và chỉ các giá trị duy nhất được trả về.

Cú pháp: **VALUES(<tableNameOrcolumnName>)**

Tham số	Mô tả
tableNameOrcolumnName	Khai báo bảng cột cần lấy giá trị duy nhất.

Giá trị trả về: cột chứa giá trị duy nhất.

Ví dụ: Tôi muốn biết có bao nhiêu đơn hàng đã thực hiện:

COUNTROWS(

VALUES('Sale Datamaker'[OrderNumber]))

RELATED()

Mục đích: trả về một giá trị liên quan từ một bảng khác.

Cú pháp: **RELATED(<column>)**

Tham số	Mô tả
column	Khai báo cột cần lấy giá trị.

Giá trị trả về: một giá trị duy nhất có liên quan đến hàng hiện tại.

Ví dụ: Tôi muốn lọc những giao dịch trừ nhóm sản phẩm liên quan đến Microsoft Azure:

FILTER('Sale Datamaker'[OrderID],

RELATED('Product Datamaker'[Nhóm sản phẩm]) <> "Microsoft Azure")

5. Nhóm hàm Thông tin (Information Functions)

Mô tả: nhóm hàm toán học sẽ thực hiện...

CONTAINS()

Mục đích: Trả về TRUE nếu tồn tại các giá trị cho tất cả các cột được đề cập hoặc được chứa trong các cột đó, ngược lại nếu không hàm trả về FALSE.

Cú pháp: **CONTAINS(<tables>,<columnName>,<value>[,<columnName>,<value>]...)**

Tham số	Mô tả
table	Bảng muốn thực hiện tìm thông tin
columnName	Cột cần tìm giá trị
value	Nhập giá trị cần tìm

Giá trị trả về: trả về TRUE nếu tồn tại các giá trị cho tất cả các cột được đề cập hoặc được chứa trong các cột đó, ngược lại nếu không hàm trả về FALSE.

Ví dụ: Tôi muốn tìm thông tin khách hàng 'ICT24h' đã mua sản phẩm "Office 365 Business":

```
CONTAINS('Sale Datamaker',  
         'Sale Datamaker'[Khach hang],"ICT24h",  
         'Sale Datamaker'[San pham], "Office 365 Business")
```

ISBLANK()

Mục đích: Trả về TRUE nếu tồn tại giá trị rỗng, ngược lại nếu không hàm trả về FALSE.

Cú pháp: **ISBLANK(<value>)**

Tham số	Mô tả
value	Nhập giá trị hoặc biểu thức muốn kiểm tra rỗng.

Giá trị trả về: Giá trị Boolean TRUE nếu giá trị trống; nếu không FALSE.

Ví dụ: Tôi có hàm tính tổng doanh thu năm 2017 (Doanhthu_2017), tôi muốn tính tỷ lệ doanh thu cùng kỳ năm nay so với năm trước nhưng muốn kiểm tra nếu doanh thu năm trước có giá trị rỗng thì trả về BLANK(), ngược lại sẽ thực hiện tính tỷ lệ doanh thu:

IF(

```
ISBLANK(Doanhthu_2017,  
BLANK(),  
(Doanhthu_2018 – Doanhthu_2017/Doanhthu_2017)
```

ISERROR()

Mục đích: Trả về TRUE nếu tồn tại giá trị bị lỗi, ngược lại nếu không hàm trả về FALSE.

Cú pháp: **ISERROR(<value>)**

Tham số

Mô tả

value

Nhập giá trị hoặc biểu thức muốn kiểm tra lỗi.

Giá trị trả về: Giá trị Boolean TRUE nếu giá trị lỗi; nếu không FALSE.

Ví dụ: Tương tự ví dụ trên, tôi muốn kiểm tra có lỗi không – thường là kiểm tra xem có chia cho giá trị 0 – nếu có giá trị lỗi thì trả về BLANK(), ngược lại sẽ thực hiện tính tỷ lệ doanh thu:

IF(ISERROR(

```
Doanhthu_2018/Doanhthu_2017  
BLANK(),  
Doanhthu_2018/Doanhthu_2017)
```

ISNUMBER()

Mục đích: Trả về TRUE nếu tồn tại giá trị là số, ngược lại nếu không hàm trả về FALSE.

Cú pháp: **ISNUMBER(<value>)**

Tham số

Mô tả

value

Nhập giá trị hoặc biểu thức muốn kiểm tra.

Giá trị trả về: Giá trị Boolean TRUE nếu giá trị là số; nếu không FALSE.

Ví dụ:

IF(ISNUMBER(1980), “Giá trị là số”, “Không phải số”)

→ Kết quả trả về: “Giá trị là số”

IF(ISNUMBER(“1988”), “Giá trị là số”, “Không phải số”)

→ Kết quả trả về: “Không phải số”

ISTEXT()

Mục đích: Trả về TRUE nếu tồn tại giá trị là ký tự, ngược lại nếu không hàm trả về FALSE.

Cú pháp: **ISTEXT(<value>)**

Tham số

Mô tả

value

Nhập giá trị hoặc biểu thức muốn kiểm tra.

Giá trị trả về: Giá trị Boolean TRUE nếu giá trị là ký tự; nếu không FALSE.

Ví dụ:

IF(ISTEXT(1980), “Giá trị là ký tự”, “Không phải ký tự”)

→ Kết quả trả về: “Không phải ký tự”

IF(ISTEXT(“1988”), “Giá trị là ký tự”, “Không phải ký tự”)

→ Kết quả trả về: “Giá trị là ký tự”

LOOKUPVALUE()

Mục đích: Trả về các dòng có giá trị thỏa điều kiện tìm kiếm.

Cú pháp: **LOOKUPVALUE(<result_columnName>,<search_columnName>,<search_value>[,<search_columnName>,<search_value>]...)**

Tham số	Mô tả
result_columnName	Cột sẽ trả về kết quả các dòng nếu thỏa điều kiện truy vấn
search_columnName	Khai báo cột có giá trị cần truy vấn
Search value	Nhập giá trị hoặc biểu thức cần truy vấn

Giá trị trả về: các dòng trong cột result_columnName nếu thỏa điều kiện có cột & có giá trị cần tìm.

Ví dụ: Tìm những sản phẩm thuộc nhóm sản phẩm có mã “Microsoft Office 365”

LOOKUPVALUE(

'Sale Datamaker'[San pham],

'Sale Datamaker'[Nhóm sản phẩm], “Microsoft Office 365”)

6. Nhóm hàm Logic (Logical Functions)

Mô tả: nhóm hàm logic hoạt động khi một biểu thức trả về thông tin về các giá trị hoặc các tập hợp trong biểu thức. Ví dụ, bạn có thể sử dụng hàm IF để kiểm tra kết quả của một biểu thức và tạo các kết quả có điều kiện.

AND()

Mục đích: Kiểm tra xem cả hai đối số đều đúng và hợp lệ thì trả về TRUE, nếu không trả về false.

Cú pháp: **AND(<logical1>, <logical2>)**

Tham số	Mô tả
logical1, logical2	Nhập giá trị logic muốn kiểm tra.

Giá trị trả về: Giá trị Boolean TRUE hay FALSE dựa trên sự kết hợp các giá trị mà bạn muốn kiểm tra.

Ví dụ:

IF(AND(10 > 9, -10 < -1), "Tất cả đều đúng", "Có giá trị sai")

Kết quả trả về: "Tất cả đều đúng"

OR()

Mục đích: Tương tự như AND() nhưng chỉ trả về giá trị FALSE khi cả 2 đối số có giá trị sai.

Cú pháp: **OR(<logical1>, <logical2>)**

Tham số	Mô tả
logical1, logical2	Nhập giá trị logic muốn kiểm tra.

Giá trị trả về: Giá trị Boolean TRUE hay FALSE dựa trên sự kết hợp các giá trị mà bạn muốn kiểm tra.

Ví dụ:

Các hàm sau cho ra kết quả: “Tất cả đều đúng”

IF(OR(10 > 9, -10 < -1), “Tất cả đều đúng”, “Có giá trị sai”)

IF(OR(10 > 9, -10 > -1), “Tất cả đều đúng”, “Có giá trị sai”)

IF(OR(10 < 9, -10 < -1), “Tất cả đều đúng”, “Có giá trị sai”)

Hàm sau cho kết quả: “Có giá trị sai”

IF(OR(10 < 9, -10 > -1), “Tất cả đều đúng”, “Có giá trị sai”)

IF()

Mục đích: Kiểm tra xem một điều kiện được cung cấp như là đối số đầu tiên được đáp ứng. Trả về một giá trị nếu điều kiện là TRUE và trả về một giá trị khác nếu điều kiện là FALSE.

Cú pháp: **IF(logical_test, <value_if_true>, value_if_false)**

Tham số	Mô tả
logical_test	Bất kỳ giá trị hoặc biểu thức nào có thể được kiểm tra là TRUE hoặc FALSE.
value_if_true	Giá trị được trả về nếu kết quả kiểm tra là TRUE. Nếu bỏ qua, TRUE được trả về.
value_if_false	Giá trị được trả về nếu kết quả kiểm tra là FALSE. Nếu bỏ qua, FALSE được trả về.

Giá trị trả về: Bất kỳ giá trị nào có thể được trả lại bằng một biểu thức.

Ví dụ: Tôi muốn thực hiện phân loại Khách hàng theo điều kiện Doanh thu sau:

- Doanh thu < 5,000,000 VND sẽ đưa vào nhóm Đạt
- Doanh thu < 3,000,000 VND sẽ đưa vào nhóm Không Đạt
- Ngoài ra, doanh thu vượt mức trên sẽ đưa vào nhóm Tốt.

IF(‘Sale Datamaker’[Doanh thu] < 3,000,000, “Không Đạt”,

IF((‘Sale Datamaker’[Doanh thu] < 5,000,000, “Đạt”, “Tốt”))

SWITCH()

Mục đích: Đánh giá một biểu thức dựa vào danh sách các giá trị và trả về một trong nhiều biểu thức kết quả có thể có.

Cú pháp: **SWITCH(<expression>, <value>, <result>[,<value>,<result>]...[,<else>])**

Tham số	Mô tả
expression	Bất kỳ biểu thức DAX nào trả về một giá trị vô hướng duy nhất trong đó biểu thức được đánh giá nhiều lần (cho mỗi hàng / ngữ cảnh).
value	Giá trị không đổi được so khớp với kết quả biểu thức.
result	Bất kỳ biểu thức vô hướng nào được đánh giá nếu kết quả của biểu thức khớp với giá trị tương ứng.
else	Bất kỳ biểu thức vô hướng nào được đánh giá nếu kết quả của biểu thức không khớp với bất kỳ đối số giá trị nào.

Giá trị trả về: Giá trị vô hướng đến từ một trong các biểu thức kết quả, nếu có giá trị khớp hoặc từ biểu thức khác (VD: thông báo) nếu không khớp với bất kỳ giá trị nào.

Ví dụ: Tôi muốn thực hiện thêm cột thể hiện tên của tháng tương ứng với tháng cụ thể (theo số):

```
SWITCH([Month],
    1, "January",
    2, "February",
    3, "March",
    4, "April",
    5, "May",
    6, "June",
    7, "July",
    8, "August",
    9, "September",
```

10, "October",
11, "November",
12, "December"
, "Giá trị không xác định")

NOT()

Mục đích: Chuyển đổi trạng thái giá trị Boolean từ TRUE sang FALSE và ngược lại.

Cú pháp: **NOT(<logical1>)**

Tham số	Mô tả
logical1	Nhập giá trị hoặc biểu thức muốn đổi trạng thái.

Giá trị trả về: Giá trị Boolean TRUE hay FALSE.

Ví dụ:

Tôi muốn chuyển trạng thái TRUE của hàm có giá trị đúng sau thành FALSE

```
NOT(  
    IF(10 > 9, "TRUE", "FALSE"))
```

IFERROR()

Mục đích: Thường dùng trong trường hợp bẫy lỗi giá trị hay biểu thức, thay giá trị lỗi bằng giá trị hay biểu thức mong muốn.

Cú pháp: **IFERROR(<value>, < value_if_error>)**

Tham số	Mô tả
value	Nhập giá trị hoặc biểu thức muốn kiểm tra

value_if_error

Nhập giá trị hoặc biểu thức trả về nếu có lỗi xảy ra

Giá trị trả về: Giá trị vô hướng có cùng kiểu với tham số value.

Ví dụ:

Kết quả sẽ là 999999 nếu biểu thức xảy ra lỗi:

IFERROR(10/0, 999999)

dataMAKER.VN

7. Nhóm hàm Text (Text Functions)

Mô tả: nhóm hàm text trả về một phần của chuỗi, tìm kiếm văn bản trong một chuỗi hoặc ghép các ký tự.

BLANK()

Mục đích: Trả về giá trị rỗng.

Cú pháp: **BLANK()**

Giá trị trả về: Giá trị rỗng.

Ví dụ: Kiểm tra nếu Doanh thu năm 2017 có giá nhỏ hơn hoặc bằng 0 thì trả về Blank, ngược lại tiến hành thực hiện phép chia.

```
IF(Doanhthu_2017 <= 0,
    BLANK(),
    Doanhthu_2018/Doanhthu_2017)
```

FIND()

Mục đích: Trả về vị trí bắt đầu của một chuỗi văn bản trong một chuỗi văn bản khác. FIND phân biệt chữ hoa chữ thường.

Cú pháp: **FIND(<find_text>,<within_text>[,<start_num>],[<NotFoundValue>])**

Tham số	Mô tả
find_text	Chuỗi văn bản cần tìm, có thể dùng các ký tự đặc biệt: (?), (*) theo quy định sau: <ul style="list-style-type: none"> - Dấu chấm hỏi khớp với bất kỳ ký tự đơn nào - Dấu sao thị khớp với bất kỳ chuỗi ký tự nào
within_text	Chuỗi văn bản có chứa chuỗi hay ký tự cần tìm.
start_num	Các ký tự mà tại đó để bắt đầu tìm kiếm; nếu bỏ qua, start_num = 1. Ký tự đầu tiên trong within_text là ký tự số 1.

NotFoundValue	Giá trị trả về trong trường hợp không tìm thấy chuỗi văn bản, ký tự khớp với nội dung cần tìm. Thông thường là: 0, -1, hoặc BLANK().
---------------	--

Ví dụ: tìm vị trí của chữ cái đầu tiên của tên Sản phẩm “Office 365”, trong chuỗi chứa mô tả sản phẩm.

FIND(“Office 365”, “Microsoft Office 365 Enterprise E5”)

CONCATENATE()

Mục đích: Nối 2 chuỗi lại với nhau.

Cú pháp: **CONCATENATE(<text1>, <text2>)**

Tham số	Mô tả
text1, text2	Khai báo các chuỗi ký tự hoặc số cần nối lại với nhau. Có thể áp dụng cho cột tham chiếu

Kết quả trả về: chuỗi đã nối.

Ví dụ: nối 2 chuỗi sau:

CONCATENATE(“dataMAKER”, “a member of ICT24h TEAM”)