

Phân tích dữ liệu với DAX



DAX là gì

DAX là viết tắt của cụm từ **Data Analysis Expressions** là tập hợp các hàm, toán tử và hằng số, được sử dụng để giải quyết những bài toán từ cơ bản đến phức tạp. Từ đó, trả về một hoặc nhiều giá trị từ dữ liệu có sẵn.

- Một cách dễ hiểu, DAX được coi là hàm trong Power BI
- Tuy nhiên, vượt xa khả năng của các hàm thuần túy, với các chức năng được xây dựng chuyên biệt để làm việc với các data model, DAX rất mạnh mẽ và linh hoạt

02 cách để sử dụng DAX

3 yếu tố của DAX

3 vấn đề quan trọng mà nhất định phải nắm. Đó là **Syntax**(cú pháp), **Function**(chức năng) và **Context**(ngữ cảnh).

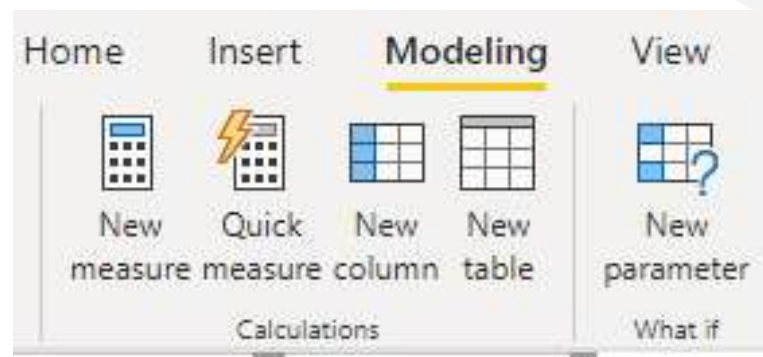
- **Syntax** có thể hiểu là cú pháp, trước khi tạo ra các công thức tính toán, bạn cần biết cú pháp của Power BI. Dưới đây là một ví dụ công thức DAX đơn giản cho một measure.
- **Function** được định nghĩa là các công thức được xác định trước, thực hiện những phép tính bằng các giá trị cụ thể được gọi là đối số, theo một thứ tự hoặc cấu trúc cụ thể.
- **Context** là ngữ cảnh sử dụng DAX. Có 2 loại Context trong DAX đó là Row Context và Filter Context.
 - **Row Context** áp dụng bất cứ khi nào một công thức có một hàm áp dụng bộ lọc để xác định một hàng trong bảng
 - **Filter Context** giống như là một hoặc nhiều bộ lọc được áp dụng trong một phép tính mà được xác định kết quả hoặc giá trị

So sánh DAX và hàm

DAX (Power BI)	Hàm (Excel)
Thường quy định kiểu dữ liệu đầu vào	Thường không quy định, nếu sai thì báo lỗi
Chỉ tham chiếu theo Cột hoặc Bảng	Tham chiếu theo Ô-vùng và cả Cột-Bảng
Kết quả trả về có thể là một Giá trị, một Bảng	Kết quả trả về là một Giá trị
DAX yêu cầu tạo ra các mối quan hệ giữa các bảng để sử dụng	Có thể làm việc với các bảng độc lập

THÊM COLUMNS & MEASURES

1. Cách 1: Chọn “**New Measure**” hoặc “**New Column**” trong tab **Modeling**



Khi bạn chèn một Columns hoặc Measures sử dụng tab **Modeling**, PowerBI sẽ mặc định bảng đang chọn làm dữ liệu tham chiếu, hoặc bảng đầu tiêu trong list thông thường

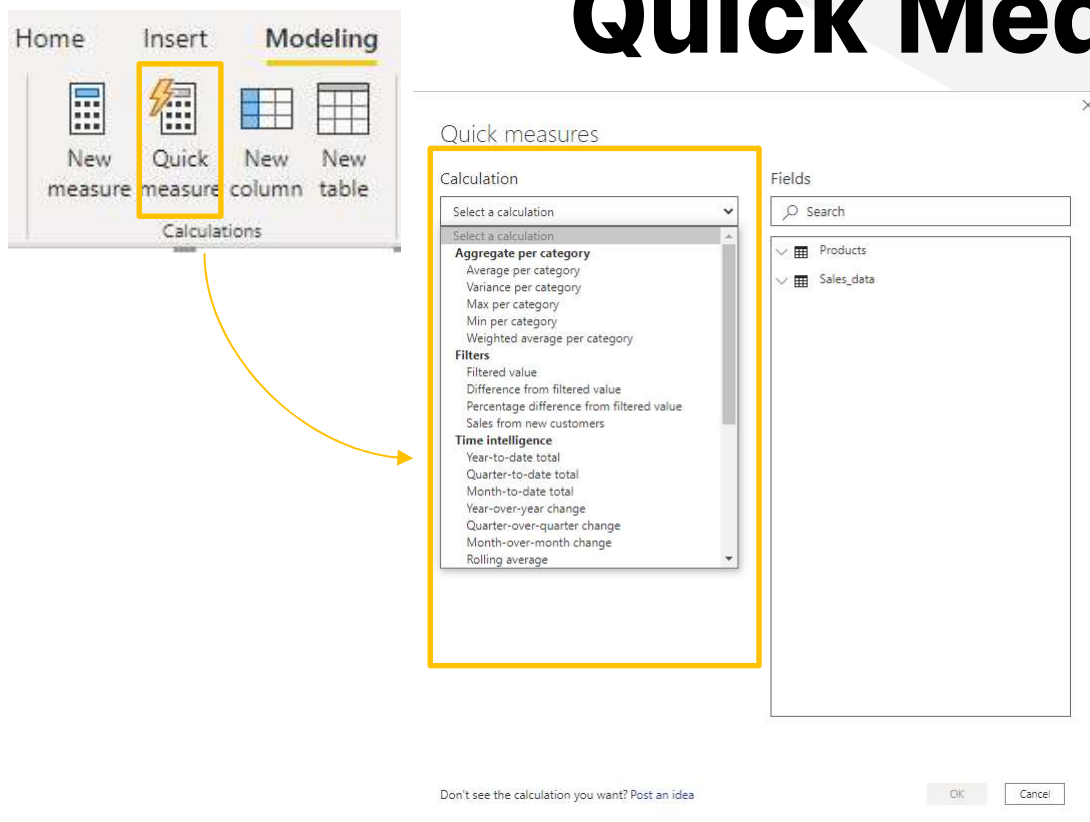
THÊM COLUMNS & MEASURES

Cách 2: Chuột phải trong
table (trong **Data view**)
hoặc trong **Field List** (cả
Data và **Report view**)

The screenshot displays the Power BI Desktop interface. The main area shows the 'Product' table in Data view. A right-click context menu is open over the 'ProductCost' column, with the 'New measure' option highlighted. The Field List pane on the right shows the 'Product' table with various fields listed. The 'New measure' option is also highlighted in the Field List pane.

ModelName	ProductDescription	ProductSize	ProductStyle	ProductCost	ProductPrice	Margin
-100	Universal fit, well-vented, lightweight,...	0	0	13.0863	34.99	21.9
-100	Universal fit, well-vented, lightweight,...	0	0	12.0278	33.6442	21.6
-100	Universal fit, well-vented, lightweight,...	0	0	12.0278	33.6442	21.6
ig Cap	Traditional style with a flip-up brim, or...	0	U	5.7052	8.6442	2.54
'k	Stout design absorbs shock and offers...	0	0	65.8097	148.22	82.4
rk	Composite road fork with an aluminum...	0	0	77.9176	175.49	97.5
rk	High-performance carbon road fork wi...	0	0	101.8936	229.49	127.5
adset	Threadless headset provides quality a...	0	0	15.1848	34.2	19.0
eadset	Sealed cartridge keeps dirt out.	0	0	45.4168	102.29	56.8
adset	High-quality 1" threadless headset wi...	0	0	55.3801	124.73	69.5
untain Handlebars	All-purpose bar for on or off-road.	0	0	17.978	40.4909	22.5
ountain Handlebars	Tough aluminum alloy bars for downhi...	0	0	24.9932	56.2909	31.2
untain Handlebars	Flat bar strong enough for the pro circ...	0	0	48.5453	109.3364	60.7
sd Handlebars	Unique shape provides easier reach to...	0	0	17.978	40.4909	22.5
oad Handlebars	Anatomically shaped aluminum tube b...	0	0	24.9932	56.2909	31.2
ad Handlebars	Designed for racers; high-end anatomi...	0	0	48.5453	109.3364	60.7
untain Front Wheel	Replacement mountain wheel for entr...	0	0	26.9708	60.745	33.7
ountain Front Wheel	Replacement mountain wheel for the c...	0	0	92.8071	209.025	116.2
untain Front Wheel	High-performance mountain replacem...	0	0	133.2955	300.215	166.9
sd Front Wheel	Replacement road front wheel for entry-level cyclist.	Black	0	37.9909	85.565	47.5
oad Front Wheel	Sturdy alloy features a quick-release hub.	Black	0	110.2829	248.385	138.1
ad Front Wheel	Strong wheel with double-walled rim.	Black	0	146.5466	330.06	183.5
1g Front Wheel	Aerodynamic rims for smooth riding.	Black	0	96.7964	218.01	121.2136
untain Rear Wheel	Replacement mountain wheel for entry-level rider.	Black	0	38.9588	87.745	48.7862
ountain Rear Wheel	Replacement mountain wheel for the casual to serious rid	Black	0	104.7951	236.025	131.2299
untain Rear Wheel	High-performance mountain replacement wheel.	Black	0	145.2835	327.215	181.9315
sd Rear Wheel	Replacement road rear wheel for entry-level cyclist.	Black	0	49.9789	112.565	62.5861
oad Rear Wheel	Aluminum alloy rim with stainless steel spokes; built for sp	Black	0	122.2709	275.385	153.1141
ad Rear Wheel	Strong rear wheel with double-walled rim.	Black	0	158.5346	357.06	198.5254
1g Rear Wheel	Excellent aerodynamic rims guarantee a smooth ride.	Black	0	108.7844	245.01	136.2256
1g-Panniers	Durable, water-proof nylon construction with easy access.	Grey	0	51.5625	125	73.4375
Lock	Wraps to fit front and rear tires, carrier and 2 keys include	NA	0	10.3125	25	14.6875
ump	Designed for convenience. Fits in your pocket. Aluminum i	NA	0	8.2459	19.99	11.7441

Quick Measure



Không nên lạm dụng Quick Measure

Quick Measures là các công thức dựng trước dạng templates, cho phép bạn kéo và thả vào các trường, hơn là phải viết DAX từ đầu

Mặc dù công cụ này khá hữu ích khi sử dụng, đặc biệt là với các hàm có tính chất phức tạp (*như weighted averages hay time intelligence...*), nhưng nó sẽ khiến bạn lười hơn và đôi khi không hiểu rõ về DAX

CALCULATED COLUMNS

Calculated columns cho phép bạn thêm mới cột vào data, dựa trên các biểu thức tính toán từ các cột đã có sẵn

- DAX tham chiếu tới các trường theo cột và bảng, không tham chiếu theo ô hay vùng
- Calculated columns tạo ra các giá trị trong **tables**
- Calculated columns rất tốt để tạo ra các cột giá trị mới dựa trên các dữ liệu tham chiếu từ hàng, nhưng lại không hữu dụng cho việc tính toán để lên trực tiếp báo cáo

Ví dụ CALCULATED COLUMNS

Trong bảng bên, chúng ta thêm một *calculated column* tên là “**Parent**”, kết quả trả về “**Yes**” nếu trường [TotalChildren] lớn hơn 0, và “**No**” trong trường hợp còn lại (giống *IF* trong *Excel*!)

- Khi này, DAX đọc các giá trị theo từng hàng, giá trị mới được tính toán dựa trên điều kiện logic từ trường [TotalChildren]
- Đây là một điển hình của *calculated columns*; nó tạo ra các giá trị mà chúng ta có thể lọc từ các giá trị đó trong bảng

1 Parent = IF(Customers[TotalChildren]>0,"Yes","No")													
CustomerKey	Prefix	FirstName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	AnnualIncome	TotalChildren	EducationLevel	Occupation	HomeOwner	Parent
24885	MR.	COREY	XIE	Sunday, 2 November 1975	S	M	corey3@adventure-works.com	\$60,000	0	Bachelors	Professional	Y	No
24893	MR.	ALVIN	CHANDE	Tuesday, 26 March 1974	S	M	alvin36@adventure-works.com	\$60,000	0	Bachelors	Professional	N	No
25809	MR.	NICHOLAS	WILLIAMS	Wednesday, 26 November 1975	M	M	nicholas4@adventure-works.com	\$60,000	0	Bachelors	Professional	Y	No
25810	MR.	JIMMY	SANZ	Friday, 14 March 1975	S	M	jimmy24@adventure-works.com	\$60,000	0	Bachelors	Professional	Y	No
26431	MR.	SHAUN	RAI	Friday, 16 August 1974	S	M	shaun18@adventure-works.com	\$60,000	0	Bachelors	Professional	Y	No
26432	MR.	SHAUN	RAI	Friday, 16 August 1974	S	M	shaun18@adventure-works.com	\$100,000	0	Bachelors	Professional	N	No
26433	MR.	SHAUN	RAI	Friday, 16 August 1974	S	M	shaun18@adventure-works.com	\$60,000	0	Bachelors	Professional	Y	No
26434	MR.	SHAUN	RAI	Friday, 16 August 1974	S	M	shaun18@adventure-works.com	\$90,000	2	Bachelors	Professional	Y	Yes
26435	MR.	SHAUN	RAI	Friday, 16 August 1974	S	M	shaun18@adventure-works.com	\$80,000	5	Bachelors	Professional	Y	Yes
11768	MR.	CLAYTON	NARA	Saturday, 8 October 1966	M	M	clayton34@adventure-works.com	\$90,000	2	Bachelors	Professional	Y	Yes
11900	MR.	BYRON	CARLSON	Saturday, 7 March 1970	S	M	byron12@adventure-works.com	\$90,000	2	Bachelors	Professional	Y	Yes
13120	MR.	LEVI	WEBER	Friday, 9 July 1965	M	M	levi3@adventure-works.com	\$80,000	5	Bachelors	Professional	Y	Yes
13130	MR.	DALTON	MILLER	Monday, 26 October 1959	M	M	dalton5@adventure-works.com	\$90,000	1	Bachelors	Professional	Y	Yes
13132	MR.	CASEY	VAZQUEZ	Saturday, 21 February 1959	S	M	casey38@adventure-works.com	\$90,000	1	Bachelors	Professional	Y	Yes
13133	MR.	DEVON	KENNEDY	Wednesday, 19 August 1959	S	M	devon5@adventure-works.com	\$100,000	5	Bachelors	Professional	Y	Yes
13955	MR.	ANDRES	RAJI	Sunday, 8 February 1970	M	M	andres17@adventure-works.com	\$90,000	2	Bachelors	Professional	Y	Yes
14057	MR.	LANCE	ALONSO	Tuesday, 24 November 1959	M	M	lance8@adventure-works.com	\$90,000	1	Bachelors	Professional	Y	Yes
14059	MR.	RODNEY	RAMOS	Wednesday, 22 December 1961	M	M	rodney13@adventure-works.com	\$80,000	5	Bachelors	Professional	Y	Yes
14060	MR.	CEDRIC	SHARMA	Saturday, 27 November 1965	M	M	cedric30@adventure-works.com	\$80,000	5	Bachelors	Professional	Y	Yes
14070	MR.	RICKY	ALONSO	Monday, 10 November 1958	S	M	ricky8@adventure-works.com	\$90,000	2	Bachelors	Professional	Y	Yes
14299	MR.	JERMAINE	CHANDRA	Sunday, 11 June 1967	M	M	jermaine1@adventure-works.com	\$90,000	2	Bachelors	Professional	N	Yes

Parent = IF(Customers[TotalChildren]>0,"Yes","No")

Ví dụ CALCULATED COLUMNS

TotalQua = SUM(Sales_data[OrderQuantity])

Còn trong bảng ví dụ này, chúng

ta sử dụng hàm (SUM) để tính

giá trị cột **TotalQuantity**

- Khi này, DAX không đọc các giá trị theo từng hàng, kết quả trả về là giống nhau ở tất cả các hàng và là tổng giá trị tính toán

- Đây không phải một điển hình của *calculated columns*; các giá trị cố định trong bảng, không thể lọc hay sử dụng làm sliced

1 TotalQua = SUM(Sales_data[OrderQuantity])										
Date	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Column	Revenue	TotalQua
1 January 2017	Friday, 10 January 2003	SO61305	583	20236	9	1	1	bad	1700.99	45314
1 January 2017	Friday, 31 October 2003	SO61276	488	17252	9	1	1	bad	53.99	45314
1 January 2017	Tuesday, 10 June 2003	SO61307	582	20260	9	1	1	bad	1700.99	45314
1 January 2017	Wednesday, 26 November 2003	SO61309	354	13625	9	1	1	bad	2071.4196	45314
1 January 2017	Tuesday, 9 December 2003	SO61306	580	23431	9	1	1	bad	1700.99	45314
1 January 2017	Friday, 19 September 2003	SO61275	220	13996	9	1	1	bad	33.6442	45314
1 January 2017	Saturday, 25 October 2003	SO61308	371	20246	9	1	1	bad	2181.5625	45314
1 January 2017	Saturday, 13 September 2003	SO61312	566	29082	9	1	1	bad	742.35	45314
February 2017	Tuesday, 16 September 2003	SO61367	575	11099	9	1	1	bad	2384.07	45314
February 2017	Friday, 19 September 2003	SO61315	539	15146	9	1	1	bad	24.99	45314
February 2017	Thursday, 13 November 2003	SO61370	561	11464	9	1	1	bad	2384.07	45314
February 2017	Sunday, 28 September 2003	SO61362	358	13648	9	1	1	bad	2049.0982	45314
February 2017	Wednesday, 22 October 2003	SO61369	565	29084	9	1	1	bad	742.35	45314
February 2017	Friday, 12 December 2003	SO61368	576	11904	9	1	1	bad	2384.07	45314
February 2017	Monday, 15 September 2003	SO61371	606	27515	9	1	1	bad	539.99	45314
February 2017	Wednesday, 11 June 2003	SO61321	235	14319	9	1	1	bad	48.0673	45314
1 March 2017	Thursday, 10 April 2003	SO61420	362	13122	9	1	1	bad	2049.0982	45314
1 March 2017	Monday, 12 May 2003	SO61419	387	24484	9	1	1	bad	1000.4375	45314
1 March 2017	Friday, 31 October 2003	SO61425	579	13050	9	1	1	bad	1214.85	45314
1 March 2017	Saturday, 22 November 2003	SO61418	385	24585	9	1	1	bad	1000.4375	45314
1 March 2017	Friday, 14 November 2003	SO61427	606	27512	9	1	1	bad	539.99	45314
1 March 2017	Friday, 12 September 2003	SO61374	215	13046	9	1	1	bad	33.6442	45314
1 March 2017	Friday, 24 October 2003	SO61375	215	13150	9	1	1	bad	33.6442	45314
1 March 2017	Wednesday, 12 February 2003	SO61417	387	17734	9	1	1	bad	1000.4375	45314

Measures

Measures là công thức DAX sử dụng để tính toán giá trị

- Giống như calculated columns, measures cũng tham chiếu tới **tables** và **columns**
- Tuy nhiên, **measure** không hiện hữu giá trị trong bảng; nó chỉ có thể “*nhìn thấy*” khi visualize như khi sử dụng chart hay matrix (*giống như một trường tính toán trong Excel pivot*)
- Measures tính toán dựa trên các dữ liệu lọc từ bảng, nghĩa là nó sẽ cập nhật lại khi các trường hoặc dữ liệu lọc từ bảng thay đổi

NOTE: Thông thường, measures dùng khi cần tổng hợp dữ liệu

IMPLICIT vs EXPLICIT MEASURES

Measures ẩn là các measures được tạo ra khi bạn kéo các trường dữ liệu số (như “*OrderQuantity*”) trong trình tạo visual và chọn một dạng tính toán nào đó (*Sum, Average, Min/Max, ...*)

Measures hiện là các measures được tạo ra bằng việc nhập các hàm DAX (hoặc thêm một “*quick measures*”) để xác định giá trị

So sánh CALCULATED COLUMNS vs MEASURES

CALCULATED COLUMNS	MEASURES
Giá trị được tính toán dựa trên tham chiếu tới vị trí theo hàng trong cột	Giá trị được tính toán dựa vào các kết quả lọc từ các dữ liệu trong
Nội các giá trị tính trong bảng và lưu trữ trong model (<i>làm tăng kích thước file</i>)	Không tạo ra các data mới trong model (<i>không làm tăng kích thước file</i>)
Cập nhật khi data source được refresh hoặc thay đổi được tạo ra trong bảng	Cập nhật khi các kết quả lọc từ các dữ liệu thay đổi trong báo cáo
Thường sử dụng để tạo thêm các columns , slicers or filters	Thường sử dụng trong các trường values lên báo cáo hoặc biểu đồ

X ✓ 1 Revenue = Sales_data[OrderQuantity]*RELATED(Products[ProductPrice])

OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Column	Revenue
Sunday, 1 January 2017	Friday, 10 January 2003	SO61305	583	20236	9	1	1	bad	1700.99
Sunday, 1 January 2017	Friday, 31 October 2003	SO61276	488	17252	9	1	1	bad	53.99
Sunday, 1 January 2017	Tuesday, 10 June 2003	SO61307	582	20260	9	1	1	bad	1700.99
Sunday, 1 January 2017	Wednesday, 26 November 2003	SO61309	354	13625	9	1	1	bad	2071.4196
Sunday, 1 January 2017	Tuesday, 9 December 2003	SO61306	580	23431	9	1	1	bad	1700.99
Sunday, 1 January 2017	Friday, 19 September 2003	SO61275	220	13996	9	1	1	bad	33.6442
Sunday, 1 January 2017	Saturday, 25 October 2003	SO61308	371	20246	9	1	1	bad	2181.5625
Sunday, 1 January 2017	Saturday, 13 September 2003	SO61312	566	29082	9	1	1	bad	742.35
ednesday, 1 February 2017	Tuesday, 16 September 2003	SO61367	575	11099	9	1	1	bad	2384.07
ednesday, 1 February 2017	Friday, 19 September 2003	SO61315	539	15146	9	1	1	bad	24.99
ednesday, 1 February 2017	Thursday, 13 November 2003	SO61370	561	11464	9	1	1	bad	2384.07
ednesday, 1 February 2017	Sunday, 28 September 2003	SO61362	358	13648	9	1	1	bad	2049.0982
ednesday, 1 February 2017	Wednesday, 22 October 2003	SO61369	565	29084	9	1	1	bad	742.35
ednesday, 1 February 2017	Friday, 12 December 2003	SO61368	576	11904	9	1	1	bad	2384.07
ednesday, 1 February 2017	Monday, 15 September 2003	SO61371	606	27515	9	1	1	bad	539.99
ednesday, 1 February 2017	Wednesday, 11 June 2003	SO61321	235	14319	9	1	1	bad	48.0673
Wednesday, 1 March 2017	Thursday, 10 April 2003	SO61420	362	13122	9	1	1	bad	2049.0982
Wednesday, 1 March 2017	Monday, 12 May 2003	SO61419	387	24484	9	1	1	bad	1000.4375
Wednesday, 1 March 2017	Friday, 31 October 2003	SO61425	579	13050	9	1	1	bad	1214.85
Wednesday, 1 March 2017	Saturday, 22 November 2003	SO61418	385	24585	9	1	1	bad	1000.4375
Wednesday, 1 March 2017	Friday, 14 November 2003	SO61427	606	27512	9	1	1	bad	539.99
Wednesday, 1 March 2017	Friday, 12 September 2003	SO61374	215	13046	9	1	1	bad	33.6442
Wednesday, 1 March 2017	Friday, 24 October 2003	SO61375	215	13150	9	1	1	bad	33.6442
Wednesday, 1 March 2017	Wednesday, 12 February 2003	SO61417	387	17734	9	1	1	bad	1000.4375
Wednesday, 1 March 2017	Sunday, 19 October 2003	SO61426	606	27500	9	1	1	bad	539.99
Saturday, 1 April 2017	Saturday, 12 July 2003	SO61435	488	17250	9	1	1	bad	53.99
Saturday, 1 April 2017	Monday, 10 November 2003	SO61471	389	17730	9	1	1	bad	1000.4375
Saturday, 1 April 2017	Monday, 10 March 2003	SO61473	358	13135	9	1	1	bad	2049.0982
Saturday, 1 April 2017	Friday, 11 July 2003	SO61472	358	13123	9	1	1	bad	2049.0982
Saturday, 1 April 2017	Friday, 12 September 2003	SO61429	371	20616	9	1	1	bad	2181.5625
Saturday, 1 April 2017	Tuesday, 23 September 2003	SO61428	379	20615	9	1	1	bad	2181.5625
Saturday, 1 April 2017	Thursday, 12 June 2003	SO61481	587	17233	9	1	1	bad	769.49
Monday, 1 May 2017	Saturday, 13 September 2003	SO61522	583	23423	9	1	1	bad	1700.99

HIỂU VỀ FILTER CONTEXT

Chúng ta luôn nhớ rằng, measures được tính toán dựa trên **filter context**, có nghĩa là nó sẽ tính toán lại khi các trường hoặc filter thay đổi

ProductName	Total Orders	Return Rate
Water Bottle - 30 oz.	1,164	1.96 %
Road Tire Tube	829	1.63 %
AWC Logo Cap	803	0.93 %
Patch Kit/8 Patches	798	1.57 %
Sport-100 Helmet, Red	755	2.79 %
Touring Tire Tube	702	1.35 %
Sport-100 Helmet, Blue	666	3.15 %
Sport-100 Helmet, Black	626	3.67 %
Road Bottle Cage	560	1.58 %
Mountain Tire Tube	554	1.95 %
Mountain Bottle Cage	539	1.38 %
Touring Tire	427	1.16 %
LL Road Tire	421	2.02 %
Fender Set - Mountain	378	1.82 %
ML Road Tire	297	1.72 %
ML Mountain Tire	266	1.94 %
HL Mountain Tire	206	3.40 %
Mountain-200 Silver, 46	199	1.51 %
Mountain-200 Black, 46	196	3.06 %
LL Mountain Tire	195	2.09 %
Mountain-200 Silver, 38	189	2.65 %
Bike Wash - Dissolver	187	2.38 %
Mountain-200 Black, 42	182	3.85 %
Mountain-200 Black, 38	180	3.33 %
Long-Sleeve Logo Jersey, M	161	4.35 %
HL Road Tire	158	5.06 %
Mountain-200 Silver, 42	156	1.28 %
Hydration Pack - 70 oz.	147	4.08 %
Long-Sleeve Logo Jersey, L	147	2.72 %
Long-Sleeve Logo Jersey, S	130	2.31 %
Total	7,380	2.17 %

Một ví dụ cụ thể trong matrix, **Total Orders** measure được tính dựa trên **filter context** sau: $Products[ProductName] = \text{“Touring Tire Tube”}$

- Nó cho phép measure trả về kết quả **Total order** cho mỗi sản phẩm cụ thể theo tên sản phẩm(hoặc bất kỳ hàng hoặc cột có nhãn riêng biệt – *product categories, customer names, ...*)

Giá trị này **không** tính toán bởi nguyên tắc như ở trên; nó vẫn là một measure, không dùng **filter context** (nghĩa là chúng ta không thể tính toán cho một sản phẩm cụ thể)

MEASURE: **Total Revenue**

FILTER CONTEXT:

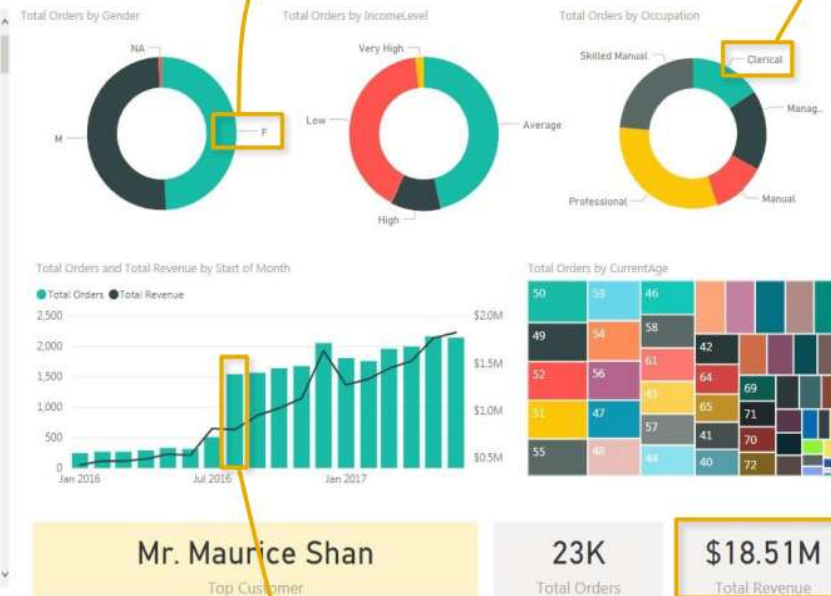
- **Calendar[Year] = 2016 or 2017**
- **Customers[Full Name] = Mr. Larry Munoz**

Full Name	Total Orders	Total Revenue
Mr. Maurice Shan	6	\$12,407.91
Mrs. Janet Munoz	6	\$12,015.39
Mrs. Lisa Chen	7	\$11,330.44
Mrs. Lacey Zheng	7	\$11,065.74
Mr. Jordan Turner	7	\$11,021.18
Mr. Larry Munoz	7	\$10,817.50
Mrs. Ariana Gray	6	\$9,191.40
Mr. Marco Lopez	6	\$10,289.60
Mr. Franklin Xu	5	\$10,164.34
Mrs. Margaret He	4	\$9,166.74
Mrs. Kathryn Henderson	4	\$9,156.82
Mrs. Nichole Nara	4	\$9,134.66
Mr. Randall Dominguez	4	\$9,110.28
Mrs. Ross Wu	4	\$8,201.2
Adriana Gonzalez	4	\$8,195.89
Mrs. Dominique Prasad	6	\$9,160.95
Mrs. Brandi Gill	4	\$9,146.38
Mr. Brad Shé	4	\$9,161.01
Mr. Francisco Sara	4	\$9,123.94
Mr. Kevin Coleman	4	\$7,750.55
Mr. Johnathan Suri	4	\$7,721.33
Mrs. Crystal Zeng	4	\$7,706.81
Mrs. Felicia Bianco	4	\$7,669.86
Mrs. Jill Suarez	4	\$7,632.61
Mr. Preston Raman	4	\$7,596.48
Mr. Willie Xu	4	\$7,553.55
Mrs. Abby Subram	4	\$7,306.39
Mr. Lance Bianco	4	\$7,207.07
Mrs. Audrey Bianco	4	\$7,139.17
Mr. Ricky Navarro	3	\$7,125.63
Mr. Eddie Dominguez	3	\$7,044.38
Mrs. Molly Madan	3	\$7,043.25
Mr. Jamod Mehta	3	\$7,036.37
Mrs. Susan Zhou	3	\$7,027.96
Mr. Brent Zhang	3	\$7,016.89
Mrs. Alyssa Bradley	3	\$7,018.84
Total	22,534	\$18,509,633.2

MEASURE: **Total Orders**

FILTER CONTEXT:

- **Calendar[Year] = 2016 or 2017**
- **Customers[Gender] = F (Female)**



MEASURE: **Total Orders**

FILTER CONTEXT:

- **Calendar[Year] = 2016 or 2017**
- **Customers[Occupation] = Clerical**

Values

Drag data fields here

FILTERS

Page level filters

Drag data fields here

Drillthrough filters

Drag drillthrough fields here

Report level filters

Year is 2016 or 2017

Đây là một **Filter** cấp trang (Page), nó tác động tới tất cả biểu đồ trên báo cáo

MEASURE: **Total Orders**

FILTER CONTEXT:

- **Calendar[Year] = 2016 or 2017**

MEASURE: **Total Orders**

FILTER CONTEXT:

- **Calendar[Year] = 2016 or 2017**
- **Calendar[Month] = August 2016**

MEASURE: **Total Revenue**

FILTER CONTEXT:

- **Calendar[Year] = 2016 or 2017**

HIỂU CÁC BƯỚC KHI CÓ MEASURE CALCULATION

CategoryName	Total Returns
Accessories	1,115
Bikes	342
Clothing	267

Làm như thế nào để measure tính đúng?

- Nhớ rằng: Tất cả xảy ra ngay lập tức phía sau, mỗi khi filter context thay đổi

STEP 1

Filter context được phát hiện và áp dụng

CategoryName	Total Returns
Accessories	1,115
Bikes	342
Clothing	267

Product[CategoryName] = "Accessories"

Product Table	
Accessories	

STEP 2

Quá trình lọc xảy ra ở tất cả các bảng liên quan

Product Table	
Accessories	

AW_Sales_Data	
Accessories	

AW_Returns_Data	
Accessories	

STEP 3

Measure tính toán lại dựa trên các bảng đã lọc

Total Returns = COUNTROWS(AW_Returns_Data)

Đếm các hàng trong bảng Returns_Data, lọc chỉ các hàng có product category is "Accessories" = 1,115

Các hàm DAX phổ biến



Cấu trúc của DAX

TÊN MEASURE/CAL COLUMN

• **Note:** Measures luôn ở trong dấu ngoặc (vd: **[Total Quantity]**) khi tham chiếu trong công thức, do vậy, có thể sử dụng dấu cách (space) trong tên measures

Tên **TABLE**
tham chiếu

Tên **COLUMN**
tham chiếu

Total Quantity: =SUM(Transactions[quantity])

HÀM (Function)

Không phải lúc nào calculated column cũng sử dụng các hàm, nhưng measure thì luôn luôn:

- Trong một **Calculated Column**, hàm =**Transactions[quantity]** sẽ trả về giá trị là số lượng giao dịch cho từng hàng (ô) trong cột
- Trong một **Measure**, hàm =**Transactions[quantity]** sẽ trả về một giá trị **error** do Power BI không thể dịch hiểu như là một giá trị riêng lẻ

Một số biểu thức trong DAX

Loại số học	Ý nghĩa	Ví dụ
+	Cộng	$2 + 7$
-	Trừ	$5 - 3$
*	Nhân	$2 * 6$
/	Chia	$4 / 2$
^	Mũ	$2 ^ 5$

Một số biểu thức trong DAX

Loại so sánh	Ý nghĩa	Ví dụ
=	Bằng	[City]="Boston"
>	Lớn hơn	[Quantity]>10
<	Nhỏ hơn	[Quantity]<10
>=	Lớn hơn hoặc bằng	[Unit_Price]>=2.5
<=	Nhỏ hơn hoặc bằng	[Unit_Price]<=2.5
<>	Khác	[Country]<>"Mexico"

Một số biểu thức trong DAX

Loại Text/Logic	Ý nghĩa	Ví dụ
&	Nối 02 giá trị dạng text thành 01 giá trị	[City] & “ ” & [State]
&&	Tạo một điều kiện VÀ giữa 02 biểu thức logic	([State]=“MA”) && ([Quantity]>10)
(double pipe)	Tạo một điều kiện HOẶC giữa 02 biểu thức logic	([State]=“MA”) ([State]=“CT”)
IN	Tạo một điều kiện HOẶC logic dựa trên danh sách đã cho (sử dụng dấu ngoặc)	‘Store Lookup’[State] IN { “MA”, “CT”, “NY” }

Các nhóm hàm DAX phổ biến

MATH & STATS Functions	LOGICAL Functions	TEXT Functions	FILTER Functions	DATE & TIME Functions
Hàm tính toán cơ bản cũng như tính toán trên vòng lặp sử dụng ngữ cảnh hàng	Hàm trả về thông tin về các giá trị trong biểu thức dựa trên điều kiện đã cho	Các hàm để thao tác các chuỗi văn bản hoặc các định dạng điều khiển cho ngày, giờ hoặc số	Các hàm tra cứu dựa trên các bảng có liên quan và các hàm lọc để tính toán	Các chức năng ngày và giờ cơ bản cũng như các hàm thời gian nâng cao
Ví dụ: • SUM • AVERAGE • MAX/MIN • DIVIDE • COUNT/COUNTA • COUNTROWS • DISTINCTCOUNT • SUMX • AVERAGEX • MAXX/MINX • RANKX • COUNTX	Ví dụ: • IF • IFERROR • AND • OR • NOT • SWITCH • TRUE • FALSE	Ví dụ: • CONCATENATE • FORMAT • LEFT/MID/RIGHT • UPPER/LOWER • PROPER • LEN • SEARCH/FIND • REPLACE • REPT • SUBSTITUTE • TRIM • UNICHAR	Ví dụ: • CALCULATE • FILTER • ALL • ALLEXCEPT • RELATED • RELATEDTABLE • DISTINCT • VALUES • EARLIER/EARLIEST • HASONEVALUE • HASONEFILTER • ISFILTERED • USERRELATIONSHIP	Ví dụ: • DATEDIFF • YEARFRAC • YEAR/MONTH/DAY • HOUR/MINUTE/SECOND • TODAY/NOW • WEEKDAY/WEEKNUM Time Intelligence: • DATESYTD • DATESQTD • DATESMTD • DATEADD • DATESINPERIOD

Hàm DATE & TIME cơ bản

DAY/MONTH/ YEAR()	<i>Trả về ngày trong tháng (1-31), tháng trong năm (1-12) hoặc năm của một ngày</i>	= DAY/MONTH/YEAR (Date)
HOUR/MINUTE/ SECOND()	<i>Trả về giờ (0-23), phút (0-59) hoặc giây (0-59) của một giá trị thời gian đã cho</i>	= HOUR/MINUTE/SECOND (Datetime)
TODAY/NOW()	<i>Trả về ngày hiện tại hoặc thời gian chính xác</i>	= TODAY/NOW ()
WEEKDAY/ WEEKNUM()	<i>Trả về số ngày trong tuần từ 1 (Chủ nhật) đến 7 (Thứ bảy) hoặc tuần # của năm</i>	= WEEKDAY/WEEKNUM (Date, [ReturnType])
EOMONTH()	<i>Trả về ngày của ngày cuối cùng của tháng, +/- một số tháng được chỉ định</i>	= EOMONTH (StartDate, Months)
DATEIFF()	<i>Trả về chênh lệch giữa hai ngày, dựa trên khoảng thời gian đã chọn</i>	= DATEDIFF (Date1, Date2, Interval)

HÀM LOGICAL CƠ BẢN (IF/AND/OR)

IF()

Kiểm tra nếu một điều kiện nhất định được đáp ứng và trả về một giá trị nếu điều kiện là TRUE và một điều kiện khác nếu điều kiện là FALSE

=**IF**(LogicalTest, ResultIfTrue, [ResultIfFalse])

IFERROR()

Đánh giá một biểu thức và trả về một giá trị được chỉ định nếu biểu thức trả về lỗi, nếu không thì trả về chính biểu thức

=**IFERROR**(Value, ValueIfError)

AND()

Kiểm tra xem cả hai đối số có TRUE không và trả về TRUE nếu cả hai đối số đều TRUE, nếu không thì trả về FALSE

=**AND**(Logical1, Logical2)

OR()

Kiểm tra xem một trong các đối số có TRUE để trả về TRUE không và trả về FALSE nếu cả hai đối số là FALSE

=**OR**(Logical1, Logical2)

Lưu ý: Sử dụng && và || toán tử nếu bạn muốn bao gồm nhiều hơn hai điều kiện

Hàm TEXT cơ bản

LEN()

Trả về số lượng ký tự trong một chuỗi

=**LEN**(Text)

CONCATENATE()

Nối hai chuỗi văn bản thành một

=**CONCATENATE**(Text1, Text2)

LEFT/MID/
RIGHT()

Trả về một số ký tự từ đầu / giữa / cuối của chuỗi văn bản

=**LEFT/RIGHT**(Text, [NumChars])

=**MID**(Text, StartPosition, NumChars)

UPPER/LOWER/
PROPER()

Chuyển đổi các chữ cái trong một chuỗi thành trường hợp trên / dưới / thích hợp

=**UPPER/LOWER/PROPER**(Text)

SUBSTITUTE()

Thay thế một thể hiện của văn bản hiện tại bằng văn bản mới trong một chuỗi

=**SUBSTITUTE**(Text, OldText, NewText, [InstanceNumber])

SEARCH()

Trả về vị trí nơi một chuỗi được chỉ định hoặc ký tự được tìm thấy, từ trái sang phải

=**SEARCH**(FindText, WithinText, [StartPosition], [NotFoundValue])

Hàm RELATED

RELATED()

Trả về các giá trị liên quan theo hàng của bảng dựa trên *relationships* với các bảng khác

=**RELATED**(ColumnName)

Cột chứa các giá trị bạn muốn truy xuất

Examples:

- *Product_Lookup[ProductName]*
- *Territory_Lookup[Country]*

NOTE

RELATED hoạt động gần như hàm **VLOOKUP** – nó sử dụng mối quan hệ giữa các bảng để đẩy dữ liệu từ một bảng sang một bảng khác như là một cột mới

Vì hàm này sử dụng *row context*, nên nó chỉ có thể sử dụng như một **calculated column** hoặc một phần của **iterator function** (*FILTER*, *SUMX*, *MAXX*, ...)

TIP:

Tránh sử dụng *RELATED* để tạo ra một *calculated columns* trừ khi bạn cần thiết phải dùng, vì nó tăng đáng kể kích thước file; thay vào đó, nên dùng *RELATED* trong *measure* để *FILTER* hoặc *SUMX*

HÀM MATH CƠ BẢN

SUM()

Trả về kết quả là một tổng

=**SUM**(ColumnName)

AVERAGE()

Trả về trung bình (trung bình số học) của tất cả các số trong một cột

=**AVERAGE**(ColumnName)

MAX()

Trả về giá trị lớn nhất trong một cột hoặc giữa hai biểu thức vô hướng

=**MAX**(ColumnName) or
=**MAX**(Scalar1, [Scalar2])

MIN()

Trả về giá trị nhỏ nhất trong một cột hoặc giữa hai biểu thức vô hướng

=**MIN**(ColumnName) or
=**MIN**(Scalar1, [Scalar2])

DIVIDE()

Thực hiện phép chia và trả về kết quả thay thế (hoặc để trống) nếu div/0

=**DIVIDE**(Numerator, Denominator,
[AlternateResult])

Hàm CALCULATE

CALCULATE()

Thực hiện một biểu thức hoặc công thức trong một điều kiện lọc

=CALCULATE(Expression, [Filter1], [Filter2],...)

Tên của một measure đã tồn tại, hoặc một công thức DAX

Ví dụ:

- [Total Orders]
- SUM>Returns_Data[ReturnQuantity])

Các biểu thức lọc

(**Note:** yêu cầu đơn giản, giá trị cố định; không thể tạo filter trên một measures khác)

Ví dụ:

- Territory_Lookup[Country] = “**USA**”
- Calendar[Year] > **1998**

TIP:

CALCULATE hoạt động tương tự **SUMIF** hoặc **COUNTIF** trong Excel, điểm khác biệt là nó có thể tính toán dựa trên mọi hàm đơn giản (không chỉ sum, count, ...); có thể gọi dễ hiểu hơn là “**CALCULATEIF**”

Hàm CALCULATE

```
X ✓ 1 BikeRevenue = CALCULATE([TotalRevenue], Product_Categories[CategoryName]="Bikes")
```

Ở đây, chúng ta tạo ra một measures mới tên là “**Bike Revenue**”, hàm sẽ thực hiện một measures khác là “**Total Revenue**” khi mà trường *CategoryName* trong bảng **Products** là “**Bikes**”

Hàm CALCULATE

CategoryName	Revanue	BikeRevanue
Accessories	507,330.99	8,468,854.53
Bikes	8,468,854.53	8,468,854.53
Clothing	209,263.93	8,468,854.53
Components		8,468,854.53
Total	9,185,449.45	8,468,854.53



TIP:

CALCULATE *chỉnh sửa* và *ghi đè lên* tất cả các filter context!

Trong ví dụ, “Clothing” có ngữ cảnh lọc filter context là CategoryName = “**Clothing**” (xác định bởi nhãn lọc) *and* CategoryName= “**Bikes**” (xác định bởi hàm CALCULATE)

Không thể có hai kết quả khác nhau cùng lúc, nên điều kiện lọc “**Clothing**” bị thay bằng điều kiện lọc “**Bikes**”. Calculate được ưu tiên sử dụng

Mức độ ưu tiên

CALCULATE

Thay đổi do CALCULATE

[CategoryName] = "Bikes"

Nếu measure có chứa hàm **CALCULATE**, filter context sẽ bị thay thế bởi điều kiện trong hàm ở **Step 1 & Step 2**

STEP 1

Filter context được phát hiện và apply

CategoryName	Total Returns	Bike Returns
Accessories	1,115	342
Bikes	342	342
Clothing	267	342
Components		342
Total	1,724	342

Products[CategoryName] = "Accessories"

Product Table
Accessories

STEP 2

Áp dụng filter này cho tất cả các bảng liên quan

Product Table
Bikes

Product Table
Bikes

1 1

AW_Sales_Data
Bikes

AW>Returns_Data
Bikes

STEP 3

Measure thay đổi dựa trên kết quả lọc

Total Returns = COUNTROWS(AW>Returns_Data)

Hàm ALL

ALL()

Trả về tất cả các hàng trong một bảng hoặc tất cả các giá trị trong một cột, bỏ qua mọi bộ lọc

=ALL(Table or ColumnName, [ColumnName1], [ColumnName2],...)

Bảng hoặc cột muốn bỏ đi các điều kiện lọc

Examples:

- Transactions
- Products[ProductCategory]

Bảng hoặc cột muốn bỏ đi các điều kiện lọc

Examples:

- Transactions
- Products[ProductCategory]

TIP:

Thay cho việc thêm vào filter context, hàm ALL loại bỏ nó. Nó được sử dụng khi bạn muốn bỏ các filter context mà sẽ bị thay đổi các hàm liên quan

Hàm FILTER

FILTER()

Trả về một bảng biểu thị một tập hợp con của một bảng hoặc biểu thức khác

=FILTER(Table, FilterExpression)

Tên Bảng để lọc

Examples:

- Territory_Lookup
- Customer_Lookup

Biểu thức lọc, theo các hàng của bảng

Examples:

- Territory_Lookup[Country] = “**USA**”
- Calendar[Year] = **1998**
- Products[Price] > [**Overall Avg Price**]

NOTE

FILTER được sử dụng để thêm một filter context mới, và có thể kiểm soát các điều kiện lọc phức tạp này tốt hơn cho CALCULATE

Vì FILTER trả về kết quả là một bảng, nó hay được sử dụng để tạo ra các *input* cho hàm

TIP:

Vì FILTER đọc dữ liệu theo từng hàng trong bảng, nó có thể cho một tốc độ xử lý chậm hơn; không sử dụng FILTER nếu hàm CALCULATE làm được chức năng tương tự

Hàm (“X”)

Hàm (or “X”) cho phép bạn lặp qua tính toán hoặc biểu thức tương tự trên mỗi hàng của một bảng và sau đó áp dụng một số công thức (SUM, MAX, v.v.)

==SUMX(Table, Expression)

Tên hàm

Examples:

- SUMX
- COUNTX
- AVERAGEX
- RANKX
- MAXX/MINX

Tên bảng lấy dữ liệu

Examples:

- Sales
- FILTER(Sales, RELATED(Products[Category])=“Clothing”)

Biểu thức

Examples:

- [Total Orders]
- Sales[RetailPrice] * Sales[Quantity]

Tên hàm

Examples:

- SUMX

TIP:

Tưởng tượng rằng hàm này sẽ thêm một cột tạm thời trong bảng, tính toán các giá trị trong mỗi hàng (dựa trên biểu thức) và sau đó tổng hợp giá trị (giống như cách SUMPRODUCT hoạt động)

HÀM TIME INTELLIGENCE

Các hàm **Time Intelligence** cho phép bạn dễ dàng tính toán với các so sánh thời gian phổ biến trong phân tích dữ liệu:

PerformanCe
To-Date

=**CALCULATE**(Measure, **DATESYTD**(Calendar[Date]))

↘ Sử dụng **DATESQTD** cho Quý hoặc **DATESMTD** cho Tháng

Previous
Period

=**CALCULATE**(Measure, **DATEADD**(Calendar[Date], -1, **MONTH**))

Chọn một kiểu thời gian (**DAY**, **MONTH**, **QUARTER**, or **YEAR**) và chọn mốc so sánh (ví dụ previous month, quay lại 10 ngày ...)

Running
Total

=**CALCULATE**(Measure, **DATESINPERIOD**(Calendar[Date], **MAX**(Calendar[Date]), -10, **DAY**))