

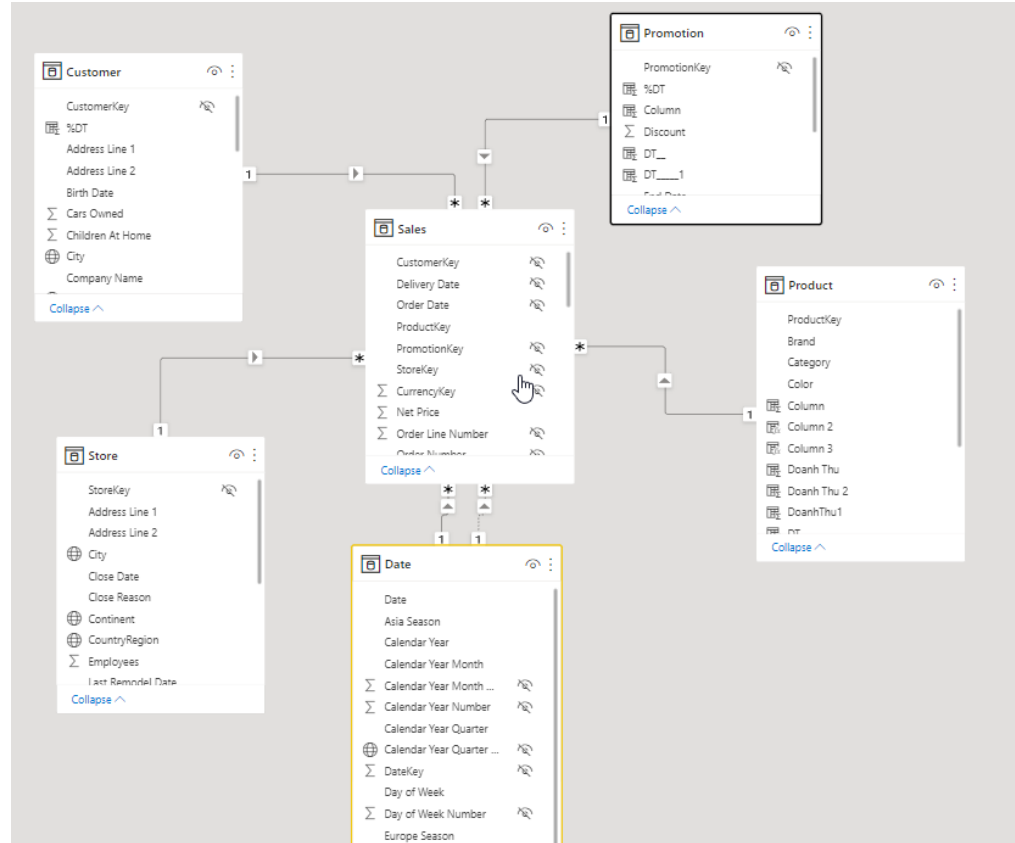


# DAX Fundamental

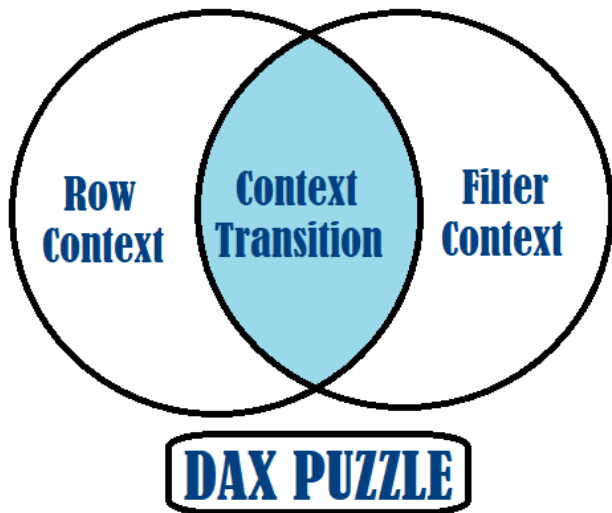
**Nguyễn Huy Hoàng**  
**Admin Data Analyst Skills (SQL, Power BI, ... )- Thực Hành Qua Tình Huống**

## DATASET SỬ DỤNG ĐỂ MÔ PHỎNG CÁC VÍ DỤ

- File thực hành sẽ được mình upload lên nha
- Mô hình STAR SCHEMA được sử dụng mô để mô phỏng các ví dụ
- Bên cạnh đó các bạn có thể sử dụng tool Bravo để xem kết cấu dữ liệu trong bảng và tool DAX Studio để demo các ví dụ hàm bảng



# CHAPTER 1: NGỮ CẢNH TRONG DAX



## 1. Row Context

- Bởi vì các **Table Data** trong Power BI được nối bằng các column database do đó chúng ta **chỉ có thể truy vấn cột** hay gọi 1 column trong power bi chứ **không thể thao tác từng dòng** trên power bi
- Row Context** là việc thực hiện các phép toán theo hướng **row by row**

Combine = [Category] & " - "&[Product Name]

Combine
Home Appliances - Adventure Works Floor Lamp X1150 Blue
Home Appliances - Adventure Works Floor Lamp M2150 Blue
Home Appliances - Adventure Works Chandelier M8150 Blue
Home Appliances - Adventure Works Chandelier M6150 Blue
Home Appliances - Adventure Works Wall Lamp E2150 Blue
Home Appliances - Adventure Works Wall Lamp E3150 Blue
Home Appliances - Adventure Works Desk Lamp E1300 Blue
Home Appliances - Adventure Works Desk Lamp E1200 Blue
Home Appliances - WWI Floor Lamp X115 Blue
Home Appliances - WWI Floor Lamp M215 Blue
Home Appliances - WWI Chandelier M815 Blue
Home Appliances - WWI Chandelier M615 Blue
Home Appliances - WWI Wall Lamp E215 Blue
Home Appliances - WWI Wall Lamp E315 Blue
Home Appliances - WWI Desk Lamp E130 Blue

→ Ở đây sẽ được hiểu là:  
Qua mỗi dòng chúng ta đang nối các dòng của cột Category và Product name lại với nhau theo row by row

## 2. Evaluation Context

- Bởi vì các **Table Data** trong Power BI được nối bằng các column database do đó chúng ta chỉ có thể nối chúng ta **có thể truy vấn cột** hay gọi 1 column trong power bi chứ **không thể thao tác từng dòng** trên power bi
- Evaluation Context** là việc tham chiếu 1 công thức lên tất cả các ô trong DAX.

Combine = [Category] & " - "&[Product Name]

Combine
Home Appliances - Adventure Works Floor Lamp X1150 Blue
Home Appliances - Adventure Works Floor Lamp M2150 Blue
Home Appliances - Adventure Works Chandelier M8150 Blue
Home Appliances - Adventure Works Chandelier M6150 Blue
Home Appliances - Adventure Works Wall Lamp E2150 Blue
Home Appliances - Adventure Works Wall Lamp E3150 Blue
Home Appliances - Adventure Works Desk Lamp E1300 Blue
Home Appliances - Adventure Works Desk Lamp E1200 Blue
Home Appliances - WWI Floor Lamp X115 Blue
Home Appliances - WWI Floor Lamp M215 Blue
Home Appliances - WWI Chandelier M815 Blue
Home Appliances - WWI Chandelier M615 Blue
Home Appliances - WWI Wall Lamp E215 Blue
Home Appliances - WWI Wall Lamp E315 Blue
Home Appliances - WWI Desk Lamp E130 Blue

Ở đây sẽ được hiểu là:  
Qua mỗi dòng chúng ta đang nối các dòng của cột Category và Product name lại với nhau theo row by row

## 2. Evaluation Context

- Một **ứng dụng khác** của Evaluation Context đó là gọi từng ô trên 1 column ra
- Việc **gọi từng ô** trên 1 column ra có thể giúp việc **kiểm tra dữ liệu** từ bảng dim đến bảng fact dễ dàng và cho thấy được ô nào đang bị lỗi dẫn đến các kết quả không mong muốn
- Ví dụ: bảng dim có SP A và SP A được bán qua các ngày các tháng. Nhưng khi chúng ta click filter vào SP A thì không trả về kết quả hoặc kết quả bị rỗng hoặc kết quả sai so với các tính toán ở bộ phận kế toán

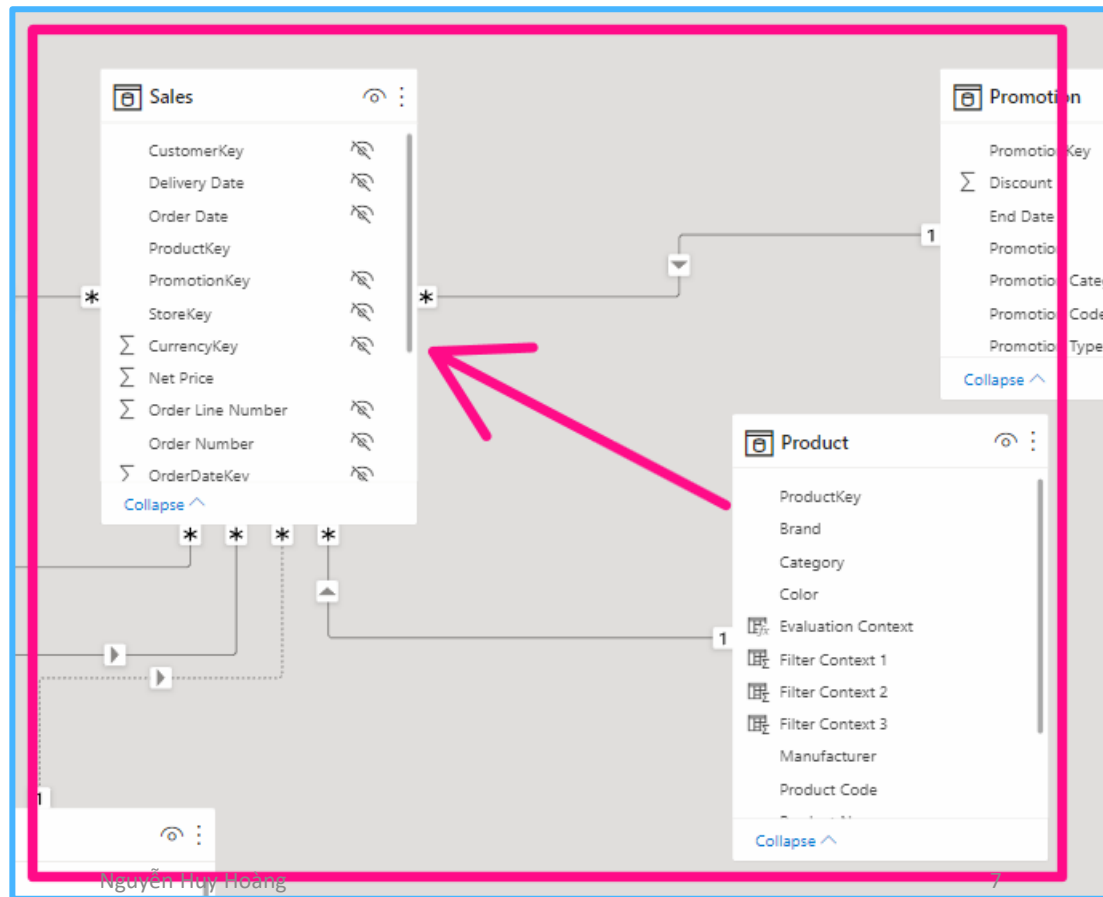
```
Ngày được mua gần nhất =  
//Gọi từng ô trên cột Product Key ra  
VAR CurrentKey = [ProductKey]  
/*  
   Qua mỗi ô thì CurrentKey thay đổi liên tục và nó đang đóng vai trò  
   như một Parameter và khi đưa vào điều kiện lọc thì điều kiện đó thay đổi  
   dựa trên sự thay đổi của các Parameter  
*/  
Return  
    CALCULATE(  
        MAX(Sales[Order Date]),  
        ALL(),  
        Sales[CustomerKey] = CurrentKey  
    )
```

Ngày được mua gần nhất
03/02/2007 12:00:00 AM
03/01/2007 12:00:00 AM
19/05/2009 12:00:00 AM
19/05/2009 12:00:00 AM
19/05/2009 12:00:00 AM

→ Kết quả thu được

### 3. Filter Context

- Filter Context** là việc chúng ta trả về các kết quả tính toán trên dữ liệu cho từng ngữ cảnh hiện hành
- Ví dụ ở đây chúng ta thấy bảng Product có Filter đến bảng sales theo hướng **one to many**
- Product[ProductKey] -> Sales[ProductKey]**
- Do đó chúng ta có thể vào bảng Product tính toán các giá trị cho từng ProductKey. Ví dụ ở đây là tính doanh thu cho từng ProductKey từ bảng Sales.



### 3. Filter Context and Context Transition

- **Context Transition** là việc chúng ta chuyển từ **Row Context** sang **Filter Context**
- Như chúng ta đã thấy thì Row Context nó chỉ thực hiện các tính toán theo Row by Row mà thôi. Do đó nếu không có **kích hoạt** Filter Context thì sẽ dẫn đến việc ghi đè một giá trị trên tất các ô
- Lấy ví dụ tính toán doanh thu cho từng ProductKey của bảng Product
- Việc tính toán hoàn toàn hợp lý nhưng kết quả lại ra không đúng. Mặc dù **bảng Product có Filter đến bảng Sales**

```
1 Doanh Thu = SUMX(Sales,Sales[Net Price]*Sales[Quantity])
```

ProductKey	Product Code	Product Name
743	0308001	Contoso Rechargeable Battery E100 Black
744	0308002	Contoso Dual USB Power Adapter - power adapter E300 Black
745	0308003	Contoso Car power adapter M90 Black
746	0308004	Contoso Notebook Peripheral Kit M69 Black
747	0308005	Contoso Mouse Lock Bundle E200 Black
748	0308006	Contoso Education Supplies Bundle E200 Black
749	0308007	Contoso Laptop Starter Bundle M200 Black
750	0308008	Contoso Education Essentials Bundle M300 Black
751	0308009	Contoso Desktop Alternative Bundle E200 Black
752	0308010	Contoso Power Inverter - DC to AC power inverter E900 Black
753	0308011	Contoso Smart Battery M901 Black
754	0308012	Contoso Laptop Cooling Hub notebook fan with 4 ports USB hub E80 Black
755	0308013	Contoso Home/Office Laptop Power Adapter E300 Black
756	0308014	Contoso USB 2.0 Dock Station docking station M800 Black
757	0308015	Contoso Enhanced Capacity Battery M800 Black
758	0308016	Contoso Connectivity Starter Kit Smart Buy M680 Black
759	0308017	Contoso 90W AC/DC Power Adapter E300 Black
760	0308018	Contoso Reserve Pen E200 Black
761	0308019	Contoso USB Data Link-direct connect adapter E600 Black
762	0308020	Contoso Primary Extended Capacity Battery Pack - notebook battery X100 Black
763	0308021	Contoso Digital camera accessory kit M200 Black
764	0308022	Contoso Leather Case - case for digital photo camera X20 Black
765	0308023	Contoso Lens cap E80 Black
766	0308024	Contoso Battery charger - bike E200 Black
767	0308025	Contoso USB Optical Mouse E200 Black
768	0308026	Contoso ADSL Modem Splitter/Filter X 1 F100 Black

[illegible]





**Vậy làm sao để kích hoạt Filter Context hay thực hiện Context Transition trong DAX ?**



1. Bọc hàm Calculate bên ngoài biểu thức
2. Dùng Measure. Bởi vì Measure có hàm calculate bọc ở ngoài
3. Dùng cách cổ điển để giải quyết (cho filter theo từng ProductKey thay vì kích hoạt )

### 3. Filter Context and Context Transition

```
1 Filter Context 1 =  
2 // Mục đích của hàm Calculate là kích hoạt Filter Context  
3 CALCULATE( SUMX(Sales,[Net Price]*[Quantity]))
```

```
Filter Context 2 =  
// Cổ điển tương tự như sumif Excel  
SUMX(  
    FILTER(  
        Sales,  
        Sales[ProductKey] = EARLIER('Product'[ProductKey])  
    ),  
    [Net Price] * [Quantity]  
)
```

```
DT =  
// New Measure  
SUMX(Sales,[Net Price]*[Quantity])
```

Filter Context 3 = [DT]

Filter Context 1	Filter Context 2	Filter Context 3
602	602	602
2,655	2,655	2655
985	985	985
1,067	1,067	1067
554	554	554
1,158	1,158	1158
419	419	419
144	144	144
182	182	182
365	365	365
1,630	1,630	1630
2,094	2,094	2094
2,253	2,253	2253
223	223	223
1,116	1,116	1116
929	929	929
1,188	1,188	1188
15	15	15
353	353	353
607	607	607
447	447	447
251	251	251

# CÁC KIẾN THỨC HỖ TRỢ TRONG DAX



Power BI

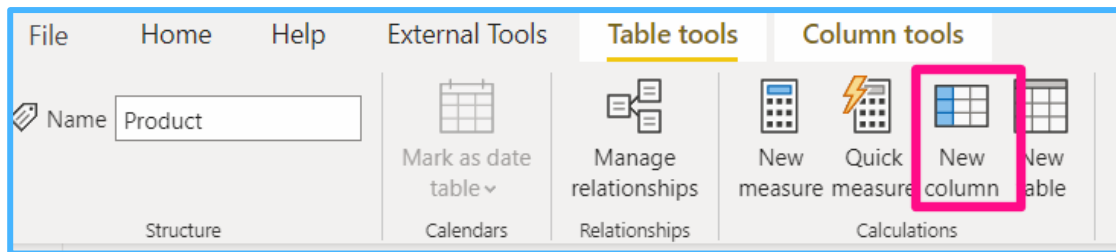


# 1. Các kiểu tính toán trong DAX

1. New Column
2. New Table
3. New Measure

## 1. Các kiểu tính toán trong DAX – New Column

## New Column: Là việc thêm một cột trong một Table của DAX



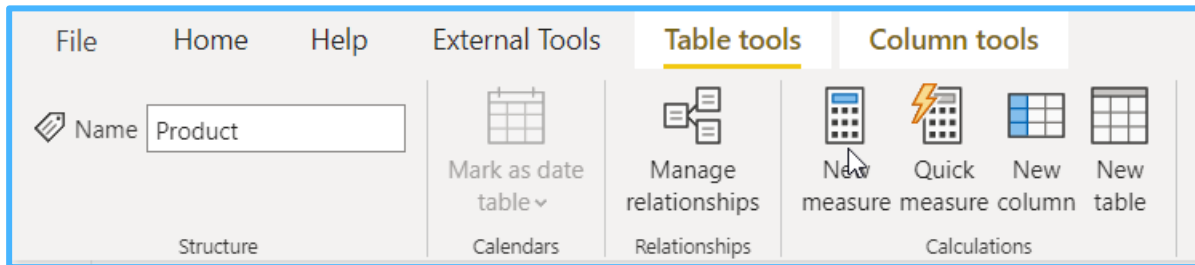
```
1 HHH =
2 // Thêm cột HHH vào bảng Product
3 [Brand] & " "&[Subcategory]
```

[illegible]

# 1. Các kiểu tính toán trong DAX – New Measure

**New Measure:** Là việc tính toán để đưa ra một con số hoặc một chuỗi string

Lưu ý: New Measure là nơi để lưu trữ các biểu thức tính toán khi đưa Measure vào Visual thì nó sẽ sử dụng các biểu thức tính toán đó để tính và trả về kết quả.



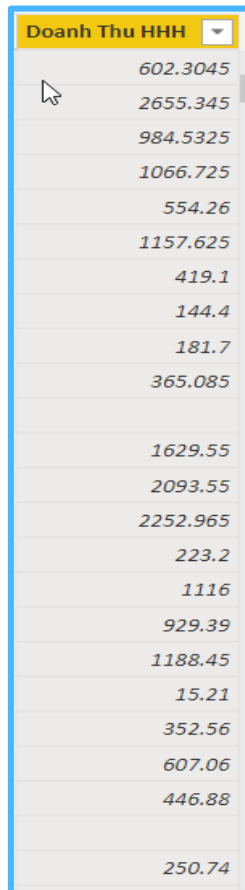
```
1 DT =  
2 // New Measure  
3 SUMX(Sales,[Net Price]*[Quantity])
```

# 1. Các kiểu tính toán trong DAX – New Measure

Vậy có thể đặt các Measure vào đâu ???

1. Vào các hàm bảng trong DAX (Summarize, Addcolumns, Filter,.....)
2. New Columns

```
1 Doanh Thu HHH =  
2 // Đặt Measure vào Column (thêm cột Doanh Thu HHH vào Table Product)  
3 [DT]
```



The screenshot shows a table with a single column titled 'Doanh Thu HHH'. The table contains 20 rows of numerical data. A mouse cursor is visible over the second row.

Doanh Thu HHH
602.3045
2655.345
984.5325
1066.725
554.26
1157.625
419.1
144.4
181.7
365.085
1629.55
2093.55
2252.965
223.2
1116
929.39
1188.45
15.21
352.56
607.06
446.88
250.74



# 1. Các kiểu tính toán trong DAX – New Measure

Vậy có thể đặt các Measure vào đâu ???

1. Vào các hàm bảng trong DAX (Summarize, Addcolumns, Filter,.....)

```
Table =  
// Tạo Table và thêm Measure vào  
ADDCOLUMNS(  
    VALUES('Product'[Category]),  
    "@DT", [DT]  
)
```

Category	@DT
Audio	384518.160300001
TV and Video	4392768.29230005
Computers	6741548.72819989
Cameras and camcorders	7192581.95279992
Cell phones	1604610.25989999
Music, Movies and Audio Books	314206.738000002
Games and Toys	360652.807400016
Home Appliances	9600457.03849981

# 1. Các kiểu tính toán trong DAX – New Measure

Vậy có thể đặt các Measure vào đâu ???

## 2. New Columns

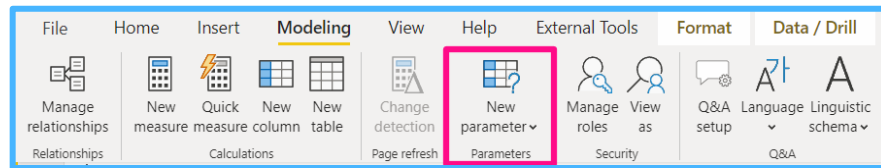
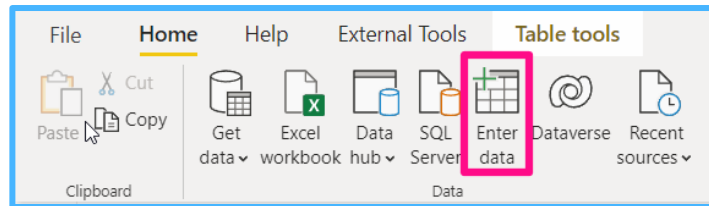
```
1 Doanh Thu HHH =  
2 // Đặt Measure vào Column (thêm cột Doanh Thu HHH vào Table Product)  
3 [DT]
```

Doanh Thu HHH
602.3045
2655.345
984.5325
1066.725
554.26
1157.625
419.1
144.4
181.7
365.085
1629.55
2093.55
2252.965
223.2
1116
929.39
1188.45
15.21
352.56
607.06
446.88
250.74

# 1. Các kiểu tính toán trong DAX – New Table

**New Table:** Tức là tạo một bảng mới từ một biểu thức DAX hoặc từ các nguồn khác như

1. Enter Data
2. Parameter (Table tạo Parameter)
3. New Table



```
Table =  
// Tạo Table và thêm Measure vào  
ADDCOLUMNS(  
    VALUES('Product'[Category]),  
    "@DT", [DT]  
)
```



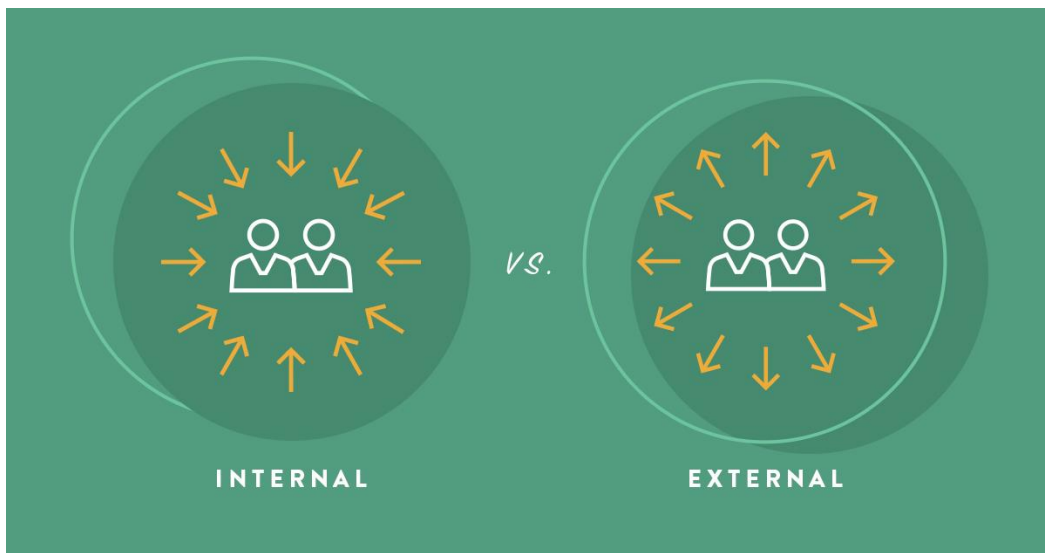
Category	@DT
Audio	384518.160300001
TV and Video	4392768.292300005
Computers	6741548.728199989
Cameras and camcorders	7192581.95279992
Cell phones	1604610.25989999
Music, Movies and Audio Books	314206.738000002
Games and Toys	360652.807400016
Home Appliances	9600457.03849981

## CHAPTER 2: Internal và External Filter (All, Allselected, Removefilter, Allexcept) kết hợp với Caculate, Caculatable



## 1. Các loại Filter

- Có hai loại Filter mà chúng ta cần quan tâm: **Internal Filter** và **External Filter**
- **Internal Filter:** Là các Filter nằm chung cùng một Visual với Measure
- **External Filter:** Là các Filter nằm ngoài Visual với Measure

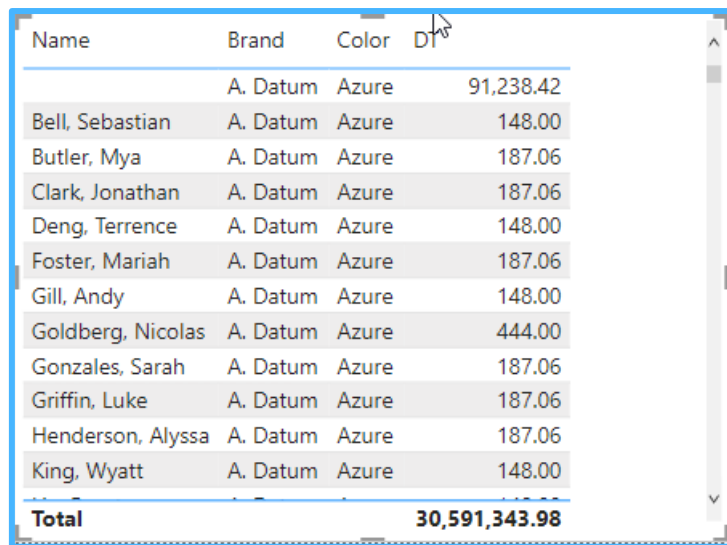


## 1. Các loại Filter – Internal Filter

- Internal Filter xuất phát từ những nơi nào ???
- Từ Visual nằm chung với Measure
- Từ các cột trong bảng Dimension có Filter đến bảng Fact
- Từ việc tạo các bảng ảo trong DAX (với các cột được lấy từ các bảng Dimension có Filter đến bảng Fact)

## 1. Các loại Filter – Internal Filter

- **Internal Filter xuất phát từ những nơi nào ???**
- Từ Visual nằm chung với Measure
- Ví dụ: Chúng ta tạo một Matrix gồm các trường Product[Name], Product[Brand], Product[Color] và Measure [DT]



Name	Brand	Color	DT
	A. Datum	Azure	91,238.42
Bell, Sebastian	A. Datum	Azure	148.00
Butler, Mya	A. Datum	Azure	187.06
Clark, Jonathan	A. Datum	Azure	187.06
Deng, Terrence	A. Datum	Azure	148.00
Foster, Mariah	A. Datum	Azure	187.06
Gill, Andy	A. Datum	Azure	148.00
Goldberg, Nicolas	A. Datum	Azure	444.00
Gonzales, Sarah	A. Datum	Azure	187.06
Griffin, Luke	A. Datum	Azure	187.06
Henderson, Alyssa	A. Datum	Azure	187.06
King, Wyatt	A. Datum	Azure	148.00
<b>Total</b>			<b>30,591,343.98</b>

Các trường Name, Brand, Color được gọi là các **Internal Filter**. Measure [DT] có biểu thức là  $SUMX(Sales, [Net Price] * [Quantity])$  bị Filter bởi các trường này

## 1. Các loại Filter – Internal Filter

- **Internal Filter** xuất phát từ những nơi nào ???
- Từ các cột trong bảng Dimension có Filter đến bảng Fact
- Ví dụ: Vào bảng Product tạo cột Doanh Thu HHH

```
1 Doanh Thu HHH =  
2 // Đặt Measure vào Column (thêm cột Doanh Thu HHH vào Table Product)  
3 [DT]
```

-> Các trường trong bảng Product đều được gọi là các Internal Filter



Doanh Thu HHH
602.3045
2655.345
984.5325
1066.725
554.26
1157.625
419.1
144.4
181.7
365.085
1629.55
2093.55
2252.965
223.2
1116
929.39
1188.45
15.21
352.56
607.06
446.88
250.74



## 1. Các loại Filter – Internal Filter

- **Internal Filter xuất phát từ những nơi nào ???**
- Từ việc tạo các bảng ảo trong DAX (với các cột được lấy từ các bảng Dimension có Filter đến bảng Fact)

```
Table =  
// Tạo Table và thêm Measure vào  
ADDCOLUMNS(  
    VALUES('Product'[Category]),  
    "@DT", [DT]  
)
```



Category	@DT
Audio	384518.160300001
TV and Video	4392768.29230005
Computers	6741548.72819989
Cameras and camcorders	7192581.95279992
Cell phones	1604610.25989999
Music, Movies and Audio Books	314206.738000002
Games and Toys	360652.807400016
Home Appliances	9600457.03849981

-> Trường Category được gọi là Internal Filter

## 1. Các loại Filter – External Filter

- External Filter xuất phát từ những nơi nào ???
- Từ các Visual ngoài Measure
- Ví dụ: Tạo một Matrix với các trường Name, Brand, Color và Measure DT

Các trường Name, Brand, Color đều được gọi là Internal Filter bởi vì chúng **nằm trong** một Visual với Measure [DT]

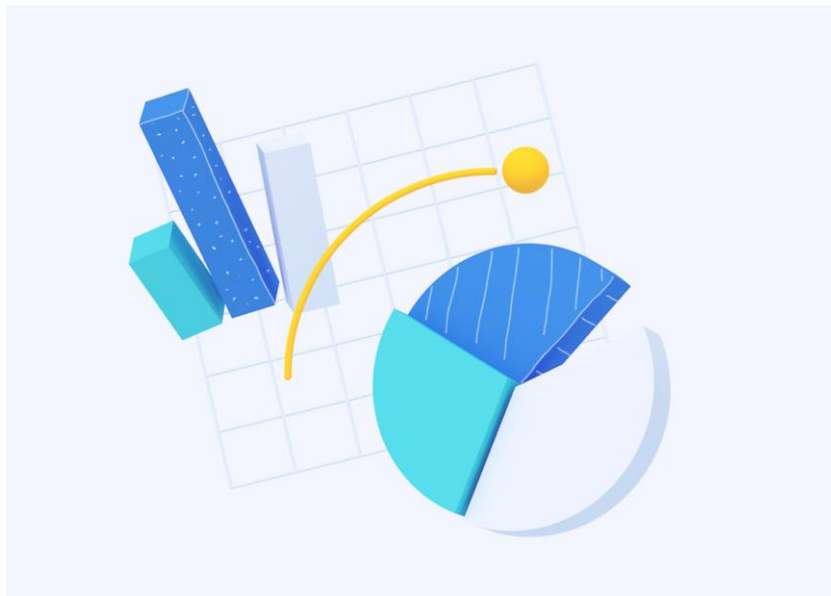
Các Slicer bên ngoài Matrix hoặc các biểu đồ... đều được gọi là các External Filter và chúng có các động đến Measure [DT] bởi vì chúng **nằm ngoài** Visual với Measure [DT]

The screenshot shows a Power BI report interface. At the top, there are two 'External Filter' slicers. The left slicer is for 'Category' and the right is for 'City'. Below them is a matrix table with columns: Name, Brand, Color, and DT. A blue arrow points from the 'Category' slicer to the matrix, and another blue arrow points from the 'City' slicer to the matrix. The matrix displays data for various individuals, all with 'A. Datum' as the Brand and 'Azure' as the Color. The total DT value is 30,591,343.98.

Name	Brand	Color	DT
	A. Datum	Azure	91,238.42
Bell, Sebastian	A. Datum	Azure	148.00
Butler, Mya	A. Datum	Azure	187.06
Clark, Jonathan	A. Datum	Azure	187.06
Deng, Terrence	A. Datum	Azure	148.00
Foster, Mariah	A. Datum	Azure	187.06
Gill, Andy	A. Datum	Azure	148.00
Goldberg, Nicolas	A. Datum	Azure	444.00
Gonzales, Sarah	A. Datum	Azure	187.06
Griffin, Luke	A. Datum	Azure	187.06
Henderson, Alyssa	A. Datum	Azure	187.06
King, Wyatt	A. Datum	Azure	148.00
<b>Total</b>			<b>30,591,343.98</b>

## 2. Các hàm bỏ bộ lọc

1. Hàm All, Removefilter
2. Hàm Allselected
3. Hàm Allexcept
4. Các định nghĩa lại bộ lọc
5. Cách sử dụng các hàm bỏ bộ lọc với hàm CALCULATE và CALCULATETABLE



## 2. Các hàm bỏ bộ lọc – ALL, REMOVEFILTER

- Nhắc lại khái niệm Internal và External Filter ??
- Hàm All và hàm Removefilter mặc dù tên hàm khác nhau nhưng về bản chất hầu như không có sự khác biệt.
- Hàm All, Removefilter: đều bỏ các Internal Filter và External Filter của một column trong một table, hoặc của tất cả các column trong một table hoặc tất cả các table.
- Ví dụ: Chúng ta tạo một Matrix với trường Product[Brand] và measure [DT All]

```
DT All =  
    CALCULATE(  
        [DT],  
        ALL('Product')  
    )
```

Brand	DT All
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>

## 2. Các hàm bỏ bộ lọc – ALL, REMOVEFILTER

- Chúng ta tiến hành tạo các External Filter từ các trường Product[Brand], Product[Category], Product[Name]

The screenshot displays a Power BI report interface with three 'External Filter' cards at the top and a data table below them.

**External Filter: Brand**

- ☐ A. Datum
- ☐ Adventure Works
- ☐ Contoso
- ☐ Fabrikam
- ☐ Litware
- ☐ Northwind Traders

**External Filter: Category**

- ☐ Audio
- ☐ Cameras and camcorders
- ☐ Cell phones
- ☐ Computers
- ☐ Games and Toys
- ☐ Home Appliances

**External Filter: Product Name**

- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...

**Data Table:**

Brand	DT All
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>

## 2. Các hàm bỏ bộ lọc – ALL, REMOVEFILTER



designed by freepik

**Khi nhấn các slicer từ các External Filter thì  
các giá trị trong matrix có thay đổi hay không  
???**

## 2. Các hàm bỏ bộ lọc – ALLSELECTED

- Nhắc lại khái niệm Internal ???
- Hàm Allselected r: bỏ các Internal Filter của một column trong một table, hoặc của tất cả các column trong một table hoặc tất cả các table.
- Ví dụ: Chúng ta tạo một Matrix với trường Product[Brand] và measure [DT Allselected]

```
DT Allselected =  
    CALCULATE(  
        [DT],  
        ALLSELECTED('Product')  
    )
```

Brand	DT Allselected
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>

## 2. Các hàm bỏ bộ lọc – ALLSELECTED

- Chúng ta tiến hành tạo các External Filter từ các trường Product[Brand], Product[Category], Product[Name]

### External Filter

**Brand**

- ☐ A. Datum
- ☐ Adventure Works
- ☐ Contoso
- ☐ Fabrikam
- ☐ Litware
- ☐ Northwind Traders

### External Filter

**Category**

- ☐ Audio
- ☐ Cameras and camcorders
- ☐ Cell phones
- ☐ Computers
- ☐ Games and Toys
- ☐ Home Appliances

### External Filter

**Product Name**

- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...

Brand	DT allselected
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>



## 2. Các hàm bỏ bộ lọc – ALLSELECTED



designed by freepik

**Khi nhấn các slicer từ các External Filter thì  
các giá trị trong matrix có thay đổi hay không  
???**

### 3. Các hàm bỏ bộ lọc – ALLEXCEPT

- Nhắc lại khái niệm Internal ???
- Hàm ALLEXCEPT : bỏ các Internal Filter và External của một column trong một table, hoặc của tất cả các column trong một table hoặc tất cả các table.
- Ví dụ: Chúng ta tạo một Matrix với trường Product[Brand] và measure [DT Allexcept]

```
DT allexcept =  
    CALCULATE(  
        [DT],  
        ALLEXCEPT('Product', 'Product'[Category])  
    )
```

Brand	DT allexcept
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>

### 3. Các hàm bỏ bộ lọc – ALLEXCEPT

- Chúng ta tiến hành tạo các External Filter từ các trường Product[Brand], Product[Category], Product[Name]

#### External Filter

**Brand**

- ☐ A. Datum
- ☐ Adventure Works
- ☐ Contoso
- ☐ Fabrikam
- ☐ Litware
- ☐ Northwind Traders

#### External Filter

**Category**

- ☐ Audio
- ☐ Cameras and camcorders
- ☐ Cell phones
- ☐ Computers
- ☐ Games and Toys
- ☐ Home Appliances

#### External Filter

**Product Name**

- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...
- ☐ A. Datum Advanced Digital Camera M3...

Brand	DT allexcept
A. Datum	30,591,343.98
Adventure Works	30,591,343.98
Contoso	30,591,343.98
Fabrikam	30,591,343.98
Litware	30,591,343.98
Northwind Traders	30,591,343.98
Proseware	30,591,343.98
Southridge Video	30,591,343.98
Tailspin Toys	30,591,343.98
The Phone Company	30,591,343.98
Wide World Importers	30,591,343.98
<b>Total</b>	<b>30,591,343.98</b>

### 3. Các hàm bỏ bộ lọc – ALLEXCEPT



designed by freepik

**Khi nhấn các slicer từ các External Filter thì  
các giá trị trong matrix có thay đổi hay không  
???**

**Liệu có External Filter nào ảnh hưởng đến  
Measure và làm thay đổi giá trị không**



## LƯU Ý QUAN TRỌNG KHI SỬ DỤNG CÁC HÀM BỎ BỘ LỌC

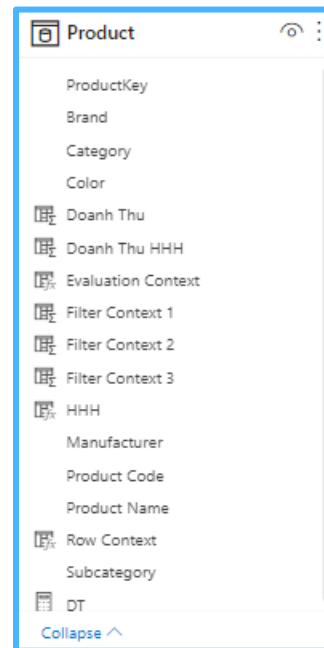
- Trong trường hợp chúng ta sử dụng các hàm bỏ bộ lọc ngoại trừ hàm ALLEXCEPT nếu không chỉ định bỏ bộ lọc ở bất kỳ bảng nào hoặc cột nào thì nó mặc định là bỏ tất cả các Filter đang Filter đến Measure mà chúng ta đang tính toán
- **Ví dụ: ALL(), ALLSELECTED(), REMOVEFILTERS()**
- Từ bản chất của các hàm mà chúng ta tự suy ra bỏ các Internal hay External Filter đang Filter đến Measure

## 4. Cách tái định nghĩa bộ lọc

- Sau khi sử dụng các hàm bỏ bộ lọc chúng ta có thể **sử dụng thêm** các hàm **Values**, **Distinct** và tái định nghĩa lại bộ lọc
- Tái định nghĩa lại bộ lọc được hiểu như thế vào: Ví dụ chúng ta đã bỏ các bộ lọc của bảng Product bằng hàm ALL hoặc hàm ALLSELECTED ( ALL(Product), ALLSELECTED(Product) )
- Lúc này chúng ta chỉnh lại và thêm vào đối số filter của hàm CALCULATE bằng hàm Values hoặc Distinct(Product[Category]) thì Measure đó bị lọc bởi trường Category trong bảng Product.
- Nhưng lưu ý các trường trong bảng Product có liên kết với nhau và tạo thành các Dependent Filter tức là trường Product[Color] bị Filter thì trường Product[Category] cũng bị Filter theo

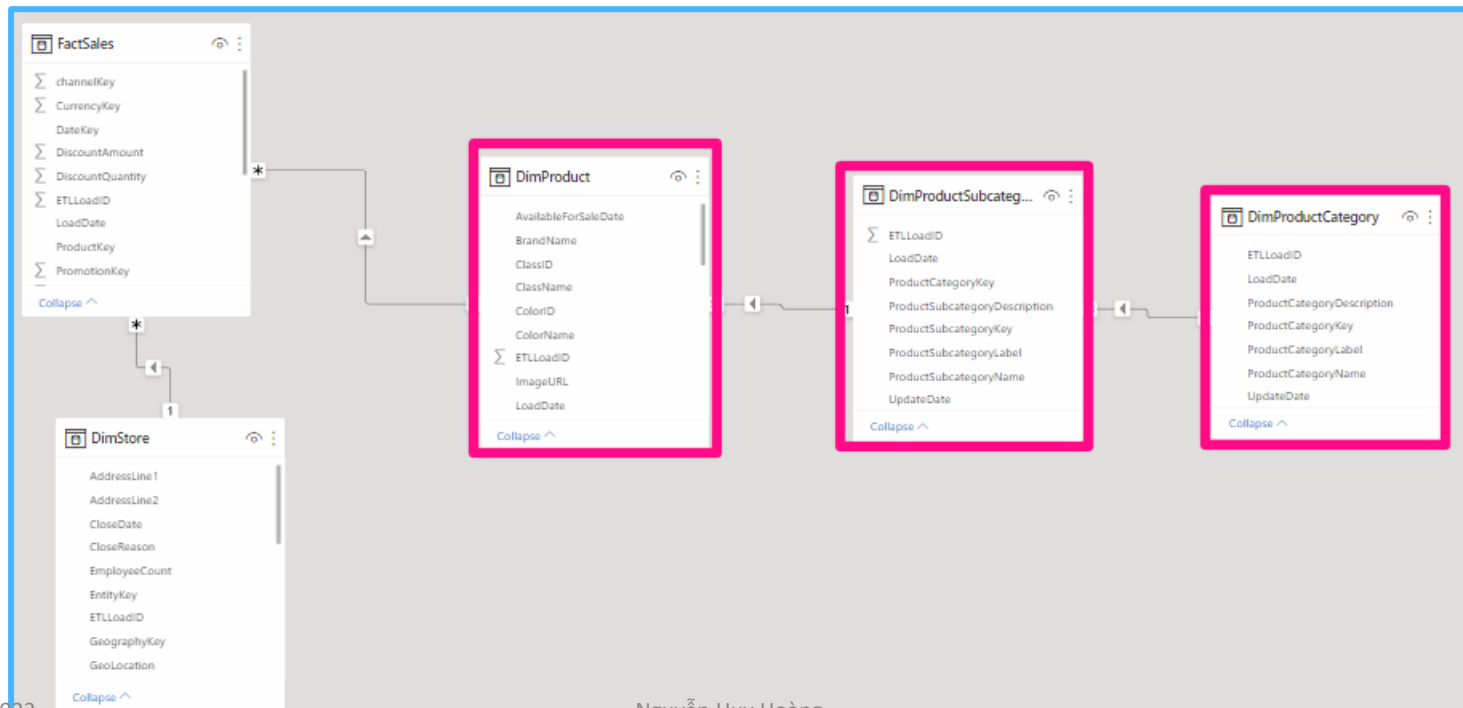
## 4. Cách tái định nghĩa bộ lọc – Dependent Filter

- Dependent Filter: là việc lọc filter ở trường này nó sẽ filter đến trường kia
- Ví dụ: Filter ở trường Product[Category] nó sẽ ảnh hưởng đến các trường Product[Color], Product[Brand], Product[SubCategory].....
- Như trong bảng Product các trường đều có liên kết với nhau việc Filter ở trường này sẽ ảnh hưởng đến trường khác
- Hoặc chúng ta có thể thấy trong Data Model dạng Snowflake...



## 4. Cách tái định nghĩa bộ lọc – Dependent Filter

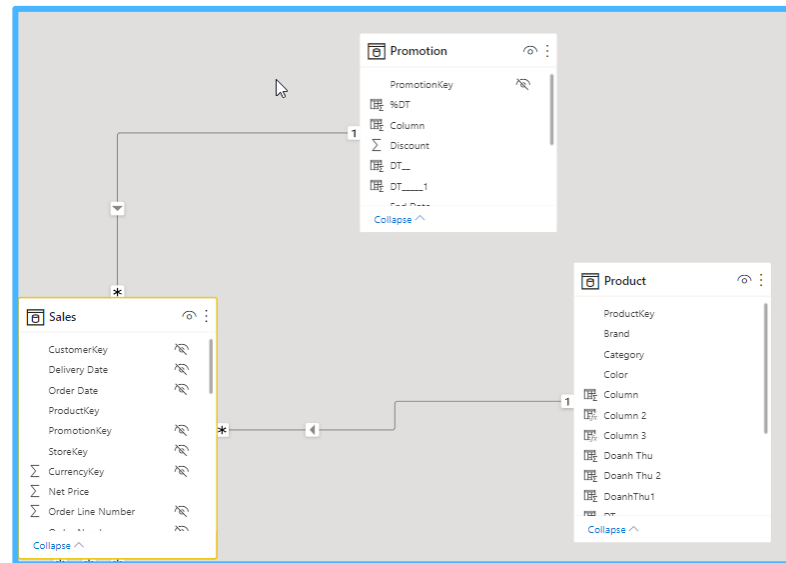
### Mô hình Snowflake





## 4. Cách tái định nghĩa bộ lọc – Independent Filter

- Independent Filter: là việc lọc filter ở trường này nó sẽ không filter đến các trường của bảng kia
- Ví dụ: Filter ở trường Product[Category], Product[Color], Product[Brand], Product[SubCategory].. Nó sẽ không ảnh hưởng đến các trường của bảng Promotion
- Bởi vì các trường trong bảng Product không nằm chung bảng với các trường của bảng Promotion và hai bảng không có liên kết relationship với nhau do đó các Filter từ các trường của bảng Promotion hoàn toàn độc lập so với các trường của bảng Product



## 5. Cách sử dụng các hàm bỏ bộ lọc với hàm CALCULATE và CALCULATETABLE

# CALCULATE

DAX Function (Filter) CONTEXT TRANSITION ⓘ

### Syntax

```
CALCULATE ( <Expression> [, <Filter> [, <Filter> [, ... ] ] ] )
```

- Theo DAX Guide: hàm Calculate có chức năng điều chỉnh ngữ cảnh bằng Filter
- Filter ở đây được hiểu là đưa vào một bộ lọc có thể sử dụng hàm bằng như Filter, Intersect, Values, Distinct (các hàm có liên quan đến định nghĩa lại bộ lọc)
- Và được sử dụng chung với các hàm bỏ bộ lọc như ALL, REMOVEFILTERS, ALLSELECTED, ALLEXCEPT..

```
Measure 2 =  
    CALCULATE(  
        [Doanh Thu],  
        FILTER(  
            'Date',  
            'Date'[Calendar Year] = "CY 2008"  
        ),  
        ALLSELECTED()  
    )
```

## 5. Cách sử dụng các hàm bộ lọc với hàm CALCULATE và CALCULATETABLE

# CALCULATETABLE

DAX Function (Filter) **CONTEXT TRANSITION** ⓘ

### Syntax



```
CALCULATETABLE ( <Table> [, <Filter> [, <Filter> [, ... ] ] ] )
```

- Tương tự như hàm CALCULATE nhưng tham số đầu tiên là một Table chứ không phải là một Expression để tính toán ra giá trị.

## 5. Cách sử dụng các hàm bỏ bộ lọc với hàm CALCULATE và CALCULATETABLE

### Cách hoạt động của tham số Filter trong các hàm CALCULATE và CALCULATETABLE

<b>Filter</b>	Optional Repeatable	A boolean (True/False) expression or a table expression that defines a filter.
---------------	------------------------	--

- Đối với các điều kiện True thì nó sẽ trả về kết quả, ví dụ như Filter các Customer[CustomerType] là Company chẳng hạn hoặc những người có thu nhập cao....
- Định nghĩa bộ lọc ở đây có thể hiện là truyền một table chứa các list bộ lọc hoặc các hàm tạo bộ lọc ảo như USERELATIONSHIP (active Relationship), TREATAS(Tạo bộ lọc ảo)...., hoặc các hàm bỏ bộ lọc

# CHAPTER 3: CÁC HÀM TÍNH TOÁN TRONG DAX

## *Các hàm X và các hàm thông thường trong DAX*



# CHAPTER 3: CÁC HÀM TÍNH TOÁN TRONG DAX

## *Các hàm X và các hàm thông thường trong DAX*

1. Các hàm X thường dùng
  - SUMX
  - MAXX
  - MINX
  - AVERAGEX
2. Các hàm thông thường
  - COUNTROWS
  - DISTINCTCOUNT
  - COUNTROWS phối hợp với DISTINCT



## TẠI SAO NÊN DÙNG SUMX THAY VÌ DÙNG SUM ???



- Bởi vì hàm SUM chỉ sử dụng với một cột chúng ta có thể thấy qua Syntax của hàm SUM

Syntax



```
SUM ( <ColumnName> )
```

- Do đó chúng ta không thể cộng trừ nhân trừ khi sử dụng hàm SUM
- Hàm SUM chỉ sử dụng với các bảng vật lý
- Còn SUMX chúng ta có thể cộng trừ nhân chia được điển hình là chúng ta hay tính doanh thu và tính toán trên cả bảng vật lý, bảng ảo khai báo bằng biến
- **SUMX(Sales, Sales[Net Price] \* Sales[Quantity])**



## CÁCH VẬN HÀNH CỦA CÁC HÀM X

**Ví dụ:** Chúng ta có một Matrix với MEASURE Doanh Thu được viết như thế này

`SUMX(Sales,[Net Price]*[Quantity])`

Quy trình 1: Bảng Sales sẽ bị lọc bởi đồng thời hai bảng Date và Product

Trường Date[Calendar Year] tác động lên bảng Date làm thay đổi bảng Date sau đó Filter đến bảng Sales tương tự với bảng Product

Quy trình 2: Sau khi bảng Sales bị Filter bởi từng ngữ cảnh của Date[Calendar Year] và Product[Color] thì tiến hành tính toán nhân từng Net Price với Quantity lại với nhau theo Row Context

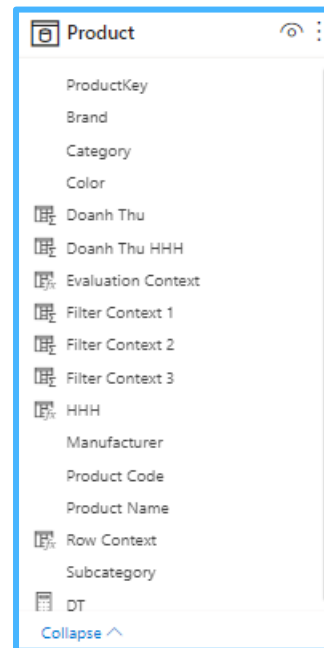
Quy trình 3: Tổng tất cả các giá trị và trả về ngữ cảnh hiện hành

Calendar Year	Color	Doanh Thu
CY 2007	Azure	27,895
CY 2008	Azure	33,072
CY 2009	Azure	36,423
CY 2007	Black	1,988,876
CY 2008	Black	1,955,354
CY 2009	Black	1,915,836
CY 2007	Blue	715,829
CY 2008	Blue	998,866
CY 2009	Blue	720,750
CY 2007	Brown	201,368
CY 2008	Brown	405,671
CY 2009	Brown	422,470
CY 2007	Gold	133,740
CY 2008	Gold	122,991
CY 2009	Gold	104,765
CY 2007	Green	429,480
CY 2008	Green	582,339
CY 2009	Green	391,365
CY 2007	Grey	1,419,966
CY 2008	Grey	1,016,073



## 4. Cách tái định nghĩa bộ lọc – Dependent Filter

- Dependent Filter: là việc lọc filter ở trường này nó sẽ filter đến trường kia
- Ví dụ: Filter ở trường Product[Category] nó sẽ ảnh hưởng đến các trường Product[Color], Product[Brand], Product[SubCategory].....
- Như trong bảng Product các trường đều có liên kết với nhau việc Filter ở trường này sẽ ảnh hưởng đến trường khác
- Hoặc chúng ta có thể thấy trong Data Model dạng Snowflake...



# 1. Các hàm X thông dụng - SUMX

## Syntax

SUMX ( <Table>, <Expression> )

- Table: Truyền một bảng vào có thể là một **bảng ảo** nằm trên biến hoặc là một bảng **vật lý**
- Expression: Được thực hiện dưới dạng Row Context, tính toán các cột dựa nằm trong Table đó

```
Doanh Thu = SUMX(Sales,[Net Price]*[Quantity])
```

```
Doanh Thu =  
VAR A =  
    ADDCOLUMNS(  
        SUMMARIZE(Sales, Customer[CountryRegion]),  
        "@DT",  
        [Doanh_Thu]  
    )  
Return  
    SUMX(A,[@DT])
```

## 1. Các hàm X thông dụng – AVERAGEX, MINX, MAXX

- **Các hàm trên cũng có cách vận hành tương tự như hàm SUMX**
- AVERAGEX: Tính giá trị trung bình trong một bảng vật lý hoặc một bảng ảo hoặc bảng vật lý
- MINX: Lấy ra giá trị nhỏ nhất trong một bảng vật lý hoặc một bảng ảo hoặc bảng vật lý
- MAXX: Lấy ra giá trị lớn nhất trong một bảng vật lý hoặc một bảng ảo hoặc bảng vật lý

## 2. Các hàm thông thường hay sử dụng - COUNTROWS

### Syntax

COUNTROWS ( [<Table>] )

- Hàm COUNTROWS được dùng để đếm số lượng dòng trong một bảng thực hoặc một bảng ảo được tạo ra bằng cách khai báo biến
- Bảng có thể chứa một hoặc nhiều cột

```
So GD = COUNTROWS(Sales)
```

```
So GD_T =  
VAR A =  
    ADDCOLUMNS(  
        SUMMARIZE(Sales, Customer[CountryRegion]),  
        "@DT",  
        [Doanh Thu]  
    )  
Return  
    COUNTROWS(A)
```

## 2. Các hàm thông thường hay sử dụng - DISTINCTCOUNT

### Syntax

```
DISTINCTCOUNT ( <ColumnName> )
```

- Hàm DISTINCTCOUNT được dùng để đếm số lượng dòng riêng biệt (tức là đã remove duplicate rồi) trong một bảng thực
- Bảng ở đây có thể chứa một hoặc nhiều cột

```
So Loai SP Da ban = DISTINCTCOUNT(Sales[ProductKey])
```



**Vậy làm sao có thể đếm các giá trị riêng biệt trên  
một bảng ảo khi không thể sử dụng hàm  
DISTINCTCOUNT??**



## CHÚNG TA CÓ THỂ DÙNG KẾT HỢP HÀM DISTINCT VỚI HÀM COUNTROWS

```
Countrows + Distinct =  
VAR A =  
    DISTINCT(  
        SELECTCOLUMNS(  
            ADDCOLUMNS(  
                SUMMARIZE(Sales, Customer[CountryRegion]),  
                "@DT",  
                [Doanh Thu]  
            ),  
            "@Country",  
            [CountryRegion]  
        )  
    )  
Return  
COUNTROWS(A)
```

# CHAPTER 4: CÁC HÀM BẢNG TRONG DAX





# CHAPTER 4: CÁC HÀM BẢNG TRONG DAX

1. Các sử dụng để nhóm, lấy và lọc dữ liệu
  - SUMMARIZE
  - ADDCOLUMNS
  - FILTER
  - GROUPBY
  - CROSSJOIN
2. Các hàm tìm điểm chung điểm khác biệt và nối dữ liệu
  - INTERSECT
  - EXCEPT
  - UNION
3. Các hàm liên quan đến bậc lọc (Values, Distinct)



**HÀM BẢNG LÀ GÌ ???**



**Hàm bảng là những hàm được sử dụng  
cho việc ra các bảng có thể sử dụng  
trong New Table, tạo bảng việc khai  
báo biến trong New Measure, New  
Column, New Table...**

# 1. Các hàm để nhóm, lấy và lọc dữ liệu – SUMMARIZE

## Syntax

```
SUMMARIZE ( <Table> [, <GroupBy_ColumnName> [, [<Name>] [,  
[<Expression>] [, <GroupBy_ColumnName> [, [<Name>] [,  
[<Expression>] [, ... ] ] ] ] ] ] )
```

- Theo DAX Guide hàm SUMMARIZE là hàm tạo một table tóm tắt được nhóm theo các cột chỉ định
- Các cột chỉ định ở đây sẽ được ưu tiên từ trái qua phải tức là nếu chúng ta đặt cột chỉ định nào trước thì sẽ ưu tiên nhóm theo cột đó trước
- Bên cạnh đó chúng ta có thể thêm một cột mới với hàm SUMAMRIZE và đặt các biểu thức tính toán như SUMX, AVERAGEX... và biểu thức này sẽ tự động chuyển từ ROW CONTEXT -> FILTER CONTEXT cho dù chúng ta không bọc hàm CALCULATE (Nhưng hiệu năng không tốt nên không được khuyến nghị khi sử dụng)

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – SUMMARIZE

- Hàm SUMMARIZE cho phép thêm một Column vào và đưa các biểu thức tính toán để tính toán các giá trị cho Column đó
- Việc đưa biểu thức vào như vậy không cần chuyển đổi từ Row Context sang Filter Context mà nó tự động chuyển. Do đó chúng ta không cần dùng các phương pháp Context Transition
- Nhưng cách này ít được sử dụng vì nếu lượng data lớn thì nó sẽ ảnh hưởng rất nhiều đến hiệu năng. Do đó dùng **ADDCOLUMNS** để thay thế là một giải pháp hữu hiệu

```
EVALUATE  
SUMMARIZE(  
    Sales,  
    'Product'[Subcategory],  
    'Product'[Category],  
    "@DT",  
    SUMX(Sales, Sales[Net Price]*Sales[Quantity])  
)
```

Subcategory	Category	@DT
MP4&MP3	Audio	170194.002200001
Recording Pen	Audio	89873.3650000003
Bluetooth Headphones	Audio	124450.793099999
Televisions	TV and Video	1834257.0543
VCD & DVD	TV and Video	428571.267
Home Theater System	TV and Video	1525526.261
Car Video	TV and Video	604413.709999998
Laptops	Computers	1925105.27650001
Desktops	Computers	1017127.267
Monitors	Computers	604286.220000000

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – ADDCOLUMNS

- Hàm ADDCOLUMNS cho phép thêm một hoặc nhiều cột mới, đặt tên cho nó và dung một biểu thức tính toán để tính toán các giá trị cho cột đó.
- Lưu ý khi đưa các biểu thức tính toán vào tham số Expression thì chúng ta phải thực hiện Context Transition. Bởi vì tham số này chỉ hoạt động dưới dạng Row Context chứ không có Filter Context
- Có 2 cách thường sử dụng: đó là bọc hàm calculate bên ngoài biểu thức hoặc là xây dựng Measure rồi đưa trực tiếp vào

### Syntax

```
ADDCOLUMNS ( <Table>, <Name>, <Expression> [, <Name>,  
<Expression> [, ... ] ] )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
<b>Table</b> ITERATOR ⓘ		The table to which new columns are added.
<b>Name</b>	Repeatable	The name of the new column to be added.
<b>Expression</b> ROW CONTEXT ⓘ	Repeatable	The expression for the new column to be added.

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – ADDCOLUMNS

### Trường hợp không bọc CALCULATE

```
EVALUATE  
  ADDCOLUMNS(  
    SUMMARIZE(  
      Sales,  
      'Product'[Subcategory],  
      'Product'[Category]  
    ),  
    "@DT",  
    SUMX(Sales, Sales[Net Price]*Sales[Quantity])  
  )
```

Subcategory	Category	@DT
MP4&MP3	Audio	30591343.9773999
Recording Pen	Audio	30591343.9773999
Bluetooth Headphones	Audio	30591343.9773999
Televisions	TV and Video	30591343.9773999
VCD & DVD	TV and Video	30591343.9773999
Home Theater System	TV and Video	30591343.9773999
Car Video	TV and Video	30591343.9773999
Laptops	Computers	30591343.9773999
Desktops	Computers	30591343.9773999
Monitors	Computers	30591343.9773999

### Trường hợp bọc CALCULATE

```
EVALUATE  
  ADDCOLUMNS(  
    SUMMARIZE(  
      Sales,  
      'Product'[Subcategory],  
      'Product'[Category]  
    ),  
    "@DT",  
    CALCULATE(  
      SUMX(Sales, Sales[Net Price]*Sales[Quantity])  
    )  
  )
```

Subcategory	Category	@DT
MP4&MP3	Audio	170194.002200001
Recording Pen	Audio	89873.3650000003
Bluetooth Headphones	Audio	124450.793099999
Televisions	TV and Video	1834257.0543
VCD & DVD	TV and Video	428571.267
Home Theater System	TV and Video	1525526.261
Car Video	TV and Video	604413.709999998
Laptops	Computers	1925105.27650001
Desktops	Computers	1017127.267
Monitors	Computers	604296.320000000

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – FILTER

- Theo DAX Guide: Hàm Filter là hàm trả về một bảng mà các giá trị đã được lọc theo điều kiện
- Ví dụ: Lọc ra danh sách những khách hàng có doanh thu > 50000....
- Table: Truyền vào nó một Table từ bảng vật lý hoặc các từ các hàm bảng hoặc từ các biến tạo ra bảng ảo
- Filter Expression là các điều kiện lọc của chúng ta (ví dụ lọc ra những khách hàng sống tại US..)
- Lưu ý Thay vì dùng ADDCOLUMNS để thêm cột Doanh Thu thì chúng ta có thể đưa Measure vào thẳng luôn

### Syntax

**FILTER ( <Table>, <FilterExpression> )**

PARAMETER	ATTRIBUTES	DESCRIPTION
<b>Table</b> ITERATOR ⓘ		The table to be filtered.
<b>FilterExpression</b> ROW CONTEXT ⓘ		A boolean (True/False) expression that is to be evaluated for each row of the table.

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – FILTER

### Trường hợp sử dụng ADDCOLUMNS

```
DEFINE
VAR A =
    ADDCOLUMNS(
        VALUES('Product'[Category]),
        "@DT",
        [Doanh Thu]
    )
EVALUATE
    FILTER(
        A,
        [@DT] > 30000000
    )
```

Category	@DT
TV and Video	30185201.7869
Computers	30591343.9774
Cameras and camcorders	30591343.9774
Cell phones	30403409.5189
Home Appliances	30591343.9774

### Trường hợp không sử dụng

```
EVALUATE
    FILTER(
        VALUES('Product'[Category]),
        [Doanh Thu] > 30000000
    )
```

Category
TV and Video
Computers
Cameras and camcorders
Cell phones
Home Appliances



## 1. Các hàm để nhóm, lấy và lọc dữ liệu – GROUPBY

- Theo DAX Guide: Hàm Groupby là hàm tạo ra một bảng mà được nhóm lại theo cột chỉ định
- Trông khái niệm có thể giống với hàm SUMMARIZE tuy nhiên trong thực tế thì lại khác
- Bởi vì hàm SUMMARIZE chỉ thao tác tính nhóm lại theo cột chỉ định một lần với các bảng vật lý và nếu sử dụng với bảng ảo thì nó không hiểu các ngữ cảnh hiện hành của cột được chỉ định nhóm lại
- Hàm Groupby vừa dùng được cho bảng ảo và bảng thực
- Thường hay sử dụng với hàm CURRENTGROUP()

### Syntax

```
GROUPBY ( <Table> [, <GroupBy_ColumnName> [, [<Name>] [, <Expression>] [, <GroupBy_ColumnName> [, [<Name>] [, <Expression>] [, ... ] ] ] ] ] )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
Table		The input table.
GroupBy_ColumnName	Optional Repeatable	A column to group by.
Name	Optional Repeatable	A column name to be added.
Expression	Optional Repeatable	The expression of the new column.

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – GROUPBY

### Trường hợp sử dụng GROUPBY

```
DEFINE
VAR A =
    ADDCOLUMNS(
        SUMMARIZE(
            Sales,
            'Product'[Category],
            'Date'[Calendar Year]
        ),
        "@DT",
        [Doanh Thu]
    )
EVALUATE
    GROUPBY(A, [Category], "@DT", SUMX(CURRENTGROUP(), [@DT]))
```

Category	@DT
Audio	29786953.4364002
TV and Video	30022164.4369002
Computers	30591343.9774002
Cameras and camcorders	30428306.6274002
Cell phones	30164392.5269002
Music, Movies and Audio Books	29221804.2963002
Games and Toys	28849752.9296002
Home Appliances	30428306.6274002

### Trường hợp không sử dụng

```
DEFINE
VAR A =
    ADDCOLUMNS(
        SUMMARIZE(
            Sales,
            'Product'[Category],
            'Date'[Calendar Year]
        ),
        "@DT",
        [Doanh Thu]
    )
EVALUATE
    SUMMARIZE(A, [Category], [@DT])
```

Category	@DT
Audio	11309946.1215001
TV and Video	11309946.1215001
Computers	11309946.1215001
Cameras and camcorders	11309946.1215001
Cell phones	11309946.1215001
Music, Movies and Audio Books	11309946.1215001
Games and Toys	11309946.1215001
Home Appliances	11309946.1215001
Audio	9764545.63850009

## 1. Các hàm để nhóm, lấy và lọc dữ liệu – CROSSJOIN

- Theo DAX Guide: Hàm CROSSJOIN là hàm trả về một bảng mà bảng đó được CROSS với một bảng chỉ định
- Nói một cách ngắn gọn nếu chúng ta có một bảng có m dòng, m' cột và một bảng có n dòng, n' cột sau khi sử dụng hàm crossjoin sẽ tạo ra một bảng gồm m x n dòng và n' cột
- Tương tự nếu chúng ta có nhiều bảng thì kết quả trả về:
- Số dòng = Số dòng tất cả các bảng nhân lại với nhau
- Số cột = Số cột tất cả các bảng nhân lại với nhau

### Syntax

```
CROSSJOIN ( <Table> [, <Table> [, ... ] ] )
```

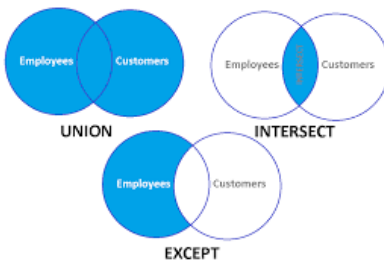
PARAMETER	ATTRIBUTES	DESCRIPTION
Table	Repeatable	A table that will participate in the crossjoin.

```
EVALUATE  
CROSSJOIN(  
VALUES('Product'[Category]),  
VALUES('Date'[Calendar Year]))
```

Category	Calendar Year
Cell phones	CY 2007
Music, Movies and Audio Books	CY 2007
Games and Toys	CY 2007
Home Appliances	CY 2007
Audio	CY 2008
TV and Video	CY 2008
Computers	CY 2008
Cameras and camcorders	CY 2008
Cell phones	CY 2008
Music, Movies and Audio Books	CY 2008

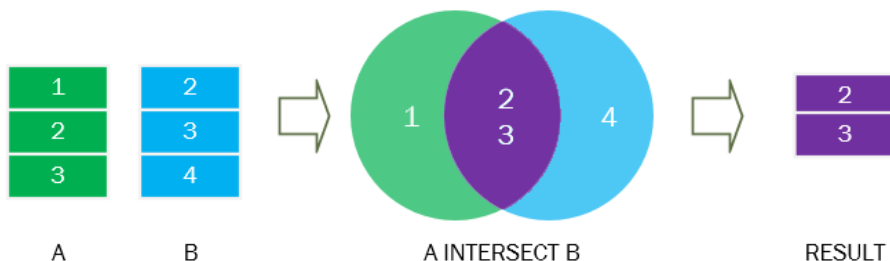


## CÁCH HOẠT ĐỘNG CỦA CÁC HÀM INTERSECT, UNION, EXCEPT



- Các hàm này thường sẽ ưu tiên các tham số bên trái
- Hàm INTERSECT như chúng ta thấy thì không có sự ưu tiên bởi vì nó đang tìm điểm chung giữa hai vùng dữ liệu. Tuy nhiên khi áp dụng thực tế thì INTERSECT có thể được dùng để tái kích hoạt FILTER CONTEXT
- Hàm EXCEPT thì sẽ ưu tiên tìm những điểm khác biệt của vùng dữ liệu bên trái so với bên phải. EXCEPT cũng được dùng để tái kích hoạt FILTER CONTEXT
- Nếu tên cột giữa hai bên có sự khác biệt thì sẽ ưu tiên lấy tên của cột bên trái

# 1. Các hàm tìm điểm giao điểm chung và nối dữ liệu – INTERSECT



- INTERSECT: Là hàm tìm điểm chung giữa hai tập dữ liệu

## Syntax

```
INTERSECT ( <LeftTable>, <RightTable> )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
LeftTable		The Left-side table expression to be used for Intersect.
RightTable		The Right-side table expression to be used for Intersect.

```
DEFINE
// Tìm những sản phẩm đã bán trong năm 2007 và năm 2008
VAR List_P2007 =
    SELECTCOLUMNS(
        CALCULATETABLE(
            VALUES(Sales[ProductKey]),
            'Date'[Calendar Year Number] = 2007
        ),
        "@P1",
        [ProductKey]
    )
VAR List_P2008 =
    SELECTCOLUMNS(
        CALCULATETABLE(
            VALUES(Sales[ProductKey]),
            'Date'[Calendar Year Number] = 2008
        ),
        "@P2",
        [ProductKey]
    )
EVALUATE
    INTERSECT(List_P2007, List_P2008)
```

@P1	864 rows
1715	
180	
1846	
2042	
52	
1334	
1664	
526	
591	

## 1. Các hàm tìm điểm giao điểm chung và nối dữ liệu – INTERSECT

### TÁI KÍCH HOẠT NGŨ CẢNH VỚI INTERSECT

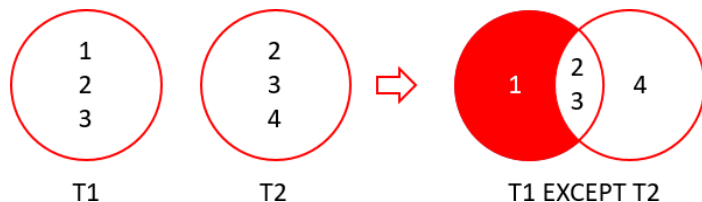
```
DEFINE
// Tìm doanh thu của những sản phẩm đã bán trong năm 2007 và năm 2008
VAR List_P2007 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2007
    )

VAR List_P2008 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2008
    )
EVALUATE
{
    CALCULATE(
        [Doanh Thu],
        INTERSECT(
            VALUES('Product'[ProductKey]),
            INTERSECT(List_P2007, List_P2008)
        )
    )
}
```

Value

15517416.2568999

## 1. Các hàm tìm điểm giao điểm chung và nối dữ liệu – INTERSECT



```
DEFINE
// Tìm những sản phẩm bán trong năm 2008 nhưng không được bán trong năm 2007
VAR List_P2007 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2007
    )

VAR List_P2008 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2008
    )

EVALUATE
    EXCEPT(List_P2008, List_P2007)
```

- EXCEPT: Là hàm tìm khác biệt tập dữ liệu 1 và tập dữ liệu 2

Syntax

```
EXCEPT ( <LeftTable>, <RightTable> )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
LeftTable		The Left-side table expression to be used for Except.
RightTable		The Right-side table expression to be used for Except.

ProductKey	614 rows
1683	
1245	
1246	
2096	
2323	
2062	
2145	
224	
1021	
1691	

## 1. Các hàm tìm điểm giao điểm chung và nối dữ liệu – INTERSECT

### TÁI KÍCH HOẠT NGŨ CẢNH VỚI EXCEPT

```
DEFINE
// Tính doanh thu những sản phẩm bán trong năm 2008 nhưng không được bán trong năm 2007
VAR List_P2007 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2007
    )

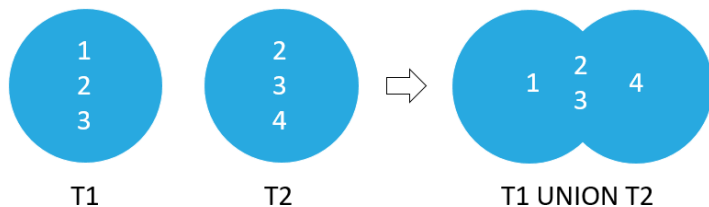
    VAR List_P2008 =
        CALCULATETABLE(
            VALUES(Sales[ProductKey]),
            'Date'[Calendar Year Number] = 2008
        )
EVALUATE
{
    CALCULATE(
        [Doanh Thu],
        INTERSECT(
            VALUES('Product'[ProductKey]),
            EXCEPT(List_P2007, List_P2008)
        )
    )
}
```

Value

5620619.75139998



## 1. Các hàm tìm điểm giao điểm chung và nối dữ liệu – UNION



- UNION: Là hàm nối hai tập dữ liệu với nhau với tên COLUMN được lấy ưu tiên từ column bên trái

```
DEFINE
// Nối các sản phẩm được bán trong năm 2007 và 2008 lại với nhau
// Lưu ý: Dữ liệu trả về không có unique và bị duplicate
VAR List_P2007 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2007
    )

VAR List_P2008 =
    CALCULATETABLE(
        VALUES(Sales[ProductKey]),
        'Date'[Calendar Year Number] = 2008
    )

EVALUATE
    UNION(List_P2007, List_P2008)
```

ProductKey
1715
180
1846
2042
52
1334
1664
246
526
1117

2736 rows