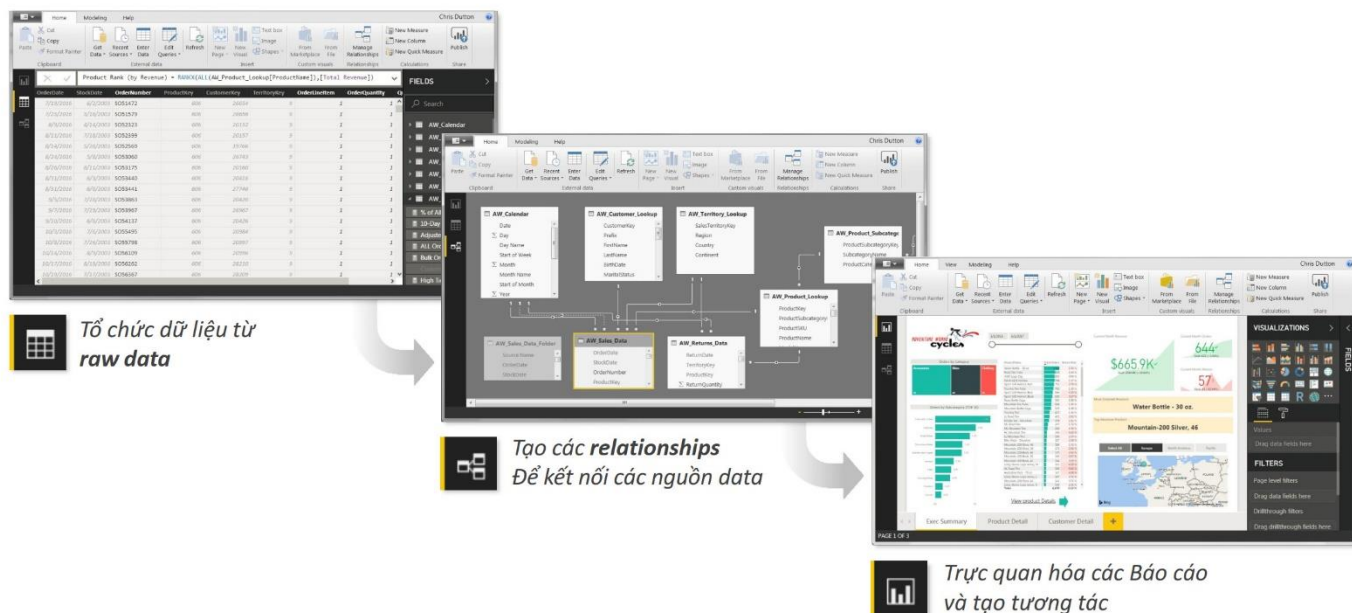


# BÀI 5: KẾT NỐI VÀ XỬ LÝ DỮ LIỆU TRONG POWER BI

## 1. Tổng quan về Power BI và cách cài đặt

Quy trình làm việc với Power BI bao gồm 3 bước

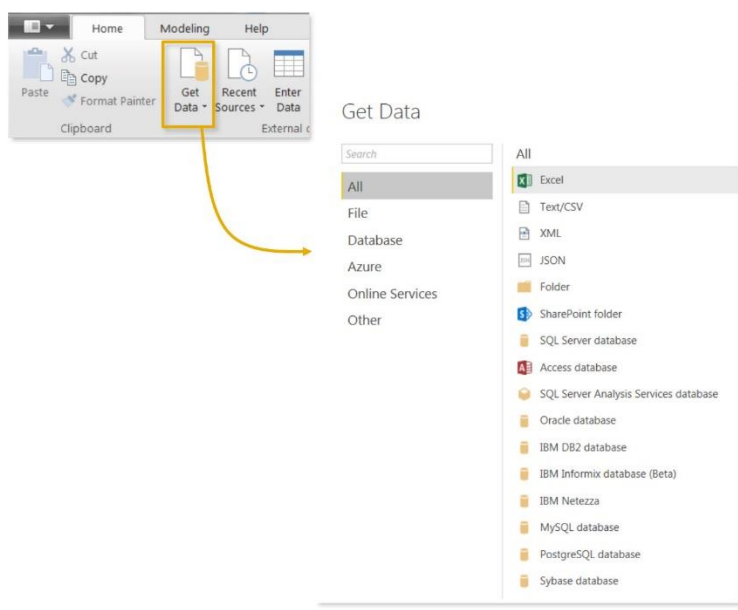


## 2. Kết nối dữ liệu

- Để kết nối dữ liệu từ bên ngoài vào Power BI, chúng ta truy cập vào Tab Home / Trong nhóm Data / Click vào nút Get Data

- Power BI có thể lấy dữ liệu từ nhiều nguồn khác nhau để xử lý:

1. Các file định dạng csv, xls, text, ...
2. Các folder
3. Các loại DataBase (Access, SQL, MySQL, Oracle, ....)
4. Các Online Services



Power BI có thể kết nối tới rất nhiều nguồn data, bao gồm (nhưng không hạn chế):

- **Flat files & Folders** (csv, text, xls, etc)
- **Databases** (SQL, Access, Oracle, IBM, Azure, etc)
- **Online Services** (Sharepoint, GitHub, Dynamics 365, Google Analytics, Salesforce, Power BI Service, etc)
- **Others** (Web feeds, R scripts, Spark, Hadoop, etc)

## 3. Xử lý dữ liệu:

Chúng ta có thể Load trực tiếp các file vào PBI để xử lý, nhưng thông thường chúng ta nên tinh chỉnh và tối ưu dữ liệu trước khi xử lý, muốn vậy chúng ta cần sử dụng đến Power Query Editor

### 3.1. Power Query Editor

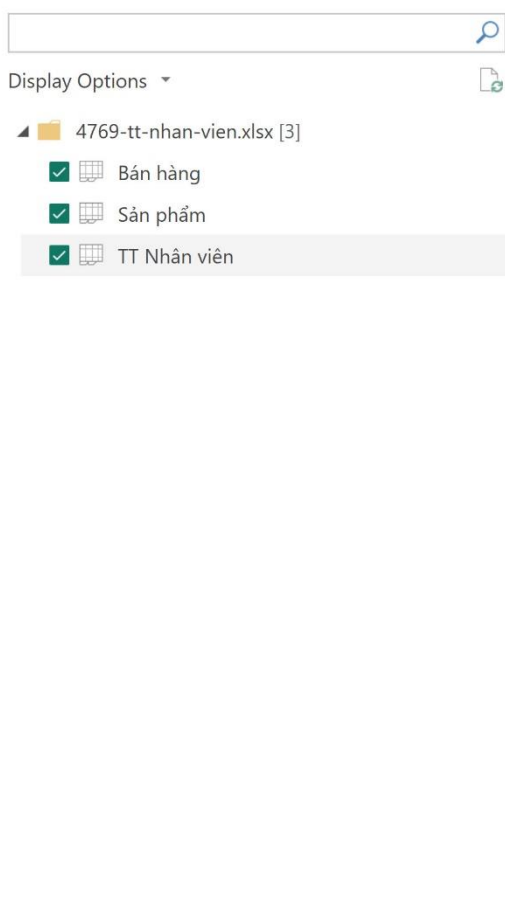
Là chương trình chỉnh sửa các file đầu vào, để tối ưu hoá dữ liệu để trực quan hoá thành dạng biểu đồ chúng ta cần phải thêm một số trường dữ liệu trung gian, hoặc trường dữ liệu mới.

Để truy cập vào Query Editor chúng ta sẽ có 2 cách:

#### **Cách 1:** Load và Transform Data ngay từ đầu

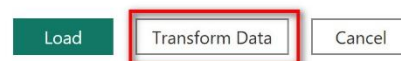
- Chọn Tab Home / Trong nhóm Data / Click vào nút Get Data / Chọn dạng file muốn load (giả sử file QuanlyBH.xlsx)
- Xuất hiện màn hình (Muốn xử lý bảng nào thì check vào bảng đó)

#### Navigator



#### TT Nhân viên

Mã nhân viên	Tên nhân viên	Chi nhánh
10001	Hoa	Chi nhánh 1
10002	Hà	Chi nhánh 1
10003	Quân	Chi nhánh 1
10004	Tiến	Chi nhánh 1
10005	Nhất	Chi nhánh 1
10006	Dương	Chi nhánh 1
10007	Dung	Chi nhánh 2
10008	Linh	Chi nhánh 2
10009	Phương	Chi nhánh 2
10010	Huy	Chi nhánh 2
10011	Ân	Chi nhánh 3
10012	Oanh	Chi nhánh 3
10013	Hải	Chi nhánh 3
10014	Khánh	Chi nhánh 3
10015	Ngọc	Chi nhánh 3
10016	Huyền	Chi nhánh 4
10017	Chinh	Chi nhánh 4
10018	Vân Anh	Chi nhánh 4
10019	Bách	Chi nhánh 4
10020	Hường	Chi nhánh 4

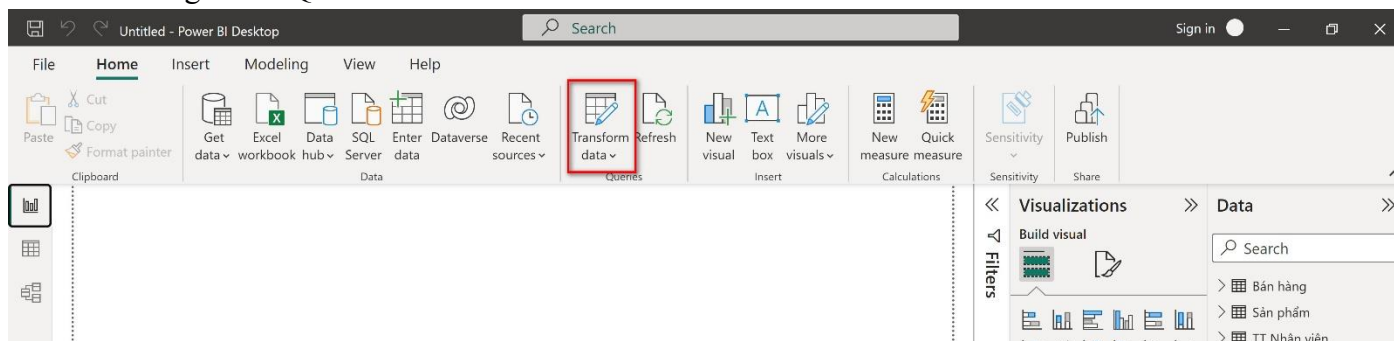


- Click vào nút **Transform Data** sẽ chuyển sang màn hình Query Editor.

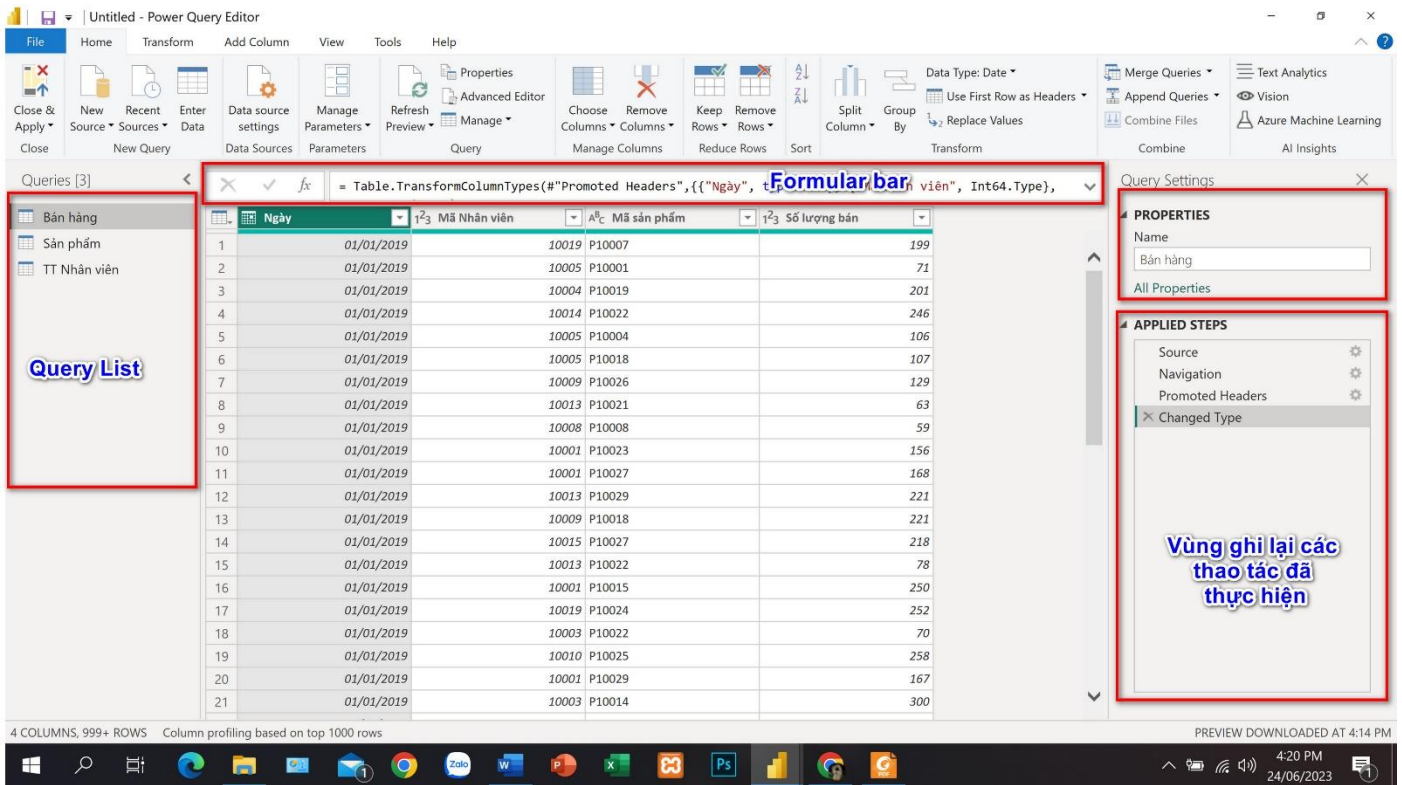
#### **Cách 2:** Load file dữ liệu xong và thực hiện Transform Data

Chúng ta vẫn thực hiện trình tự như Cách 1 tuy nhiên thay vì click vào nút **Transform Data** thì click nút **Load**.

- Sau đó Trong nhóm Queries / Click nút **Transform Data**



- Sẽ chuyển sang màn hình Query Editor như sau:



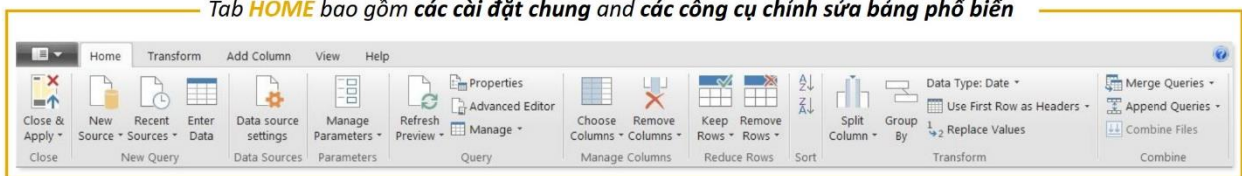
Trên giao diện của Query Editor có:

- Thanh Ribbon, chứa các Tab Home, Transform (giúp chỉnh sửa thông tin trong các cột (trường, field) của bảng), Add Column (Thêm các cột bổ sung cần thiết cho tính toán sau này).
- Khung Query List chứa các query chúng ta sẽ tạo ra sau khi chỉnh sửa các bảng.
- Thanh công thức (Formular Bar) dùng để lập các công thức tính toán.
- Khung Properties ( giúp thay đổi các thuộc tính cho query)
- Khung Applied Steps (Chứa các bước thay đổi khi có 1 chỉnh sửa nào đó trên Query)

### 3.2. Thao tác edit table cơ bản

## QUERY EDITING TOOLS

Tab **HOME** bao gồm các cài đặt chung and các công cụ chỉnh sửa bảng phổ biến



Tab **TRANSFORM** bao gồm các công cụ chỉnh sửa các bảng sẵn có đã tồn tại (chia/gộp, chỉnh sửa, nhóm xử lý text ...)



Tab **ADD COLUMN** bao gồm các công cụ tạo bảng mới (dựa trên một điều kiện, tính toán, công thức, ngày ...)



Những thao tác chỉnh sửa bảng cơ bản:

# CHỈNH SỬA BẢNG CƠ BẢN

**Sắp xếp dữ liệu (A-Z, Z-A, ...)**      **Đổi loại dữ liệu (date, \$, %, text, ...)**

**Cài đặt Header**

**Chọn hoặc xóa Cột**  
*Tip: sử dụng "Remove Other Columns" chỉ muốn giữ lại 1 cột nào đó*

**Giữ hoặc Xóa các Hàng**  
*Tip: sử dụng "Remove Duplicates" để xóa các dữ liệu trùng từ bảng cũ*

**Duplicate, move & rename cột**  
*Tip: Chuột phải để sử dụng các công cụ phổ biến*

Các thao tác trên Tab Home:

1. Trong nhóm Manage Column:
  - a. Nút Choose Columns cho phép chúng ta chọn 1 cột hay chọn nhiều cột để xử lý, hoặc chọn nhanh 1 cột bằng cách sử dụng Go to Column.
  - b. Nút Remove Columns: Cho phép chúng ta Xóa bớt các cột.
2. Trong nhóm Reduce Rows: Cho phép chúng ta giữ lại hoặc xóa bớt các dòng
3. Trong nhóm Transform:
  - a. Data Type: Dùng để thay đổi kiểu dữ liệu cho các cột do quá trình load vào PBI bị nhận dạng không đúng kiểu dữ liệu.
  - b. Use First Row as Header: Dùng để chọn dòng đầu tiên làm Header chứa tên các cột

## 3.2.1. Xử lý dữ liệu Text

**Chia một text column dựa trên một dấu ngăn cách hoặc ký tự đặc biệt**

**Extract characters từ một text column dựa trên chiều dài, ký tự đầu/cuối, dấu ngăn cách ...**  
*Tip: Chọn hai hoặc nhiều cột để gộp các trường (tương tự hàm concatenate trong excel)*

**Format a text column: upper, lower or proper case, hoặc thêm tiền tố - hậu tố**  
*Tip: Sử dụng "Trim" để loại bỏ spaces, hoặc "Clean" để xóa các ký tự không thể in*

**NOTE!**  
 Bạn có thể kết nối tới các công cụ trong cả "Transform" và "Add Column" - sự phân biệt chỉ ở chỗ bạn muốn thao tác trên một **cột sẵn có** hay **thêm một cột mới**

Muốn làm việc với dữ liệu kiểu Text chúng ta cần truy cập vào Tab Transform / Trong nhóm Text Column; bao gồm các công cụ sau:

- Nút Split Column cho phép ta tách một cột ra thành các cột khác nhau dựa vào:
  - By Delimiter: Tức là dựa vào 1 dấu phân cách chẳng hạn dấu , ; space ....
  - By Number of Character: Dựa vào số ký tự muốn chia
- Format: Cho phép chúng ta định dạng lại nội dung text của 1 cột: upper (Hoa tất cả); lower (Thường tất cả); thêm tiền tố Add Prefix; thêm hậu tố Add Suffix
- Extract: Dùng để trích xuất từ 1 text column dựa trên chiều dài; ký tự đầu/ cuối, dấu ngăn cách...

Bài tập thực hành: Xử lý file AdventureWorks\_Customers (Phần 2 trong Bài tập)

### 3.2.2. Xử lý dữ liệu Number

**Statistics functions** cho phép bạn thực hiện một số tính toán cơ bản từ cột được chọn (sum, min/max, average, count, ...)

**Note:** Những tools này sẽ trả về một Kết quả đơn (SINGLE value), và thường sử dụng để tính toán trên bảng thay vì sử dụng ở bước xử lý data (load data)

**Standard, Scientific và Trigonometry tools** cho phép bạn apply các công thức ở mức độ cơ bản (cộng, trừ, nhân, chia, ...) hoặc nâng cao ( Absolute (Tính giá trị tuyệt đối); Square Root (Khai căn); Exponent (hàm mũ) cho các giá trị trong 1 cột

**Note:** Khác với Statistics options, các công cụ này áp dụng cho các giá trị trong cột chứ không trả về một Kết quả đơn

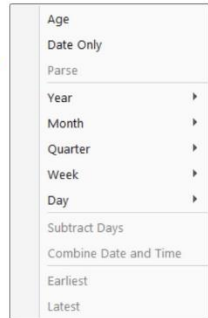
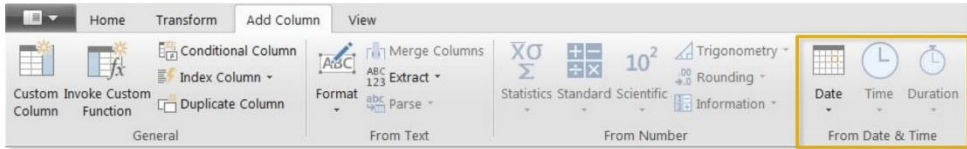
**Information tools** cho phép bạn xác định kết quả Logic hoặc Nhị phân (TRUE/FALSE or 1/0) dựa trên giá trị Chẵn/Lẻ

Tương tự như việc xử lý dữ liệu dạng Text, để xử lý dữ liệu dạng Number, chúng ta truy cập vào Tab Transform (Nếu muốn chỉnh sửa trực tiếp trên cột), hoặc truy cập vào Tab Add Column (Nếu muốn tạo ra cột mới).

- Công cụ Statistics: Cho phép chúng ta thực hiện các phép toán thống kê như: sum, min, max, ... từ một cột mà chúng ta đang chọn, nó chỉ trả về 1 kết quả đơn để đối chiếu mà thôi.
- Các công cụ khác như: Standard, Scientific, Trigonometry cho phép thực hiện các công thức ở mức độ cơ bản (cộng, trừ, nhân, chia...) hoặc nâng cao ( Absolute (Tính giá trị tuyệt đối); Square Root (Khai căn); Exponent (hàm mũ) cho các giá trị trong 1 cột.
- Công cụ Rounding (Làm tròn).
- Công cụ Information: Xác định kết quả đúng sai. True / False dựa trên giá trị chẵn lẻ

Bài tập thực hành: Xử lý file AdventureWorks\_Products (Phần 3 và 4 trong Bài tập)

### 3.2.3. Xử lý dữ liệu Date



**Date & Time** tools tương đối đơn giản, bao gồm một số tùy chọn như:

- **Age:** sự khác nhau giữa thời gian hiện tại và thời gian trong cột
- **Date Only:** Bỏ đi các thành phần khác ngoài định dạng Date trong cột
- **Year/Month/Quarter/Week/Day:** Xuất ra các thành phần từ trường Date
- **Earliest/Latest:** Trả về kết quả về dữ liệu Date sớm nhất hoặc muộn nhất từ Date Column dưới dạng Kết quả đơn

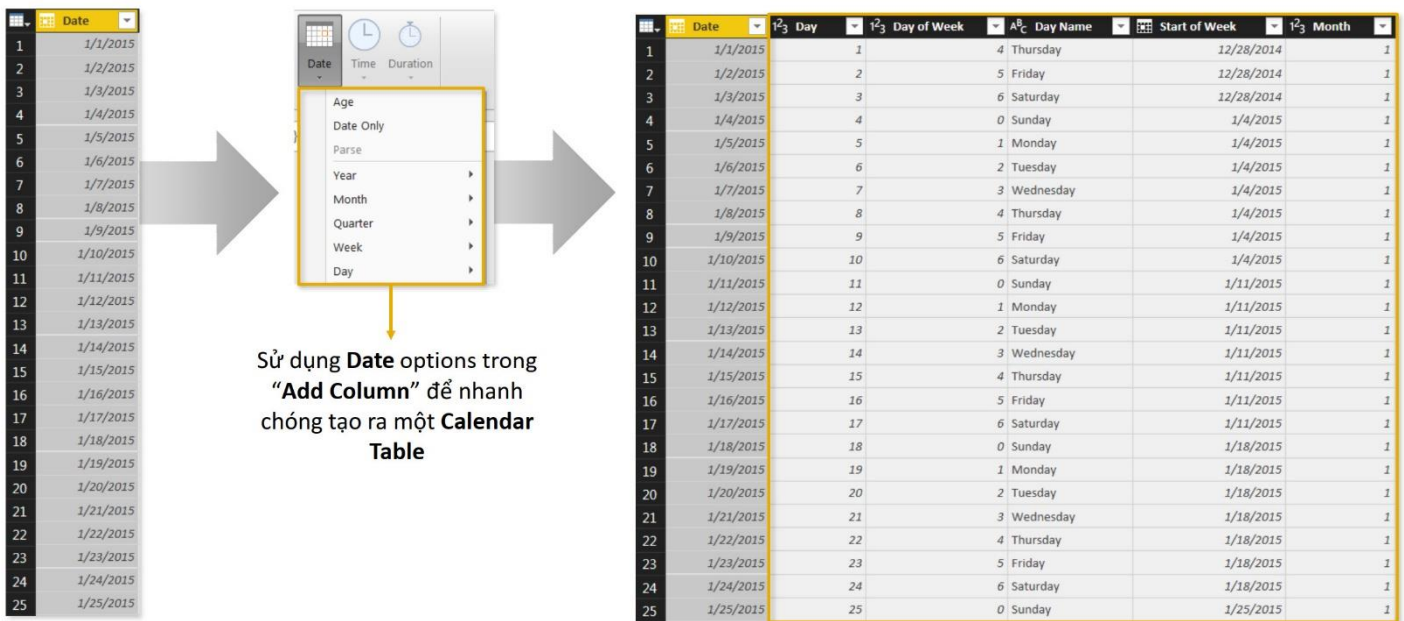
**Note:** Với dữ liệu dạng Date, chúng ta nên thực hiện "Add Column" để thêm một trường/cột mới, hơn là chỉnh sửa trực tiếp dữ liệu trong "Transform"



**PRO TIP:**

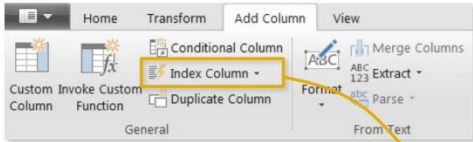
Nên tải một bảng chỉ chứa một cột **date column** và sử dụng Date tools để tạo ra một **calendar table** – sẽ cực kỳ hữu ích khi làm việc trong một file PowerBI có nhiều bảng

Để dễ thao tác với dữ liệu có những thống kê theo thời gian chúng ta nên tạo ra 1 Calendar Table như sau:



### 3.2.4. Thêm cột Index và Conditional

#### Thêm cột Index

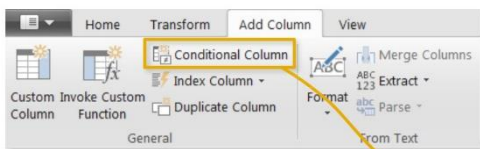


**Index Columns** chứa một list giá trị để xác định vị trí trong cột hoặc trong bảng (điển hình thường bắt đầu từ 0 hoặc 1)

Cột này cũng thường được dùng để tạo ra các **IDs** để có thể xác định quan hệ giữa các bảng (chi tiết ở phần sau của khóa học!)

Index	OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey
1	1/1/2015	9/21/2001	SO45080	332	14657
2	1/1/2015	12/5/2001	SO45079	312	29255
3	1/1/2015	10/29/2001	SO45082	350	11455
4	1/1/2015	11/16/2001	SO45081	338	26782
5	1/2/2015	12/15/2001	SO45083	312	14947
6	1/2/2015	10/12/2001	SO45084	310	29143
7	1/2/2015	12/18/2001	SO45086	314	18747
8	1/2/2015	10/9/2001	SO45085	312	18746
9	1/3/2015	10/3/2001	SO45093	312	18906
10	1/3/2015	9/29/2001	SO45090	310	29170
11	1/3/2015	12/11/2001	SO45088	345	11398
12	1/3/2015	10/24/2001	SO45092	313	18899
13	1/3/2015	12/16/2001	SO45089	351	25977
14	1/3/2015	10/26/2001	SO45091	314	18909
15	1/3/2015	9/11/2001	SO45087	350	11388
16	1/3/2015	9/11/2001	SO45094	310	22785
17	1/4/2015	10/30/2001	SO45096	312	12483
18	1/4/2015	10/30/2001	SO45097	313	29151

### Thêm cột điều kiện



**Conditional Columns** cho phép bạn tạo ra một trường mới dựa trên một nguyên tắc hoặc điều kiện nào đó (giả định **NẾU/THÌ**)

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name  
QuantityType

Column Name	Operator	Value	Output
If OrderQuantity	equals	ABC 123 1	Single Item
Else if OrderQuantity	is greater than	ABC 123 1	Multiple Items
Otherwise		ABC 123	Other

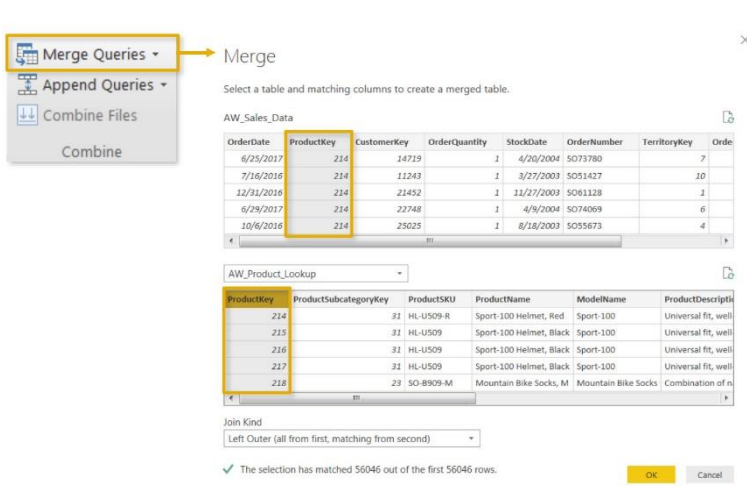
OK Cancel

### 3.2.5. Grouping Data

### 3.2.6. Gộp bảng

Có thể ghép nội dung các bảng lại với nhau, trong Power BI có 2 kiểu gộp như sau:

#### a. Merge Queries



Merging queries cho phép bạn **gộp bảng** dựa trên những cột chung (giống hàm **VLOOKUP** trong Excel)

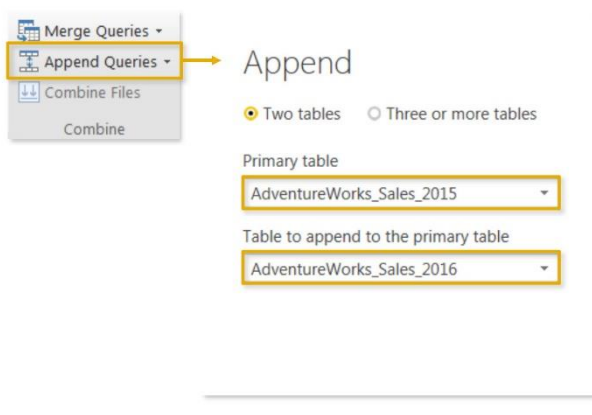
**NOTE:** Merging queries **thêm cột** tới một bảng đã có sẵn

#### NOTE!

Không bởi vì chúng ta có thể gộp bảng, mà chúng ta luôn gộp để có số lượng bảng ít.

Thực tế trong PBI, sẽ tốt hơn nếu chia ra thành các bảng và tạo **relationships** cho chúng (chi tiết hơn ở phần sau khóa học!)

## b. Append Queries



Appending queries cho phép bạn **combine** tables có cùng kiểu cấu trúc và kiểu định dạng

**NOTE:** Appending **thêm dòng** tới một bảng đã có sẵn

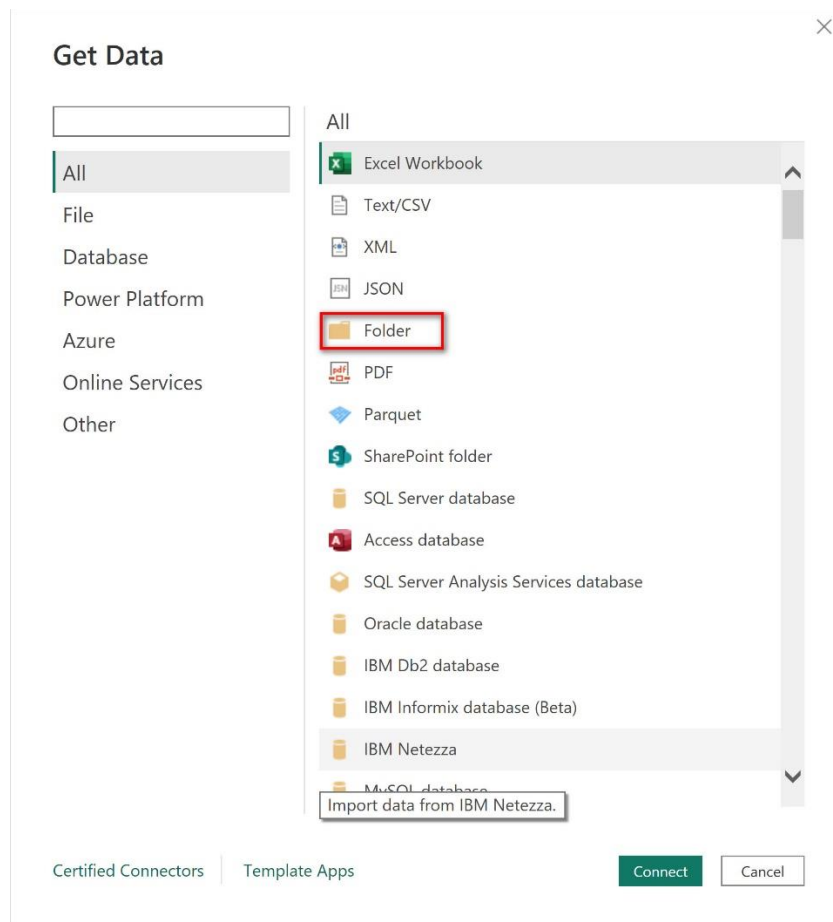


### PRO TIP:

Sử dụng "**Folder**" option (Get Data > More > Folder) để gộp các file trong một folder (nhưng phải có chung cấu trúc); nếu bạn thêm 01 file mới, bạn chỉ cần refresh queries và mọi thứ sẽ tự động update!

Cách Load nhiều file trong 1 folder để ghép lại thành 1 bảng:

B1: Trong Tab **Home** / chọn nhóm **Data** / Click nút **Get Data** / Chọn **More ...** xuất hiện hộp thoại

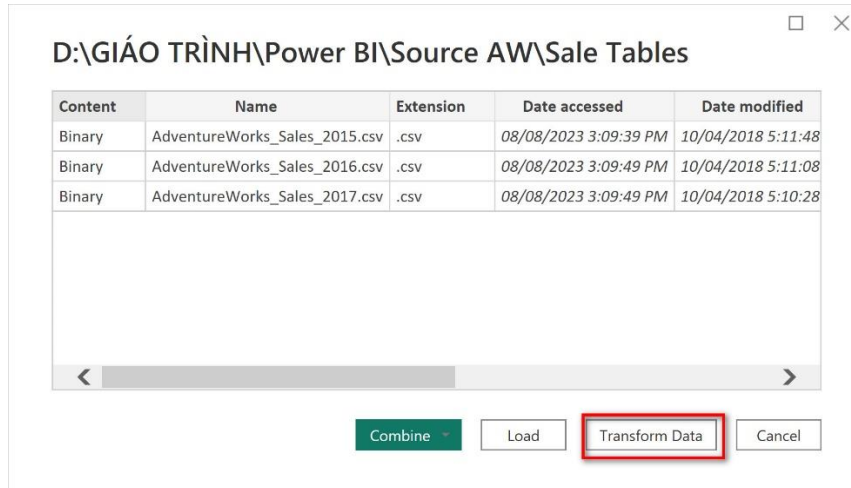


B2: Chọn Folder, xuất hiện hộp thoại bạn hãy chọn thư mục chứa các file nguồn muốn ghép, click Browse để chọn thư mục và OK

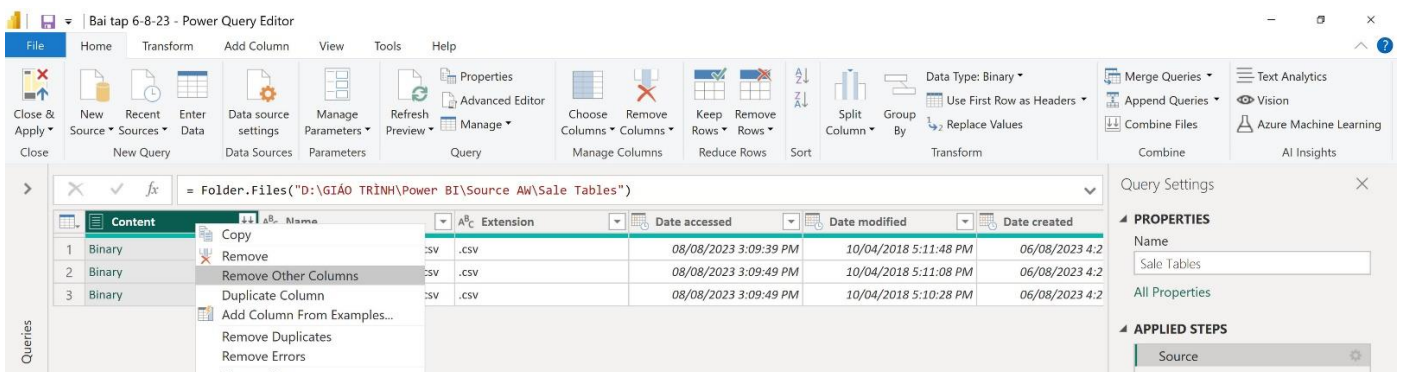




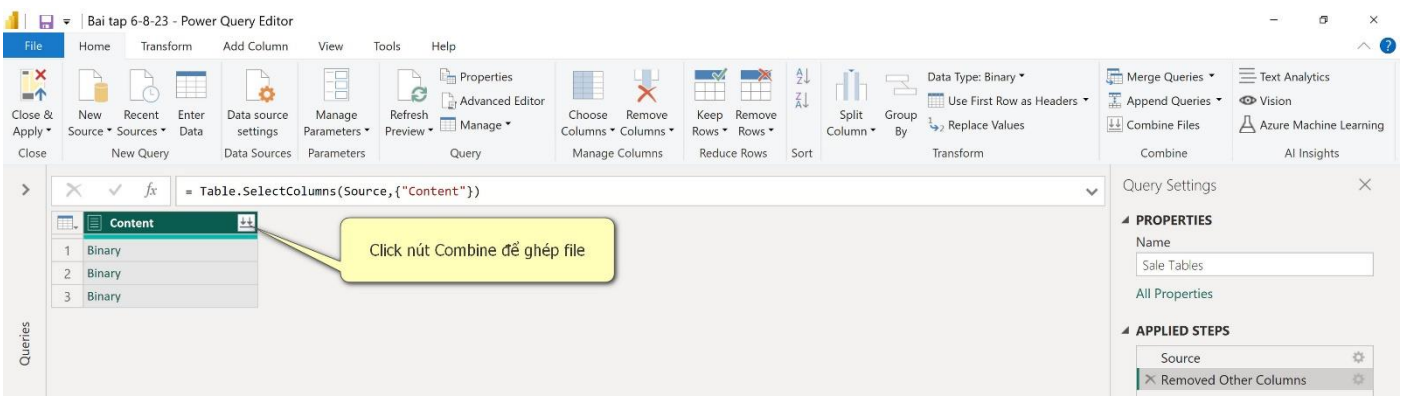
B3: Xuất hiện hộp thoại, click nút **Transform Data** để nạp dữ liệu chỉnh sửa



B4: Trong cửa sổ Query Editor chọn cột **Content** và loại bỏ các cột còn lại bằng cách click chuột phải vào cột này và chọn lệnh **Remove Other Columns**



B5: Ghép file



## 4. Tạo Table Relationship và Data Model

### 4.1. Chuẩn hoá DataBase

### 4.1.1. Data Model là gì?

Data Model còn gọi là “Mô hình dữ liệu” trong tất cả các hệ quản trị cơ sở dữ liệu đều sử dụng Data Model. Trong Power BI Data model nói một cách ngắn gọn là một tập hợp:

#### 1. Các bảng dữ liệu:

- Bảng dữ liệu này bao gồm một hoặc nhiều cột (dòng) được load vào trong Power BI (thông qua Power Query) vì một lý do nhất định (Chúng ta không load thừa dữ liệu từ nguồn nếu không có lý do làm vậy)

#### 2. Các mối quan hệ giữa các bảng được load vào cùng data model

3. Các công thức (measures) được viết ra để tính toán business logic, các chỉ số trong doanh nghiệp (ví dụ: margin, cost) bao gồm:

- Công thức và các định dạng áp dụng cho kết quả của công thức

- Tên chỉ số kinh doanh được gán cho công thức (Measure Name)

Tất cả những thành phần trên có thể được gọi là “Mô hình dữ liệu trong Power BI” (Điều này cũng đúng cho Analysis Service Tabular, Power Pivot trong Excel, và nhiều công cụ BI khác).

*Nếu chúng ta chỉ đưa vào các bảng mà không chuẩn hoá dữ liệu và không thiết lập quan hệ cho chúng thì không phải là 1 Data Model.*

### 4.1.2. Chuẩn hoá DataBase

Chuẩn hoá DataBase là 1 quá trình tổ chức lại dữ liệu trong các bảng, tại các cột trong một mối quan hệ dữ liệu, để giảm sự dư thừa và đảm bảo tính toàn vẹn. Thực hiện chuẩn hoá DataBase bao gồm các việc sau khi Load các bảng vào Query Editor như sau:

- Chọn lại đúng kiểu dữ liệu cho các cột

- Tách hoặc ghép cột cho phù hợp với nội dung lập báo cáo

- Xoá bớt các cột không cần thiết

- Mỗi bảng cần phải có 1 cột được chọn làm khoá chính (Primary key) để phân biệt các dòng, nếu không có chúng ta nên tạo thêm cột Index; Mỗi bảng phải chứa dữ liệu có 1 mục đích riêng biệt.

## 4.2. Tạo Relationship và Snowflake

### 4.2.1. Data Table và Lookup Table

Các bảng khi chúng ta đưa vào Data Model thông thường có 2 loại:

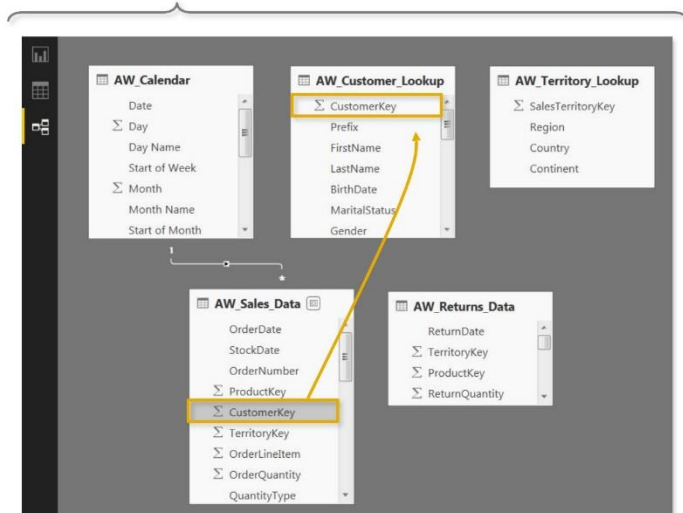
- **Data Table** (hoặc “Fact”) là các bảng chứa dữ liệu chính mà chúng ta thu thập được (Ví dụ: các Bảng kê hoá đơn bán hàng, Bảng kê xuất nhập kho).

- **Lookup Table** (hoặc “dimension”) là các bảng chứa các thông tin bổ sung cho **Data Table** (Ví dụ: Danh mục hàng, danh mục khách, kho, ...)

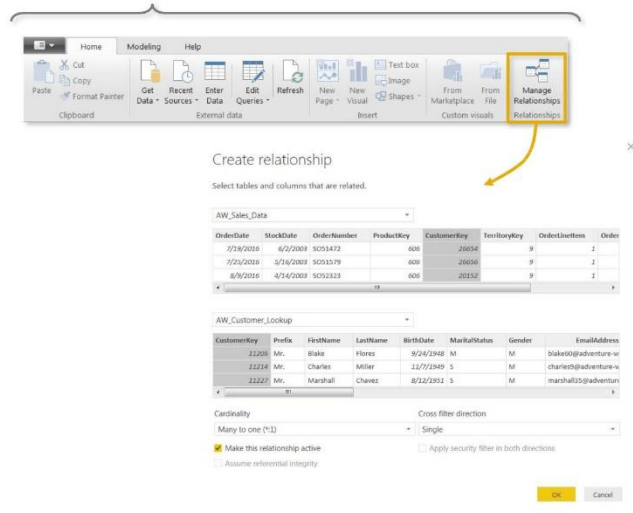
### 4.2.2. Tạo Relationship

Có 2 cách để tạo Relationship

**Option 1:** Click và kéo các trường dữ liệu chung giữa các bảng trong **Relationships** pane

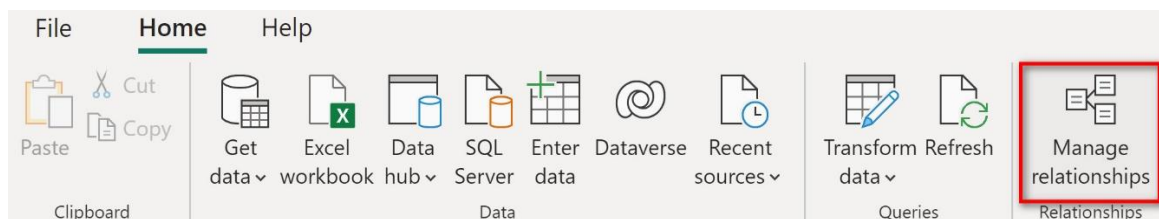


**Option 2:** Thêm hoặc phát hiện relationships sử dụng hộp thoại **“Manage Relationships”**

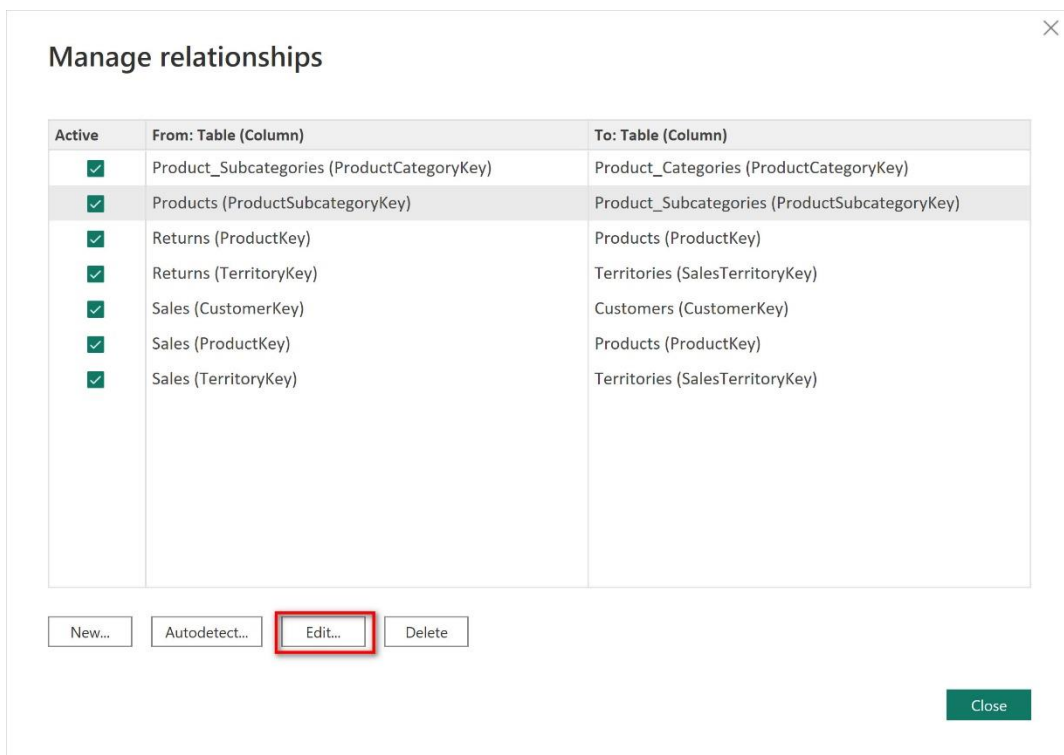


### 4.3. Các thao tác biên tập và quản lý Relationship

Để quản lý chỉnh sửa Relationship trong **Model View** chúng ta click chọn nút **Manage Relationships**



Xuất hiện cửa sổ **Manage Relationships**



Chúng ta có thể chọn một liên kết để chỉnh sửa bằng cách ấn vào nút **Edit...** (nếu muốn xoá mối quan hệ ấn nút **Delete...**) sẽ xuất hiện cửa sổ

### Edit relationship

Select tables and columns that are related.

Sales

OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	Order
03/01/2016	06/09/2002	SO48819	352	13622	9	1	
13/01/2016	19/09/2002	SO48901	352	13960	9	1	
14/01/2016	22/09/2002	SO48909	352	13953	9	1	

Customers

CustomerKey	Prefix	FirstName	LastName	BirthDate	MaritalStatus	Gender	
11206	Mr.	Blake	Flores	Friday, September 24, 1948	M	M	blake60
11214	Mr.	Charles	Miller	Monday, November 7, 1949	S	M	charles9
11227	Mr.	Marshall	Chavez	Sunday, August 12, 1951	S	M	marshal

Cardinality: Many to one (\*:1)  
Cross filter direction: Single

Make this relationship active  
 Assume referential integrity  
 Apply security filter in both directions

OK Cancel

Giả sử chúng ta chọn mối liên kết giữa bảng **Sales** và **Customers**, sẽ thấy được 2 cột khoá chính và khoá ngoại **Customer key** (Cột tô màu xám).

Chúng ta có thể chỉnh sửa một số nội dung như sau:

### CARDINALITY

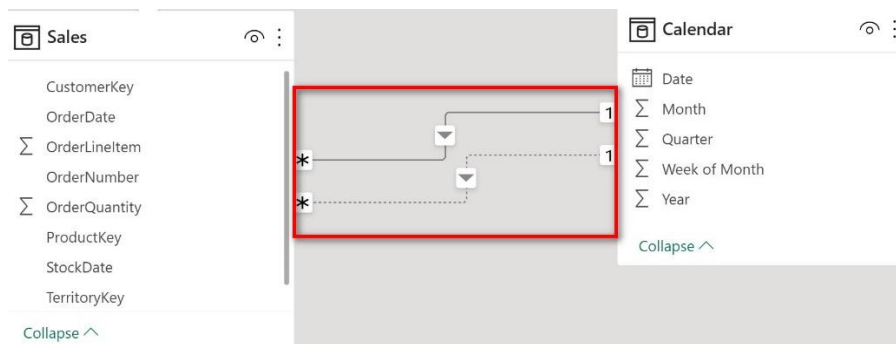
Cardinality chính là kiểu quan hệ giữa 2 Table có 1 số kiểu như sau:

1. Many to many (nhiều nhiều)
2. One to one (một một)
3. One to many (1 nhiều)

### ACTIVE và INACTIVE RELATIONSHIPS

Trong trường hợp Primary key (khoá chính) của 1 bảng liên kết đến 2 hoặc nhiều hơn các Foreign key (khoá ngoại) của 1 bảng khác, thì chỉ có 1 đường liên kết được hoạt động (Active) nếu muốn đường liên kết khác hoạt động thì chúng ta cần Inactive nó.

Trong ví dụ này chúng ta có bảng **Calendar** (khoá chính là **Date**) liên kết với bảng **Sales** (có 2 khoá ngoại **OrderDate** và **StockDate**) khi tạo relationship chúng ta thấy có 2 đường kết nối với kiểu quan hệ 1 – nhiều, đường liền nét đang được Active, đường chấm chấm chưa được kích hoạt.



Để kích hoạt 1 đường liên kết chúng ta click đúp vào đường liên kết và bật hoặc tắt mục **Make this relationship active**

#### ***4.4. Kết nối nhiều bảng trong Data Model***

Trong 1 Data Model chúng ta chỉ tạo liên kết giữa các Data Table với các Lookup Table, tuyệt đối không tạo liên kết giữa các Data Table với nhau.

*Ví dụ: Chúng ta có 2 Data Table **Sales** (chứa dữ liệu các đơn hàng đã bán) và **Returns** (chứa dữ liệu các hàng bán bị trả lại), thì không được tạo Relationship cho 2 bảng này.*

# BÀI 6: DAX TRONG POWER BI

## 1. Tổng quan về DAX

**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, Power Pivot
- 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

#### 1) Calculated Columns

EducationLevel	Occupation	HomeOwner	Full Name	User Name	Domain	HirementLevel	Parent	Co.
Partial College	Professional	Y	Mr. Blake Stone	blake90	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Charles Miller	charles5	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Marshall Chavez	marshall35	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Levi Chandra	levi1	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Sean Allen	sean49	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. James Walker	james96	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Cameron Yang	cameron23	Adventure Works	Average	Yes	
Partial College	Professional	N	Mr. Keith Raje	keith17	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Richard Coleman	richard81	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Robert Lewis	robert81	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Jonathan Robinson	jonathan72	Adventure Works	Average	Yes	
Partial College	Professional	Y	Mr. Robert Wang	robert182	Adventure Works	Average	Yes	
Partial College	Professional	N	Mr. Angel Scott	angel38	Adventure Works	Average	Yes	

#### 2) Measures

Measure Name	DAX Formula
Total Orders	DISTINCTCOUNT(Sales_Data[OrderNumber])
Total Revenue	SUMX(Sales_Data, Sales_Data[OrderQuantity] * RELATED(Product_Lookup[ProductPrice]))
Quantity Ordered	SUM(Sales_Data[OrderQuantity])

## 2. Các yếu tố khi sử dụng DAX

Chúng ta cần biết rằng có rất nhiều yếu tố quan trọng khác nhưng đây là 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, chúng ta cần biết cú pháp của DAX

- **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.

## 2. Calculated column và Measure trong Power BI

Các hàm DAX được sử dụng để có thể giúp chúng ta tạo các cột mới (New Column) hay còn gọi là các cột được tính (Calculated Columns), hoặc tạo ra các công thức (Measure), sự khác biệt giữa tạo ra 1 cột mới và tạo 1 measure ở vấn đề sau:

- Khi sử dụng DAX để tạo ra 1 cột mới, nội dung của các cột được xác định bởi biểu thức DAX được xét theo từng dòng (row, record), người ta gọi ngữ cảnh khi tạo ra cột được tính này là **Row Context**. Cột mới đó sẽ lưu trữ ngay trong Data Model, mỗi dòng đều sử dụng công thức DAX đã lập để tính ra giá trị của dòng tại cột mới đó, cho nên nếu bảng có quá nhiều dòng sẽ tốn nhiều thời gian tính toán hơn.

- Có một cách khác để xác định các phép tính trong mô hình DAX, hữu ích bất cứ khi nào chúng ta không muốn tính giá trị cho mỗi dòng mà thay vào đó, chúng ta muốn tổng hợp các giá trị từ nhiều hàng trong một bảng. Các tính toán này là các chỉ số. Gọi là các Measure nó sử dụng **Filter Context** có nghĩa là chúng ta chỉ dùng DAX tính toán 1 chỉ tiêu nào đó, ví dụ Doanh Thu (Lập 1 Measure tính Tổng doanh thu tất cả mặt hàng), nhưng khi chúng ta muốn thể hiện Doanh thu của 1 mặt hàng trong Pivot Table thì nó sẽ tự động áp dụng bộ lọc để lọc mặt hàng đó ra rồi mới áp measure đã tính ứng với mặt hàng này.

Có thể Tóm tắt việc sử dụng Calculated Columns và Measure trong bảng so sánh dưới đây:

## CALCULATED COLUMNS

- Giá trị được tính toán dựa trên tham chiếu tới vị trí theo hàng trong cột
- Nối các giá trị tính trong bảng và lưu trữ trong model (*làm tăng kích thước file*)
- Cập nhật khi data source được refresh hoặc thay đổi được tạo ra trong bảng
- Thường sử dụng để tạo thêm các **columns, slicers or filters**

Level	Occupation	HomeOwner	Full Name	User Name	Details	IncomeLevel	Parent
Page	Professional	Y	Mr. Blake Flores	blake40	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Charles Miller	charles8	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Marshall Chavez	marshall35	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Levi Chandra	levi1	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Sean Allen	sean45	Adventure Works	Average	Yes
Page	Professional	Y	Mr. James Walker	james26	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Cameron Wang	cameron23	Adventure Works	Average	Yes
Page	Professional	N	Mr. Keith Raje	keith17	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Richard Coleman	richard81	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Robert Lewis	robert81	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Jonathan Robinson	jonathan72	Adventure Works	Average	Yes
Page	Professional	Y	Mr. Robert Wang	robert36	Adventure Works	Average	Yes

Calculated columns "sống" trong tables

## MEASURES

- Giá trị được tính toán dựa vào các kết quả lọc từ các dữ liệu trong
- Không tạo ra các data mới trong model (*không làm tăng kích thước file*)
- Cập nhật khi các kết quả lọc từ các dữ liệu thay đổi trong báo cáo
- Thường sử dụng trong các trường **values** lên báo cáo hoặc biểu đồ



Measures "sống" trong visuals

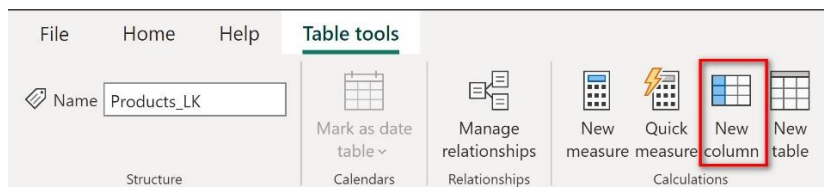
### 2.1. Thêm cột mới (Calculated Column)

Chúng ta sẽ sử dụng dữ liệu là file *Bai tap Version 2.pbix* (được sao chép từ file *Bai tap Version 2.pbix* là file chúng ta đã thực hành theo bài tập của chương trước).

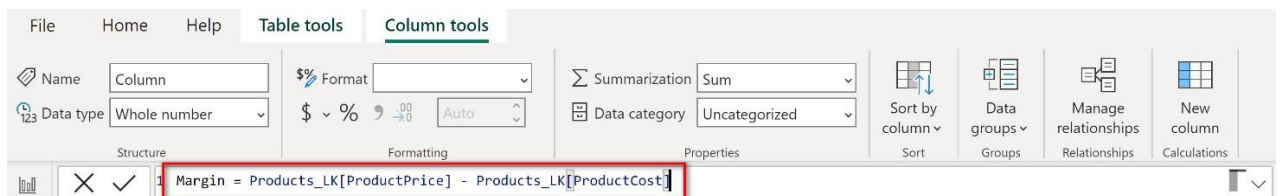
Chúng ta có thể tạo ra 1 cột mới bằng cách sau:

**Cách 1:** Trong chế độ **Data View** (hoặc **Report View**), chọn bảng muốn tạo thêm cột (giả sử chọn bảng **Products\_LK**, tạo cột **Margin**), sau đó chuyển qua Tab **Table Tools**

Trong nhóm **Calculations**, click nút **New column**



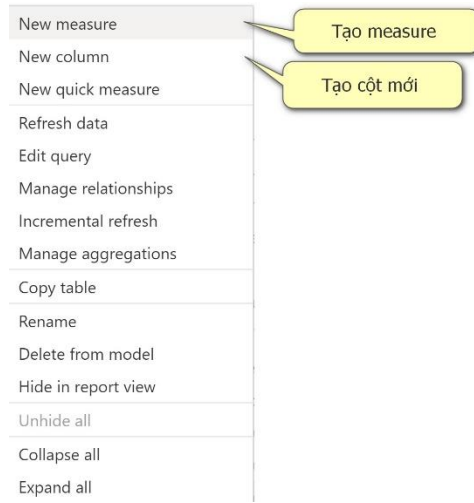
Xuất hiện dòng nhập công thức



Sau đó Enter

**Cách 2:** Click chuột phải vào tên bảng xuất hiện 1 menu

Chọn **New column**



## 2.2. Thêm Chỉ số (Measure)

Để thêm các Measure tương tự như cách tạo cột mới ở trên chỉ có khác là chọn **New Measure**.

Tuy nhiên công thức của Measure phải bắt đầu bằng 1 hàm DAX bởi vì nó không tham chiếu trực tiếp đến các cột.

Các Measure chỉ có thể được khi chúng ta Visualization (trực quan hoá dữ liệu bằng các biểu đồ trong chế độ Report View)

## 3. Các hàm DAX

### 3.1. Cú pháp và các toán tử DAX

#### 3.1.1. Cấu trúc của DAX

##### TÊN MEASURE

- **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

#### 3.1.2. Các toán tử sử dụng trong DAX



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

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"

Chú ý tới 02 biểu thức này

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" }

### 3.2. 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
<b>Ví dụ:</b> <ul style="list-style-type: none"> <li>SUM</li> <li>AVERAGE</li> <li>MAX/MIN</li> <li>DIVIDE</li> <li>COUNT/COUNTA</li> <li>COUNTROWS</li> <li>DISTINCTCOUNT</li> <li>SUMX</li> <li>AVERAGEX</li> <li>MAXX/MINX</li> <li>RANKX</li> <li>COUNTX</li> </ul>	<b>Ví dụ:</b> <ul style="list-style-type: none"> <li>IF</li> <li>IFERROR</li> <li>AND</li> <li>OR</li> <li>NOT</li> <li>SWITCH</li> <li>TRUE</li> <li>FALSE</li> </ul>	<b>Ví dụ:</b> <ul style="list-style-type: none"> <li>CONCATENATE</li> <li>FORMAT</li> <li>LEFT/MID/RIGHT</li> <li>UPPER/LOWER</li> <li>PROPER</li> <li>LEN</li> <li>SEARCH/FIND</li> <li>REPLACE</li> <li>REPT</li> <li>SUBSTITUTE</li> <li>TRIM</li> <li>UNICHAR</li> </ul>	<b>Ví dụ:</b> <ul style="list-style-type: none"> <li>CALCULATE</li> <li>FILTER</li> <li>ALL</li> <li>ALLEXCEPT</li> <li>RELATED</li> <li>RELATEDTABLE</li> <li>DISTINCT</li> <li>VALUES</li> <li>EARLIER/EARLIEST</li> <li>HASONEVALUE</li> <li>HASONEFILTER</li> <li>ISFILTERED</li> <li>USERRELATIONSHIP</li> </ul>	<b>Ví dụ:</b> <ul style="list-style-type: none"> <li>DATEDIFF</li> <li>YEARFRAC</li> <li>YEAR/MONTH/DAY</li> <li>HOURLY/MINUTE/SECOND</li> <li>TODAY/NOW</li> <li>WEEKDAY/WEEKNUM</li> </ul> <b>Time Intelligence:</b> <ul style="list-style-type: none"> <li>DATESYTD</li> <li>DATESQTD</li> <li>DATESMTD</li> <li>DATEADD</li> <li>DATESINPERIOD</li> </ul>

### 3.3. Hàm Date & Time

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

### 3.4. Hàm Logic

<b>IF()</b>	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	= <b>IF</b> (LogicalTest, ResultIfTrue, [ResultIfFalse])	
<b>IFERROR()</b>	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	= <b>IFERROR</b> (Value, ValueIfError)	
<b>AND()</b>	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	= <b>AND</b> (Logical1, Logical2)	Lưu ý: Sử dụng <b>&amp;&amp;</b> và <b>  </b> toán tử nếu bạn muốn bao gồm nhiều hơn hai điều kiện!
<b>OR()</b>	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	= <b>OR</b> (Logical1, Logical2)	

### 3.4. Hàm Text

<b>LEN()</b>	Trả về số lượng ký tự trong một chuỗi	= <b>LEN</b> (Text)	Lưu ý: Sử dụng toán tử <b>&amp;</b> như một phím tắt hoặc để kết hợp nhiều hơn hai chuỗi!
<b>CONCATENATE()</b>	Nối hai chuỗi văn bản thành một	= <b>CONCATENATE</b> (Text1, Text2)	
<b>LEFT/MID/RIGHT()</b>	Trả về một số ký tự từ đầu / giữa / cuối của chuỗi văn bản	= <b>LEFT/RIGHT</b> (Text, [NumChars]) = <b>MID</b> (Text, StartPosition, NumChars)	
<b>UPPER/LOWER/PROPER()</b>	Chuyển đổi các chữ cái trong một chuỗi thành trường hợp trên / dưới / thích hợp	= <b>UPPER/LOWER/PROPER</b> (Text)	
<b>SUBSTITUTE()</b>	Thay thế một thể hiện của văn bản hiện tại bằng văn bản mới trong một chuỗi	= <b>SUBSTITUTE</b> (Text, OldText, NewText, [InstanceNumber])	
<b>SEARCH()</b>	Trả về vị trí nơi một chuỗi được chỉ định hoặc ký tự được tìm thấy, từ trái sang phải	= <b>SEARCH</b> (FindText, WithinText, [StartPosition], [NotFoundValue])	

### 3.5. Hàm Number

<b>SUM()</b>	Trả về kết quả là một tổng	= <b>SUM</b> (ColumnName)	
<b>AVERAGE()</b>	Trả về trung bình (trung bình số học) của tất cả các số trong một cột	= <b>AVERAGE</b> (ColumnName)	
<b>MAX()</b>	Trả về giá trị lớn nhất trong một cột hoặc giữa hai biểu thức vô hướng	= <b>MAX</b> (ColumnName) or = <b>MAX</b> (Scalar1, [Scalar2])	
<b>MIN()</b>	Trả về giá trị nhỏ nhất trong một cột hoặc giữa hai biểu thức vô hướng	= <b>MIN</b> (ColumnName) or = <b>MIN</b> (Scalar1, [Scalar2])	
<b>DIVIDE()</b>	Thực hiện phép chia và trả về kết quả thay thế (hoặc để trống) nếu div/0	= <b>DIVIDE</b> (Numerator, Denominator, [AlternateResult])	

### 3.6. 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

## 3.7. Hàm Calculated

## 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 Calculated sẽ thực hiện tính toán lại một **Biểu thức (measure)** theo các điều kiện lọc, giả sử chúng ta đã tạo một Measure có tên là **Revenue (Doanh thu)** của tất cả các nhóm hàng bằng công thức dưới đây:

Revenue = SUMX(Sales, Sales[OrderQuantity]\*RELATED(Products\_LK[ProductPrice]))

Nếu muốn tính riêng doanh thu của nhóm hàng **Bikes**, chúng ta sẽ tạo ra một measure

Bikes Revenue = CALCULATE([Revenue], Product\_Categories\_LK[CategoryName]="Bikes")

## 3.8. 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]

Danh sách cột chi tiết muốn bỏ đi điều kiện lọc

Notes:

- Nếu tham số đầu tiên của bạn là một bảng, bạn có thể chỉ định các cột bổ sung
- Tất cả các cột phải bao gồm tên bảng và đến từ cùng một bảng

Examples:

- Customer\_Lookup[CustomerCity], Customer\_Lookup[CustomerCountry]
- Products[ProductName]



### 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 ALL trả về tất cả các dòng trong bảng hoặc tất cả các giá trị trong 1 cột, bỏ qua tất cả các bộ lọc đã sử dụng trước đó.

Thường sử dụng hàm ALL trong các phép tính mà muốn tính trên toàn bộ bảng hoặc toàn bộ cột mà không bị ảnh hưởng bởi các bộ lọc.

### Cách dùng

ALL( <bảng> )

ALL( <cột>, <cột>, ... )

Có thể viết theo 1 trong 2 cách trên.

### Tham số

<bảng> : bỏ lọc trên toàn bộ bảng

<cột> : khi bỏ lọc trên cột, có thể chỉ rõ bỏ lọc trên cột nào

### Giá trị trả về

Là bảng hoặc cột đã xóa bỏ các bộ lọc, bao gồm tất cả các giá trị có trong bảng, cột.

### Lưu ý

- Các giá trị dùng trong hàm ALL phải là các bảng hoặc cột được xác định cụ thể đã có sẵn, không phải là 1 biểu thức tính.
- Hàm ALL thường không dùng 1 mình mà kết hợp trong các hàm tính toán khác để tính trong trường hợp không muốn bị ảnh hưởng bởi bộ lọc.
- Khi áp dụng hàm ALL với nhiều cột sẽ bỏ qua các giá trị trùng lặp (Duplicates Value) trên các cột.

**Ví dụ:** Sau khi chúng ta đã tạo ra measure **Revenue** ở trên và sử dụng một matrix visual để biểu diễn Doanh thu theo từng nhóm hàng, do measure **Revenue** sử dụng Filter Context là **CategoryName** cho nên ta có kết quả như sau:

CategoryName	Revenue
Accessories	906,656.58
Bikes	23,642,500.06
Clothing	365,410.54
<b>Total</b>	<b>24,914,567.18</b>

Bây giờ chúng ta có nhu cầu muốn tạo thêm cột tổng doanh thu cố định trên từng dòng, muốn vậy chúng ta sẽ sử dụng hàm ALL để tạo measure này.

Total Revenue = CALCULATE([Revenue],ALL(Sales))

Trong measure **Total Revenue** đầu tiên hàm **ALL(Sales)** sẽ xóa tất cả các bộ lọc trên bảng Sales (không còn sử dụng Filter context), sau đó hàm **CALCULATE** sẽ tính lại doanh thu toàn bảng Sales, nếu chúng ta đưa thêm vào Matrix sẽ có kết quả như hình dưới:

CategoryName	Revenue	Total Revenue
Accessories	906,656.58	24,914,567.18
Bikes	23,642,500.06	24,914,567.18
Clothing	365,410.54	24,914,567.18
Components		24,914,567.18
<b>Total</b>	<b>24,914,567.18</b>	<b>24,914,567.18</b>

Bây giờ chúng ta có nhu cầu muốn tính tỉ lệ doanh thu từng nhóm hàng trên tổng doanh thu, muốn vậy chúng ta sẽ sử dụng hàm **ALL** để tạo measure này. Chúng ta sẽ tạo ra 1 measure **% Total Revenue**

% Total Revenue = DIVIDE([Revenue],CALCULATE([Revenue],ALL(Sales)))

Sau đó chúng ta thay thế vào Matrix trên có kết quả

CategoryName	Revenue	% Total Revenue
Accessories	906,656.58	0.04
Bikes	23,642,500.06	0.95
Clothing	365,410.54	0.01
<b>Total</b>	<b>24,914,567.18</b>	<b>1.00</b>

Muốn thể hiện thêm số liệu chi tiết các năm chúng ta kéo thêm cột **Year** vào Matrix có kết quả

Year	2015		2016		2017		Total	
CategoryName	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue
Clothing			156,151.21	0.01	209,259.33	0.01	<b>365,410.54</b>	<b>0.01</b>
Bikes	6,404,933.98	0.26	8,768,709.20	0.35	8,468,856.88	0.34	<b>23,642,500.06</b>	<b>0.95</b>
Accessories			399,334.94	0.02	507,321.64	0.02	<b>906,656.58</b>	<b>0.04</b>
<b>Total</b>	<b>6,404,933.98</b>	<b>0.26</b>	<b>9,324,195.35</b>	<b>0.37</b>	<b>9,185,437.85</b>	<b>0.37</b>	<b>24,914,567.18</b>	<b>1.00</b>

Với kết quả số liệu của bảng trên do chúng ta sử dụng **ALL(Sales)** nó bỏ qua tất cả các bộ lọc trên toàn bảng Sales cho nên **tỉ lệ Doanh thu / Tổng Doanh thu** của từng ô nó sẽ lấy Doanh thu từng ô chia cho Tổng doanh thu (24.914.567,18).

- Nếu muốn tính **tỉ lệ Doanh thu từng nhóm sản phẩm / trên Tổng Doanh thu của từng năm** chúng ta cần bỏ bộ lọc theo cột nhóm sản phẩm (CategoryName), vậy ta sử dụng lại công thức % Total Revenue như sau:

% Total Revenue = DIVIDE([Revenue],CALCULATE([Revenue],ALL(Product\_Categories\_LK[CategoryName])))

Có kết quả như hình

Year	2015		2016		2017		Total	
CategoryName	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue
Clothing			156,151.21	0.02	209,259.33	0.02	<b>365,410.54</b>	<b>0.01</b>
Bikes	6,404,933.98	1.00	8,768,709.20	0.94	8,468,856.88	0.92	<b>23,642,500.06</b>	<b>0.95</b>
Accessories			399,334.94	0.04	507,321.64	0.06	<b>906,656.58</b>	<b>0.04</b>
<b>Total</b>	<b>6,404,933.98</b>	<b>1.00</b>	<b>9,324,195.35</b>	<b>1.00</b>	<b>9,185,437.85</b>	<b>1.00</b>	<b>24,914,567.18</b>	<b>1.00</b>

- Nếu muốn tính tỉ lệ Doanh thu từng nhóm sản phẩm, theo từng năm / trên Tổng Doanh thu của tất cả các năm chúng ta cần bỏ bộ lọc theo cột năm (Year) , vậy ta sử dụng lại công thức % Total Revenue như sau:

% Total Revenue = `DIVIDE([Revenue],CALCULATE([Revenue],ALL(Calendar_LK[Year])))`

Có kết quả như hình

Year	2015		2016		2017		Total	
CategoryName	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue	Revenue	% Total Revenue
Clothing			156,151.21	0.43	209,259.33	0.57	<b>365,410.54</b>	<b>1.00</b>
Bikes	6,404,933.98	0.27	8,768,709.20	0.37	8,468,856.88	0.36	<b>23,642,500.06</b>	<b>1.00</b>
Accessories			399,334.94	0.44	507,321.64	0.56	<b>906,656.58</b>	<b>1.00</b>
<b>Total</b>	<b>6,404,933.98</b>	<b>0.26</b>	<b>9,324,195.35</b>	<b>0.37</b>	<b>9,185,437.85</b>	<b>0.37</b>	<b>24,914,567.18</b>	<b>1.00</b>

### 3.9. 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

Biểu thức lọc, theo các hàng của bảng

Examples:

- Territory\_Lookup
- Customer\_Lookup

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 FILTER giúp lọc bảng dữ liệu theo điều kiện.

#### Cách dùng

`FILTER(<Bảng>, <Biểu thức lọc>)`

#### Tham số

<Bảng> : bảng cần lọc dữ liệu. Bảng này có thể là 1 bảng có sẵn hoặc 1 biểu thức tính trả về kết quả dạng bảng

<Biểu thức lọc> : Chỉ rõ lọc trên cột nào, theo điều kiện nào

Dạng biểu thức lọc <Cột cần lọc = Điều kiện lọc>

#### Giá trị trả về

Là **bảng** chỉ còn lại những dòng chứa giá trị **thỏa mãn điều kiện lọc**

#### Lưu ý

- Bạn có thể dùng hàm FILTER để giảm bớt các đối tượng cần làm việc trong bảng, chỉ tính toán với những dữ liệu còn lại sau khi lọc.
- Hàm FILTER không dùng độc lập, riêng lẻ mà thường lồng ghép trong các hàm khác có sử dụng tham số dạng Bảng.

Ví dụ:

- Tạo ra bảng Sales 2015 chứa số liệu kinh doanh năm 2015. Thực hiện tạo 1 bảng mới bằng lệnh **New Table** và nhập công thức

Sales 2015 = FILTER(Sales,Year(Sales[OrderDate])=2015)

- Tạo ra bảng Sales 2017 chứa số liệu kinh doanh năm 2017 trở về sau, ngoài cách làm như trên (thay 2015 bằng 2017) chúng ta cũng có thể dùng công thức sau:

Sales 2015 = FILTER(Sales,DATEDIFF(DATE(2016,12,31),Sales[OrderDate],DAY)>0)

**Giải thích:** điều kiện lọc ở đây chúng ta sử dụng hàm DATEIFF (hàm tính số chênh lệch giữa 2 ngày 31/12/2016 và ngày đặt hàng, nếu > 0 thì đưa vào bảng, như vậy những đơn hàng có ngày đặt hàng sau 31/12/2016 sẽ được đưa vào bảng kết quả)

- Tạo ra bảng Sales Red chứa các Đơn hàng có màu sản phẩm là màu đỏ

Sales Red = FILTER(Sales,RELATED(Products\_LK[ProductColor])="Red")

- Tạo 1 measure tính Doanh thu các nhóm hàng năm 2017

Revenue 2017 = CALCULATE([Revenue],FILTER(Sales,YEAR(Sales[OrderDate])=2017))

Có kết quả

CategoryName	Revenue	Revenue 2017
Accessories	906,656.58	507,321.64
Bikes	23,642,500.06	8,468,856.88
Clothing	365,410.54	209,259.33
<b>Total</b>	<b>24,914,567.18</b>	<b>9,185,437.85</b>

### 3.10. Nhóm 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

Bảng mà trong đó công thức sẽ chạy

Examples:

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

Biểu thức

Examples:

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



**TIP:**

Tương tự như 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)

Đặc biệt Hàm SUM và SUMX được sử dụng rất nhiều chúng ta có 1 số lưu ý về cách dùng như sau:

Để sử dụng hàm SUM, chúng ta cần phải can thiệp trực tiếp vào bảng dữ liệu bằng cách tạo thêm cột mới. Việc này không khó để làm, nhưng nó tốn thời gian và tốn tài nguyên. Excel sẽ phải luôn tính toán lại kết quả trong cột mà chúng ta mới thêm vào, đồng thời nó làm thay đổi độ lớn của bảng dữ liệu. Nhưng với SUMX thì không. Việc tính toán chỉ thực hiện khi chúng ta sử dụng hàm này mà thôi, và nó cũng không làm thay đổi bảng dữ liệu.

**Lưu ý:**

1. Nếu bảng dữ liệu đã có sẵn cột mà chúng ta cần tính rồi thì dùng hàm SUM sẽ viết công thức nhanh hơn, đơn giản hơn.

2. Nếu bảng dữ liệu chưa có sẵn cột cần tính thì chúng ta dùng hàm SUMX sẽ hiệu quả hơn thay vì cố gắng tạo ra cột cần tính chỉ để dùng hàm SUM.

3. Luôn cân nhắc việc tạo ra thêm cột mới trong bảng dữ liệu, bởi nó làm file nặng hơn, tính chậm hơn trong khi tác dụng có thể không nhiều (bởi chúng ta hoàn toàn tính toán được mà không cần tạo ra cột đó với các hàm X)

### 3.11. Các hàm Time Intelligence

**Time Intelligence** – Thời gian thông minh là một chức năng của PowerBI. Time Intelligence có 2 phần:

- Time – Thời gian: Chúng ta sẽ thường bắt gặp những dữ liệu là “time” trong bộ dữ liệu của mình. Đó có thể là ngày sinh của khách hàng, ngày đặt hàng, ngày giao hàng hoặc ngày nhập kho...

- Intelligence – Thông minh: nó cho phép chúng ta thao tác dữ liệu bằng các khoảng thời gian như năm, quý, tháng, ngày và tạo các phép tính trong các khoảng thời gian đó.

Các hàm này thường dùng để so sánh dữ liệu: cùng kỳ tháng trước, quý trước, năm trước, ...

#### Hàm DATEADD

*Cú pháp:* DATEADD(Dates, NumberOfIntervals, Interval)

**Dates** (Thời điểm làm mốc) : Thường là 1 cột chứa giá trị thời gian để làm mốc

**NumberOfIntervals** (Khoảng cách thời gian so với mốc) : Là số nguyên. Nếu nhập số >0 sẽ cho ra thời gian tăng thêm (sau thời gian mốc), số <0 sẽ cho thời gian giảm đi (trước thời gian mốc) một khoảng tương ứng theo loại đơn vị thời gian được chỉ định.

**Interval** (Loại đơn vị thời gian) : Là các ký tự đại diện cho đơn vị thời gian (nhập bằng tiếng anh): year (năm), quarter (quý), month (tháng), day (ngày).

*Giá trị trả về:* Là cột chứa các giá trị thời gian (đủ ngày, tháng, năm), Giá trị thời gian mới sẽ cách thời điểm mốc bằng số ngày (hoặc số tháng, số quý, số năm) tùy theo giá trị được chỉ định trong hàm.

Thông thường chúng ta sẽ kết hợp với CALCULATE để tính toán theo mẫu:

CALCULATE(Measure, DATEADD(Dates, NumberOfIntervals, Interval))

Chúng ta cũng có thể thay thế DATEADD bằng các hàm:

- PREVIOUSMONTH(Dates)
- PREVIOUSYEAR(Dates)
- PREVIOUSDAY(Dates)
- PREVIOUSQUARTER (Dates)

#### Hàm DATESYTD

*Cú pháp:* DATESYTD(Dates)

**Dates** (Thời điểm làm mốc) : Thường là 1 cột chứa giá trị thời gian để làm mốc

*Giá trị trả về:* Trả về dãy các ngày từ ngày đầu năm đến ngày làm mốc

#### Lưu ý:

Hàm không dùng với mục đích trả về giá trị kết quả trực tiếp, mà thường dùng để tính toán với điều kiện tính toán theo từng năm. Hàm thường dùng để lồng bên trong hàm CALCULATE với vai trò là điều kiện tính toán lại.

*Ví dụ:*

1. Tính doanh thu lũy kế với chu kỳ lũy kế theo năm (Lũy kế theo từng ngày từ đầu năm đến cuối năm)

Lũy kế theo năm 2016 = CALCULATE([DT 2016], DATESYTD(Calendar\_LK[Date].[Date]))

2. Tính doanh thu lũy kế với chu kỳ lũy kế theo tháng (Lũy kế theo từng ngày từ đầu tháng đến cuối tháng)

Lũy kế theo tháng 2016 = CALCULATE([DT 2016], DATESMTD(Calendar\_LK[Date].[Date]))



## Hàm SAMEPERIODLASTYEAR

Cú pháp: SAMEPERIODLASTYEAR (Dates)

Dates (Thời điểm làm mốc) : Thường là 1 cột chứa giá trị thời gian để làm mốc

Giá trị trả về: Cột chứa giá trị ngày tương ứng của năm trước (cùng ngày, cùng tháng nhưng lùi về 1 năm) so với ngày trong Dates. Hay còn được hiểu là cùng kỳ so với năm trước.

### Lưu ý:

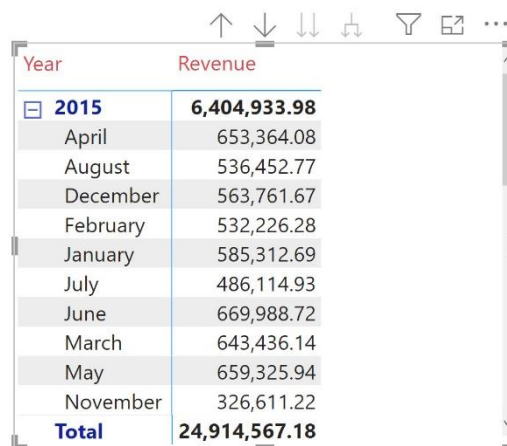
Hàm không dùng với mục đích trả về giá trị kết quả trực tiếp, mà thường dùng để tính toán với điều kiện tính toán theo từng năm. Hàm thường dùng để lồng bên trong hàm CALCULATE với vai trò là điều kiện tính toán lại.

Ví dụ:

Tạo Matrix có thông tin Doanh thu (Revenue), theo năm và tháng

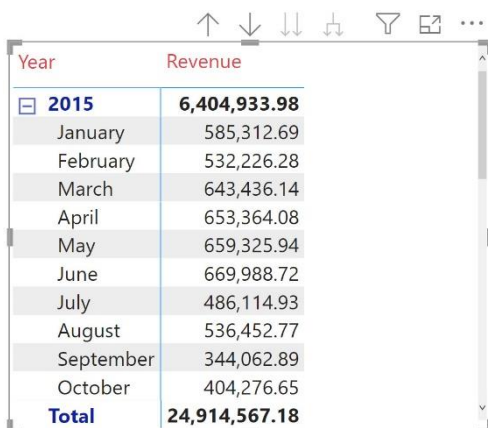
1. Tạo Measure **Revenue**
2. Tạo Matrix Visual, kéo cột **Revenue** vào *Values*
3. Sau đó kéo 2 cột **Year** và **Month name** vào *Rows*

Kết quả như hình dưới:



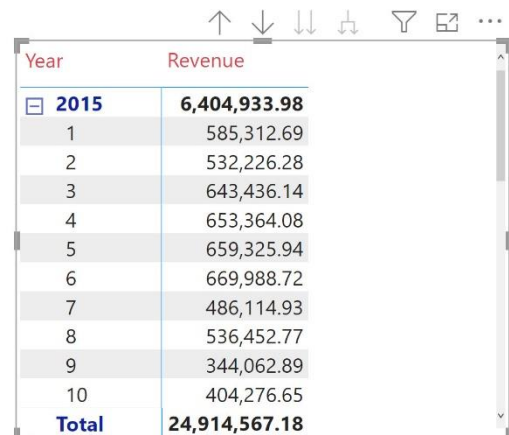
Year	Revenue
2015	6,404,933.98
April	653,364.08
August	536,452.77
December	563,761.67
February	532,226.28
January	585,312.69
July	486,114.93
June	669,988.72
March	643,436.14
May	659,325.94
November	326,611.22
<b>Total</b>	<b>24,914,567.18</b>

4. Chúng ta cần sắp xếp lại cột tên tháng tăng dần từ tháng 1 đến tháng 12, bằng cách quay về chế độ **Data View**, sau đó chọn cột **Month Name**, trên Tab **Columns Tool**, click vào nút **Sort by Column** và chọn cột **Month**, kết quả ta có trong *Figure 1* hoặc nếu kéo cột Month có kết quả tương tự trong *Figure 2*



Year	Revenue
2015	6,404,933.98
January	585,312.69
February	532,226.28
March	643,436.14
April	653,364.08
May	659,325.94
June	669,988.72
July	486,114.93
August	536,452.77
September	344,062.89
October	404,276.65
<b>Total</b>	<b>24,914,567.18</b>

Figure 2



Year	Revenue
2015	6,404,933.98
1	585,312.69
2	532,226.28
3	643,436.14
4	653,364.08
5	659,325.94
6	669,988.72
7	486,114.93
8	536,452.77
9	344,062.89
10	404,276.65
<b>Total</b>	<b>24,914,567.18</b>

Figure 1

5. Tạo thêm 1 Measure lấy dữ liệu doanh thu cùng kỳ năm trước đặt tên **Revenue Same Period Last Year**

Revenue Same Period Last Year = CALCULATE([Revenue], SAMEPERIODLASTYEAR(Calendar\_LK[Date]))

Year	Revenue	Revenue Same Period Last Year
May	659,325.94	
June	669,988.72	
July	486,114.93	
August	536,452.77	
September	344,062.89	
October	404,276.65	
November	326,611.22	
December	563,761.67	
<b>2016</b>	<b>9,324,195.35</b>	<b>6,404,933.98</b>
January	432,425.86	585,312.69
February	474,162.94	532,226.28
March	471,962.02	643,436.14
April	494,957.58	653,364.08
May	545,535.02	659,325.94
June	533,825.21	669,988.72
...	...	...
<b>Total</b>	<b>24,914,567.18</b>	<b>9,357,802.61</b>

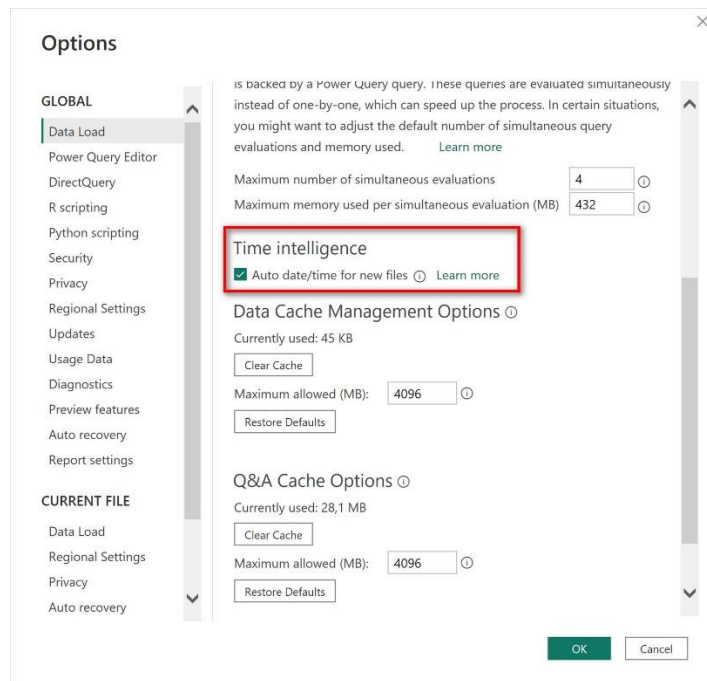
## 3.12. Date Dimension

### 3.12.1. Giới thiệu về Default Date Dimension

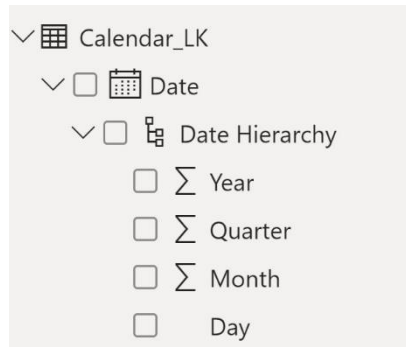
Trong Power BI, Dimension (Kích thước) là một khái niệm quan trọng trong việc xây dựng mô hình dữ liệu và tạo báo cáo. Kích thước là một thuộc tính hoặc mô tả các khía cạnh của dữ liệu mà ta muốn phân loại, nhóm hoặc lọc dữ liệu theo đó.

Thông thường trong PBI có rất nhiều bảng chứa các cột dữ liệu dạng Date, để thuận tiện chúng ta nên tạo 1 Date Dimension, trong đó có 1 cột dữ liệu dạng Date chứa tất cả các ngày trong các báo cáo sẽ lập sau này. Trong Power BI khi chúng ta tạo các cột dạng Date thì nó mặc định tạo cho chúng ta 1 Date Dimension gọi là Default Date Dimension được định sẵn tự động trong Power BI Desktop và chúng ta có thể tùy chỉnh option này ở Data Load cho Global (tất cả các báo cáo Desktop) hoặc Current File (cho báo cáo này thôi). Lựa chọn này được đặt sẵn on và khi có nó, các trường dữ liệu dạng Date của chúng ta sẽ được đi kèm theo một bảng **Date Hierarchy** tạo sẵn chính là Default Date Dimension.

Chọn Tab **File** / chọn **Options and Settings** / chọn **Options** / trong **GLOBAL** / **Data Load** chúng ta sẽ thấy tùy chọn **Time intelligence**



Các cột kiểu Data thông thường với tùy chọn này sẽ có bảng **Date Hierarchy** tạo sẵn gọi là Default Date Dimension.



**Lưu ý:** Khi chúng ta thiết lập Relationship cho 1 cột kiểu Date từ 1 Lookup Table đến 1 cột kiểu Date của 1 Data table thì ký hiệu dạng Date (hình cuốn lịch) ở Data table sẽ không xuất hiện (không có Date Hierarchy)

### 3.12.2. Custom Date Dimension

Khi bật chế độ Default Date Dimension, Power BI tự động tạo ra những bảng ẩn với mọi trường định dạng Date và xuất hiện hierarchy gồm Year/Quarter/Month/Day. Kể cả khi chúng ta tạo một trường mới từ các trường khác và định dạng date thì PowerBI sẽ tự động tạo một date hierarchy dimension ở các trường đó và đây thực chất là một bảng ẩn dựa trên trường Date đó.

Default Date Dimension không thể tùy chỉnh được, do vậy nếu cần thiết chúng ta có thể tạo ra Custom Date Dimension. Có nhiều cách tạo ra Custom Date Dimension

Cách 1:

- Trong PBI, đầu tiên chúng ta tạo 1 bảng Date, bằng cách mở chế độ Data View, chọn New Table, trên thanh công thức nhập

```
Calendar_LK = CALENDAR(DATE(2021,01,01), DATE(2023,12,31))
```

Với Calendar\_LK là tên bảng, và dùng hàm CALENDAR(StartDate, EndDate) để tạo ra bảng này, sẽ có 1 cột Date kèm theo Date Hierarchy của nó

- Sau đó chúng ta lần lượt tạo ra các cột, bằng cách chọn lệnh New Column và nhập các công thức sau

```
Day = FORMAT('Calendar_LK'[Date], "DD")
Month = FORMAT('Calendar_LK'[Date], "MMM YYYY")
Year = FORMAT('Calendar_LK'[Date], "YYYY")
Quarter = FORMAT('Calendar_LK'[Date], "\Quý Q")
Week = FORMAT('Calendar_LK'[Date], "\Week WW")
```

Cách 2:

- Trong PBI, chúng ta vẫn tạo bảng Date Dimension bằng hàm CALENDAR(StartDate, EndDate) như sau:

Giả sử chúng ta đã có bảng Sales\_DT, có cột OrderDate (Ngày đặt hàng) chứa dữ liệu dạng ngày của các năm 2015, 2016, 2017. Vậy để tạo bảng Calendar\_LK có cột Date (mặc định) chúng ta chỉ cần tạo 1 bảng mới với công thức:

```
Calendar_LK = CALENDAR(MIN(Sale_DT[OrderDate]), MAX(Sale_DT[OrderDate]))
```

Với công thức này sẽ phát sinh ra cột Date trong bảng Calendar\_LK chứa tất cả các ngày trong 3 năm trên, mà cận trên là ngày nhỏ nhất và cận dưới là ngày lớn nhất của cột Sale\_DT[OrderDate]

Tuy nhiên trong các bảng Date Dimension thì các ngày phải đầy đủ cho các năm (365 ngày), liên tục và không được ngắt quãng, cho nên công thức chuẩn nhất sẽ là:

```
Calendar_LK = CALENDAR(DATE(YEAR(MIN(Sale_DT[OrderDate])), 1, 1),
DATE(YEAR(MAX(Sale_DT[OrderDate])), 12, 31))
```

Trong trường hợp chúng ta muốn tạo thêm các cột mô tả ngày, tháng, quý, năm, .. chúng ta dùng như cách 1 bằng cách sử dụng New Column hoặc chúng ta dùng hàm ADDCOLUMN có cú pháp:

```
ADDCOLUMNS(Table, Name1, Expression1, ...)
```

Trong đó:

- **Table:** Tên bảng (trong ví dụ này là Calendar\_LK, tuy nhiên chúng ta sẽ dùng ghép nên sẽ dùng hàm CALENDAR đã tạo phía trên)

- **Name1:** Tên cột 1

- **Expression1:** Biểu thức tạo cột 1

để bổ sung vào công thức như sau:

```
Calendar_LK = ADDCOLUMNS(CALENDAR(DATE(YEAR(MIN(Sale_DT[OrderDate])),1,1),
    DATE(YEAR(MAX(Sale_DT[OrderDate])),12,31)),
    "Year", YEAR([Date]),
    "Month", FORMAT([Date], "mmm"),
    "Month No", MONTH([Date]),
    "Quater", FORMAT([Date], "\Quater Q"),
    "Week", FORMAT([Date], "\Week W")
))
```

### 3.12.3. Cách sử dụng Default Date Dimension với DAX Time Intelligence

Khi sử dụng một trường Date trong DAX với các công thức Time Intelligence, bạn sẽ có hai cách để sử dụng trường Date đó trong bảng, đó là chọn sử dụng trường Date đó như chính giá trị của nó hoặc chọn show Date Hierarchy tức là sử dụng Default Date Dimension.

- Trong 1 visual (Giả sử Matrix Visual) nếu chúng ta sử dụng các hierarchy **Year/Quarter/Month/Day** sẵn có của nó đưa vào các rows thì chúng ta chú ý phải bổ sung thêm `[Date]` ở đằng sau cột dữ liệu dạng Date (Mục đích để nó hiểu rõ dữ liệu dạng ngày)

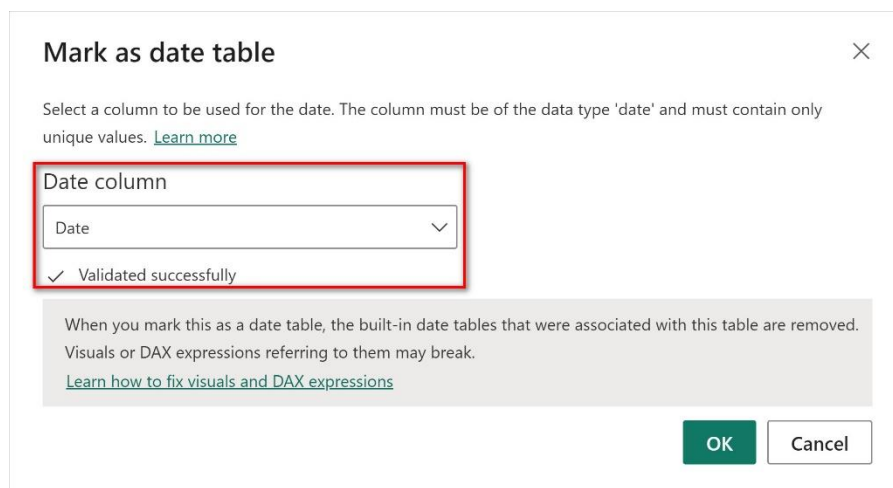
```
Prev Month Revenue = CALCULATE([Revenue],DATEADD(Calendar_LK[Date].[Date],-1,MONTH))
```

- Trường hợp sử dụng các **Custom Date Dimension** đã tạo như trên đưa vào các rows thì không cần bổ sung `[Date]`

```
Prev Month Revenue = CALCULATE([Revenue],DATEADD(Calendar_LK[Date],-1,MONTH))
```

**Lưu ý:** Để khỏi phải dùng `[Date]` đằng sau cột dữ liệu dạng Date trong cả 2 trường hợp trên chúng ta thực hiện như sau.

- Chọn bảng *Date Dimension* (trong ví dụ này chúng ta click chọn bảng *Calendar\_LK*), sau đó vào *Tab Tools*, chọn *Make as Date Table* xuất hiện hộp thoại. Trong *Date column* hãy chọn cột *Date* và *OK*



Lúc này biểu tượng cột Date sẽ thay đổi như hình:

**Sale\_DT**

- CustomerKey
- OrderDate**
- OrderLineItem
- OrderNumber
- OrderQuantity
- ProductKey
- StockDate
- TerritoryKey
- Collapse ^



**Calendar\_LK**

- Date**
- Month
- Month No
- Quarter
- Week
- Collapse ^

**Properties**

**Data**

Search

- Calendar\_LK
  - Date**
  - Month
  - Month No
  - Quarter
  - Week
  - Year
- Sale\_DT

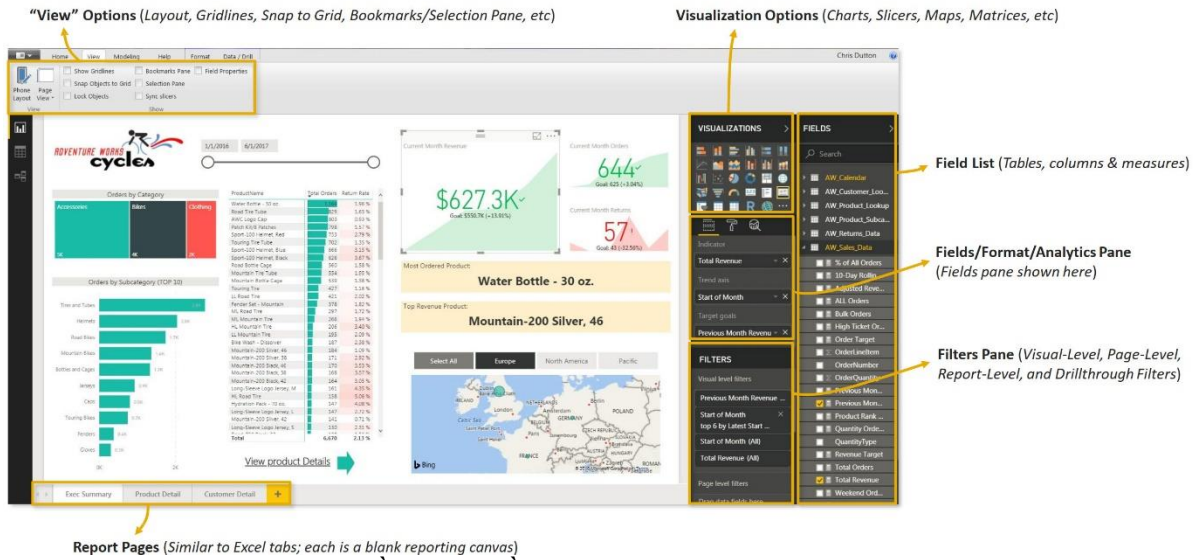
# BÀI 7: TRỰC QUAN HOÁ BÁO CÁO BẰNG POWER BI

## 1. Trực quan hoá báo cáo kinh doanh bằng Power BI

### 1.1. Report View và cách thêm đối tượng cho report

Sau khi chúng ta đã thực hiện tất cả các bước kể trên, giai đoạn cuối cùng trong Power BI là trực quan hoá (Visualization) các dữ liệu thành các loại biểu đồ.

Phần đầu tiên chúng ta cần nghiên cứu là giao diện Report View



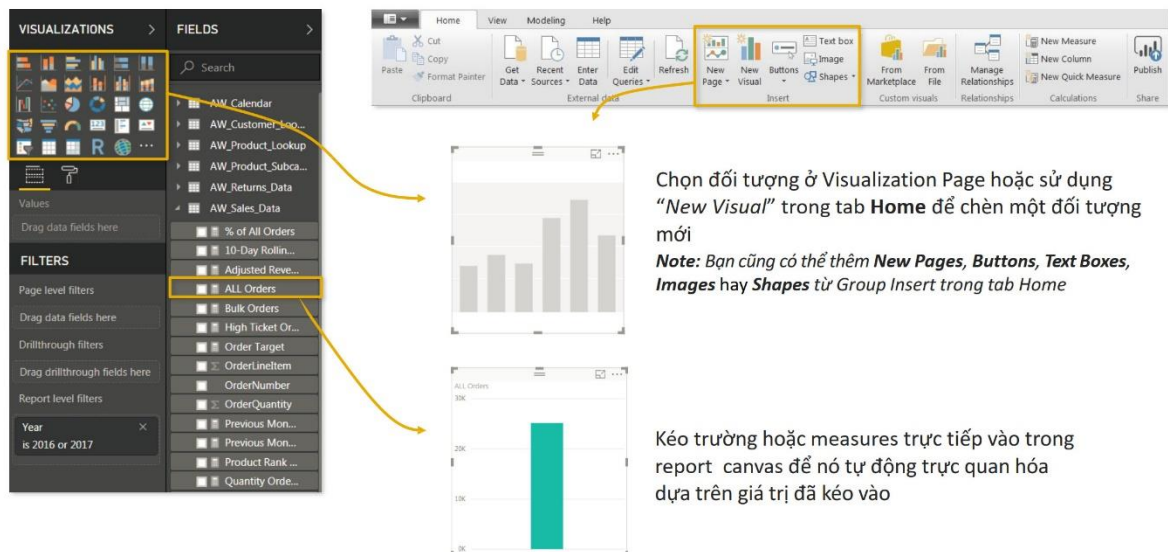
Giao diện Report View bao gồm các phần chính như sau:

- Phần Dashboard là phần lớn nhất của Report nó chứa tất cả các đồ thị mô tả, chúng ta có thể chia các báo cáo ra nhiều phần chứa trên các Page (Report Pages)

Bên phải của Dashboard chứa các pane:

- Visualizations Pane: Phần này chứa các Visual cho phép chúng ta tạo ra các loại đồ thị, slicer bằng các thao tác kéo thả. Mỗi khi chúng ta tạo ra một đối tượng visual thì bên dưới sẽ có phần tùy chỉnh nội dung của Visual đó ( các trường chọn làm nội dung, định dạng các visual)
- Fields Pane: Chứa tất cả các bảng mà chúng ta đã đưa vào Data Model
- Filter Pane: Tạo ra các bộ lọc

### 1.2. Chèn Objects & Charts cơ bản



Có 2 cách để chèn các đối tượng vào báo cáo:

**Cách 1:** Chọn trực tiếp các đối tượng từ Visualizations Pane và kéo vào báo cáo

Cách 2: Chọn Tab Home / trong nhóm Insert / chọn New Visual

Cách 3: Chọn Tab Insert / trong nhóm elements / chọn đối tượng muốn chèn

## 2. Tạo và định dạng các loại Visual

### 2.1. Sử dụng Card Visual

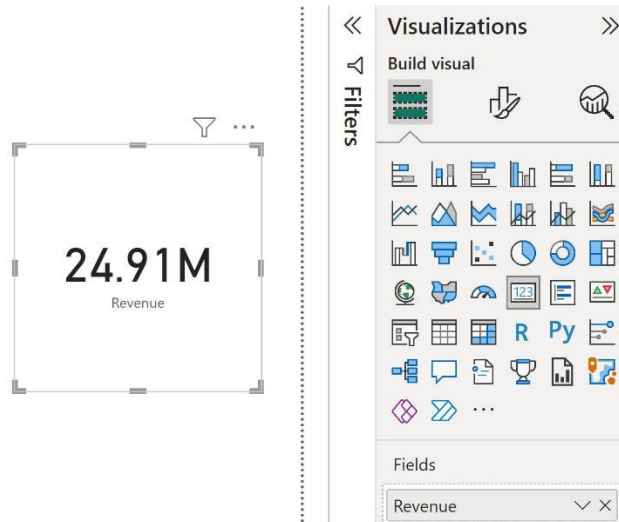
Trong Power BI có 3 loại Card Visual:

#### 1. Single Card: Loại này chỉ hiển thị 1 giá trị

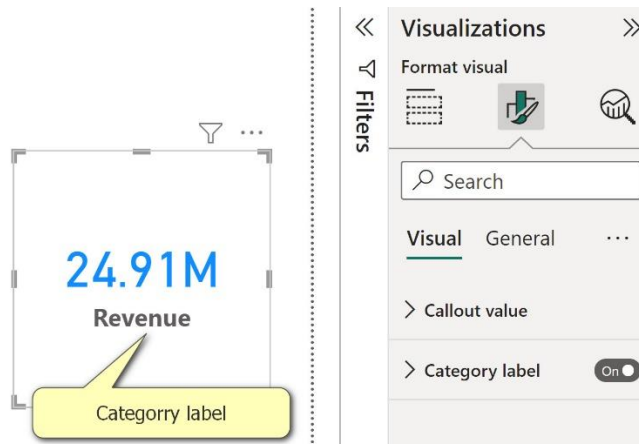
Ví dụ: Sử dụng Single Card để hiển thị tổng doanh thu (**Revenue**)

- Click vào nút **Card** để tạo ra 1 Single Card

- Đưa dữ liệu muốn hiển thị vào vùng **Fields** (ở đây chúng ta kéo measure **Revenue** vào) sẽ có kết quả như hình.



- Để định dạng Card, chúng ta click sang Tab **Format your Visual**

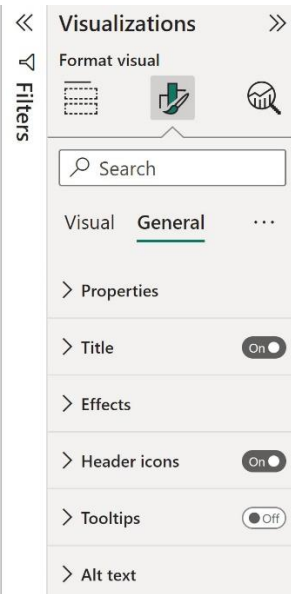


+ Trong Tab **Visual** có 2 nội dung định dạng:

\* **Callout value** dùng để định dạng con số hiển thị (đây là số 24.91 M tổng doanh thu), chúng ta có thể định dạng về font size, font color, ...

\* **Category label** dùng để định dạng nhãn của nhóm, hoặc bật tắt nhãn

+ Trong Tab **General** dùng để định dạng chung cho Card (Màu nền, đường viền, tiêu đề)



## 2. Multi row Card

Loại này chỉ khác nó thể hiện được nhiều giá trị cùng 1 lúc, chẳng hạn Doanh thu, Chi phí, Lợi nhuận

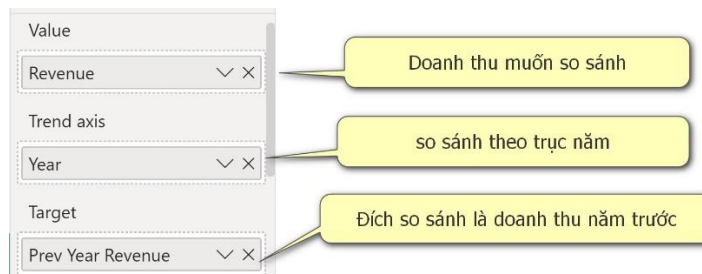
Tổng hợp Doanh thu, Chi phí, Lợi nhuận		
24,914,567.18	14,456,986.32	10,457,580.86
<a href="#">Revenue</a>	<a href="#">Total Cost</a>	<a href="#">Total Profit</a>

## 3. KPI Card

KPI Card cũng tương tự như Single Card, chỉ có 1 điểm khác biệt dữ liệu trong KPI Card bao gồm 2 loại: Dữ liệu muốn so sánh đồ KPI và dữ liệu đích để so sánh

*Ví dụ: Muốn so sánh Doanh thu của 1 năm với doanh thu của năm trước*

Chúng ta sẽ đưa các chỉ tiêu vào theo nội dung sau



Kết quả sẽ có 1 KPI Card

Bảng so sánh doanh thu năm trước		
Year	Revenue	Prev Year Revenue
2015	6,404,933.98	
2016	9,324,195.35	6,404,933.98
2017	9,185,437.85	2,952,868.63
<b>Total</b>	<b>24,914,567.18</b>	<b>9,357,802.61</b>



## 2.2. Sử dụng Map Visual

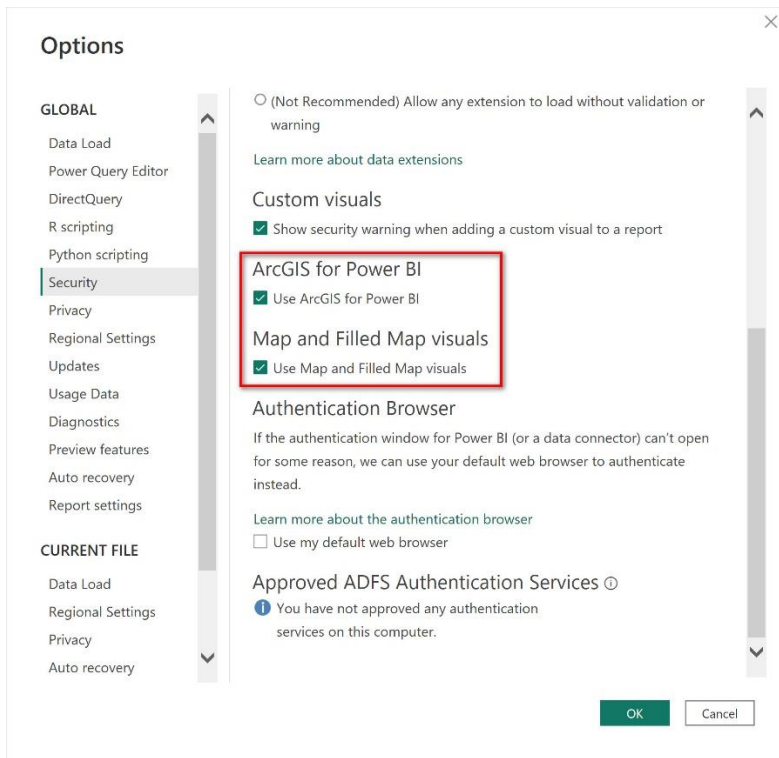
Map Visual thông thường sẽ có 3 dạng:

1. Map (là dạng cơ bản)
2. Filled Map
3. ArcGIS Map

3 dạng này đều sử dụng Bing Map để hiển thị vị trí địa lý cùng với Data muốn hiển thị.

Muốn sử dụng nó, đầu tiên chúng ta phải vào **File** / chọn **Options and Settings** / **Options** / trong **GLOBAL** / chọn **Security**



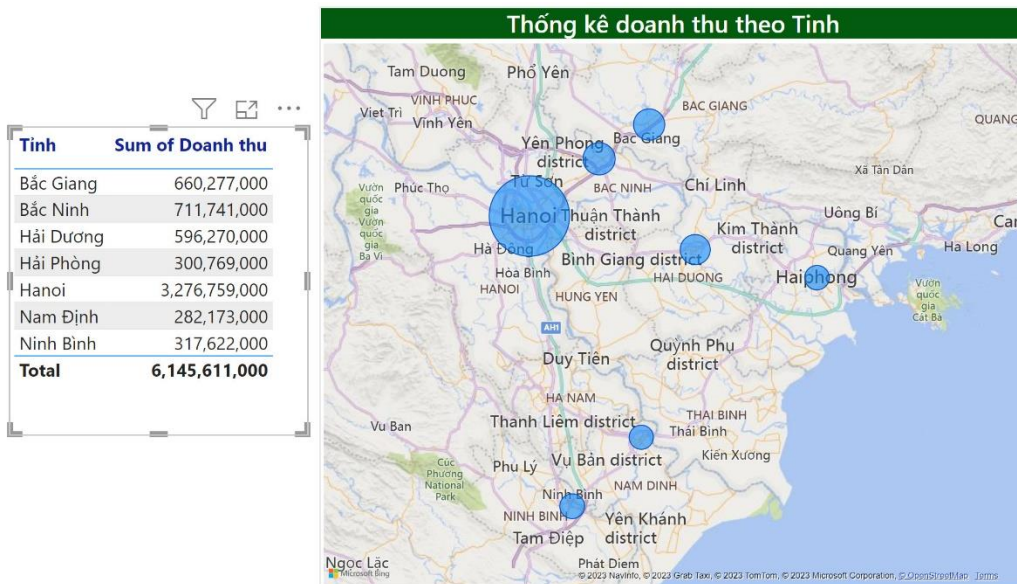


Ví dụ 1: Chúng ta có file *Map-data.xlsx* chứa dữ liệu doanh thu của một số cửa hàng, hãy nạp vào Power BI và dùng *Map Visual* để mô tả Doanh thu theo từng Tỉnh, thực hiện như sau:

- Click chọn **Map** visual và nạp vào các thông tin sau:

1. Phần **Location** kéo trường **Tỉnh** vào
2. Phần **Bubble size** kéo trường **Doanh thu** vào (tùy chọn Sum of Doanh thu)
3. Nếu cần chi tiết theo các cấp nhỏ hơn thì kéo phần **Quận, Cửa hàng** vào phần **Legend**

- Kết quả



**Lưu ý:**

- Đối với dữ liệu của Việt Nam chúng ta cần kiểm tra lại trên bảng đồ đôi khi một số vị trí hiển thị không đúng, do Bing Map sử dụng các vị trí nó đã thiết lập hầu hết là bằng chữ không dấu, nên chúng ta cần cập nhật lại dữ liệu vị trí trên cơ sở dữ liệu của chúng ta theo các vị trí mà Bing đã nhận dạng, thì nó mới hiển thị chính xác được.

- Để update chúng ta mở Bing map và copy các địa chỉ có trong CSDL và dán vào ô tìm kiếm, sau khi ra kết quả chúng ta lại copy nó lại CSDL

- Các vị trí (Tỉnh, thành phố Việt Nam) trong cơ sở dữ liệu nên dùng chữ không dấu.

- Có thể sử dụng kinh độ (**Longitude**) và vĩ độ (**Latitude**) của vị trí để nhập vào chính xác

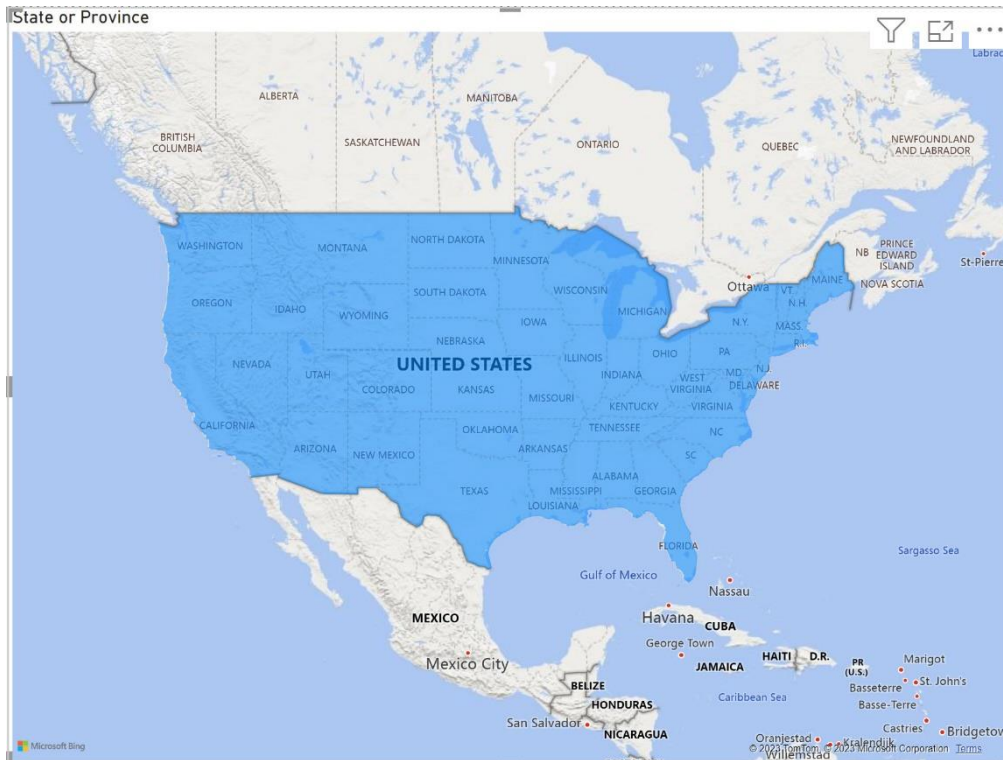
*Ví dụ 2: Sử dụng file P6-SuperStoreUS-2015.xlsx để mô tả*

- Nạp file P6-SuperStoreUS-2015 vào Power BI

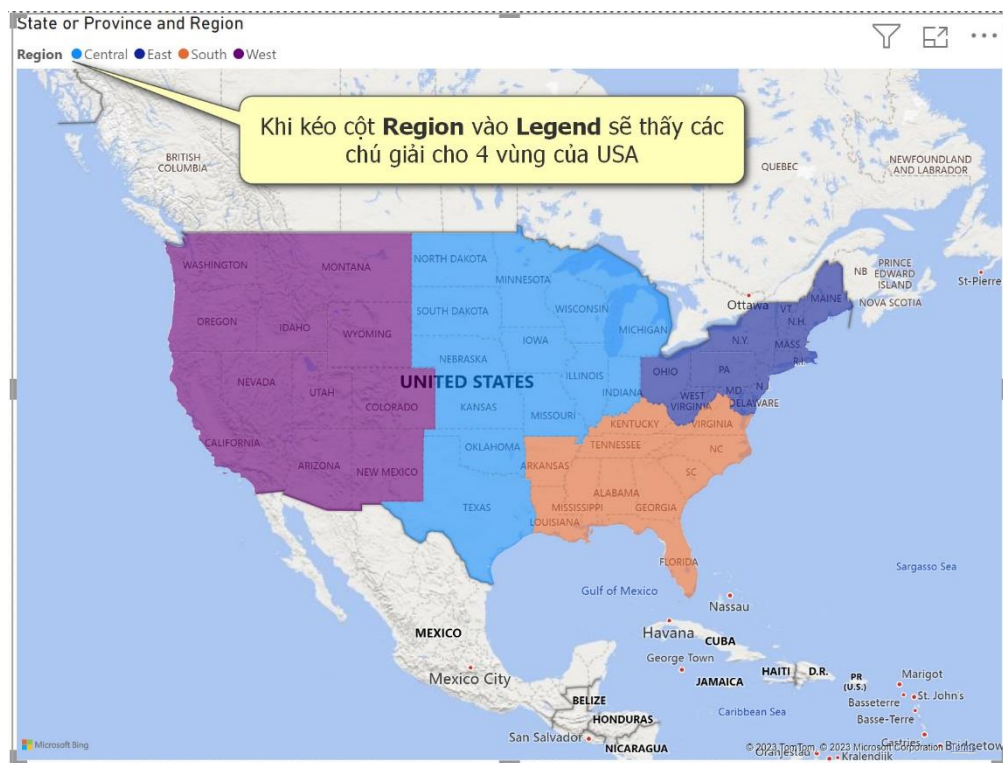
- Chọn tất cả các Sheet có trên file và Load vào Data Model

- Click chọn **Filled Map**

- Kéo trường **State or Province** vào mục **Location**, sẽ có kết quả trong hình dưới do nó nhận dạng được tất cả các bang của Mỹ có trong trường này, ban đầu nó chỉ có đồng nhất một màu xanh nước biển.



- Nếu chúng ta muốn thể hiện các khu vực trong bản đồ này thì kéo tiếp trường **Region** vào mục **Legend** sẽ thấy các khu vực



- Kéo trường **Sales** vào **Tooltips** để thể hiện chi tiết doanh thu theo từng bang

### 2.3. Sử dụng TreeMap Visual

TreeMap Visual là một dạng đồ thị biểu diễn các giá trị theo các tiêu chí khác nhau dưới hiển thị là các hình chữ nhật có màu khác nhau.

*Ví dụ: Chúng ta có file Map-data.xlsx chứa dữ liệu doanh thu của một số cửa hàng, hãy nạp vào Power BI và dùng Map Visual để mô tả Doanh thu theo từng Nhân viên, thực hiện như sau:*

- Click chọn **TreeMap** visual và nạp vào các thông tin sau:

1. Trong mục **Category** chúng ta kéo vào một trường muốn phân loại cấp 1 ( nhập trường **Nhân viên Sale**)

2. Trong mục **Details** chúng ta kéo vào một trường muốn phân loại cấp 2 ( giả sử chúng ta muốn hiển thị Doanh thu theo từng nhân viên và theo từng sản phẩm thì chúng ta kéo thêm trường **Sản phẩm** vào)

3. Trong mục **Values** chúng ta kéo vào các trường dạng số vào để tổng hợp ( trường hợp ví dụ này ta kéo trường **Doanh thu** vào)

4. Trong mục **Tooltips** chúng ta kéo vào trường để bổ trợ thêm thông tin cần tổng hợp (ví dụ muốn đếm thêm số lượng sản phẩm đã bán của từng Nhân viên theo từng sản phẩm thì kéo thêm trường **Sản phẩm** vào)

Kết quả ta có đồ thị mô tả như sau:



### 2.4. Sử dụng Gauge Visual

Biểu đồ Gauge thường dùng để biểu diễn 1 giá trị cần so sánh với 1 giá trị đích.

*Ví dụ: Chúng ta vẫn sử dụng dữ liệu nguồn là file Map-data.xlsx, nạp vào Power BI, sau đó tạo thêm bảng Calendar\_LK, với cú pháp:*

```
Calendar_LK = ADDCOLUMNS(CALENDAR(
    DATE(YEAR(MIN(Data_Sale[Ngày mua hàng])),1,1),
    DATE(YEAR(MAX(Data_Sale[Ngày mua hàng])),12,31)),
    "Year", YEAR([Date]),
    "Month", FORMAT([Date], "mmm"),
    "Month No", MONTH([Date]),
    "Quater", FORMAT([Date], "\Quater Q"),
    "Week", FORMAT([Date], "\Week WW")
))
```

- Lập Relationship cho 2 bảng Calendar\_LK và Data\_Sale.

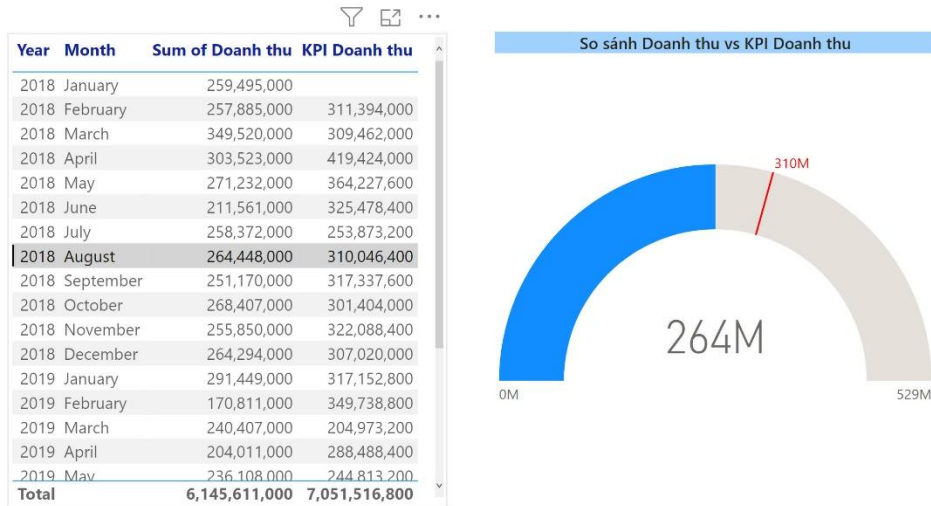
- Thiết lập cột Month sắp xếp (Sort by column) theo Month No

- Giả sử chúng ta muốn so sánh Doanh thu từng tháng đối chiếu với đích muốn đạt được là KPI Doanh thu = Doanh thu tháng trước x 1,2. Vậy chúng ta sẽ tạo 1 measure như sau:

KPI Doanh thu = `CALCULATE(SUM(Data_Sale[Doanh thu]),DATEADD(Calendar_LK[Date],-1,MONTH))*1.2`

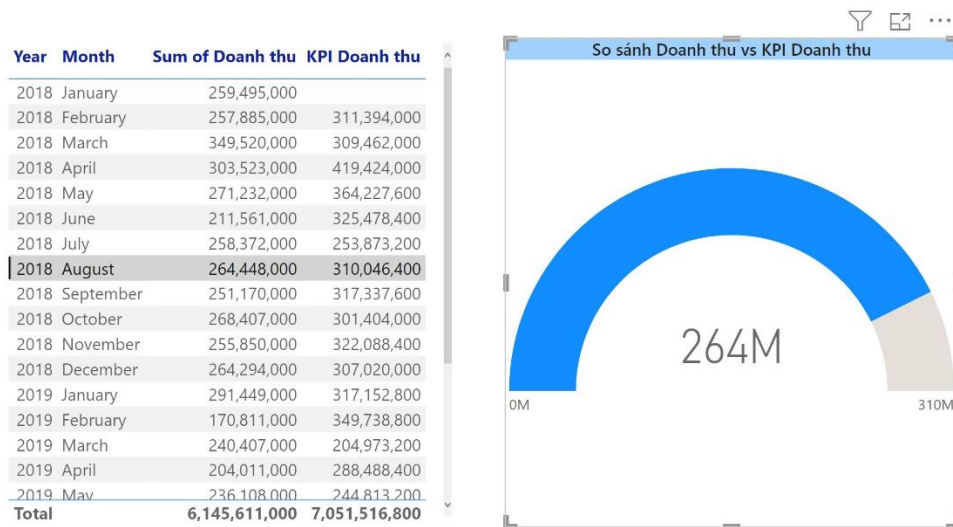
- Tạo 1 Table Visual sau đó chúng ta đưa vào nó các cột Year, Month, Doanh thu và KPI Doanh thu

- Tạo 1 Gauge Visual chúng ta đưa cột **Doanh thu (Sum of Doanh thu)** vào phần **Value**; và measure **KPI Doanh thu** vào **Target value** (Giá trị đích muốn so sánh), ta sẽ có hình:



Khi chúng ta click chuột vào từng dòng trên Table Visual tương ứng bên Gauge Visual sẽ thể hiện Doanh thu tháng đó (phần màu xanh dương với con số hiển thị doanh thu tháng (264M), đồng thời Doanh thu KPI (đích muốn đạt được) sẽ thể hiện ở phần màu đỏ.

Lưu ý: chúng ta cũng có thể kéo KPI Doanh thu vào mục Max Value thì sẽ có cách hiển thị khác như sau:



## 2.5. Sử dụng Line và Column Visual

Các biểu đồ dạng dòng (Line) và dạng cột (Column) là các loại biểu đồ phổ biến thường được sử dụng để trực quan hoá dữ liệu tốt nhất, giúp cho người dùng có góc nhìn tổng quan và chi tiết về sự biến thiên dữ liệu. Có 1 số dạng thường dùng như sau:

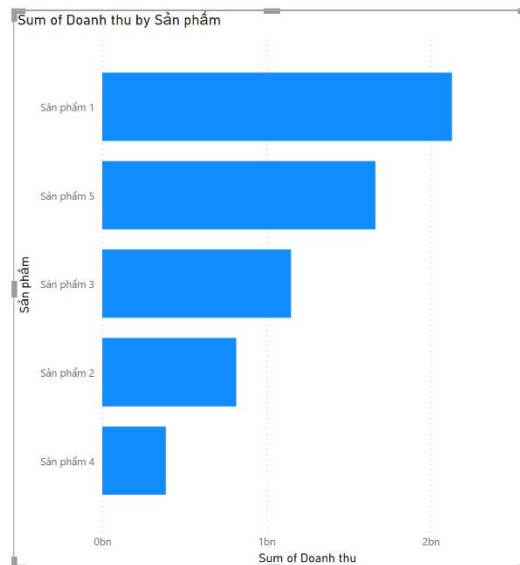
### 2.5.1. Stacked Bar Chart

Ví dụ 1: Chúng ta vẫn sử dụng file Map-data.xlsx chứa dữ liệu doanh thu của một số cửa hàng, hãy nạp vào Power BI và dùng Stacked Bar Chart Visual để mô tả **Doanh thu theo từng Sản phẩm**, thực hiện như sau:

Click nút **Stacked Bar Chart** để tạo biểu đồ (Trong Tab **Add Data to your Visual**) chúng ta thực hiện kéo các cột vào các phần sau:

- Trong **Y-axis (trục y)** kéo cột Doanh thu vào

- Trong **X-axis (trục x)** kéo cột Sản phẩm vào
- Chúng ta sẽ có biểu đồ sau:



Để định dạng lại biểu đồ chúng ta chuyển sang Tab **Format your Visual** và thực hiện một số nội dung:

### **Trên tab Visual**

1. Định dạng trục Y chọn **Y-axis** có 2 nội dung:

- **Value:** trong mục này cho phép ta định dạng font cho các nội dung hiển thị trên trục Y của biểu đồ
- **Title:** cho phép chúng ta bật hoặc tắt, định dạng font chữ cho tiêu đề trục Y (chữ *sản phẩm*)

2. Định dạng trục X chọn **X-axis** có các nội dung:

- **Value:** trong mục này cho phép ta định dạng font cho các nội dung hiển thị trên trục X của biểu đồ, đồng thời có thể điều chỉnh đơn vị (ngàn, triệu, tỉ) xuất hiện bằng cách chọn mục **Display units**, và số phần thập phân muốn hiển thị chọn trong phần **Value Decimal places**
- **Title:** cho phép chúng ta bật hoặc tắt, định dạng font chữ cho tiêu đề trục Y (chữ *sản phẩm*)

3. Định dạng các bar chọn **Bars**, mặc định các thanh ngang hiển thị có cùng 1 màu xanh, tuy nhiên chúng ta cũng có thể cho các bar có màu khác nhau, bằng cách chọn **Show All** sau đó chỉnh màu lại cho từng bar.

4. Để hiển thị thêm dữ liệu trên từng bar chúng ta bật mục **Data Labels**

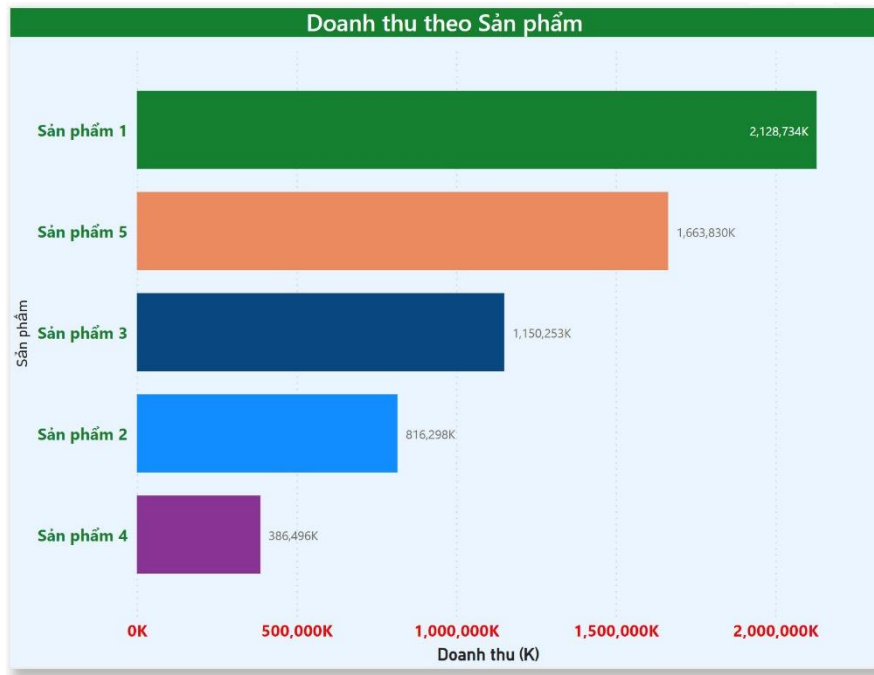
### **Trên tab General**

1. Để định dạng lại tiêu đề của biểu đồ chọn mục **Title**, chúng ta có thể bật tắt tiêu đề này, hoặc muốn chỉnh sửa thì có thể nhập lại tiêu đề mới và định dạng lại font chữ.

2. Để tăng hiệu ứng cho biểu đồ chọn mục **Effect**, trong đó:

- **Background:** dùng để thay đổi màu nền của biểu đồ
- **Visual border:** dùng để tạo đường viền cho biểu đồ
- **Shadow:** tạo bóng nổi 3D cho biểu đồ

Kết quả ta có biểu đồ sau khi định dạng:



Ví dụ 2: Dùng *Stacked Bar Chart Visual* để mô tả *Doanh thu theo từng cửa hàng của từng Sản phẩm*, thực hiện như sau:

Click nút **Stacked Bar Chart** để tạo biểu đồ (Trong Tab **Add Data to your Visual**) chúng ta thực hiện kéo các cột vào các phần sau:

- Trong **Y-axis (trục y)** kéo cột Cửa hàng vào
- Trong **X-axis (trục x)** kéo cột Doanh thu vào
- Trong **Legend** kéo cột Sản phẩm vào

Chúng ta sẽ có biểu đồ sau:



Nếu chúng ta kéo cột Sản phẩm vào vùng **Small Multiples** sẽ có dạng:



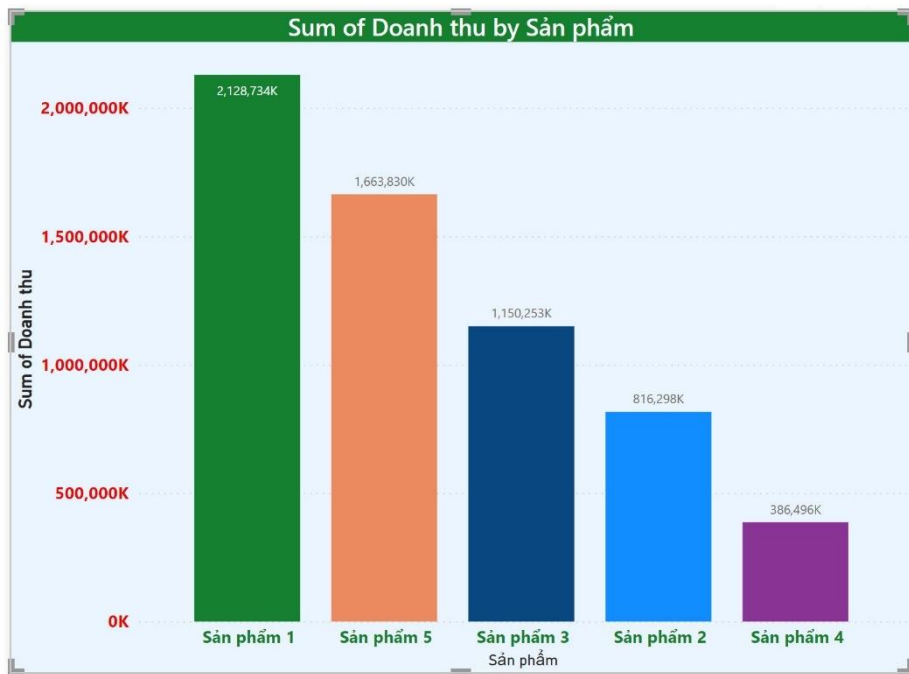
### 2.5.2. Stacked Column Chart

Ví dụ 1: Chúng ta vẫn sử dụng file *Map-data.xlsx* chứa dữ liệu doanh thu của một số cửa hàng, hãy nạp vào Power BI và dùng *Stacked Column Chart Visual* để mô tả **Doanh thu theo từng Sản phẩm**, thực hiện như sau:

Click nút **Stacked Column Chart** để tạo biểu đồ (Trong Tab **Add Data to your Visual**) chúng ta thực hiện kéo các cột vào các phần sau:

- Trong **Y-axis (trục y)** kéo cột Doanh thu vào
- Trong **X-axis (trục x)** kéo cột Sản phẩm vào

Chúng ta sẽ có biểu đồ sau:



Phần tùy biến và định dạng tương tự như biểu đồ dạng **Stacked Bar Chart**

### 2.5.3. Line Chart và Area Chart

Loại biểu đồ dạng Line Chart thường được biểu diễn cho các đường xu hướng

Ví dụ 1: Chúng ta vẫn sử dụng file *Map-data.xlsx* chứa dữ liệu doanh thu của một số cửa hàng, hãy nạp vào Power BI và dùng *Line Chart Visual* để mô tả **Doanh thu theo tháng**, thực hiện như sau:

Trong Tab **Add Data to your Visual** , click nút **Line Chart** để tạo biểu đồ chúng ta thực hiện kéo các cột vào các phần sau:

- Trong **Y-axis (trục y)** kéo cột Doanh thu vào
- Trong **X-axis (trục x)** kéo cột Month vào

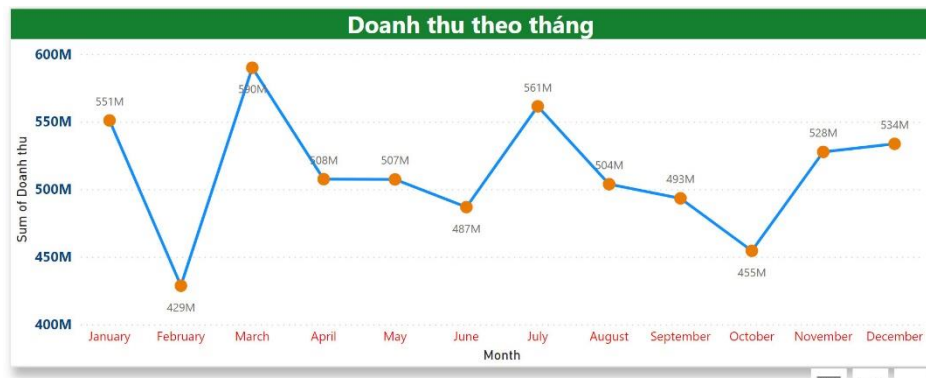
Để định dạng lại biểu đồ chúng ta chuyển sang Tab **Format your Visual** và thực hiện một số nội dung:

### **Trên tab Visual**

Chúng ta thực hiện các định dạng tương tự như các loại trên, có 1 số định dạng riêng cho Line Chart như sau:

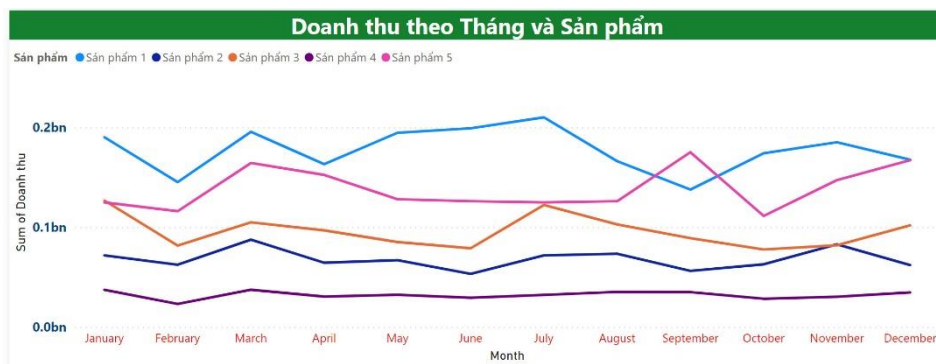
- **Markers:** điểm đánh dấu dữ liệu có thể bật hoặc tắt, trong đó có mục **Shape** để chọn kiểu nút, và **Size** để thay đổi kích thước nút và **Color** dùng để thay đổi màu của nút.

- **Data Labels:** Bật hoặc tắt nhãn dữ liệu trên các nút



*Lưu ý: Chúng ta nên tạo bộ lọc Slicer để xem được chính xác hơn dữ liệu theo thời gian*

Nếu muốn xem thêm 1 tiêu chí nữa giả sử mô tả **Doanh thu theo tháng và từng sản phẩm** thì kéo thêm cột Sản phẩm vào vùng **Legend** ta sẽ có dạng:



## 2.5.4. Combo Chart

Trong Power BI, một Combo chart là một kiểu trực quan được kết hợp giữa một biểu đồ đường (line chart) và một biểu đồ cột (column chart). Sử dụng 2 loại biểu đồ trên cùng một trực quan giúp công tác phân tích dữ liệu hiệu quả hơn.

Khi nào nên sử dụng Combo chart:

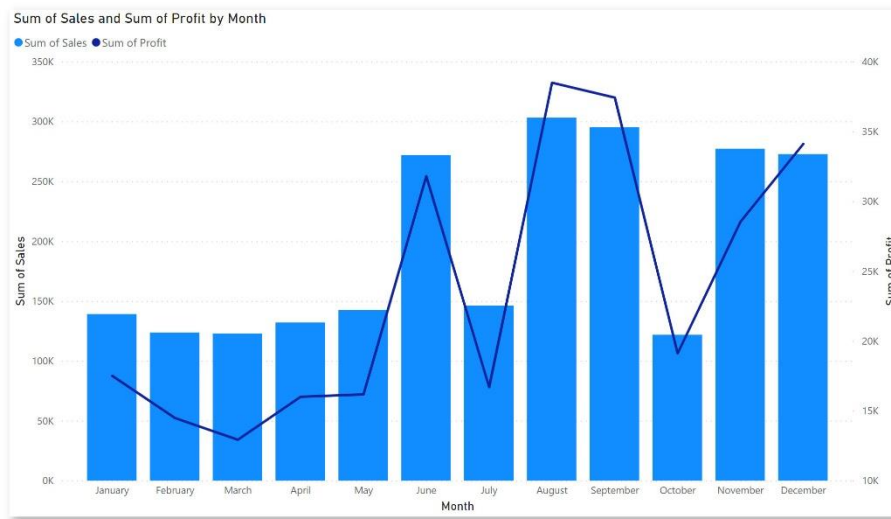
- Khi chúng ta có một biểu đồ đường và một biểu đồ cột có cùng trục Y.
- Khi cần so sánh nhiều measures với các khoảng giá trị khác nhau.
- Để minh họa mối tương quan giữa hai measures trong một trực quan.
- Để kiểm tra một measure có đáp ứng mục tiêu được xác định bởi một measure khác hay không.
- Để tiết kiệm không gian làm việc.

*Ví dụ: Hãy nạp file P6-AmazingMartEU2Geo.xlsx vào Power BI và tạo biểu đồ combo chart để mô tả Doanh thu (Sales) và Lợi nhuận (Profit) theo các tháng*

Trong Tab **Add Data to your Visual** , click nút **Line and Stacked Column Chart** để tạo biểu đồ chúng ta thực hiện kéo các cột vào các phần sau:



- Trong **X-axis (trục x)** kéo cột **Month** vào
- Trong **Column Y-axis (trục y thứ nhất)** kéo cột **Sales** vào
- Trong **Line Y-axis (trục y thứ hai)** kéo cột **Profit** vào

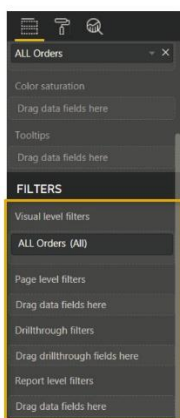


## 2.6. Kỹ thuật lọc dữ liệu và Slicer

Có 2 cách lọc dữ liệu trong Power BI:

- Lọc bằng Filters Pane
- Lọc bằng Slicer

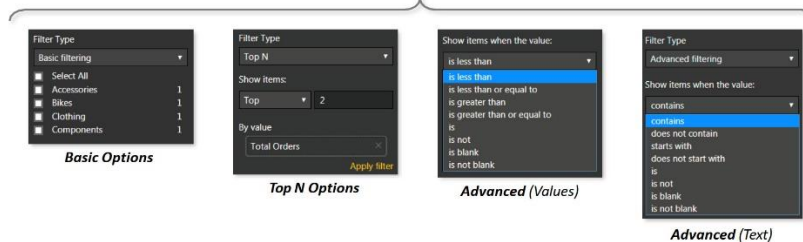
Có 4 cấp độ lọc trong Power BI



Có 04 cấp độ lọc trong báo cáo Power BI:

1. **Visual Level:** Áp dụng cho một *visual cụ thể*
2. **Page Level:** Áp dụng cho tất cả *visuals trong một trang*
3. **Report Level:** Áp dụng cho tất cả *visuals trong báo cáo*
4. **Drillthrough:** Áp dụng cho một *pages*, và *thay đổi* dựa trên user path trong máy

Filter settings include Basic, Advanced, and Top N options

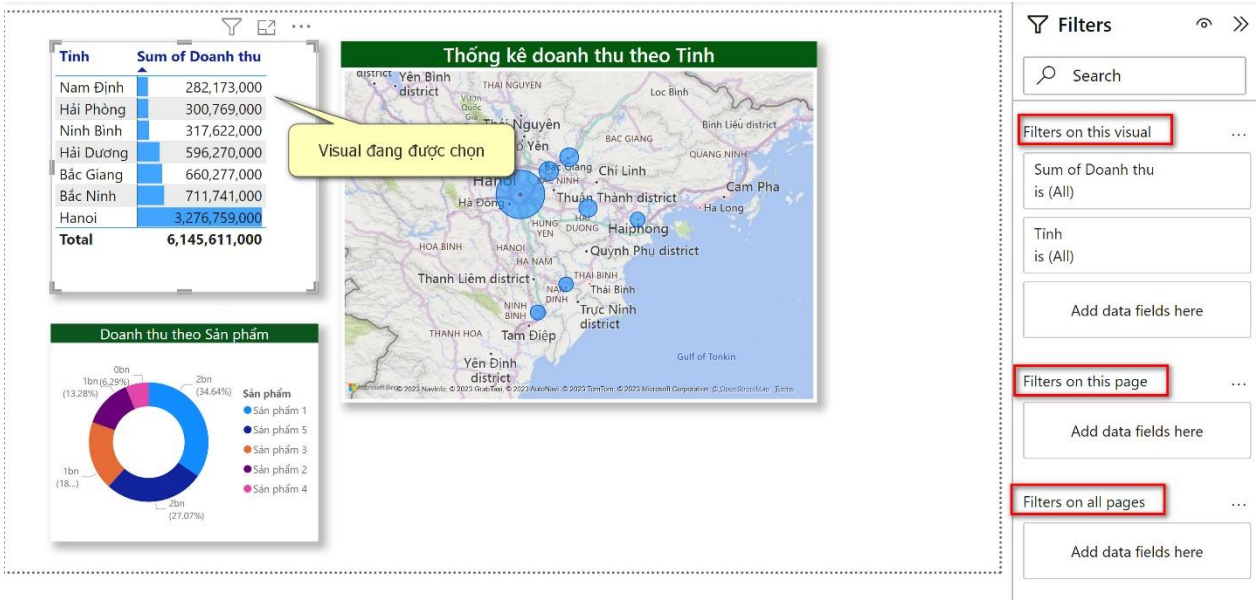


### 2.6.1. Lọc bằng Filter Pane

Khi chúng ta chọn cụ thể 1 cái Visual thì Filter Pane sẽ xuất hiện nội dung gồm 3 phần:

- **Filter on this Visual:** Thiết lập bộ lọc cho Visual đang được chọn
- **Filter on this page:** Thiết lập bộ lọc cho tất cả các Visual trên trang hiện hành
- **Filter on All pages:** Thiết lập bộ lọc cho tất cả các Visual trên tất cả các trang

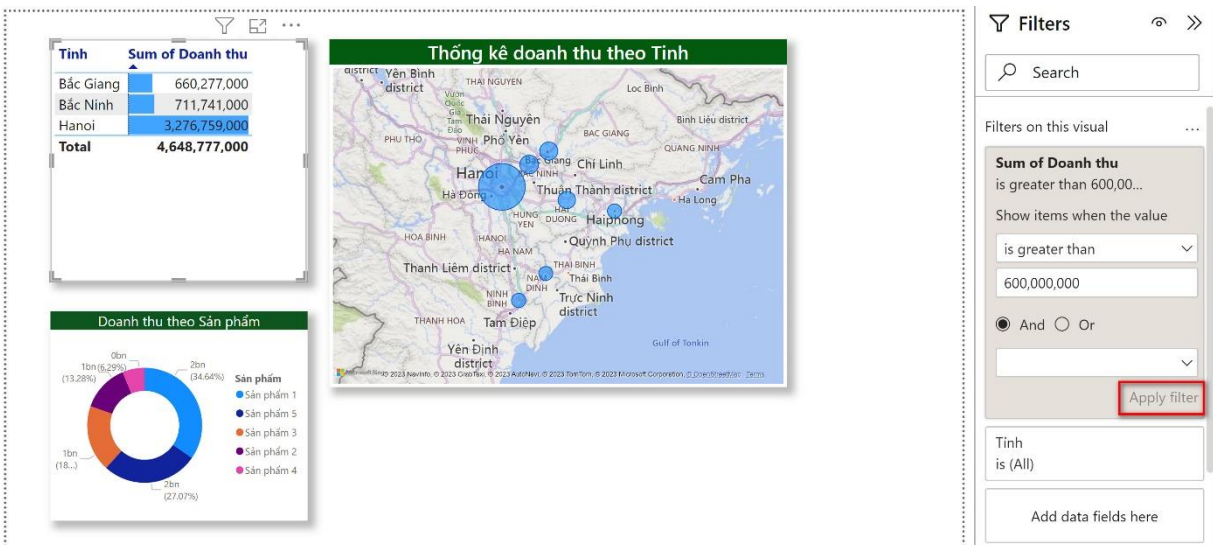
#### Cấp độ Visual Level



Ví dụ: Chúng ta cần thiết lập bộ lọc cho Table Visual. Thực hiện như sau:

- Click chuột vào Table Visual
- Chuyển sang Filter Pane, trong vùng **Filter on this Visual** chúng ta sẽ thiết lập điều kiện lọc ở đây, dựa vào 2 cột **Doanh thu** và **Tỉnh**

Chẳng hạn chỉ muốn hiển thị các Tỉnh có doanh thu lớn hơn 600.000.000, chúng ta nhập điều kiện vào và click nút **Apply Filter** là xong có kết quả như hình.

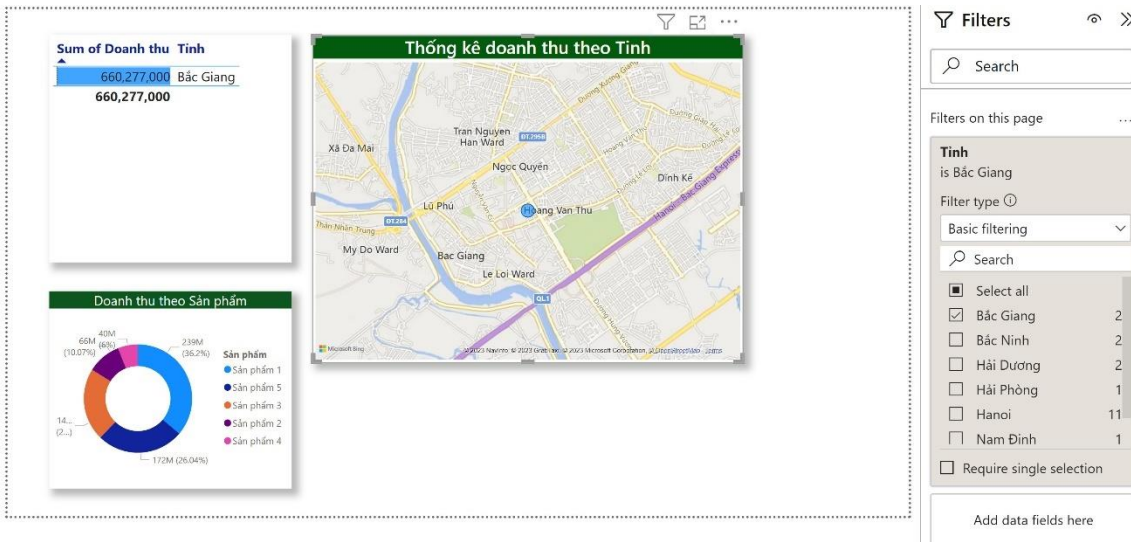


### Cấp độ Page Level

Với cấp độ này chúng ta sẽ kéo 1 trường muốn làm bộ lọc vào khung **Filter on this page** để làm điều kiện lọc.

Ví dụ: Chúng ta cần làm bộ lọc cho tất cả các Visual trên trang theo điều kiện lọc là theo **Tỉnh**

- Chúng ta kéo cột Tỉnh vào ô **Add data fields here**
- Giả sử chúng ta chỉ cần xem tỉnh **Bắc giang** thì chọn tỉnh này và sẽ có kết quả



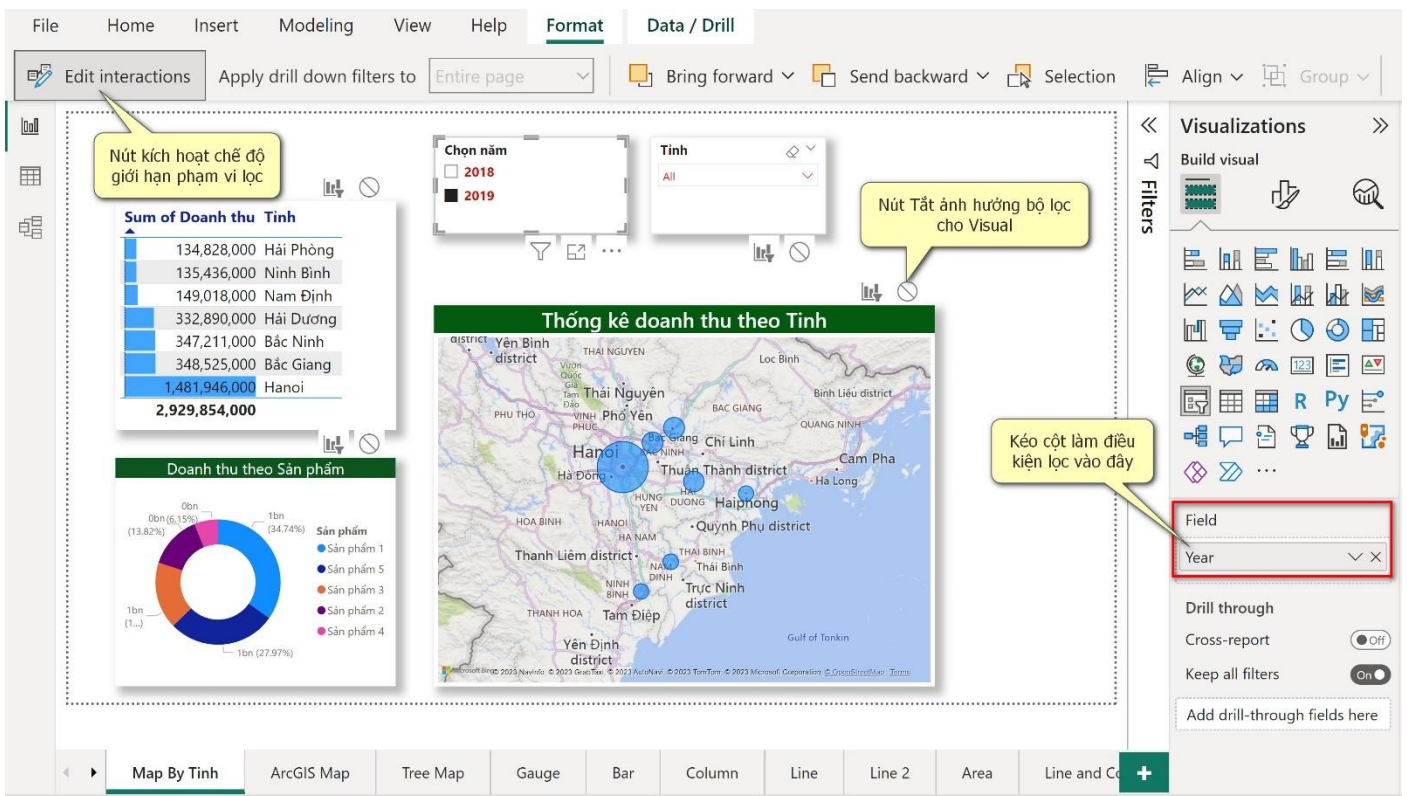
### Cấp độ Report Level

Với cấp độ này tương tự chúng ta sẽ kéo 1 trường muốn làm bộ lọc vào khung **Filter on All Pages** để làm điều kiện lọc.

### 2.6.2. Lọc bằng Slicer

Để tạo bộ lọc bằng Slicer, chúng ta phải tạo 1 **Slicer Visual** trong trang, với cách làm này nhìn vào Report sẽ trực quan hơn.

Cách thực hiện tương tự, chúng ta chỉ cần Click chọn **Slicer Visual** để đưa Slicer vào trang, và chỉ cần kéo cột chọn làm tiêu chí bộ lọc vào vùng **Field**



### 2.6.3. Lọc chi tiết bằng Drill through

Kỹ thuật lọc bằng Drill through (Khoan xuyên qua) là 1 kỹ thuật lọc đặc biệt của Power BI, cho phép chúng ta lọc chi tiết dữ liệu từ một dữ liệu tổng hợp.

Giả sử chúng ta vẫn sử dụng file Map-data.xlsx chứa dữ liệu doanh thu của một số cửa hàng và nạp vào Power BI. Có yêu cầu sau:

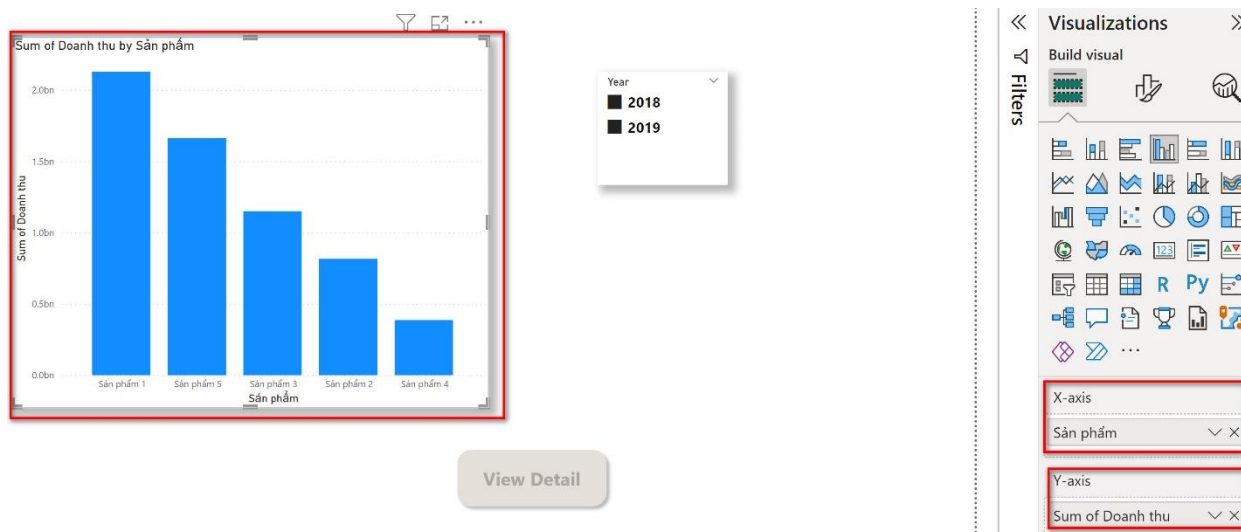
1. Lập một **Clustered Column Chart Visual** để mô tả Doanh thu theo từng sản phẩm
2. Lập một **Slicer** để lọc theo năm

3. Mỗi khi click vào 1 cột sản phẩm để xem doanh thu thì chúng ta có thể xem chi tiết về doanh thu riêng cho sản phẩm đó trên 1 trang khác.

Trình tự thực hiện như sau:

**Bước 1:** Tạo **Trang chủ (Main)** chứa **Clustered Column Chart** Visual để mô tả Doanh thu theo từng sản phẩm.

Thiết lập cho **Clustered Column Chart** Visual như hình dưới:



**Bước 2:** Tạo **Trang chi tiết (Detail)**, trên trang này chúng ta tạo 2 Table Visual:

1. Table mô tả Cửa hàng, Tháng, Doanh thu

The table displays the following data:

Cửa hàng	Month	Sum of Doanh thu
Cửa hàng 1	January	21,628,000
Cửa hàng 10	January	27,407,000
Cửa hàng 11	January	21,429,000
Cửa hàng 12	January	37,282,000
Cửa hàng 13	January	31,395,000
Cửa hàng 14	January	25,416,000
Cửa hàng 15	January	32,674,000
Cửa hàng 16	January	43,891,000
Cửa hàng 17	January	25,614,000
Cửa hàng 18	January	31,914,000
Cửa hàng 19	January	13,246,000
Cửa hàng 2	January	24,720,000
Cửa hàng 20	January	19,237,000
Cửa hàng 3	January	38,856,000
Cửa hàng 4	January	19,134,000
Cửa hàng 5	January	29,802,000
Cửa hàng 6	January	27,102,000
<b>Total</b>		<b>6,145,611,000</b>

The visualization pane configuration is as follows:

- Columns: Cửa hàng, Month, Sum of Doanh thu

2. Table mô tả Year, Sản phẩm

Cửa hàng	Month	Sum of Doanh thu
Cửa hàng 1	December	1,497,000
Cửa hàng 10	December	4,491,000
Cửa hàng 12	December	998,000
Cửa hàng 13	December	4,990,000
Cửa hàng 14	December	4,990,000
Cửa hàng 15	December	4,990,000
Cửa hàng 16	December	9,481,000
Cửa hàng 17	December	2,994,000
Cửa hàng 18	December	5,988,000
Cửa hàng 2	December	4,491,000
Cửa hàng 20	December	5,489,000
Cửa hàng 4	December	9,980,000
Cửa hàng 5	December	7,485,000
Cửa hàng 6	December	9,481,000
Cửa hàng 7	December	4,990,000
Cửa hàng 11	November	2,495,000
Cửa hàng 11	November	2,495,000
<b>Total</b>		<b>1,110,774,000</b>

**Bước 3:** Quay về Trang chủ tạo 1 Button và cấu hình (giống nội dung các khung màu đỏ)

Nếu dữ liệu nhiều gian đoạn thời gian chúng ta nên tạo bộ lọc ( 2018, 2019), bằng cách tạo một Slicer và cấu hình phần **Field** là cột **Year**.

Để sử dụng chúng ta chỉ cần chọn 1 cột bất kỳ trên biểu đồ và Bấm Ctrl + Click vào nút **View Detail** nó sẽ chuyển sang trang Detail để chúng ta xem chi tiết nội dung.