# TẠO TABLE RELATIONSHIP VÀ DATA MODEL TRONG POWER BI

# TẠO TABLE RELATIONSHIP VÀ DATA MODEL TRONG POWER BI

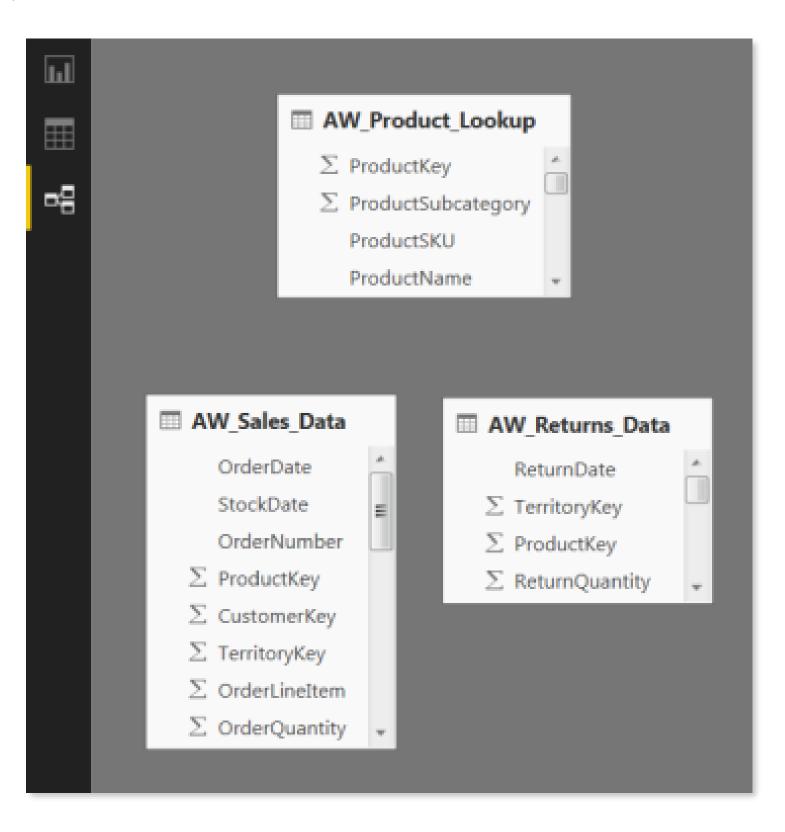
01. Data Model, chuẩn hóa Database

02. Phân biệt Data Table và Lookup Table

03. Thao tác với Relationship, Snowflake

### 01. Data Model, chuẩn hóa Database

# Pata Model là gì?



# Đây *không là* một Data Model



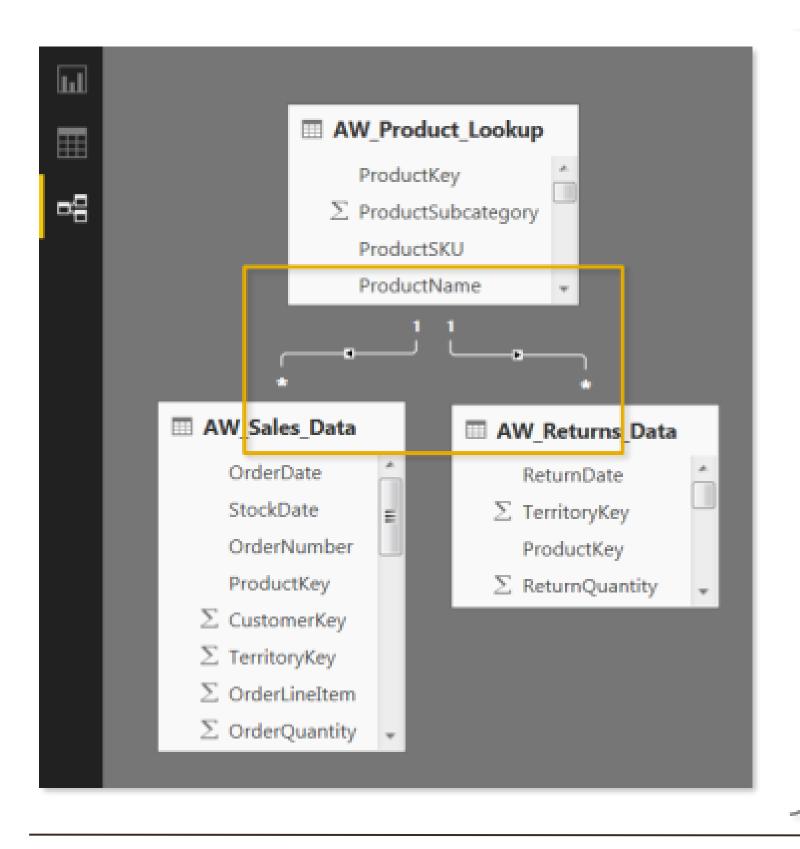
- Đây là tổng hợp của các bảng độc lập, không có kết nối hay mối quan hệ nào
- Nếu bạn thử visualize Orders và Returns theo Product, đây sẽ là kết quả:

ProductName	OrderQuantity	ReturnQuantity	Λ
All-Purpose Bike Stand	84,174	1,828	
AWC Logo Cap	84,174	1,828	
Bike Wash - Dissolver	84,174	1,828	
Cable Lock	84,174	1,828	
Chain	84,174	1,828	
Classic Vest, L	84,174	1,828	
Classic Vest, M	84,174	1,828	
Classic Vest, S	84,174	1,828	
Fender Set - Mountain	84,174	1,828	v
Total	84,174	1,828	
iotai	64,1/4	1,020	

# 01. Data Model, chuẩn hóa Database



#### Data Model là gì?



# Đây mới là một Data Model



- Các bảng được kết nối với nhau bằng các relationships, qua trường dữ liệu chung là Product Key
- Bây giờ, nếu bạn thử visualize Sales và Returns, đây là kết quả:

ProductName	OrderQuantity	ReturnQuantity	A
All-Purpose Bike Stand	234	8	
AWC Logo Cap	4,151	46	
Bike Wash - Dissolver	1,706	25	
Classic Vest, L	182	4	
Classic Vest, M	182	7	
Classic Vest, S	157	8	
Fender Set - Mountain	3,960	54	
Half-Finger Gloves, L	840	18	
Half-Finger Gloves, M	918	16	V
Total	84,174	1,828	

# 01. Data Model, chuẩn hóa Database



#### Chuẩn hóa Database

Chuẩn hóa Database là quá trình tổ chức dữ liệu tại các cột - bảng trong một mối quan hệ dữ liệu, để giảm sự dư thừa và đảm bảo tính toàn vẹn. Một số quy tắc:

- Loại bỏ dữ liệu dư thừa để giảm kích thước bảng và cải thiện tốc độ & hiệu quả xử lý.
- Giảm thiểu lỗi và sự bất thường từ sửa đổi dữ liệu (chèn, cập nhật hoặc xóa...)
- Đơn giản hóa các queries và cấu trúc database có ý nghĩa phân tích.

TIP: Trong việc chuẩn hóa database, các bảng nên đảm bảo mục đích riêng biệt và cụ thể.

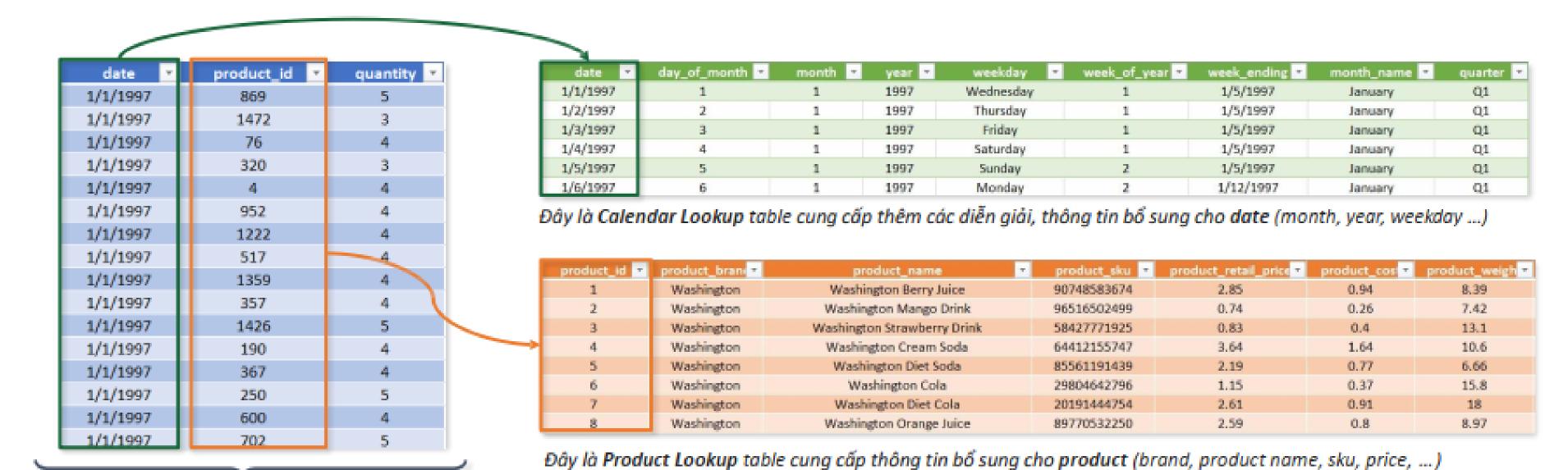
date 💌	product_id *	quantity *	product_brand 💌	product_name *	product_sku 🕶	product_weight •
1/1/1997	869	5	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/7/1997	869	2	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/3/1997	1	4	Washington	Washington Berry Juice	90748583674	8.39
1/1/1997	1472	3	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/6/1997	1472	2	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/5/1997	2	4	Washington	Washington Mango Drink	96516502499	7.42
1/1/1997	76	4	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/1/1997	76	2	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/5/1997	3	2	Washington	Washington Strawberry Drink	58427771925	13.1
1/7/1997	3	2	Washington	Washington Strawberry Drink	58427771925	13.1
1/1/1997	320	3	Excellent	Excellent Cranberry Juice	36570182442	16.4

Khi bạn không quan tâm tới việc chuẩn hóa
Database, bảng của bạn sẽ trông như vậy, tất cả các
hàng sẽ bị duplicate thông tin có thể tìm kiếm trong
một bảng khác (cụ thể là bảng product\_id)
Ngay lập tức, nó có vẻ không ảnh hưởng tới data,
nhưng khi vấn đề sẽ tới khi data scale up lên kích
thước lớn.

# 02. Phân biệt Data Table và Lookup Table

Data Model thông thường bao gồm 2 loại bảng: data (hoặc "fact") tables và lookup (or "dimension") tables:

- Data tables thường dạng number hoặc value, thông thường ở cấp độ đầu tiên, với ID or "key" columns có thể sử dụng để tạo ra các relationships.
- Lookup tables cung cấp thêm các diễn giải, để bổ sung thông tin, thường dạng text.



Đây là Data Table gồm cột "quantity", và có thể kết nối với lookup tables qua cột "date" và "product id"



#### Relationships và Merged Tables



Tôi có thể **merge queries** hoặc sử dụng hàm **LOOKUP** hoặc **RELATED** để thêm dữ liệu trong bảng, vậy tại sao tôi không gộp hết vào một bảng ??

-Anonymous confused man

Bảng gốc Bổ sung thông tin về Calendar Lookup

Bổ sung thông tin về Product Lookup

•		•	•					•	•			•
date *	product_id =	quantity -	day_of_month *	month *	year *	weekday *	month_name *	quarter 💌	product_brand *	product_name -	product_sku *	product_weight *
1/1/1997	869	5	1	1	1997	Wednesday	January	Q1	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/7/1997	869	2	7	1	1997	Tuesday	January	Q1	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/3/1997	1	4	3	1	1997	Friday	January	Q1	Washington	Washington Berry Juice	90748583674	8.39
1/1/1997	1472	3	1	1	1997	Wednesday	January	Q1	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/6/1997	1472	2	6	1	1997	Monday	January	Q1	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/5/1997	2	4	5	1	1997	Sunday	January	Q1	Washington	Washington Mango Drink	96516502499	7.42
1/1/1997	76	4	1	1	1997	Wednesday	January	Q1	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/1/1997	76	2	1	1	1997	Wednesday	January	Q1	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/5/1997	3	2	5	1	1997	Sunday	January	Q1	Washington	Washington Strawberry Drink	58427771925	13.1
1/7/1997	3	2	7	1	1997	Tuesday	January	Q1	Washington	Washington Strawberry Drink	58427771925	13.1
1/1/1997	320	3	1	1	1997	Wednesday	January	Q1	Excellent	Excellent Cranberry Juice	36570182442	16.4

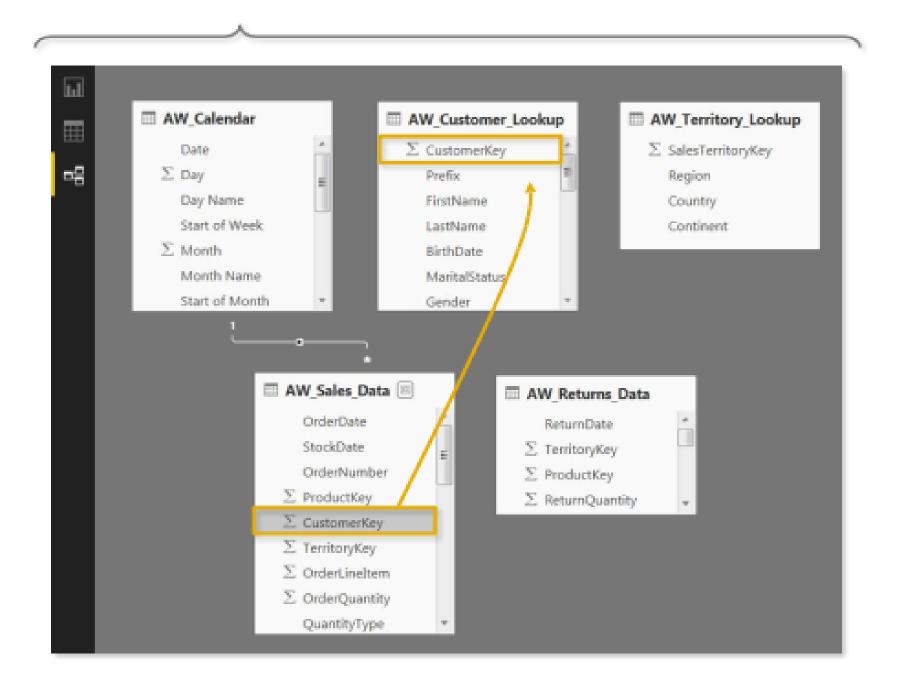
#### Chắc chắn bạn có thể, nhưng nó không hiệu quả!

 Merging data trong cách này tạo ra data thừa và sử dụng đáng kể bộ nhớ và trình xử lý thông tin của máy tính hơn là tạo ra các relationships.

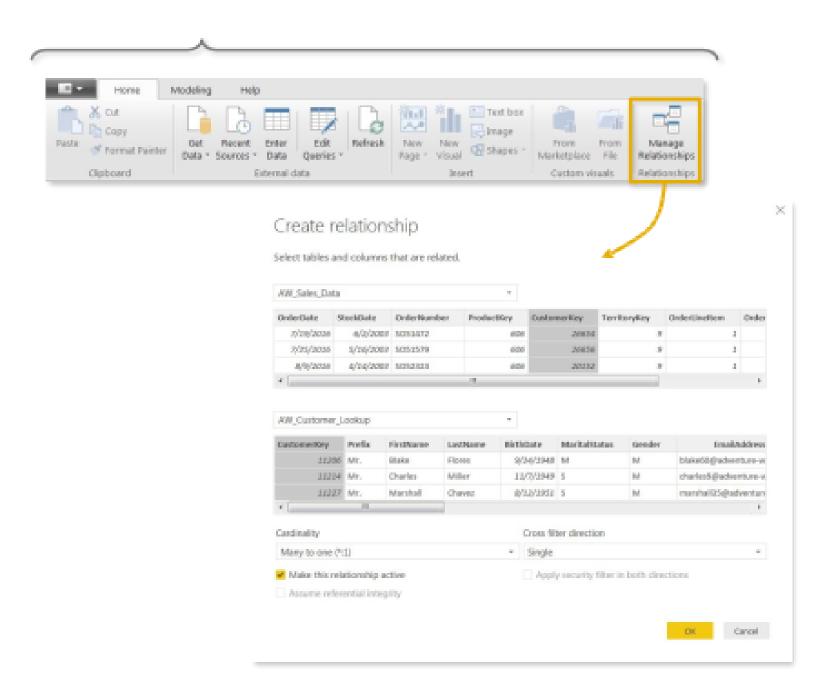


#### Cách tạo Relationships

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

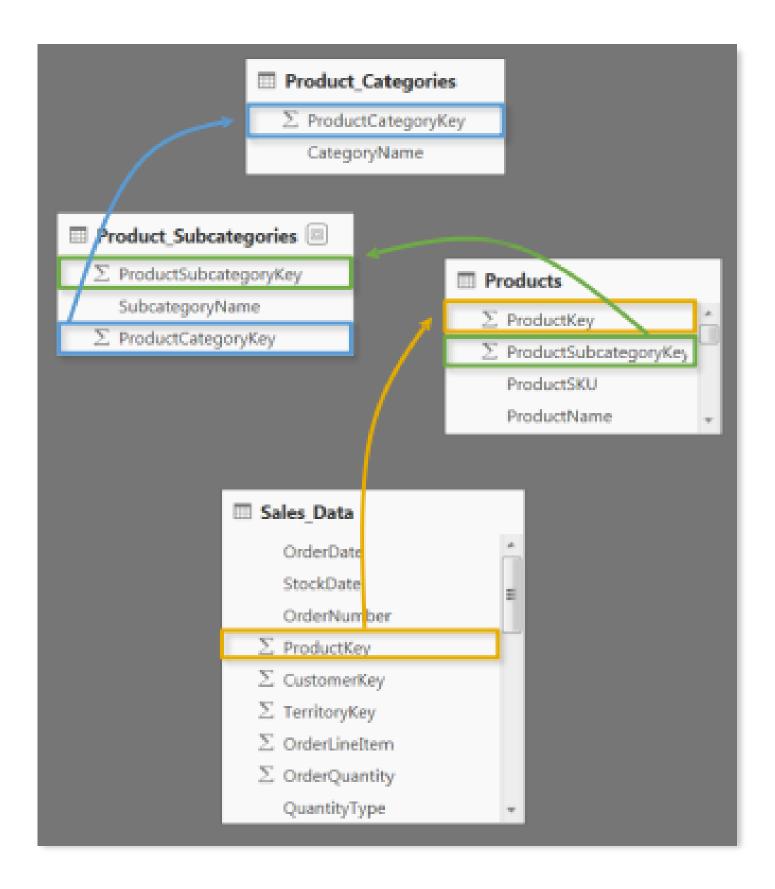


**Option 2**: Thêm hoặc phát hiện relationships sử dụng hộp thoại **"Manage Relationships"** 





#### Tạo một Snowflake



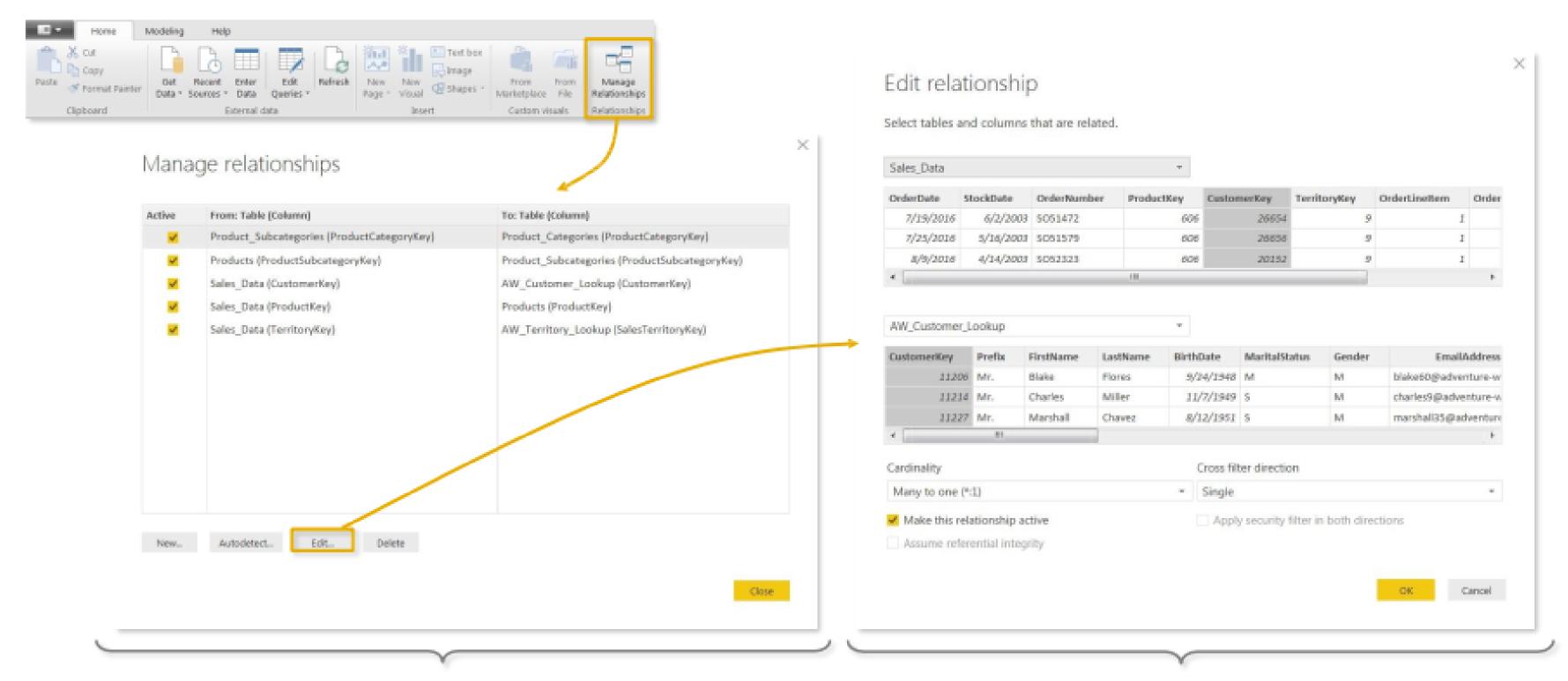
Sale\_Data table có thể kết nối tới Products bằng trường ProductKey, nhưng không thể kết nối trực tiếp tới Subcategories hoặc Categories.

Để tạo ra relationships từ **Products** to **Subcategories** (sử dụng **ProductSubcategoryKey**) và **Subcategories** tới **Categories** (sử dụng **ProductCategoryKey**), chúng ta phải kết nối **Sales\_Data** tới các lookup table; các bảng sẽ được kết nối bắc cầu.

NOTE: Models với chuỗi như hình bên được gọi là "snowflake" (phân biệt chuỗi "star" là chuỗi mà lookup tables được kết nối xung quanh một data tables trung tâm)



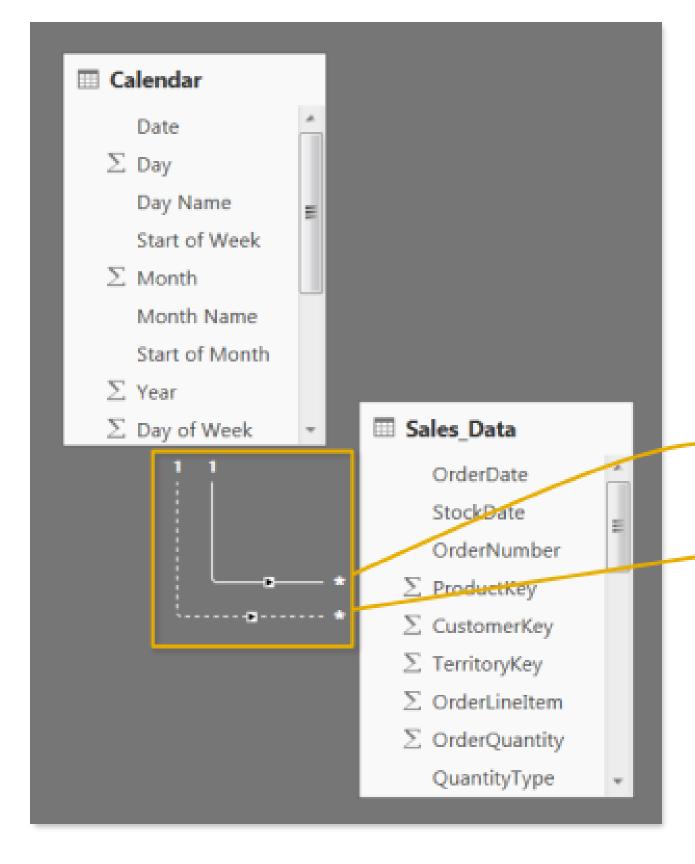
#### Quản lý & Edit Relationships

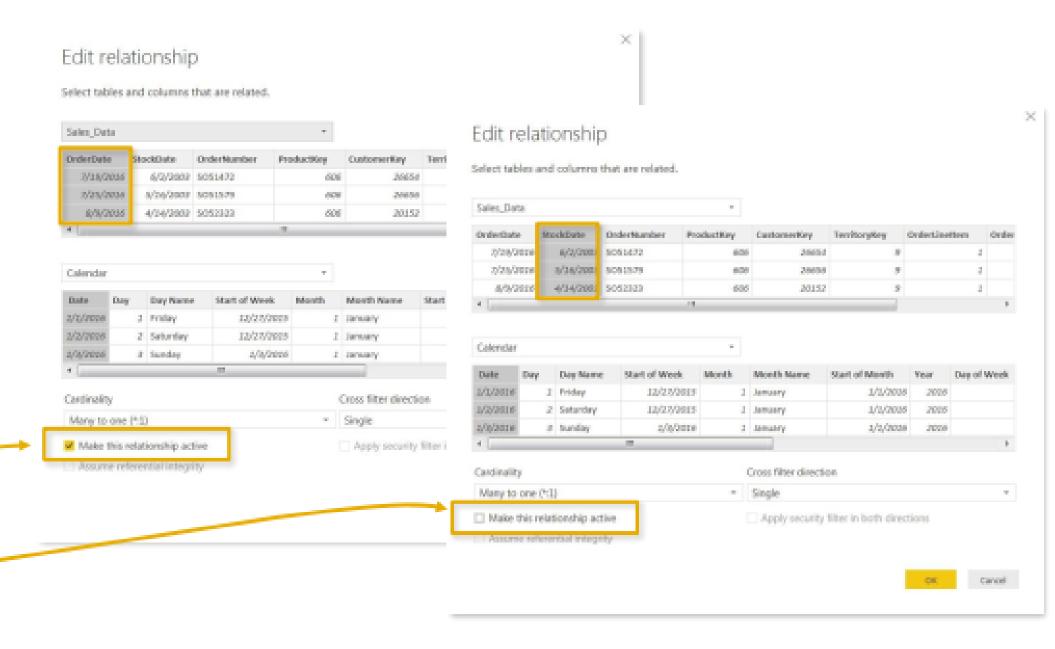


Hộp thoại "Manage Relationships" cho phép thêm, sửa hoặc xóa các relationships Editing tools cho phép activate/deactivate relationships, xem cardinality, và sửa cross filter direction



#### Active và Inactive Relationships



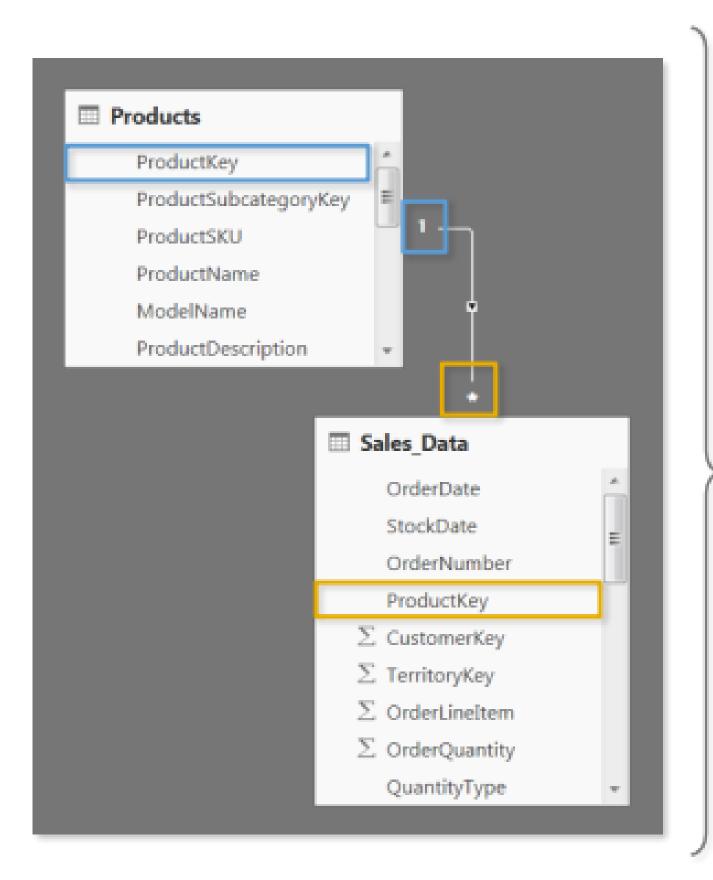


Bảng Sales\_Data gồm 02 trường date (*OrderDate* & *StockDate*), nhưng chúng ta có thể chỉ *active một* relationship tới một trường date trong bảng Calendar

Click đúp vào đường relationship, và check trong hộp thoại tùy chọn "*Make this* relationship active" (chú ý rằng bạn phải deactivate một trường để active trường còn lại)



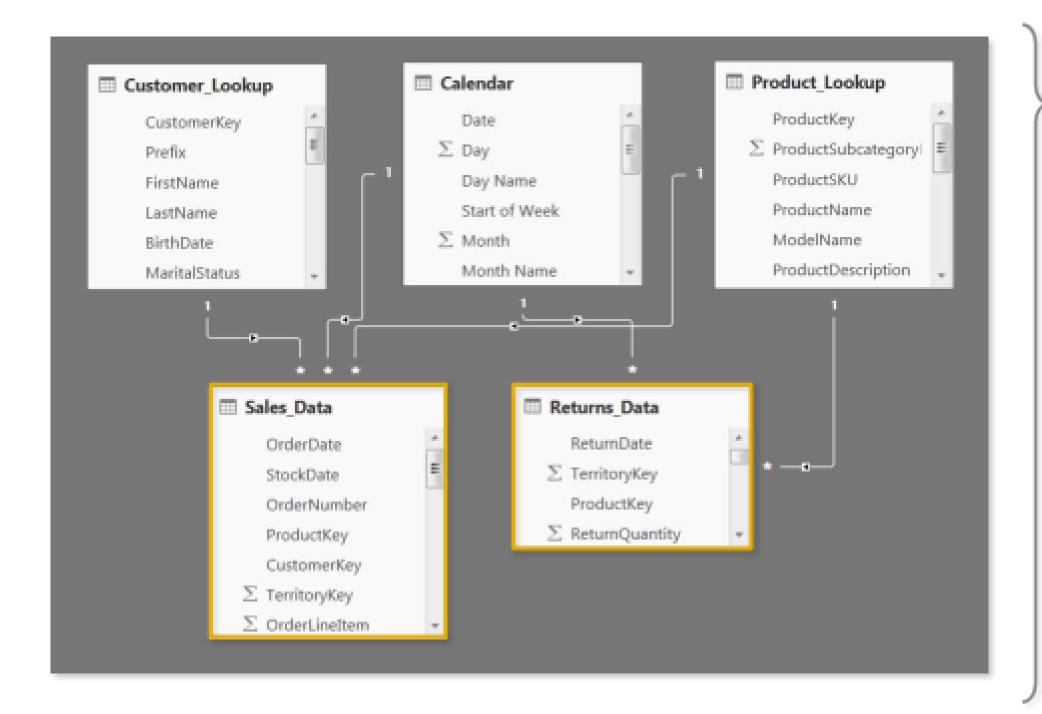
#### **Relationship Cardinality**



Cardinality đề cập tới tính duy nhất của giá trị trong cột



#### Kết nối nhiều bảng trong Data Model



# Dữ liệu gồm 02 bảng dạng *Data table*: Sales\_Data và Returns\_Data

- Chú ý rằng Returns table kết nối tới
   Calendar và Product\_Lookup giống như
   Sales table, nhưng không có trường
   CustomerKey để kết nối tới
   Customer\_Lookup
- Chúng ta không nên tạo relationships giữa Sale\_Data và Returns\_Data để tạo Snowflake vì dữ liệu có thể xử lý sai bản chất
- Ví dụ: chúng ta muốn biết Return theo
  Customerkey, chúng ta tạo liên kết qua trường
  ProductKey với Sale\_Data, PBI vẫn trả ra một kết
  quả nhưng đó là sai vì chúng ta tạo sai



#### NOTE!

Nhìn chung, đừng tạo **relationships trực tiếp** giữa các **data table**; thay vào đó, **kết nối qua shared lookups** 



#### Sử dụng Fields



Trong ví dụ, chúng ta có trường TerritoryKey ở cả 3 bảng, và khi chúng ta sử dụng trường để visualization data, đã có 3 kết quả:

- TH1: Chúng ta dùng đúng trường ở Lookup Table, kết quả trả về ĐÚNG
- TH2: Chúng ta dùng trường ở Sale\_Data (Data Table 1), Kết quả trả về chỉ ĐÚNG cho Sale Data, SAI cho Return Data
- TH3: Chúng ta dùng trường ở Return\_Data (Data Table 2), Kết quả trả về chỉ ĐÚNG cho Return\_Data, SAI cho Sale\_Data

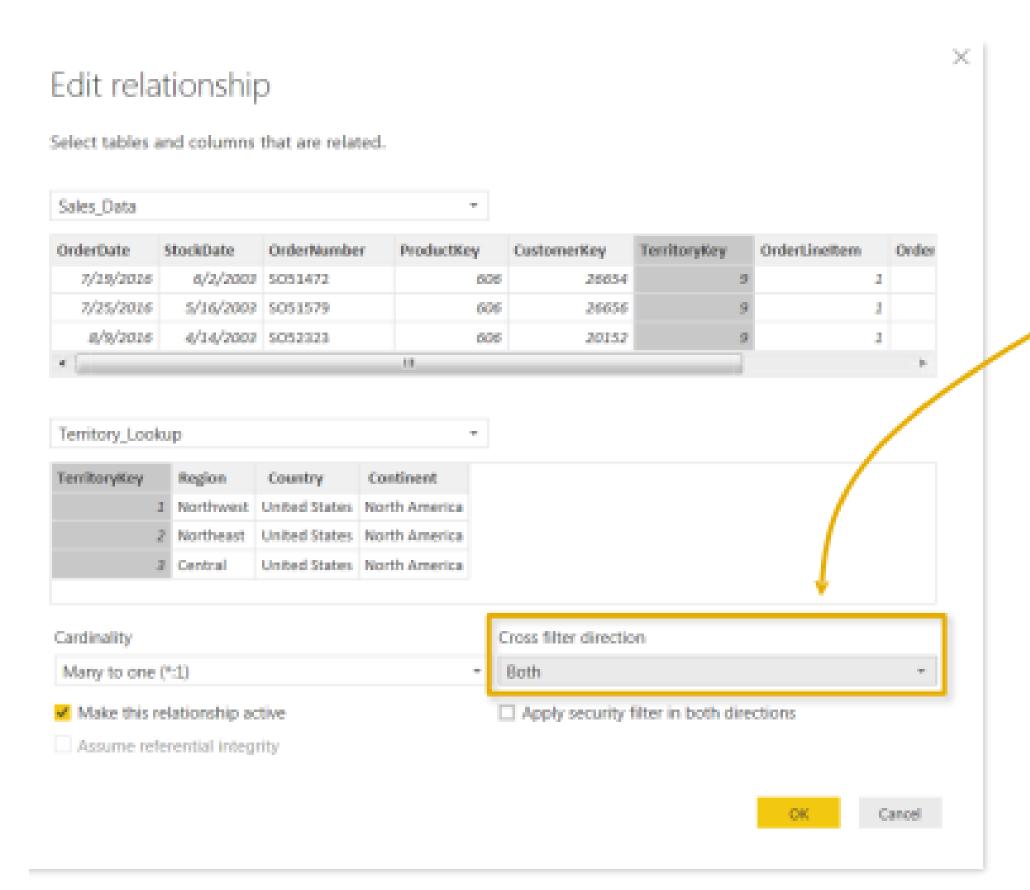
TerritoryKey	OrderQuantity	ReturnQuantity
1	12,513	270
2	40	
3	30	
4	17,191	362
5	49	
6	894	238
7	7,862	186
8	7,950	163
9	17,951	404
10	9,694	204
Total	84,174	1,828

TerritoryKey	OrderQuantity	ReturnQuantity
1	12,513	1,828
2	40	1,828
3	30	1,828
4	17,191	1,828
5	49	128
6	20,894	1,828
7	7,862	1,828
8	7,950	1,828
9	17,951	1,828
10	9,694	1,828
Total	84,174	1,828

TerritoryKey	OrderQuantity	ReturnQuantity
1	84,174	270
4	84,174	362
5	84,174	1
6	174	238
7	84,174	186
8	84,174	163
9	84,174	404
10	84,174	204
Total	84,174	1,828



#### Cài đặt Filters hai chiều







#### Một số lưu ý: Data Modeling

- Xây dựng một Model tiêu chuẩn ngay từ khi bắt đầu.
  - Đảm bảo mỗi bảng là một bảng riêng biệt, cho một mục đích cụ thể.
  - Sử dụng Relationship thay cho Merge Table; bảng dài và hẹp luôn tốt hơn bảng ngắn và rộng.
- Tổ chức lookup tables phía trên data tables.
  - Điều này luôn nhắc cho chúng ta về flow kết nối và phân loại các loại table.
- Tránh những relationship phức tạp nếu không cần thiết.
  - Hạn chế sử dụng filter hai chiều, chỉ nên sử dụng filter 1 chiều.