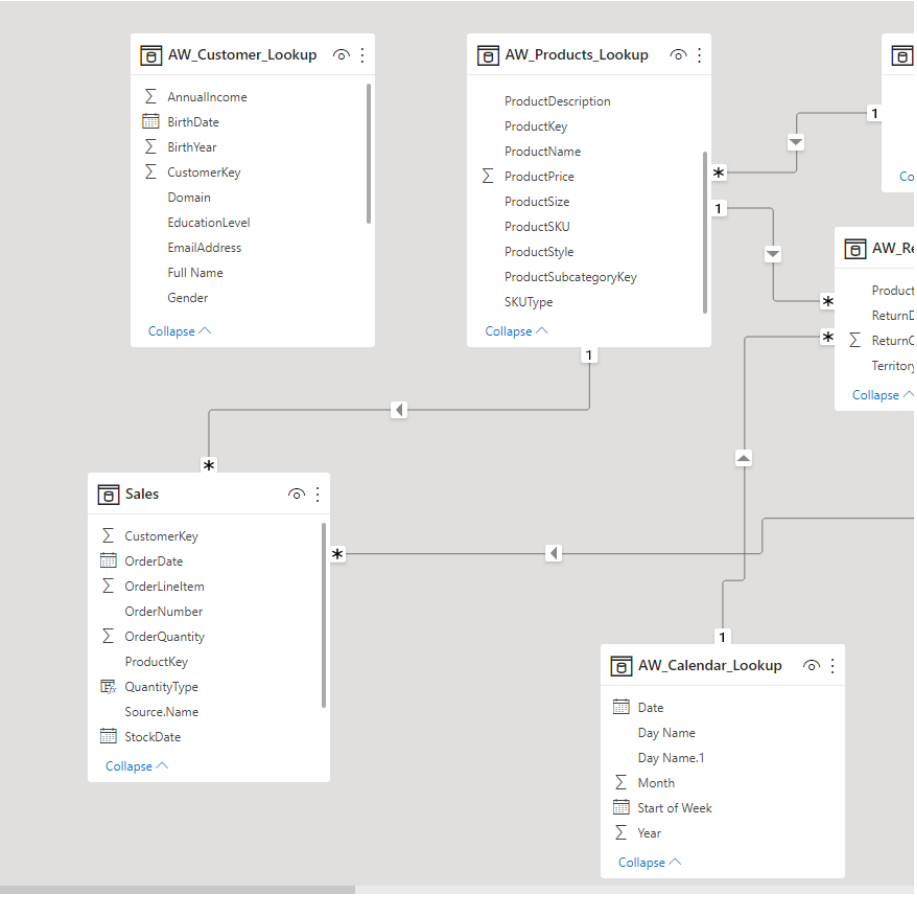
Exercise: Creating Table Relationships & Data Models in Power BI

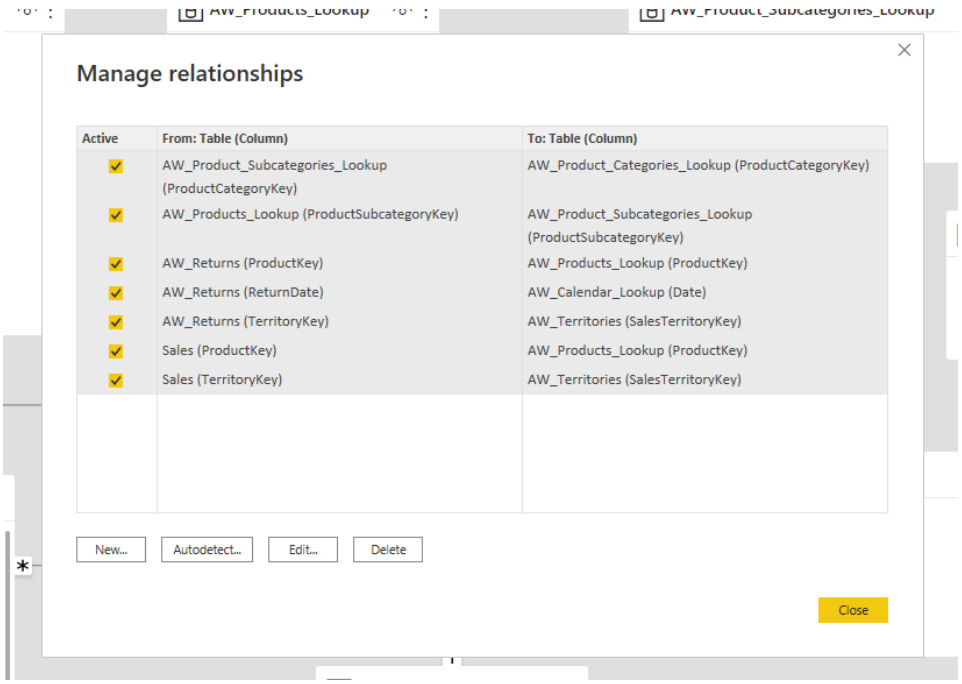
- Using your Adventure Works report file, complete the following:

- **1)** Navigate to the **RELATIONSHIPS** view, and perform the following actions
- Right-click to delete each relationship between **AW_Sales**, **AW_Customer_Lookup** and **AW_Calendar_Lookup** (*including both date fields*)

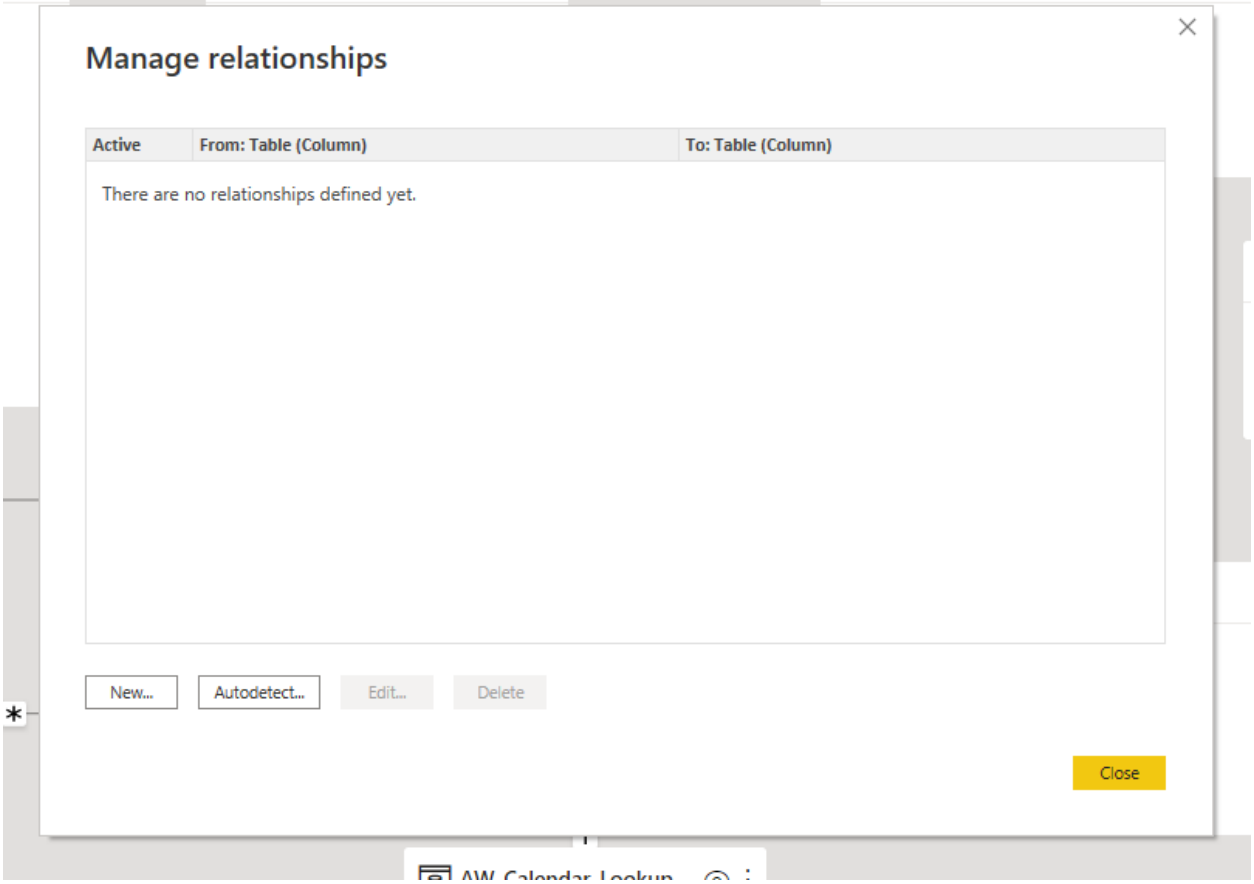


- Use the **Manage Relationships** tool to delete *all* remaining relationships between all tables

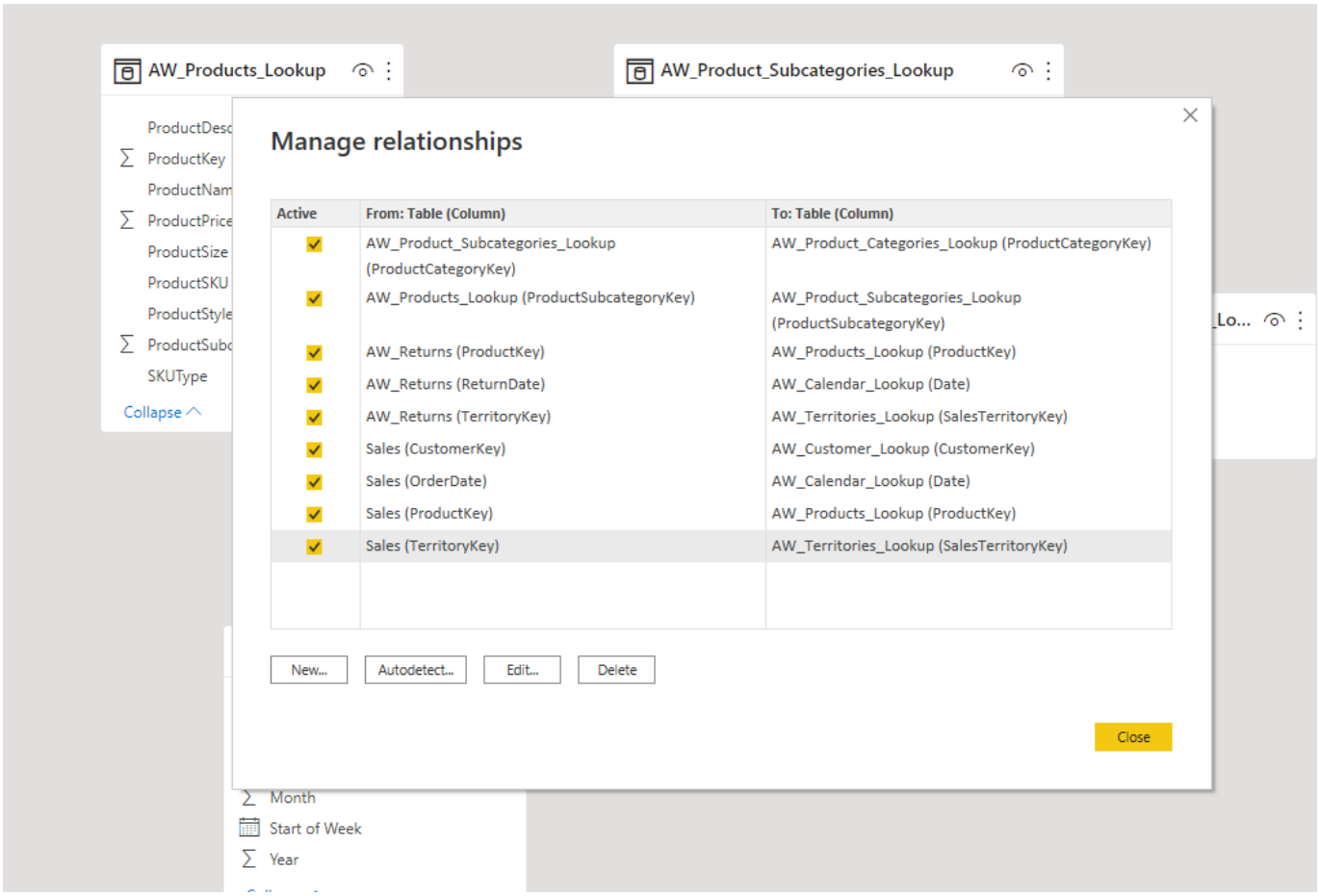
Before



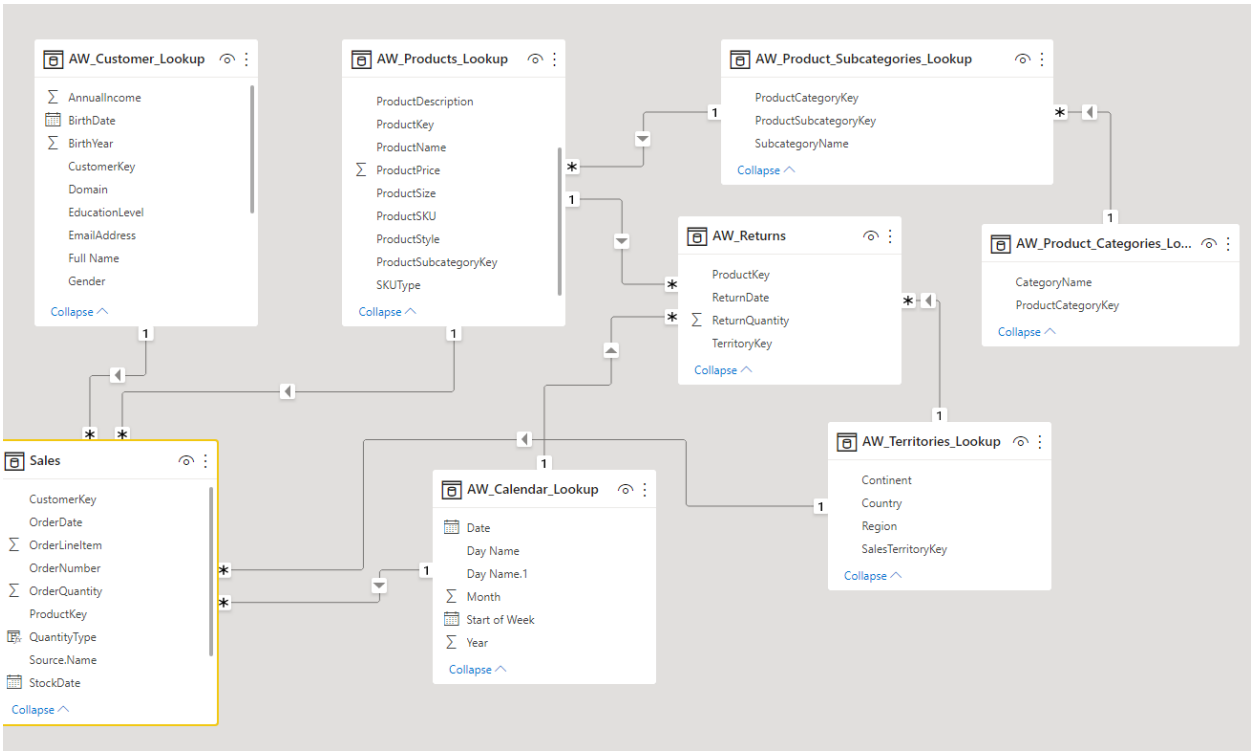
After



- **2)** Recreate all table relationships (*using any method you prefer*), and confirm the following :



- Cardinality is **1-to-Many** for all relationships
- Filters are all **One-Way** (*no two-way filters*)
- Filter direction correctly flows "**downstream**" to data tables
- Data tables are **not connected** directly to one another
- Both data tables are connected to **all valid lookup tables**
- Product-related tables follow a **snowflake schema**



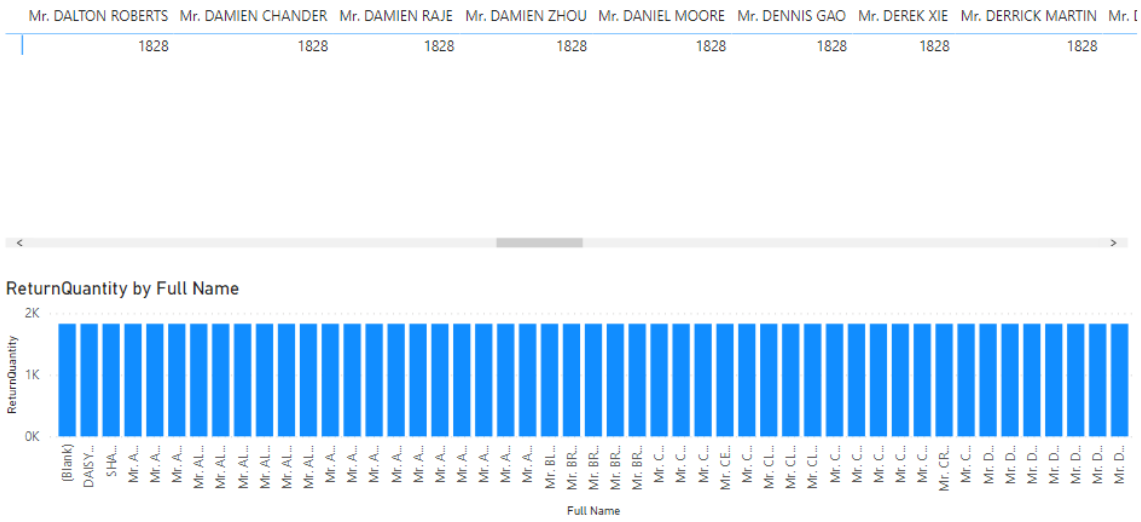
- **3)** Return to the **REPORT** view, and complete the following:
- Edit (*or insert*) the matrix visual to show **ReturnQuantity** (*values*) by **CategoryName** (*rows*) from the **AW_Product_Category_Lookup** table
  - *Which category saw the highest volume of returns? How many?*

CategoryName	ReturnQuantity
Accessories	1130
Bikes	429
Clothing	269
Total	**1828**

- Replace **CategoryName** with **Year** from the **AW_Calendar_Lookup** table
  - *How many returns do you see in 2015 vs. 2016?*

Year	ReturnQuantity
2015	86
2016	770
2017	972
Total	**1828**

- Replace **Year** with **FullName** from the **AW_Customer_Lookup** table
  - *What do you see, and why?*



No applicable information or relationship field between the “Customer_Lookup” table and the “Return” Items.

- Update the matrix to show both **OrderQuantity** and **ReturnQuantity** (*values*) by **ProductKey** (*rows*) fro m the **AW_Product_Lookup** table
  - *What was the total OrderQuantity for Product #338?*

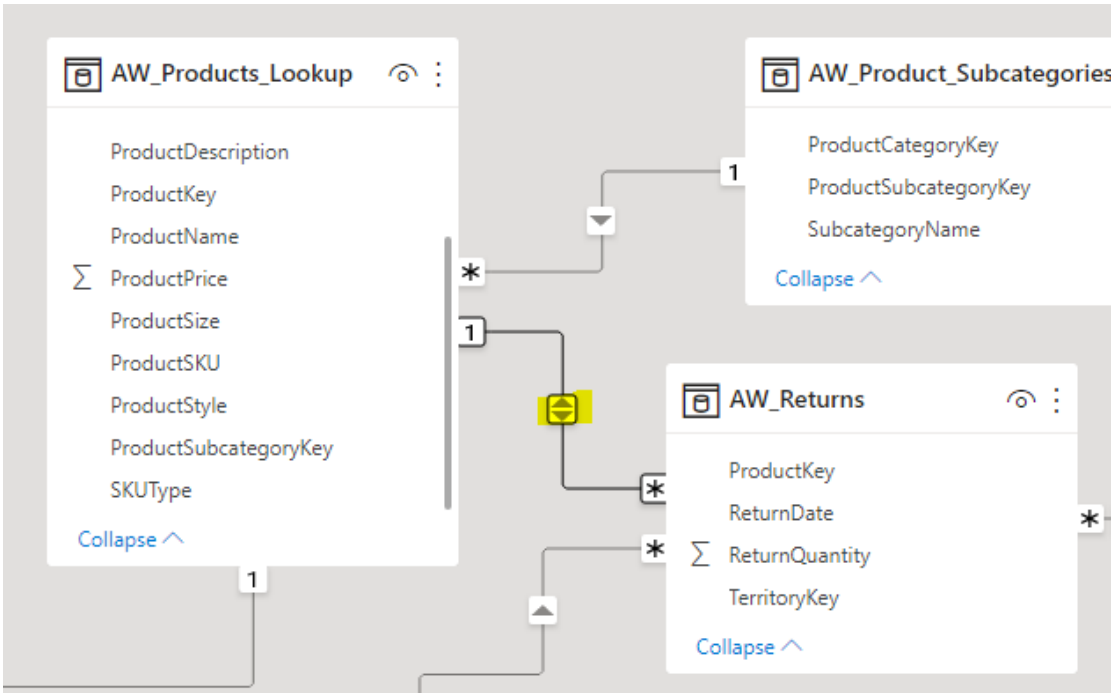
ProductKey	ReturnQuantity	OrderQuantity
326	3	65
328	4	75
330	6	51
332	2	64
334	2	63
336	1	50
338		50
340	1	56
342	1	72
344		29
345		25
Total	**1828**	**84174**

- **4)** Unhide the **ProductKey** field from the **AW_Returns** tables (*using either the **DATA** or **RELATIONSHIPS** view*):
- In the matrix,
  replace **ProductKey** from **AW_Product_Lookup** with **ProductKey** from the **AW_Returns** table
  - *Why do we the same repeating values for OrderQuantity?*

ProductKey	ReturnQuantity	OrderQuantity
214	70	84174
215	52	84174
220	66	84174
223	46	84174
226	12	84174
229	15	84174
232	15	84174
235	10	84174
310	4	84174
311	7	84174
312	8	84174
313	5	84174
Total	**1828**	**84174**

--- The single filter direction limits the actual count per product key within the “Return table” only although the repeated value represents the total count from the other relationship or associated table (Sales). Meanwhile, the recorded/listed product key in the Return table will be the basis of the above matrix.

- Edit the relationship between **AW_Returns** and **AW_Product_Lookup** to change the cross filter direction from *Single* to *Both*
  - *Why does the visual now show OrderQuantity values by product, even though we are using **ProductKey** from **AW_Returns**?*
  - *How many orders do we see now for Product #338? What's going on here?*

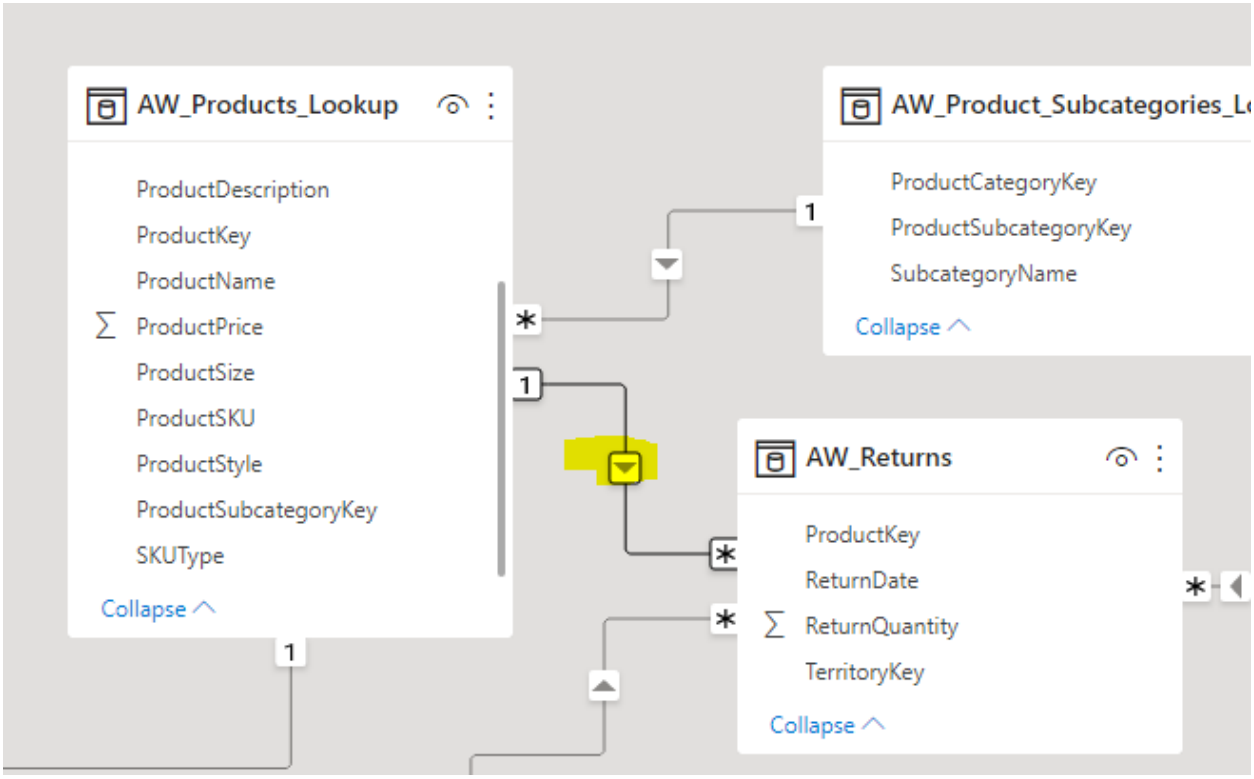


The dual filter direction allows cross order quantity count per product key @ Sales table although the "product key" in the Return Table remained the primary reference. Since there is no return quantity count related to product key # 338 ; then , product #338 creturn & order quantity count will not appear in the matrix.

ProductKey	ReturnQuantity	OrderQuantity
326	3	20
328	4	47
330	6	52
332	2	28
334	2	16
336	1	19
340	1	10
342	1	8
346	2	11
347	1	7
348	1	11
Total	**1828**	**84174**

- **5)** Complete the following:
- Change the cross filter direction between **AW_Returns** and **AW_Product_Lookup** back to *single* (*One-Way*)

- Hide the **ProductKey** field in the **AW_Returns** table from report view (*and any other foreign keys, if necessary*)



- Update the matrix to show **ProductKey** from the **AW_Product_Lookup**, rather than **AW_Returns**