# Chapter 8: Modeling Data Project

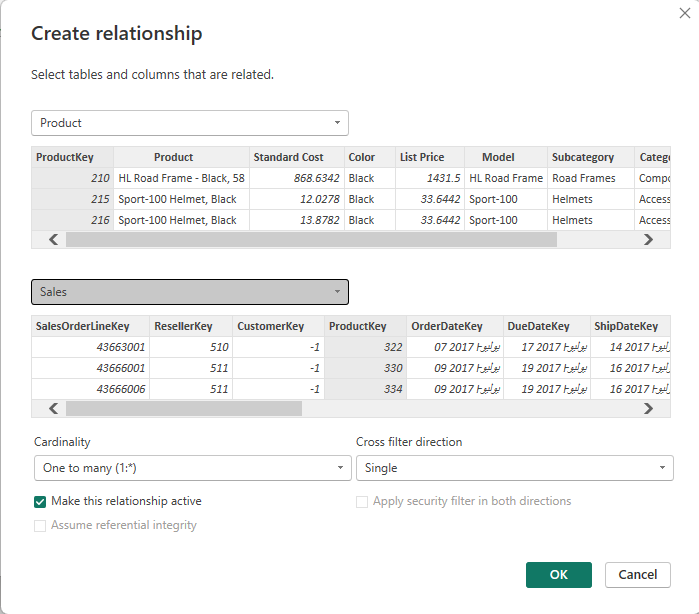
## Step 1: Create Tables Relationships

1. The first step in data modeling is defining the initial relationships between the tables. In this task. You will define relationships using different user interface techniques.
2. Select the **model view** icon on the left bar.
3. zoom out to show our tables to make things a little bit more visible using the icon on bottom right corner.
4. A screenshot of a computer

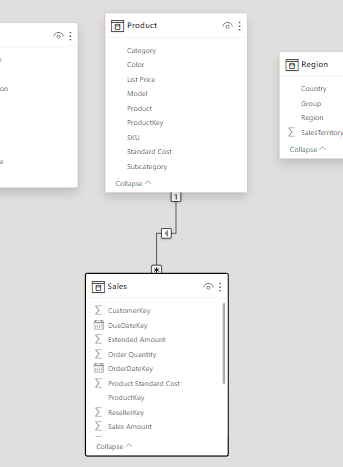
   Description automatically generatedGrab the sales table and drag it down. So you can see all the tables that are involved inside the data model.
5. We have **customer reseller**, **product region**, **sales order** and **sales**.
6. There are currently no defined relationships between the tables. This is our task. You are going to manually create the relationships instead of using the auto detect feature.
7. First select the **manage relationship** button on the ribbon.
8. This screen currently shows all of our relationships between the tables. And of course, we don't have any yet because we're going to manually.

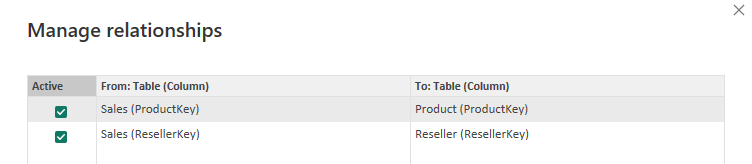
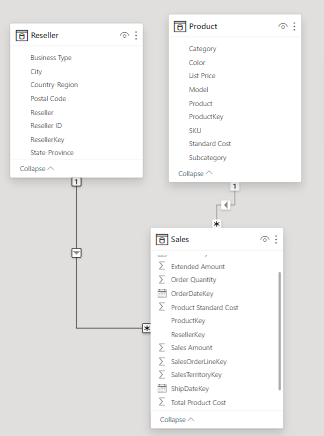
A screenshot of a computer

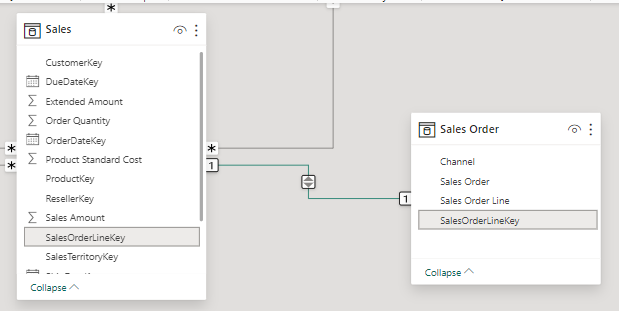
Description automatically generated

1. **Do not** select the **auto detect button**, select the **new button**.
2. And for the table, we're going to select **product**. And for the second table, we are going to select **sales**
3. next, confirm their **cardinality** and **cross filter direction**.
4. The cardinality has **one to many** selected. This is correct. The product is used in many sales.
5. The **single filter** direction means that the filter propagates from the one side to the mini side. In this case, it means filters applied to the **product** table will **propagate** to the **sales** table but not in the opposite direction.

1. A screenshot of a computer

   Description automatically generatedSelect the **OK** button, you'll now see in the list our newly created **relationship**.
2. If you need to **edit**, you can always select the edit button.
3. Now select **close**.
4. You will now see a **line** from the **product** table to the **sales** table. The product shows a one with the line including an arrow, which is the cross-filter direction to the sales table that shows the star, which is the many sideof the cardinality.
5. Next, you will create the same type of relationship using the **drag and drop capability** by dragging a field with the same name from one table to another.
6. The relationship will automatically be defined.
7. At the **reseller** table, drag and drop the **reseller key** from the reseller table onto the **sales reseller key** field.
8. Now the line appears to show the relationship was created.
9. You can check with the **managed relationships button** on the ribbon to see that we now have two relationships.



1. Next, create a relationship from **region** to **sales** by using the **sales** **territory key** and inside sales, we have the **sales territory key** too.
2. Define a relationship from the **customer**, **customer key** to the sales **customer key**.
3. Define the relationship between **sales order**, and **sales**, using field **sales order line key**.
4. The line between the two tables show a **1 to 1 relationship**.
5. when creating the relationship. Power bi I detected a 1 to 1 relationship. This would also set the **cross filter** to **both**.
6. You need to explore the **data** to confirm this is correct.
7. Select the table view button on the left menu. Then select the **sales** table. We can now view the data with inside the **sales** table.
8. A screenshot of a computer

   Description automatically generatedNow sort the **sales order line key** ascending by selecting the arrow next to the column name.
9. Do we see any **duplicate numbers**?
10. I don't, this table seems to have **unique numbers** that explains the one side of the relationship.
11. Now on the **table view**, select the **sales order** table, let's go ahead and, and sort the **sales order line key** column.
12. Once again, let's take a look for any duplicate numbers. The table seems to have **unique numbers**.
13. So once again, this explains why in the relationship we had a one on that side.
14. So, everything looks good. Let's select our model view on the left menu and you have now defined the base relationships between all the tables. You are ready for the next task.

## A screenshot of a computer Description automatically generatedStep 2: Configure Tables

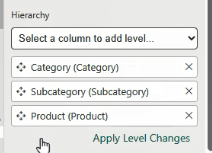
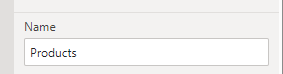
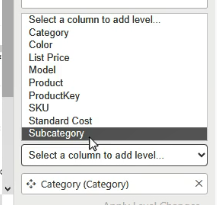
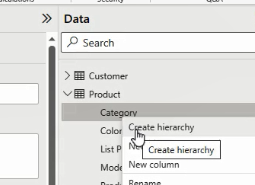
1. Once you have defined the basic table relationships, your next task is to configure table specific properties.
2. This includes **field formatting** and **field groupings.**
3. Select the model view and expand the **properties** and **data** panes.

### Configure Product table

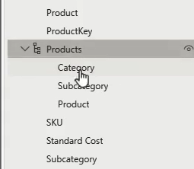
1. Select the **product** table.

### Creating Hierarchy

1. You'll be creating a **hierarchy** on **category subcategory** and **product**:
2. Right Click on the **category** column and select **create hierarchy.**
3. In the **properties** **pane.**
   1. For the **name**, enter **products**,
   2. Scroll to the bottom, you'll see an area to build a **field list**.
   3. select a column to add: select **subcategory**. And then select **product**.
   4. The next most important thing to do, select, **apply level changes**.

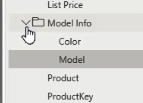
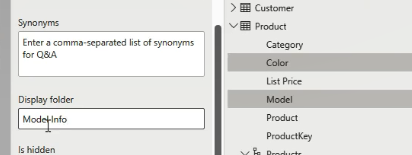


1. You can now look at the hierarchy inside the data pane.
2. You have products categories, subcategory and then the product.



### Organize Columns into Display Folder

1. Next, we will organize columns into a display folder.
2. select the **model** field and **products** and while holding the **control key**, select the **color** field.

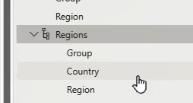


1. In the **properties** pane and the **display folder** field enter “**model info”**.
2. If you leave that field, it will save and under the folder, we now have the model info, we have the color and the model.

### Configure Region Table

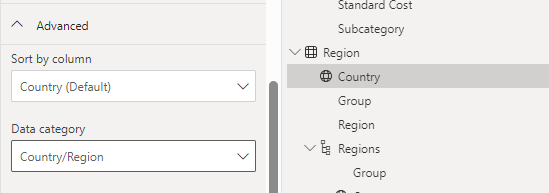
### Create Hierarchy Regions

1. Next, you need to configure the region table,
2. Select the **region** table.
3. Create a **hierarchy** on **group** **country** and **region**.
4. Don't forget to select **apply level changes** and you should see our regions listed in the table.



### Data Category

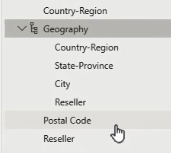
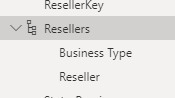
1. Next, select the **country** field **that is not part of our hierarchy**.
2. And then with inside the **properties**, scroll down to the **advanced** options and change the **data category** to “**country region**”.



### Configure Reseller Table

### Hierarchies:

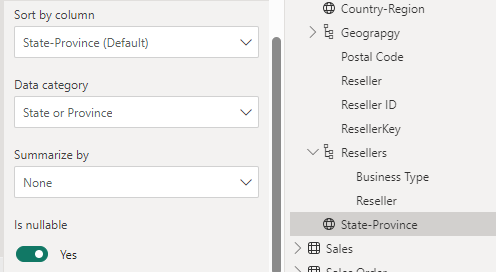
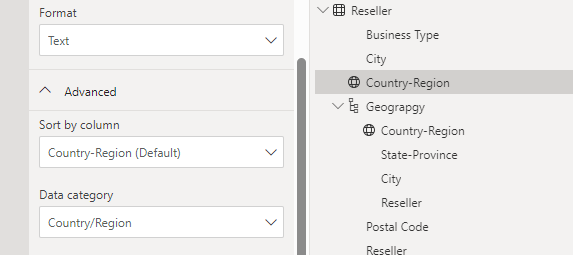
1. Next, we will configure the **reseller** table,
2. Select the reseller table.
3. Create a **hierarchy** on **business type** and **reseller** columns.
4. For the name type **resellers**.
5. apply level changes. And once again, you should see the hierarchy in the table.



1. Create a second hierarchy in the reseller table. This time on **country region**, **state**, **province**, **city** and **reseller** columns.
2. For name type “**geography”.**
3. **Apply level changes** and you should see **geography** and the reseller table.

### Data Category

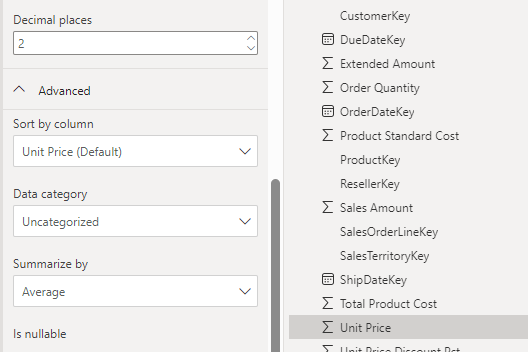
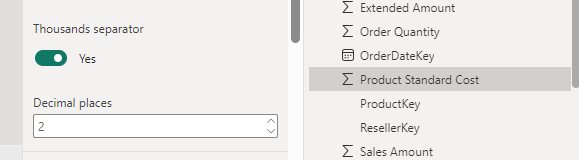
1. Next we'll set the **data category** for **country region, state province** and **city**
2. select the **country** region field that's **not part of** the **geography** **hierarchy**.
3. And then in our properties select **advanced** and for **data category**, select **country region**.



1. Next select **state province**, which is not part of our geography hierarchy. And for the data category, select **state** **or province**
2. and then our final field will be **city** once again, not part of our hierarchy for data category. Configure as **city**.

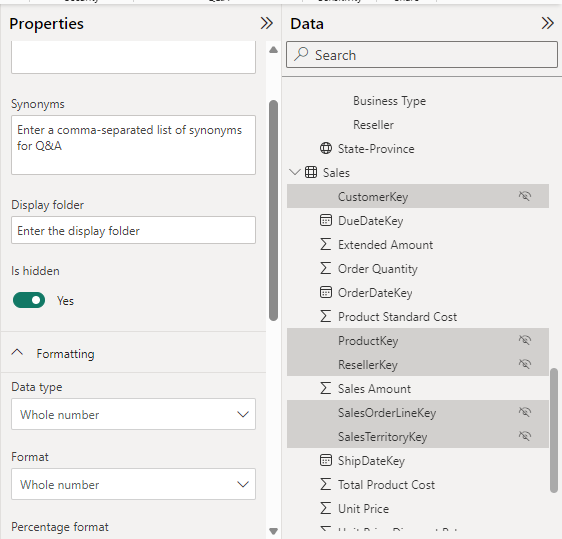
### Configure Sales Table

1. Select the sales table.
2. select the **product** **standard cost** column on the **properties** tab slide the 1000 separator to **yes**.
3. Now select the **unit price** column and inside properties set the **decimal places** to 2.
4. In the **advanced** section set, the **summarize b**y to **average**.



### Bulck Update Columns

1. as you could see if you had to individually update every column for the same setting. This could take a while.
2. Fortunately, you can **bulk update columns**.
3. For example, you can hide all the **keys** because they provide no other value than linking the tables together.
4. To make this a little bit easier, expand all of the tables just so we can see.
5. select the **customer key**. Then while holding down the **control key**,select the **product key**, **the sales territory key**, continue holding down the control key and the **reseller key**,then the **sales order line key** with inside **sales** table and then also the **sales territory key** with inside sales table, the **product key**, the **reseller key** and the **customer keys**.
6. Finally, down in **sales order** table, we should have our **sales order line key**.
7. In **properties** pane we will select **is hidden to Yes**.



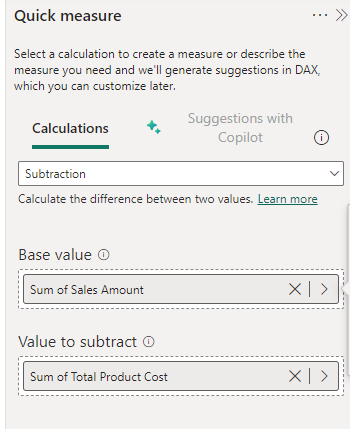
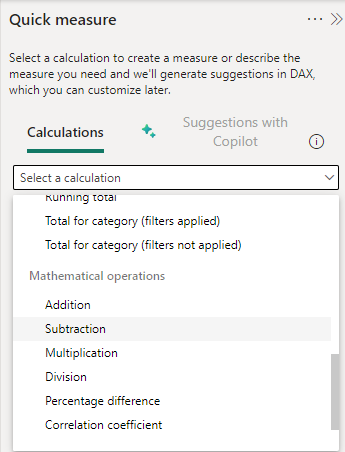
1. Now this has updated all of those columns to be hidden.
2. You are ready for the next task but remember to also save your file.

## Step 3: Create Quick Measures

1. To quickly create calculated fields without the need for writing DAX calculations, you can create quick measures.
2. in this task, you will create quick measures to enhance the data.
3. Select the **table view**,
4. select the **sales** table and expand it.
5. Right Click **sales** table, you'll see an option for **new quick measure**.
6. New quick measure is also available to you with inside the **tables tools**, ribbon menu.

### Quick Subtraction Measure

1. Select **subtraction** for the **base value**



1. drag and dropped the **sales amount** field from the sales table
2. for the **value to subtract** drag and drop the **total product cost** field from the sales table.
3. Then select **add.**
4. As you can see the measure has been added to our sales table.
5. Looks like it has a very long name. Let's rename this measure to **profit**.
6. So right mouse on the column, select **rename** and type in **profit**.
7. Notice the DAX code in the function bar for the new measure

Profit =

SUM('Sales'[Sales Amount]) - SUM('Sales'[Total Product Cost])

### Quick Division Measure

1. Our calculation is going to be division.
2. So select **division** for the **mathematic operations**
3. for the **numerator** drag and dropped the **profit** field from the **sales** table and for the **denominator** drag and dropped the **sales of Amount** field.
4. Then select **add.**
5. as you can see our new measure has been created with a very long descriptive name. rename it to **profit margin**.
6. Notice the DAX code of the new quick measure

Profit Margin =

DIVIDE([Profit], SUM('Sales'[Sales Amount]))

### Format measures

1. Now that we have the **profit margin** column, we need to format the field.
2. So, in the ribbon, we have the formatting section set the format to **percentage** and confirm that the decimals are 2.

### Test The Measure

1. test our quick measures. Select the report icon on the left menu. This is the report view area.
2. double click on **table** that now drops a table into our canvas
3. from the sales table. Select the check box for **profit**, **profit** **margin** and **sales amount.**
4. A close-up of a number

   Description automatically generatedThis shows an overall total for the sales table.
5. I think we need to see a little bit more detail.
6. select the **product** table and then select the check box for **category**.
7. Now we have the profit, profit margin sales amount for each category. A screenshot of a graph

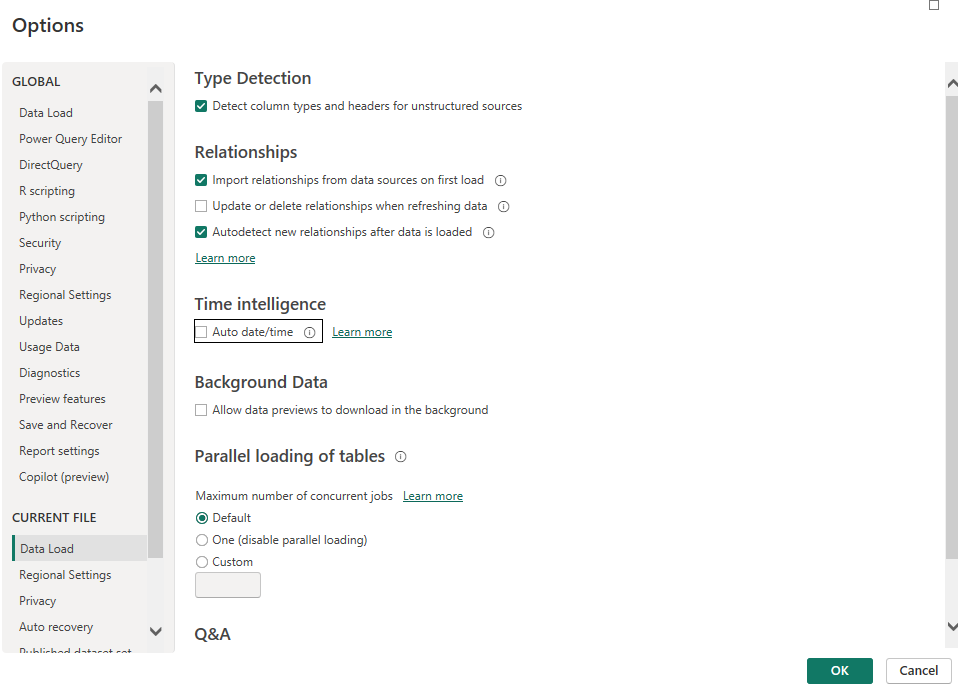
   Description automatically generated
8. You are ready for the next task. Don't forget to press the save button first.

## Step 4: Create a Date Dimension Table

1. Common analysis scenarios require date related groupings.
2. An example of this is a custom **Fiscal year** instead of using a **calendar year**.
3. In this task, you will create a date table for use in your data.

### Turn off the system auto date functionality

1. From the File select **file**🡺 **options and settings** 🡺 **options**
2. in the section, **current file**, select **data load**, then **uncheck** the checkbox **auto detect time** and then select **OK**.



### Create Date Table Using DAX

1. Select the **table view** from the left menu.
2. From the ribbon, select **new table**
3. in the formula bar, you will enter a DAX calculation.
4. Enter the following calculation:

Date = CALENDARAUTO(6)

1. Press Enter the function generated a new table named **Date.**
2. The (**6**) is an extra parameter that is used to define **the last month of the year**. In this case, **June**.
3. by default, it is **12** or **December.**
4. take a minute to review the data that was generated.
5. Format the Date Column as **Date** only

### Add More Columns

1. Now you need to add in a few extra columns to enable filtering and grouping for different time periods.

### Create Year Column

1. On the **table tools** ribbon select **new column**.
2. You will see that the formula bar has a new formula for us to enter in the calculation for the column. Enter the formula:

A screenshot of a computer

Description automatically generatedYear = "FY" & YEAR('Date'[Date]) +   
IF(MONTH('Date'[Date]) > 6 , 1)

### Create Quarter Column

1. Create the Quarter Column using this formula:

Quarter = 'Date'[Year] & " Q" &   
 IF(MONTH('Date'[Date]) <=3,3,

IF(MONTH('Date'[Date]) <=6 ,4,

A screenshot of a computer

Description automatically generated IF(MONTH('Date'[Date]) <=9,1,2)))

### Create a Month Column

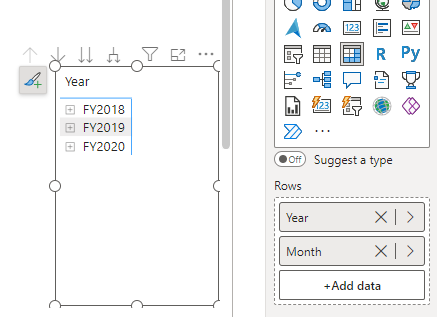
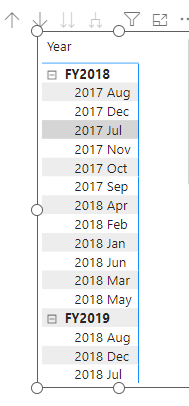
1. Create a Month Column with the code

A screenshot of a computer

Description automatically generatedMonth = FORMAT('Date'[Date] , "yyyy MMM")

### Test Your Date Table

1. Go to Report view.
2. Create new Page.
3. Add a **Matrix**.
4. Add **Year** and **Month** Field to **Rows**.
5. Click on the Fork Icon to expand the next level (Months)



1. Notice the Month is not sorted well.
2. Let us fix that

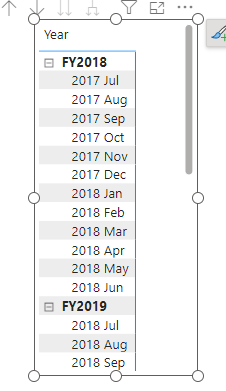
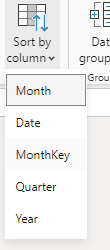
### Create MonthKey column

1. Go to **table view**.
2. Select **Date** table.
3. Create new Column **MonthKey** to filter the date on
4. Use the following DAX code:

MonthKey = YEAR('Date'[Date]) \*100 + MONTH('Date'[Date])

1. A screenshot of a computer

   Description automatically generatedNow Select the **Month** column



1. From the ribbon select **Sort by Column**.
2. Select **MonthKey**.
3. Go to Matrix now and it should be updated.
4. Now go to Model View
5. Select **MonthKey** and Hide
6. Select **is Hidden** and make it **Yes** .

### Create Date Hierarchy

1. A screenshot of a computer

   Description automatically generatedSelect **Year** column and right click and select **Create New Hierarchy.**
2. Make the **Name** : **Fiscal.**
3. Add **Quarter** then **Month**
4. Click Apply Level Changes

### Set Relationship

1. Now set the relationship between Date table and Sales table.
2. A screenshot of a computer

   Description automatically generatedUse **Date** field from Date table and **OrderDateKey** from Sales table.

### Mark As a Date Table

1. Go to table view.
2. From the ribbon select **Mark as a Date Table**.
3. In the Dialogue box make it **on**.
4. Select **Date** field.
5. Save.

### A screenshot of a computer Description automatically generatedCreate Some Measure

1. Select Sales table and create some measures.
2. **Avg Price**:

Avg Price = AVERAGE(Sales[Unit Price])

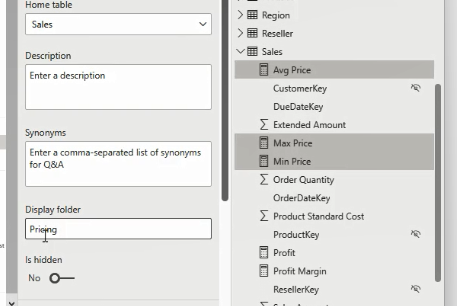
1. **Min Price:**

Min Price = MIN(Sales[Unit Price])

1. **Max Price:**

Max Price = MAX(Sales[Unit Price])

1. Go to Model view.
2. Arrange those measures to a Display Folder : **Pricing**



### Display Measure in Matrix

1. Go to your **Report View**.
2. A screenshot of a computer screen

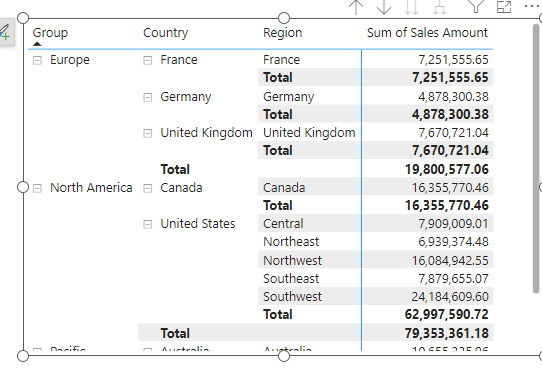
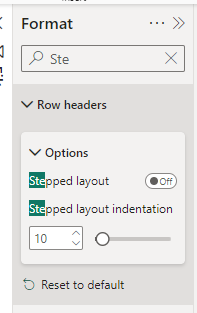
   Description automatically generatedSelect your **Matrix**.
3. Chek the measures **Avg Price**, **Min Price** , and **Max Price** to add to matrix.
4. Save your File.

## Step 5: Create DAX Calculation

1. In this task you will create advanced DAX calculations to solve some business problems.
2. Go to **Report View**.
3. Create New Page.
4. Add a **Matrix** to report.
5. A screenshot of a computer

   Description automatically generatedFrom **Region** table drag the **Regions** Hierarchy to Rows.
6. From **Sales** table Check **Sales Amount** to Add in **Values**.
7. A screenshot of a computer

   Description automatically generatedSelect the **Fork** icon twice to expand the hierarchy.
8. In **format** pane search for **Stepped Layout** and make it **off**



### Create Sales % All Region Measure

1. Select Sales table and create new measure:

Sales All Regions = CALCULATE(SUM(Sales[Sales Amount]) , REMOVEFILTERS(Region))

1. A screenshot of a data

   Description automatically generatedSelect **Sales All Region** measure to add to matrix.
2. Edit the measure to become **Sales % All Region** measure as follow:
3. Format as **Percentage** with **2** decimals.
4. A screenshot of a graph

   Description automatically generatedSee your matrix now

### Create Sales YTD Measure

1. Go back to matrix of years in Page 2
2. Create new measure:

Sales YTD = TOTALYTD(SUM(Sales[Sales Amount]) , 'Date'[Date],"30-6")

1. Add to your matrix.

### A screenshot of a data table Description automatically generatedCreate Sales YoY Growth

1. Create the measure

Sales YoY Growth =

    Var TotalSales =SUM(Sales[Sales Amount])

    var SalesPriorYear =

        CALCULATE(SUM(Sales[Sales Amount]) ,

        PARALLELPERIOD('Date'[Date],-12,MONTH))

    RETURN

        DIVIDE((TotalSales -SalesPriorYear), SalesPriorYear)

1. Format as Percentage with 2 decimals
2. A screenshot of a graph

   Description automatically generatedAdd to your matrix.
3. Add your two measures to a **Display folder** and name it **Time Intelligence**