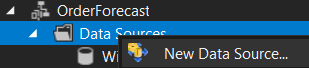
The Order Manager wants to have a report on order forecast in orders

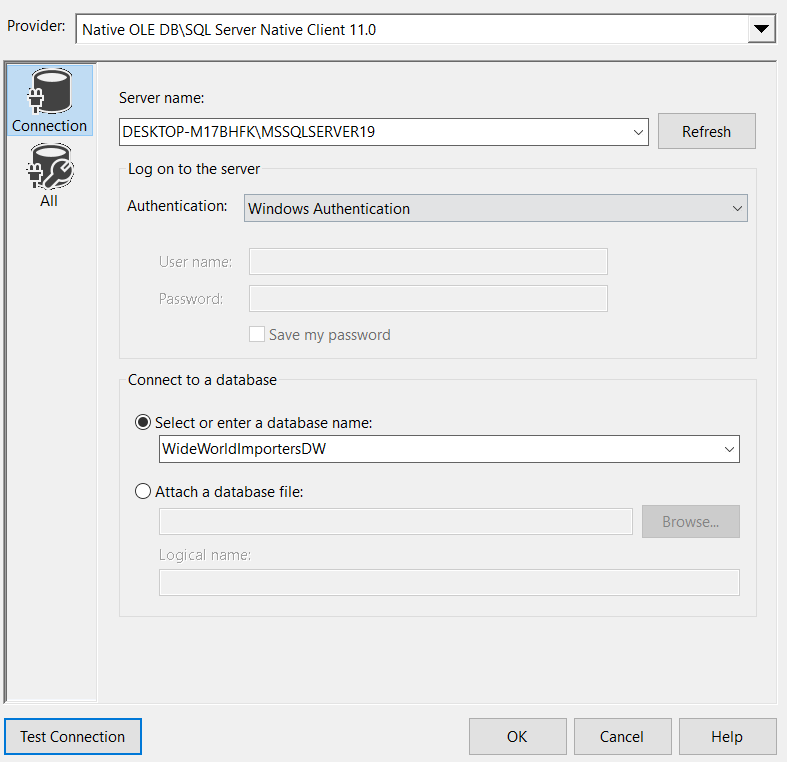
1. Firstly, we will create a new Analysis Services multidimensional project and then add a data source to it



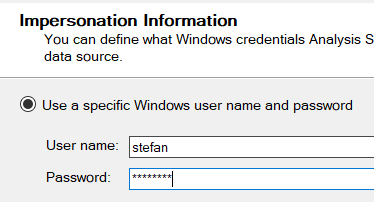
1. We will need to create a new connection to a database



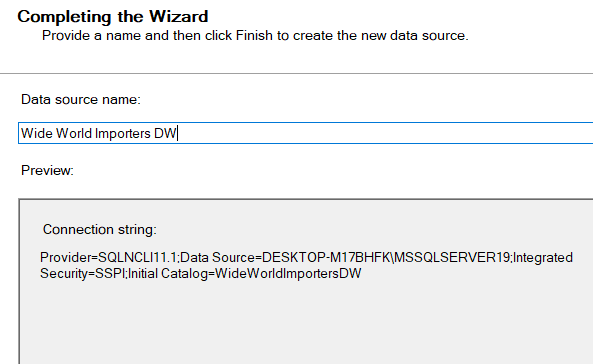
In the new connection dialog, we need to provide the name of the SQL Server our database is on, specify using Windows Authentication if it isn’t by default and select WideWorldImportersDW as the database we want as data source. After using the Test Connection button to make sure that the information was provided is ok, we can click ok.



1. Next we need to provide a valid Windows Authentication login information in the Impersonation Information dialog in order for SSDT to be able to successfully connect to our SQL Server



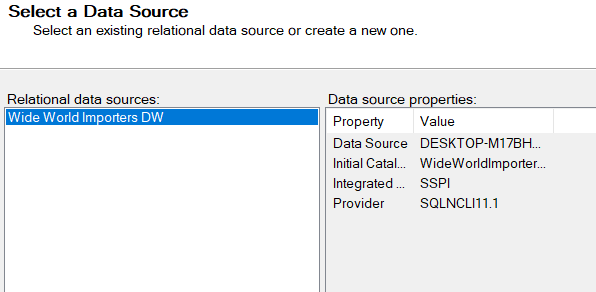
1. Then we can provide a name for our data source, click on finish and have it created



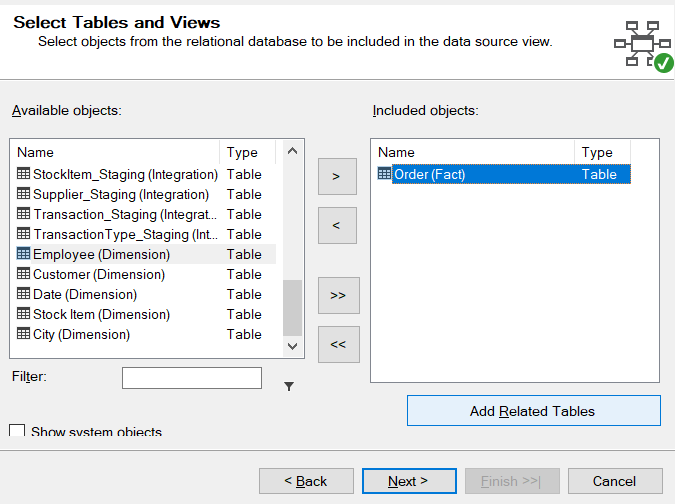
1. Now we need to create a **data source view**



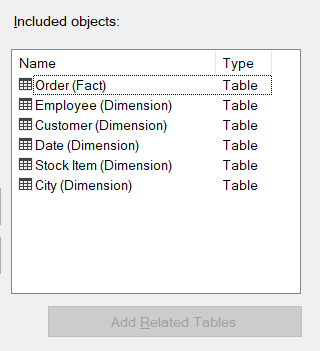
We base our view on the data source we created earlier



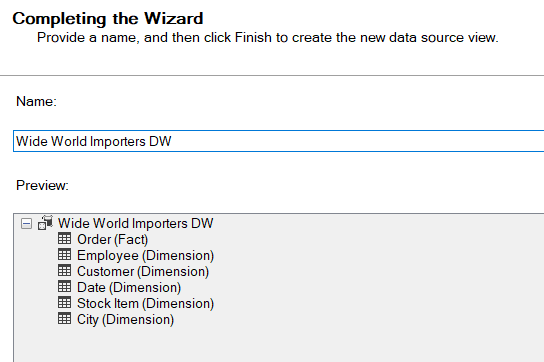
1. We select Order as fact and then add the related tables



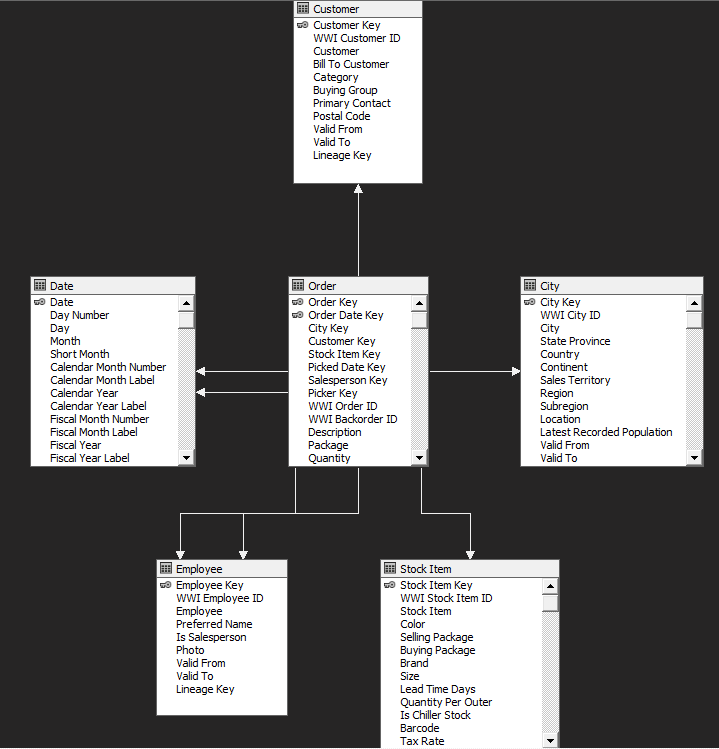
This is how the Included objects window should look like



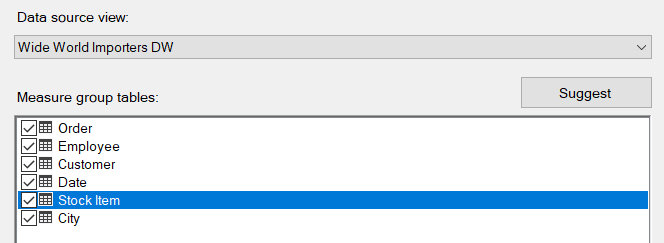
1. Then we can name our data view and click finish to create it



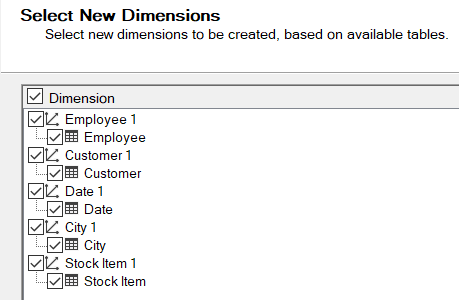
1. This is the diagram of the data view



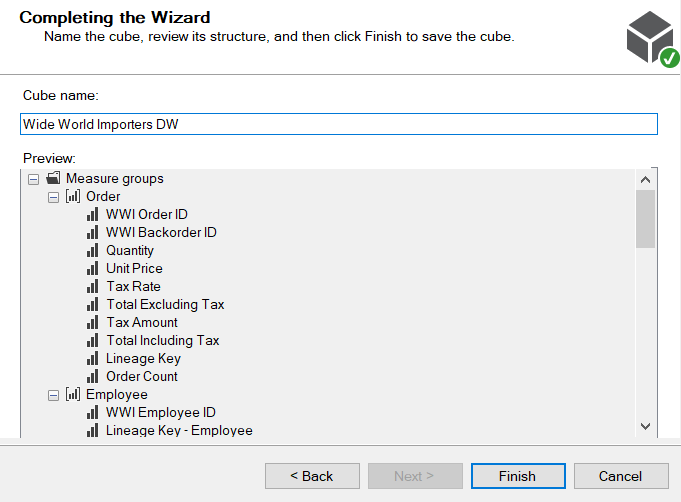
1. Now we can get to creating the **cube**. We will use the data source view created earlier



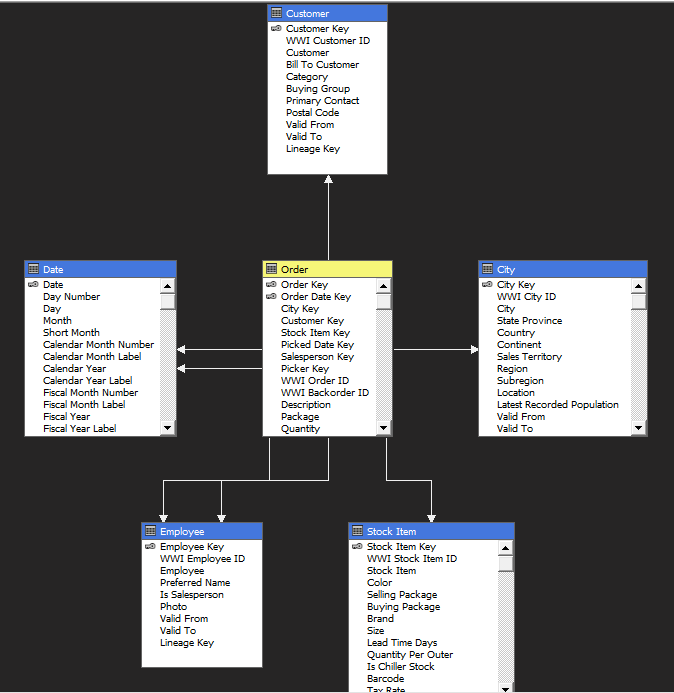
We will end up with these new dimensions to be created



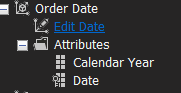
1. Now we can finish creating the cube



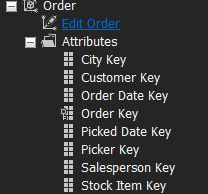
This is the cube structure we will have



1. Now we will **add attributes to the dimensions**

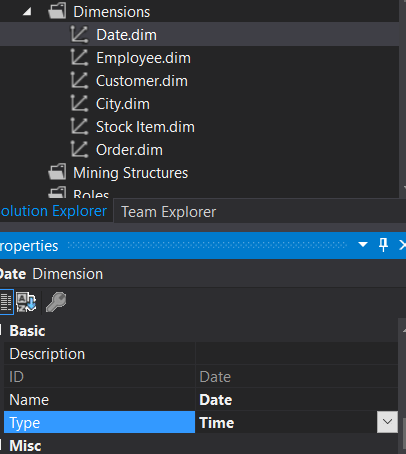


We will use Calendar Year for Order Date so we can draw insights on the forecast of orders by years as the predictions can get less accurate with more specific time periods.

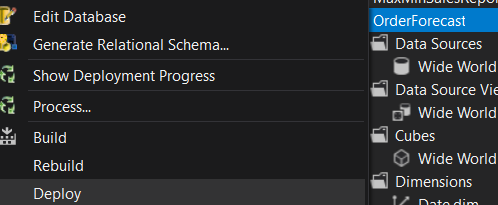


For Order what we will need is Order Count, Quantity and Total Including Tax

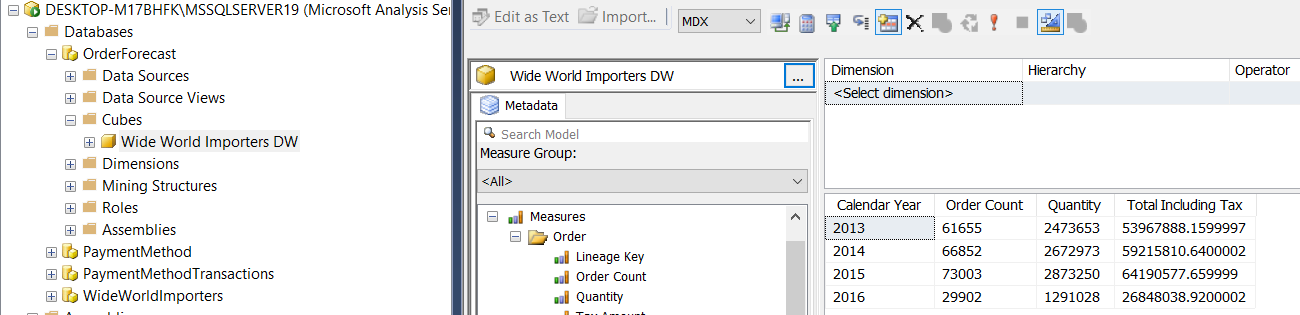
1. Next we should also **set** the Date dimension’s **type** to Time



1. Now the cube is ready to be **deployed to the Analysis Services SQL** server

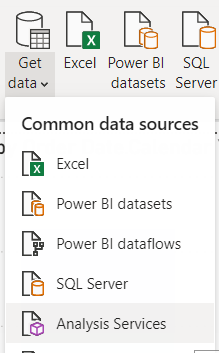


After deploying we can check in SSMS and see that the cube is there and can be **browsed**.

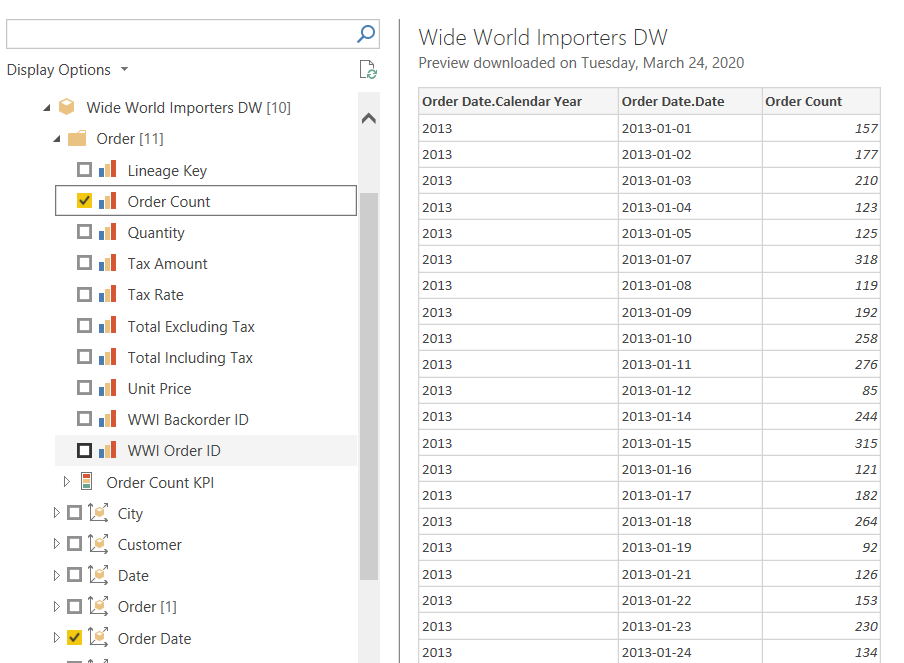


Now we can use the year, order count, quantity and total including tax to see the amount of orders over the years and link them with the quantity of produce ordered and also the total income for that year.

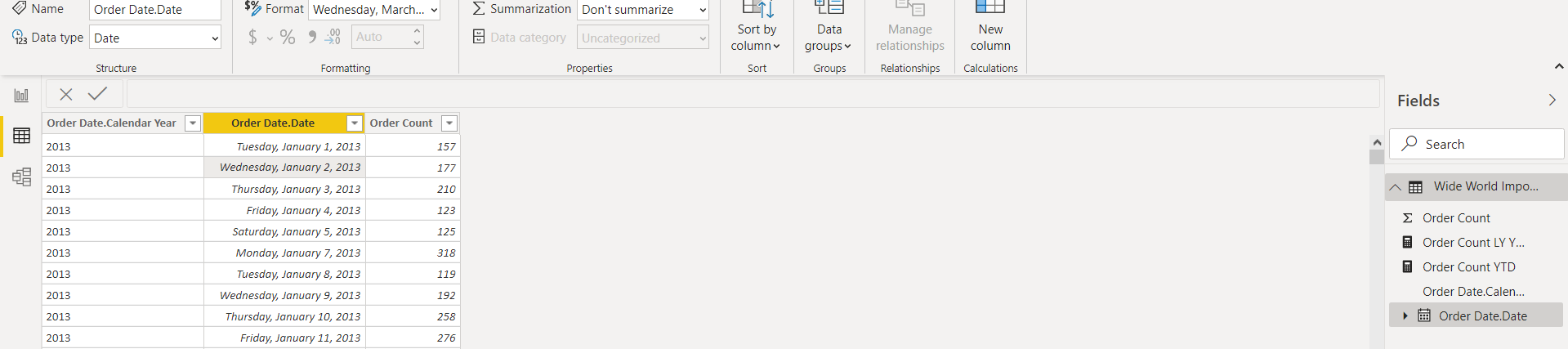
1. Next step is the **PowerBI report**



We need to **import the data** from the analysis services server and select what data to use

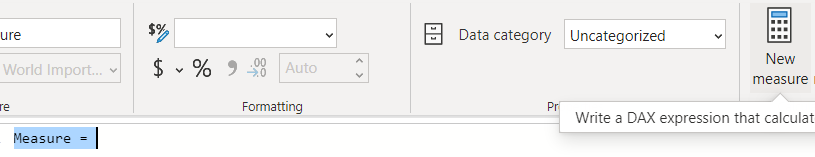


The data should be like this

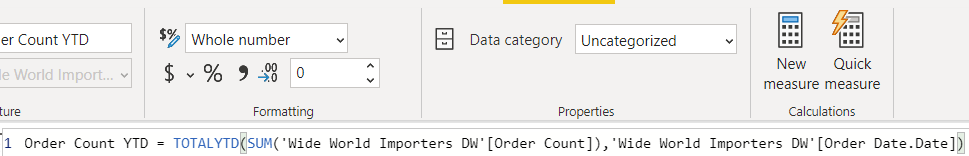


If the Date doesn’t show up as a Date data type, we can format it here

Now we can create some helper **measures**. We will make an *Order Count* **year-to-date** (Order Count YTD) and an *Order Count* **last year year-to-date** (Order Count LY YTD)



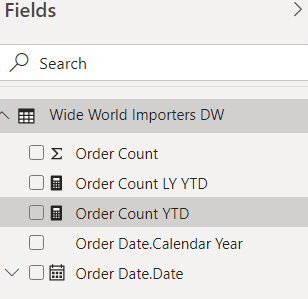
We **click new measure** and then we enter the following **DAX formula**: Order Count YTD = TOTALYTD(SUM('Wide World Importers DW'[Order Count]),'Wide World Importers DW'[Order Date.Date])



For the last year we do the same but with the following formula: Order Count LY YTD = CALCULATE(SUM('Wide World Importers DW'[Order Count]),PARALLELPERIOD(DATESYTD('Wide World Importers DW'[Order Date.Date]),-12,MONTH))

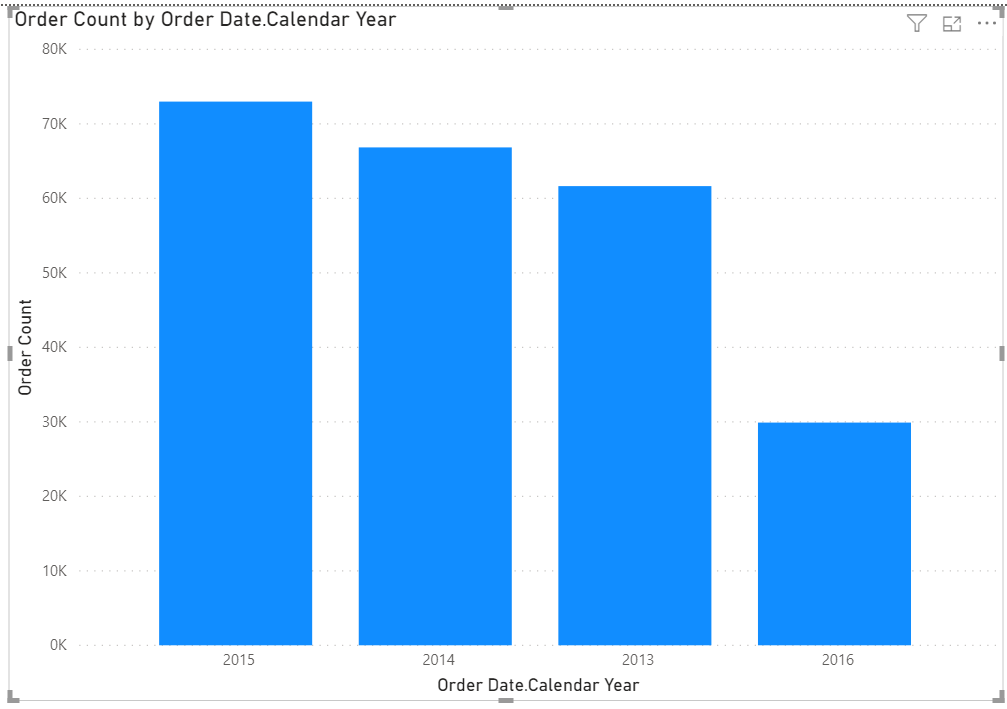


These are the fields we will end up with

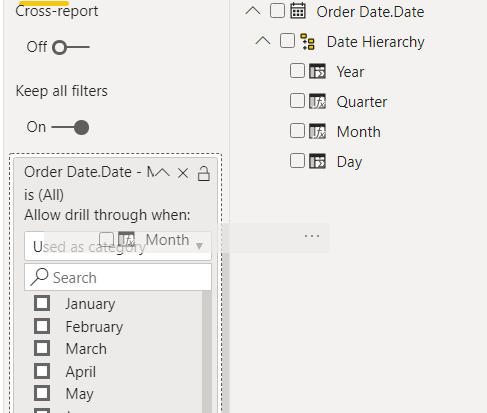


Now we can create some graphical representations of the data.

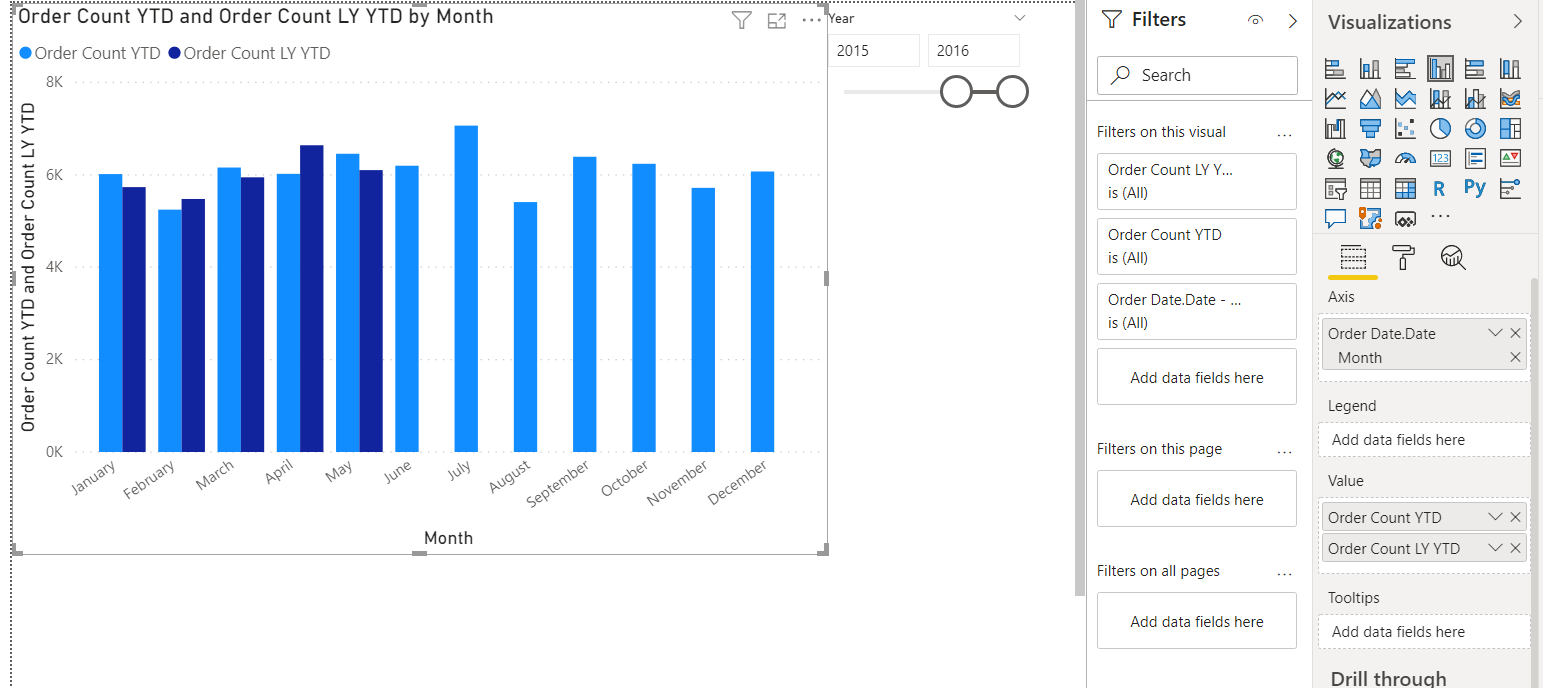
A **bar chart** will allow us to compare the different values of Order Count over the years.



Adding the Month value from the Date hierarchy to the **Drill Through** filter will allow us the see more specific data for a certain month from each year



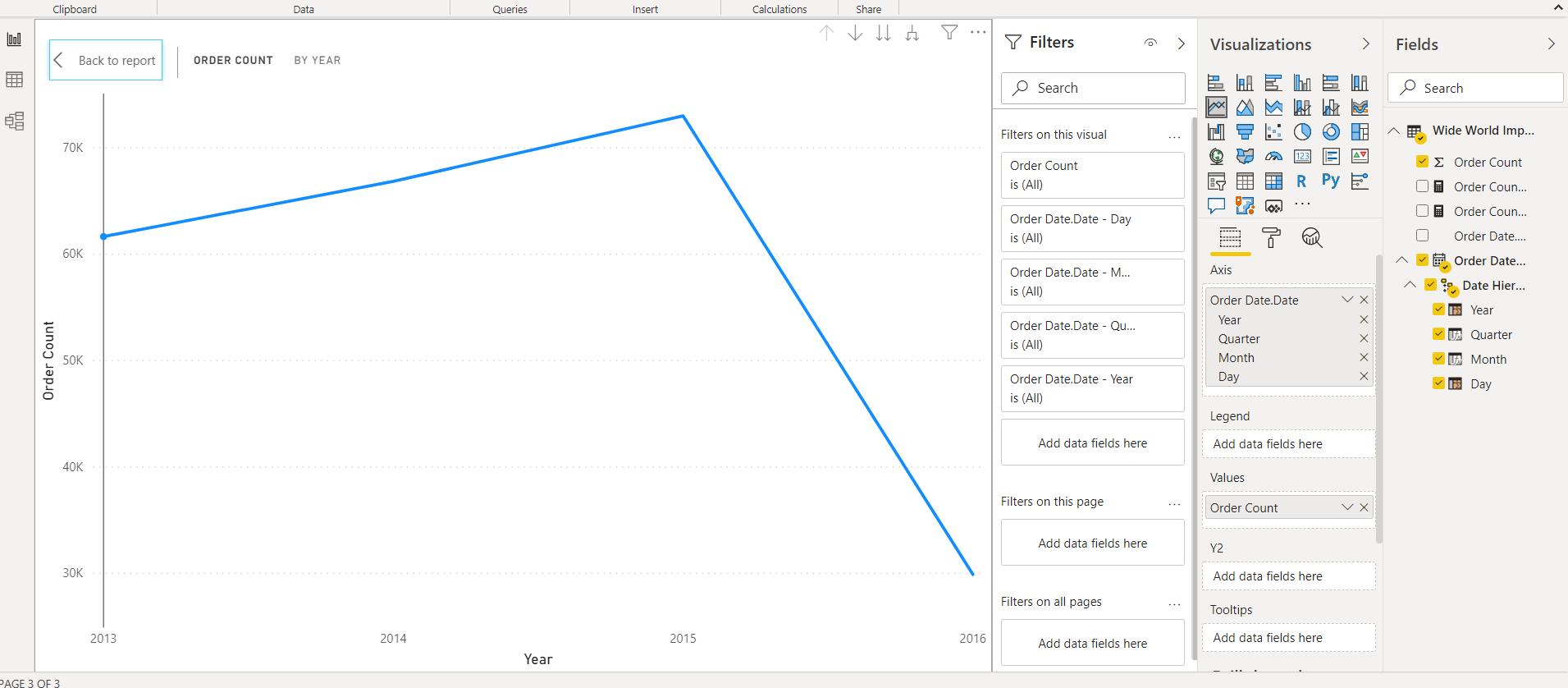
For a better overview of amount of Orders compared to past years we can make another **bar chart** and use the **measures** we just created



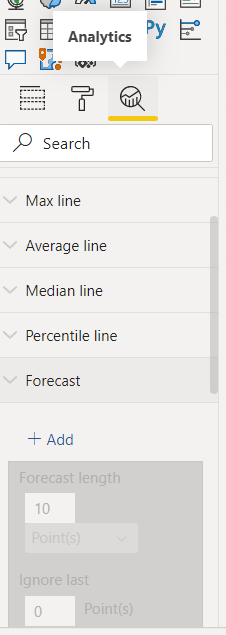
We can **also add a slicer** to limit the range of years the data is taken from

1. Making a **forecast** line chart

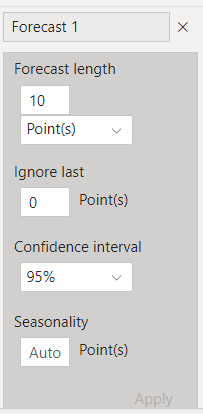
Because we want to see what the prediction for the future is we are going to create a **line chart** and add a forecast to it



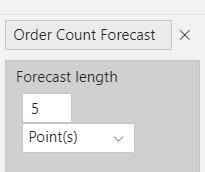
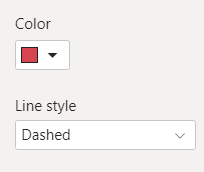
We use *Order Count* as values and the *Order Date* as axis. In order for PowerBI to enable the forecasting feature it is important that the axis field uses a *Date* type. We made sure it is in the above steps and we can also tell from the calendar icon next to *Order Date* in the fields menu



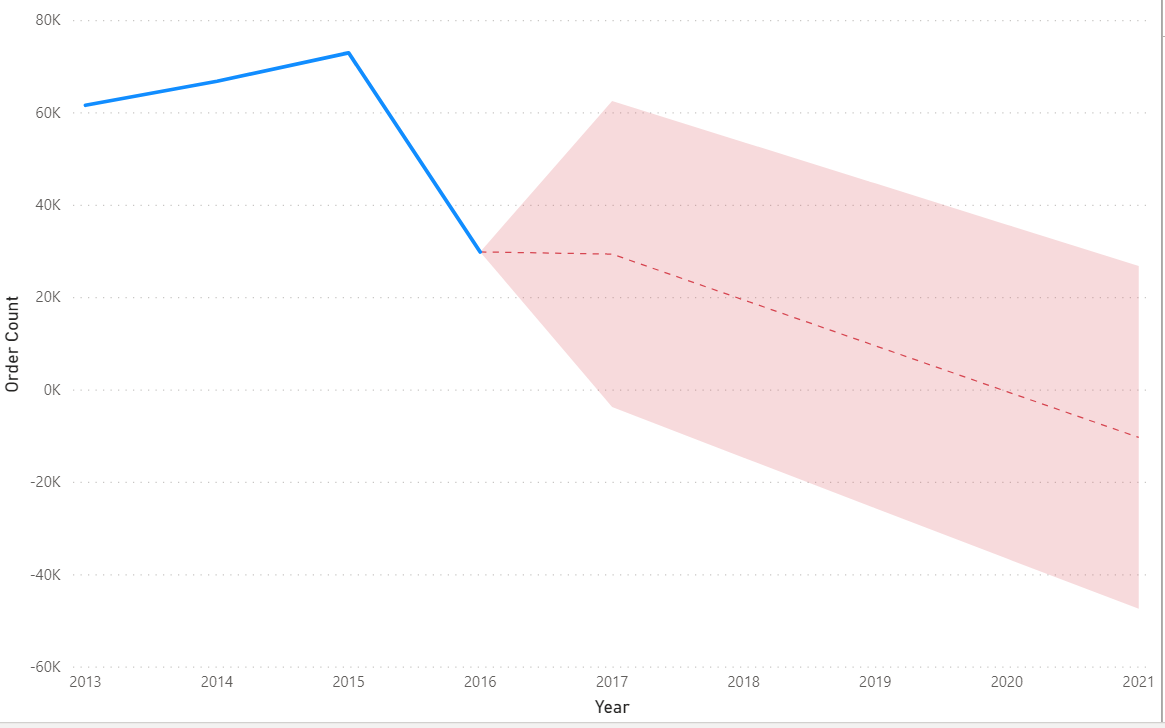
By going into the analytics section of the line chart we can now add a forecast to it



We can now modify this to look how we want it to. Because there is a downwards trend in the forecast we will highlight this with colors

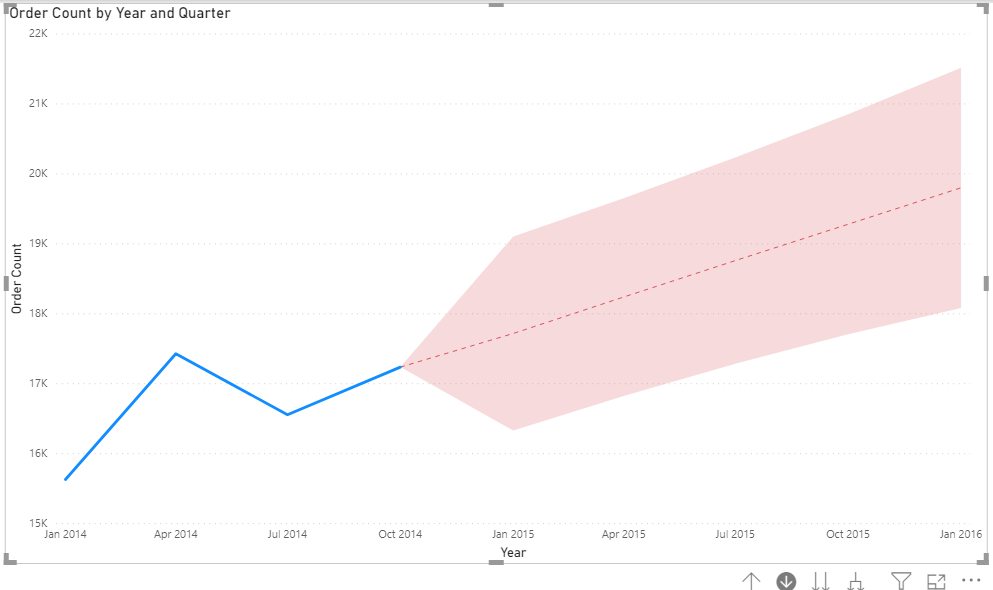
 

The graph will now look like this

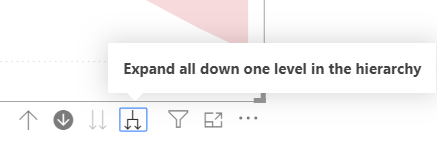


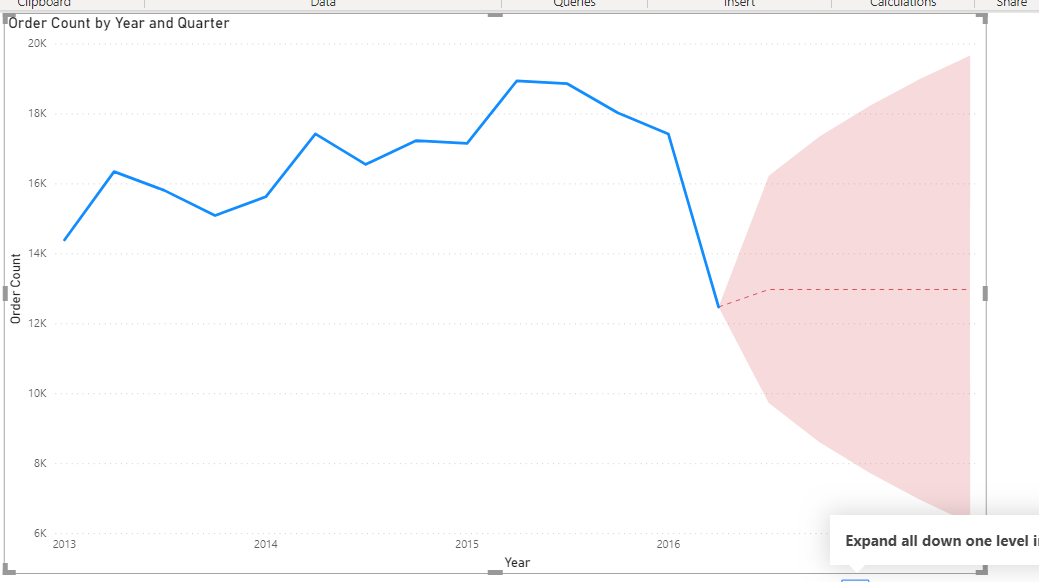
By **enabling drill down** on the report we can also check predictions at **certain data points**



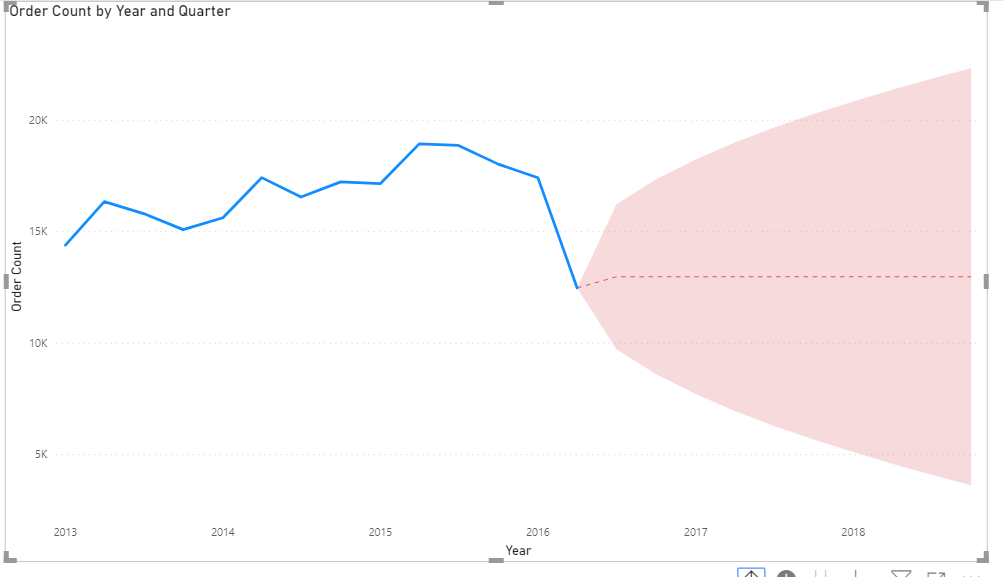


The initial line chart that only shows years is a bit short. That is because it only shows the value of *Order Count* as a sum for every year. We can **expand one level down in the hierarchy** and see the progress from year to year by quarters





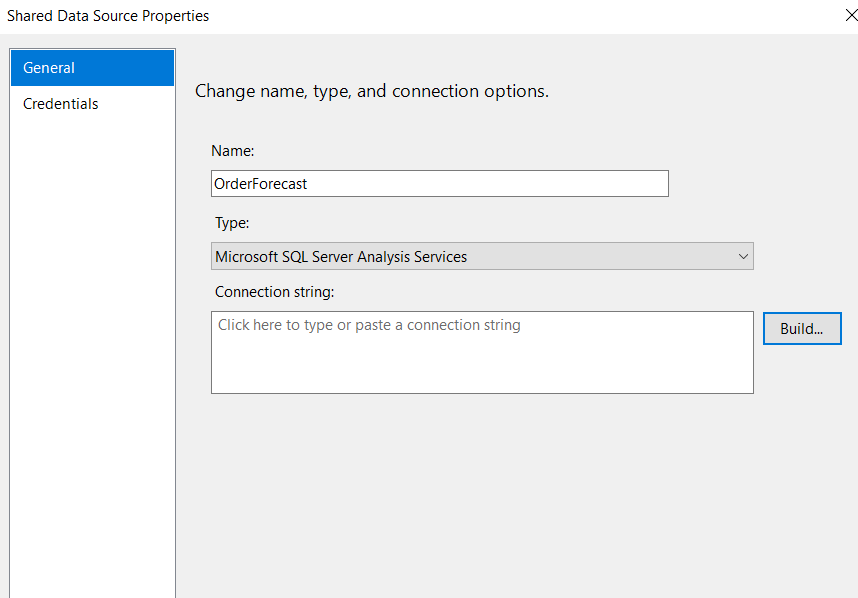
We can also make the forecast longer now



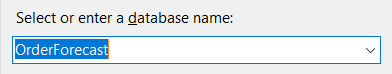
1. Next we will create **reports in SSRS**. After creating the reporting services project we add a data source to it.



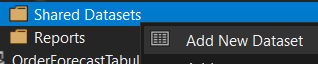
We name the data source *Order Forecast* and then by pressing build we setup the connection to a SQL Server Analysis Services instance.

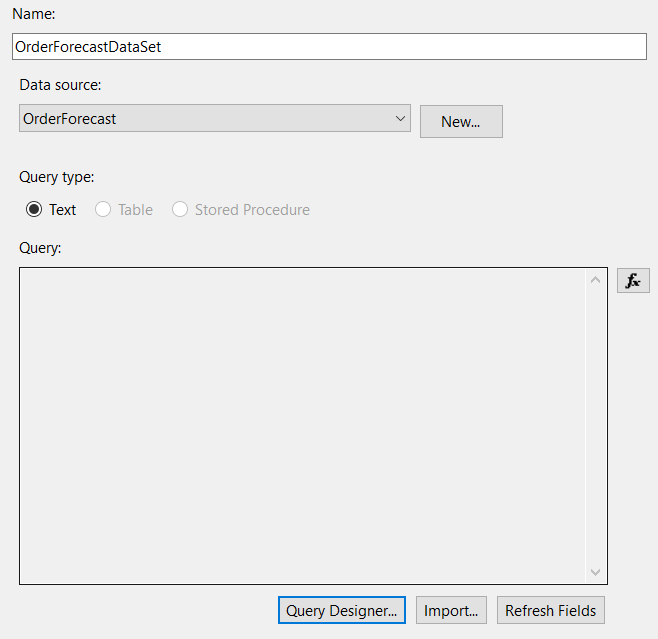


For the database we use *OrderForecast*

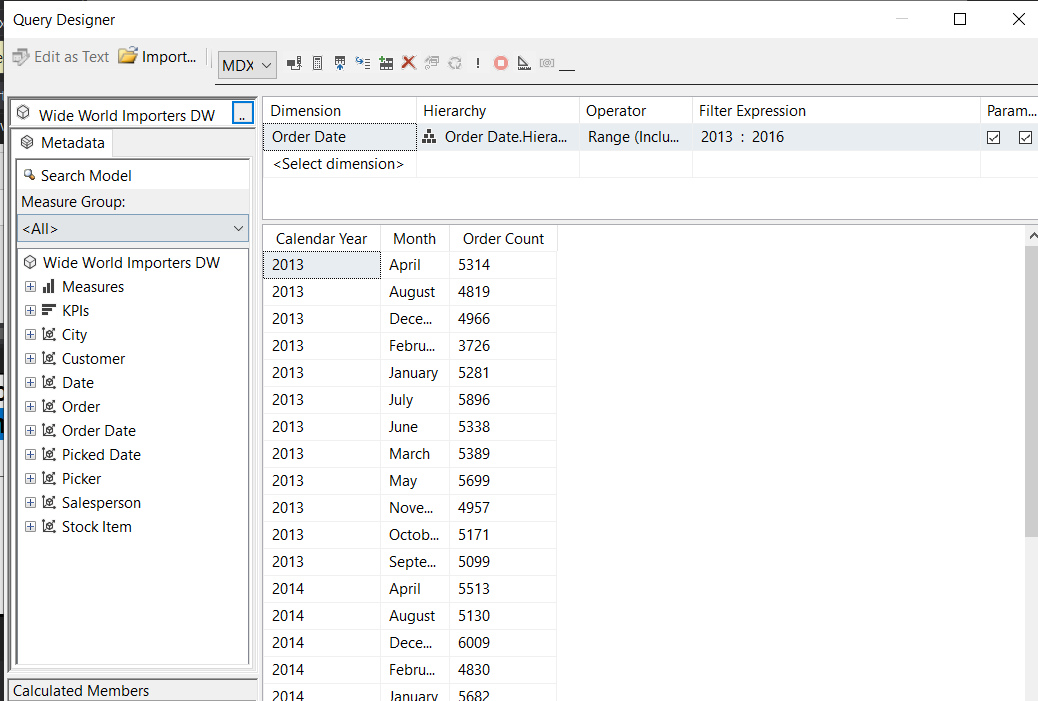


1. The next step is to create a *Shared Data Set*



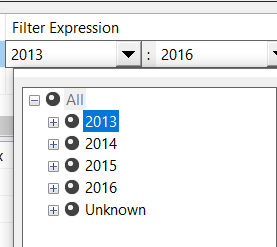


We name it OrderForecastDataSet and then we go to the Query Designer to select what we need from the data source. Then we can go in the query designer and choose the data we want to use



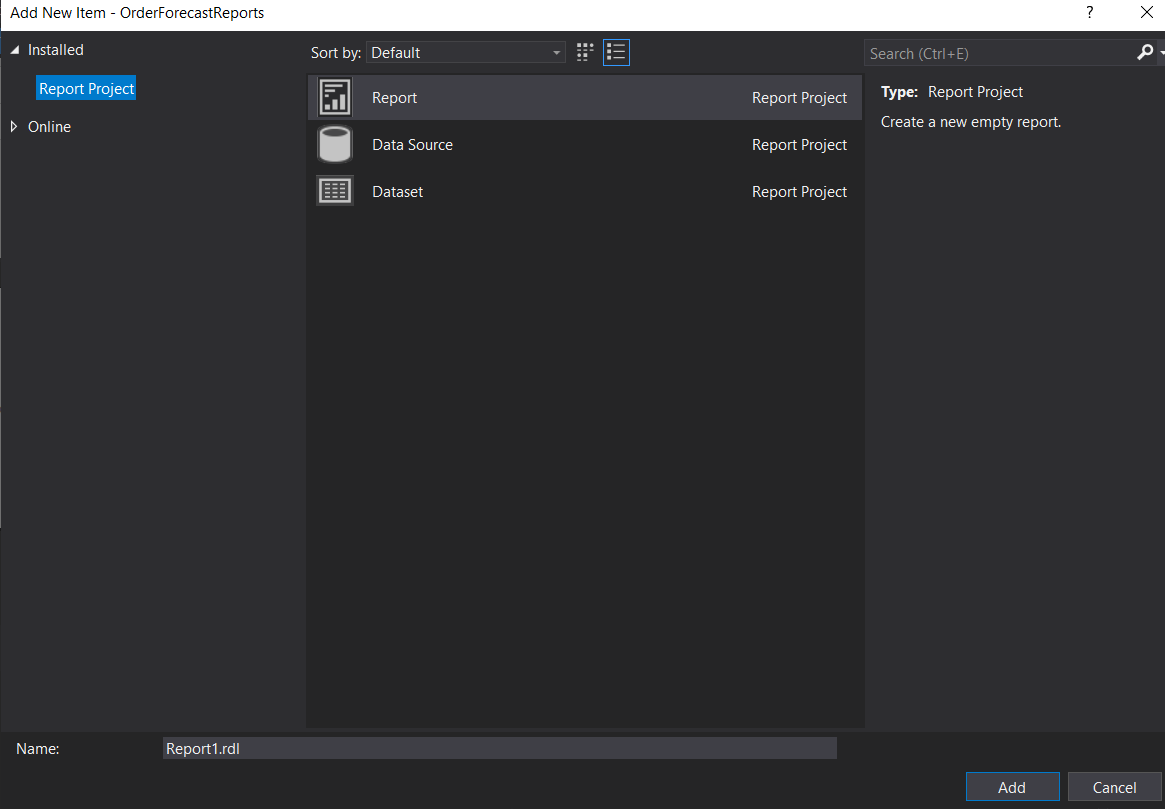
For this we selected the *Calendar Year* and *Month* values from the hierarchy in the *Order Date* dimension and the *Order Count* measure.

The values for the date contain an *Unkown* value and we won’t need that for our report so we create a filter for the values that interest us

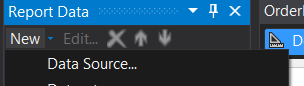


We also mark those values as parameters so that the user can choose for themselves the range of years

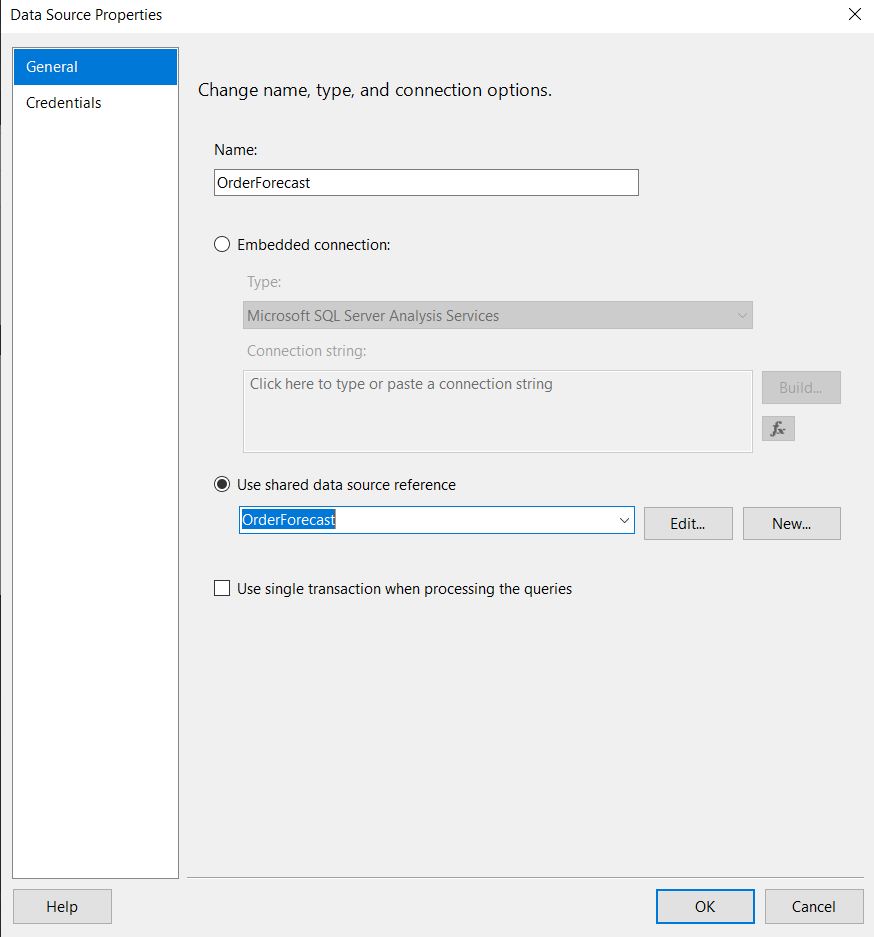
1. Now we can go and **create a report** in the Reports folder



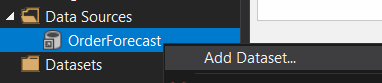
We’ll name it *OrderForecastReport* and then we will add our data source to it



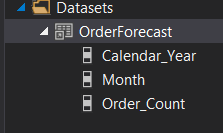
We make sure we choose the shared data source and then we click *Ok*



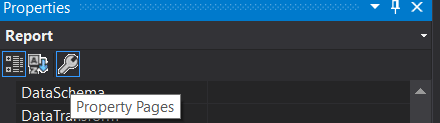
Then we need to add a **dataset** to it



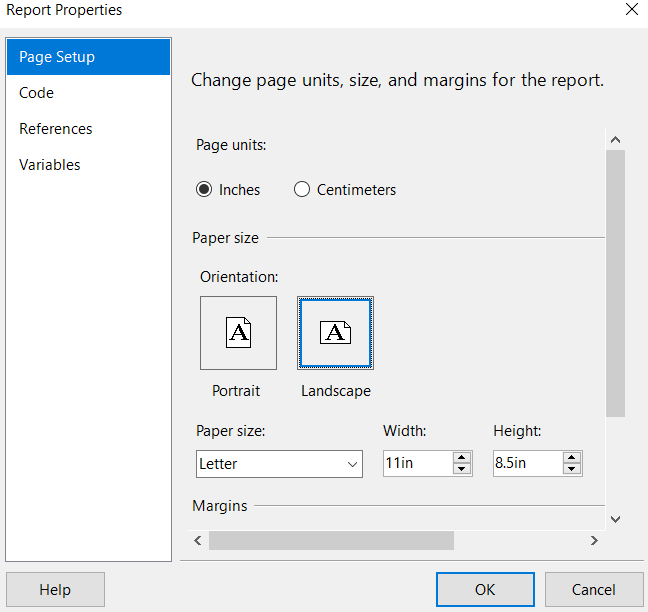
From the dialog we choose use shared data set and then select *OrderForecastDataSet*. After clicking *Ok*, the report structure should look like this



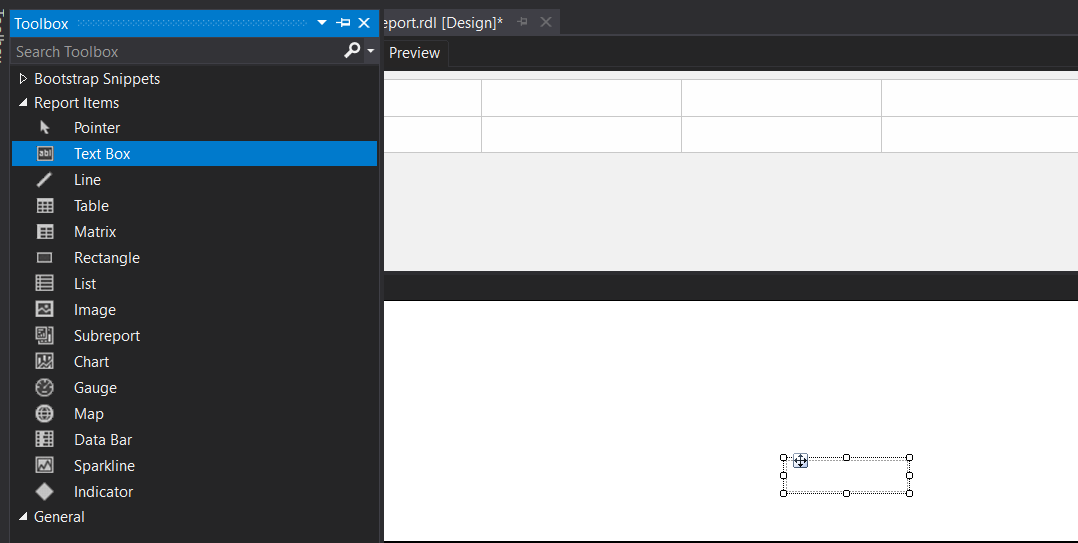
1. Then we can start **formatting** the look of our report. In property pages we can select the orientation of our page



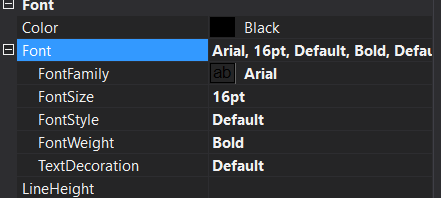
For this I set it to landscape



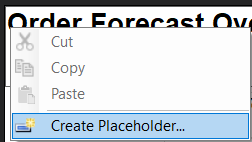
Then we can add a *Text Box* in which to add the title of the report

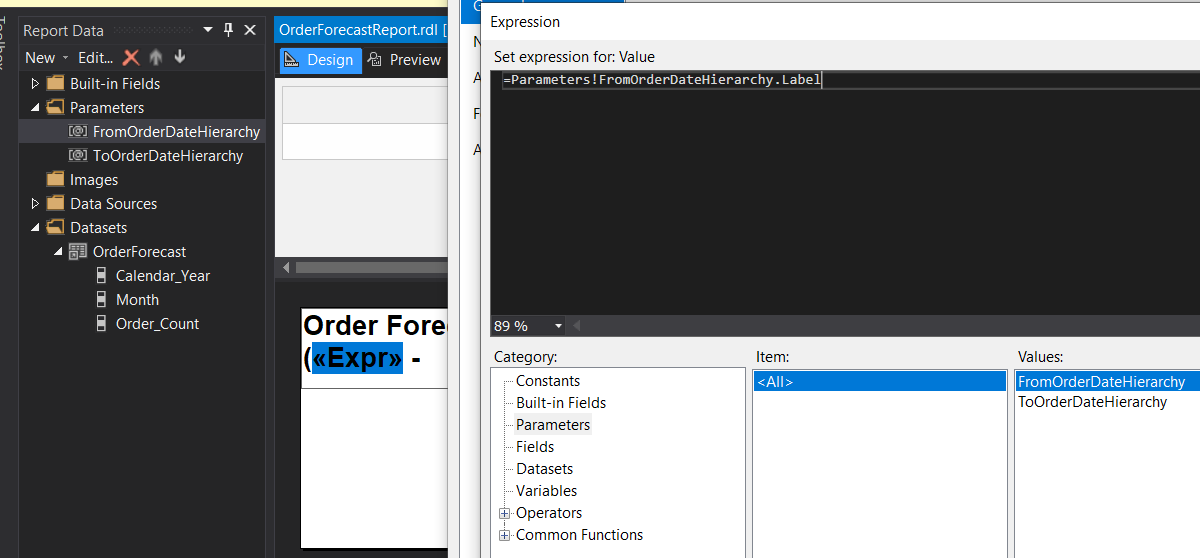


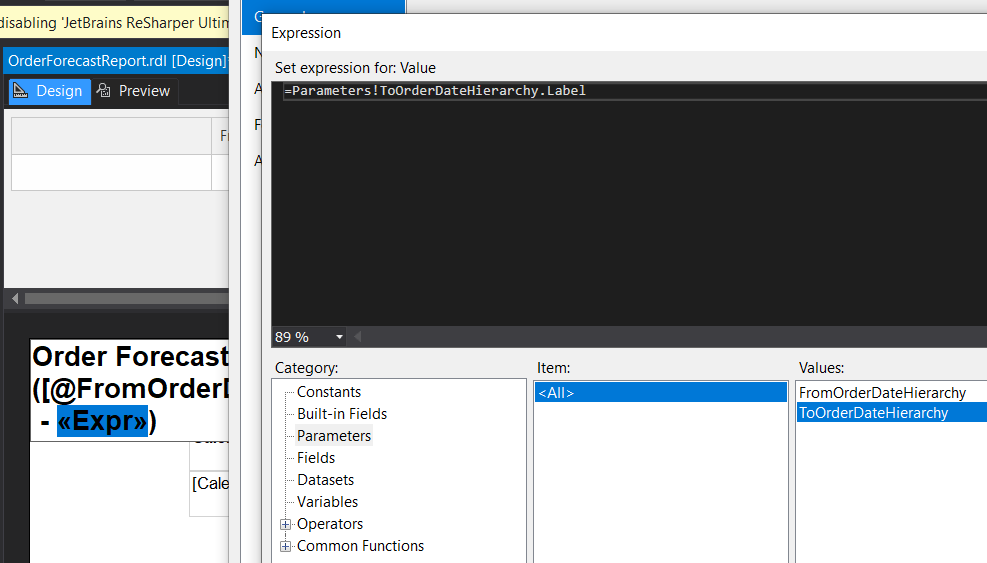
It will have the following **properties**



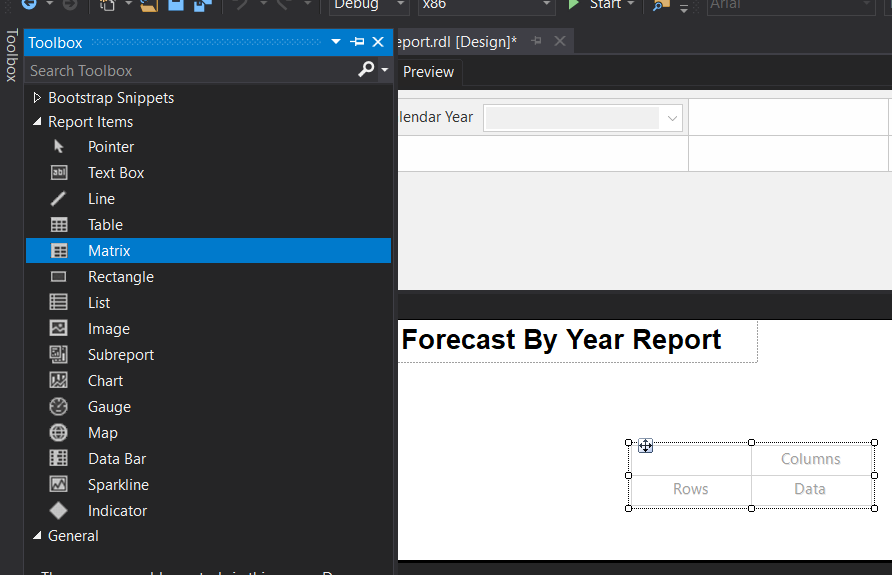
At the end of the title we will create two placeholders in which we will use the properties created when adding the dataset. This will display the range of years in the title





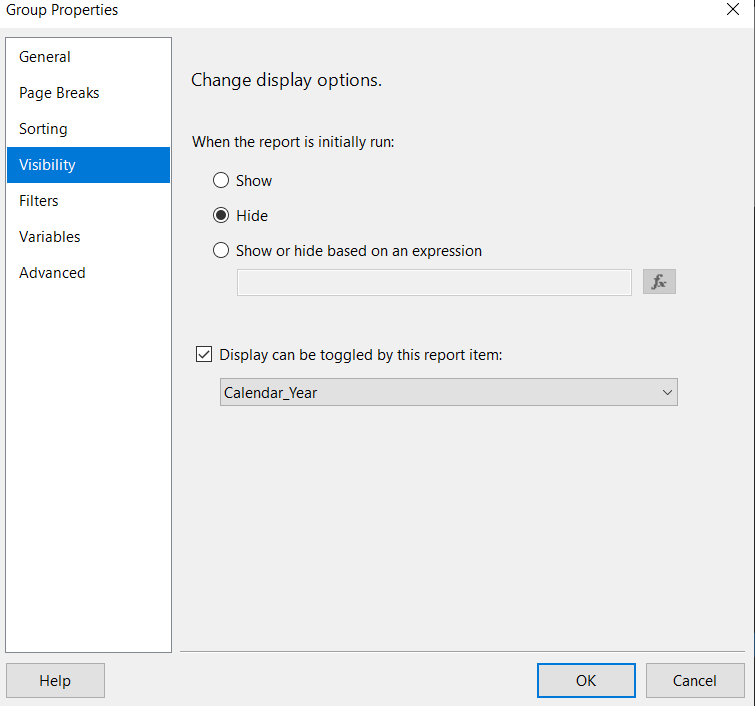


1. Now we **add a** **Matrix** to the report in which we can choose how our data will be displayed



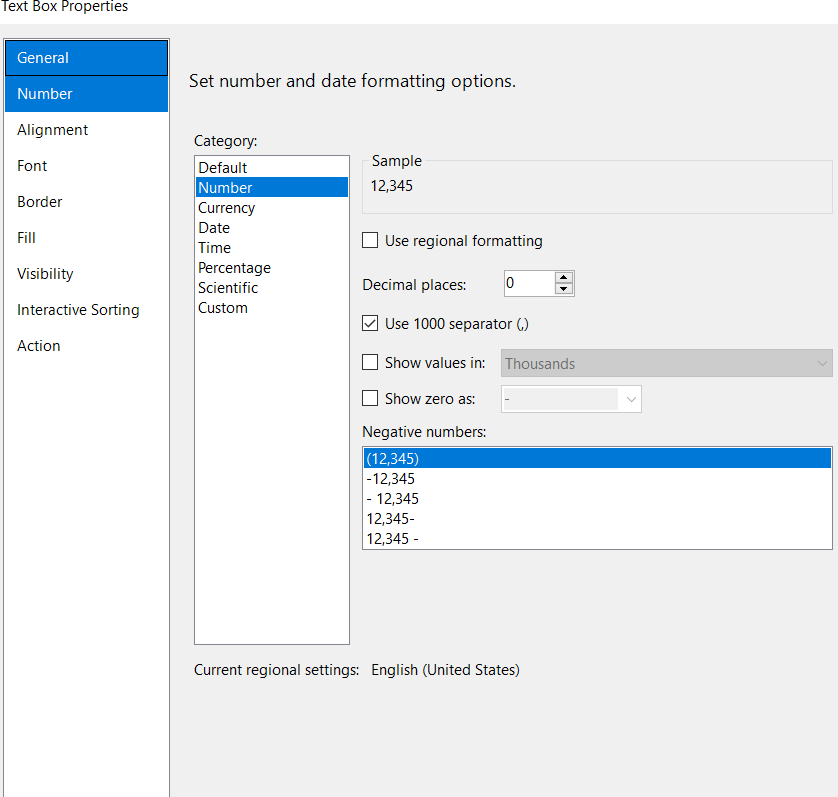
In the *Data* cell we will add the *Order Count* measure, and for the rows we will first add *Month* and to the left of it *Calendar Year*.

Now we need to set some properties to **enable drill down**. For *Month* in the visibility options we will hide it and use *Calendar Year* to toggle the *Month* display

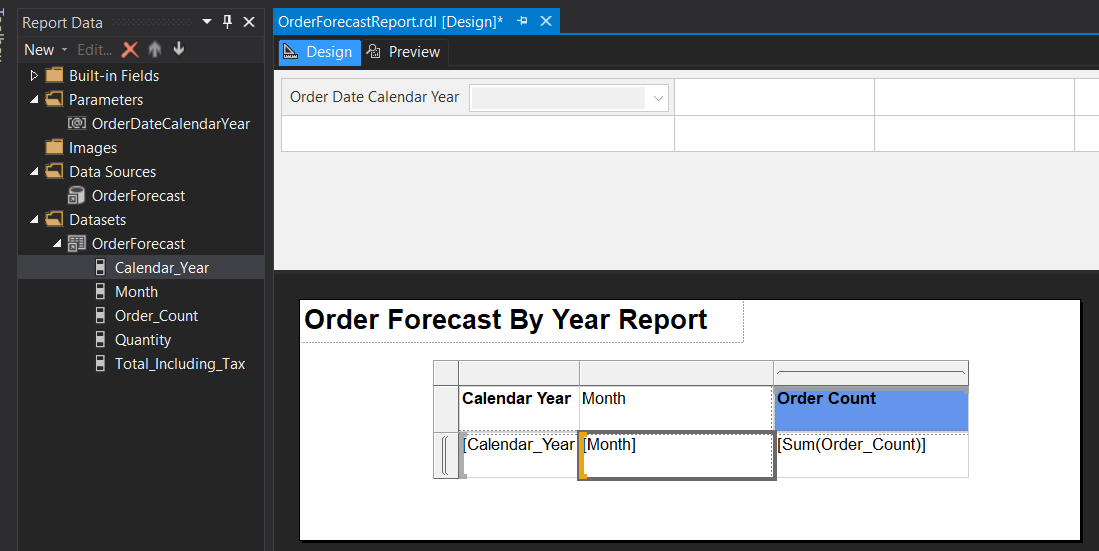


1. Now to make some adjustments to make the table look better

For the *Order Count* cell, we will set it as *Number* and use a thousand separator and reduce the decimal places to 0 as the number will always be a whole one.

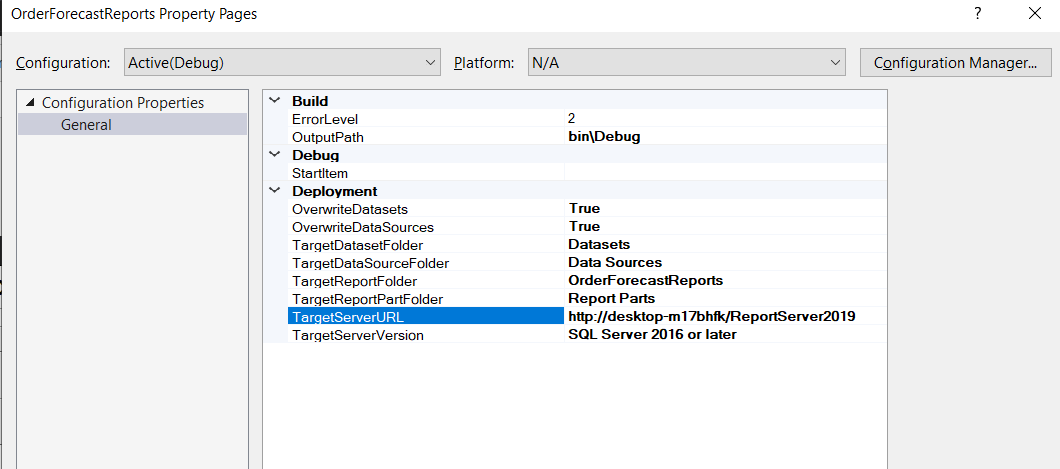


I also set the *Calendar Year* and *Order Count* headers to bold and added a background for the *Order Count*

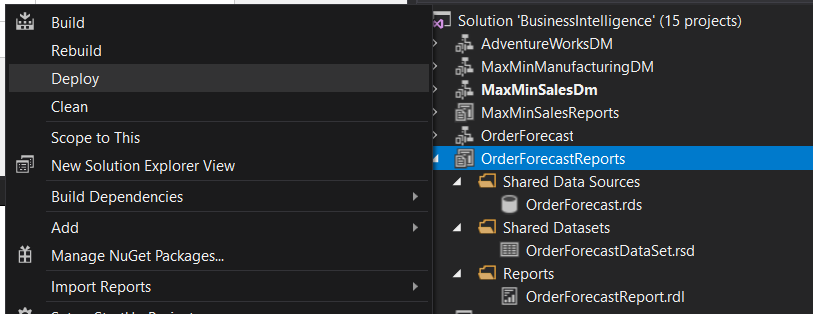


1. Now we can get to **deploying the report to our SSRS server**

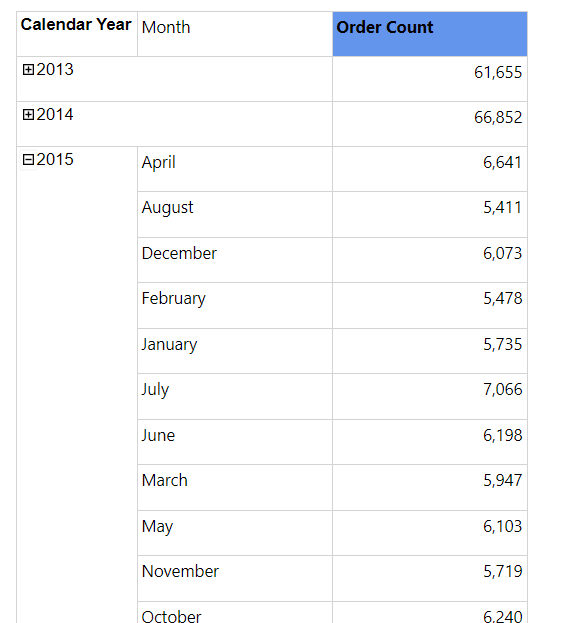
I set *OverwriteDatasets* and *OverwriteDataSources* to **True** in the deployment settings to avoid issues with multiple deployments



The report is ready to be deployed and viewed in the browser

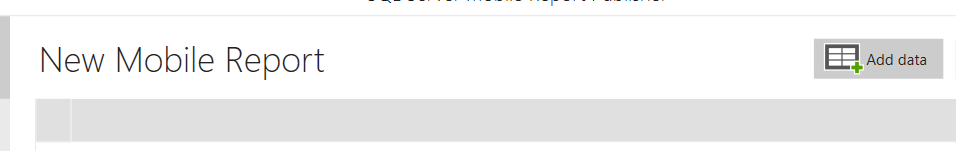


This is what the report looks like after it was deployed. We can also see that **drilling to month level works**

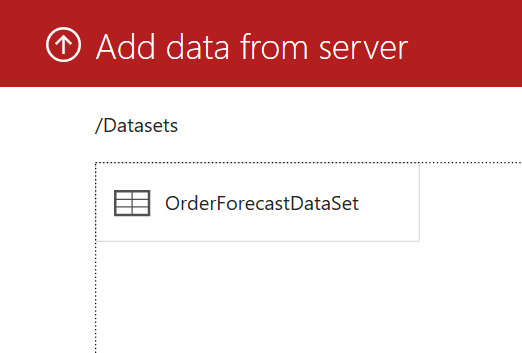


1. Next step is to **create a mobile report**

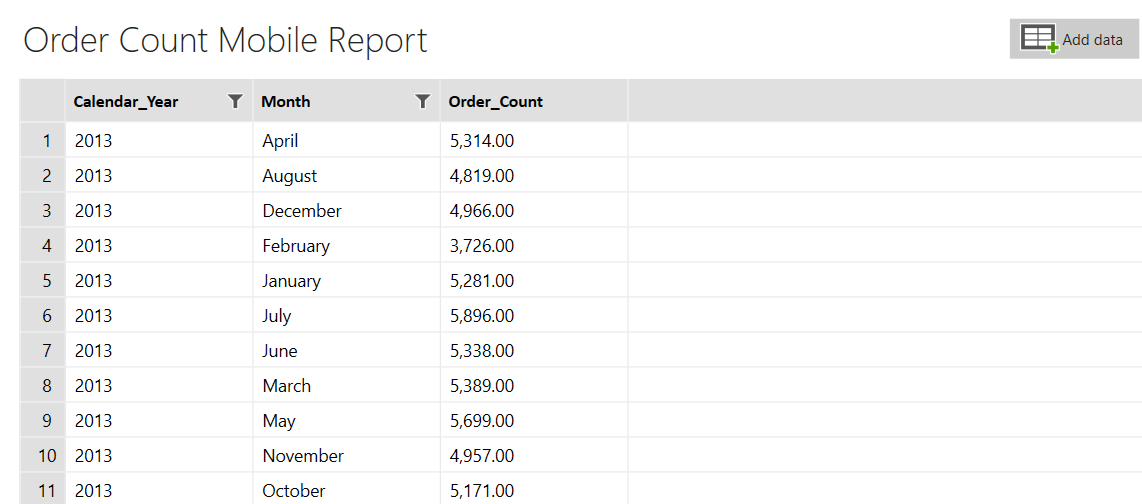
We start by **adding data**



We will use the previously created dataset



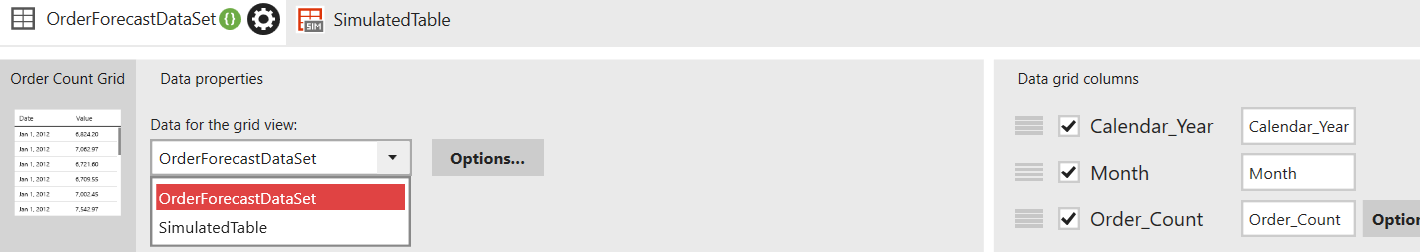
We also rename the report to *Order Count Mobile Report*



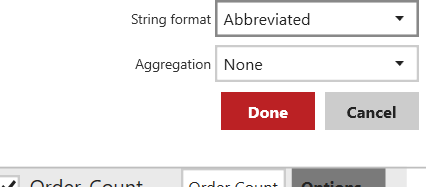
Now we can add some visualization. We add a simple **data grid** and name it *Order Count Grid*

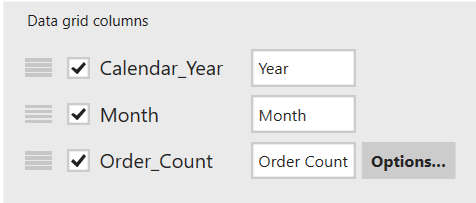


Then we can set the **grid to use our dataset**



For design reasons we also rename the *Calendar\_Year* and *Order\_Count* headers and also set *Order\_Count* to be displayed in abbreviated form.

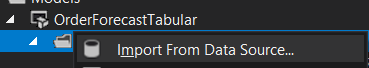


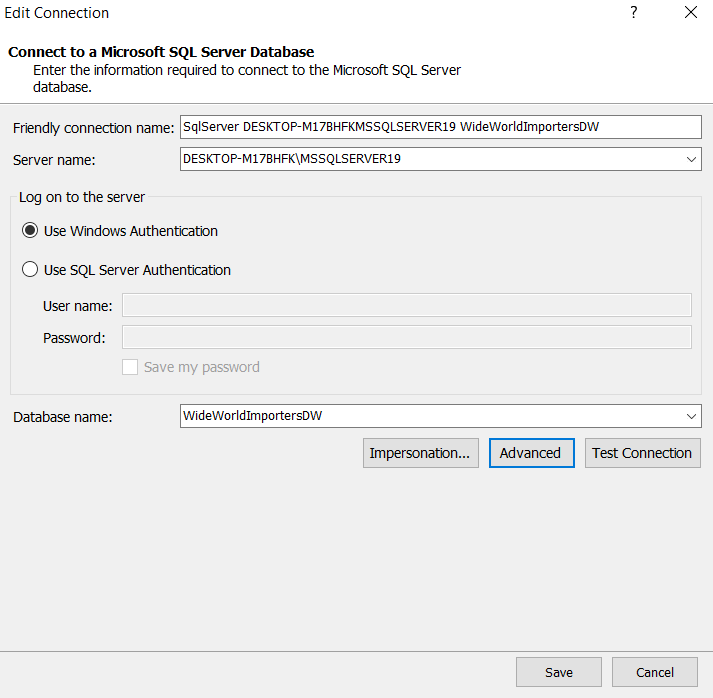




1. Next we will **create a tabular model** for the orders

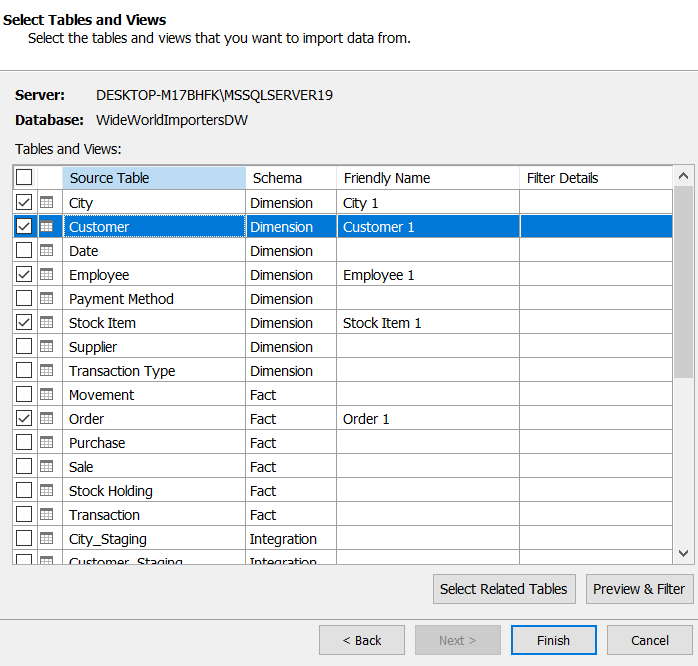
After **creating the project**, we add a data source to it



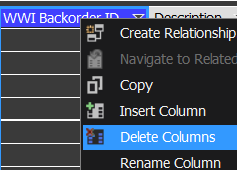


We will choose the Order table first and then add the related tables. Then we deselect date from the menu because we will be **creating our own calculated Time table** in the model. The Time table we will make will contain only what we need for **creating YTD measures** and also some helper columns for properly sorting by *Month* and *Month of Year*

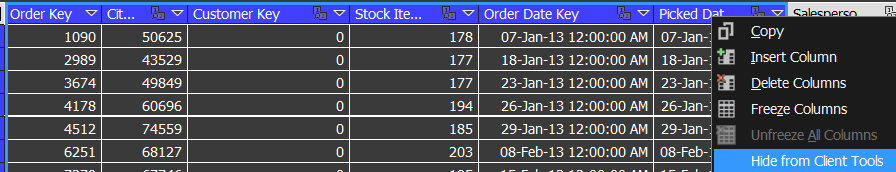


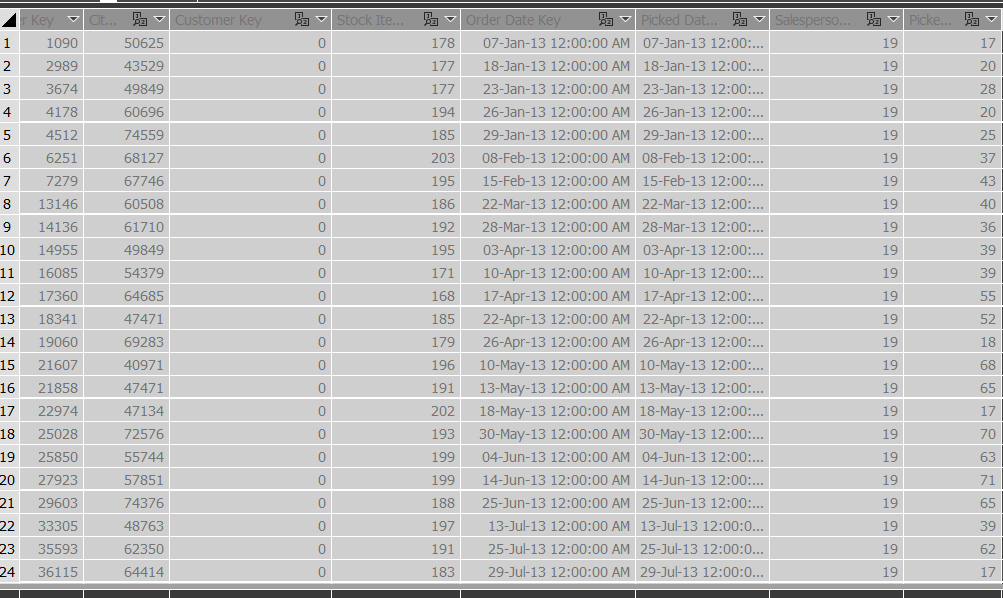


Then we **remove some columns** we don’t need like the *WWI Backorder ID* and the Lineage *Key*



We will also **hide the foreign key columns**

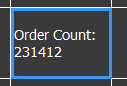




1. Now we can get to **adding some measures** to our model

First we add the *Order Count* by counting the rows in the *Order* table





We also make a measure for the total quantity of produce and the total value in dollars for the orders







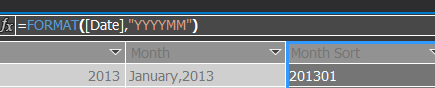
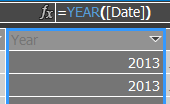


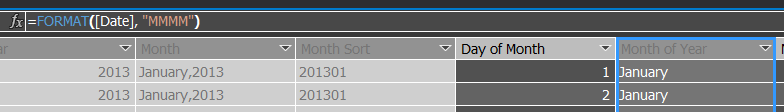
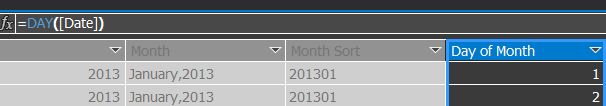
1. **We create a time table** to be able to attribute the values we see to specific points in time

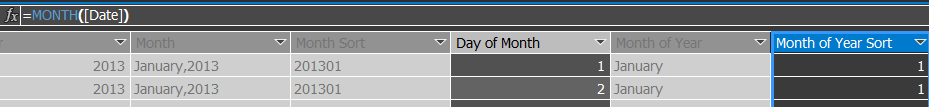


Our data about orders dates from 2013 up until May 2016 so we will create a table from the beginning of 2013 until the end of 2016

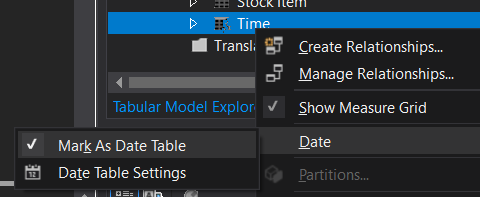
Because we have full dates we will add some more columns to help us split up this information. We will make a *Year*, *Month*, *Day of Month* and *Month of Year* column. *Month Sort* and *Month of Year Sort* will help with sorting the dates properly.

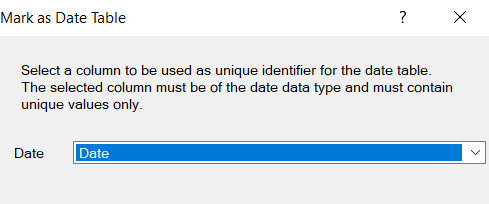




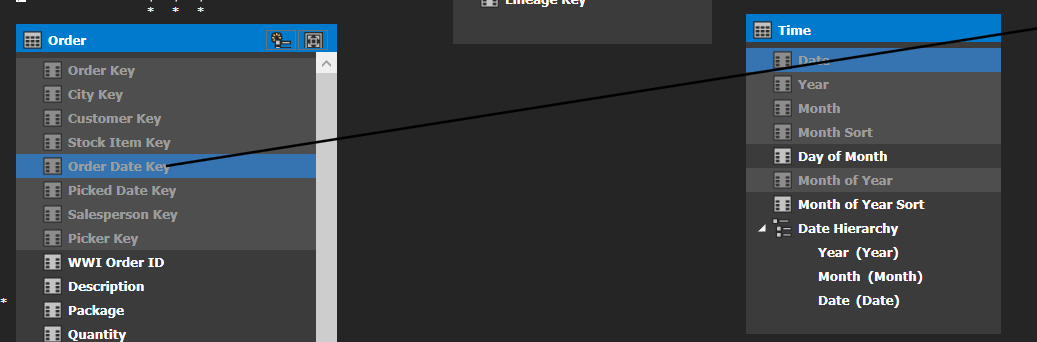


Next we will **mark this table as a date table**. Also we should make sure that the *Date* field is **marked as unique identifier** for the *Time* table

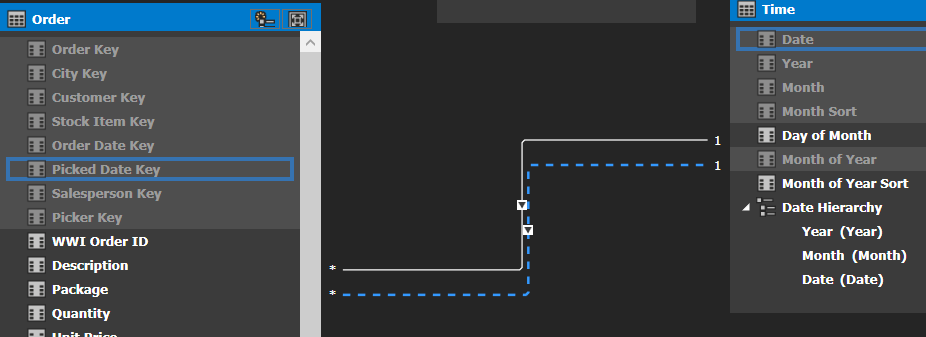




Now we need to **add the relationship** linking the *Order* table with the table we just created



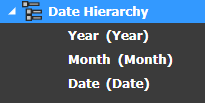
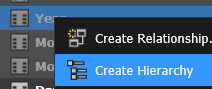
We can also link the *Picked Date Key* to *Date* as well



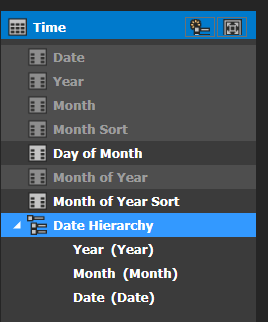
Now we can **set the sorting**. For *Month* and *Month of Year* we go to properties and set the *Sort by Column* property to the columns that have *Sort* in the name



We can also **make a** **hierarchy** for our *Date* columns now that the table is all setup. Our hierarchy will start from the *Year* being the top most level, then we will add *Month* and *Date* bellow it in that order

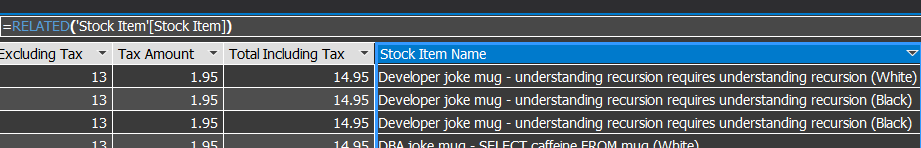


The *Time* table will look like this in the end

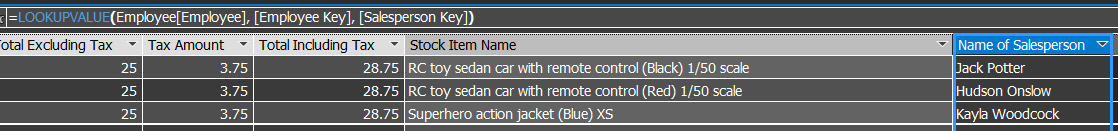


To help us with visualization we will also add 2 **calculated columns** to the *Order* table

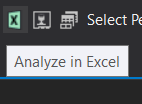
For the first one we will add the actual name of the item that was sold in that order by using the **RELATED** function



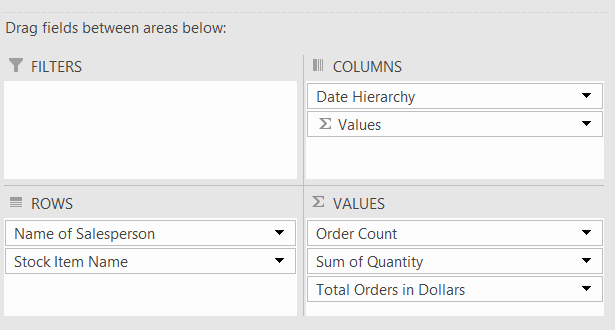
For the second column we will add the name of the salesperson responsible for handling that order. For this we used the **LOOKUPVALUE** function

****

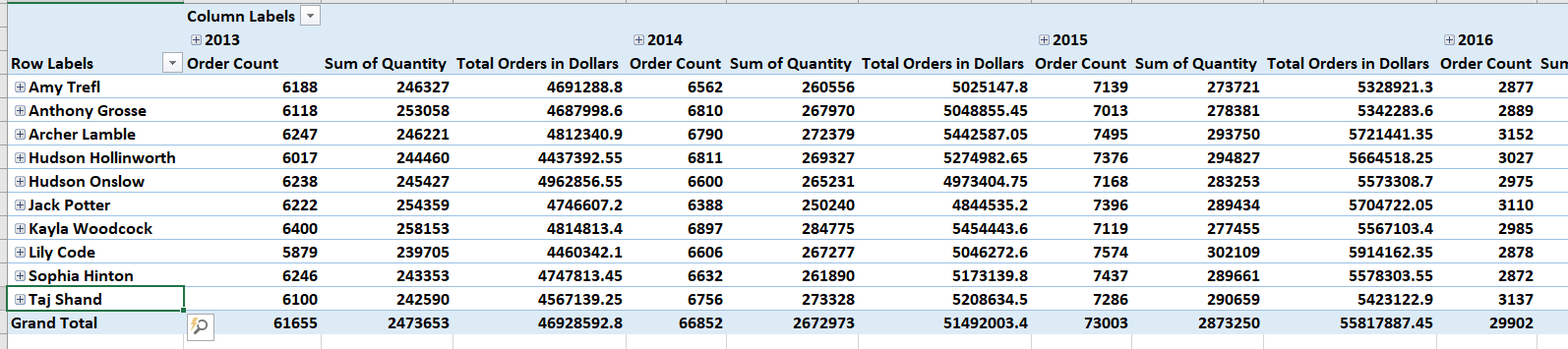
1. We can now analyze our model in Excel

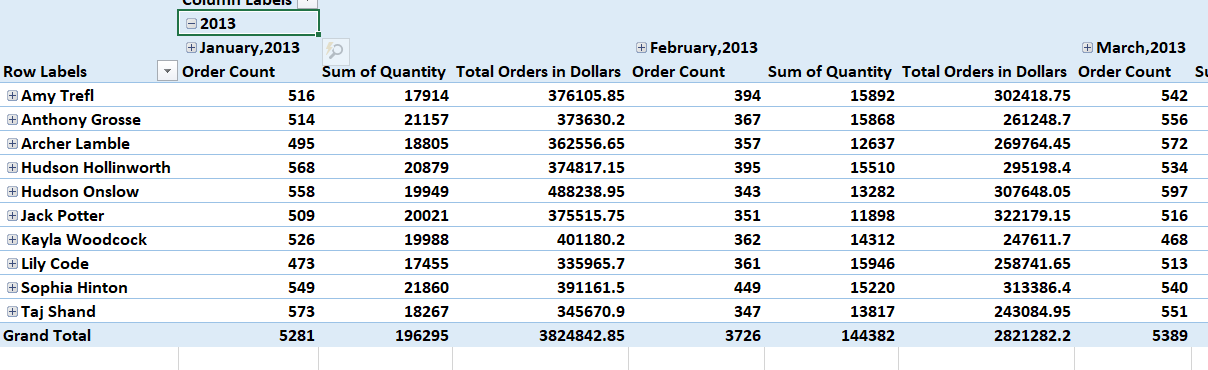


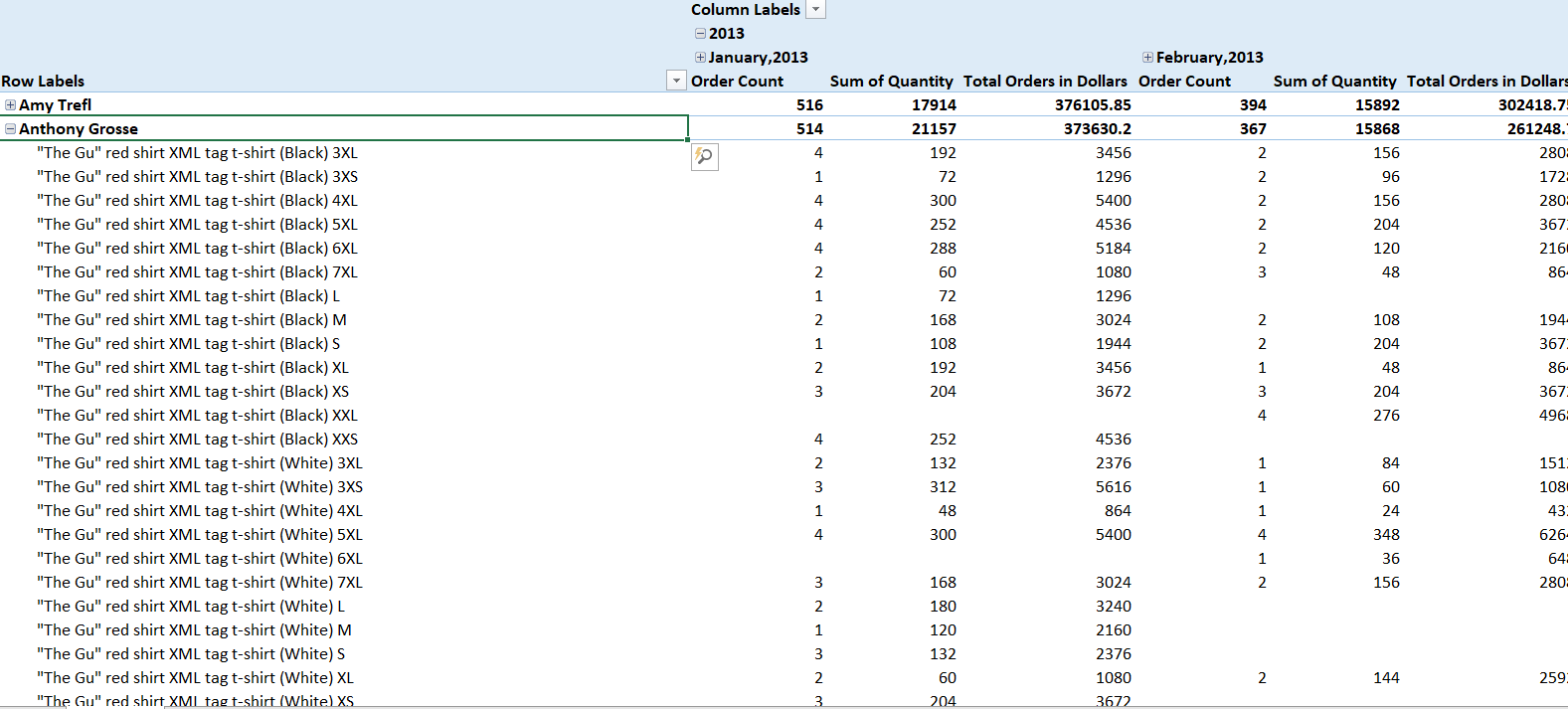
We will setup the pivot table like this. For the rows section we use the **calculated columns** created previously



Now in the table we can see details about *Orders* attributed to the employee that was in charge of completing that order and also the item that was sold. The table also allows to **drill down on the time hierarchy** we created



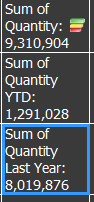




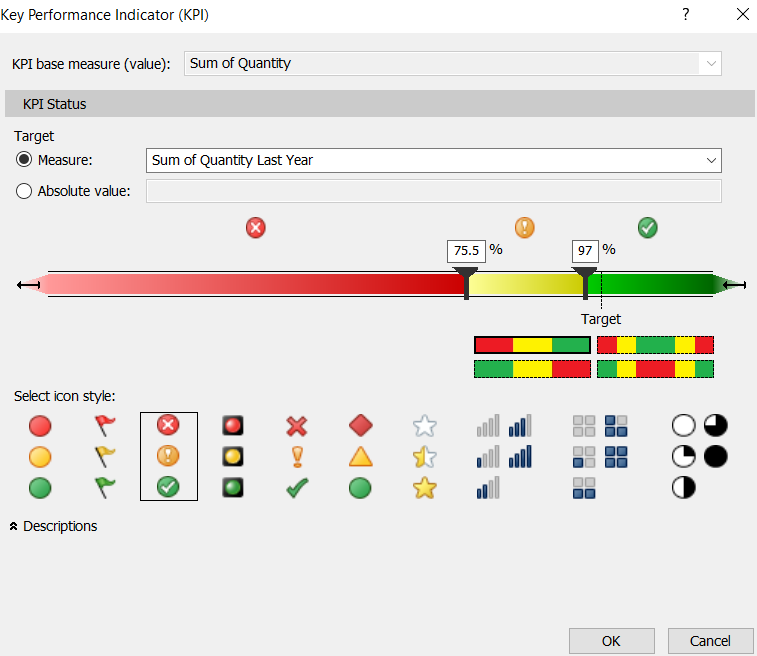
1. We can expand our *Order* table now.

Because we have the *Time* table we can **create some YTD measures**

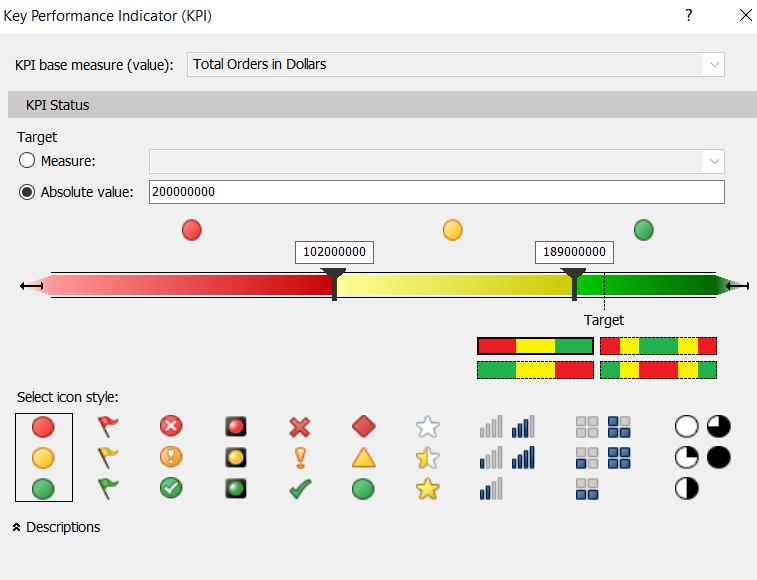




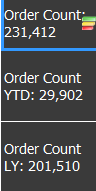
With the Last Year value, we created we can **make a KPI** on the quantity



We can also create a KPI for *Total Orders in Dollars* by using an absolute value

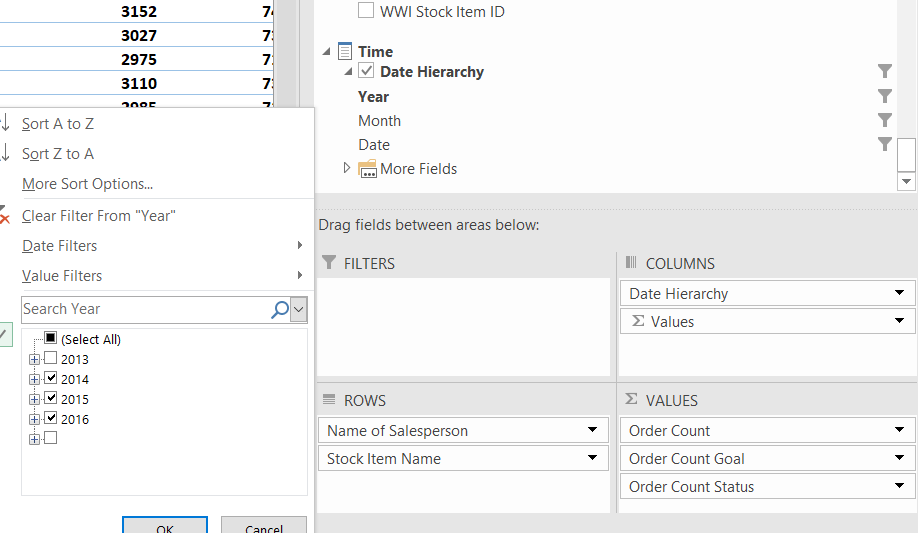


Through the same steps we created the same measures for *Order Count*

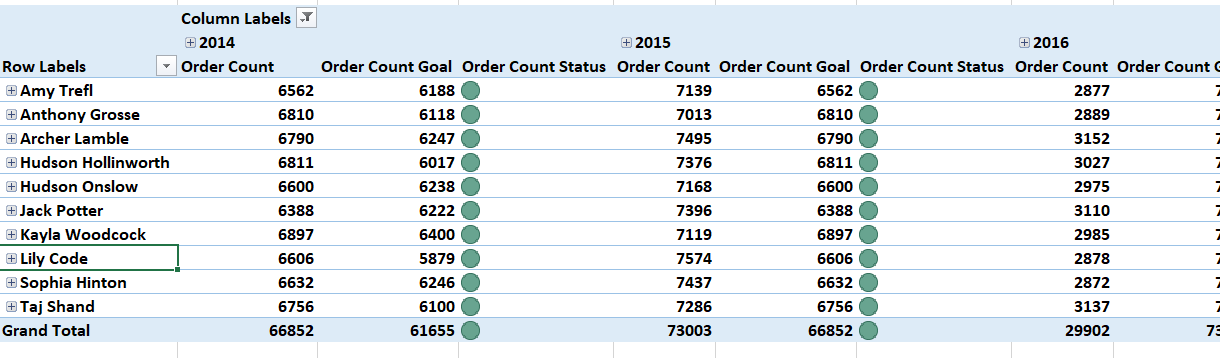


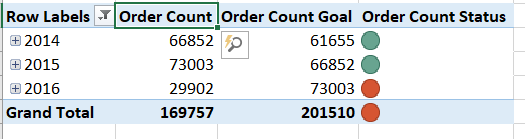
We will go back to visualize our model in Excel now and see how the **KPI**’s we created can be used

In the **pivot table** we will now add the goal and status to our values from the KPI group

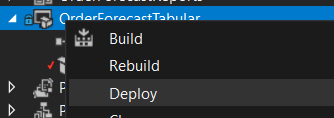


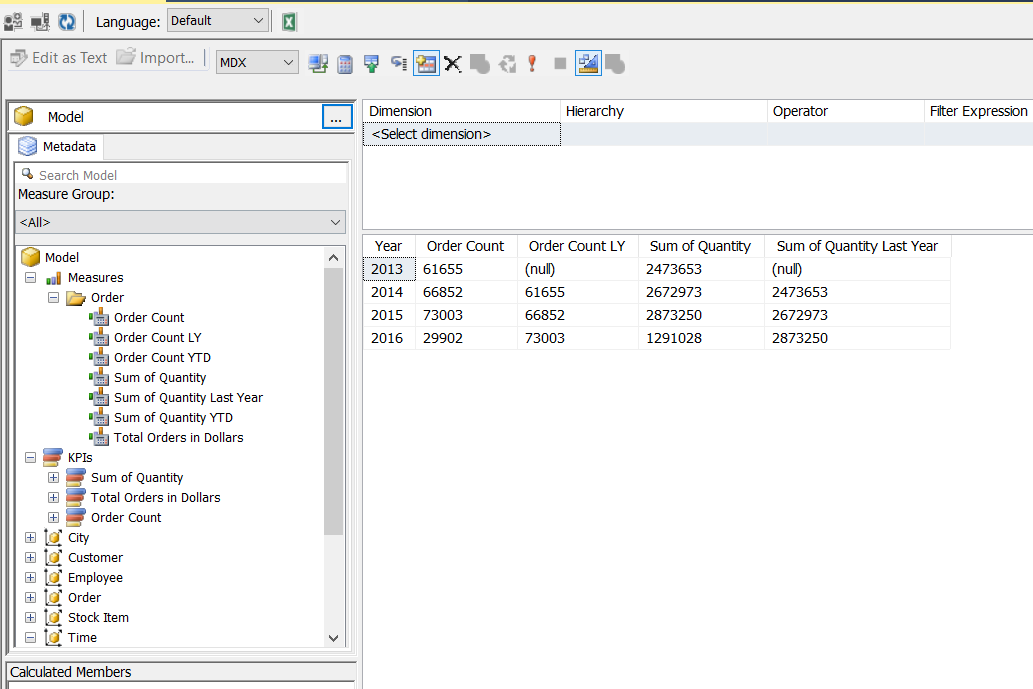
We can see now that the table displays if the goal was reached or not for every row in the table

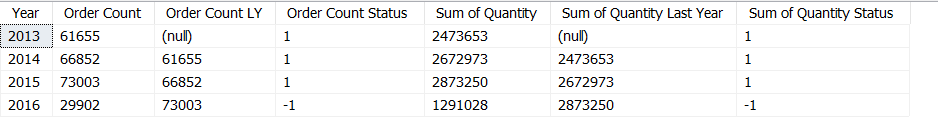




We can now **deploy our project and view on our tabular server**

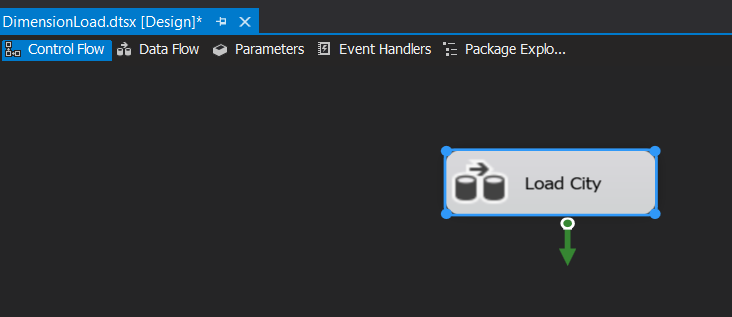




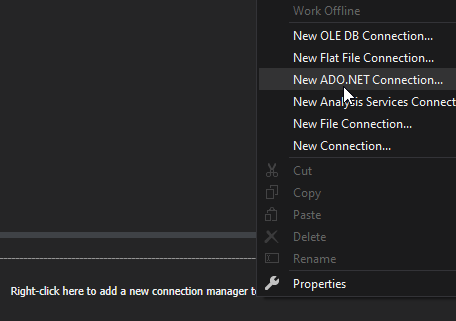


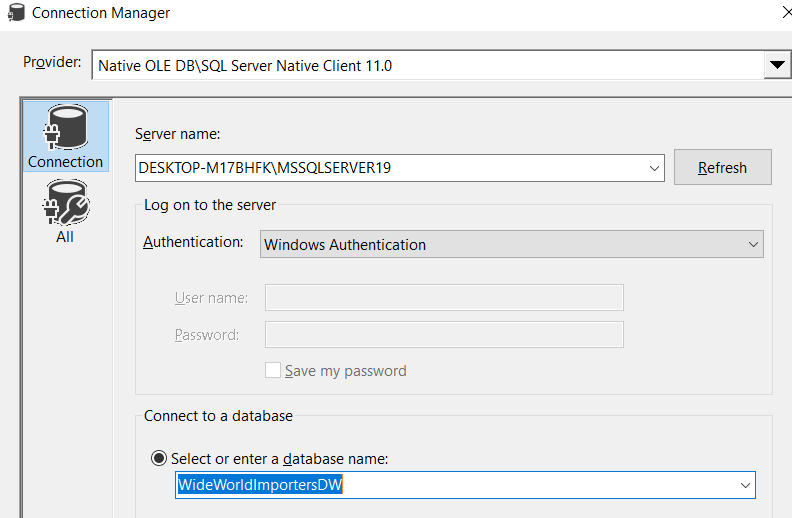
1. Lastly, we will **create an Integration Services project** for the *Order*

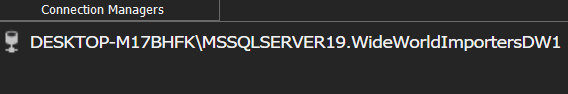
First we start by **creating a data flow**



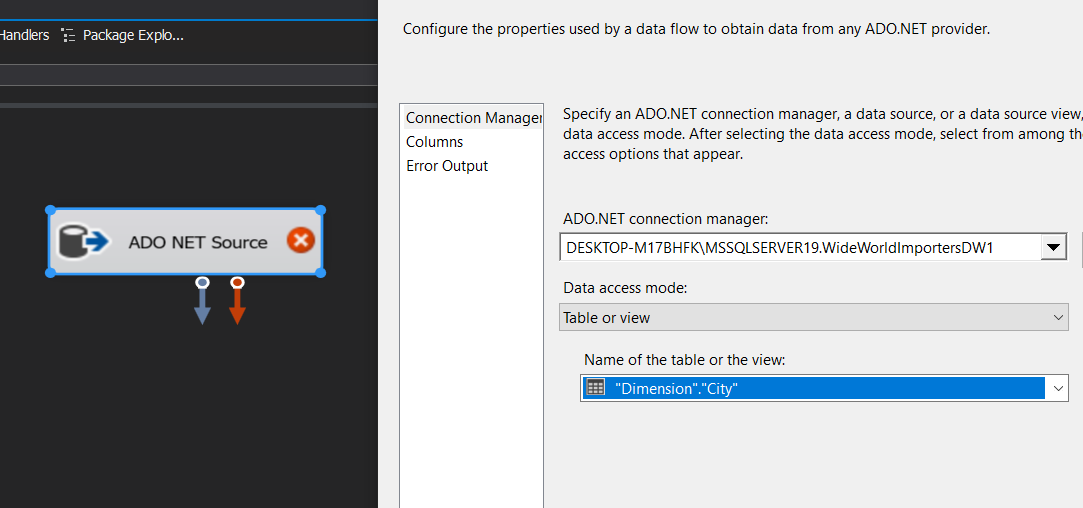
We will start with the **dimensions** and then the **fact** last. We also need to setup a **connection to our data source**



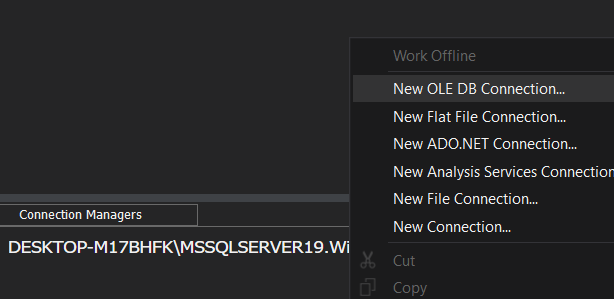


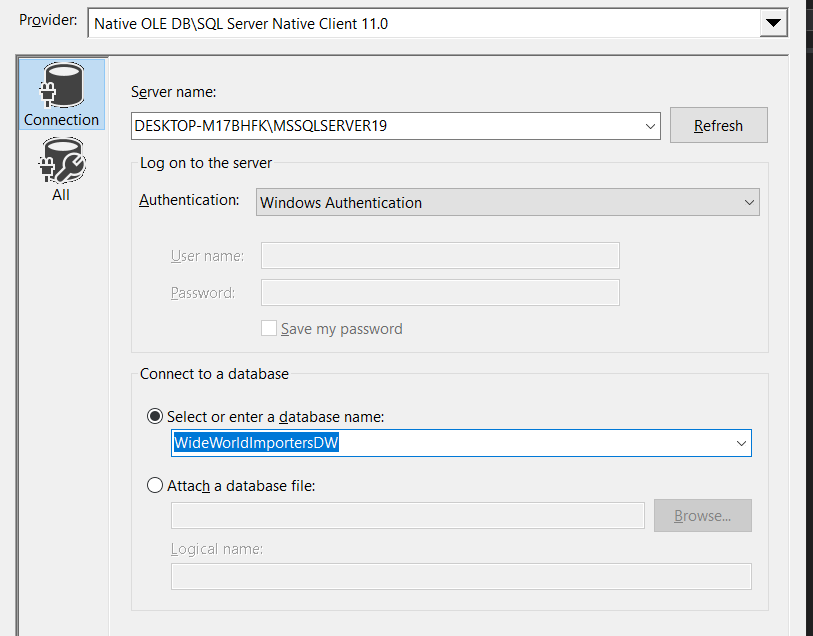


Then we **add an ADO NET Source task** to it

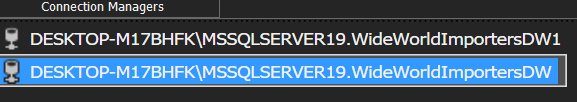


Now we can **setup a destination connection**

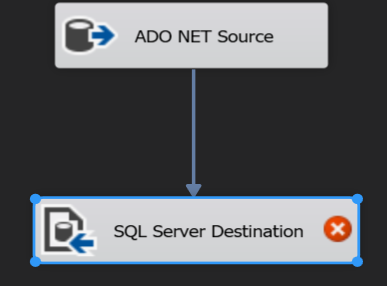


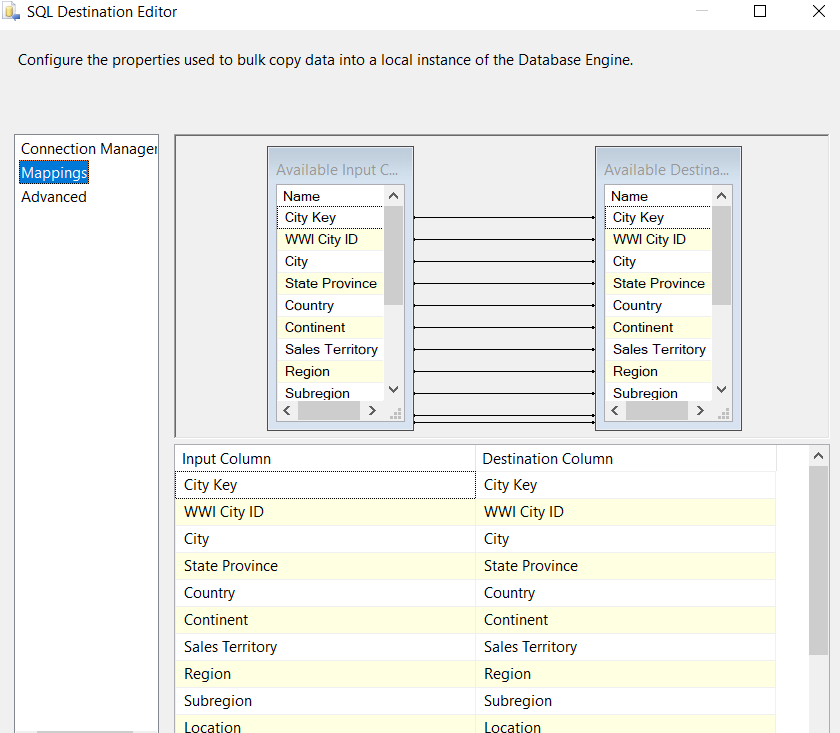


We now have 2 connections available

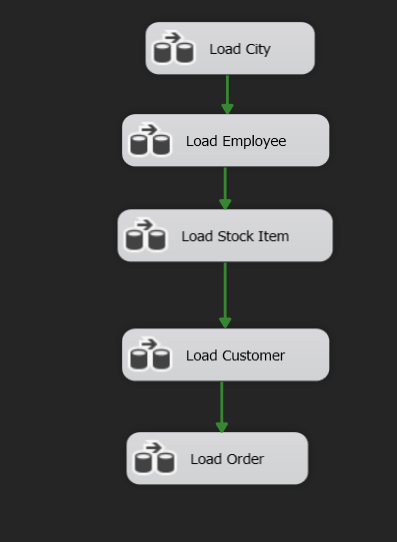


Now we add a **destination item** to the flow, connect the source to it and then select the *City* table in the destination. By viewing the **mappings**, we can see that they were auto assigned properly



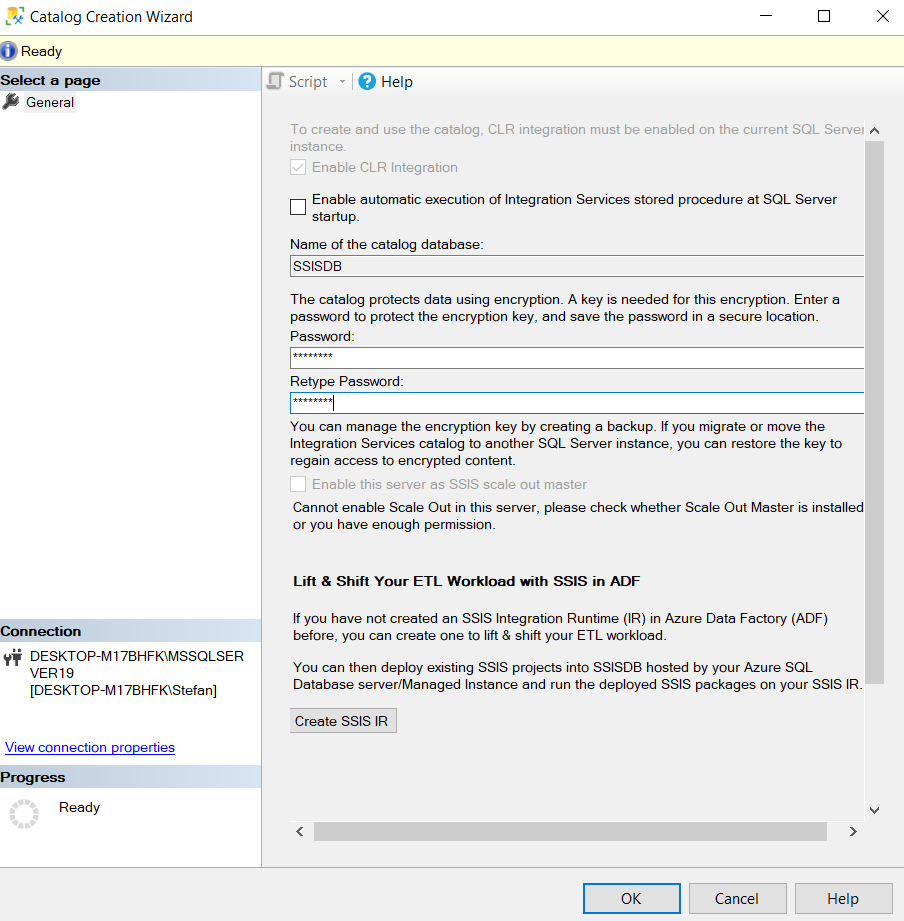


By repeating these steps, we can add the rest of the dimensions



We can **now deploy our project**

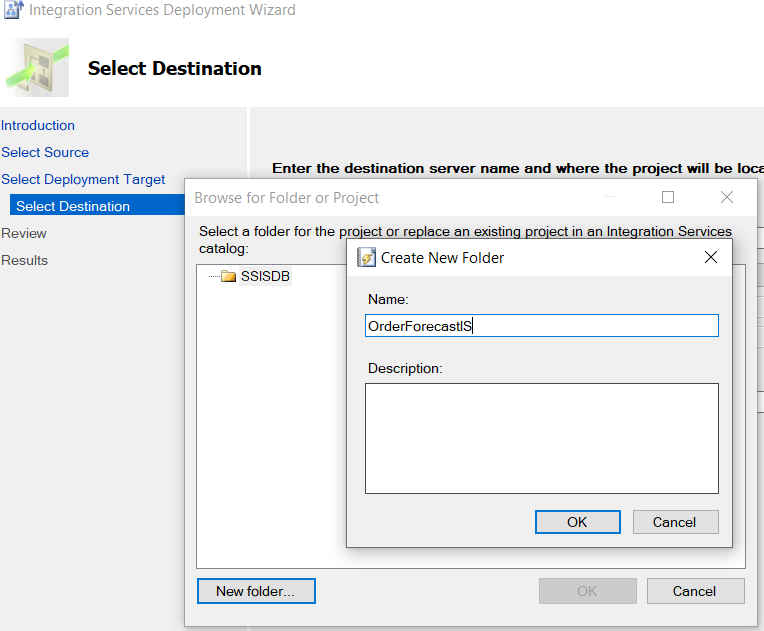
After **adding integration services as a feature** to our SQL Server instance we can now go and create a catalog in our database engine

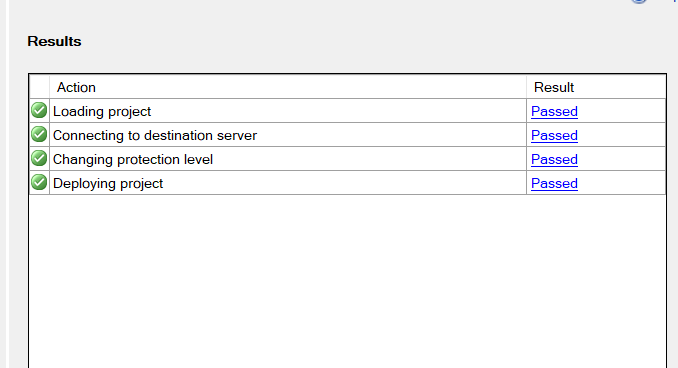




The project can now be deployed







We now have our project deployed on our server

