The SMB Focused Dynamics Partner Community

DIRECTIONS EMEA

Table of Contents

Abstract 3
Prerequisites 4
The objective 5
The approach 8
Workshop Steps 9
- Create a new project in Visual Studio Code 9
- Add queries to the extension 21
- Publish the queries as web services 29
- Obtain the ODATA web services URL 33
- Import the ODATA queries in Power BI Desktop 34
- Create the Data Model and relations 45
- Enrich the Data Model with DAX 47
- Create the Report(s) 55
- Publish your pbix file to the Power BI Service 67
- Create a Dashboard 70

Abstract

In this workshop you will learn how you can create stunning dashboards with the Power BI toolset.

Using a hands-on approach, you will get an overview of the Power BI toolset and how to apply it to create stunning dashboards that not only bring your data to life but also learn from it.

At the end, you will have the skills, so you can turbo-charge your reporting capabilities with Business Central and Power BI.

Prerequisites

For the workshop we will use:

- **An Office 365 trial**
  - Please go to https://www.microsoft.com/nl-nl/microsoft-365/business/office-365-enterprise-e5-business-software
  - Select: Create a Trial
  - Go over the complete registration process
  - Make a note (in notepad) of your email-address and password.
  - As an alternative you could also use demos.microsoft.com to create a trial tenant.
- **Power BI Desktop**
  - Please visit https://powerbi.microsoft.com/en-us/desktop/, download and install Power BI Desktop. You can install it on your local machine (laptop) or on the VM on Azure.
- An **account on powerbi.microsoft.com**
  - Please visit https://powerbi.microsoft.com/en-us/ and use your Office 365 trial credentials to sign in.
- **Visual Studio Code (VS-Code)**
  - Please visit https://code.visualstudio.com and then download and install VS-Code.

Important

You can use your own office 365 credentials, if you have them. But be aware that we will use a Business Central Cloud tenant, create and publish an extension. If your company is already using their Business Central tenant, then it would strongly recommend you create **a new** Office **trial** instead.

Summary:

If you decide to use an already existing Office 365 environment, then do so at your own risk, you have been warned!

The objective

The goal of this workshop is to create the following Dashboard:

PLATAAN

Amount Outstanding Tot
BY CUSTOMERNAME, YEAR, MONTHNAME

Year			2019			
CustomerName		Total	Apr	May	Sep	Total
	0.00	0.00				
Adatum Corporation			20,390.44	2,778.64	22,008.21	**45,177.29**
Alpine Ski House				573.00	1,231.00	**1,804.00**
Relecloud					8,542.00	**8,542.00**
School of Fine Art			5,208.80		9,115.40	**14,324.20**
Trey Research					3,560.79	**3,560.79**
Total	**0.00**	**0.00**	**25,599.24**	**3,351.64**	**44,457.40**	**73,408.28**

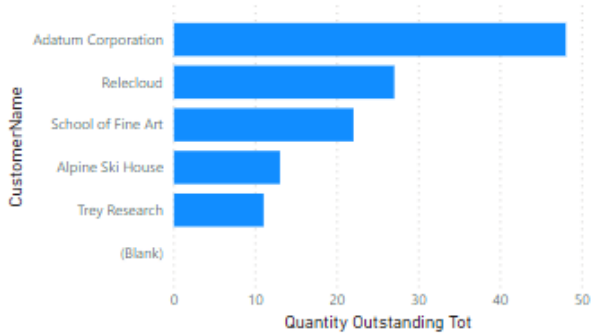
No Of Customers
6

No Of Items
8

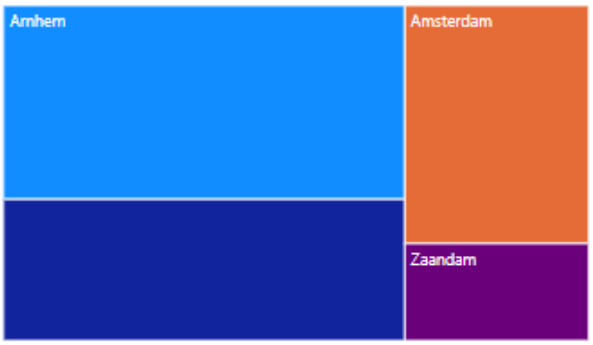
Quantity Outstanding Tot
121.00

Amount Outstanding Tot
73.41K

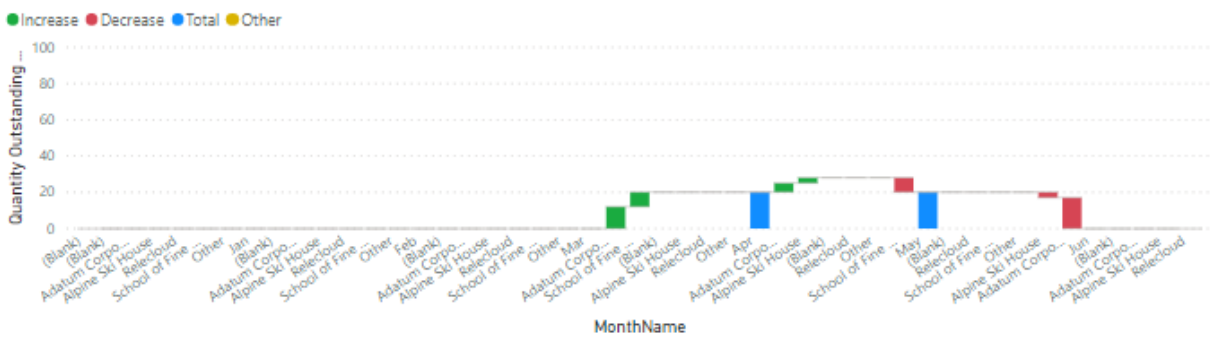
Quantity Outstanding Tot
BY CUSTOMERNAME



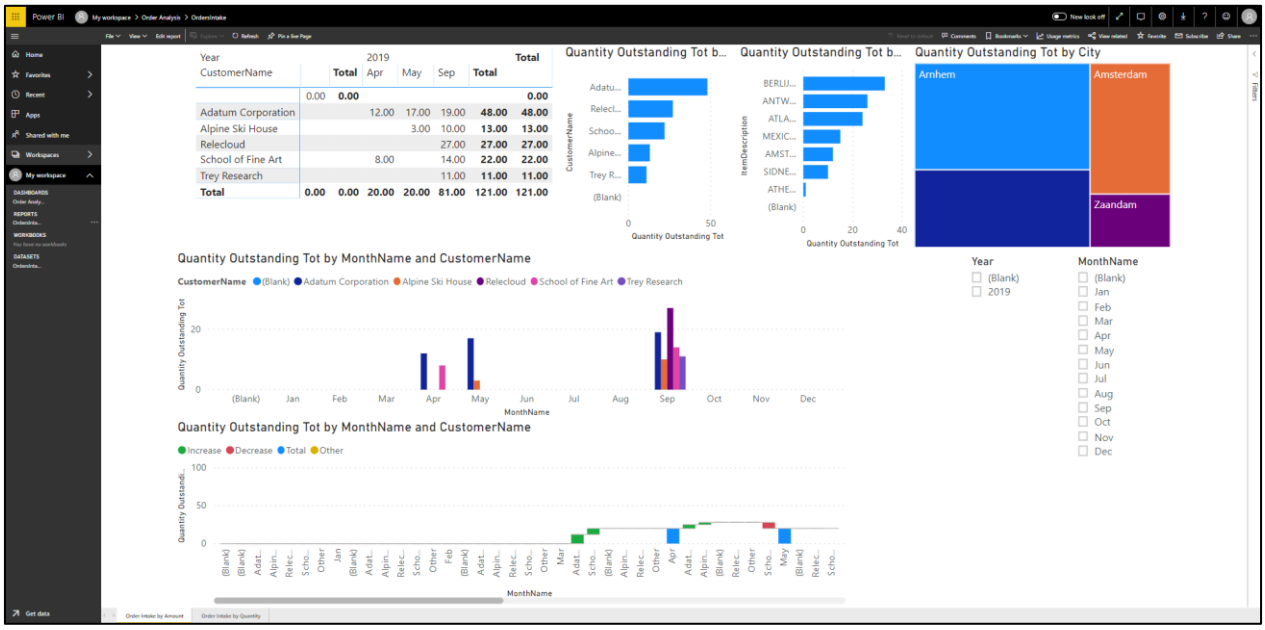
Quantity Outstanding Tot
BY CITY

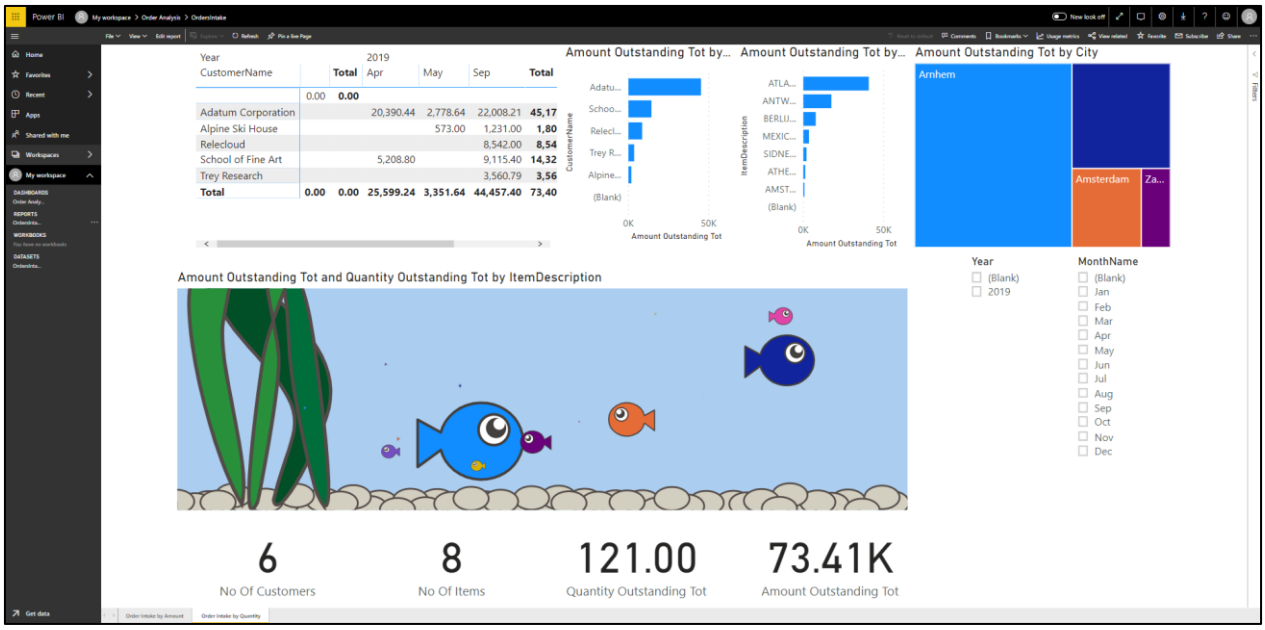


Quantity Outstanding Tot
BY CUSTOMERNAME, MONTHNAME



With an underlying report:





The approach

We will get there in the following steps:

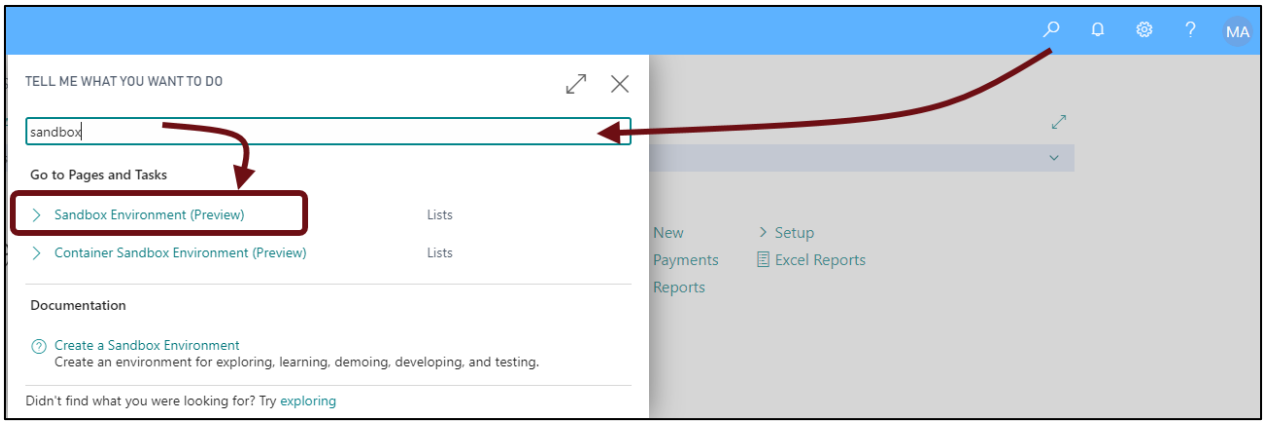
1. Create a new project in VS-Code
2. Add queries to the extension
3. Publish the queries as web services
4. Obtain the ODATA web services URL
5. Import the ODATA queries in Power BI Desktop
6. Create the Data Model and relations
7. Enrich the Data Model with DAX
8. Create the Report(s)
9. Publish your pbix file to the Power BI Service
10. Create a Dashboard

Workshop Steps

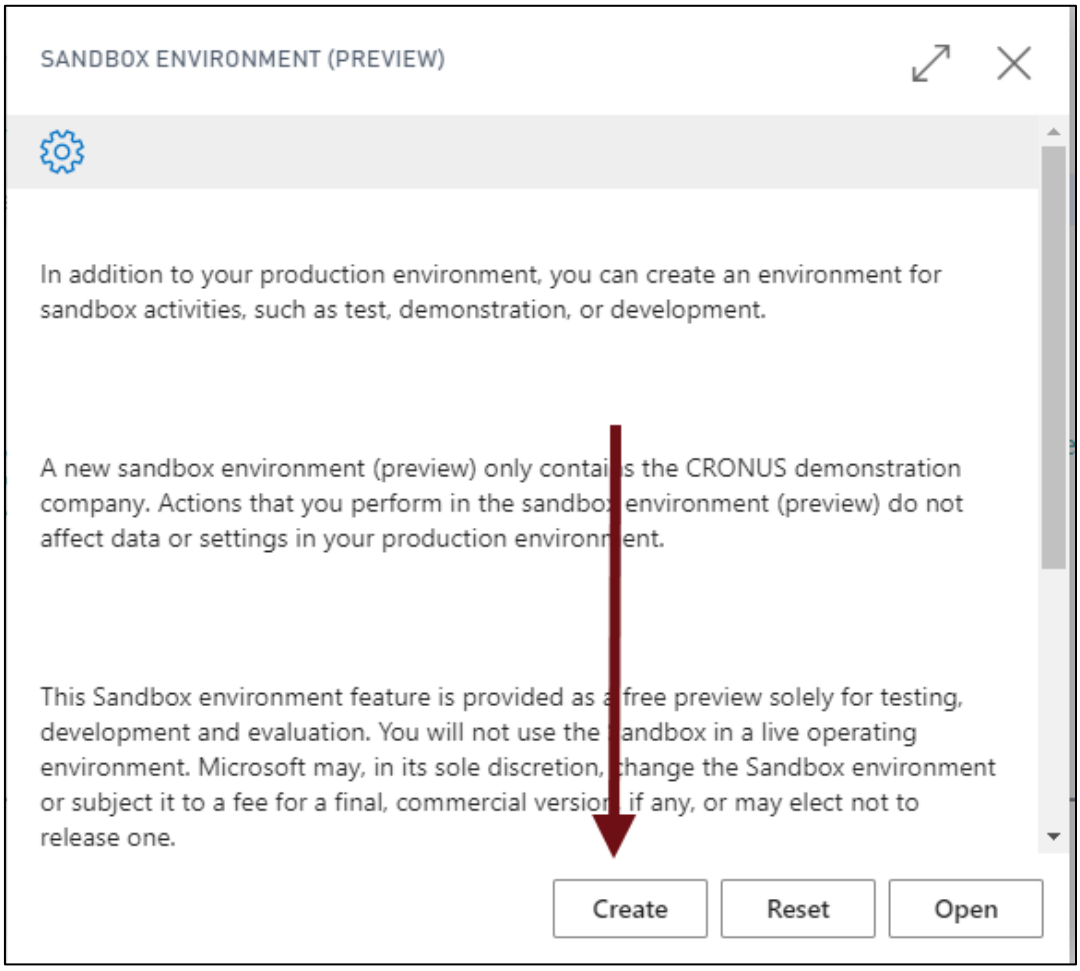
Create a new project in VS-Code

Open your Business Central sandbox. If you don't have a sandbox yet then you should create one.

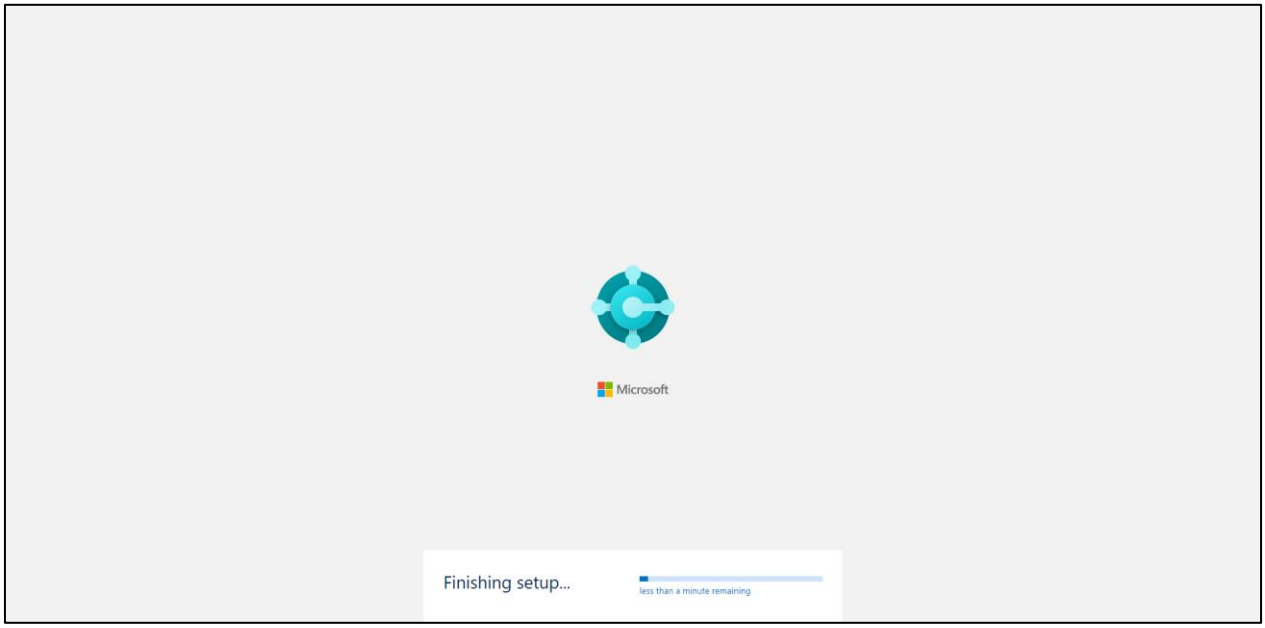
In Business Central, use the search, look for sandbox:



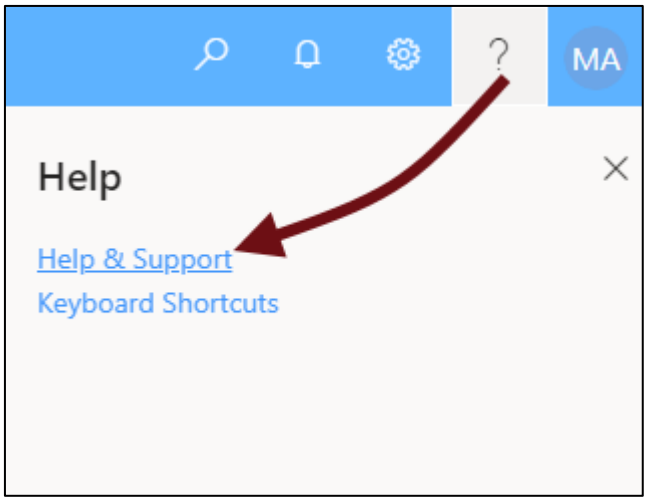
As shown in the screenshot, select Sandbox Environment (Preview), then click on Create:



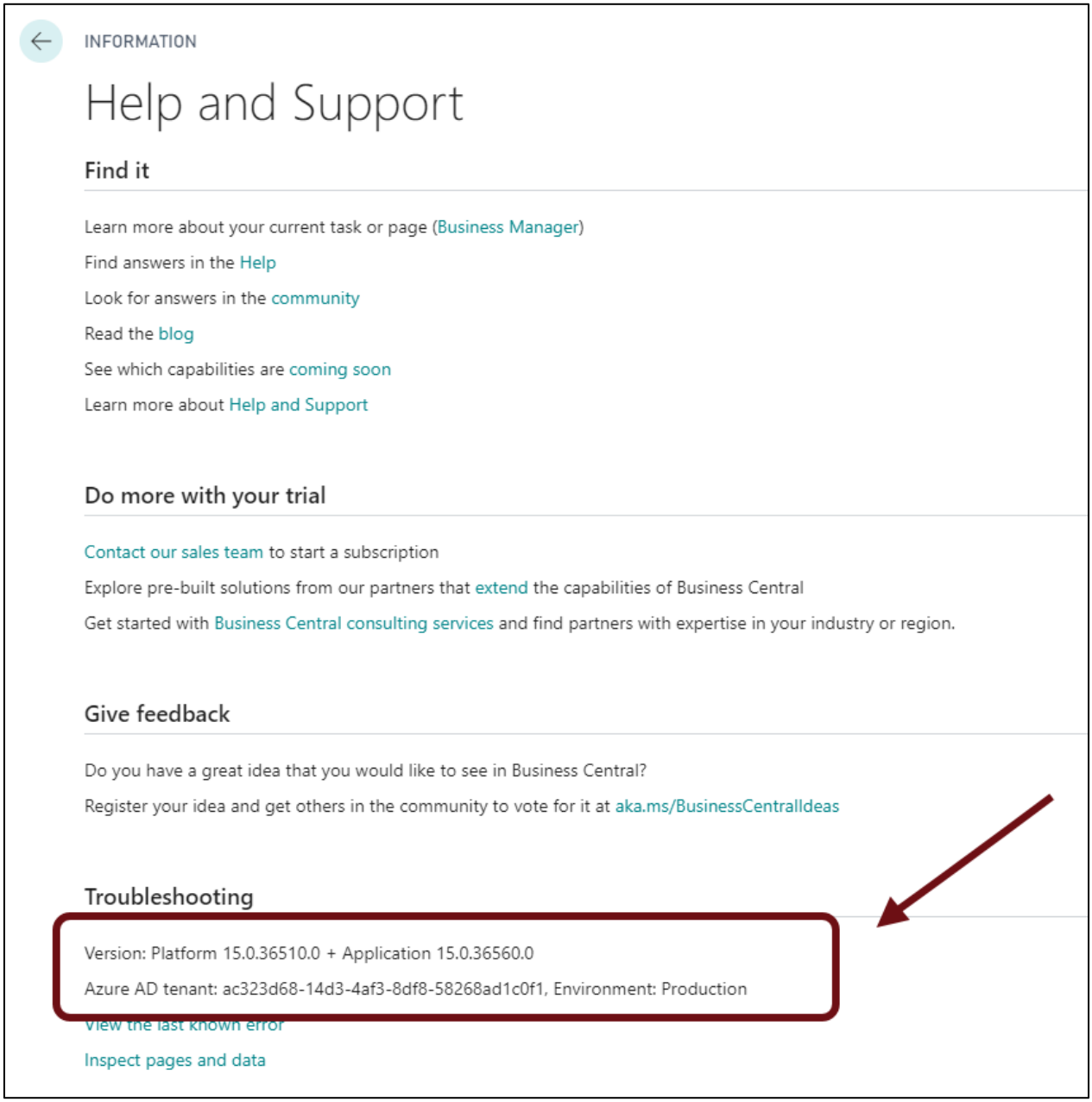
Wait until the sandbox is ready.



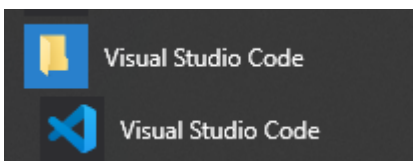
In the meantime, you can go to the help in your production tenant:

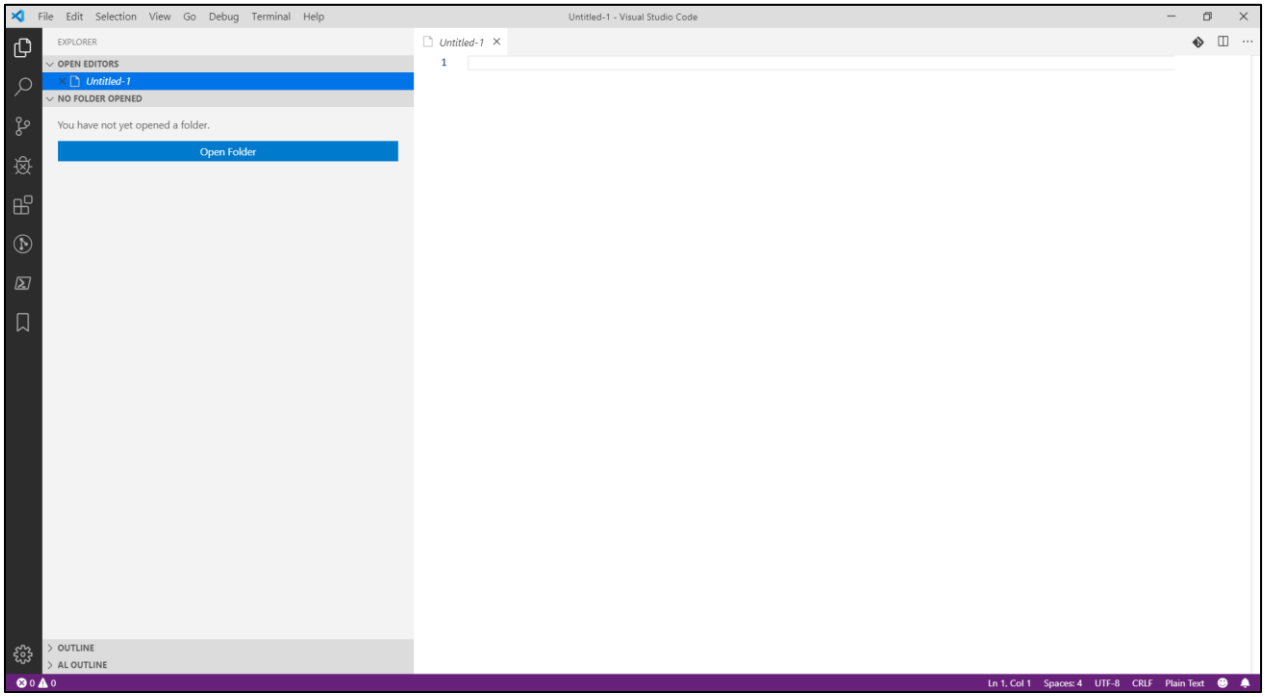


And make a note (copy in notepad) of the following settings:

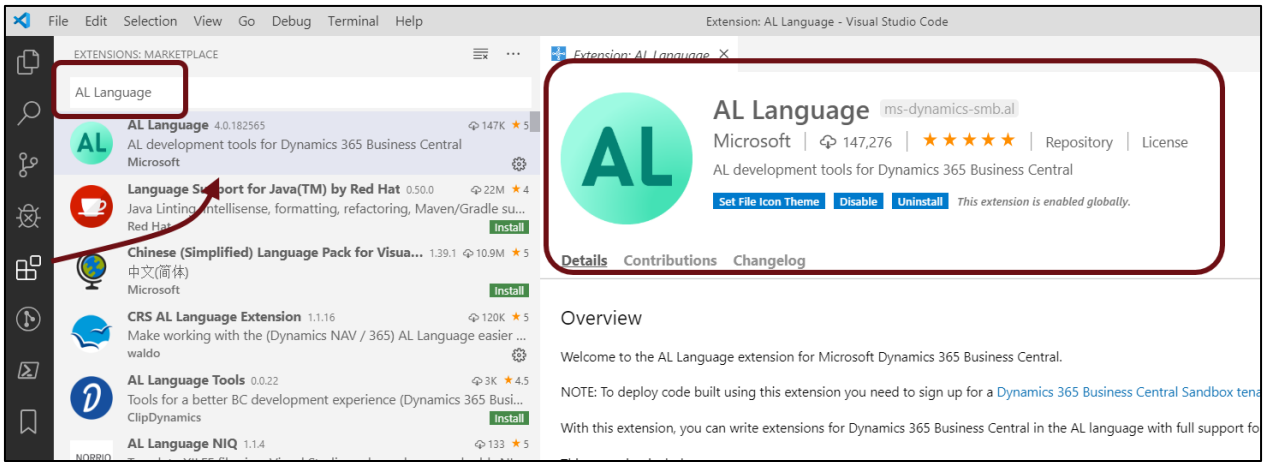


Now launch Visual Studio Code:



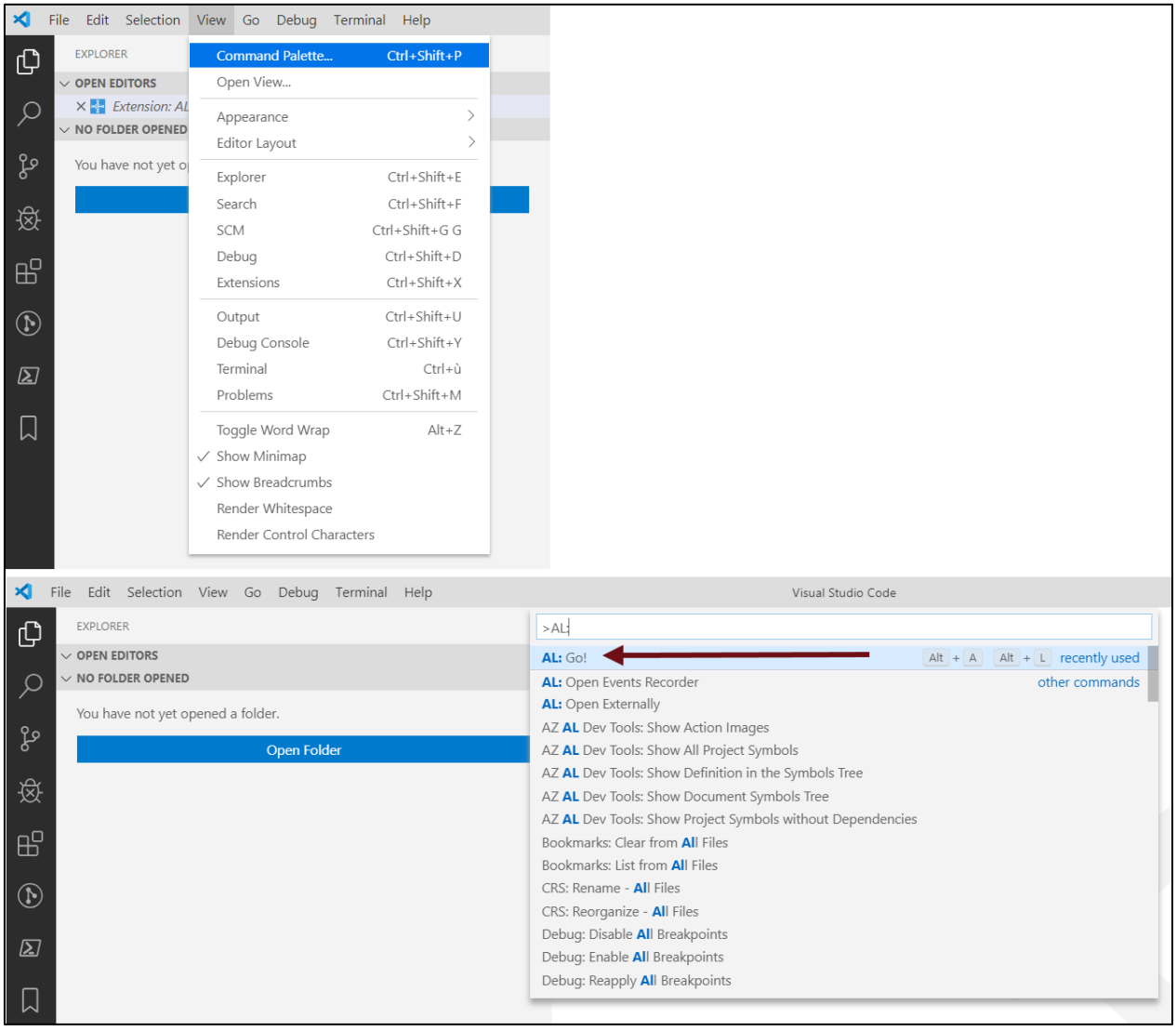


In Visual Studio Code, verify in the extensions window that the AL extension is available:

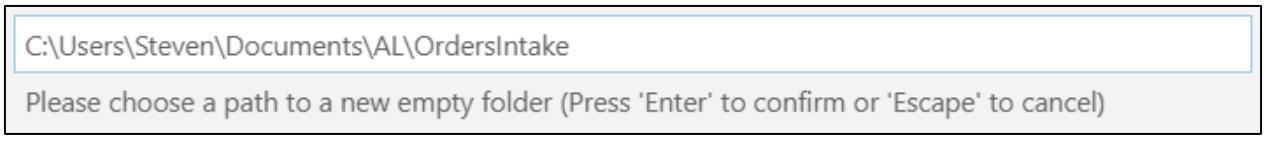


If not, then install it. You might need to reload Visual Studio Code.

Now create a new project, using **Ctrl+Shft+P** and **AL:Go!:**

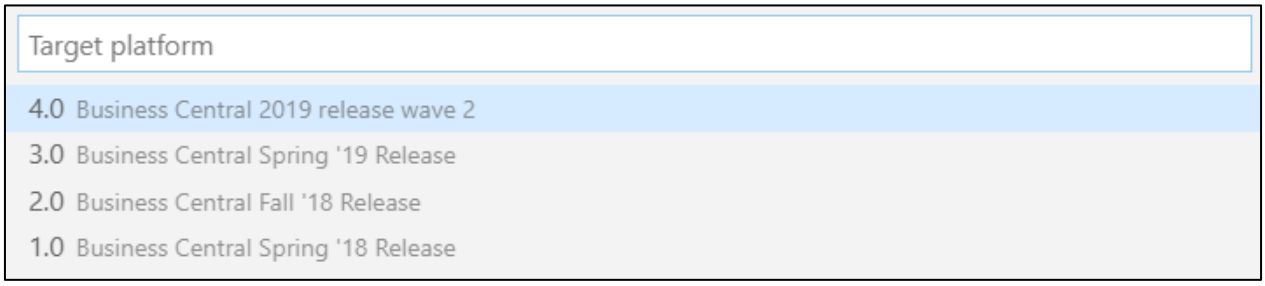


C:\Users\Steven\Documents\AL\OrdersIntake

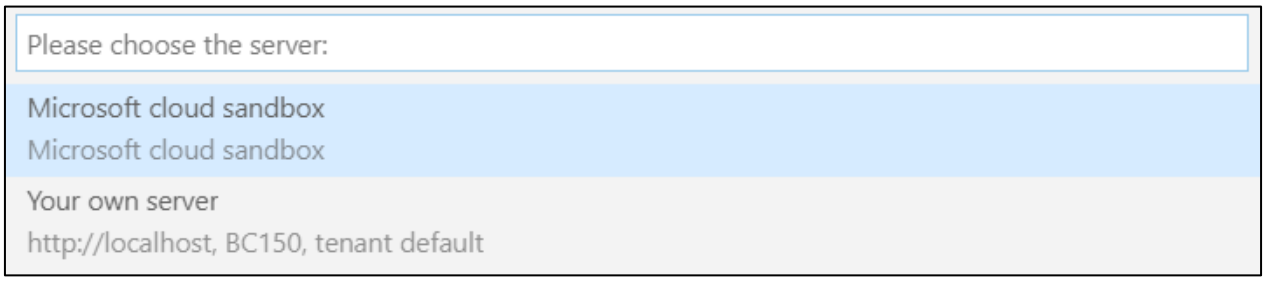


You can use another name and/or location if you prefer.

Select target platform 4.0:

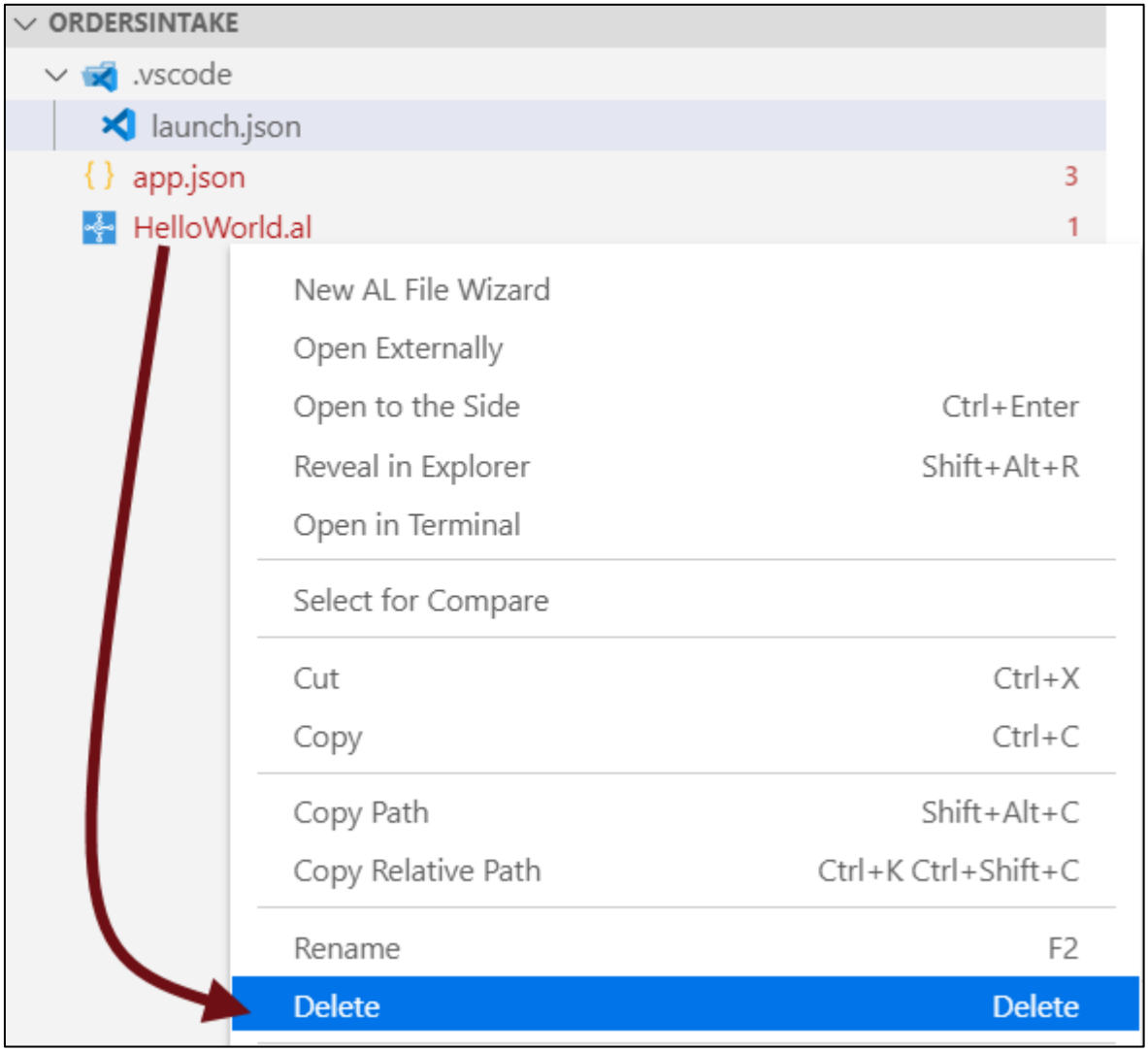


Then select Microsoft cloud sandbox:

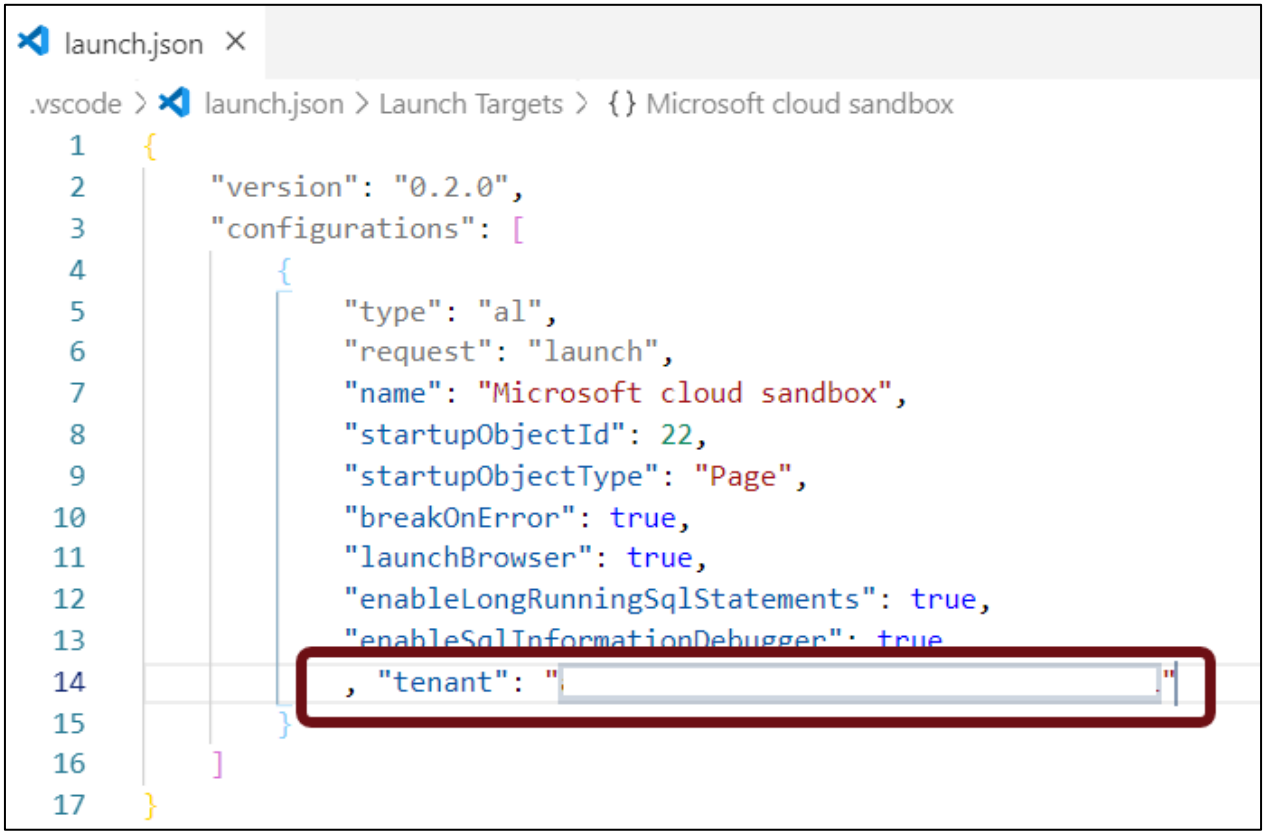


A project is created.

Now delete the HelloWorld.al file:

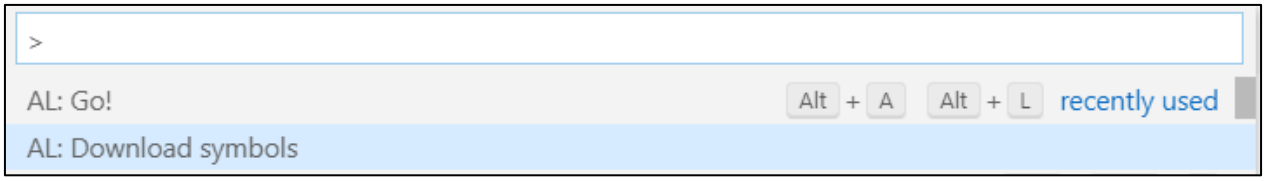


In the launch.json file enter the following information:

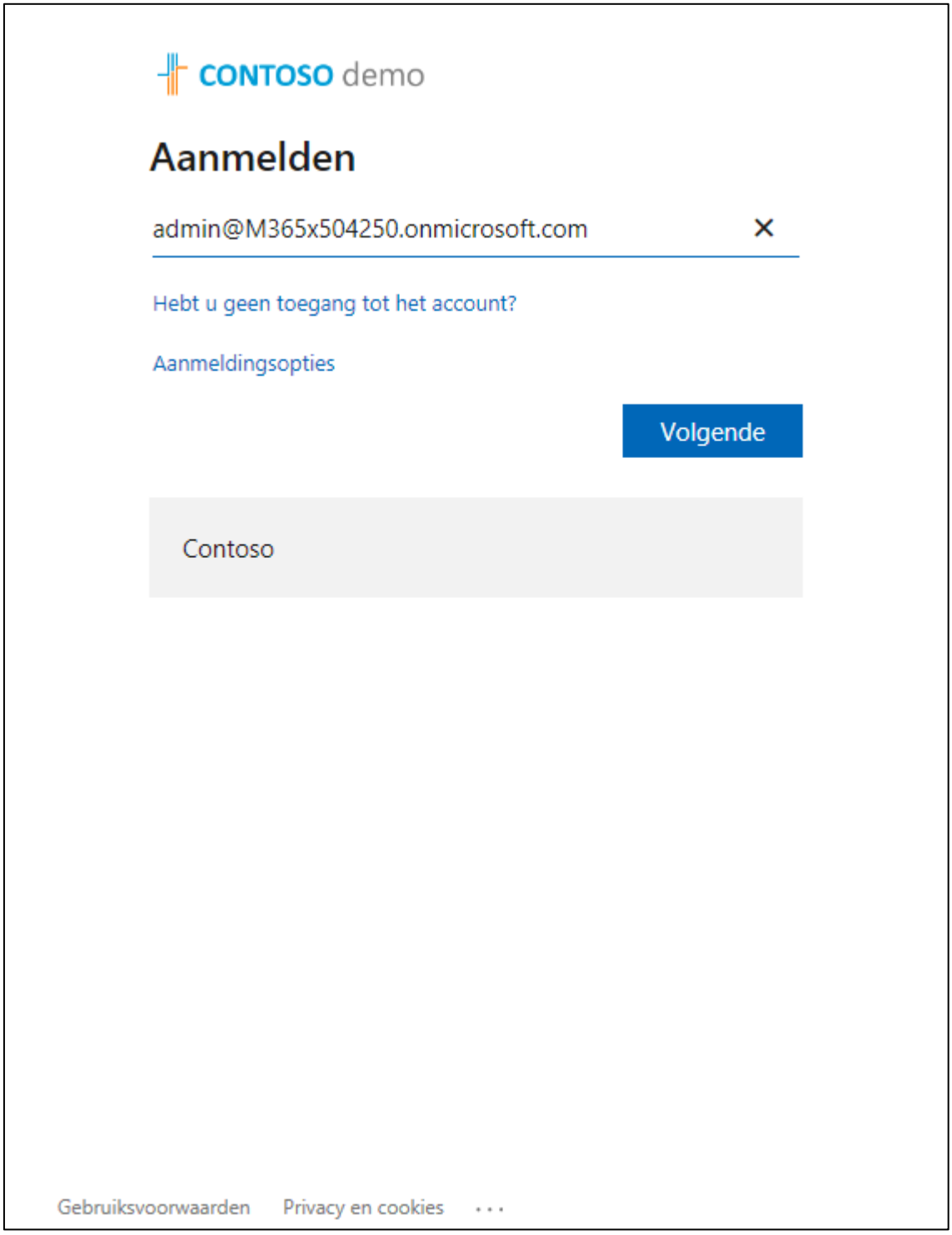


Remember, this is the tenant id you copied from the help page.

Now, in the command pallet, select AL:Download symbols:

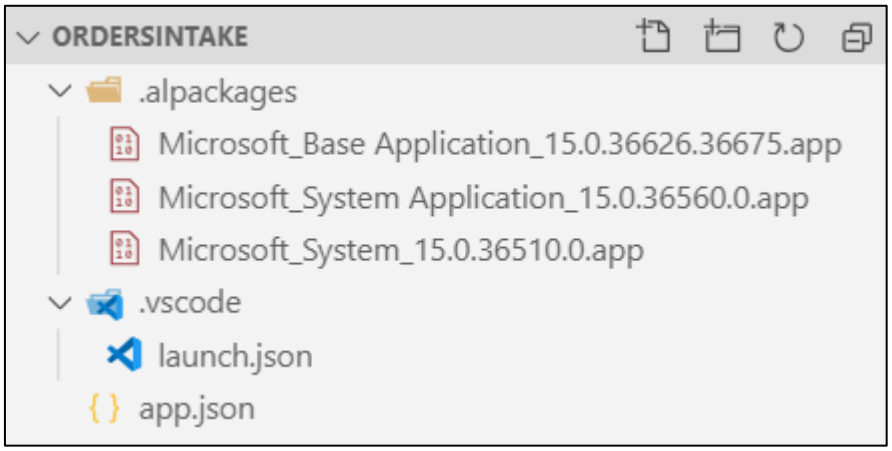


Enter your login & pw when asked for:

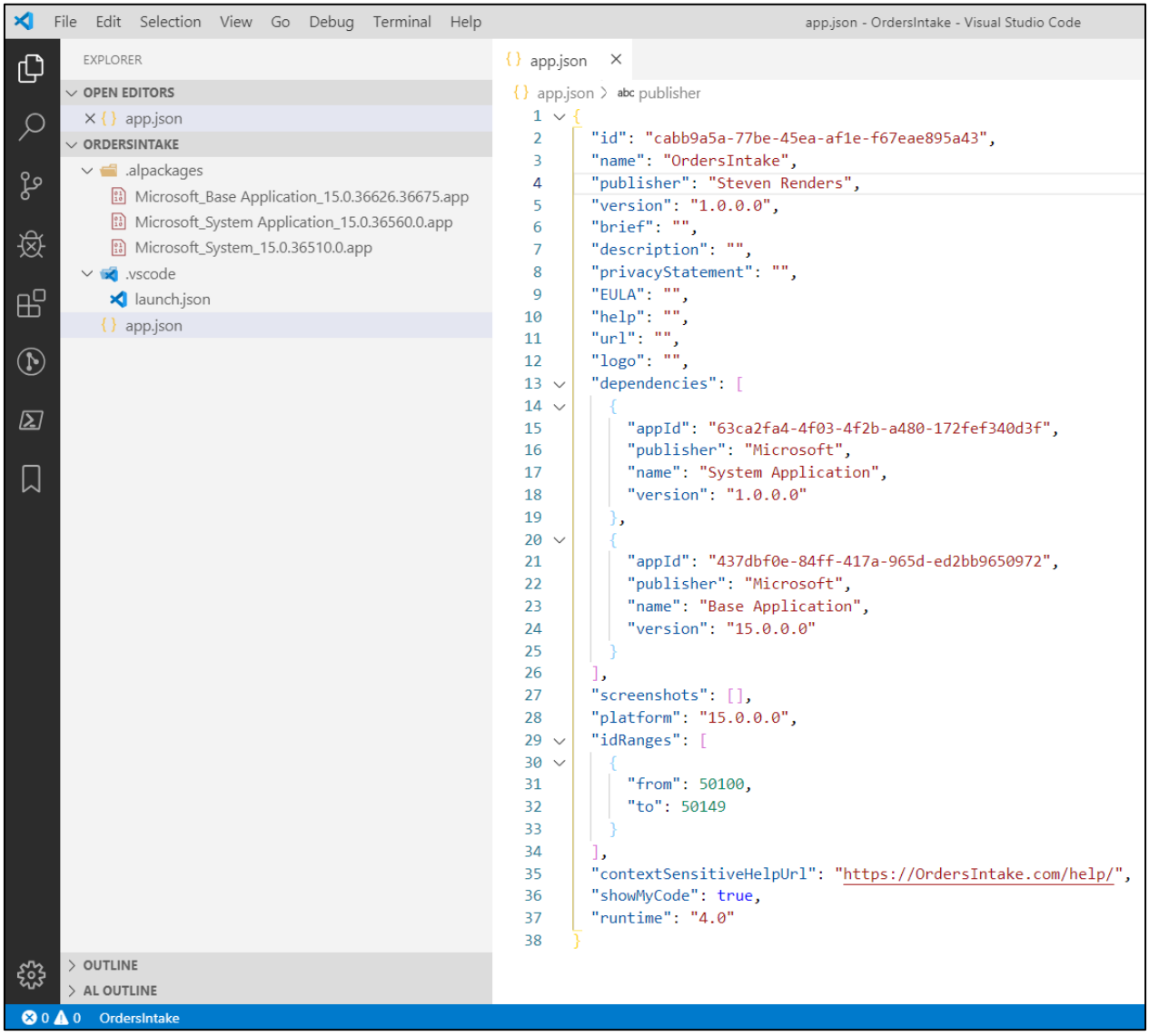


Now the system and base application files will download. This can take several minutes.

After they have been downloaded, you can see them in the .alpackages folder:



If desired, in the app.json file you can enter a name and publisher for your extension:

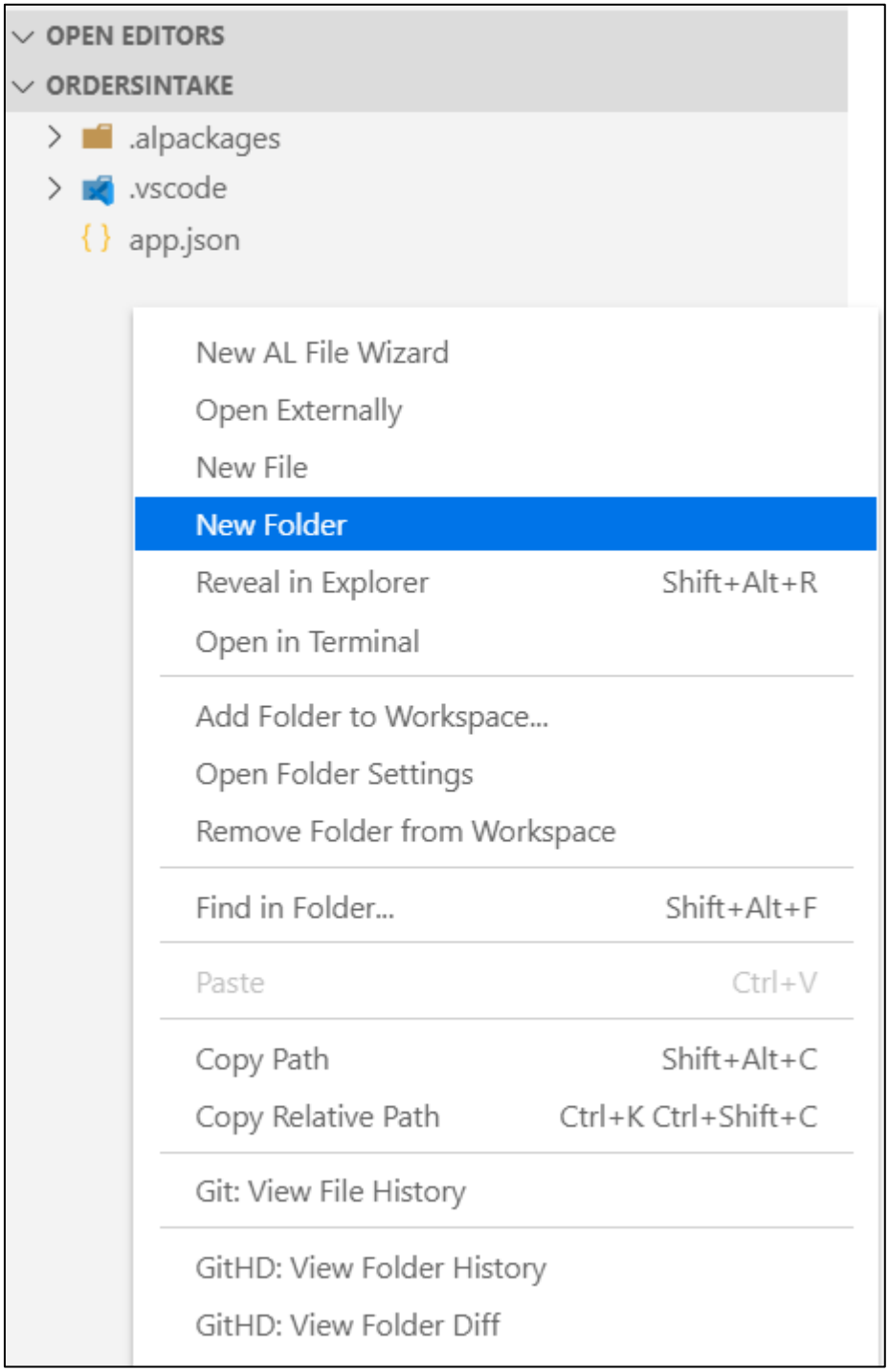


We are now ready to add queries to the extension.

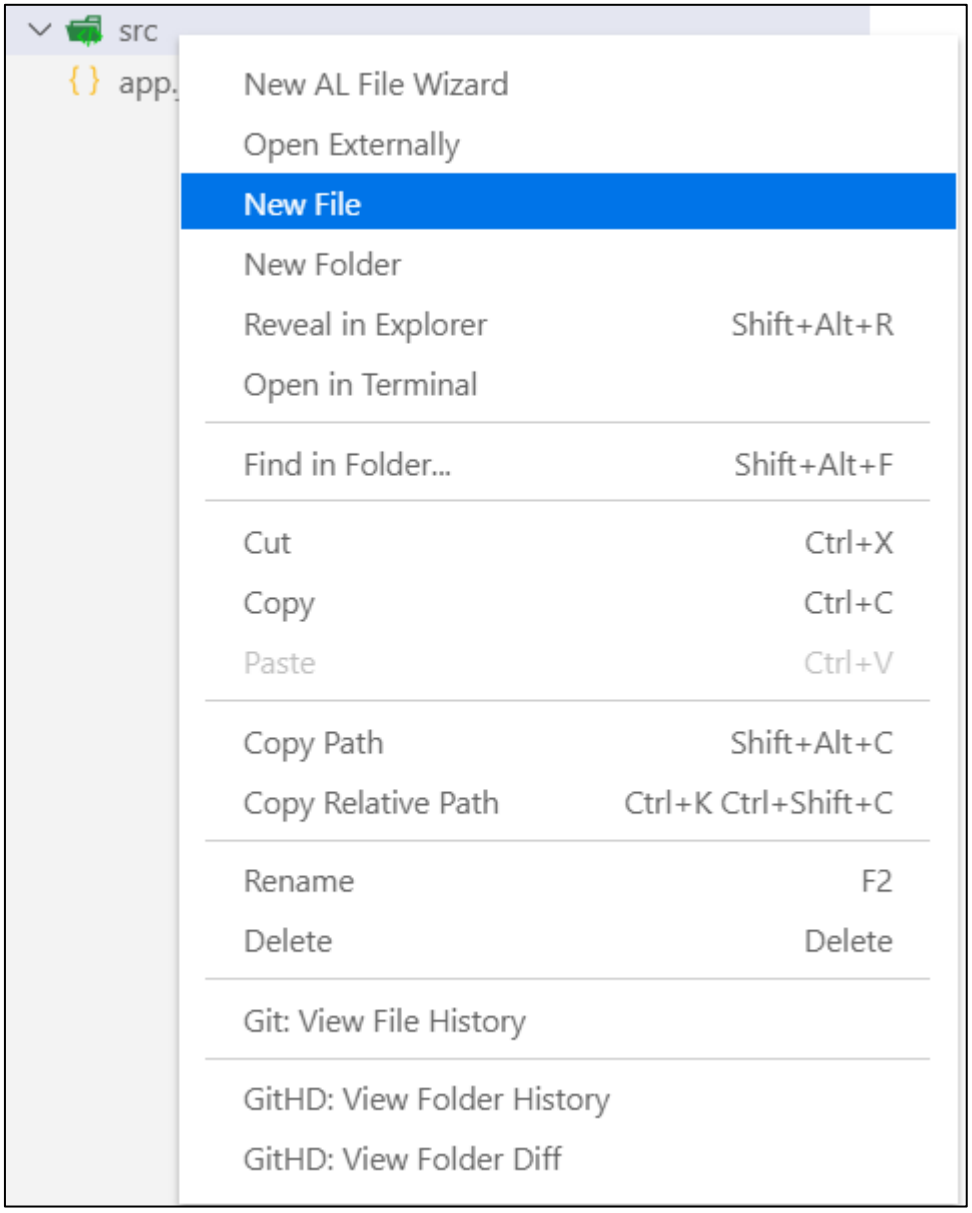
Add queries to the extension

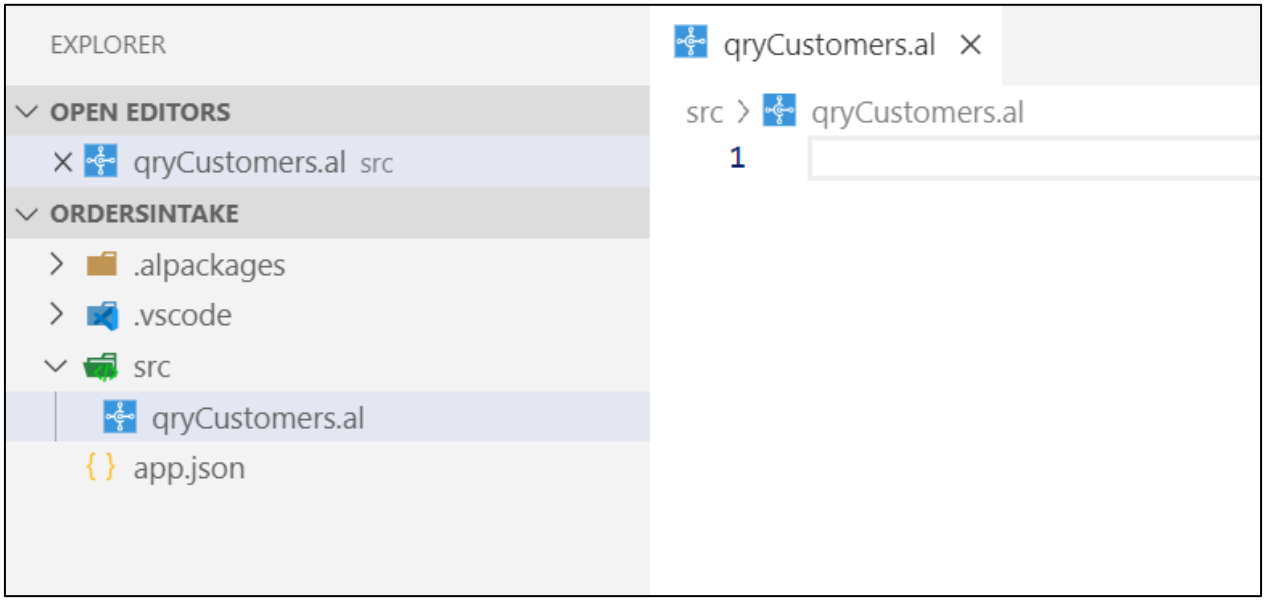
In this extension we will add 3 queries: **qryCustomers**, **qryItems** and **qryOrderIntake**.

Create a new folder named: **src**

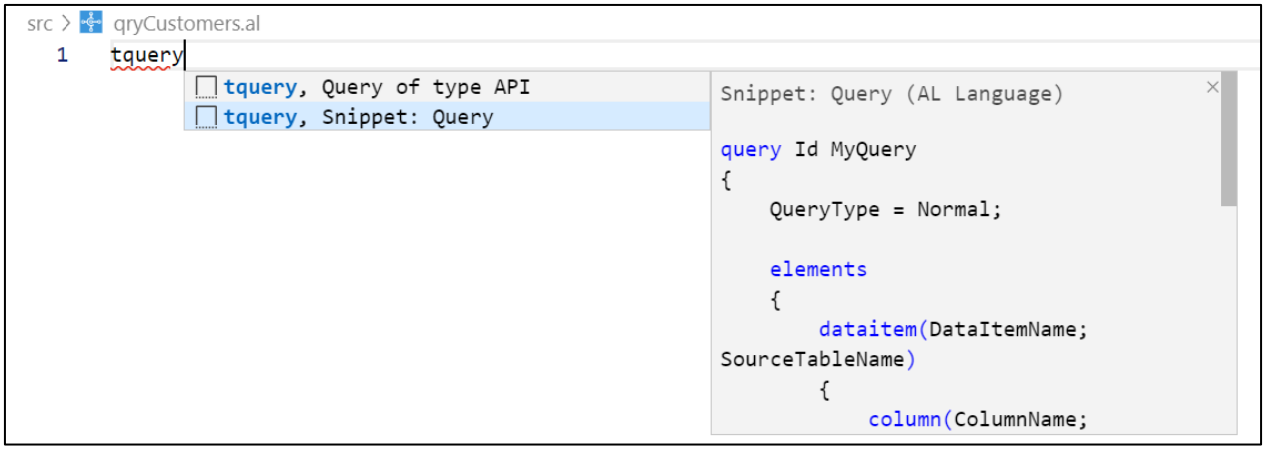


Add a new file to the folder and name it: **qryCustomers.al**:


src
app.
New AL File Wizard
Open Externally
New File
New Folder
Reveal in Explorer Shift+Alt+R
Open in Terminal
Find in Folder... Shift+Alt+F
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Copy Path Shift+Alt+C
Copy Relative Path Ctrl+K Ctrl+Shift+C
Rename F2
Delete Delete
Git: View File History
GitHD: View Folder History
GitHD: View Folder Diff



Use the **tquery** snippet to add a skeleton for a query object:



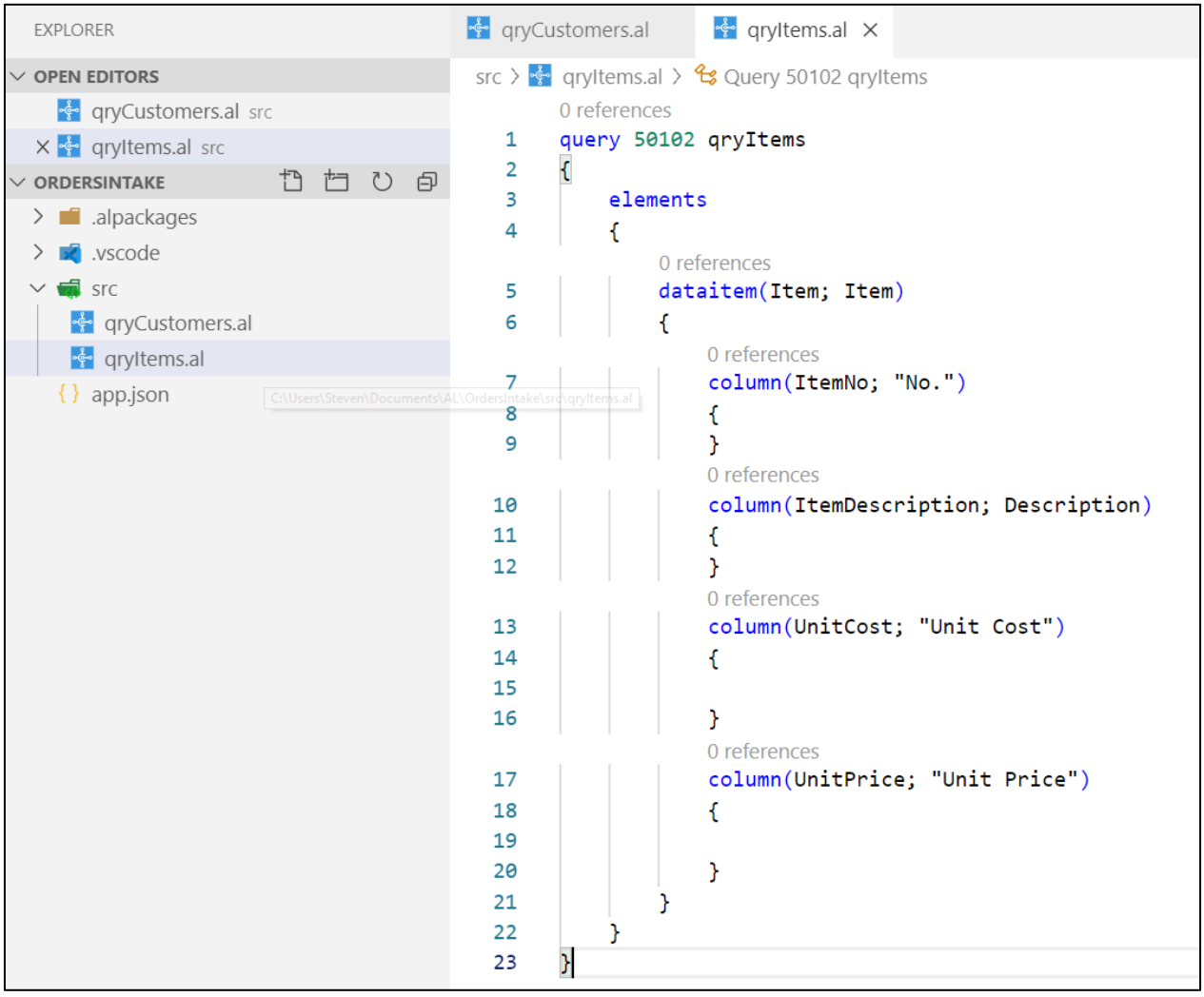
```
src > qryCustomers.al > Query 0 Id
    0 references
1   query Id MyQuery
2   {
3       QueryType = Normal;
4
5       elements
6       {
            0 references
7           dataitem(DataItemName; SourceTableName)
8           {
                0 references
9               column(ColumnName; SourceFieldName)
10              {
11
12              }
                0 references
13              filter(FilterName; SourceFieldName)
14              {
15
16              }
17          }
18      }
19
20      var
            0 references
21          myInt: Integer;
22
23      trigger OnBeforeOpen()
24      begin
25
26      end;
27  }
```

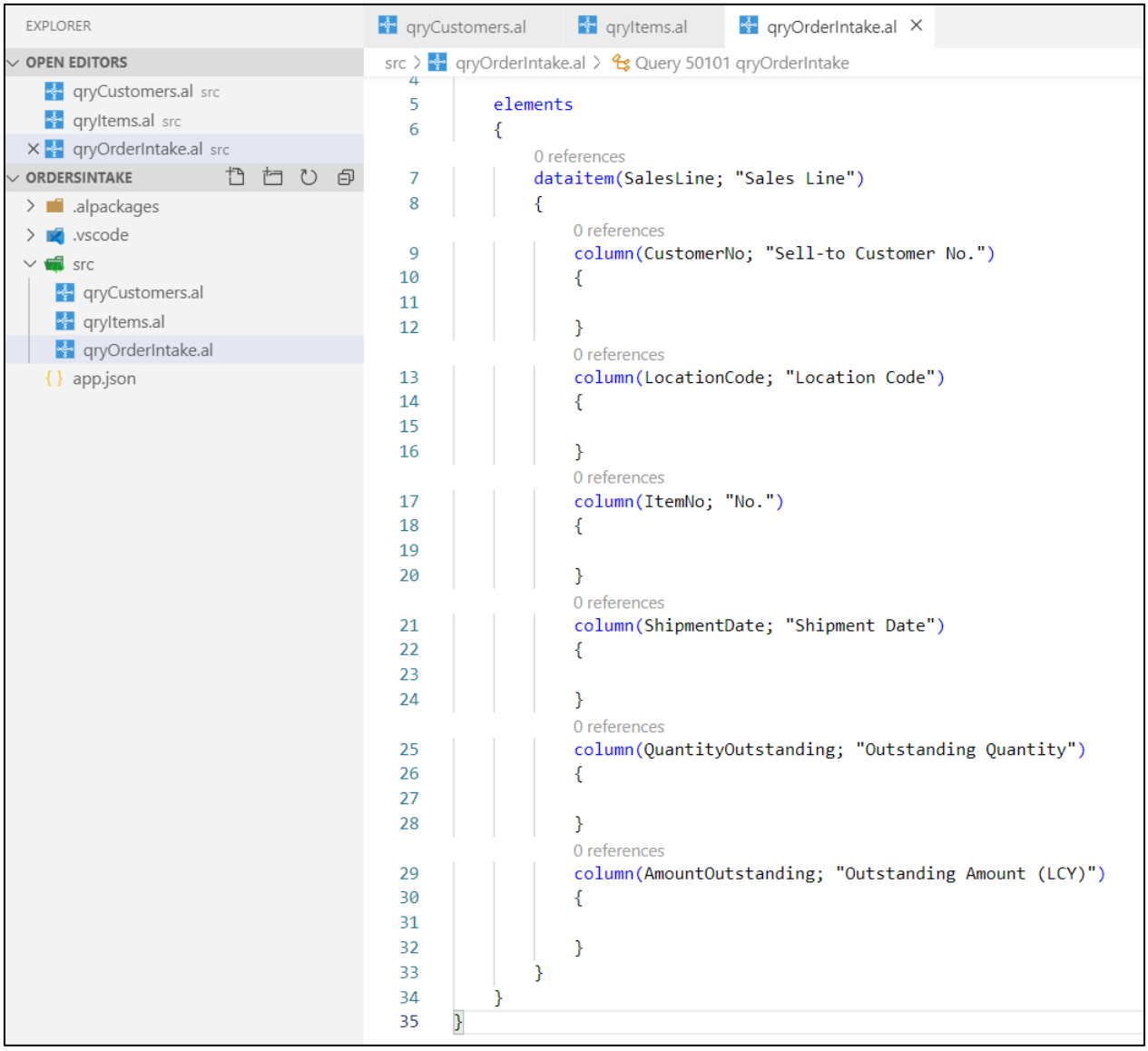
Now change the **Id** and **Name** and **content** to the following:

src > qryCustomers.al > Query 50100 qryCustomers

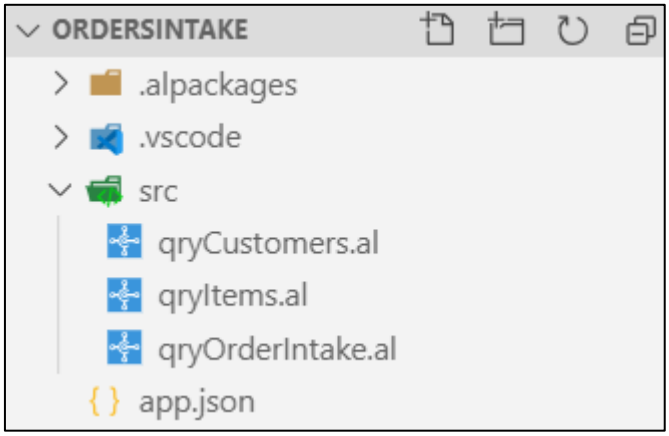
```
0 references
query 50100 qryCustomers
{
    elements
    {
        0 references
        dataitem(Customer; Customer)
        {
            0 references
            column(CustomerNo; "No.")
            {
            }
            0 references
            column(CustomerName; Name)
            {
            }
            0 references
            column(City; City)
            {

            }
        }
    }
}
```

Then in the same way that you created the **qryCustomers**, add two more queries: **qryItems** and **qryOrderIntake** as follows:


EXPLORER
qryCustomers.al
qryItems.al
OPEN EDITORS
qryCustomers.al src
qryItems.al src
ORDERSINTAKE
.alpackages
.vscode
src
qryCustomers.al
qryItems.al
app.json
src > qryItems.al > Query 50102 qryItems
0 references
query 50102 qryItems
{
elements
{
0 references
dataitem(Item; Item)
{
0 references
column(ItemNo; "No.")
{
}
0 references
column(ItemDescription; Description)
{
}
0 references
column(UnitCost; "Unit Cost")
{
}
0 references
column(UnitPrice; "Unit Price")
{
}
}
}
}



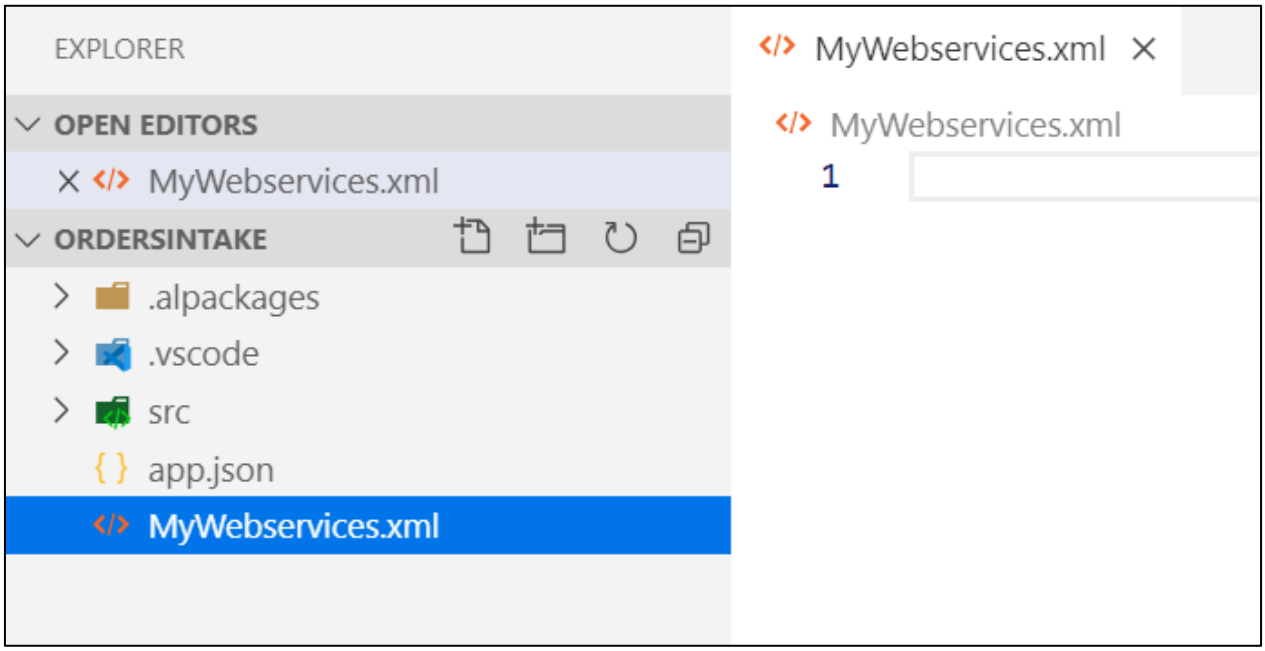
Your project now contains 3 queries:



Publish the queries as web services

To publish the queries as web services, when deploying the extension, you need to add a web services file to the project.

- Create a new (empty) file, with the name: **MyWebservices.xml**



In the xml file type, the following:

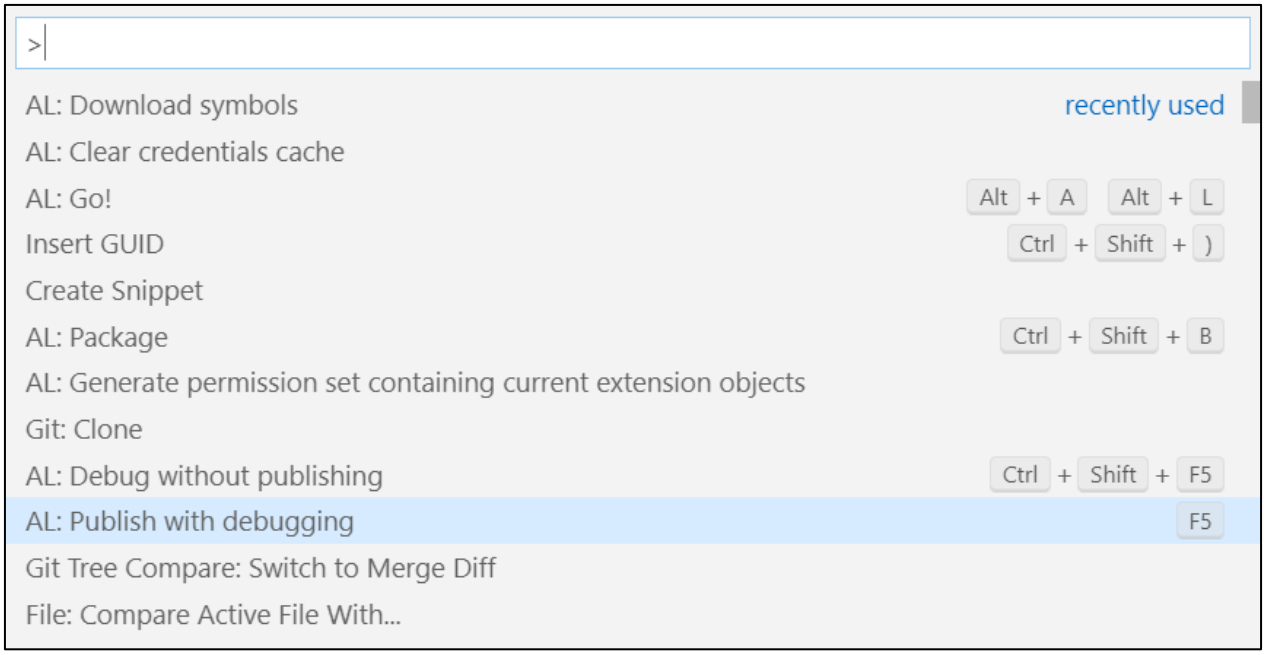
MyWebservices.xml

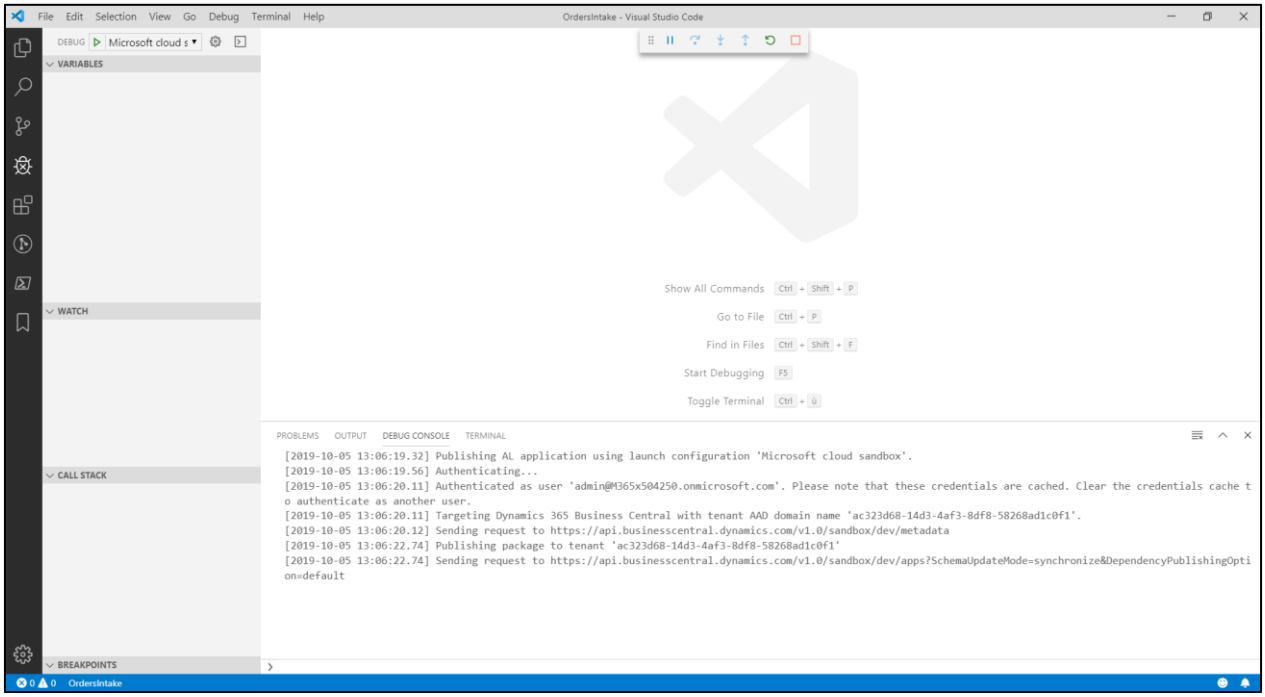
```xml
<?xml version="1.0" encoding="utf-8"?>
<ExportedData>
    <TenantWebServiceCollection>
        <TenantWebService>
            <ObjectType>Query</ObjectType>
            <ServiceName>qryCustomers</ServiceName>
            <ObjectID>50100</ObjectID>
            <Published>true</Published>
        </TenantWebService>
        <TenantWebService>
            <ObjectType>Query</ObjectType>
            <ServiceName>qryItems</ServiceName>
            <ObjectID>50102</ObjectID>
            <Published>true</Published>
        </TenantWebService>
         <TenantWebService>
            <ObjectType>Query</ObjectType>
            <ServiceName>qryOrderIntake</ServiceName>
            <ObjectID>50101</ObjectID>
            <Published>true</Published>
        </TenantWebService>
    </TenantWebServiceCollection>
</ExportedData>
```

Then in the **launch.json** file, change the **startupObjectId** to **810**:

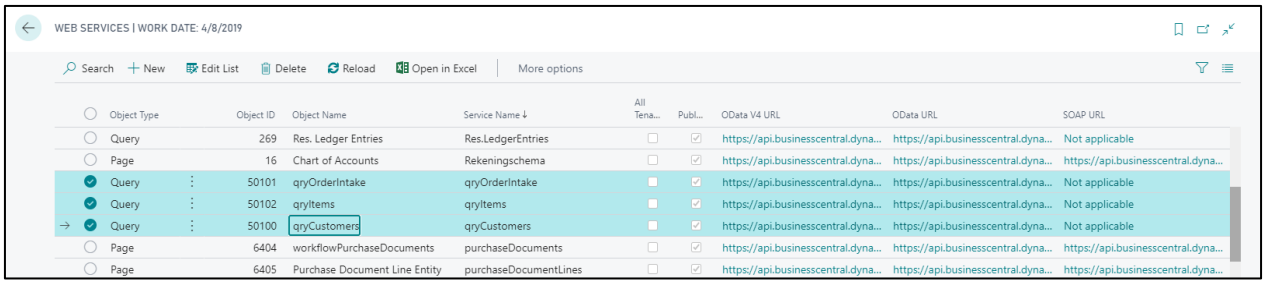
This will make sure that when you launch the extension, the web services page opens.

Now **Build** and **Publish** your extension, using the **F5** shortcut:





The application starts and opens the web services page:



Here you should now see the queries in your extension.

Obtain the ODATA web services URL

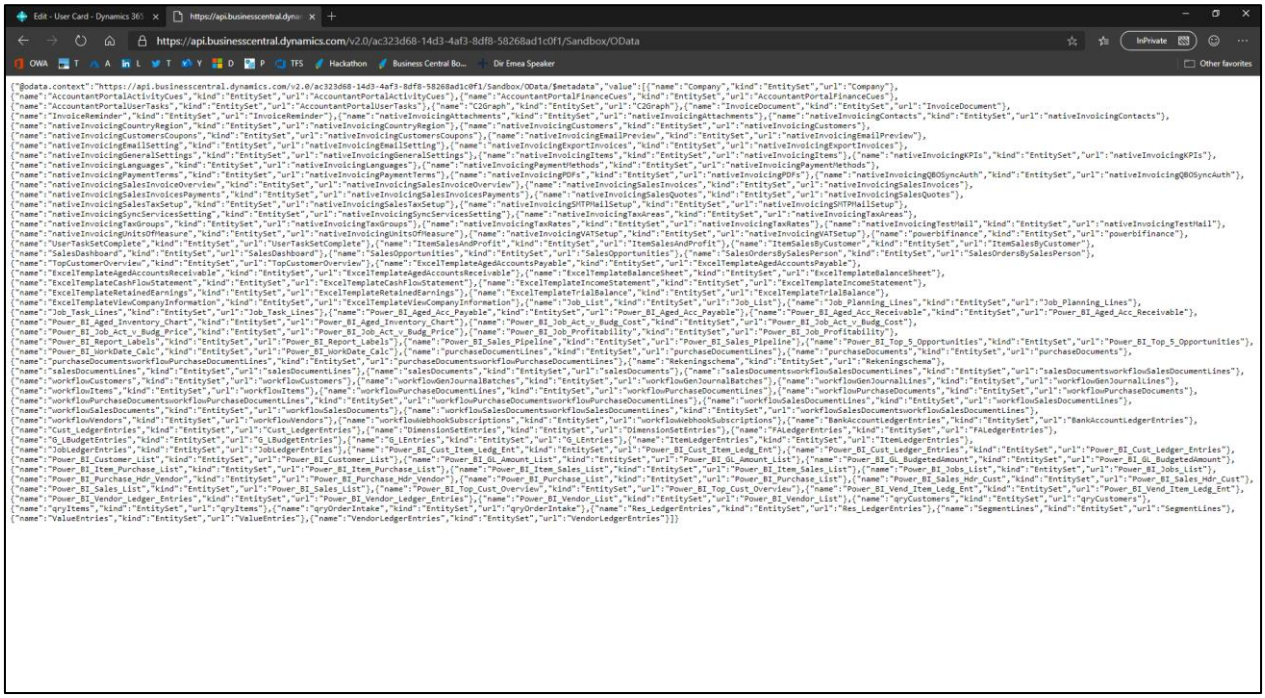
In the web services page, copy the **ODATA URL** of one of the queries. It should look like the following:

- https://api.businesscentral.dynamics.com/v2.0/ac323d68-14d3-4af3-8df8-58268ad1c0f1/Sandbox/OData/Company('CRONUS%20NL')/qryCustomers

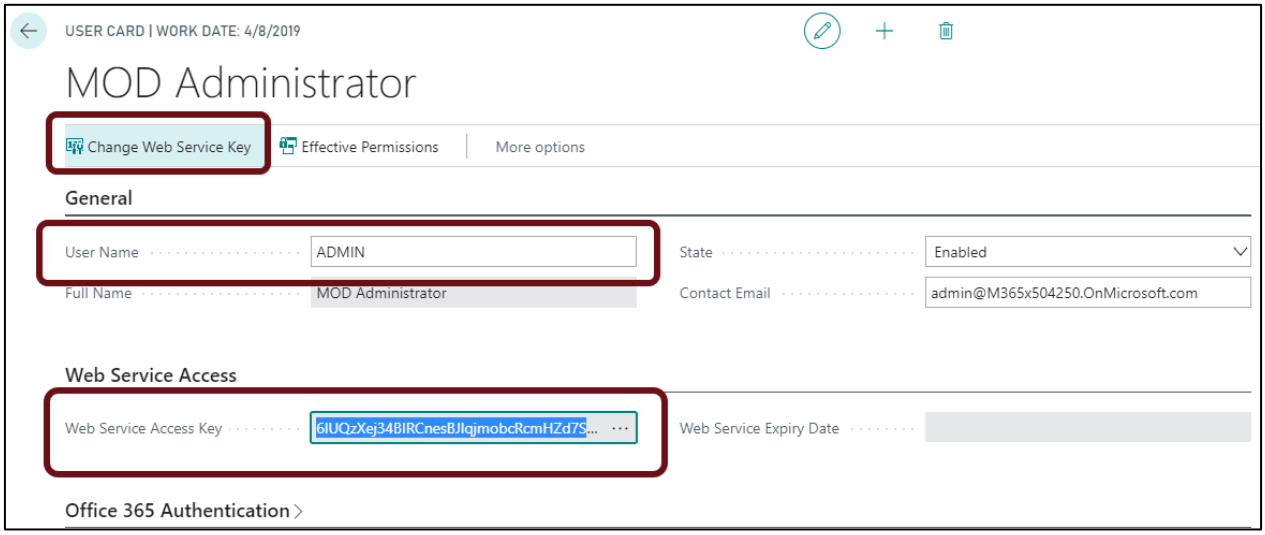
Remove everything after Company('CRONUS%20NL'), until your URL looks like the following:

https://api.businesscentral.dynamics.com/v2.0/ac323d68-14d3-4af3-8df8-58268ad1c0f1/Sandbox/OData/Company('CRONUS%20NL')

To test that it works, paste it into your browser:

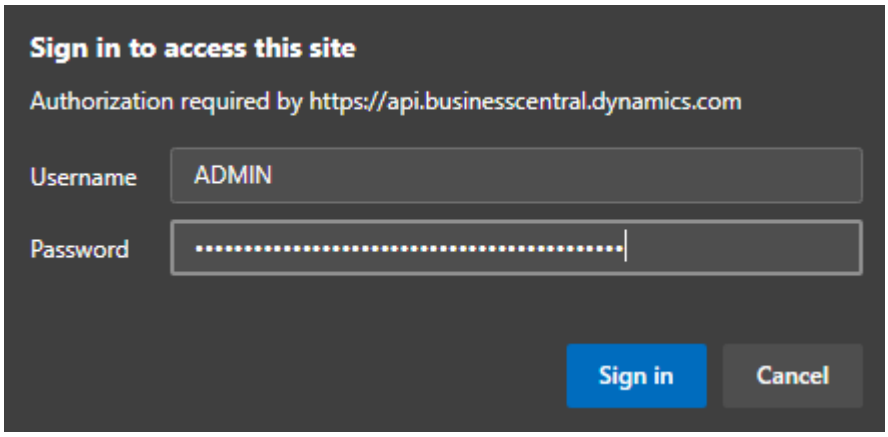


If you are prompted for credentials, then you first need to open your user card, generate and copy the Web Service Access Key:



Your credentials are:

User Name & Web Service Access Key.

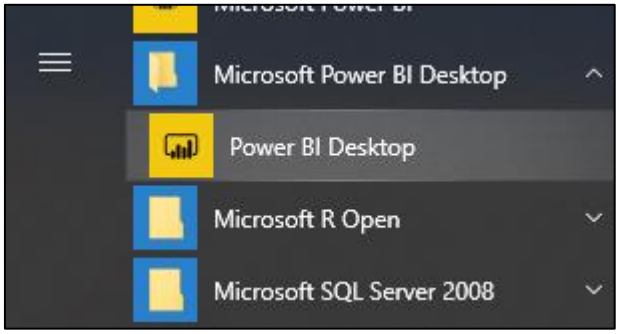


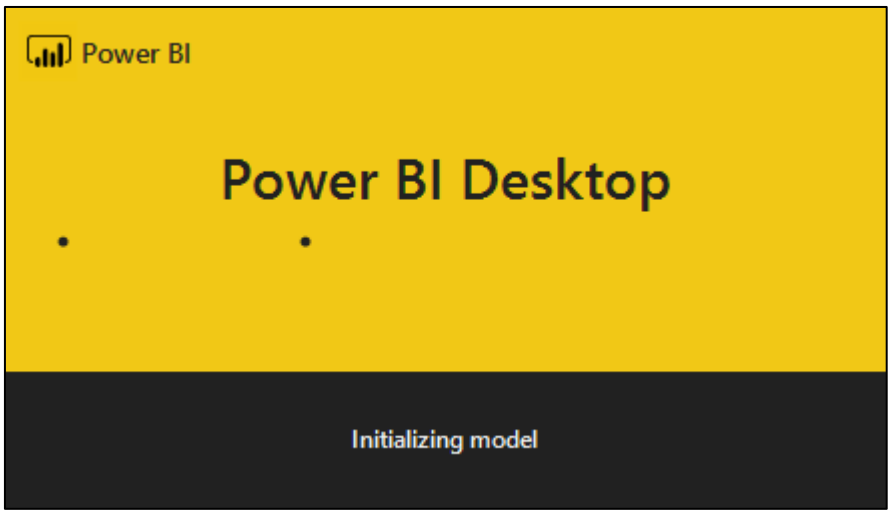
Copy them to notepad and double-check that the web service access key ends with an = sign.

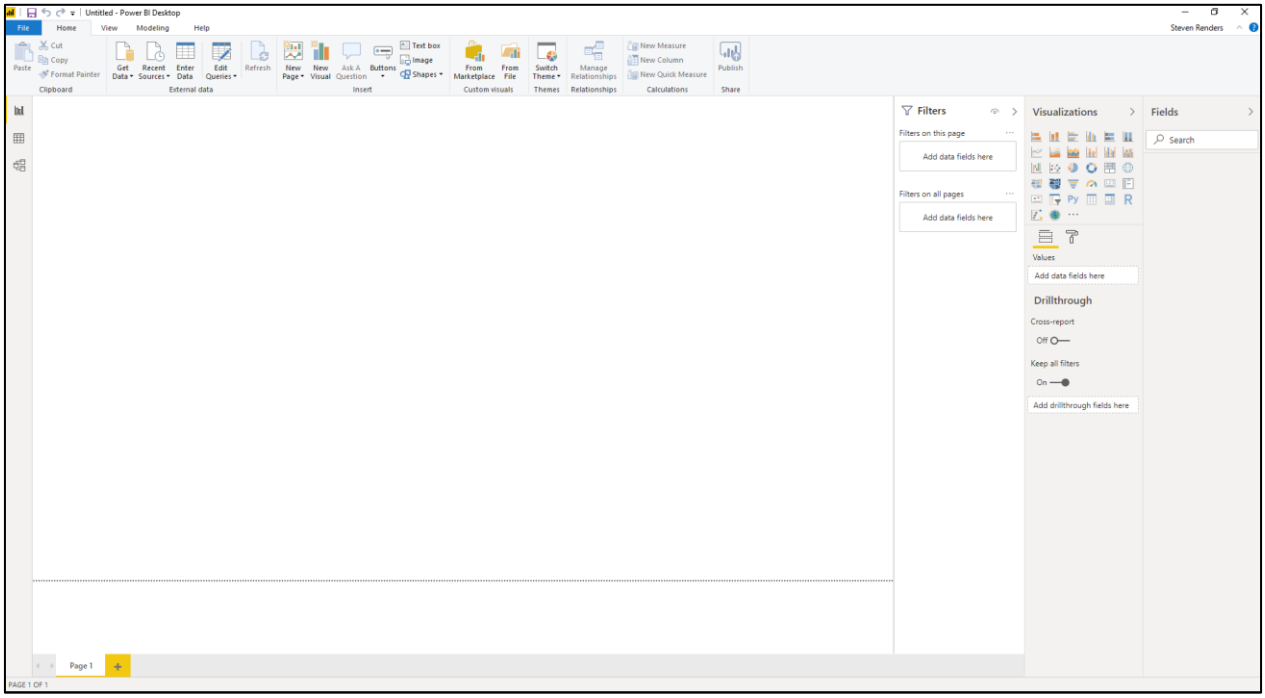
Import the ODATA queries in Power BI Desktop

Now that our queries have been published as ODATA web services and we know the URL on how to reach them, we are ready to create a report in Power BI Desktop.

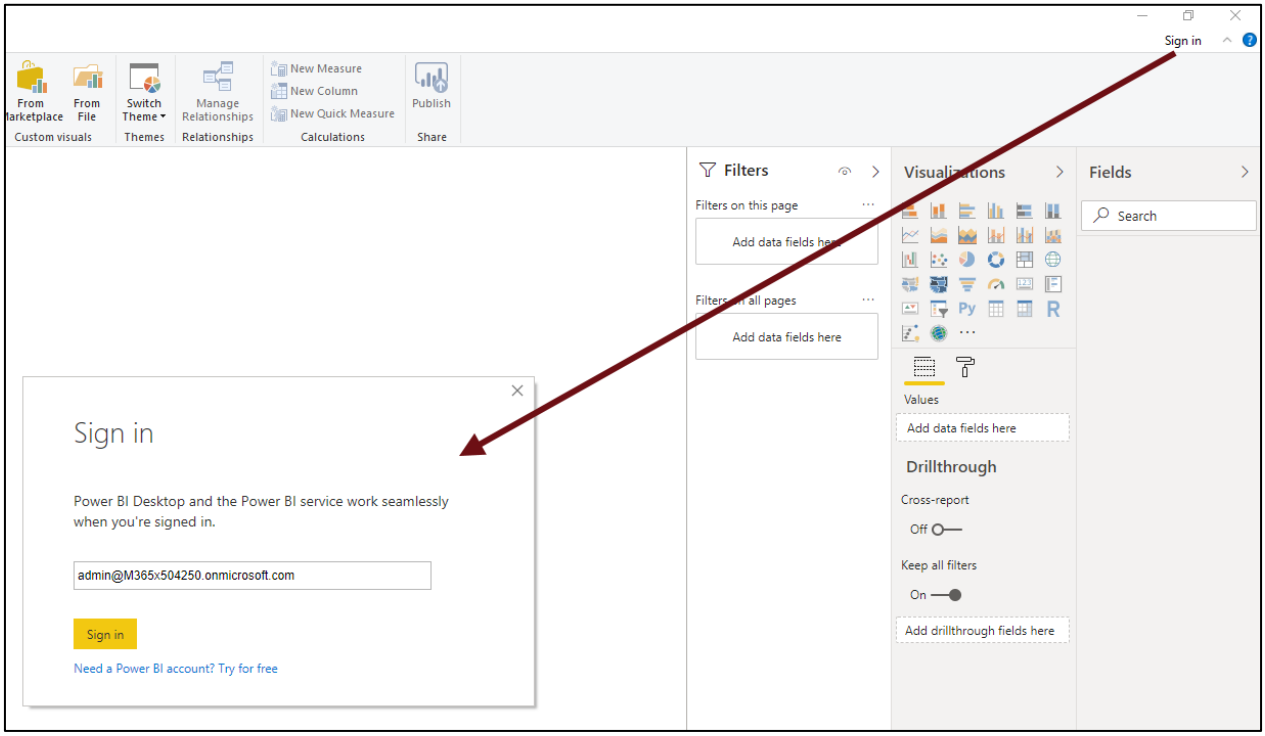
Launch Power BI Desktop:

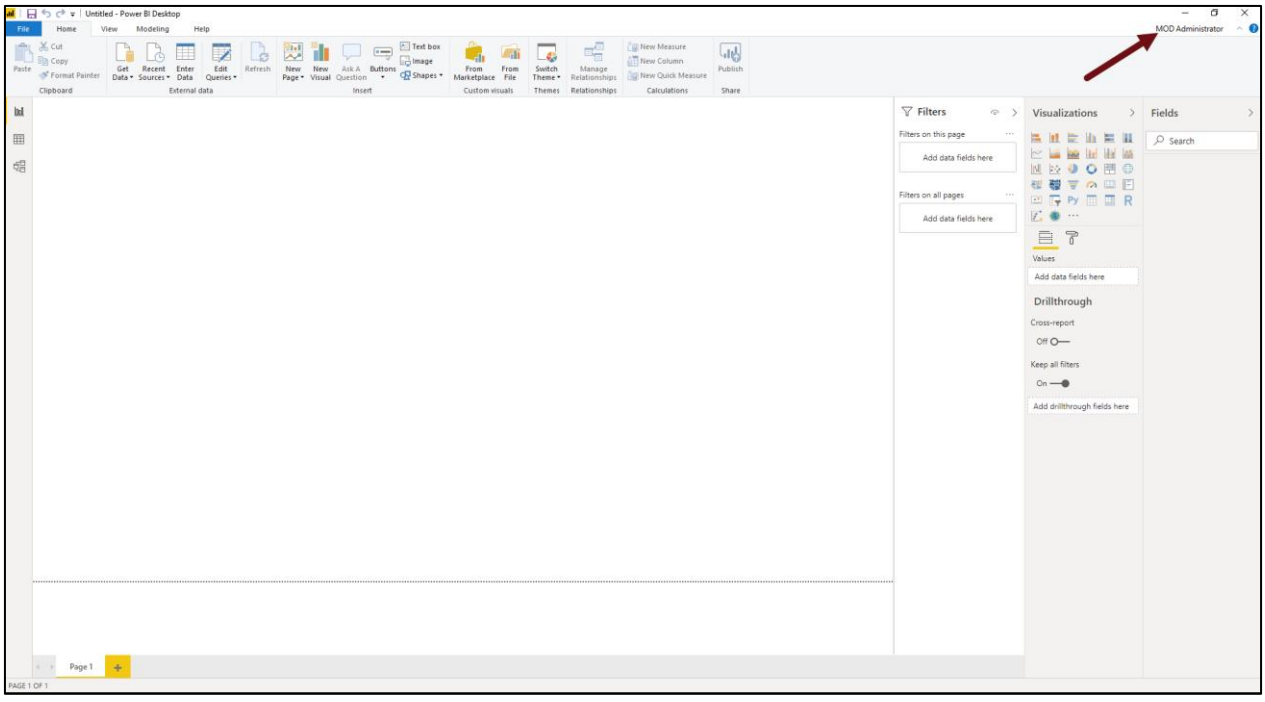

Microsoft Power BI Desktop
Power BI Desktop
Microsoft R Open
Microsoft SQL Server 2008


Power BI
Power BI Desktop
Initializing model

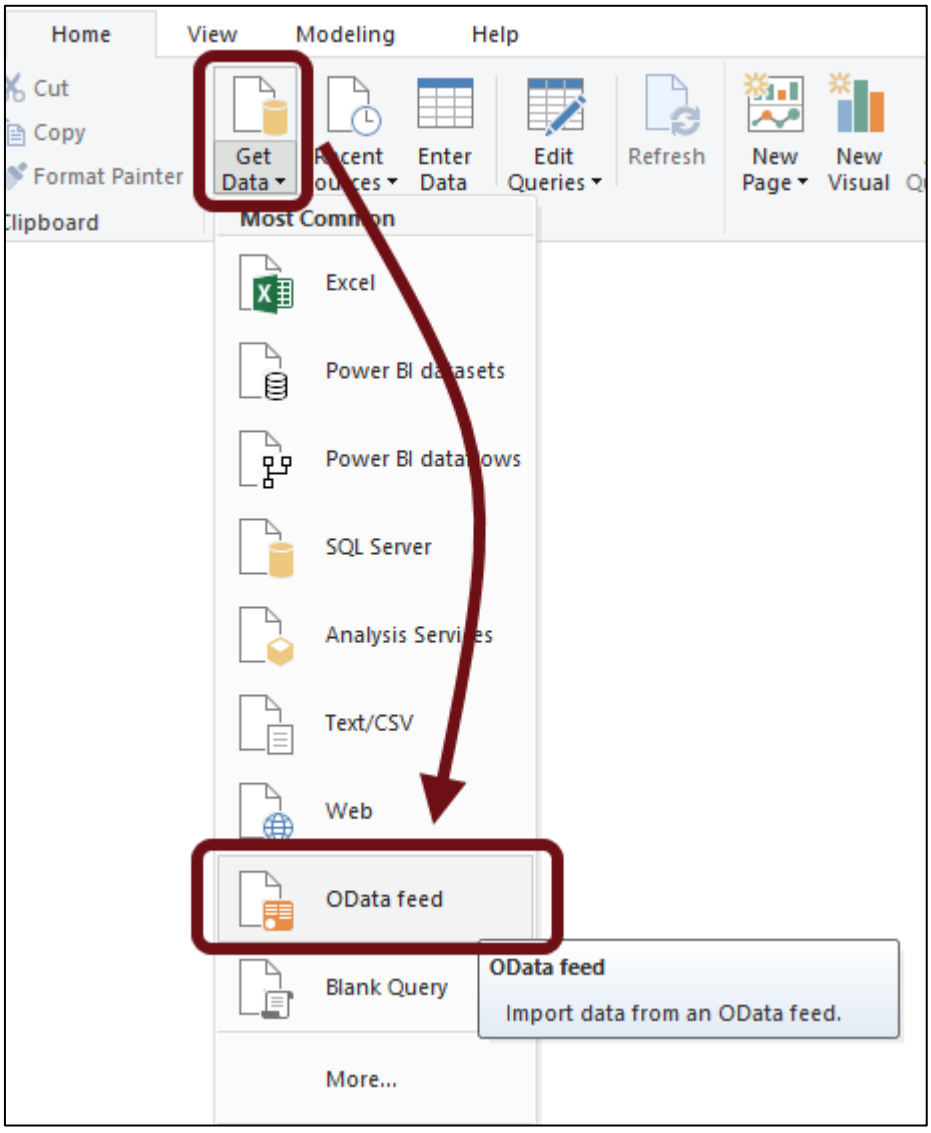


Select the **Sign In** link and enter your email and password you used when registering on the Power BI website:


Sign in
From Marketplace
From File
Switch Theme
Manage Relationships
New Measure
New Column
New Quick Measure
Publish
Custom visuals
Themes
Relationships
Calculations
Share
Filters
Filters on this page
Add data fields here
Filters on all pages
Add data fields here
Visualizations
Values
Add data fields here
Drillthrough
Cross-report
Off
Keep all filters
On
Add drillthrough fields here
Fields
Search
Sign in
Power BI Desktop and the Power BI service work seamlessly when you're signed in.
admin@M365x504250.onmicrosoft.com
Sign in
Need a Power BI account? Try for free

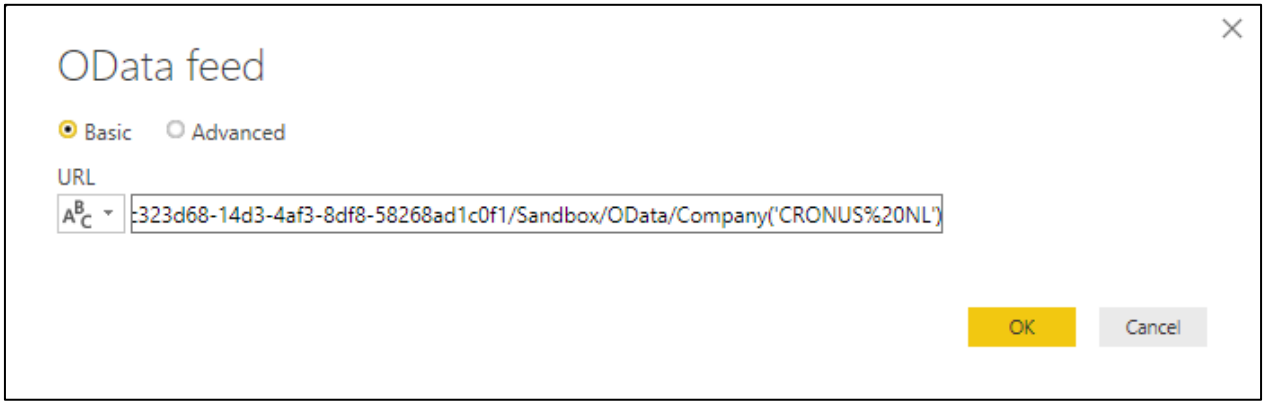

MOD Administrator
Filters
Visualizations
Fields
Page 1

Now select **Get Data**, **ODATA Feed**:

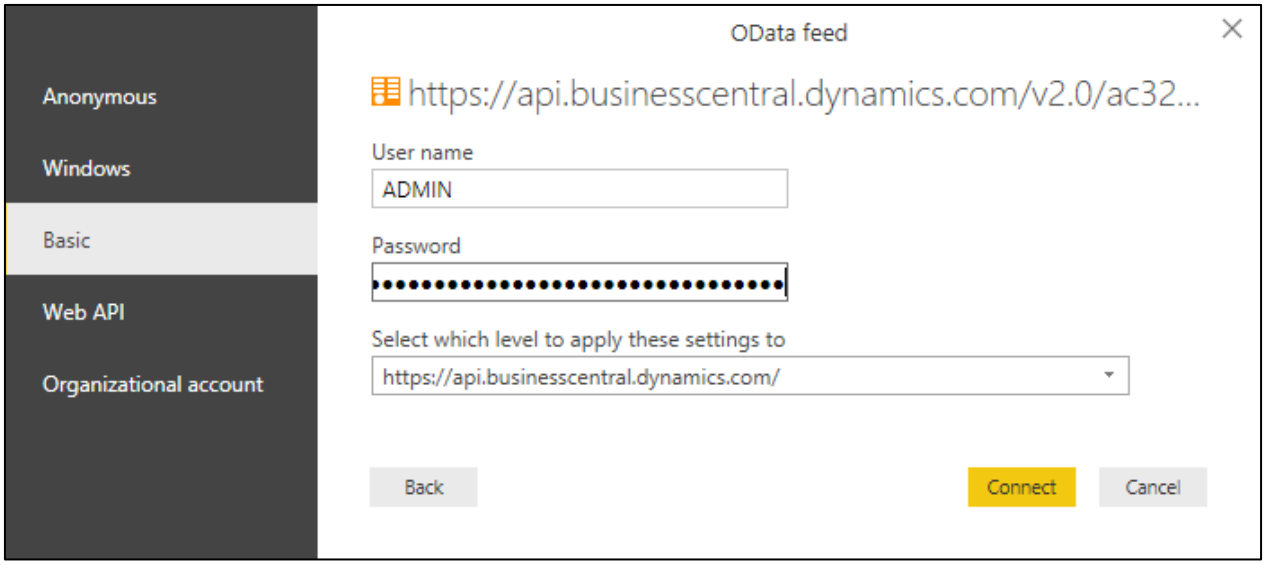


In the popup window, paste the ODATA **Url**:

https://api.businesscentral.dynamics.com/v2.0/ac323d68-14d3-4af3-8df8-58268ad1c0f1/Sandbox/OData/Company('CRONUS%20NL')

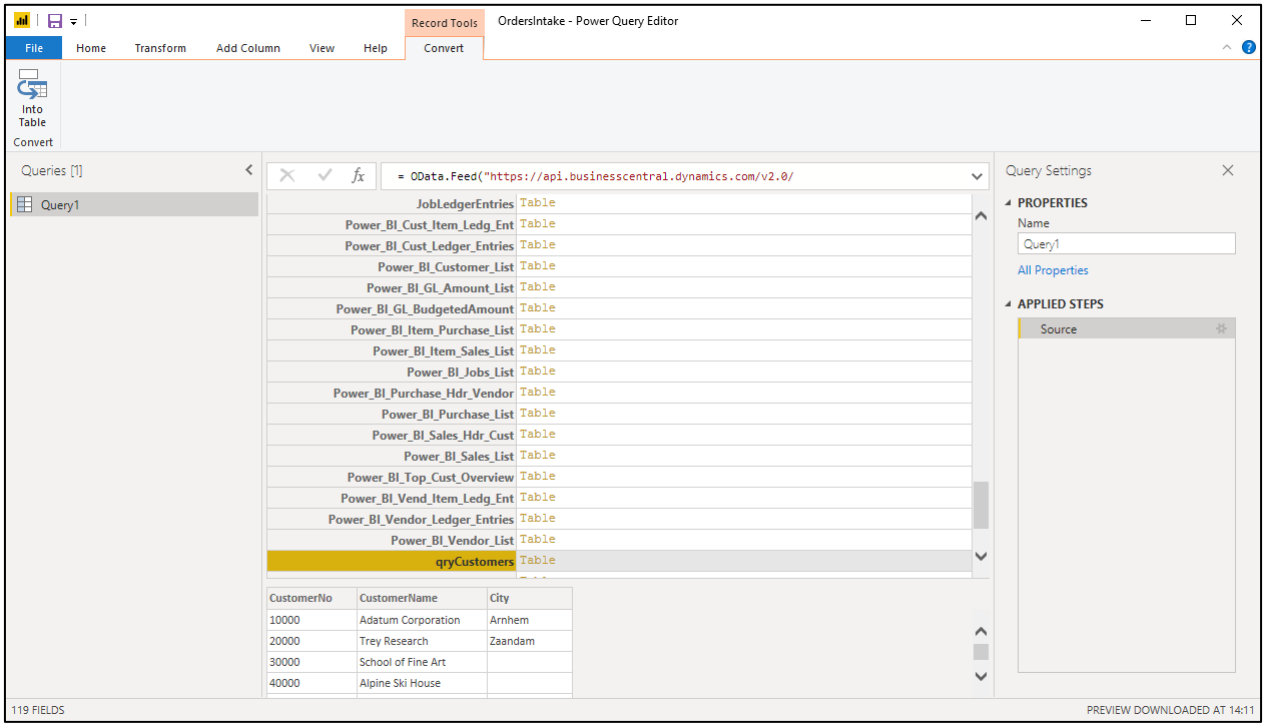


Then enter your credentials. Select **Basic**:

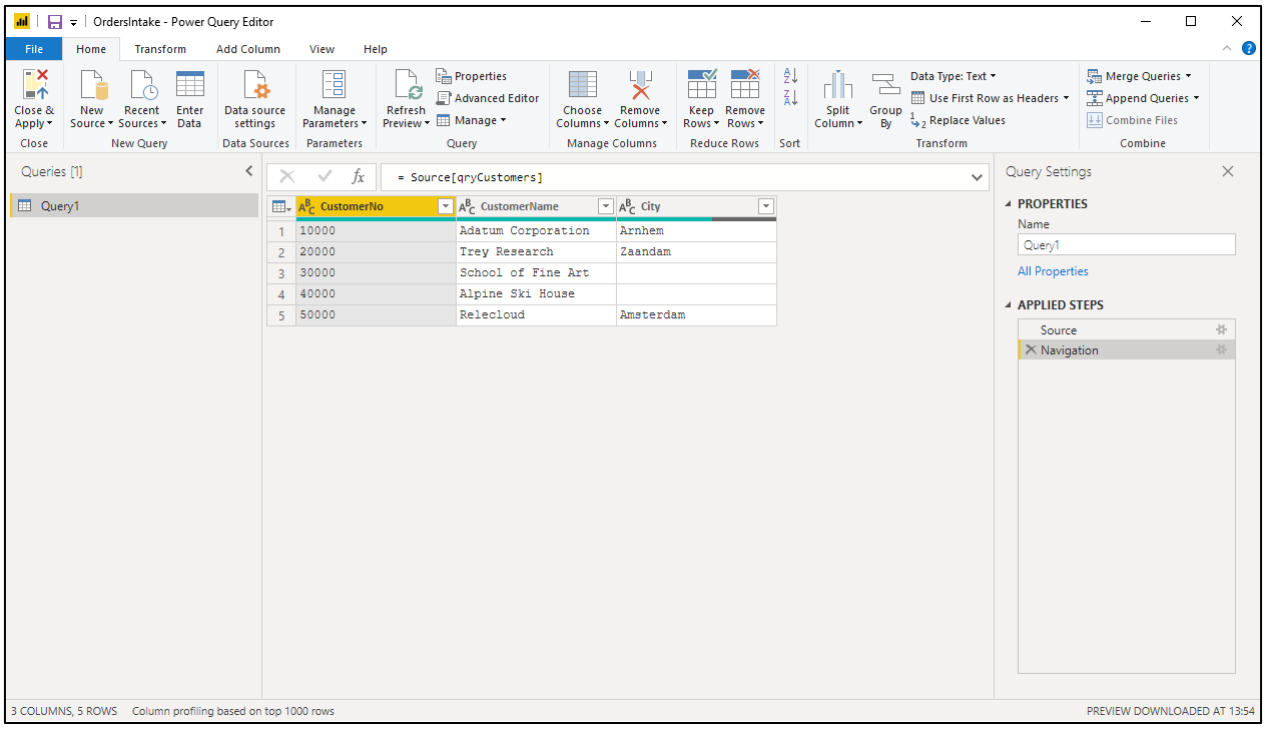


These are your Business Central username and web service access key.

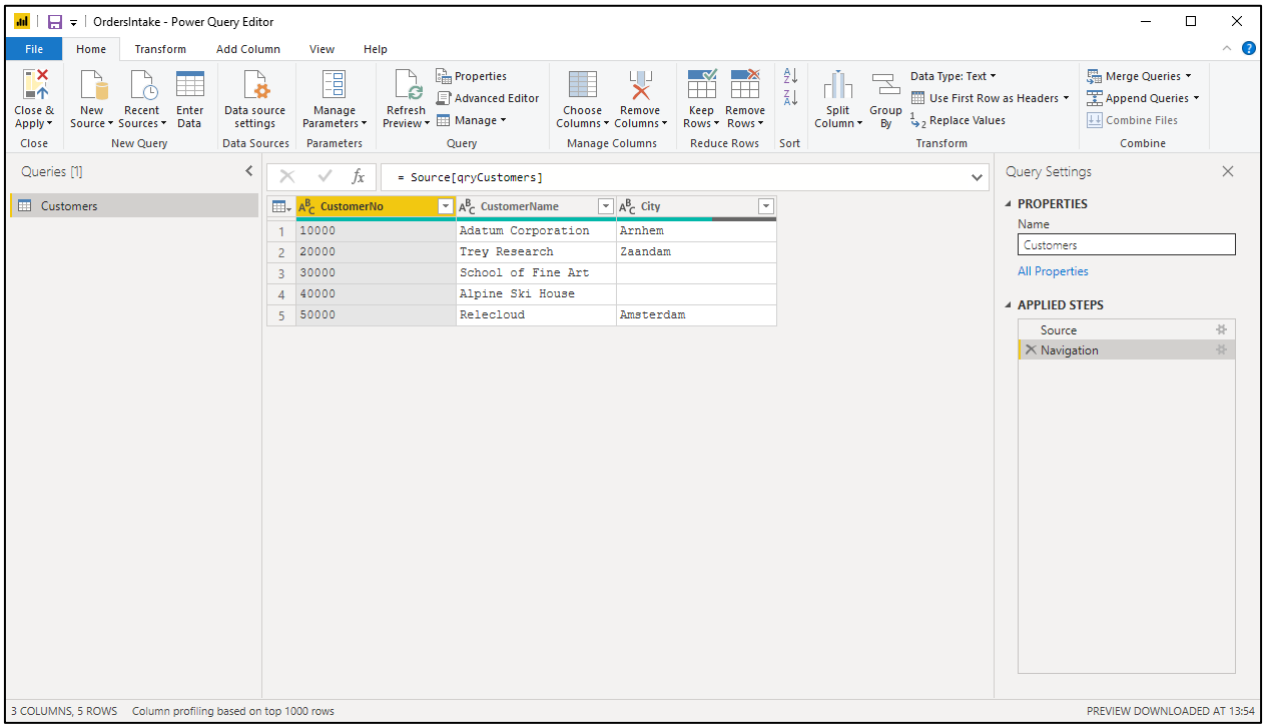
Select **Connect** and the following window opens:



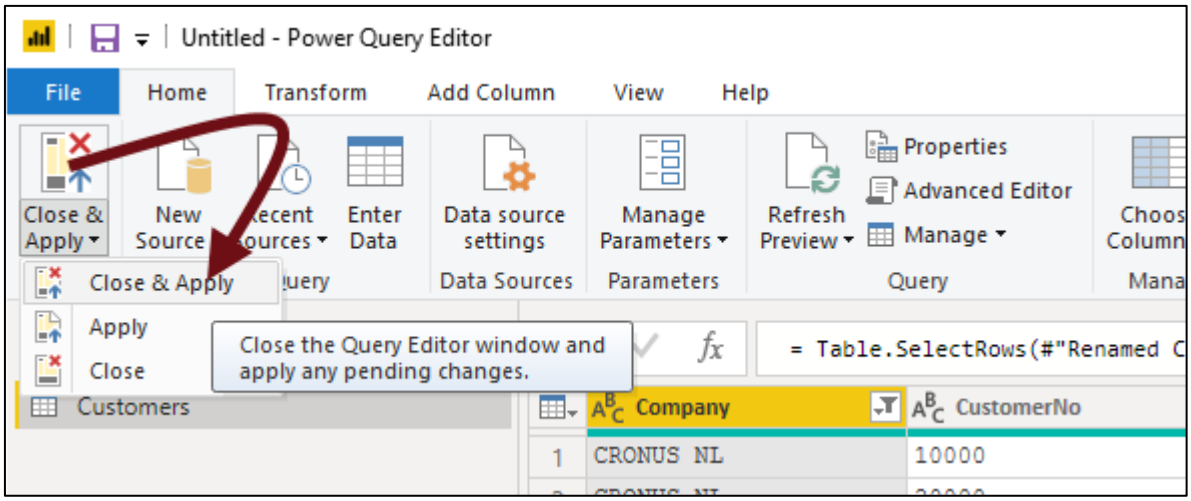
Select Table, right next to the qryCustomers:

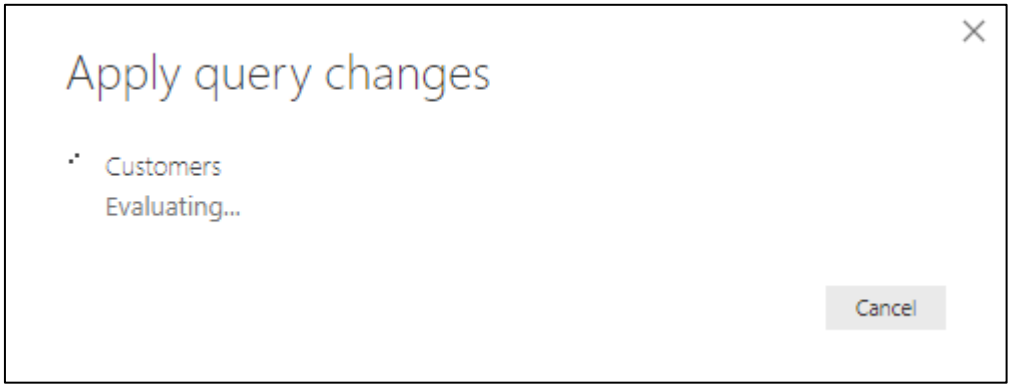


Then rename the query to **Customers**:

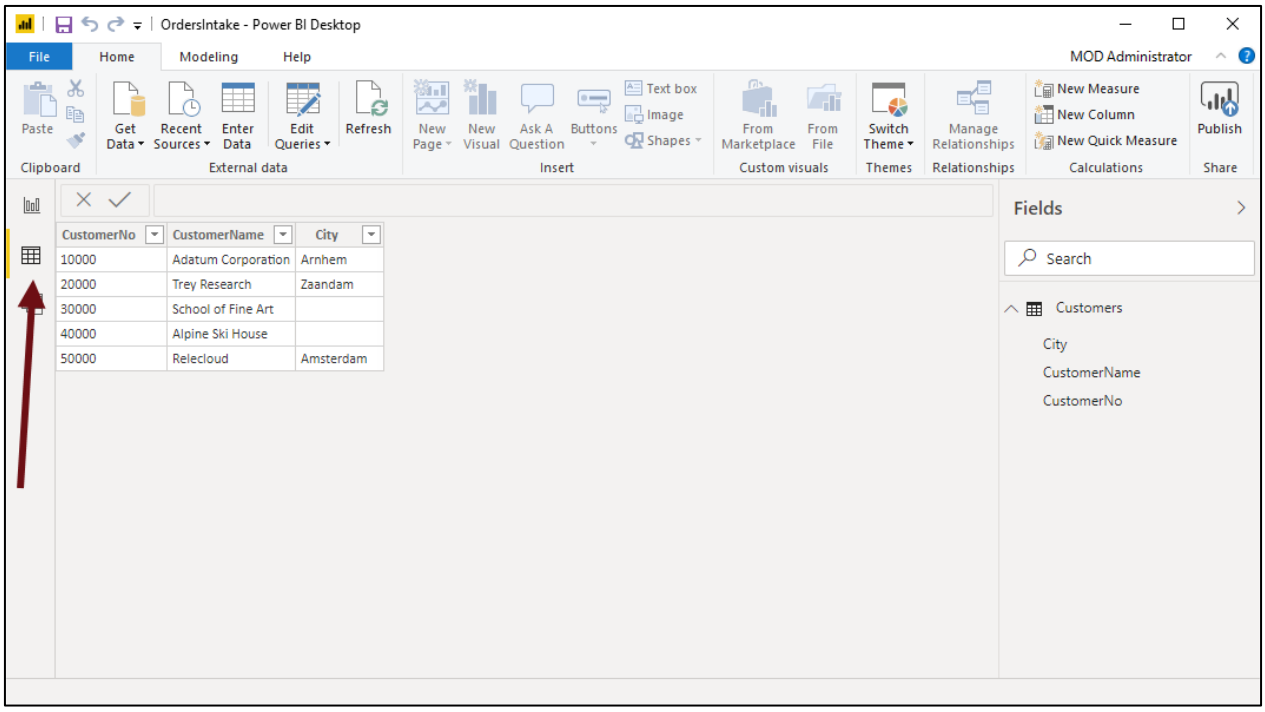


Then select **Close & Apply**:

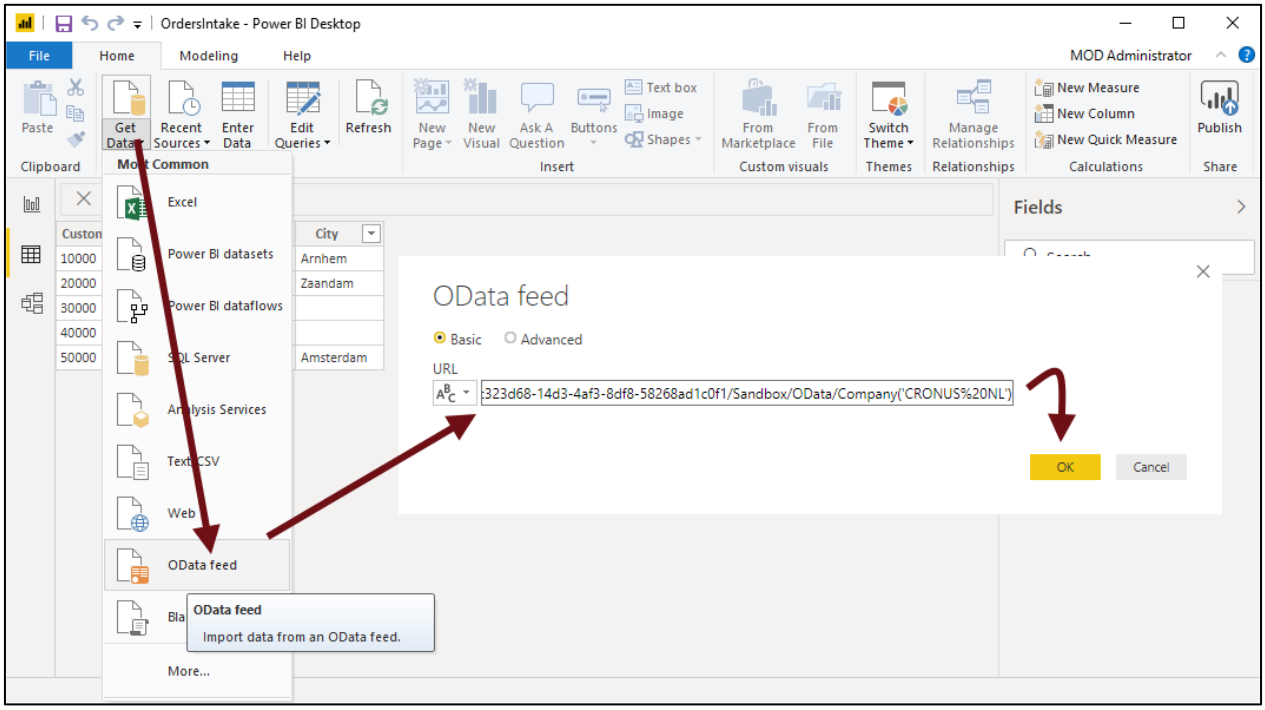


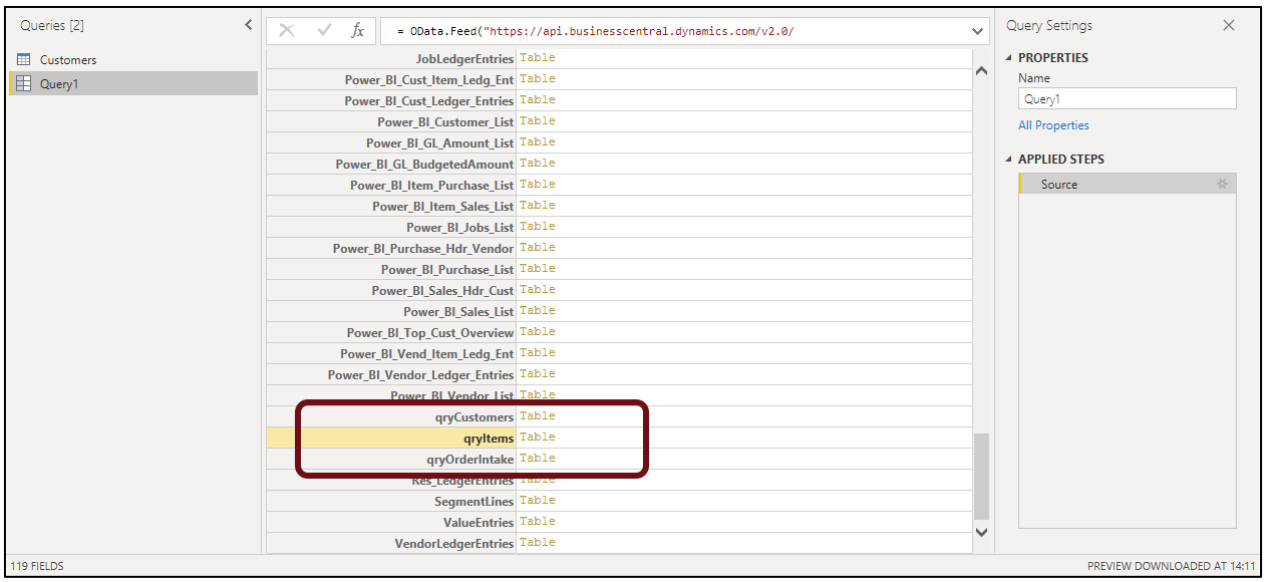


When you then select the **Table** button, you should see something like this:

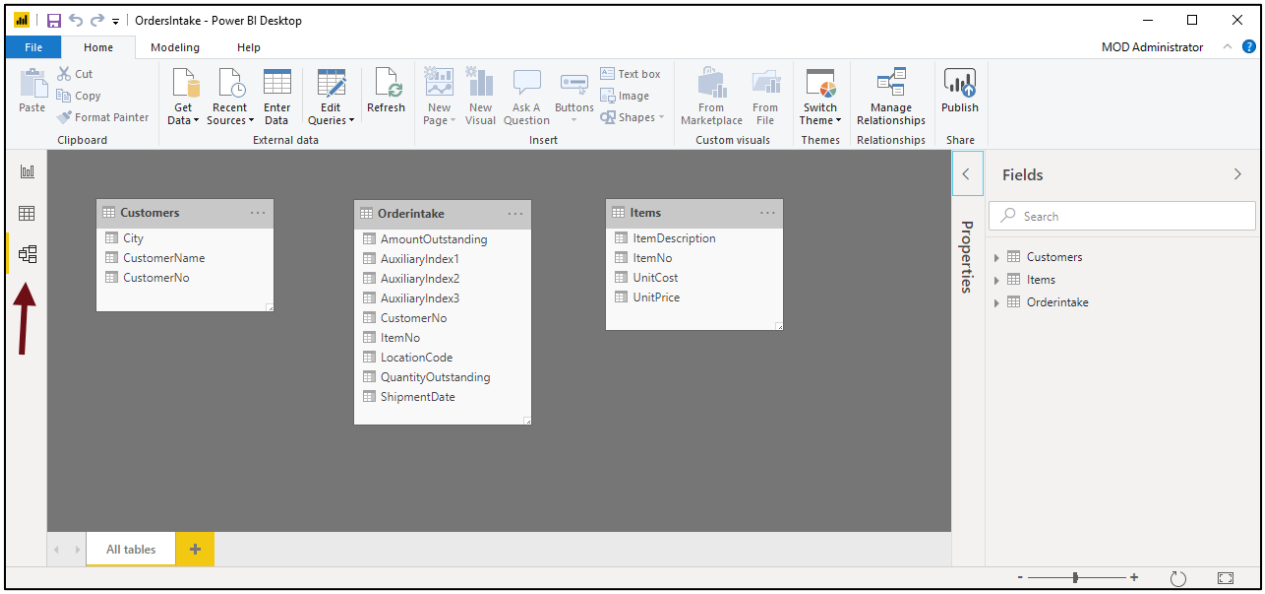


We will now also import the qryItems and qryOrderIntake in the same way:



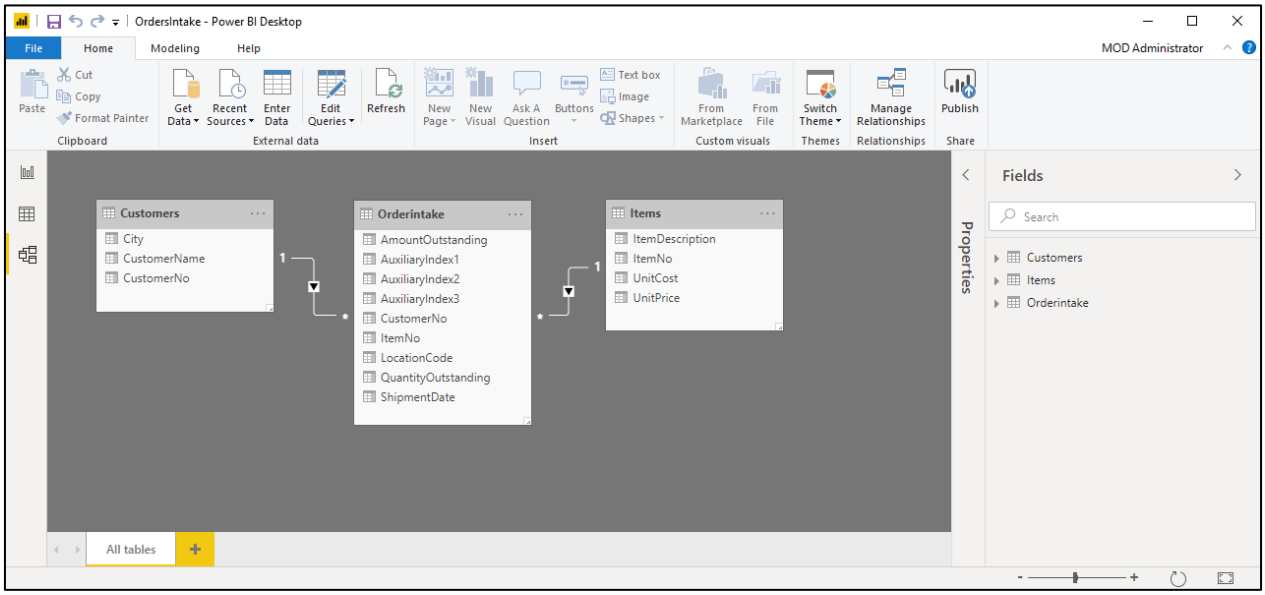


This should be the result, in the **relations** tab:


OrdersIntake - Power BI Desktop
File
Home
Modeling
Help
MOD Administrator
Paste
Cut
Copy
Format Painter
Clipboard
Get Data
Recent Sources
Enter Data
Edit Queries
Refresh
External data
New Page
New Visual
Ask A Question
Buttons
Text box
Image
Shapes
Insert
From Marketplace
From File
Custom visuals
Switch Theme
Themes
Manage Relationships
Relationships
Publish
Share
Customers
City
CustomerName
CustomerNo
Orderintake
AmountOutstanding
AuxiliaryIndex1
AuxiliaryIndex2
AuxiliaryIndex3
CustomerNo
ItemNo
LocationCode
QuantityOutstanding
ShipmentDate
Items
ItemDescription
ItemNo
UnitCost
UnitPrice
Properties
Fields
Search
Customers
Items
Orderintake
All tables

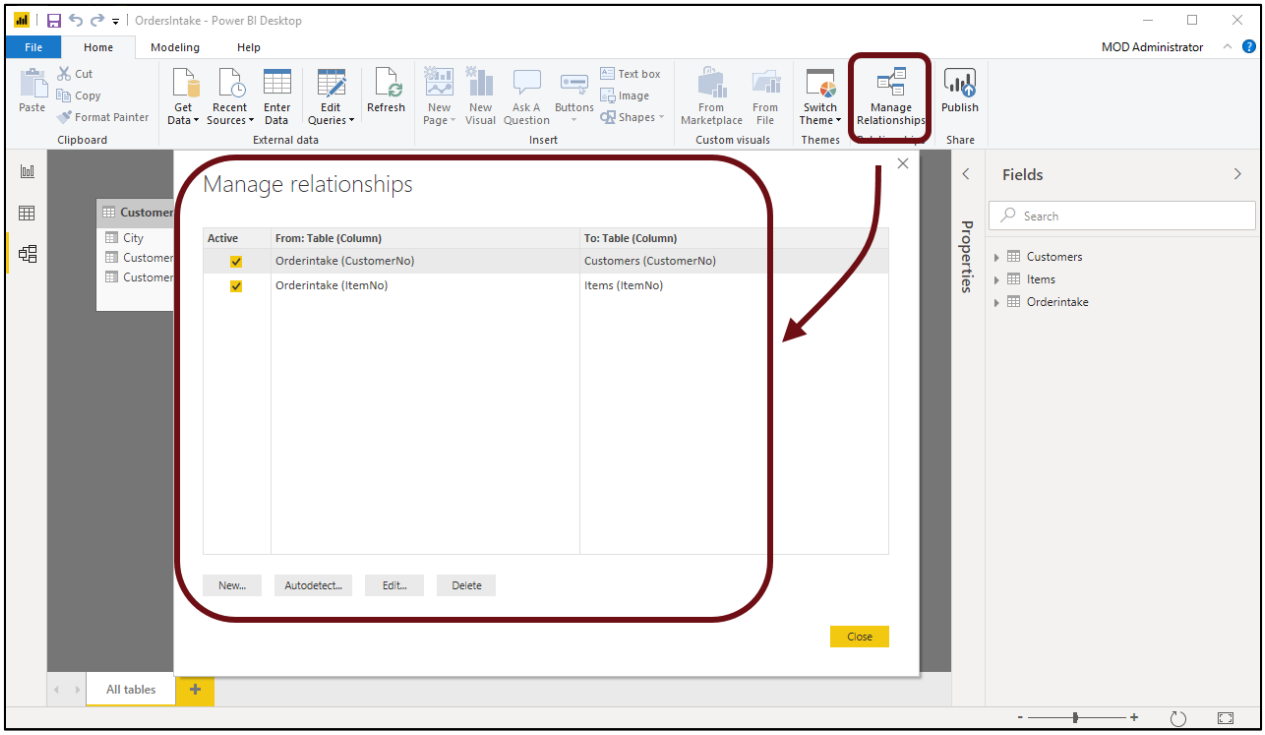
Create the Data Model and relations

You can use the **CustomerNo** and **ItemNo** fields to create relationships between the tables:



To create the relationships simply drag a field from one table to another, for example the **CustomerNo** from the Customers table on the **CustomerNo** of the **OrderIntake** table.

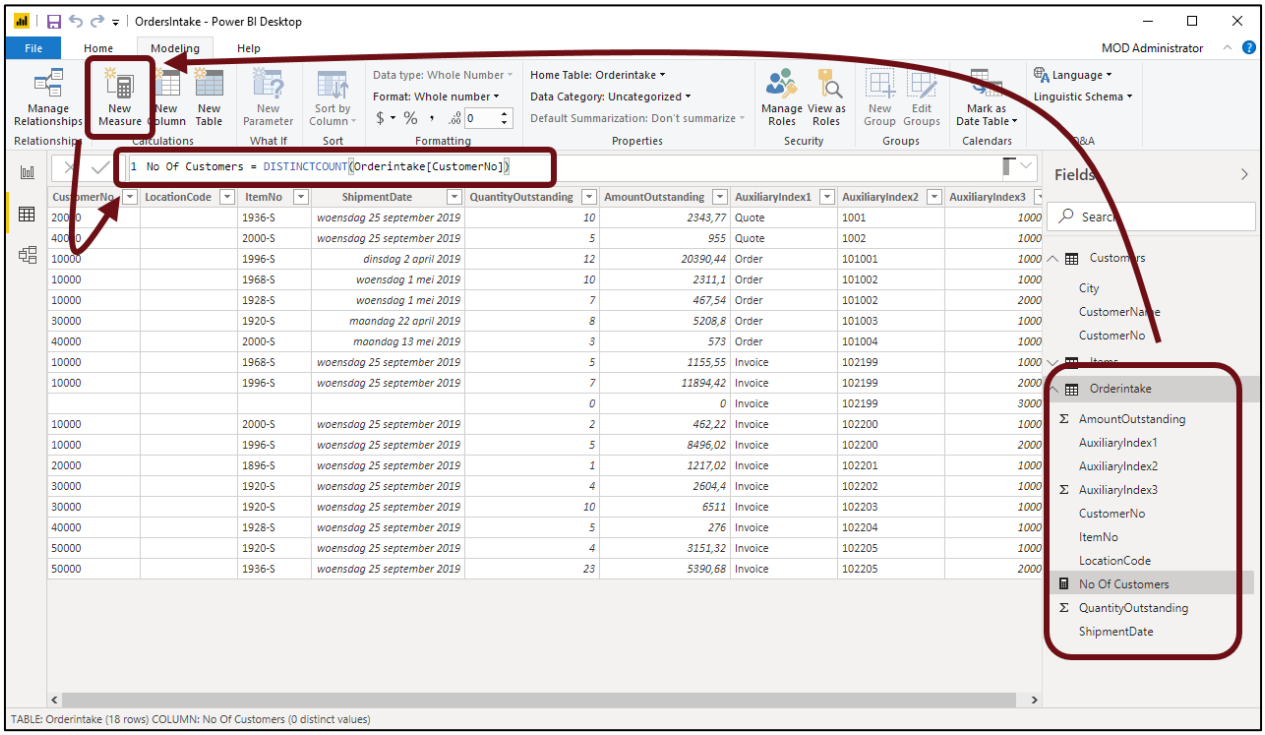
As an alternative, you could also use the **Manage Relationships** button:


OrdersIntake - Power BI Desktop
File
Home
Modeling
Help
MOD Administrator
Paste
Cut
Copy
Format Painter
Clipboard
Get Data
Recent Sources
Enter Data
Edit Queries
Refresh
External data
New Page
New Visual
Ask A Question
Buttons
Text box
Image
Shapes
Insert
From Marketplace
From File
Custom visuals
Switch Theme
Themes
Manage Relationships
Publish
Share
Manage relationships
Active
From: Table (Column)
To: Table (Column)
Orderintake (CustomerNo)
Customers (CustomerNo)
Orderintake (ItemNo)
Items (ItemNo)
New...
Autodetect...
Edit...
Delete
Close
Fields
Search
Customers
Items
Orderintake
Properties
All tables

Enrich the Data Model with DAX

To perform calculations or create KPI's we will now add some measures to the data model.

In the **OrderIntake** table, select the **Modeling** tab and click **New Measure**:

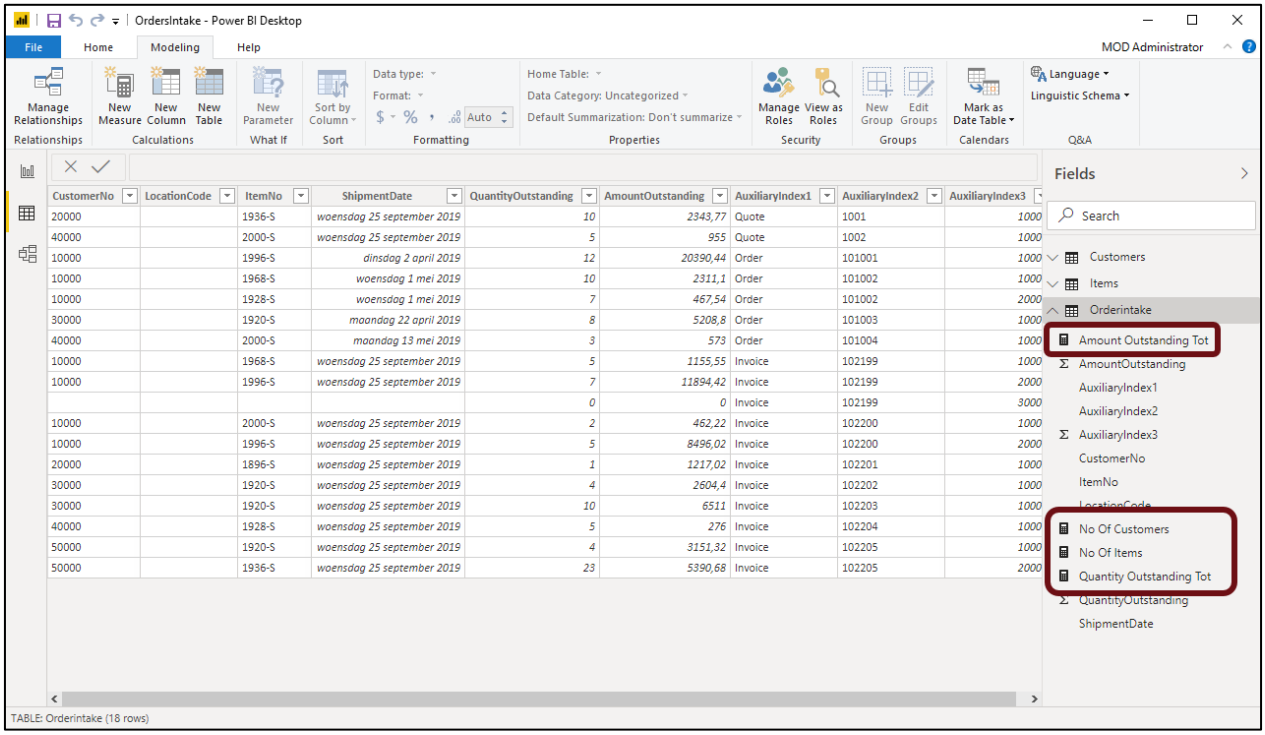


Use the following formula:

- **No Of Customers = DISTINCTCOUNT(Orderintake[CustomerNo])**

Similarly, also add the following measures to the **OrderIntake** table:

- **No Of Items = DISTINCTCOUNT(OrderIntake[ItemNo])**
- **Quantity Outstanding Tot = sum(OrderIntake[QuantityOutstanding])**
- **Amount Outstanding Tot = sum(OrderIntake[AmountOutstanding])**

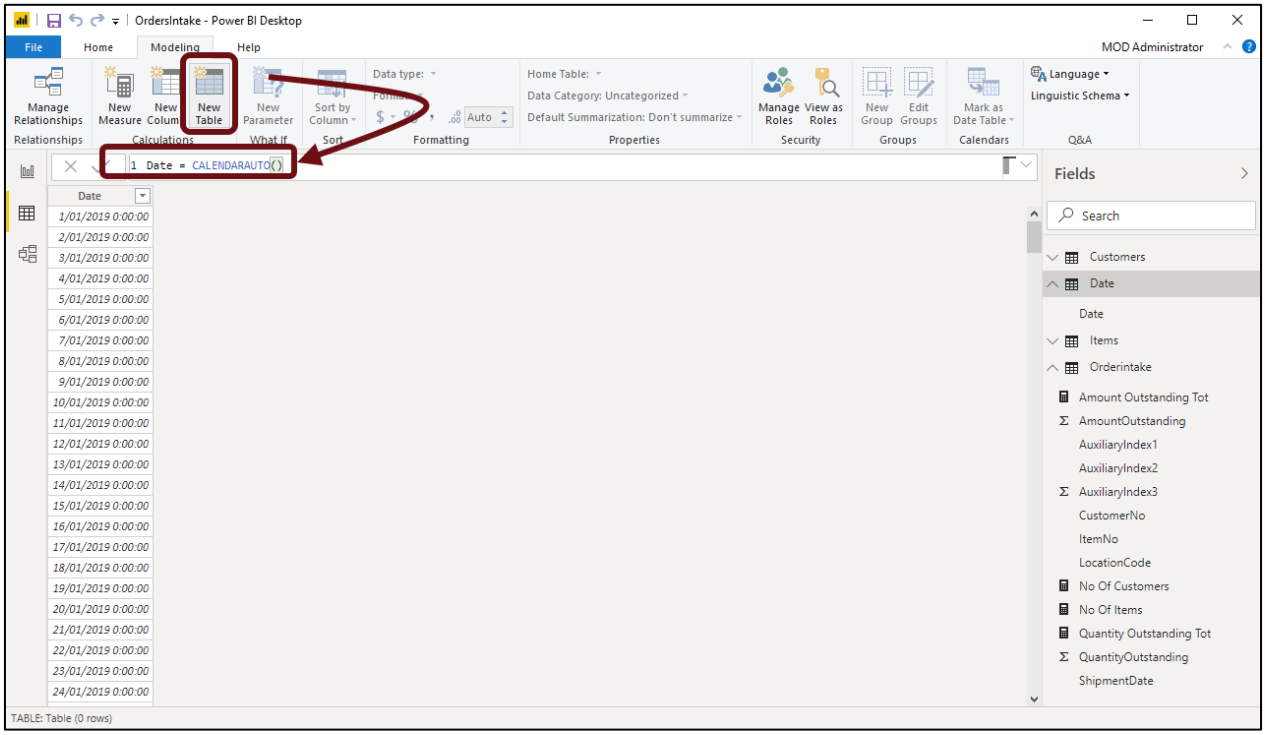


Now we will also add a **Date** table to the data model.

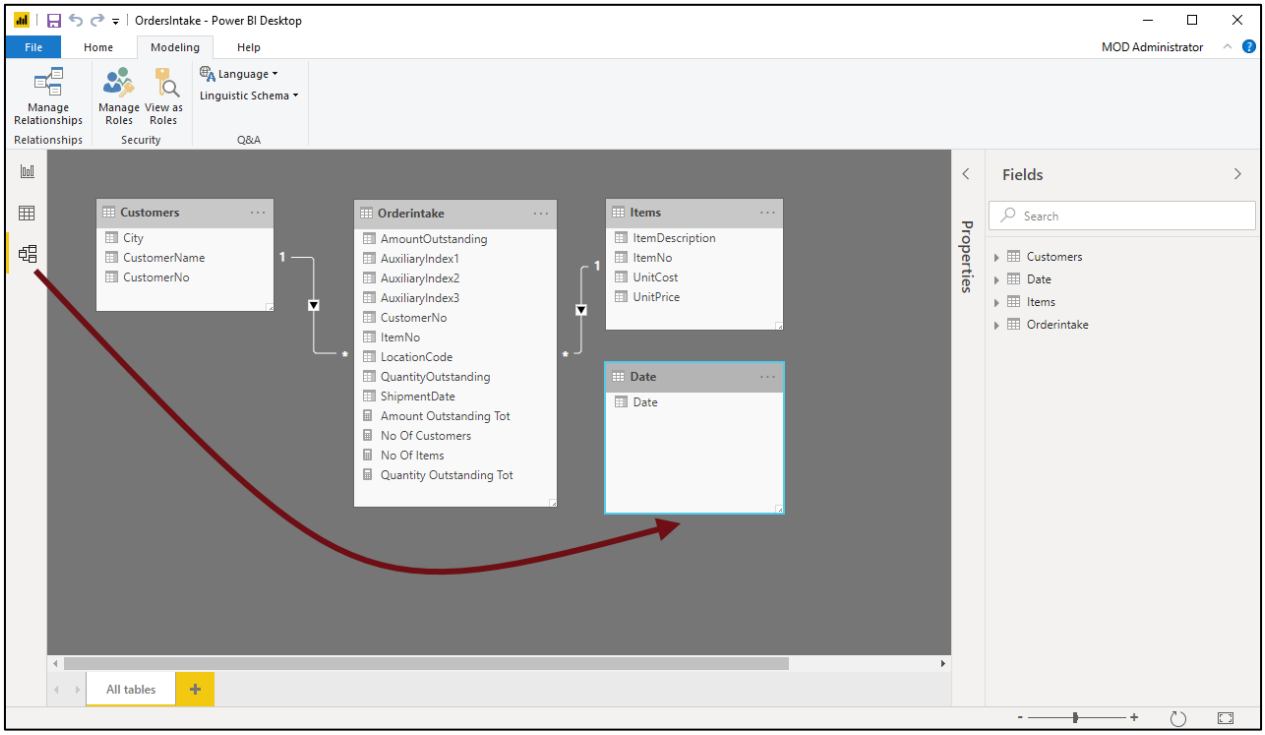
The reason is that we need a table with a **continuous** date range.

To create the Date table, in the **Modeling** tab select **New Table**, and use the following formula:

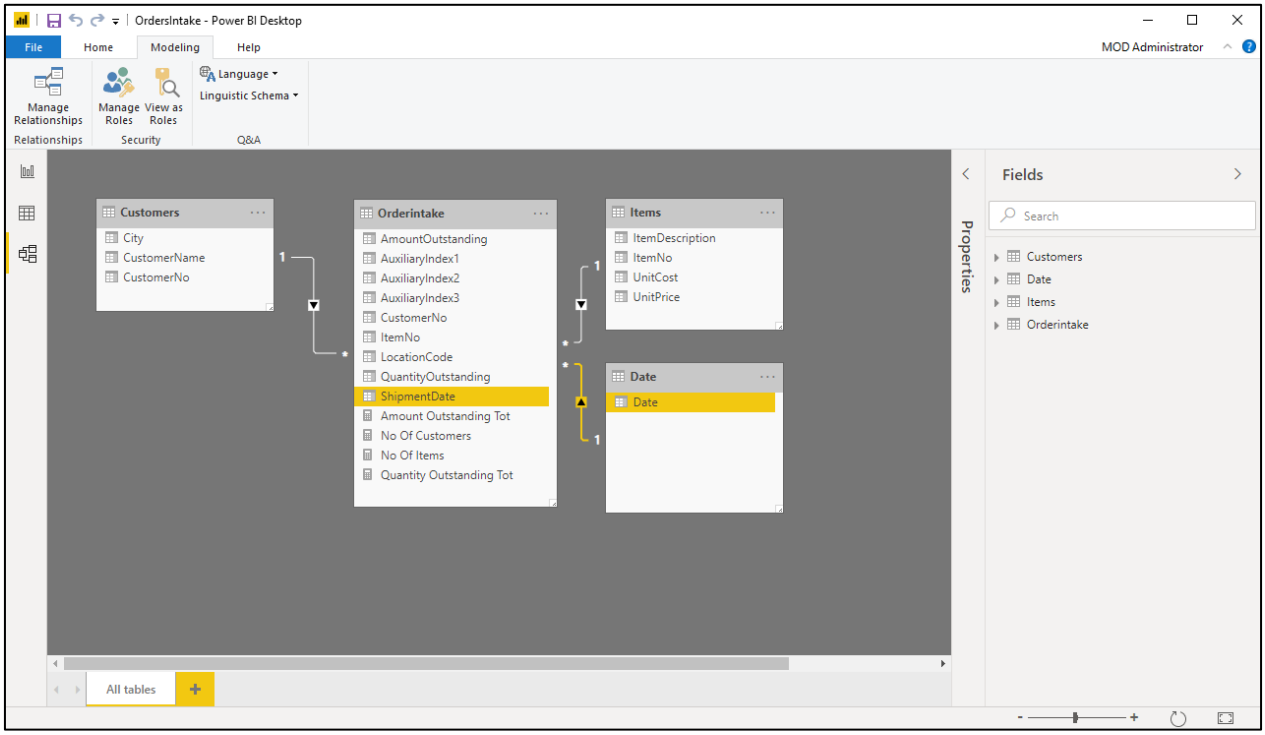
- **Date = CALENDARAUTO()**



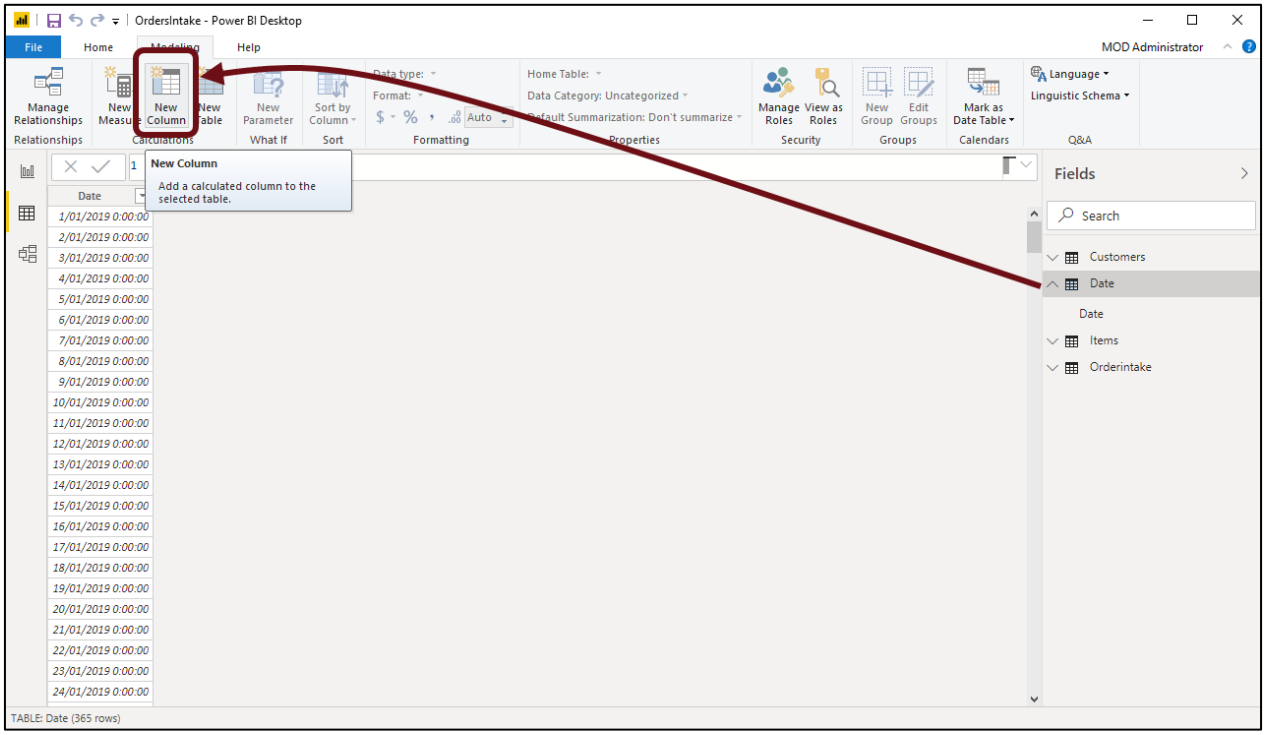
You can also see the table in the **Relationships** window:



Connect the OrderIntake table to the Date table as follows:

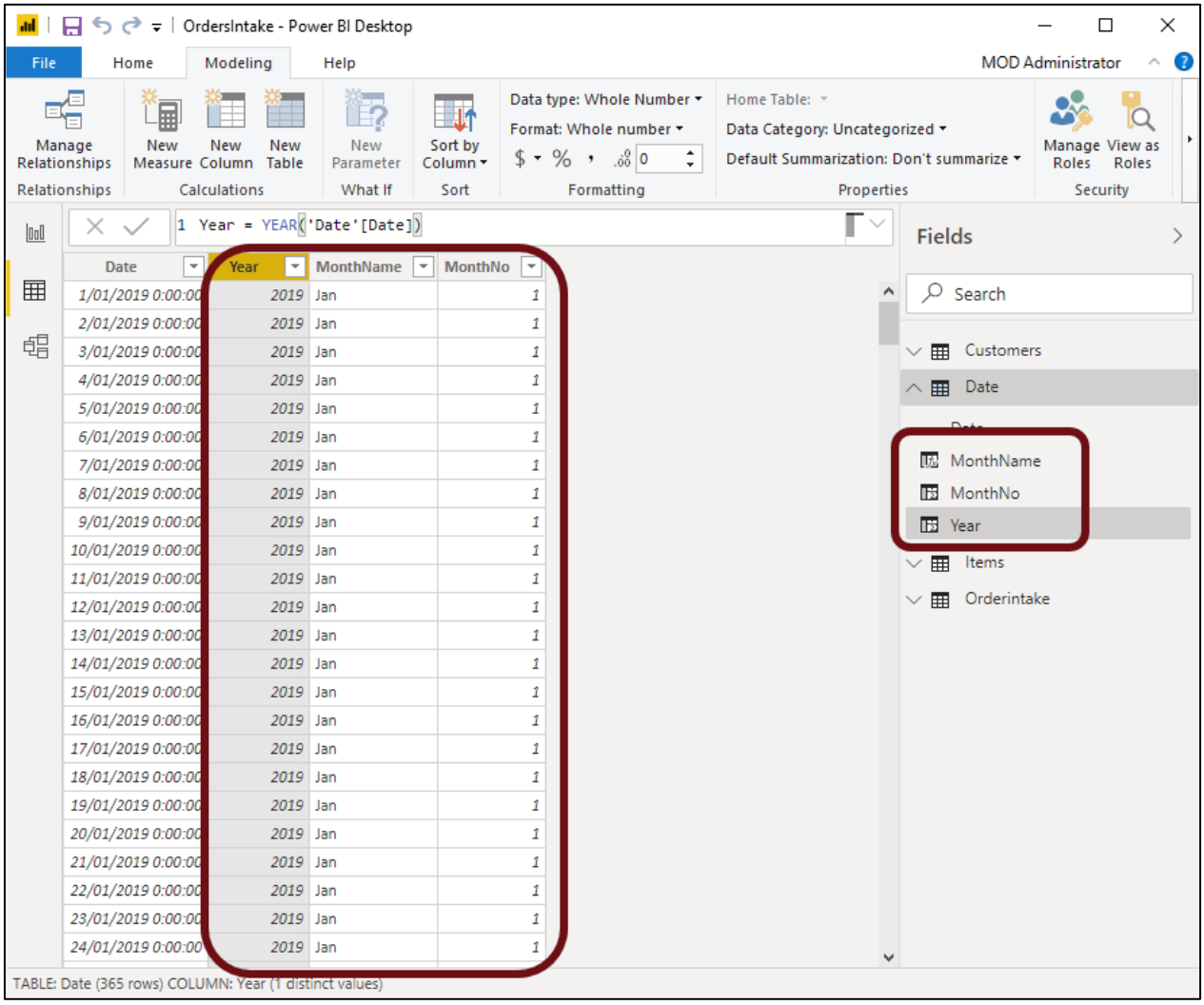


In the **Date** table we will now create new columns:

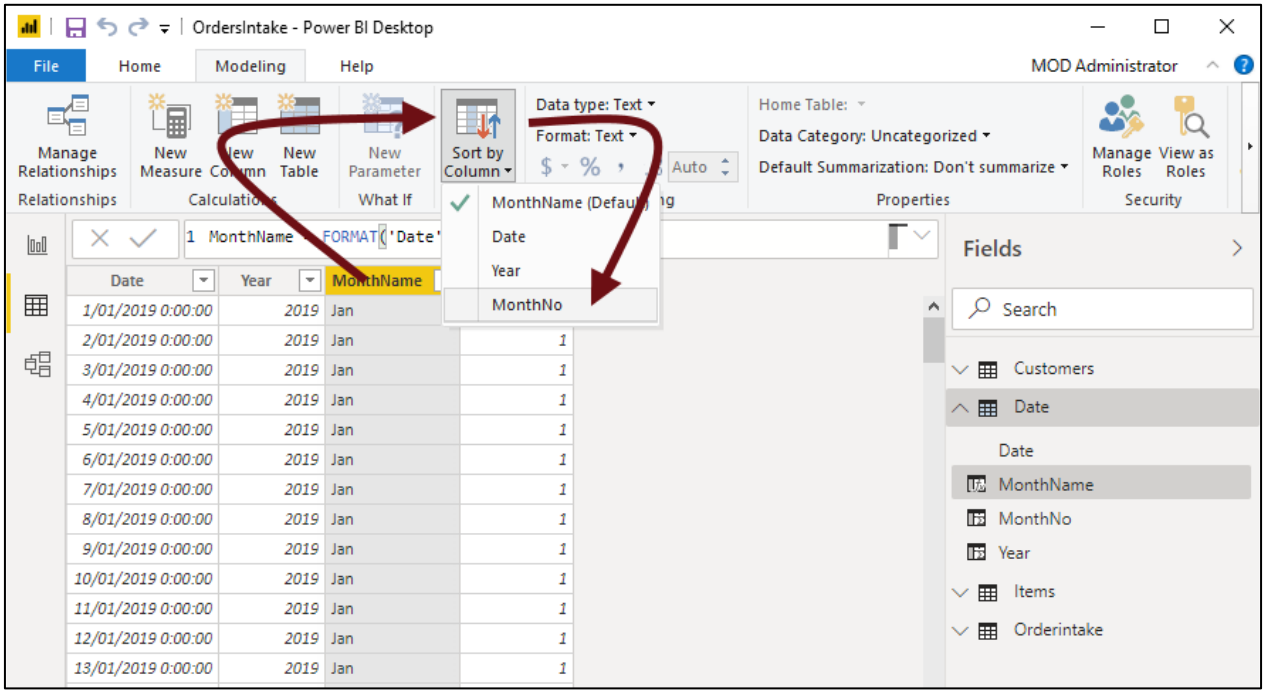


You can use the following DAX formulas to create the columns:

- Year = YEAR('Date'[Date])
- MonthName = FORMAT('Date'[Date],"MMM")
- MonthNo = MONTH('Date'[Date])



And finally, we will set the **MonthNo** column as the **SortBy** column for the **MonthName**:

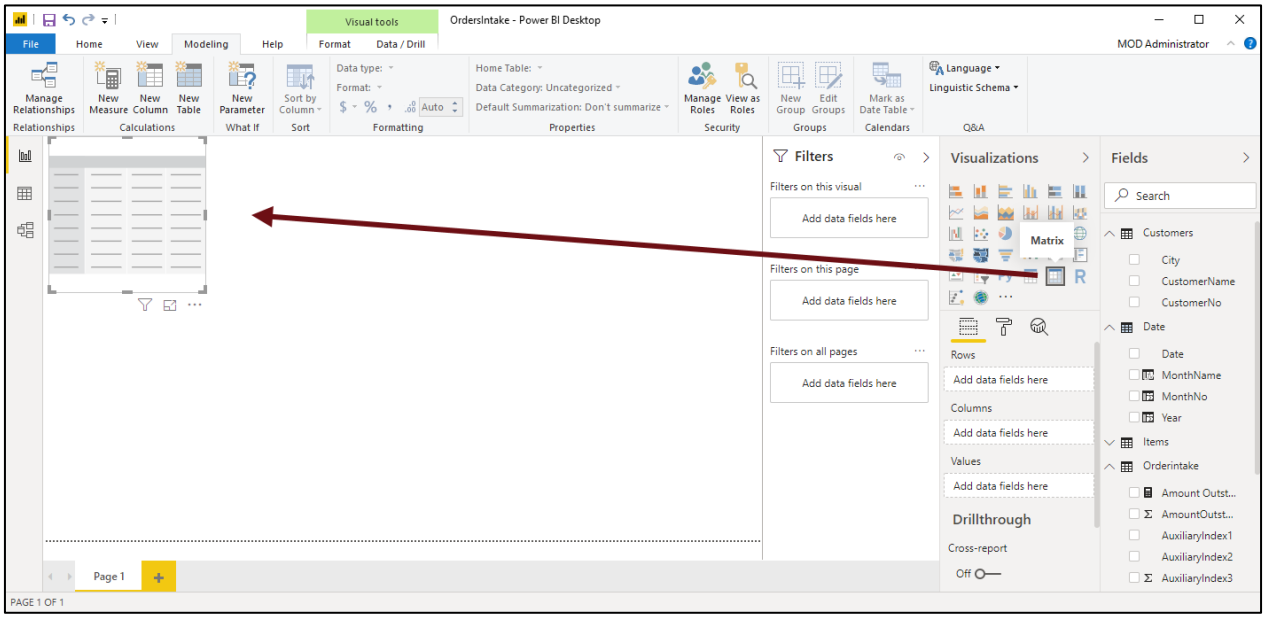


If you don't set the **SortBy** Column, then the **MonthNames** will be sorted **alphabetically**.

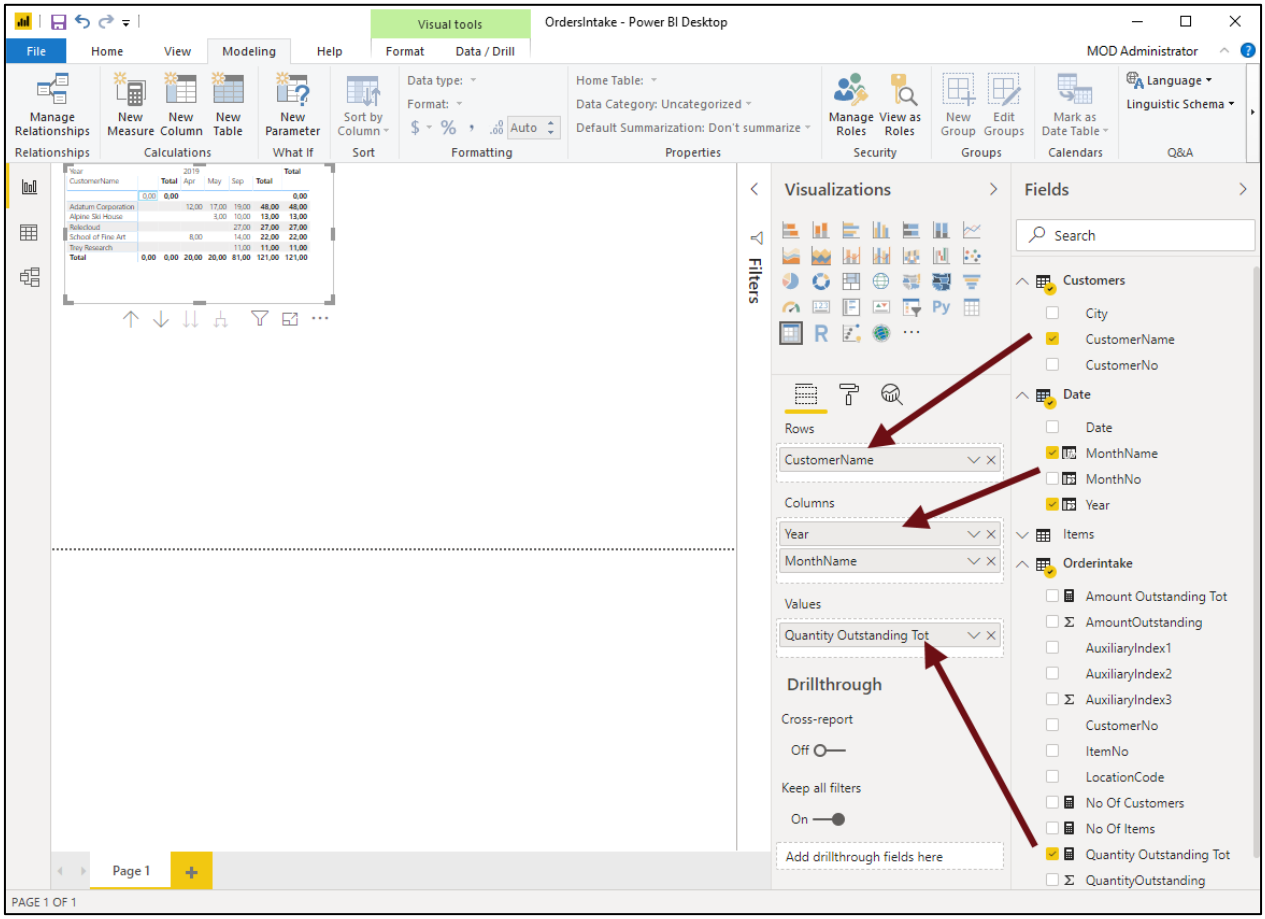
We have now enriched the data model with a Date table and some Measures. Now it's time to create some reports.

Create the Report(s)

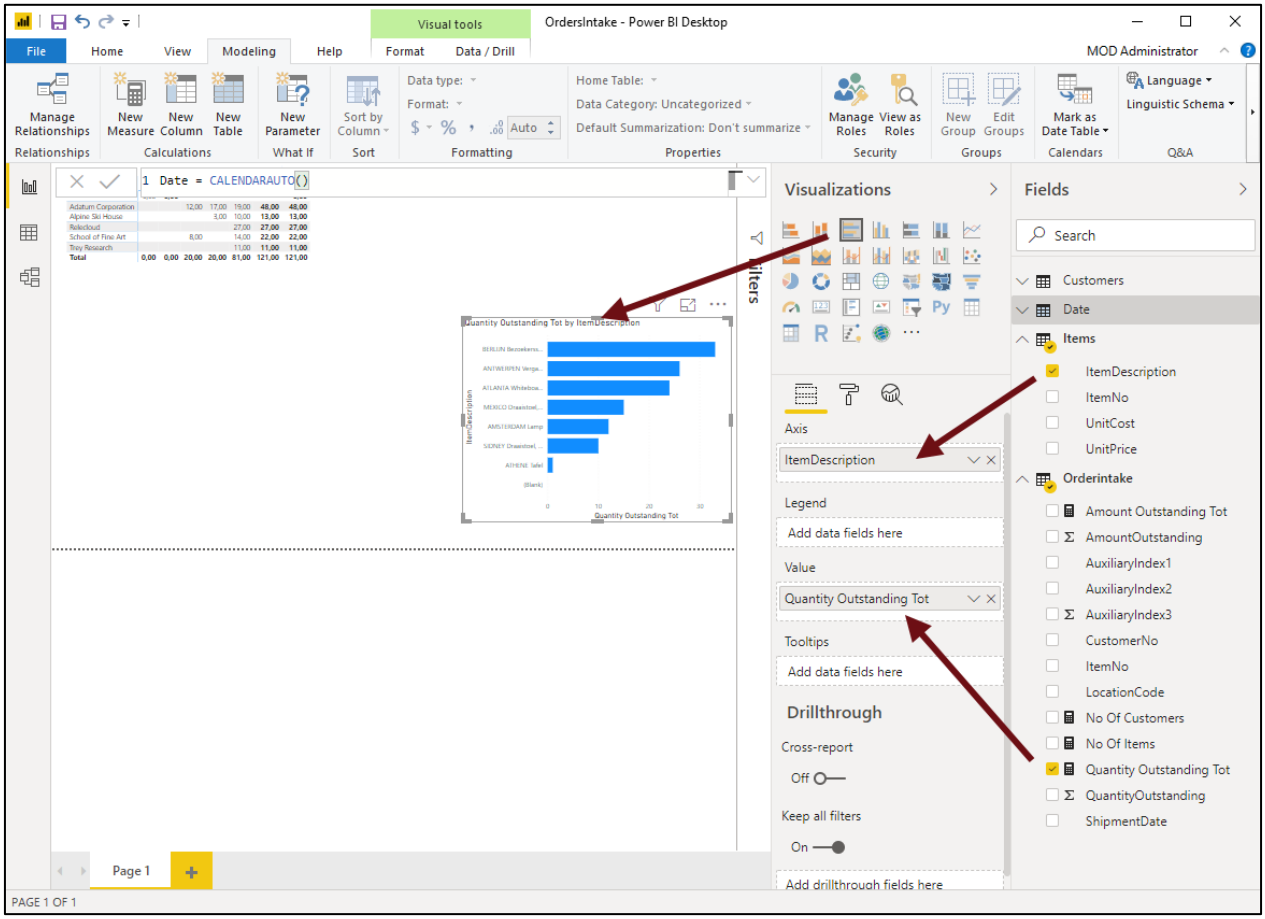
Add a **Matrix** to the report:



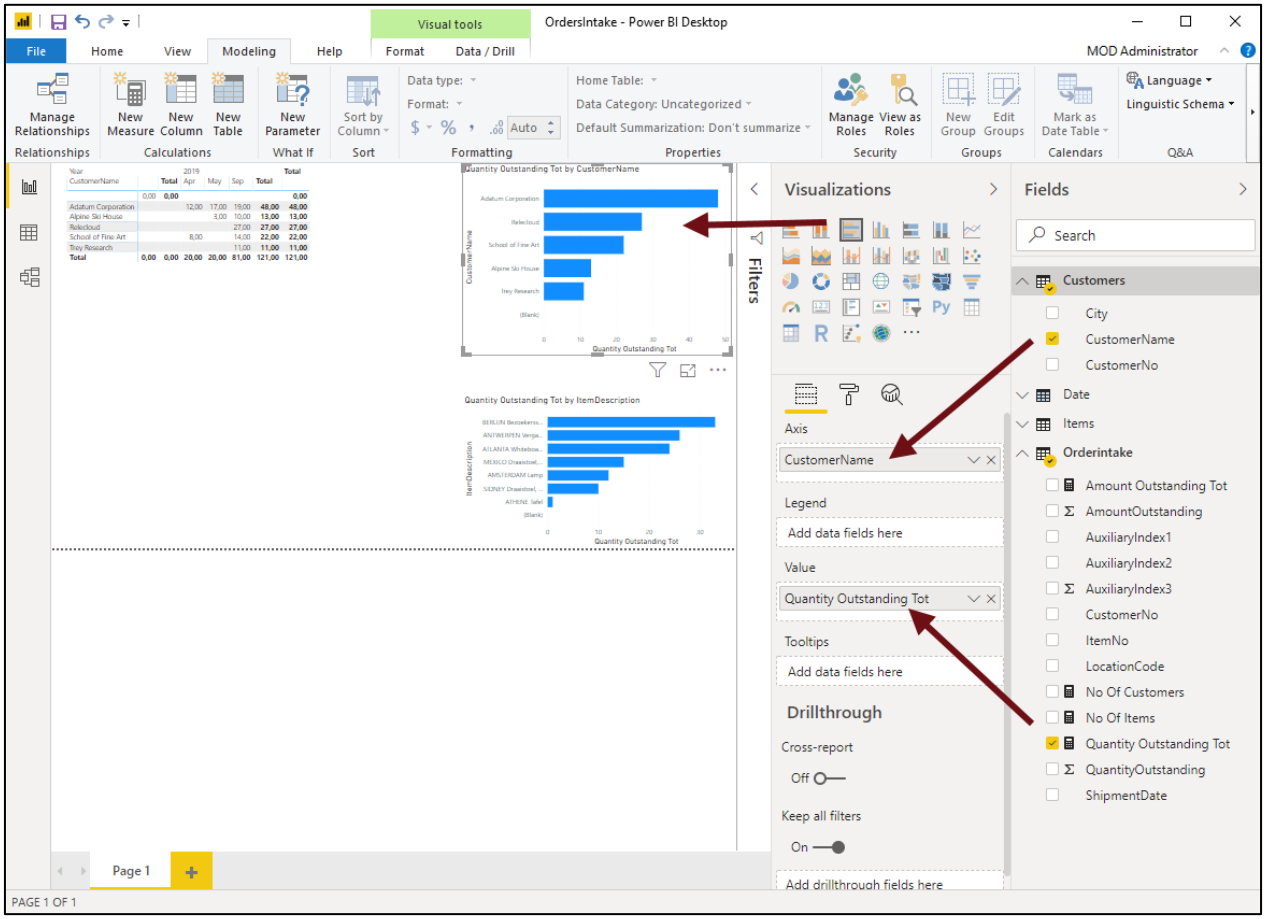
Drag the following fields into Rows, Columns and Values:



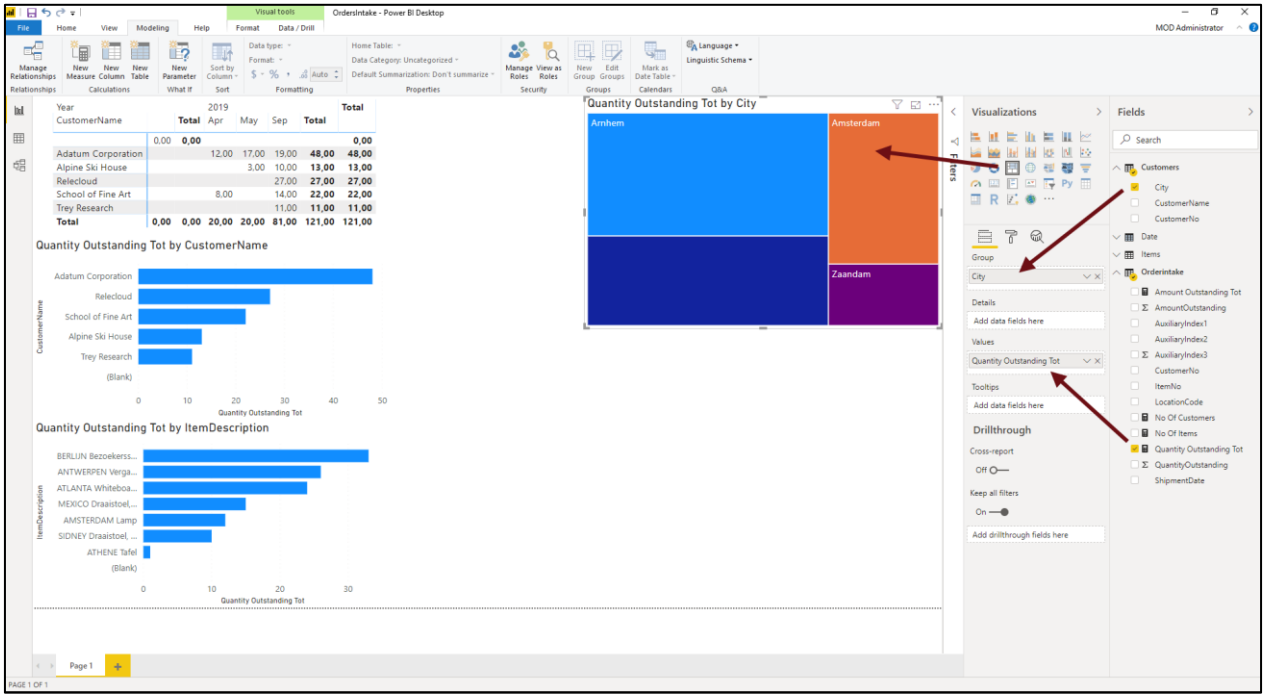
Add a Chart to the report:



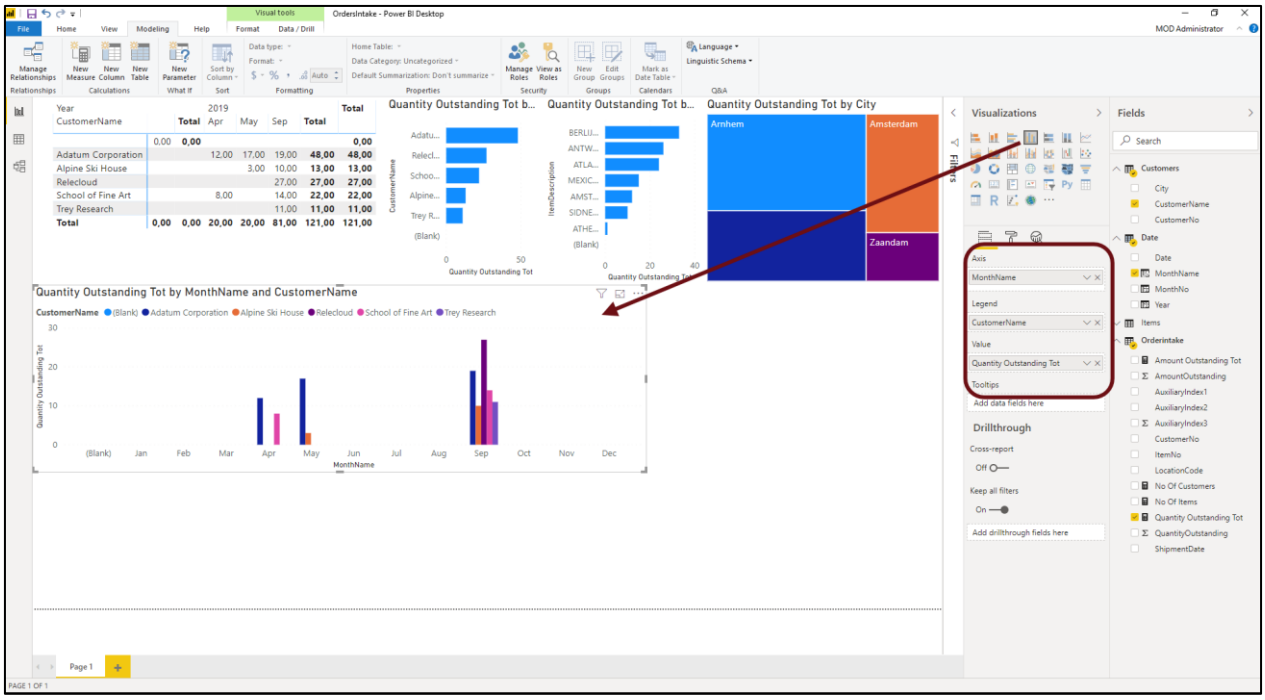
Add another Chart to the report:



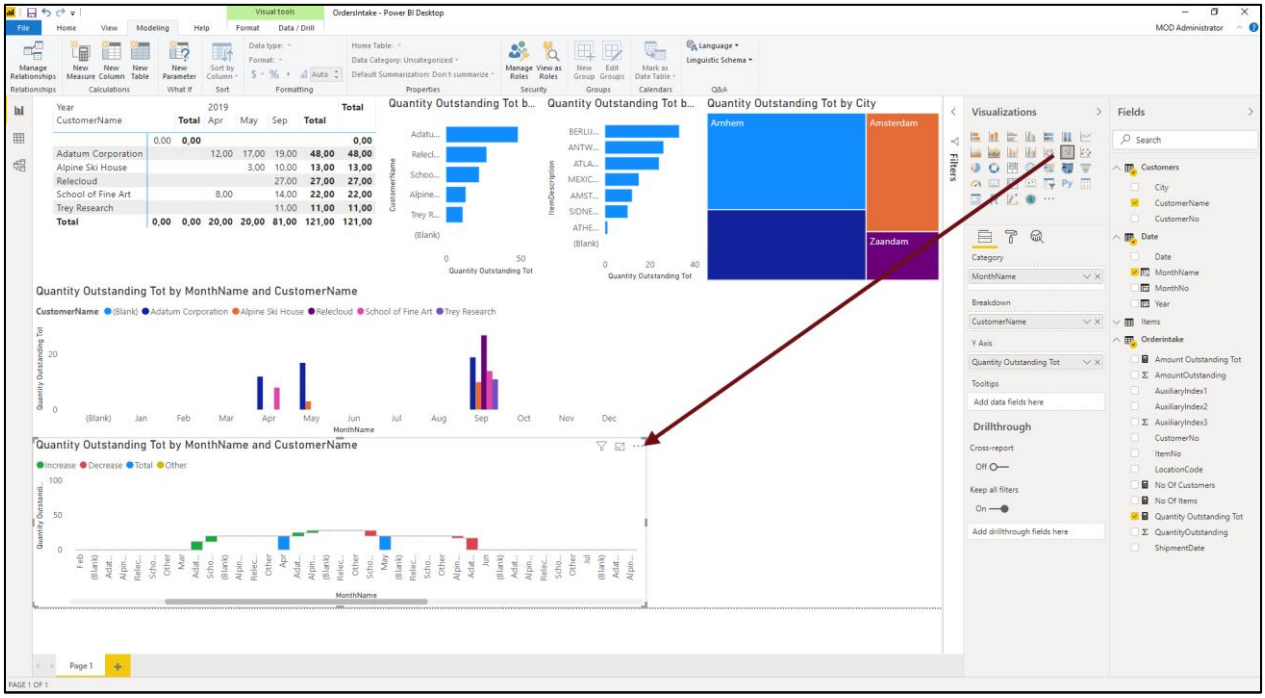
Add a Treemap to the report:



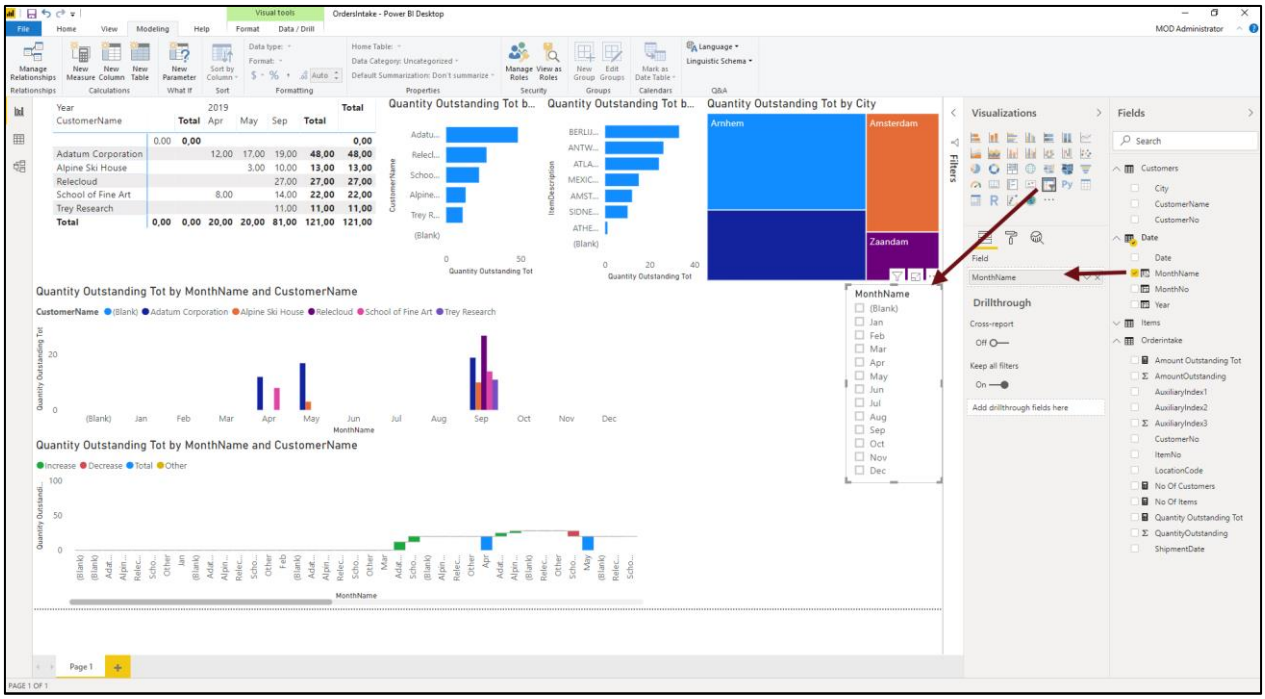
Add another Chart to the report:



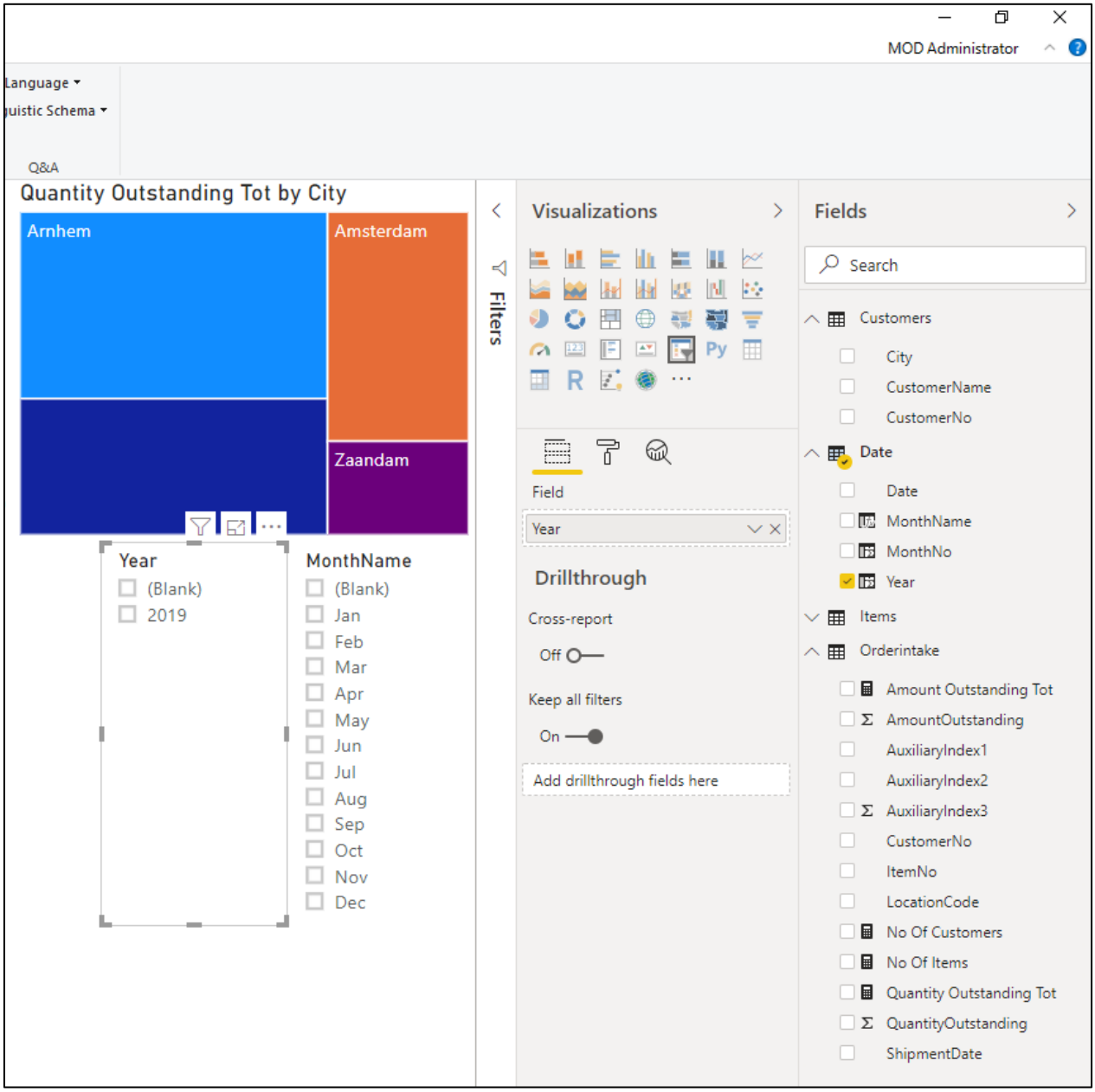
Copy/Paste the previous chart and change it into a Waterfall chart:



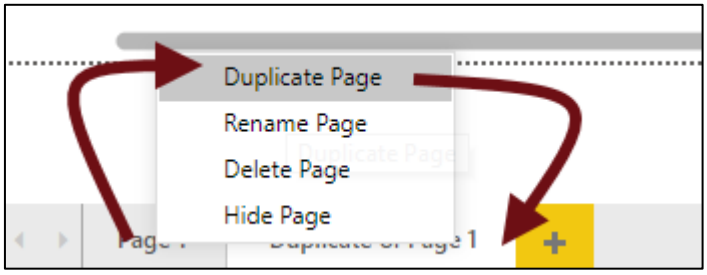
Add a Slicer:



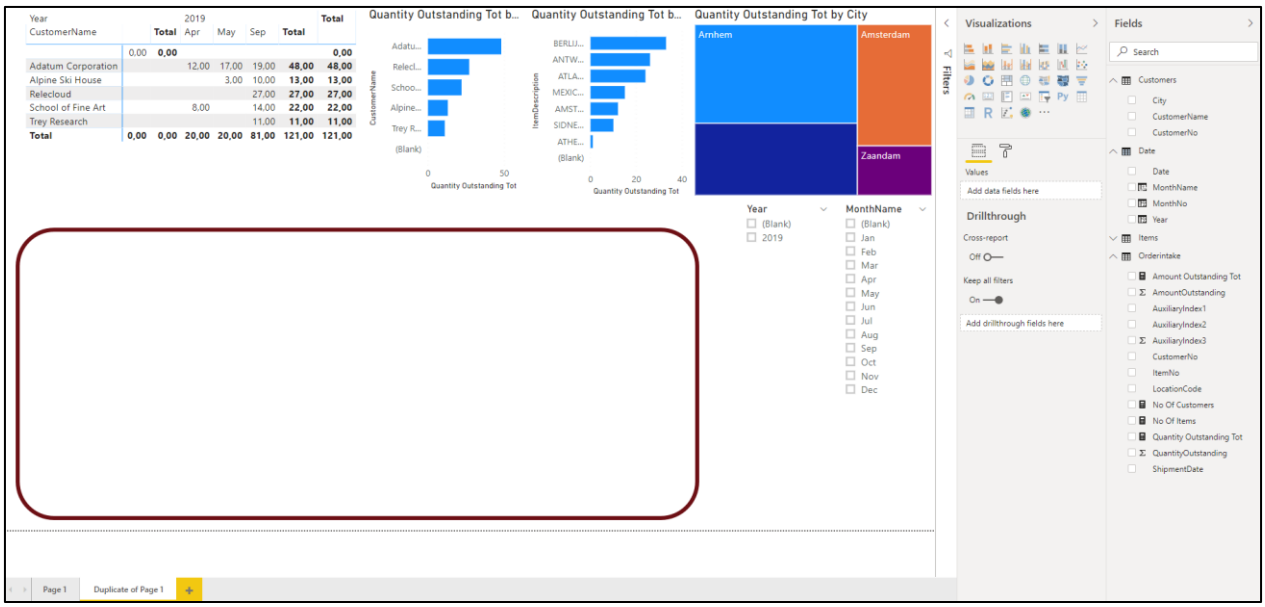
Add another Slicer:



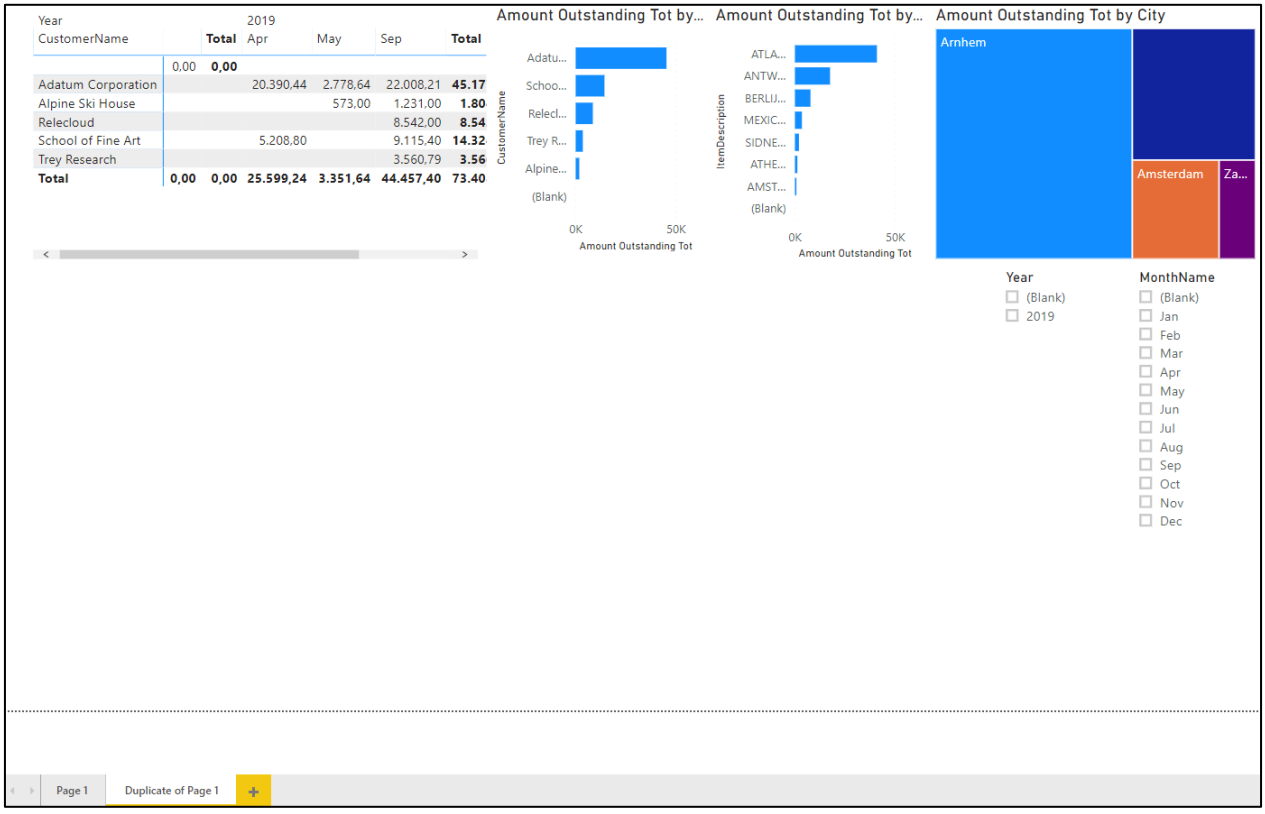
Duplicate the Report page:



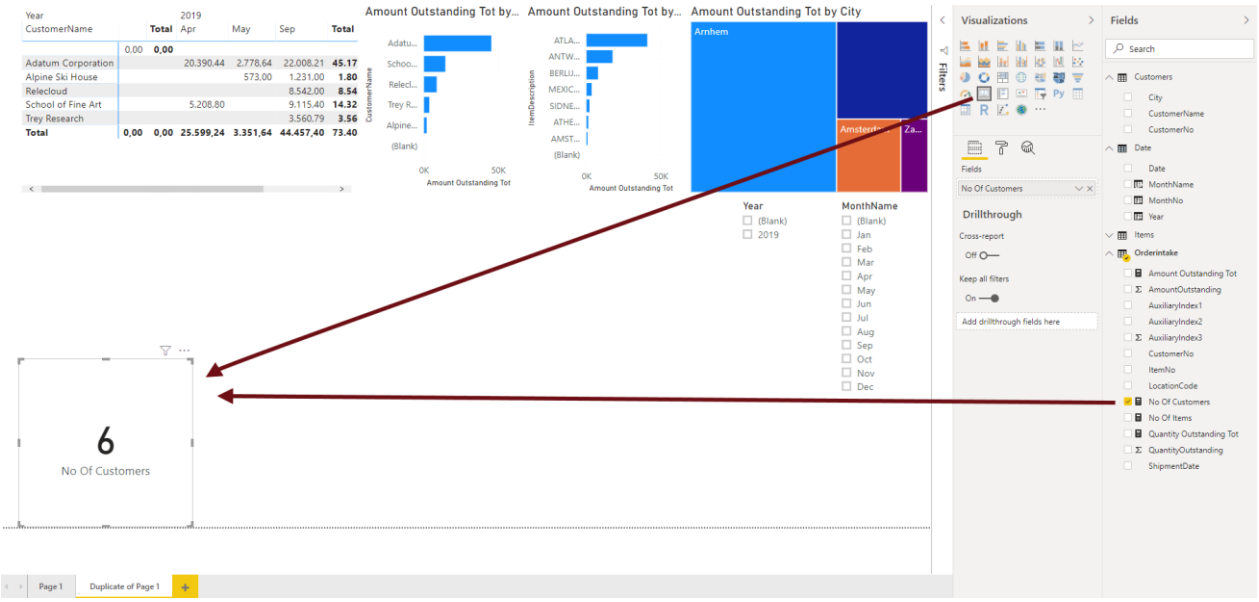
On the new page, remove the bottom charts:



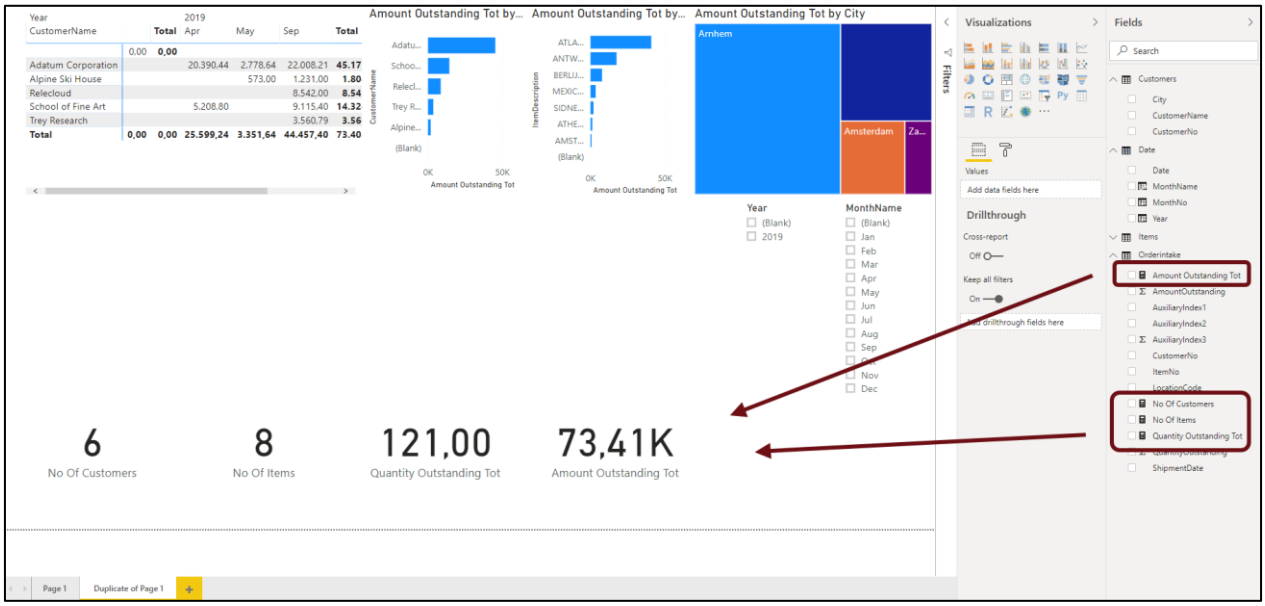
Then go into the visuals on the new page and replace the **Quantity** fields by the corresponding **Amount** fields.


Year
2019
CustomerName
Total
Apr
May
Sep
Total
0,00
0,00
Adatum Corporation
20.390,44
2.778,64
22.008,21
45.17
Alpine Ski House
573,00
1.231,00
1.80
Relecloud
8.542,00
8.54
School of Fine Art
5.208,80
9.115,40
14.32
Trey Research
3.560,79
3.56
Total
0,00
0,00
25.599,24
3.351,64
44.457,40
73.40
Amount Outstanding Tot by...
Adatu...
Schoo...
Relecl...
Trey R...
Alpine...
(Blank)
CustomerName
0K
50K
Amount Outstanding Tot
Amount Outstanding Tot by...
ATLA...
ANTW...
BERLIJ...
MEXIC...
SIDNE...
ATHE...
AMST...
(Blank)
ItemDescription
0K
50K
Amount Outstanding Tot
Amount Outstanding Tot by City
Arnhem
Amsterdam
Za...
Year
(Blank)
2019
MonthName
(Blank)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Page 1
Duplicate of Page 1

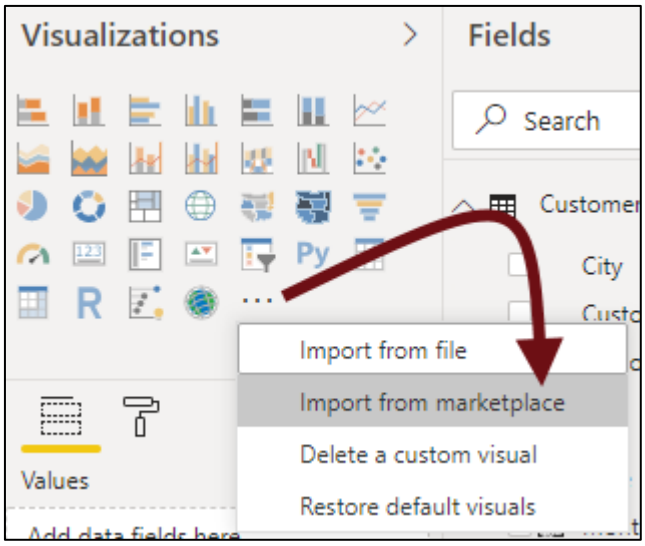
Add a Card:


Year
2019
CustomerName
Total
Apr
May
Sep
Total
0,00
0,00
Adatum Corporation
20.390,44
2.778,64
22.008,21
45.17
Alpine Ski House
573,00
1.231,00
1.80
Relecloud
8.542,00
8.54
School of Fine Art
5.208,80
9.115,40
14.32
Trey Research
3.560,79
3.56
Total
0,00
0,00
25.599,24
3.351,64
44.457,40
73.40
Amount Outstanding Tot by...
Amount Outstanding Tot by...
Amount Outstanding Tot by City
Arnhem
Year
(Blank)
2019
MonthName
(Blank)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
6
No Of Customers
Visualizations
Fields
Filters
Drillthrough
Cross-report
Keep all filters
Add drillthrough fields here
Customers
City
CustomerName
CustomerNo
Date
MonthName
MonthNo
Year
Items
Orderintake
Amount Outstanding Tot
AmountOutstanding
AuxiliaryIndex1
AuxiliaryIndex2
AuxiliaryIndex3
CustomerNo
ItemNo
LocationCode
No Of Customers
No Of Items
Quantity Outstanding Tot
QuantityOutstanding
ShipmentDate
Page 1
Duplicate of Page 1

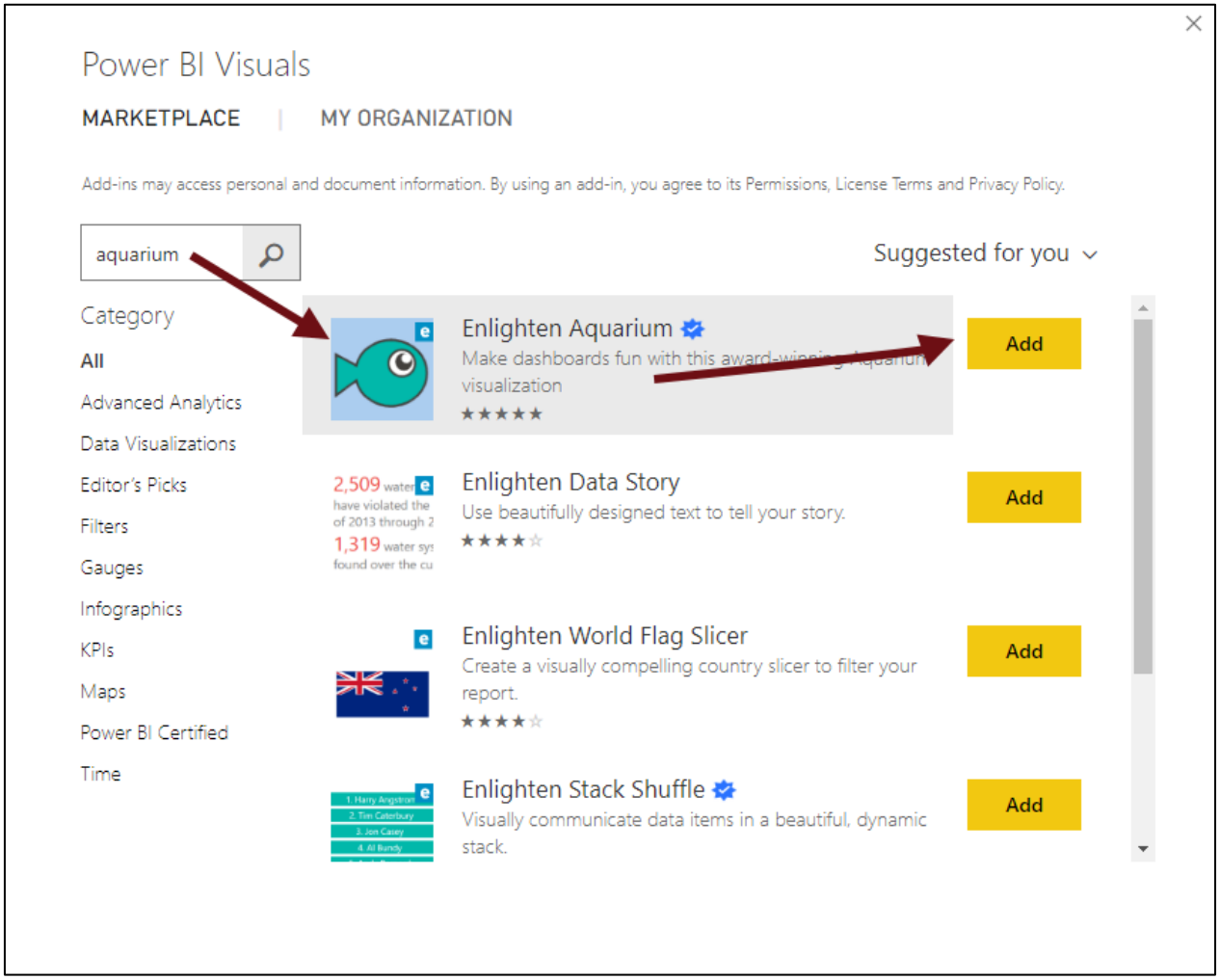
Copy/Paste the Card and add the following cards:



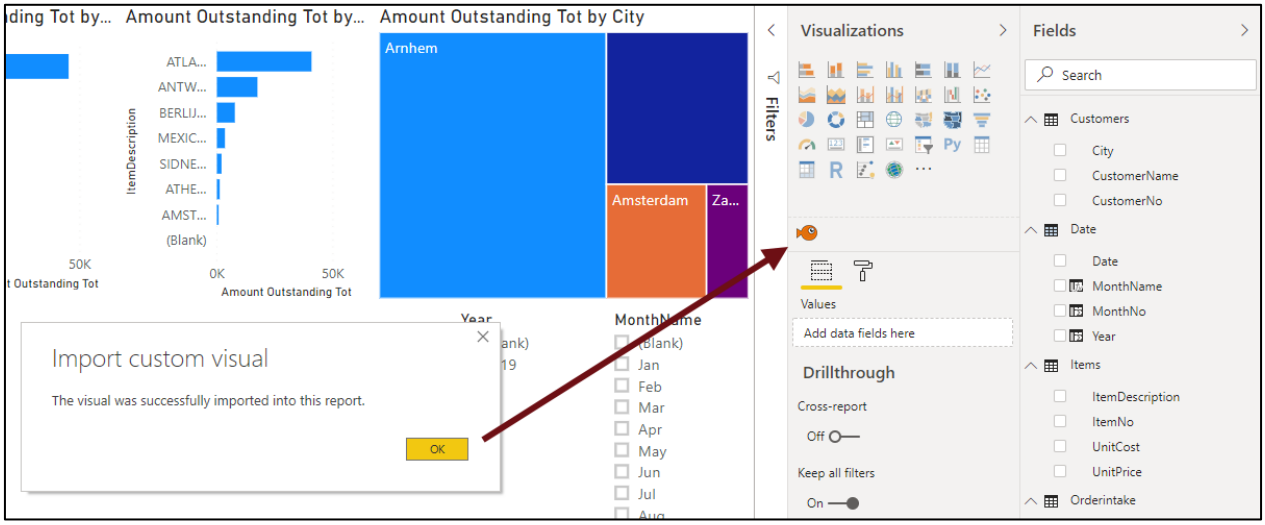
Add a custom visual:

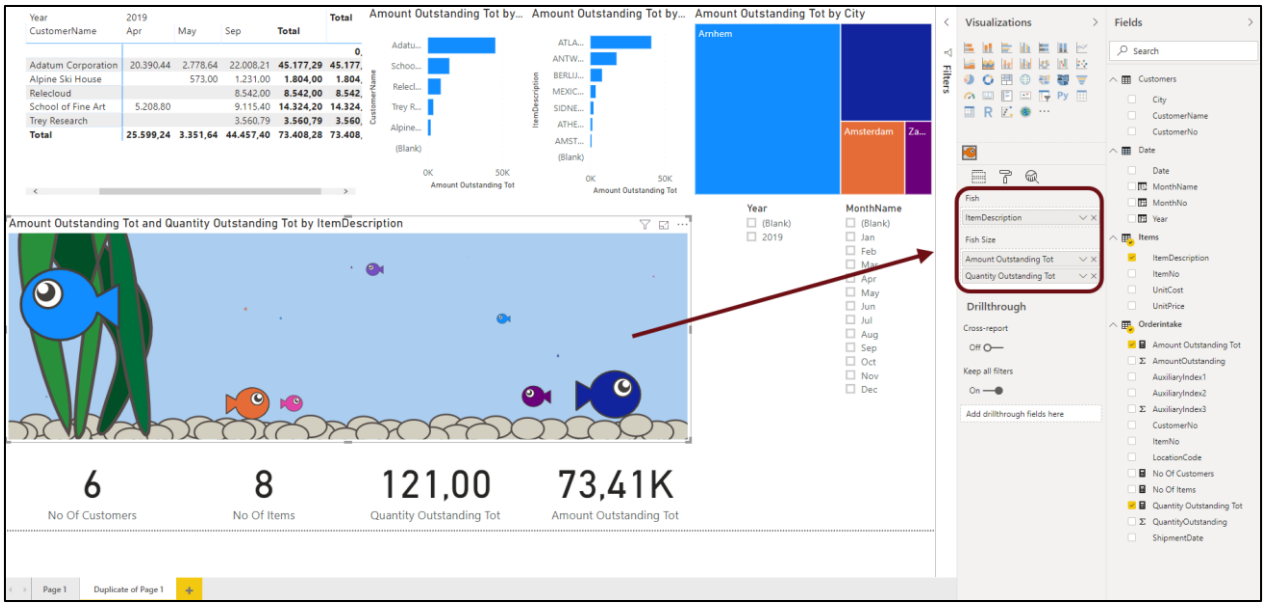


Look for the **Aquarium** visual and Add it:

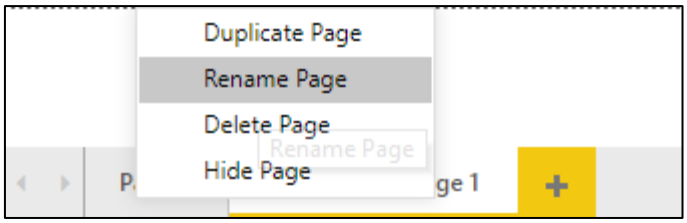


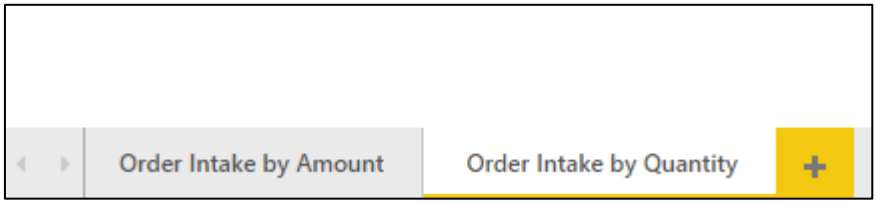
Add it to the report as follows:



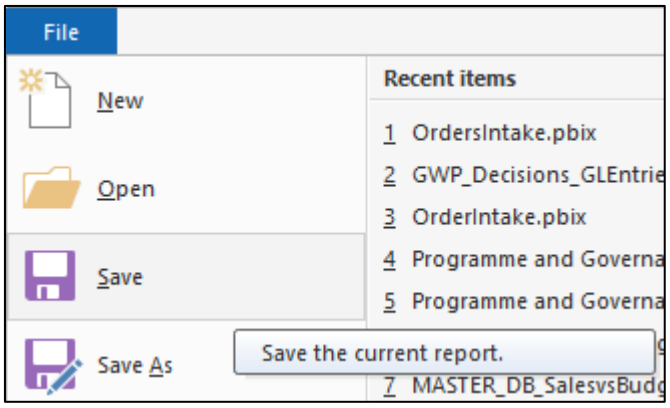


Rename the pages:



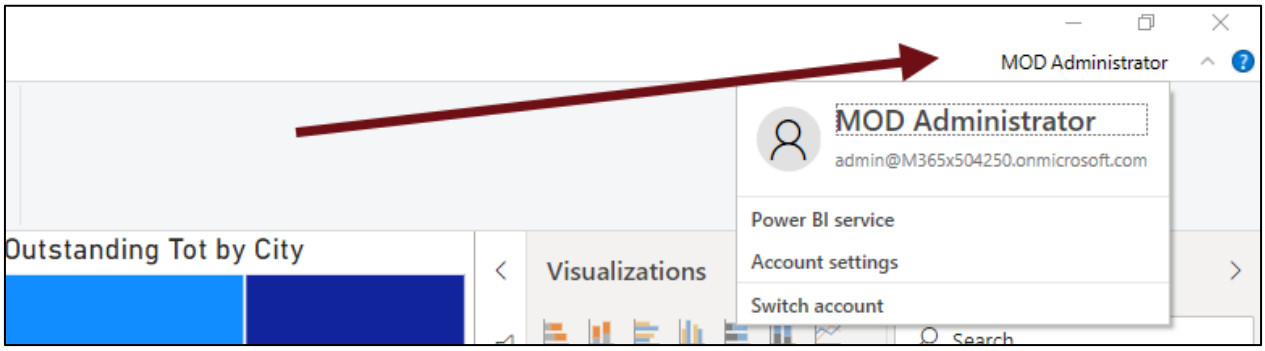


The report is now ready to publish. Please **Save** it now:



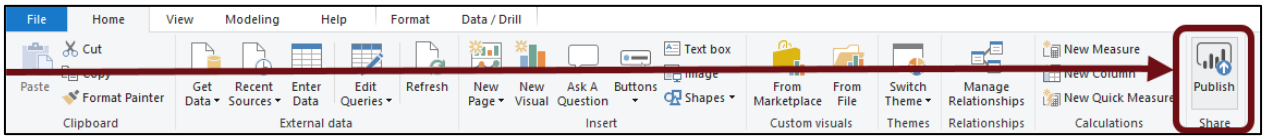
Publish your pbix file to the Power BI Service

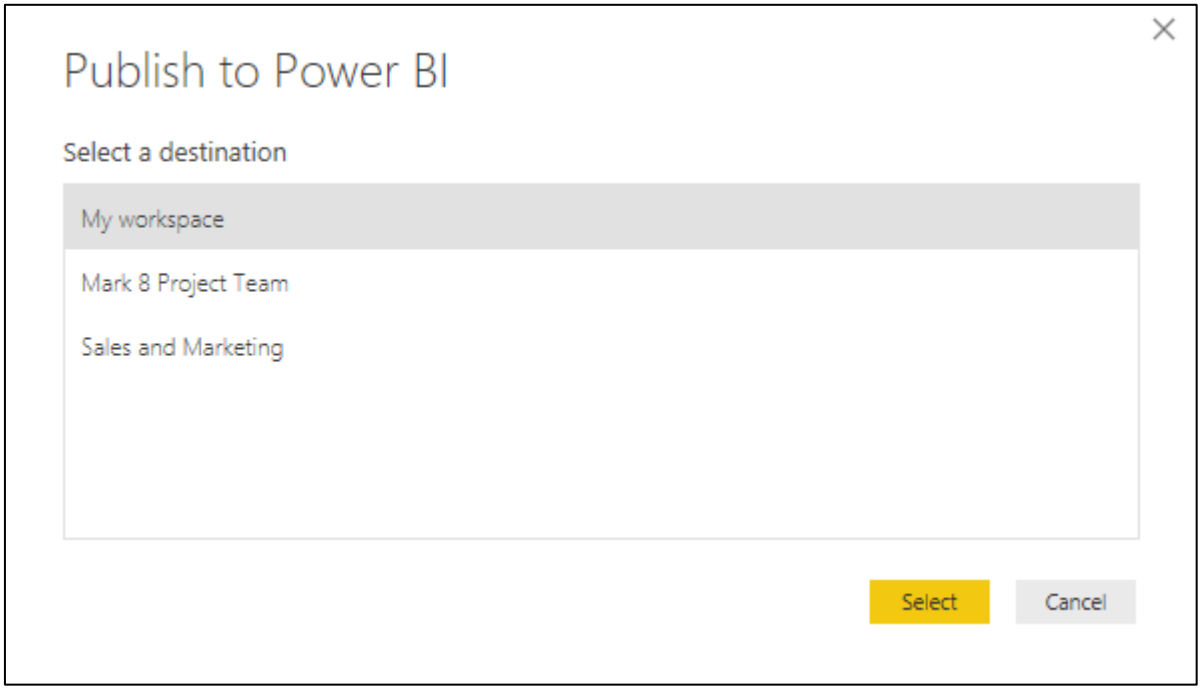
To publish our report, you need to sign in:

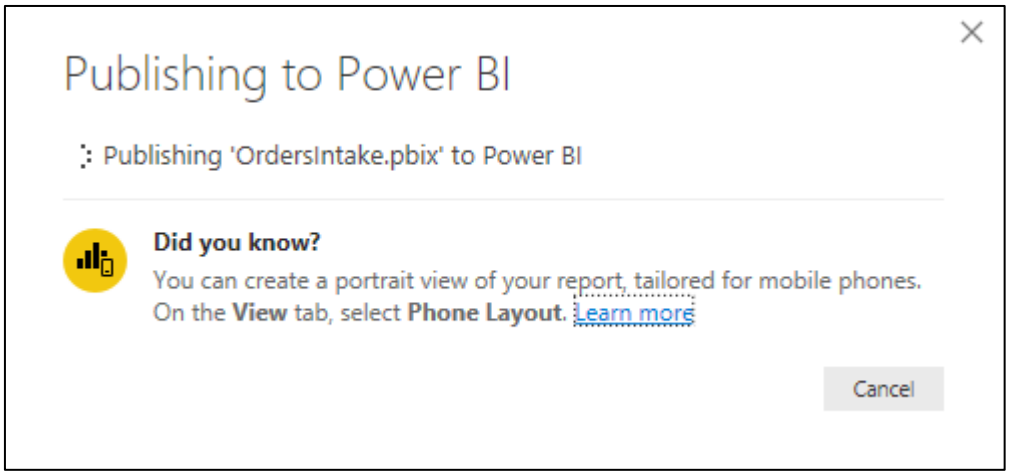


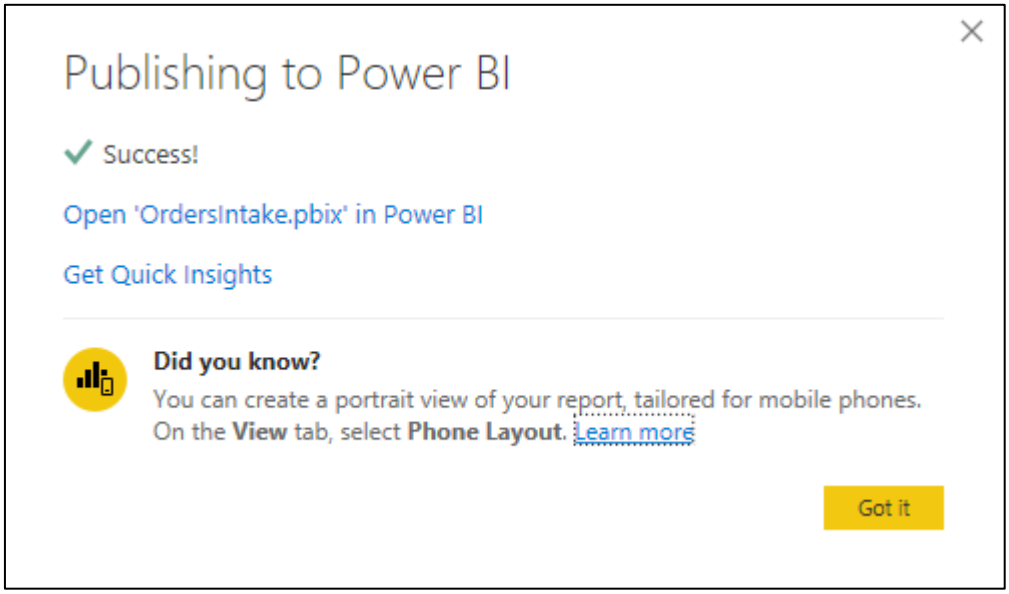
Use your email address that you used to create your Power BI account.

Then **Publish** the report:

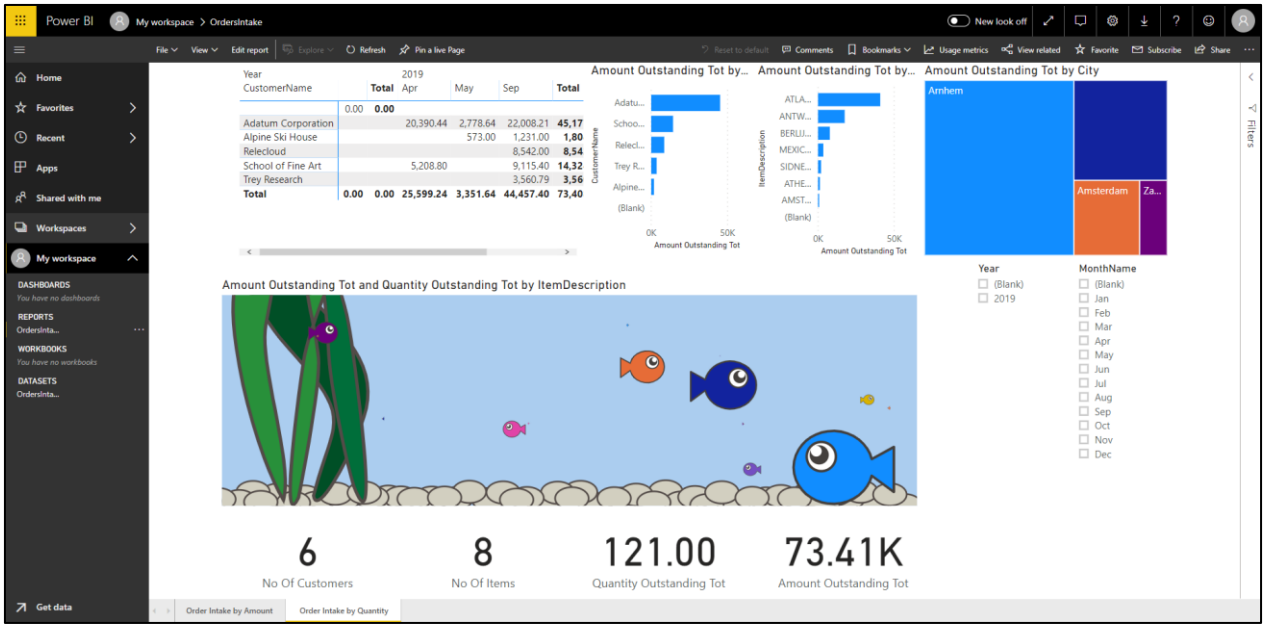



Publish to Power BI
Select a destination
My workspace
Mark 8 Project Team
Sales and Marketing
Select
Cancel


Publishing to Power BI
Publishing 'OrdersIntake.pbix' to Power BI
Did you know?
You can create a portrait view of your report, tailored for mobile phones.
On the View tab, select Phone Layout. Learn more
Cancel

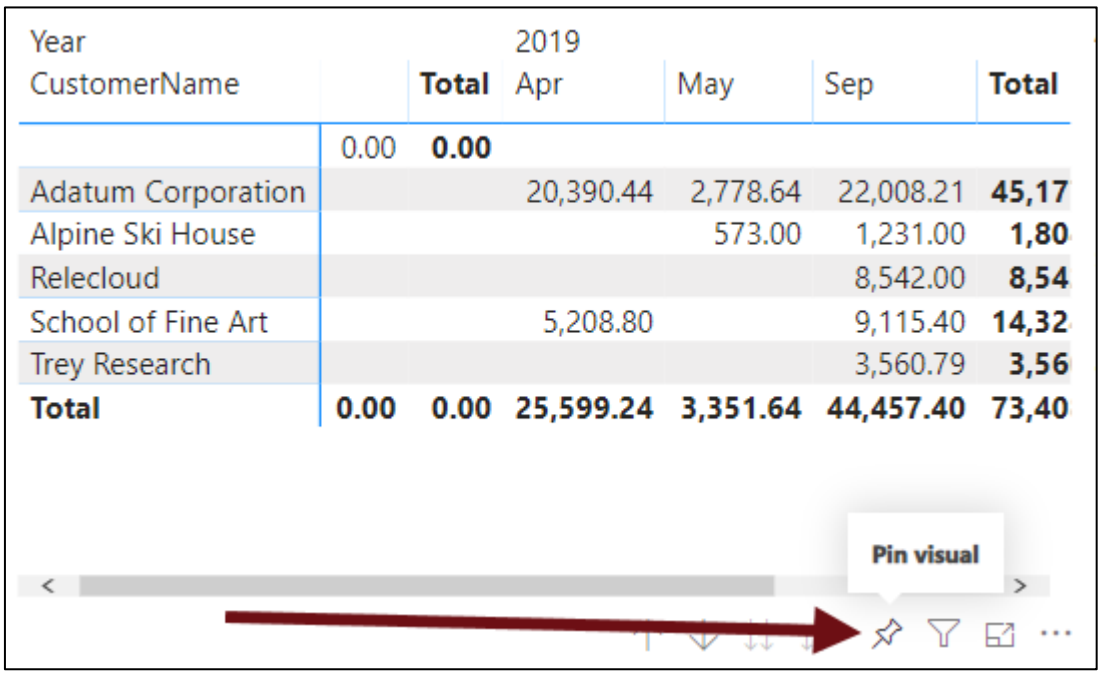


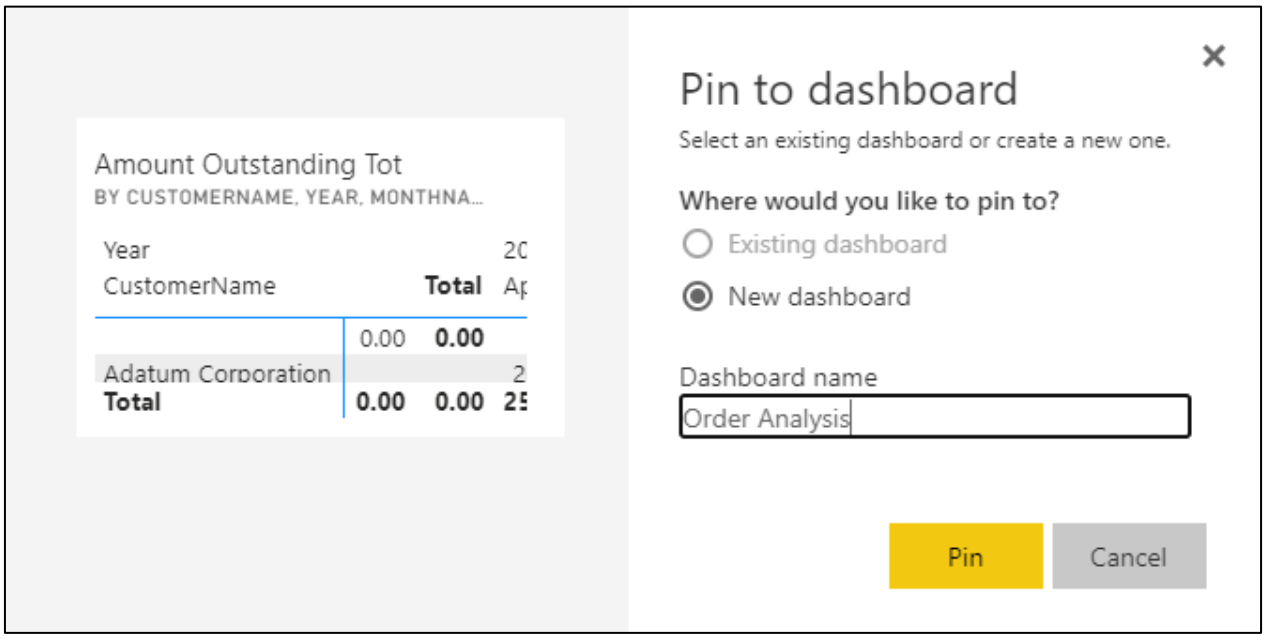
Go to powerbi.microsoft.com or select the link in the popup:



Create a Dashboard

To create a Dashboard, select the Pin icon on one of the visuals and pin it:





Once you have pinned your visuals to the dashboard, you can go to the dashboard and resize the visuals:

PLATAAN

Amount Outstanding Tot
BY CUSTOMERNAME, YEAR, MONTHNAME

Year			2019			
CustomerName		Total	Apr	May	Sep	Total
	0.00	0.00				
Adatum Corporation			20,390.44	2,778.64	22,008.21	**45,177.29**
Alpine Ski House				573.00	1,231.00	**1,804.00**
Relecloud					8,542.00	**8,542.00**
School of Fine Art			5,208.80		9,115.40	**14,324.20**
Trey Research					3,560.79	**3,560.79**
Total	**0.00**	**0.00**	**25,599.24**	**3,351.64**	**44,457.40**	**73,408.28**

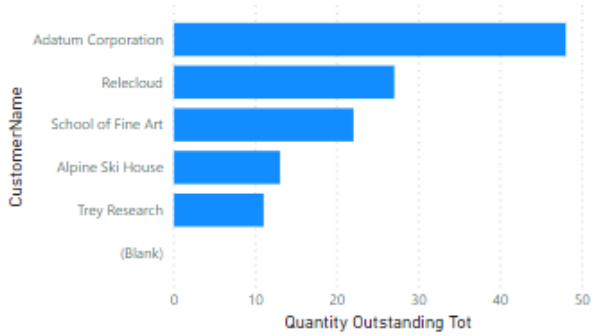
No Of Customers
6

No Of Items
8

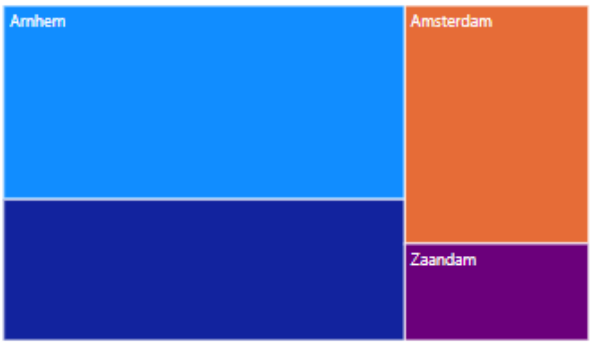
Quantity Outstanding Tot
121.00

Amount Outstanding Tot
73.41K

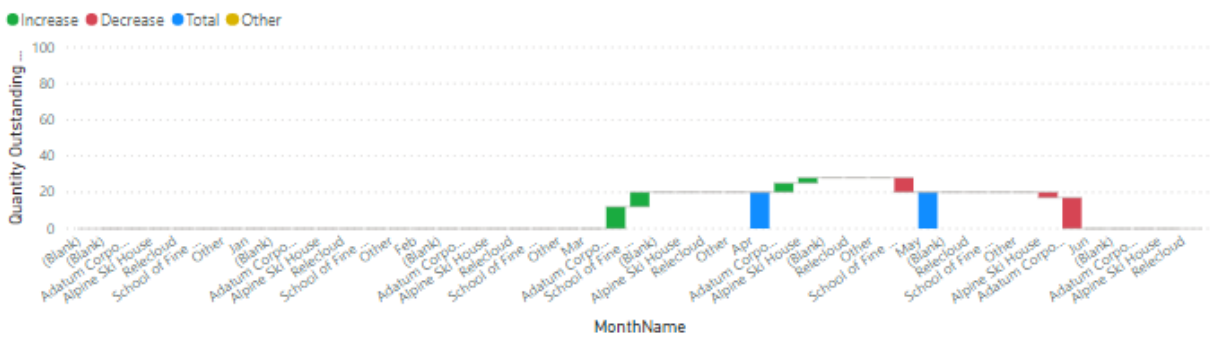
Quantity Outstanding Tot
BY CUSTOMERNAME



Quantity Outstanding Tot
BY CITY



Quantity Outstanding Tot
BY CUSTOMERNAME, MONTHNAME



Our Power BI Order Intake dashboard is ready now, congratulations!