

The SMB Focused Dynamics Partner Community

DIRECTIONS
E M E A

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

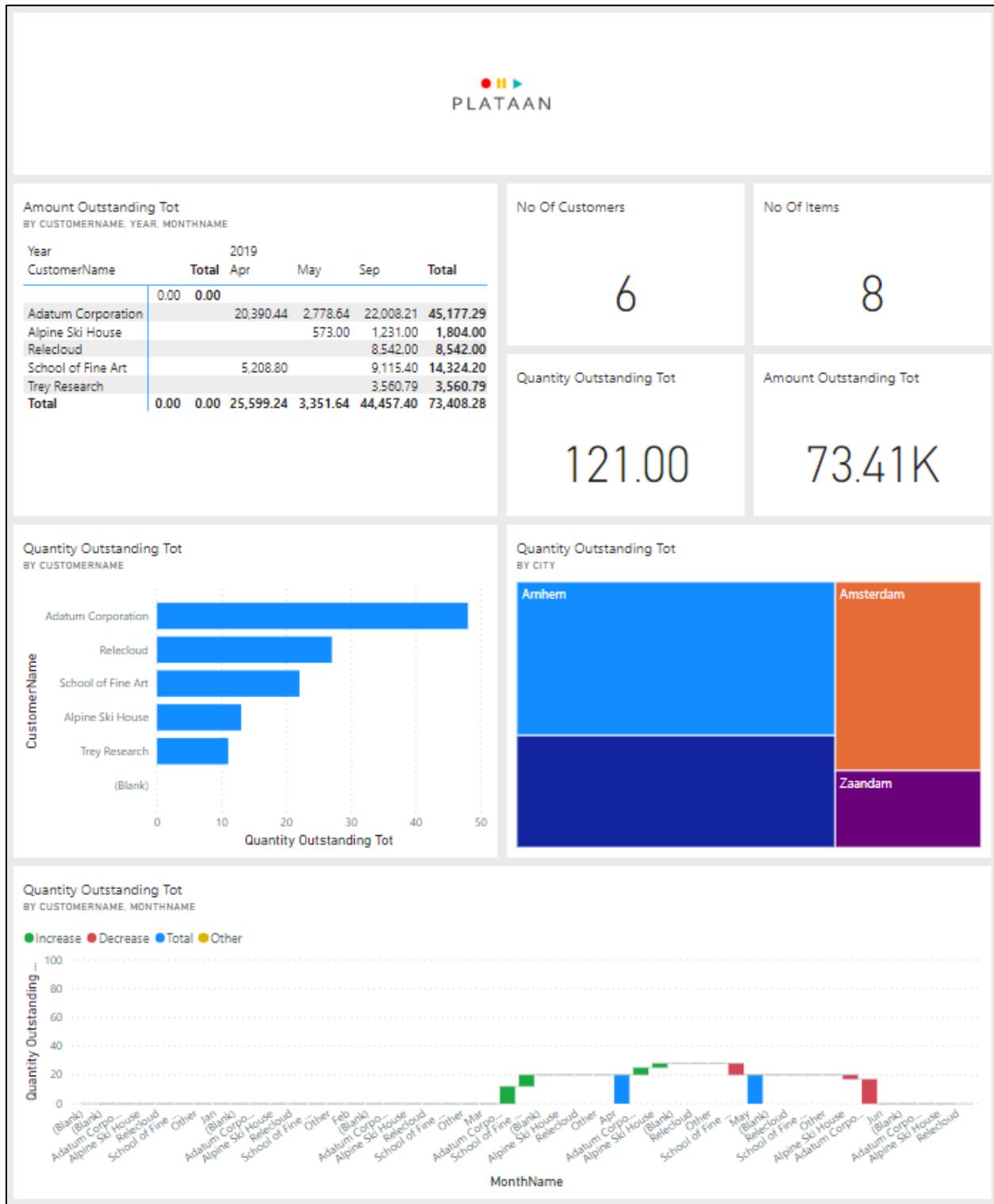
You can use your own office 365 credentials, if you have them. But be aware that we will use a Business Central Cloud tenant, create and publish an extension. If your company is already using their Business Central tenant, then it would strongly recommend you create a **new Office trial** instead.

Summary:

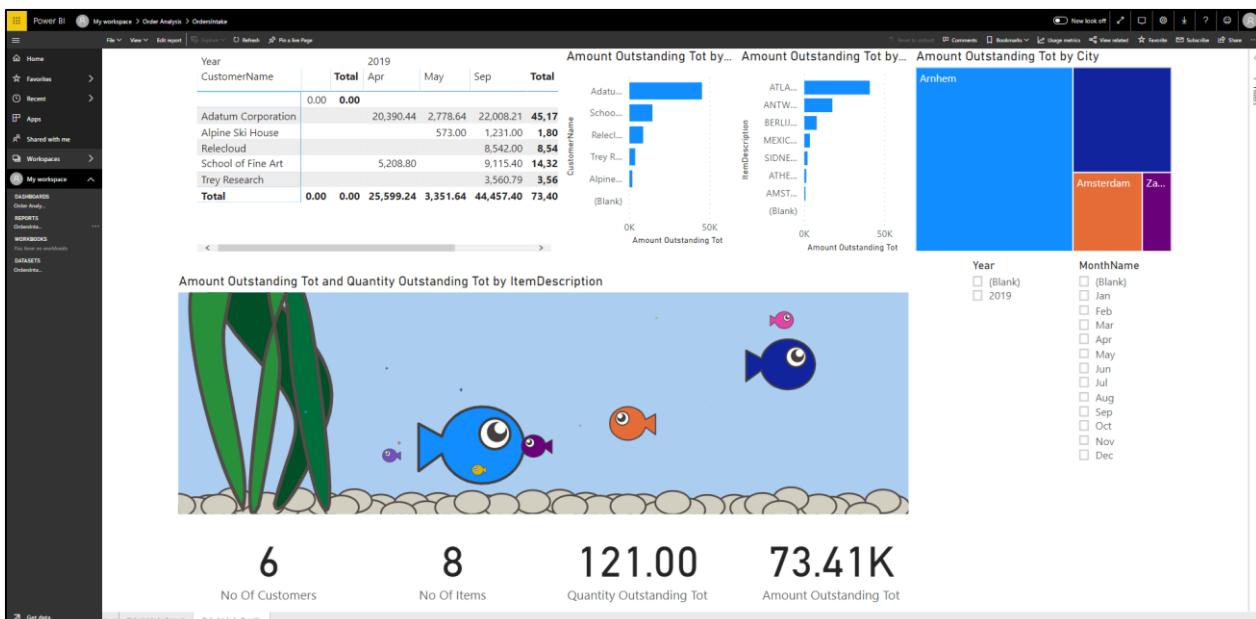
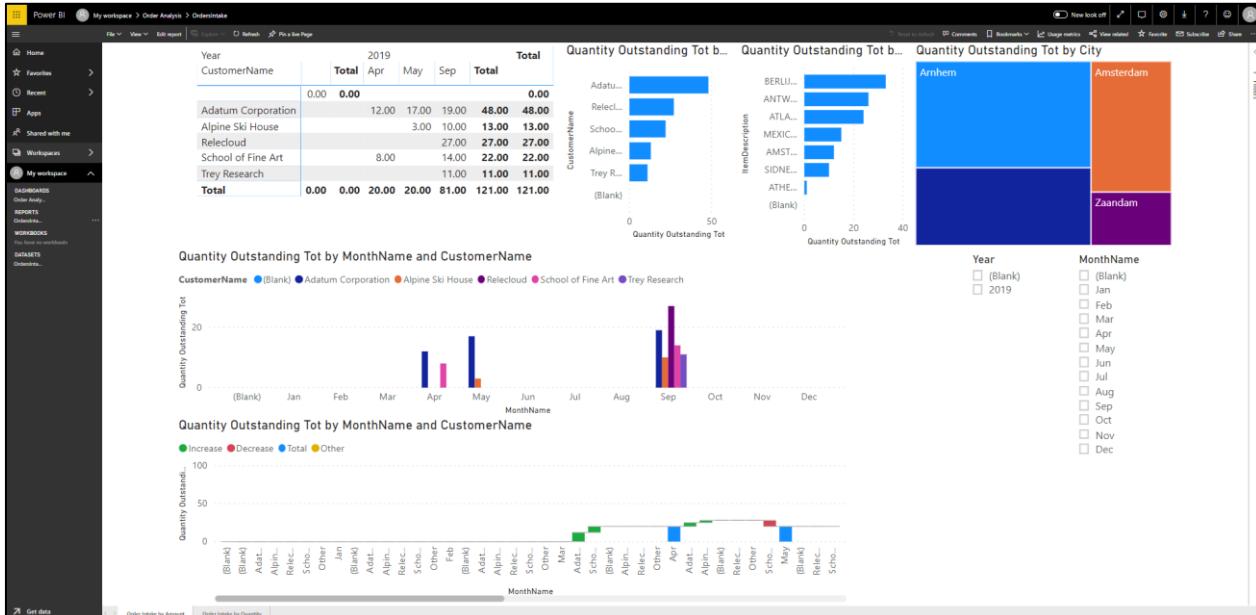
If you decide to use an already existing Office 365 environment, then do so at your own risk, you have been warned!

The objective

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



With an underlying report:



The approach

We will get there in the following steps:

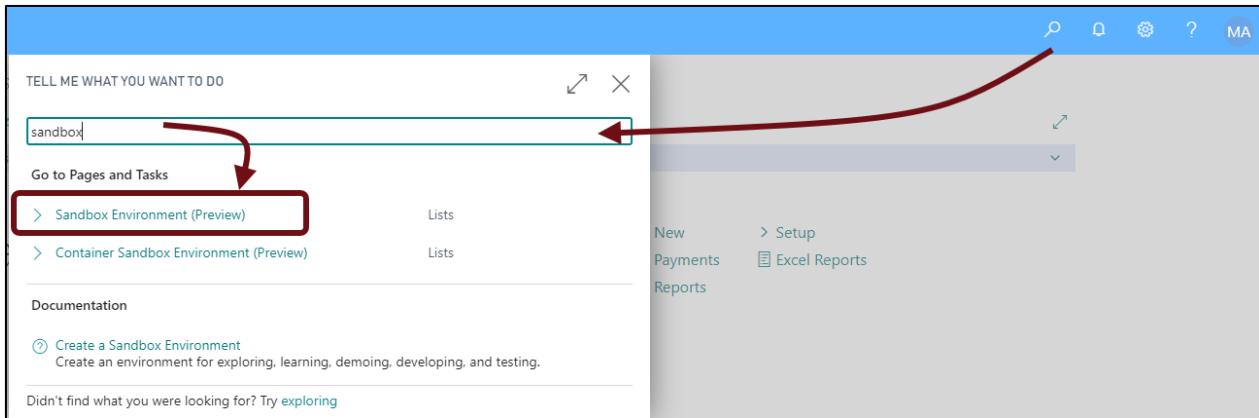
1. Create a new project in VS-Code
2. Add queries to the extension
3. Publish the queries as web services
4. Obtain the ODATA web services URL
5. Import the ODATA queries in Power BI Desktop
6. Create the Data Model and relations
7. Enrich the Data Model with DAX
8. Create the Report(s)
9. Publish your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

SANDBOX ENVIRONMENT (PREVIEW)

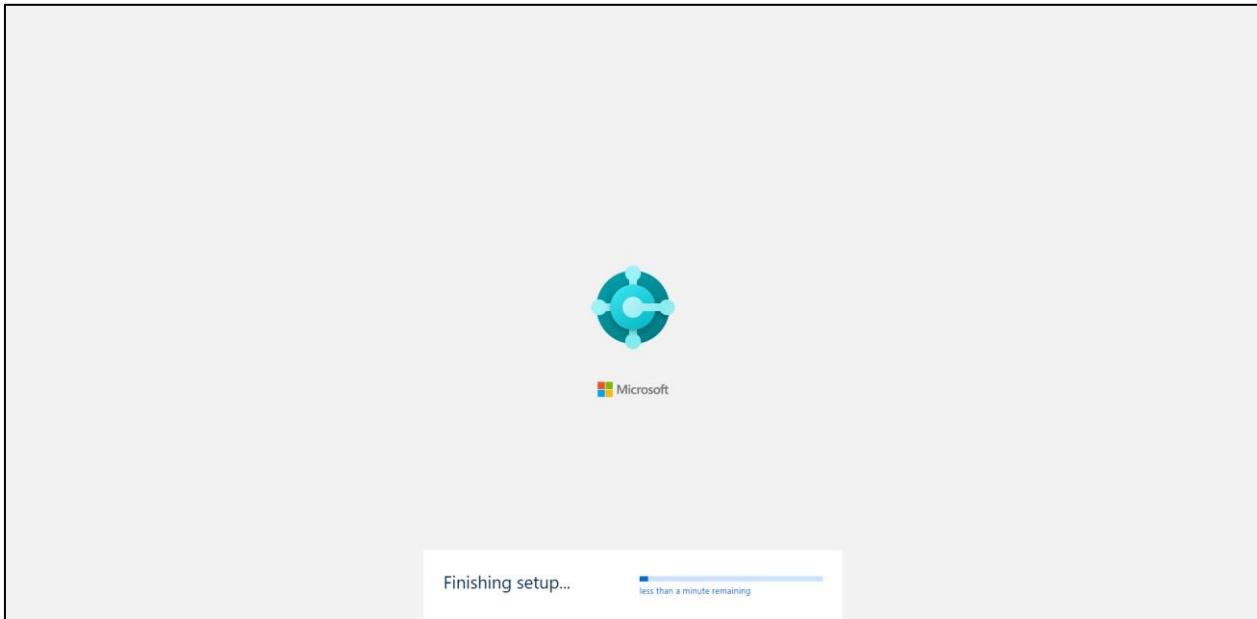
In addition to your production environment, you can create an environment for sandbox activities, such as test, demonstration, or development.

A new sandbox environment (preview) only contains the CRONUS demonstration company. Actions that you perform in the sandbox environment (preview) do not affect data or settings in your production environment.

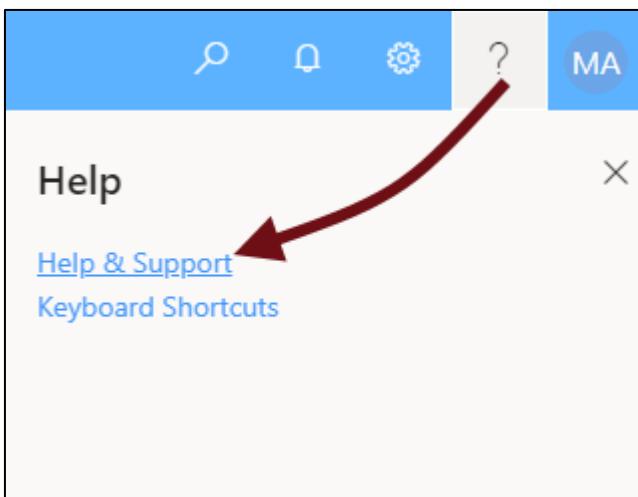
This Sandbox environment feature is provided as a free preview solely for testing, development and evaluation. You will not use the sandbox in a live operating environment. Microsoft may, in its sole discretion, change the Sandbox environment or subject it to a fee for a final, commercial version, if any, or may elect not to release one.

Create Reset Open

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:



INFORMATION

Help and Support

Find it

- Learn more about your current task or page ([Business Manager](#))
- Find answers in the [Help](#)
- Look for answers in the [community](#)
- Read the [blog](#)
- See which capabilities are [coming soon](#)
- Learn more about [Help and Support](#)

Do more with your trial

- [Contact our sales team](#) to start a subscription
- Explore pre-built solutions from our partners that [extend](#) the capabilities of Business Central
- Get started with [Business Central consulting services](#) and find partners with expertise in your industry or region.

Give feedback

- Do you have a great idea that you would like to see in Business Central?
- Register your idea and get others in the community to vote for it at aka.ms/BusinessCentralIdeas

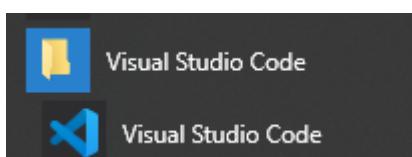
Troubleshooting

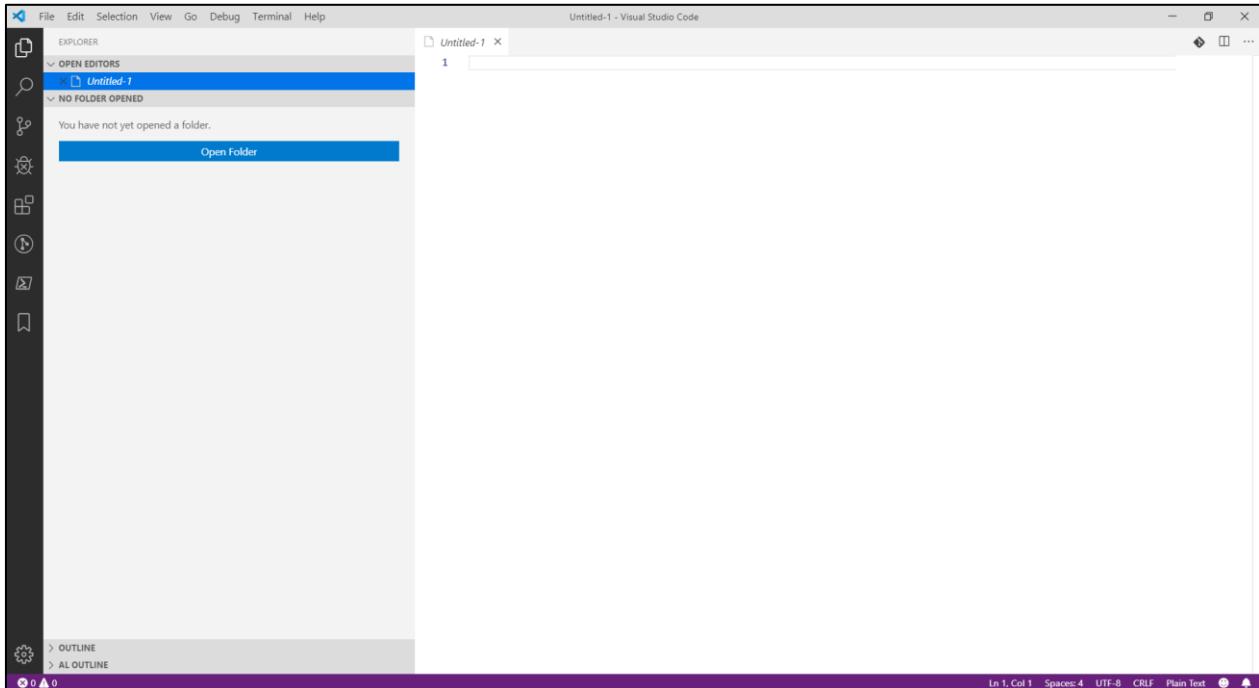
Version: Platform 15.0.36510.0 + Application 15.0.36560.0
Azure AD tenant: ac323d68-14d3-4af3-8df8-58268ad1c0f1, Environment: Production

[View the last known error](#)
[Inspect pages and data](#)

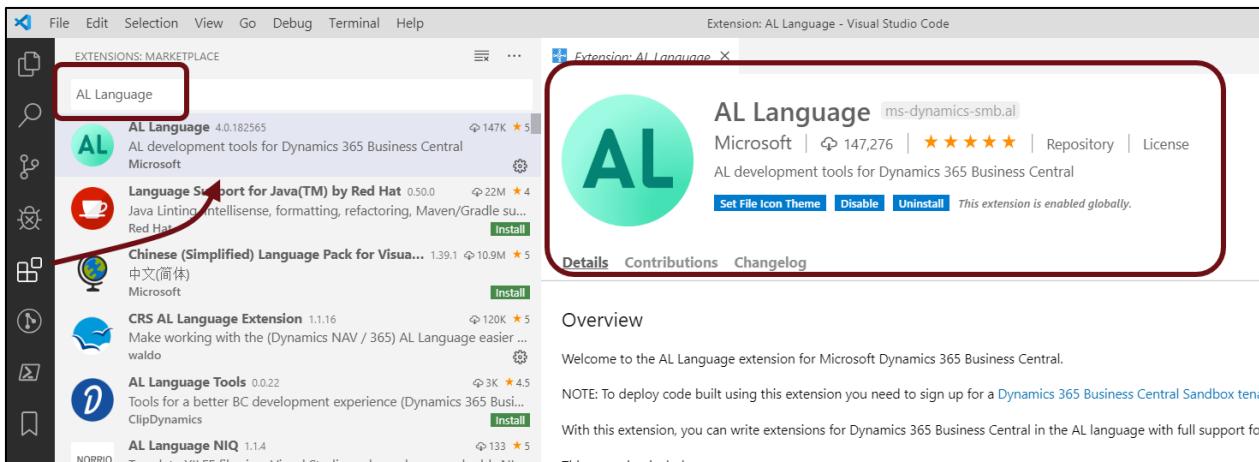


Now launch Visual Studio Code:



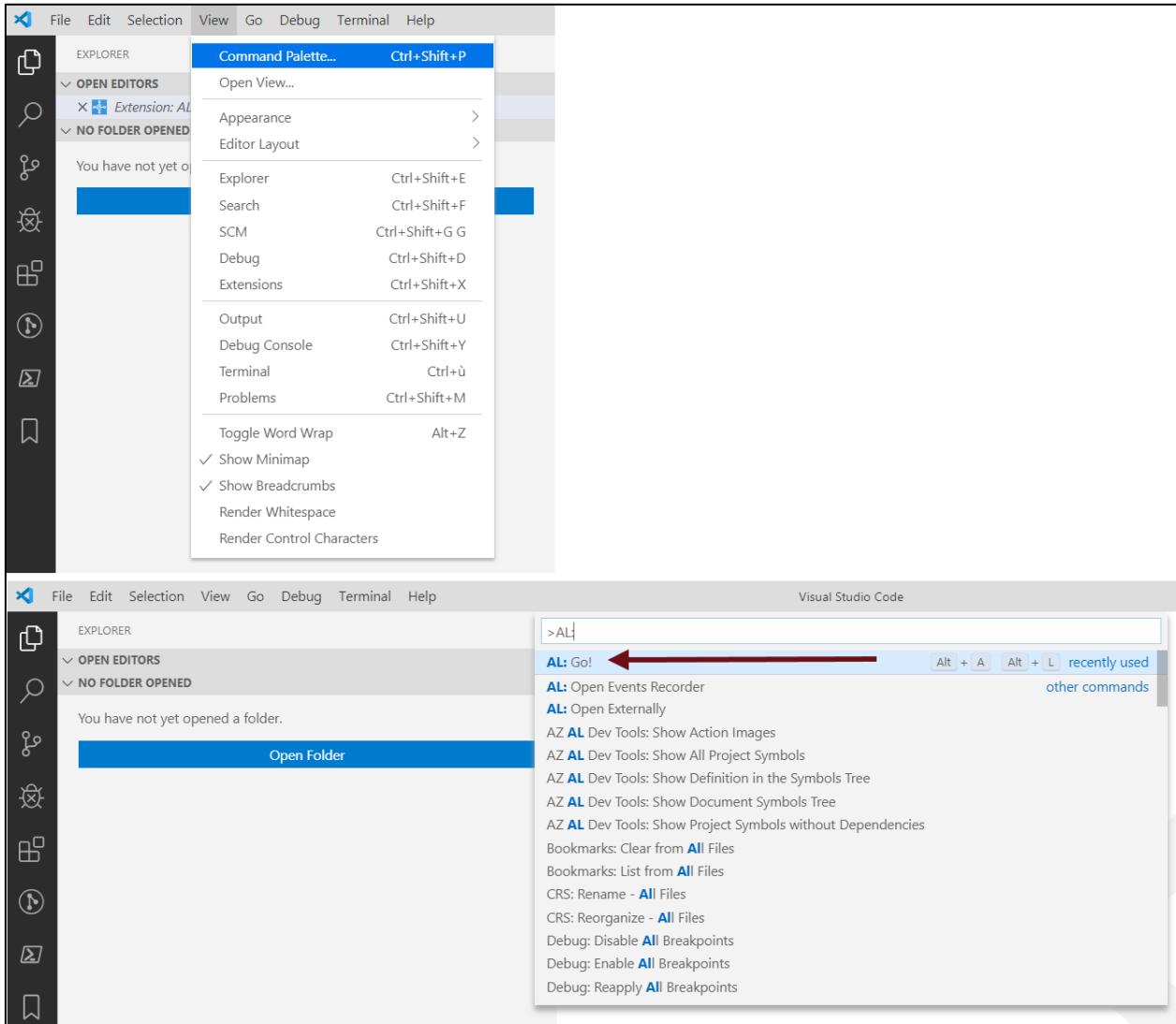


In Visual Studio Code, verify in the extensions window that the AL extension is available:



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

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



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

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

Please choose a path to a new empty folder (Press 'Enter' to confirm or 'Escape' to cancel)

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

Select target platform 4.0:

Target platform

- 4.0 Business Central 2019 release wave 2
- 3.0 Business Central Spring '19 Release
- 2.0 Business Central Fall '18 Release
- 1.0 Business Central Spring '18 Release

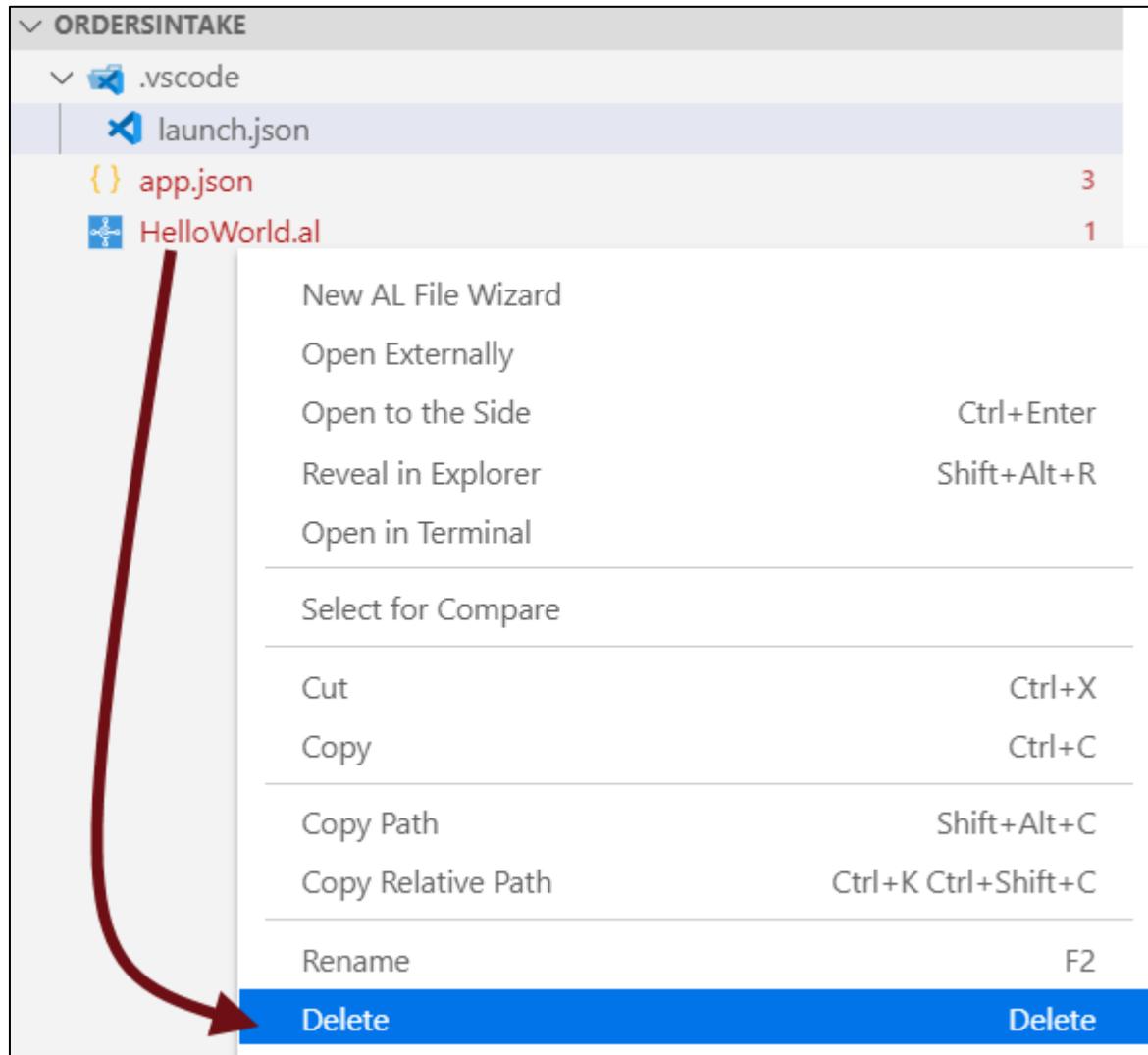
Then select Microsoft cloud sandbox:

Please choose the server:

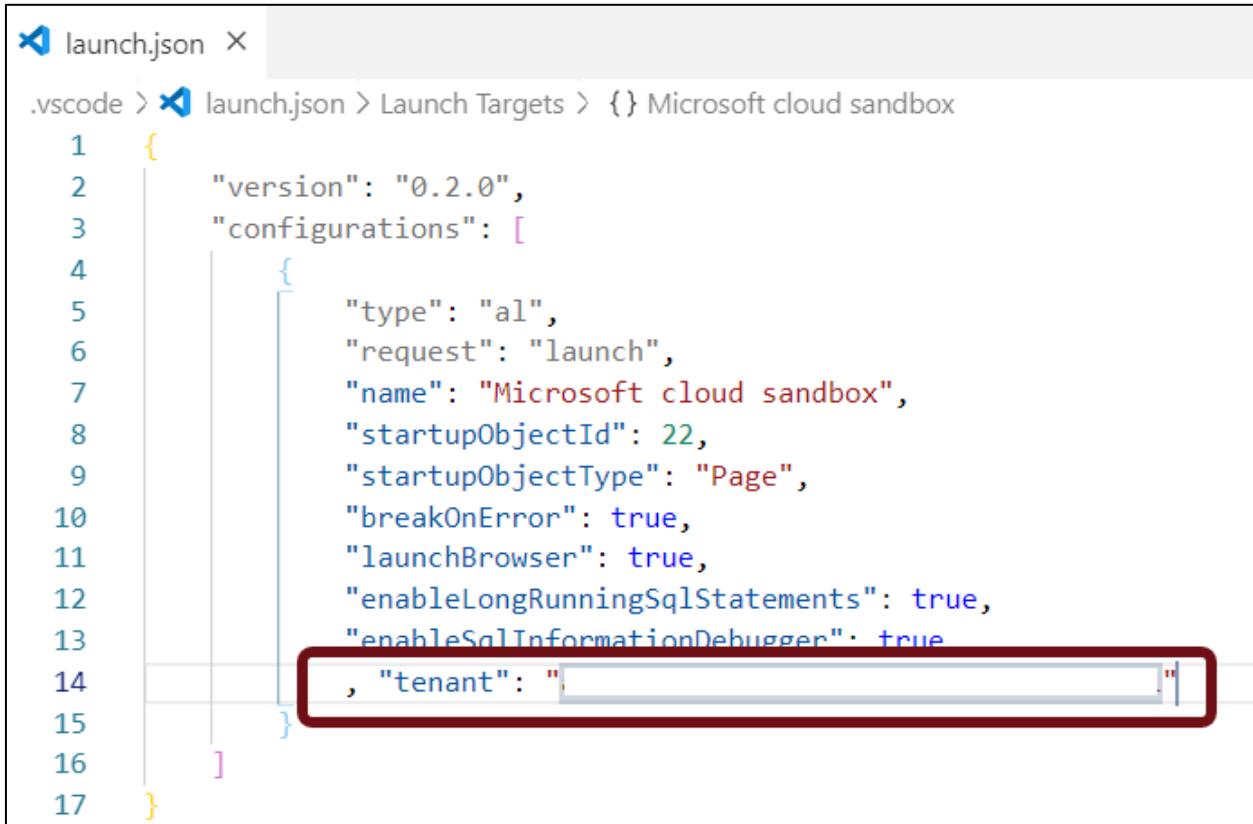
- Microsoft cloud sandbox
- Microsoft cloud sandbox
- Your own server
<http://localhost>, BC150, tenant default

A project is created.

Now delete the HelloWorld.al file:



In the launch.json file enter the following information:

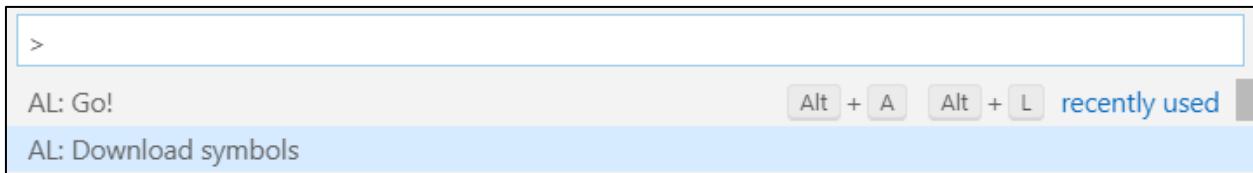


```
launch.json X
.vscode > launch.json > Launch Targets > {} Microsoft cloud sandbox

1  {
2    "version": "0.2.0",
3    "configurations": [
4      {
5        "type": "al",
6        "request": "launch",
7        "name": "Microsoft cloud sandbox",
8        "startupObjectId": 22,
9        "startupObjectType": "Page",
10       "breakOnError": true,
11       "launchBrowser": true,
12       "enableLongRunningSqlStatements": true,
13       "enableSqlInformationDebugger": true
14       , "tenant": ""
15     }
16   ]
17 }
```

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:

 **CONTOSO** demo

Aanmelden

admin@M365x504250.onmicrosoft.com X

[Hebt u geen toegang tot het account?](#)

[Aanmeldingsopties](#)

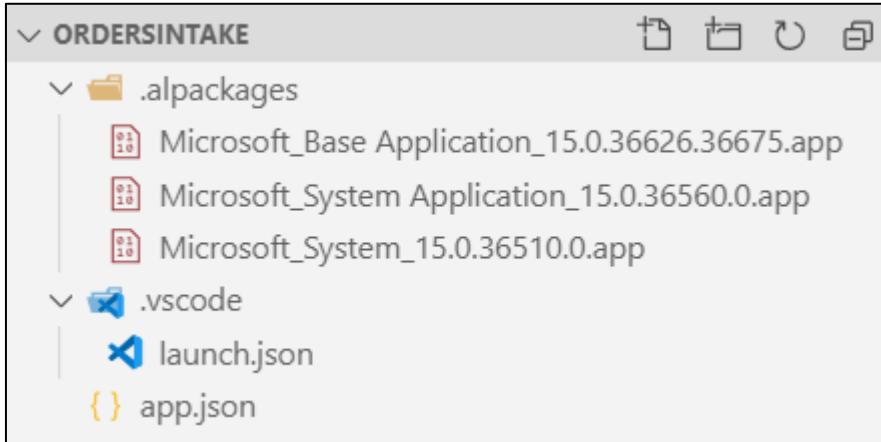
Volgende

Contoso

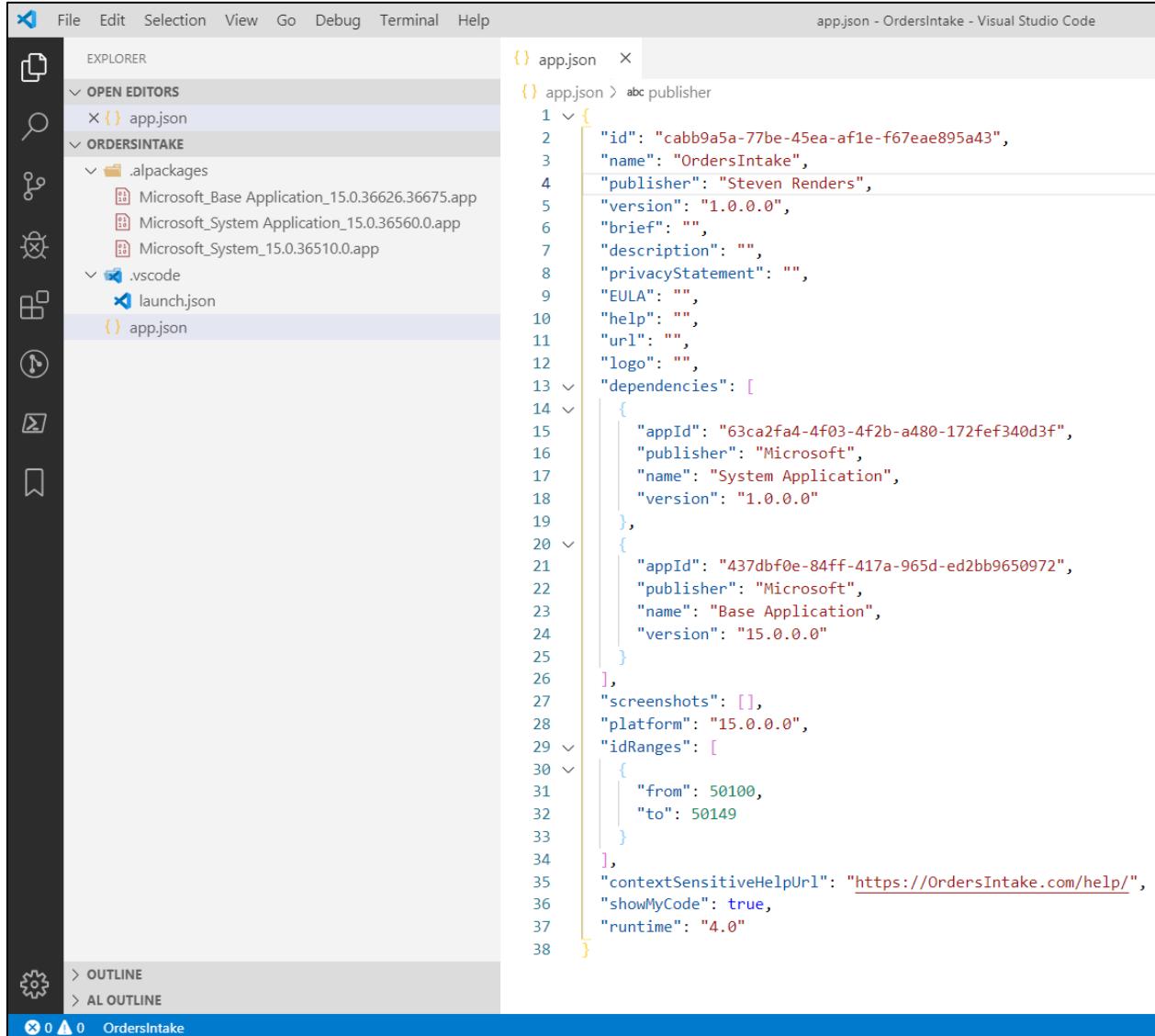
[Gebruiksvoorwaarden](#) [Privacy en cookies](#) [...](#)

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:



```

{
  "id": "cabb9a5a-77be-45ea-af1e-f67eae895a43",
  "name": "OrdersIntake",
  "publisher": "Steven Renders",
  "version": "1.0.0.0",
  "brief": "",
  "description": "",
  "privacyStatement": "",
  "EULA": "",
  "help": "",
  "url": "",
  "logo": "",
  "dependencies": [
    {
      "appId": "63ca2fa4-4f03-4f2b-a480-172fef340d3f",
      "publisher": "Microsoft",
      "name": "System Application",
      "version": "1.0.0.0"
    },
    {
      "appId": "437dbf0e-84ff-417a-965d-ed2bb9650972",
      "publisher": "Microsoft",
      "name": "Base Application",
      "version": "15.0.0.0"
    }
  ],
  "screenshots": [],
  "platform": "15.0.0.0",
  "idRanges": [
    {
      "from": 50100,
      "to": 50149
    }
  ],
  "contextSensitiveHelpUrl": "https://OrdersIntake.com/help/",
  "showMyCode": true,
  "runtime": "4.0"
}

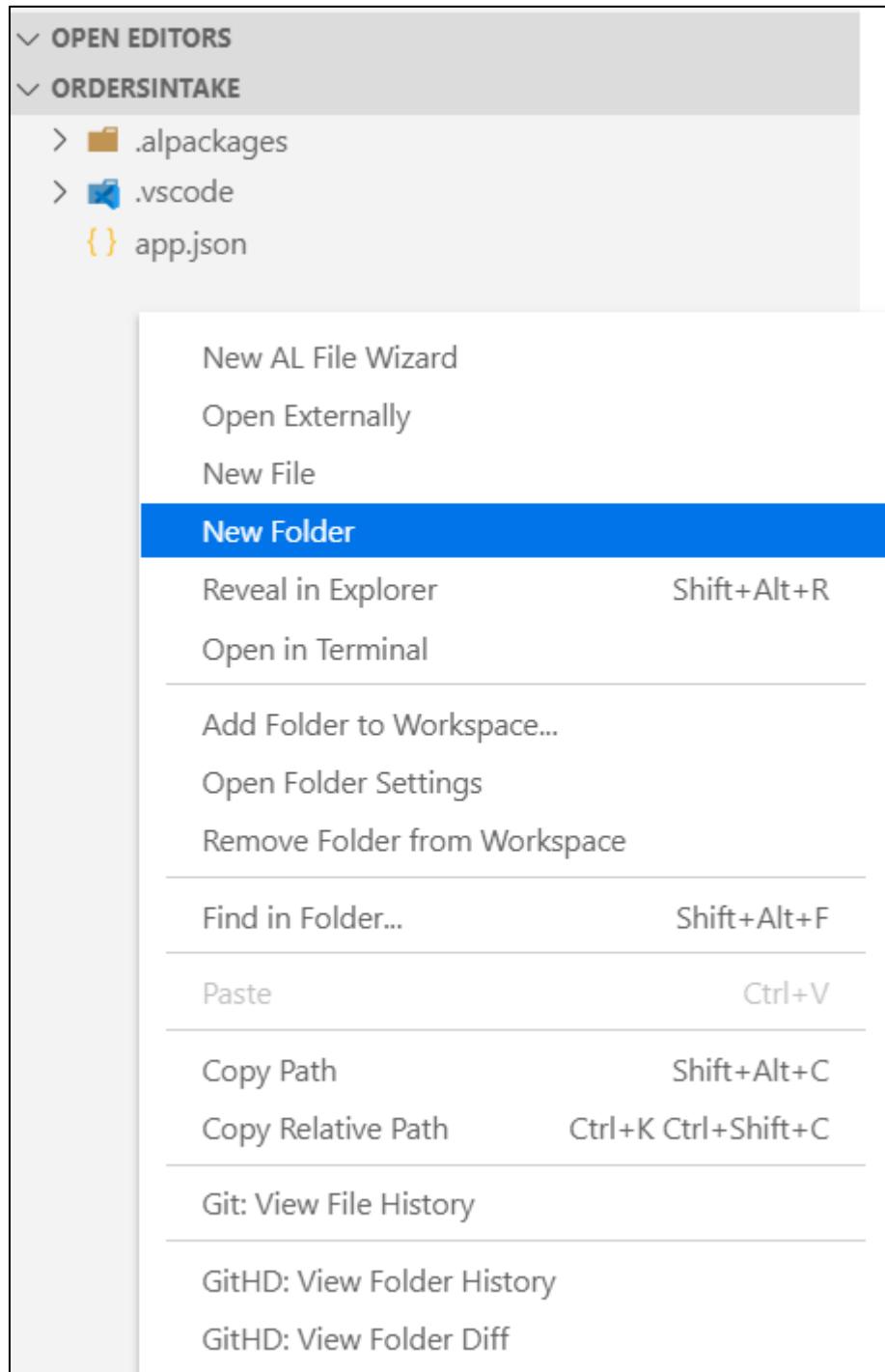
```

We are now ready to add queries to the extension.

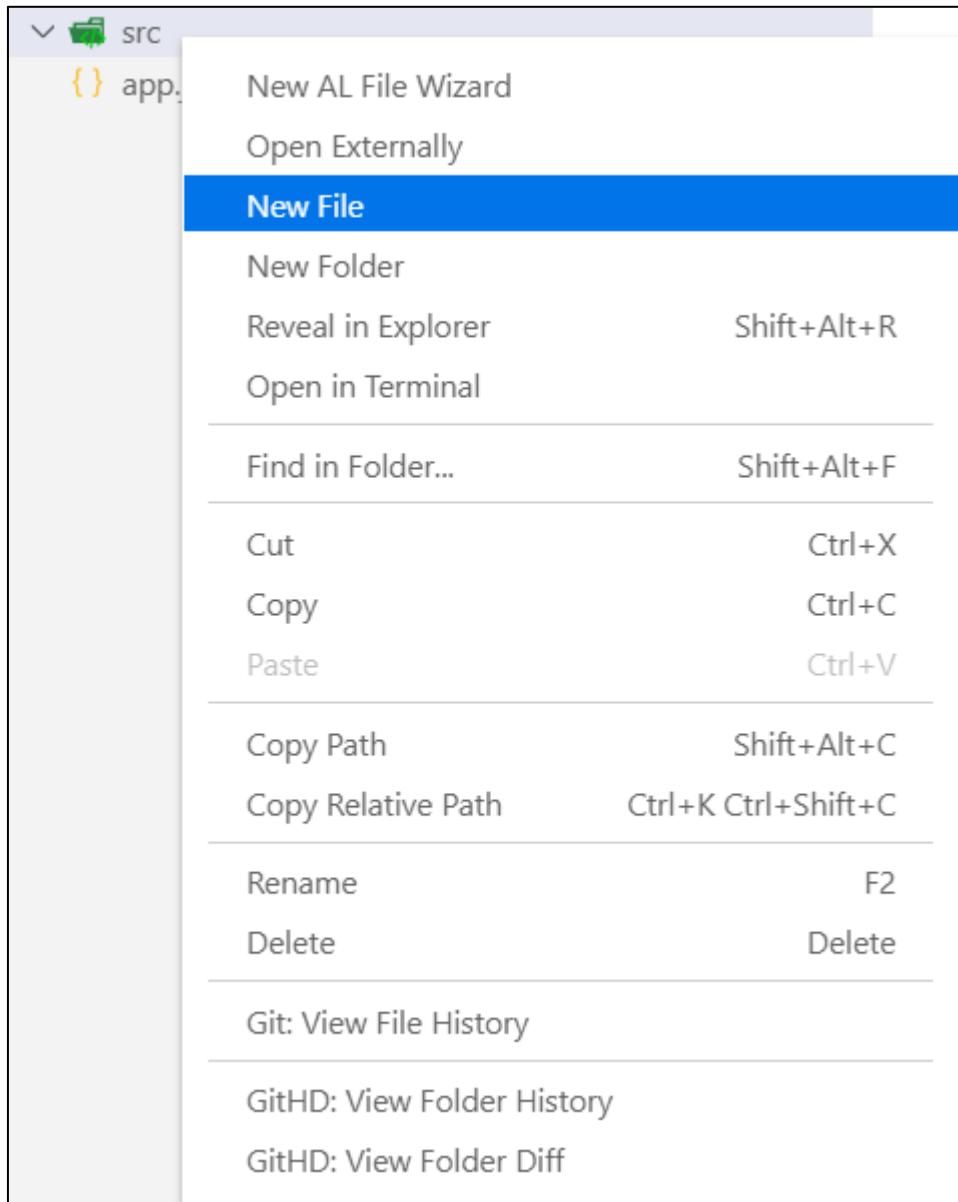
Add queries to the extension

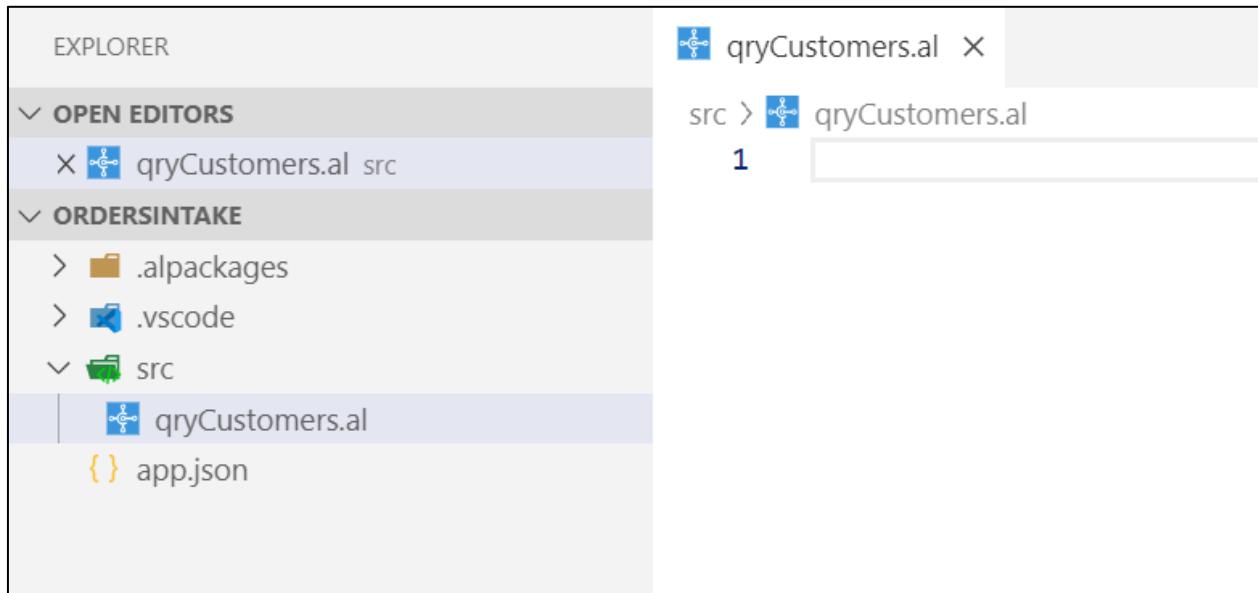
In this extension we will add 3 queries: **qryCustomers**, **qryItems** and **qryOrderIntake**.

Create a new folder named: **src**

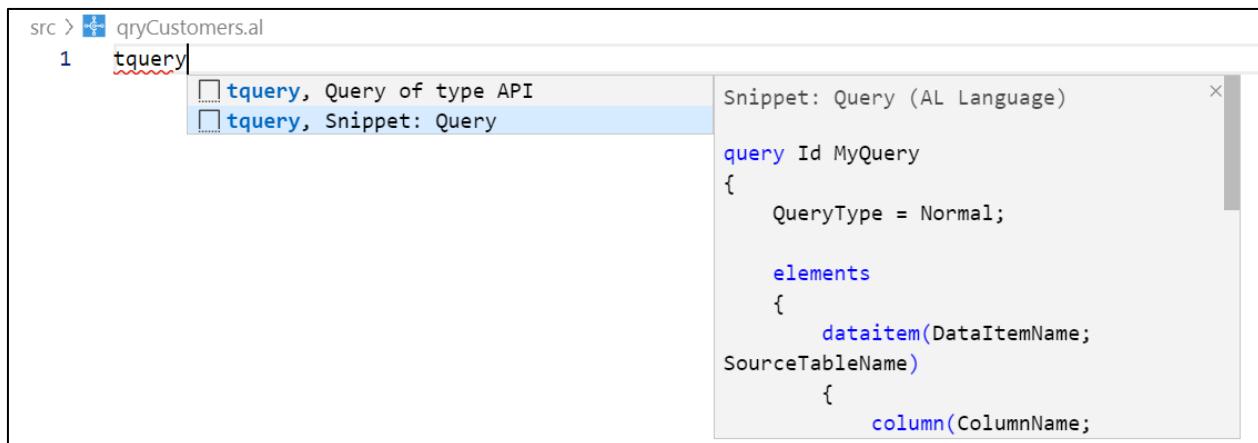


Add a new file to the folder and name it: **qryCustomers.al**:





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     {
7         0 references
8         dataitem(DataItemName; SourceTableName)
9         {
10            0 references
11            column(ColumnName; SourceFieldName)
12            {
13                0 references
14                filter(FilterName; SourceFieldName)
15                {
16                }
17            }
18        }
19
20        var
21        0 references
22        myInt: Integer;
23
24        trigger OnBeforeOpen()
25        begin
26        end;
27 }
```

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

```
src > qryCustomers.al > Query 50100 qryCustomers
      0 references
1  query 50100 qryCustomers
2  {
3      elements
4      {
5          dataitem(Customer; Customer)
6          {
7              column(CustomerNo; "No.")
8              {
9                  }
10             0 references
11             column(CustomerName; Name)
12             {
13                 }
14                 0 references
15                 column(City; City)
16                 {
17                     }
18                     }
19     }
```

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

EXPLORER

OPEN EDITORS

ORDERSINTAKE

src > qryItems.al > Query 50102 qryItems

0 references

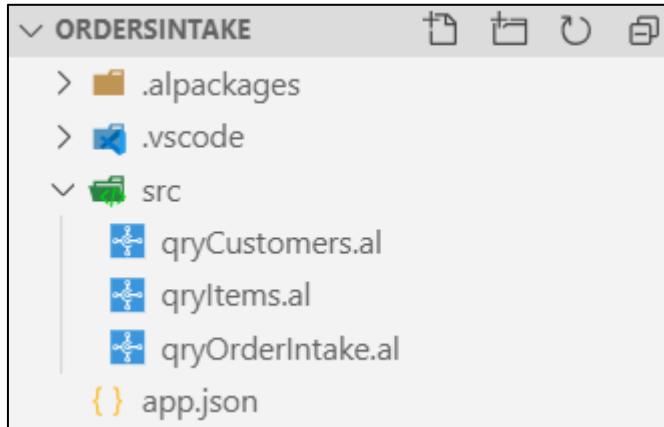
```
1 query 50102 qryItems
2 {
3     elements
4     {
5         0 references
6             dataitem(Item; Item)
7             {
8                 0 references
9                     column(ItemNo; "No.")
10                {
11                }
12                0 references
13                    column(ItemDescription; Description)
14                    {
15                    }
16                    0 references
17                        column(UnitCost; "Unit Cost")
18                        {
19                        }
20                        }
21                        }
22                }
23 }
```

```

src > qryOrderIntake.al > Query 50101 qryOrderIntake
4
5     elements
6     {
7         0 references
8             dataitem(SalesLine; "Sales Line")
9             {
10                0 references
11                    column(CustomerNo; "Sell-to Customer No.")
12                }
13                0 references
14                    column(LocationCode; "Location Code")
15                }
16                0 references
17                    column(ItemNo; "No.")
18                }
19                0 references
20                    column(ShipmentDate; "Shipment Date")
21                }
22                0 references
23                    column(QuantityOutstanding; "Outstanding Quantity")
24                }
25                0 references
26                    column(AmountOutstanding; "Outstanding Amount (LCY)")
27                }
28                0 references
29            }
30        }
31    }
32    }
33}
34}
35}

```

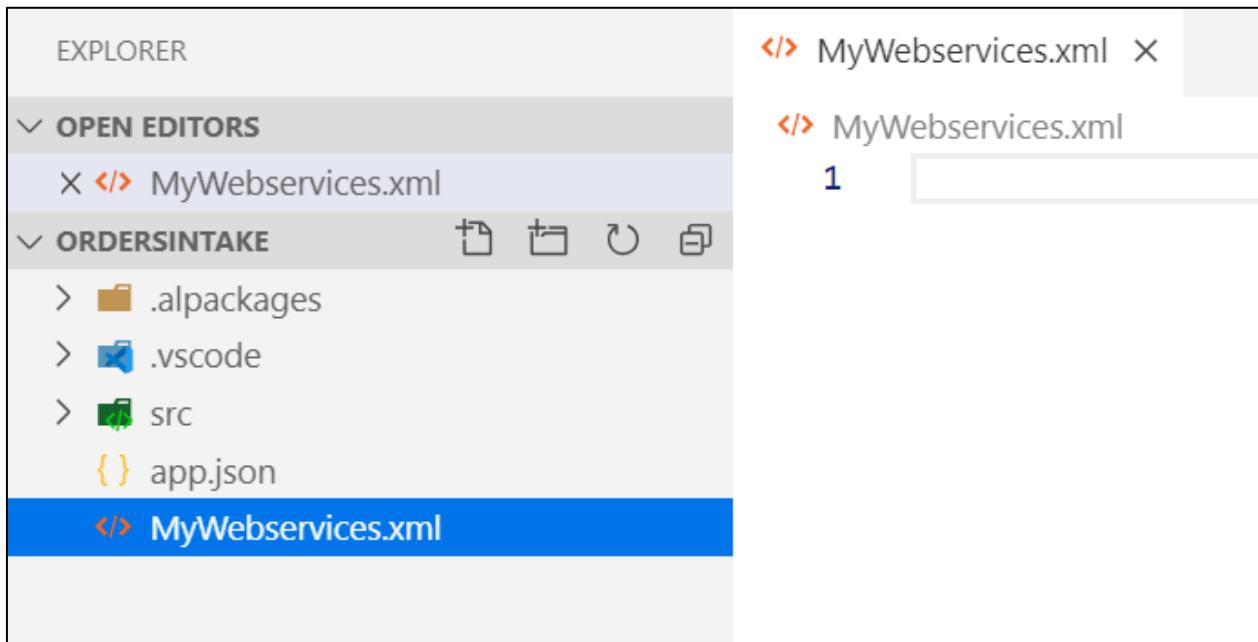
Your project now contains 3 queries:



[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 <  
  
</> MyWebservices.xml  
1   <?xml version="1.0" encoding="utf-8"?>  
2   <ExportedData>  
3       <TenantWebServiceCollection>  
4           <TenantWebService>  
5               <ObjectType>Query</ObjectType>  
6               <ServiceName>qryCustomers</ServiceName>  
7               <ObjectID>50100</ObjectID>  
8               <Published>true</Published>  
9           </TenantWebService>  
10          <TenantWebService>  
11              <ObjectType>Query</ObjectType>  
12              <ServiceName>qryItems</ServiceName>  
13              <ObjectID>50102</ObjectID>  
14              <Published>true</Published>  
15          </TenantWebService>  
16          <TenantWebService>  
17              <ObjectType>Query</ObjectType>  
18              <ServiceName>qryOrderIntake</ServiceName>  
19              <ObjectID>50101</ObjectID>  
20              <Published>true</Published>  
21          </TenantWebService>  
22      </TenantWebServiceCollection>  
23  </ExportedData>  
24 |
```

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

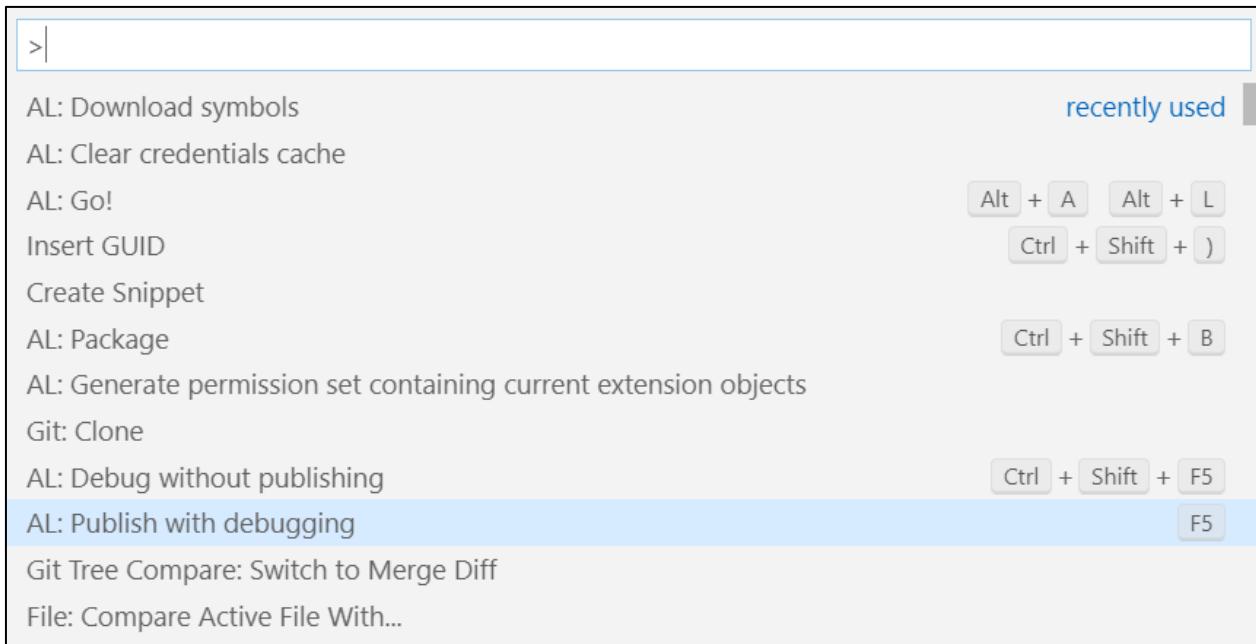
```

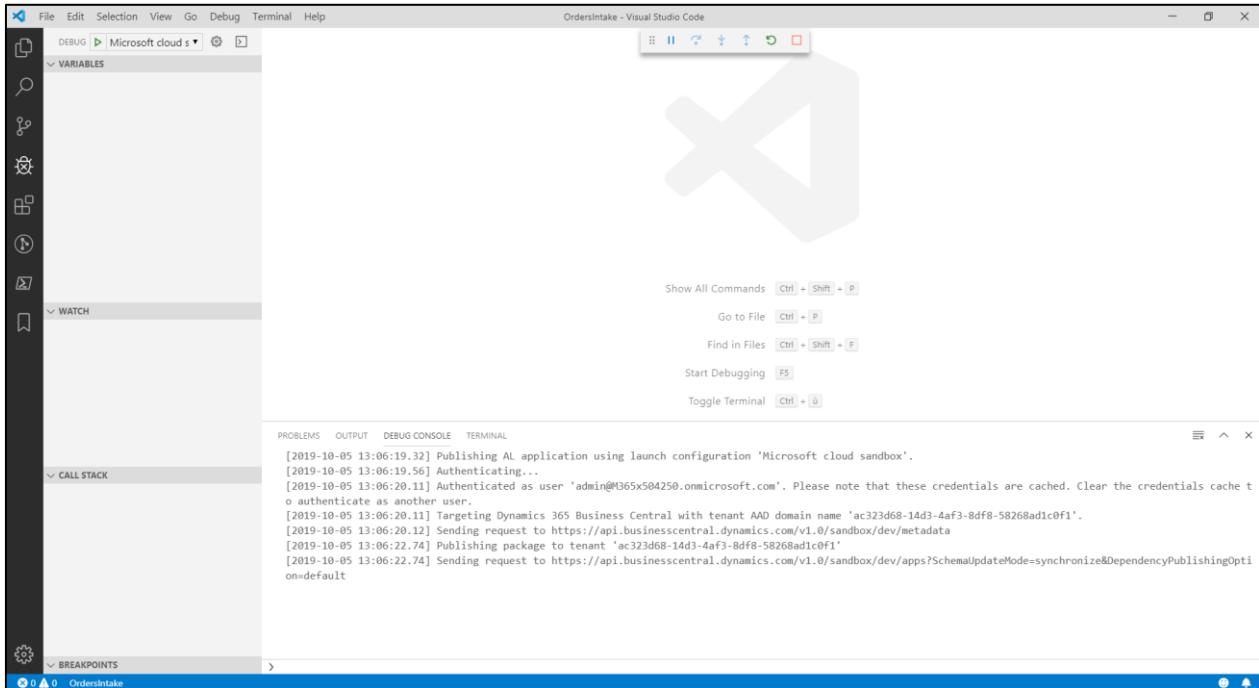
1  "version": "0.2.0",
2  "configurations": [
3      {
4          "type": "al",
5          "request": "launch",
6          "name": "Microsoft Local Database",
7          "startupObjectId": 810,
8          "startupObjectType": "Page",
9          "breakOnError": true,
10         "launchBrowser": true,
11         "enableLongRunningSqlStatements": true,
12         "enableSqlInformationDebugger": true,
13         , "tenant": "ac323d68-14d3-4af3-8df8-58268ad1c0f1"
14     }
15 ]
16
17

```

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:

Object Type	Object ID	Object Name	Service Name	All Tenants	Publ.	OData V4 URL	OData URL	SOAP URL
Query	269	Res. Ledger Entries	Res.LedgerEntries	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	Not applicable
Page	16	Chart of Accounts	Rekeningschema	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata
Query	50101	qryOrderIntake	qryOrderIntake	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	Not applicable
Query	50102	qryItems	qryItems	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	Not applicable
Query	50100	qryCustomers	qryCustomers	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	Not applicable
Page	6404	workflowPurchaseDocuments	purchaseDocuments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata
Page	6405	Purchase Document Line Entity	purchaseDocumentLines	<input type="checkbox"/>	<input checked="" type="checkbox"/>	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata	https://api.businesscentral.dynamics.com/v1.0/sandbox/dev/metadata

Here you should now see the queries in your extension.

Obtain the ODATA web services URL

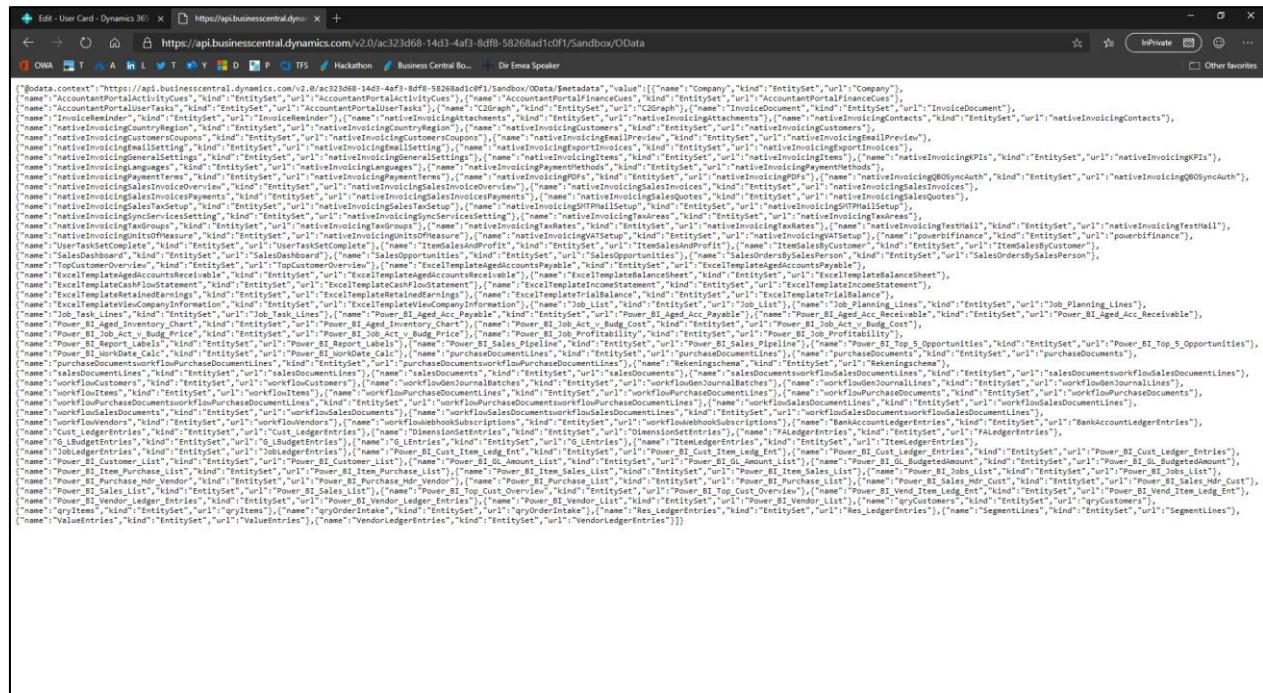
In the web services page, copy the **ODATA URL** of one of the queries. It should look like the following:

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

User Card | Work Date: 4/8/2019

MOD Administrator

Change Web Service Key Effective Permissions More options

General

User Name	ADMIN	State	Enabled
Full Name	MOD Administrator	Contact Email	admin@M365x504250.OnMicrosoft.com

Web Service Access

Web Service Access Key	6lUQzXej34BIRCnesBJlqjmobcRcmHzd7S... ...	Web Service Expiry Date
------------------------------	---	-------------------------------

Office 365 Authentication >

Your credentials are:

User Name & Web Service Access Key.

Sign in to access this site

Authorization required by https://api.businesscentral.dynamics.com

Username	ADMIN
Password

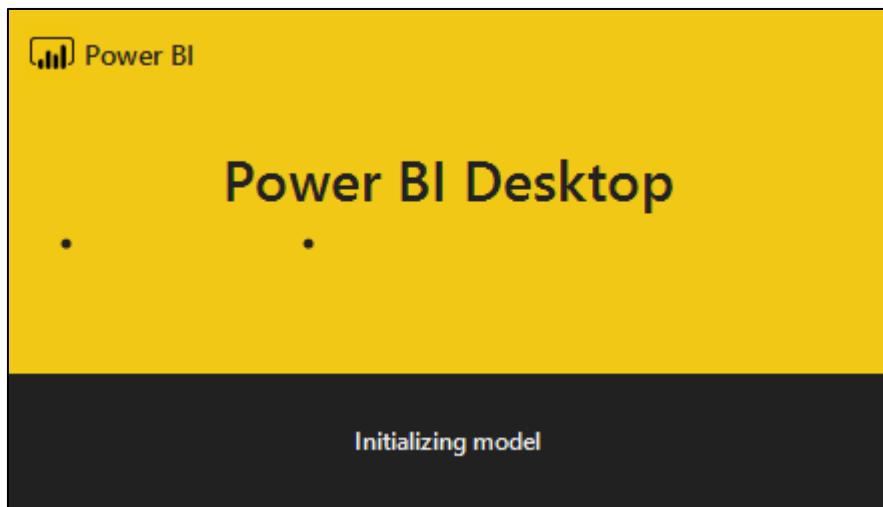
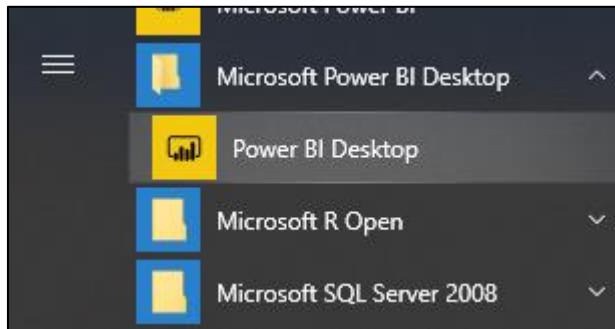
Sign in **Cancel**

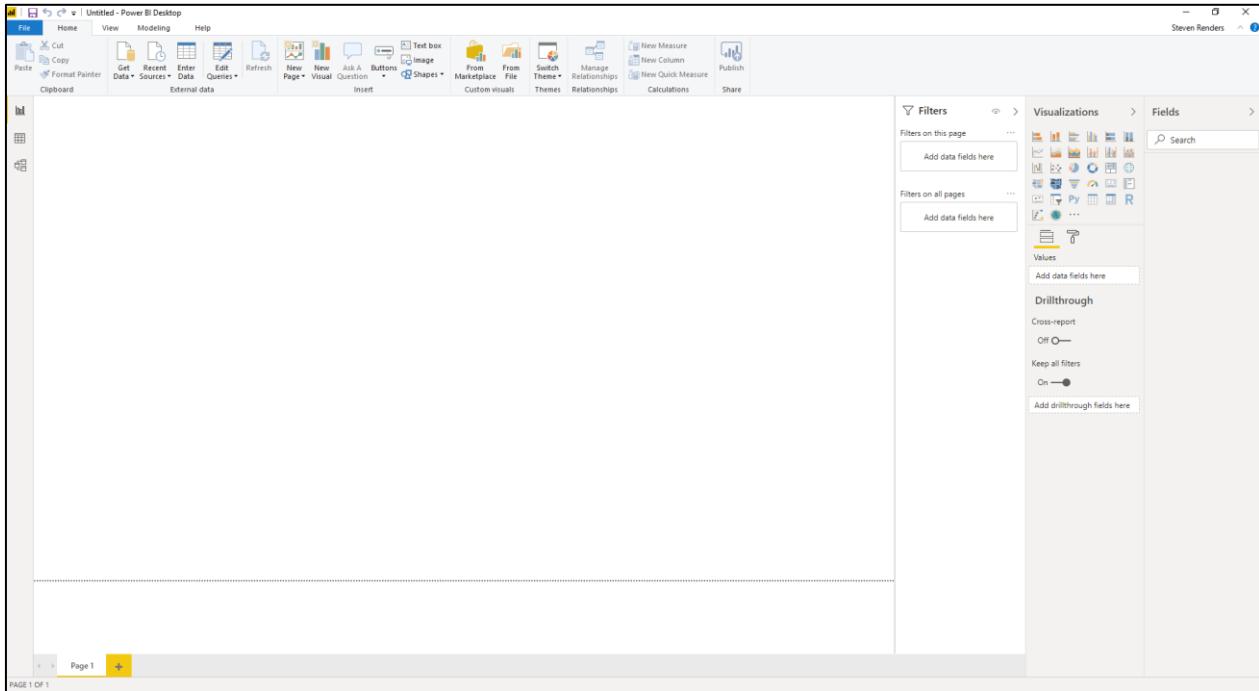
Copy them to notepad and double-check that the web service access key ends with an = sign.

Import the ODATA queries in Power BI Desktop

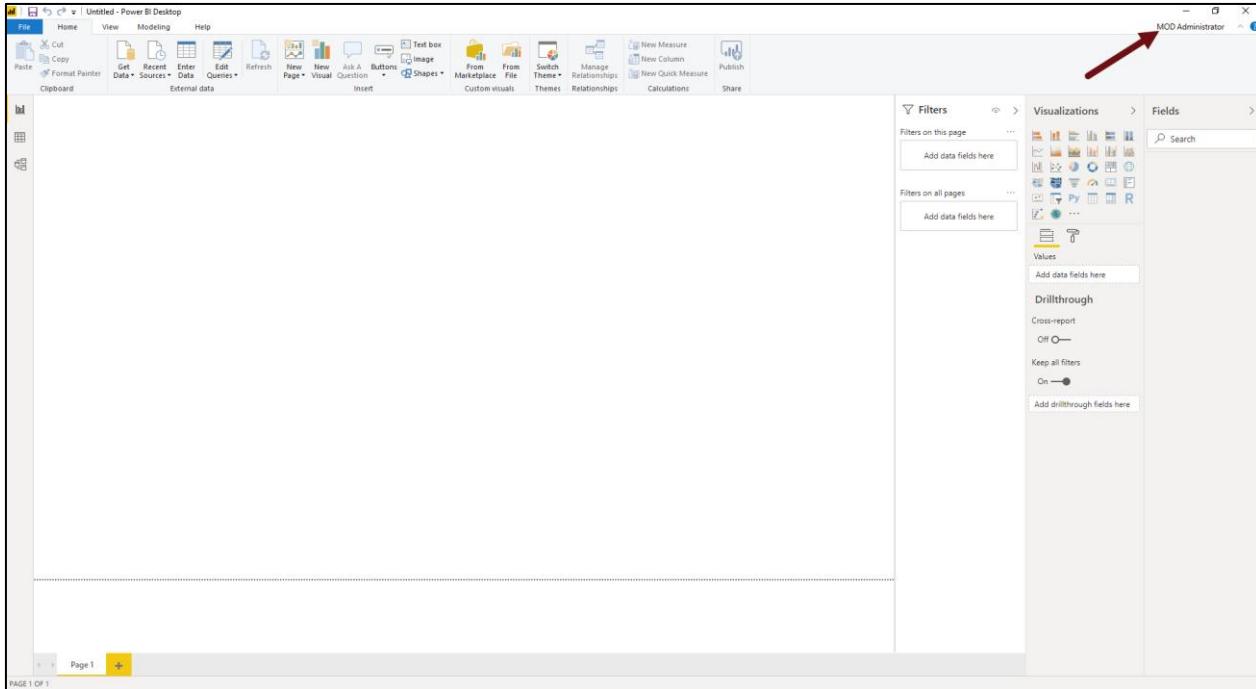
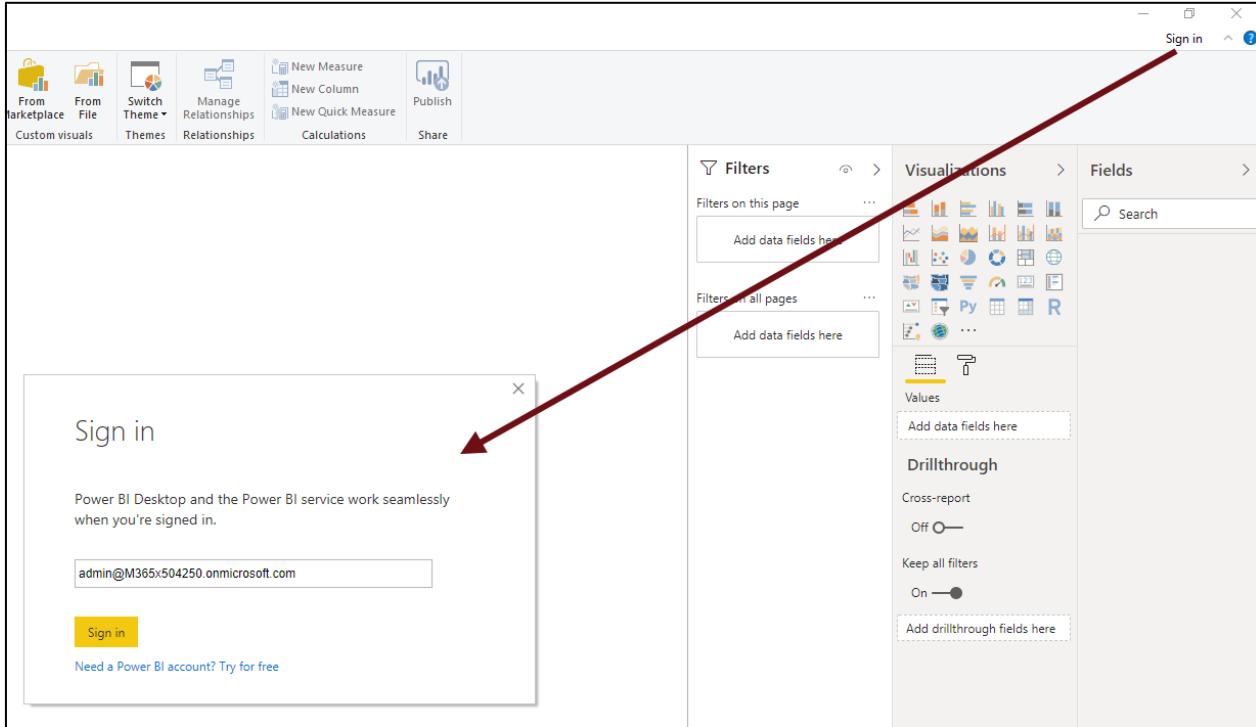
Now that our queries have been published as ODATA web services and we know the URL on how to reach them, we are ready to create a report in Power BI Desktop.

Launch Power BI Desktop:

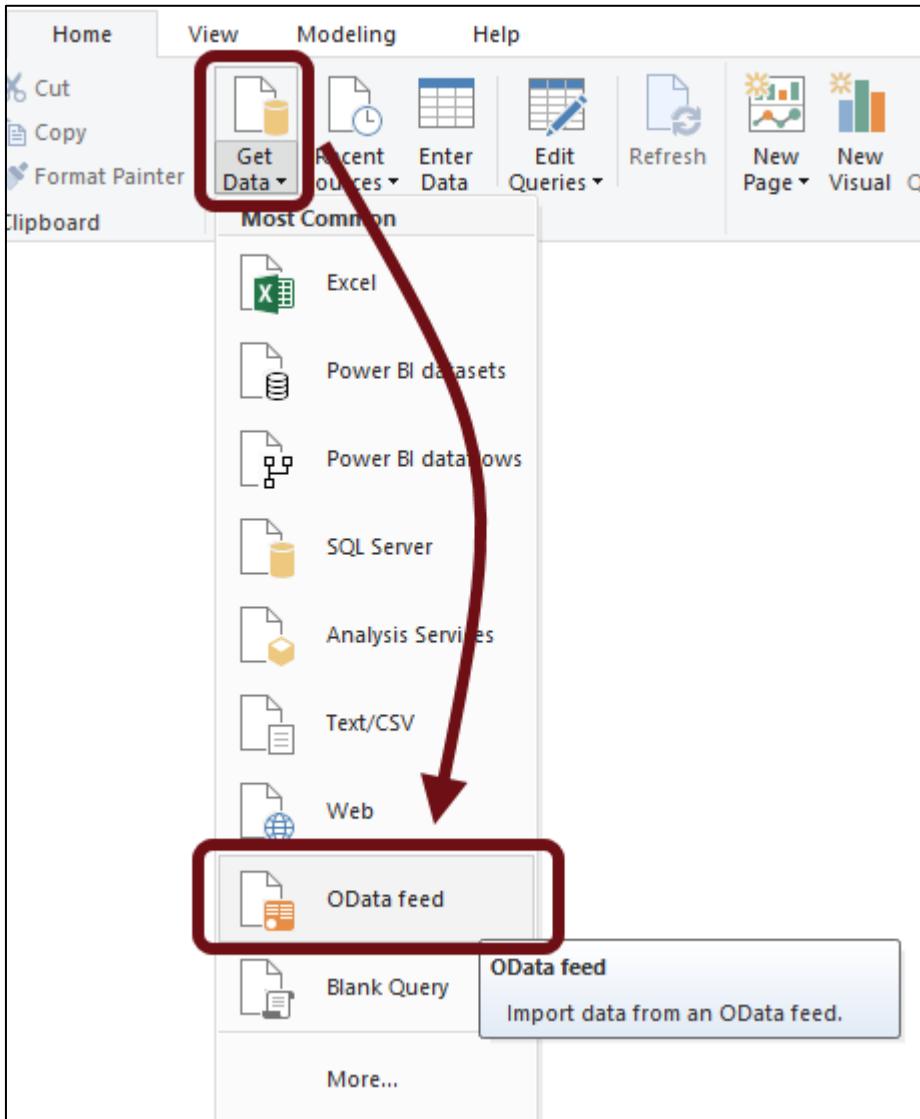




Select the **Sign In** link and enter your email and password you used when registering on the Power BI website:

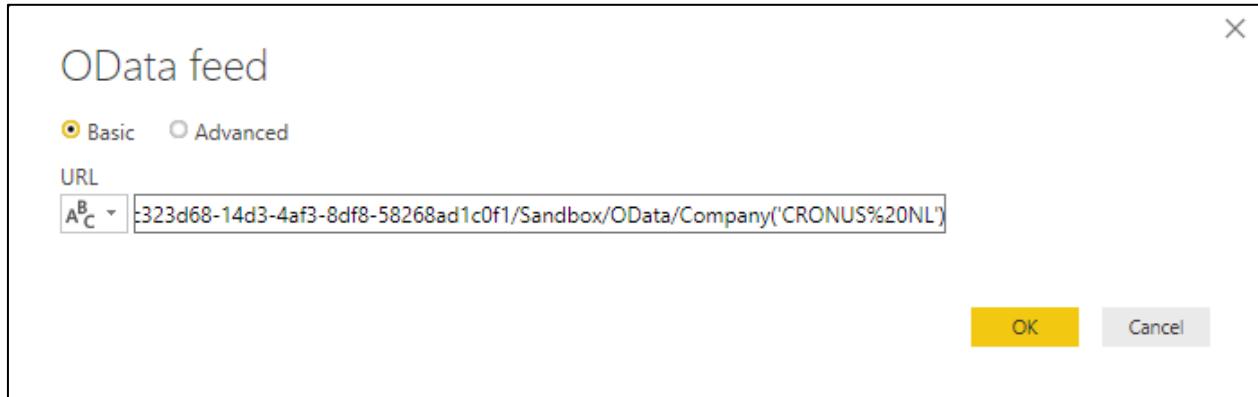


Now select **Get Data, ODATA Feed**:

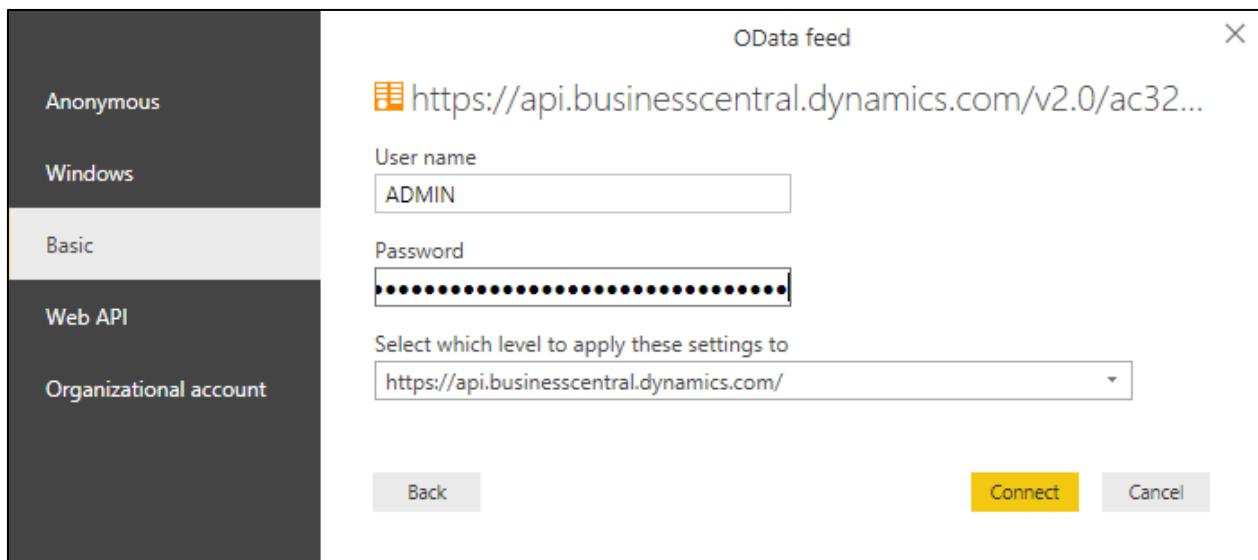


In the popup window, paste the ODATA Url:

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

The screenshot shows the Power Query Editor interface with the title bar "OrdersIntake - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Help, and Record Tools. The "File" tab is selected. The "Queries [1]" pane shows a list of tables, with "qryCustomers" highlighted. The main preview area displays the "qryCustomers" table with columns CustomerNo, CustomerName, and City, containing five rows of data. The "Query Settings" pane on the right shows the "Name" field set to "Query1".

CustomerNo	CustomerName	City
10000	Adatum Corporation	Arnhem
20000	Trey Research	Zaandam
30000	School of Fine Art	
40000	Alpine Ski House	

Select Table, right next to the qryCustomers:

The screenshot shows the Power Query Editor interface with the title bar "OrdersIntake - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Help, and Record Tools. The "File" tab is selected. The "Queries [1]" pane shows a list of tables, with "qryCustomers" highlighted. The main preview area displays the "qryCustomers" table with columns CustomerNo, CustomerName, and City, containing five rows of data. The "Query Settings" pane on the right shows the "Name" field set to "Query1". The "Transform" ribbon tab is active, showing various data transformation tools like Close & Apply, New Source, Refresh, Manage, Choose Columns, Keep Rows, Sort, and Data Type. The "Applied Steps" pane shows a step named "Navigation".

A ^B CustomerNo	A ^B CustomerName	A ^B City
1 10000	Adatum Corporation	Arnhem
2 20000	Trey Research	Zaandam
3 30000	School of Fine Art	
4 40000	Alpine Ski House	
5 50000	Relecloud	Amsterdam

Then rename the query to **Customers**:

The screenshot shows the Power Query Editor interface. The title bar says "OrdersIntake - Power Query Editor". The ribbon tabs include File, Home, Transform, Add Column, View, and Help. The Home tab is selected. The toolbar includes Close & Apply, New Source, Refresh, Properties, and others. The main area shows a table with three columns: CustomerNo, CustomerName, and City. The table has 5 rows of data. The "Properties" pane on the right shows the query is named "Customers". The "Applied Steps" pane shows the "Source" step.

Then select **Close & Apply**:

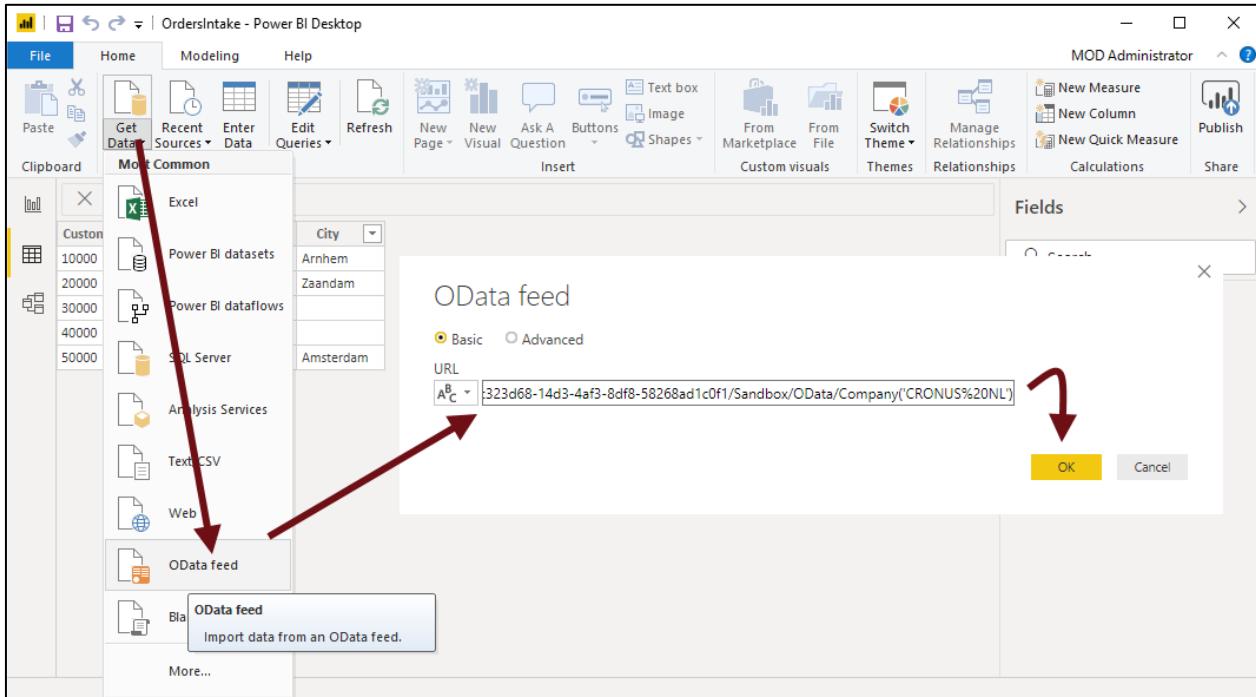
The screenshot shows the Power Query Editor interface with a red arrow pointing to the "Close & Apply" button in the ribbon. A tooltip above the button reads "Close the Query Editor window and apply any pending changes." The main area shows a table with columns Company and CustomerNo, and 2 rows of data. The "Properties" pane on the right shows the query is named "Customers".



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

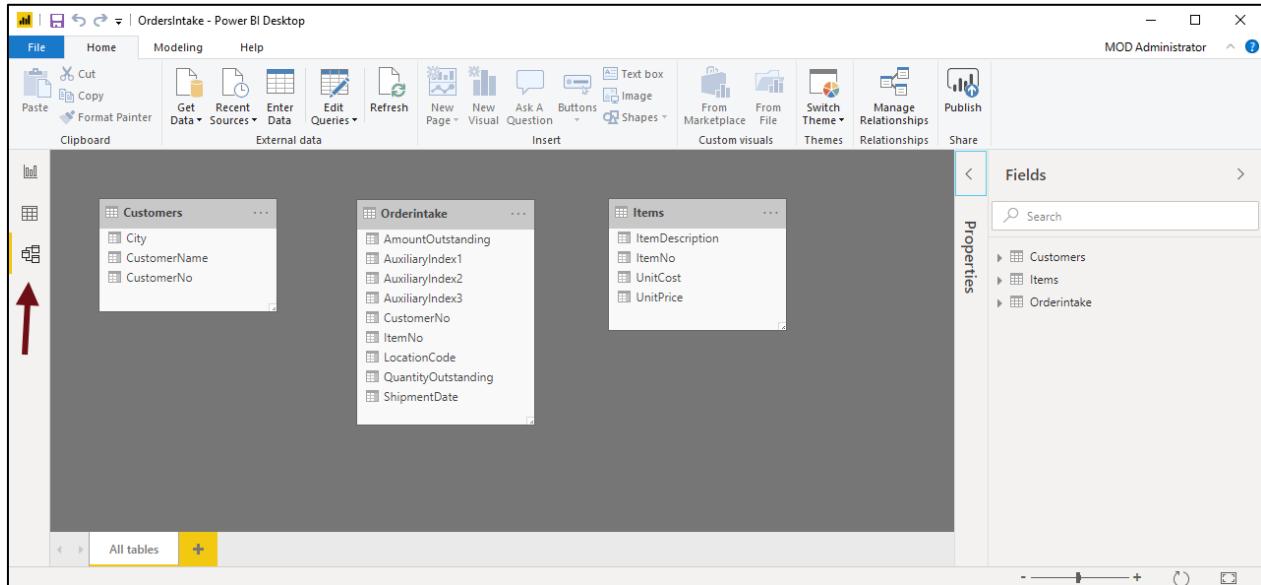
CustomerNo	CustomerName	City
10000	Adatum Corporation	Arnhem
20000	Trey Research	Zaandam
30000	School of Fine Art	
40000	Alpine Ski House	
50000	Relecloud	Amsterdam

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



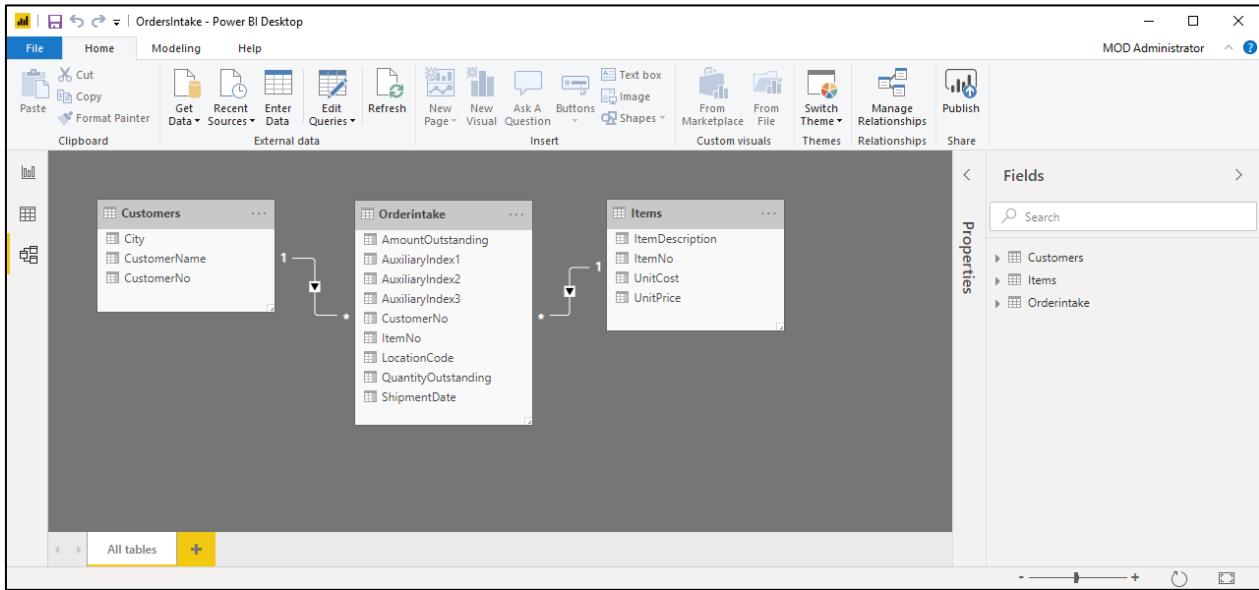
The screenshot shows the Power BI Query Editor. The 'Queries' pane displays a query named 'Query1' which connects to an OData feed at `https://api.businesscentral.dynamics.com/v2.0/`. The results show various tables such as JobLedgerEntries, Power_BI_Cust_Item_Ledg_Ent, and qryOrderIntake. A red box highlights the 'qryOrderIntake' table. The 'Query Settings' pane shows the 'Source' step under 'APPLIED STEPS'.

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



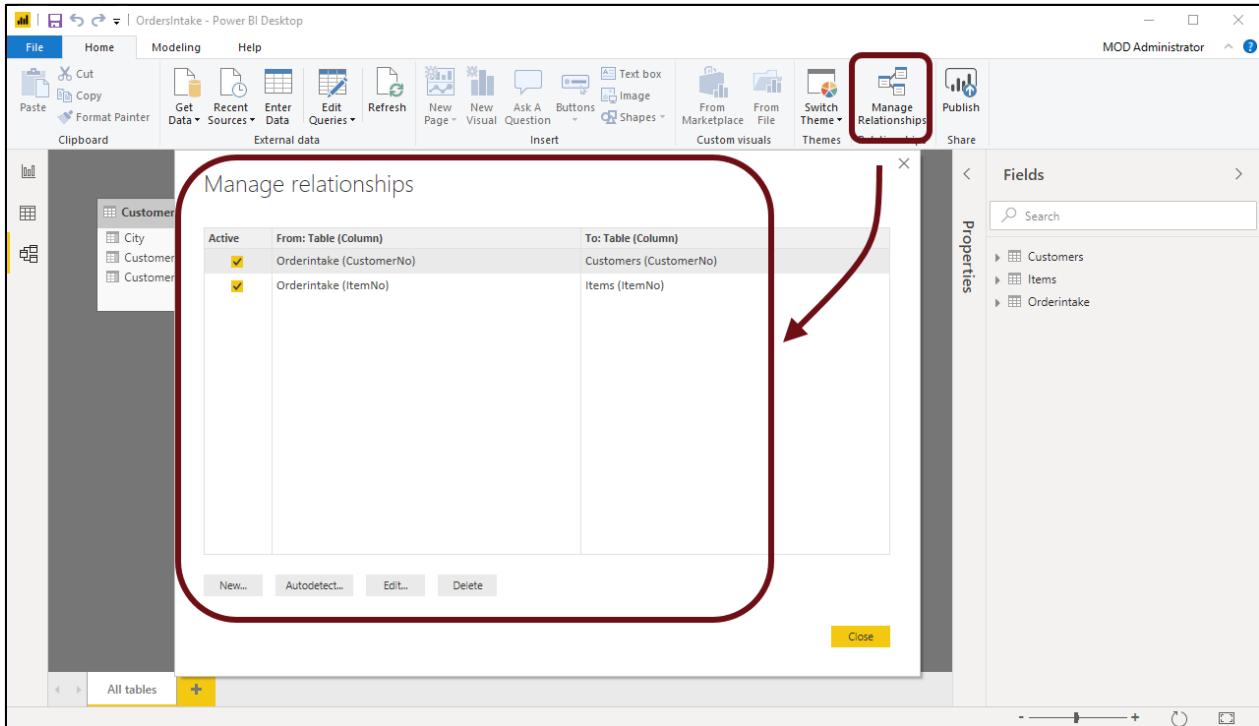
Create the Data Model and relations

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:



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**:

The screenshot shows the Power BI Desktop interface with the 'Modeling' tab selected in the ribbon. A red box highlights the 'New Measure' button. Another red box highlights the formula bar where the measure '1 No Of Customers = DISTINCTCOUNT(Orderintake[CustomerNo])' is entered. A callout bubble points to the 'Fields' pane on the right, which displays the 'Orderintake' table and its columns: CustomerNo, LocationCode, ItemNo, ShipmentDate, QuantityOutstanding, AmountOutstanding, AuxiliaryIndex1, AuxiliaryIndex2, and AuxiliaryIndex3. The 'CustomerNo' column is currently selected.

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])**

CustomerNo	LocationCode	ItemNo	ShipmentDate	QuantityOutstanding	AmountOutstanding	AuxiliaryIndex1	AuxiliaryIndex2	AuxiliaryIndex3
20000		1936-S	woensdag 25 september 2019	10	2343,77	Quote	1001	1000
40000		2000-S	woensdag 25 september 2019	5	955	Quote	1002	1000
10000		1996-S	dinsdag 2 april 2019	12	20390,44	Order	101001	1000
10000		1968-S	woensdag 1 mei 2019	10	2311,1	Order	101002	1000
10000		1928-S	woensdag 1 mei 2019	7	467,54	Order	101002	2000
30000		1920-S	maandag 22 april 2019	8	5208,8	Order	101003	1000
40000		2000-S	maandag 13 mei 2019	3	573	Order	101004	1000
10000		1968-S	woensdag 25 september 2019	5	1155,55	Invoice	102199	1000
10000		1996-S	woensdag 25 september 2019	7	11894,42	Invoice	102199	2000
				0	0	Invoice	102199	3000
10000		2000-S	woensdag 25 september 2019	2	462,22	Invoice	102200	1000
10000		1996-S	woensdag 25 september 2019	5	8496,02	Invoice	102200	2000
20000		1896-S	woensdag 25 september 2019	1	1217,02	Invoice	102201	1000
30000		1920-S	woensdag 25 september 2019	4	2604,4	Invoice	102202	1000
30000		1920-S	woensdag 25 september 2019	10	6511	Invoice	102203	1000
40000		1928-S	woensdag 25 september 2019	5	276	Invoice	102204	1000
50000		1920-S	woensdag 25 september 2019	4	8151,32	Invoice	102205	1000
50000		1936-S	woensdag 25 september 2019	23	5390,68	Invoice	102205	2000

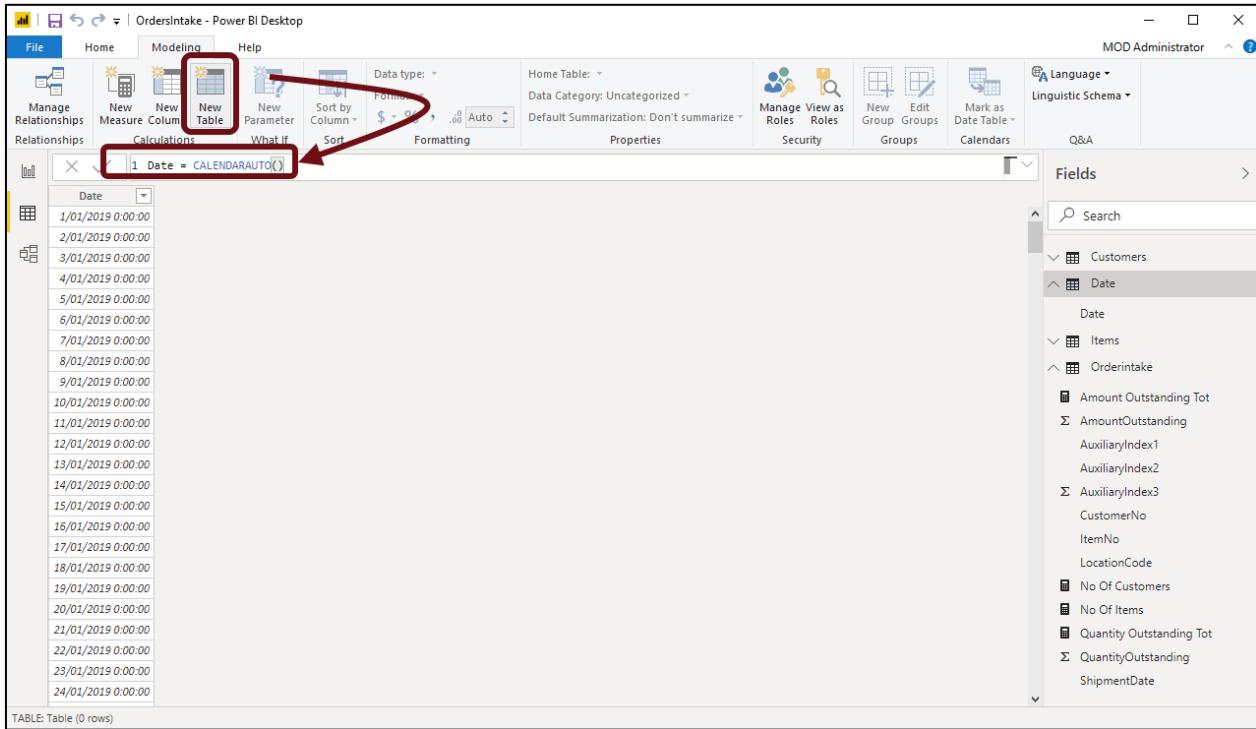
TABLE: Orderintake (18 rows)

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

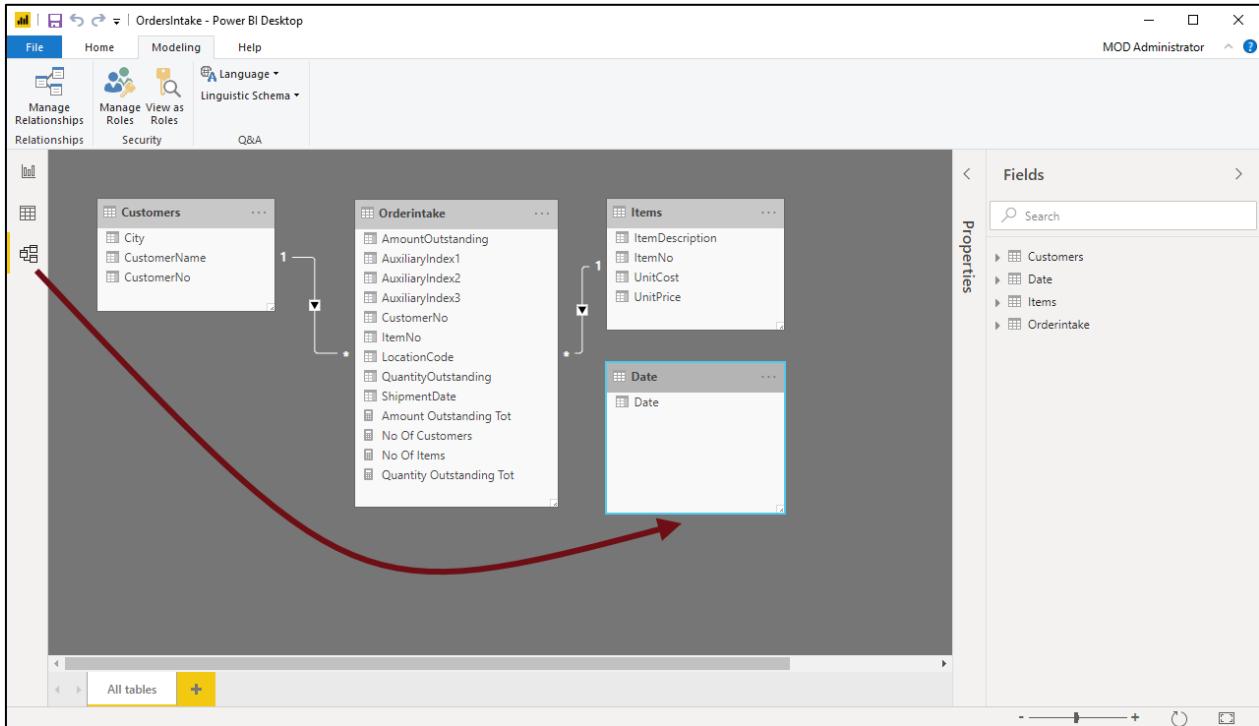
The reason is that we need a table with a **continuous date range**.

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

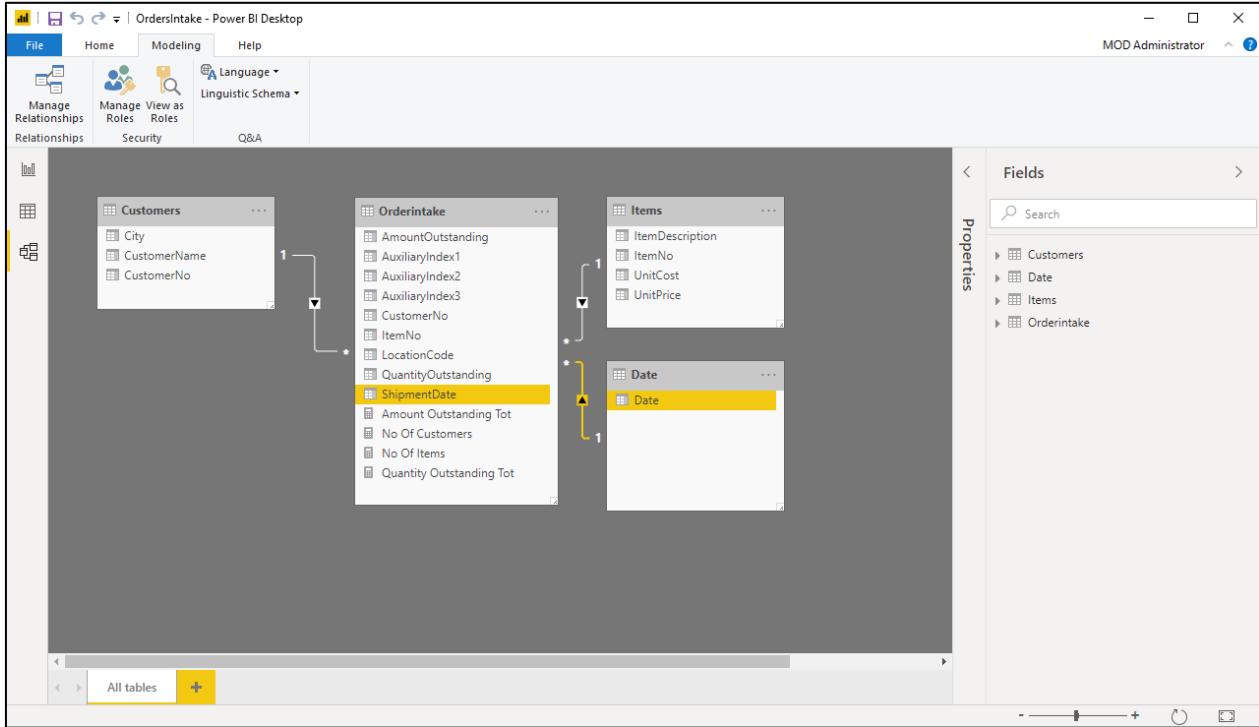
- **Date = CALENDARAUTO()**



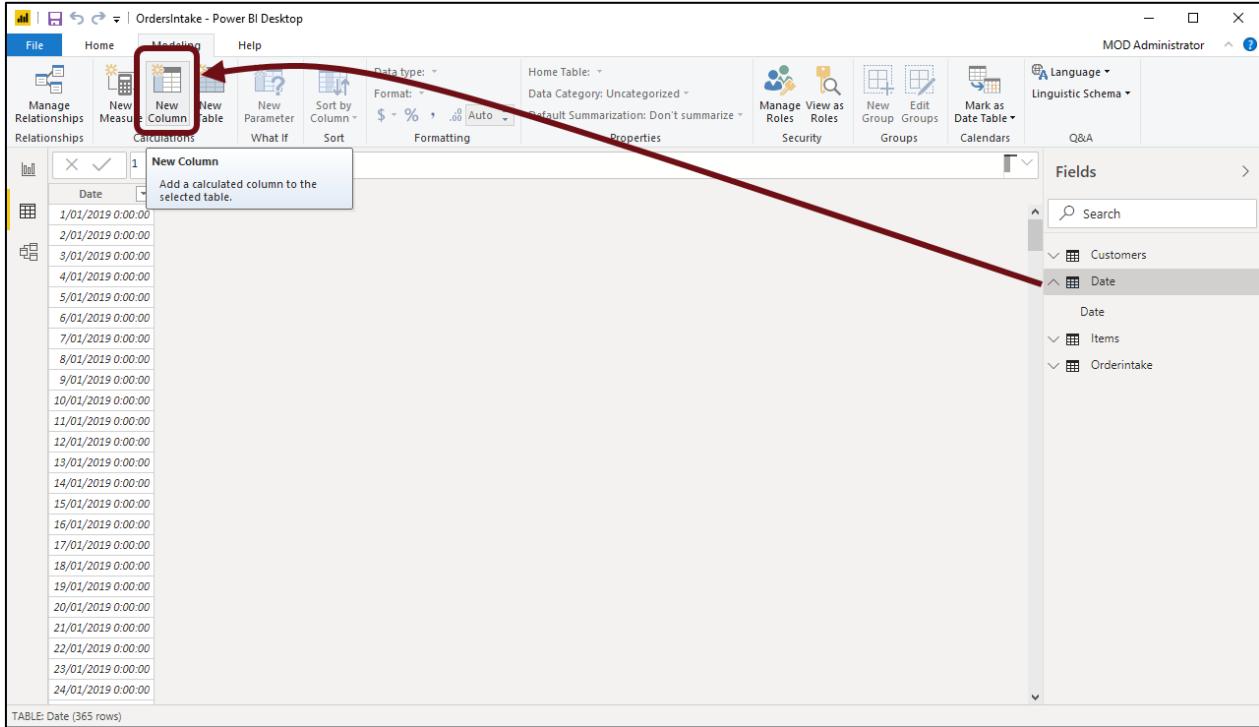
You can also see the table in the **Relationships** window:



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])

The screenshot shows the Power BI Desktop interface with the 'Ordersintake' dataset open. In the Data view, there is a table with four columns: Date, Year, MonthName, and MonthNo. The 'Year' column contains the value '2019' for all rows. The entire table has a red box around it. In the Fields pane on the right, the 'Year' column is listed under the 'Date' category, also with a red box around it. The Fields pane also lists 'MonthName' and 'MonthNo' under the 'Date' category.

Date	Year	MonthName	MonthNo
1/01/2019 0:00:00	2019	Jan	1
2/01/2019 0:00:00	2019	Jan	1
3/01/2019 0:00:00	2019	Jan	1
4/01/2019 0:00:00	2019	Jan	1
5/01/2019 0:00:00	2019	Jan	1
6/01/2019 0:00:00	2019	Jan	1
7/01/2019 0:00:00	2019	Jan	1
8/01/2019 0:00:00	2019	Jan	1
9/01/2019 0:00:00	2019	Jan	1
10/01/2019 0:00:00	2019	Jan	1
11/01/2019 0:00:00	2019	Jan	1
12/01/2019 0:00:00	2019	Jan	1
13/01/2019 0:00:00	2019	Jan	1
14/01/2019 0:00:00	2019	Jan	1
15/01/2019 0:00:00	2019	Jan	1
16/01/2019 0:00:00	2019	Jan	1
17/01/2019 0:00:00	2019	Jan	1
18/01/2019 0:00:00	2019	Jan	1
19/01/2019 0:00:00	2019	Jan	1
20/01/2019 0:00:00	2019	Jan	1
21/01/2019 0:00:00	2019	Jan	1
22/01/2019 0:00:00	2019	Jan	1
23/01/2019 0:00:00	2019	Jan	1
24/01/2019 0:00:00	2019	Jan	1

TABLE: Date (365 rows) COLUMN: Year (1 distinct values)

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

The screenshot shows the Power BI Desktop interface with the 'Modeling' tab selected. In the center, there is a table preview for the 'Date' table. The columns shown are 'Date', 'Year', and 'MonthName'. The 'MonthName' column contains values like 'Jan', 'Feb', etc. The 'MonthNo' column contains values like '1', '2', etc. A red arrow points from the 'Sort by Column' dropdown in the top ribbon to the 'MonthName' column in the preview. Another red arrow points from the 'MonthName' column in the preview to the 'MonthNo' column.

Date	Year	MonthName	MonthNo
1/01/2019 0:00:00	2019	Jan	1
2/01/2019 0:00:00	2019	Jan	1
3/01/2019 0:00:00	2019	Jan	1
4/01/2019 0:00:00	2019	Jan	1
5/01/2019 0:00:00	2019	Jan	1
6/01/2019 0:00:00	2019	Jan	1
7/01/2019 0:00:00	2019	Jan	1
8/01/2019 0:00:00	2019	Jan	1
9/01/2019 0:00:00	2019	Jan	1
10/01/2019 0:00:00	2019	Jan	1
11/01/2019 0:00:00	2019	Jan	1
12/01/2019 0:00:00	2019	Jan	1
13/01/2019 0:00:00	2019	Jan	1

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:

The screenshot shows the Power BI Desktop interface with the title bar "OrdersIntake - Power BI Desktop". The ribbon menu includes File, Home, View, Modeling, Help, Format, Data / Drill, Visual tools, and a green tab labeled "Visual tools". The main area displays a matrix visual with several rows and columns. To the right of the canvas is the "Fields" pane, which lists various data sources and their fields. The "Customers" section includes City, CustomerName, and CustomerNo. The "Date" section includes Date, MonthName, MonthNo, and Year. The "Items" section includes OrderIntake. The "Values" section includes Amount Outst..., Σ AmountOutst..., AuxiliaryIndex1, AuxiliaryIndex2, and Σ AuxiliaryIndex3. The "Drillthrough" section is set to "Off". The status bar at the bottom shows "PAGE 1 OF 1".

Drag the following fields into Rows, Columns and Values:

The screenshot shows the Power BI Desktop interface with the following details:

- Visual Tools Bar:** Includes File, Home, View, Modeling (selected), Help, Format, Data / Drill.
- Properties Bar:** Shows Home Table: OrdersIntake, Data Category: Uncategorized, Default Summarization: Don't summarize.
- Relationships Panel:** Manage Relationships, New Measure, New Column, New Table, New Parameter, What If.
- Sort and Formatting:** Sort by Column, Sort, Formatting.
- Security:** Manage Roles, View as Roles, New Group, Edit Groups, Mark as Date Table, Groups, Calendars, Language, Linguistic Schema.
- Fields Panel:** Displays a tree view of fields from various tables like Customers, Date, Items, and Orderintake. Three red arrows point to specific fields:
 - CustomerName** under Rows.
 - Year** and **MonthName** under Columns.
 - Quantity Outstanding Tot** under Values.
- Drillthrough:** Options for Cross-report (Off or On) and Keep all filters.
- Page Navigation:** Page 1 of 1.

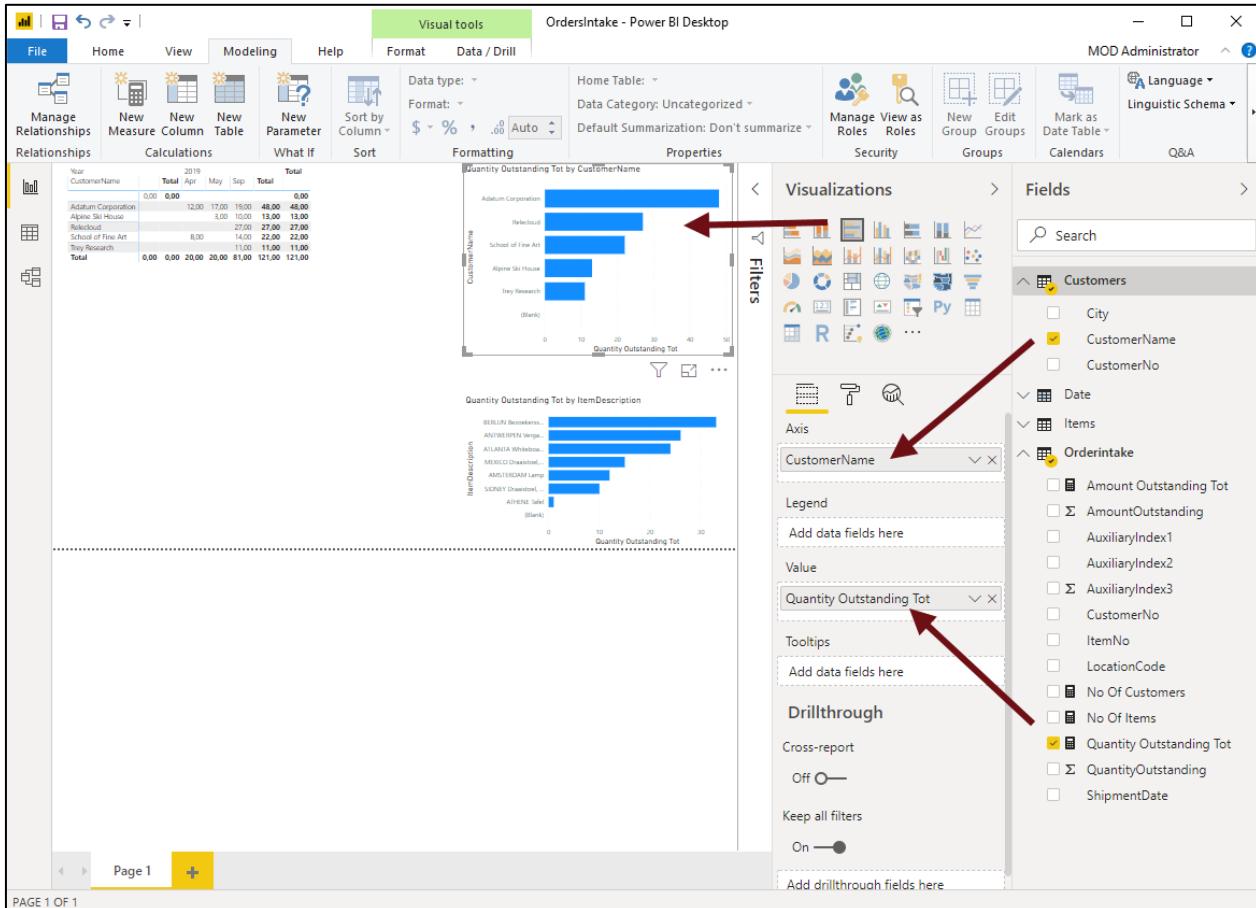
Add a Chart to the report:

The screenshot shows the Power BI Desktop interface with the following details:

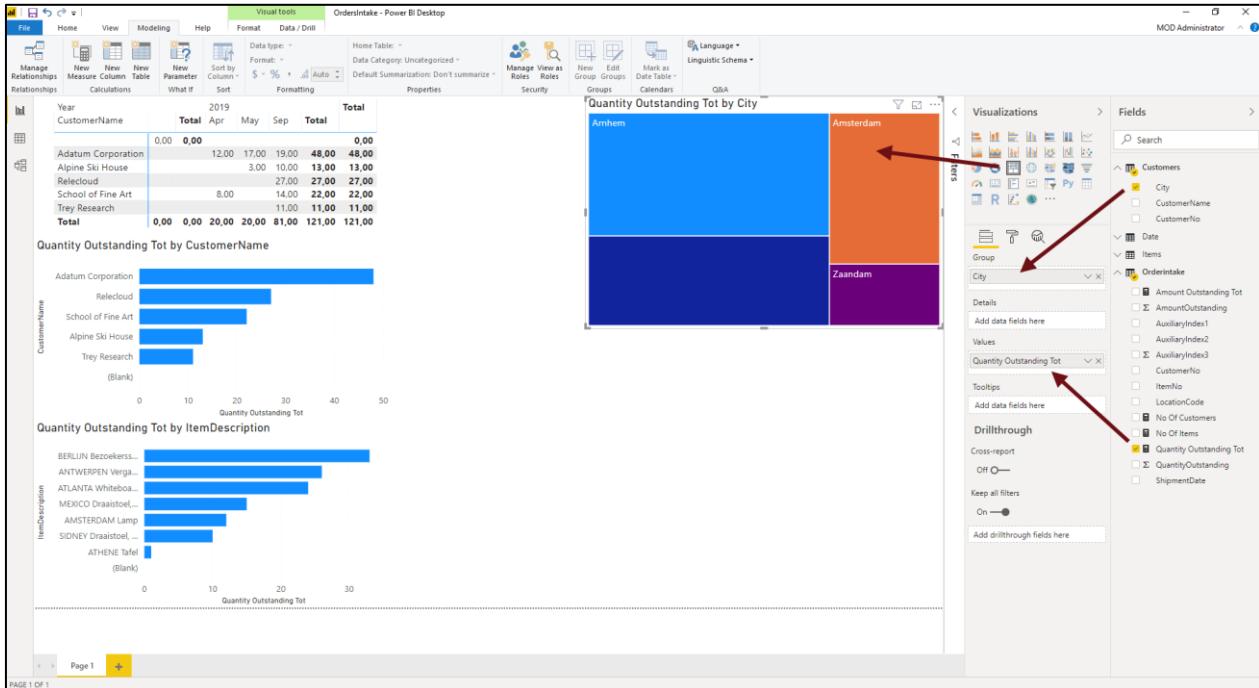
- File Bar:** File, Home, View, Modeling (selected), Help.
- Home Bar:** Manage Relationships, New Measure, New Column, New Table, New Parameter, Sort by Column, Sort, Format, Data / Drill.
- Visual Tools Bar:** Data type: \$ % .00 Auto, Home Table: Uncategorized, Data Category: Uncategorized, Default Summarization: Don't summarize.
- Properties Bar:** Properties tab selected.
- Visualizations:** A bar chart titled "Quantity Outstanding Tot by ItemDescription". The chart displays the following data (approximate values):

Item Description	Quantity Outstanding Tot
BERLIN Beeskers...	30
ANTWERPEN Verg...	25
ATLANTA Whitebow...	20
MEXICO Drawnvel...	15
AMSTERDAM Lamp...	10
SOPHIA Drawnvel...	10
ATHENE Telef...	5
(Blank)	0
- Fields pane:** Shows the hierarchy of fields. The "Items" node under "Orderintake" is expanded, with "ItemDescription" checked. Other items like "ItemNo", "UnitCost", and "UnitPrice" are also listed under "Items". The "Orderintake" node contains fields such as "Amount Outstanding Tot", "AuxiliaryIndex1", "CustomerNo", etc.
- Bottom Navigation:** Page 1, a plus sign icon, and PAGE 1 OF 1.

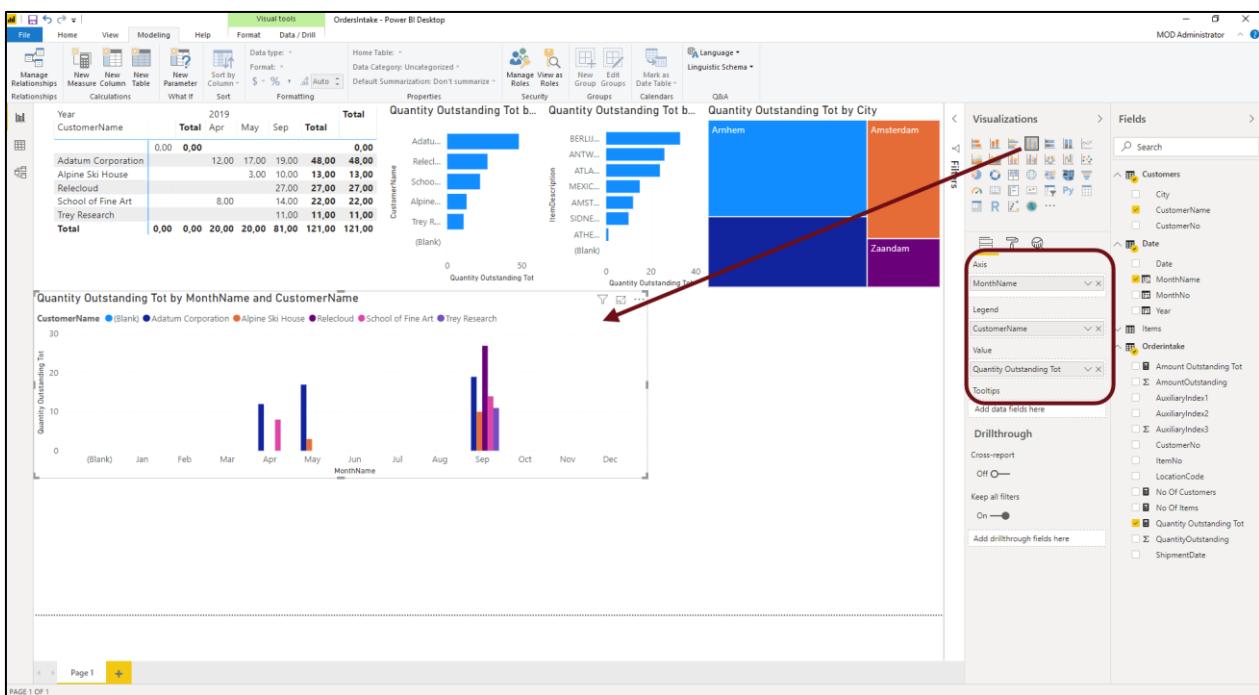
Add another Chart to the report:



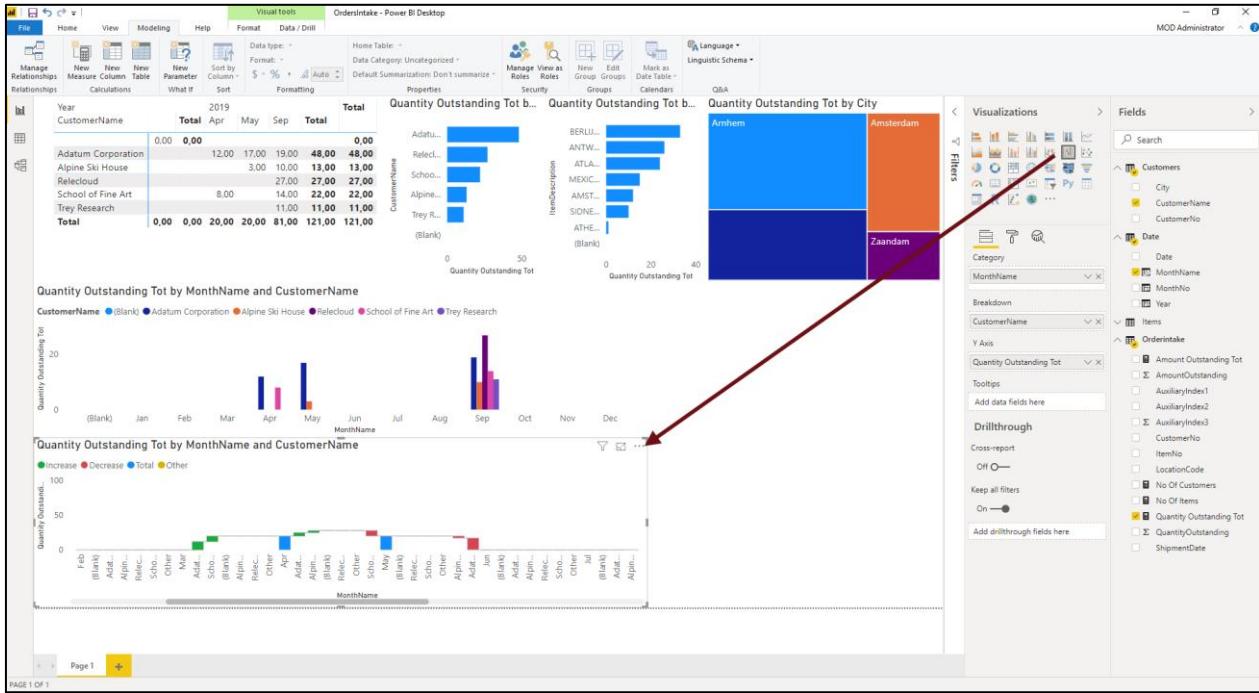
Add a Treemap to the report:



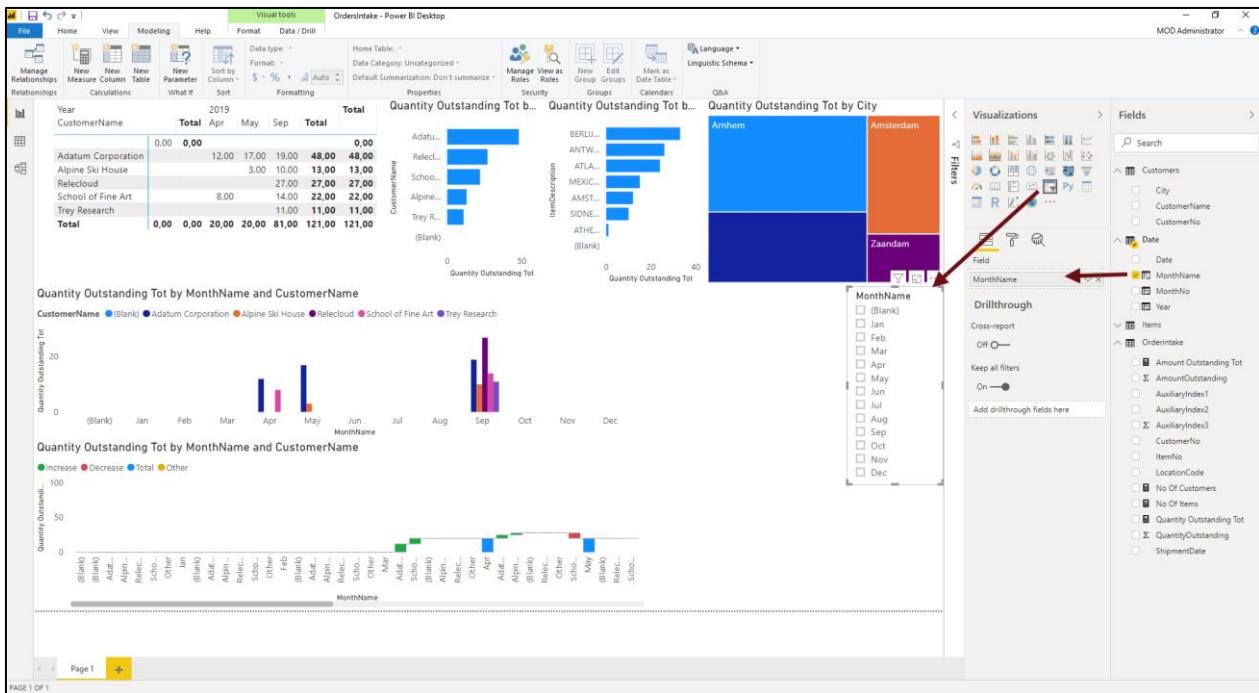
Add another Chart to the report:



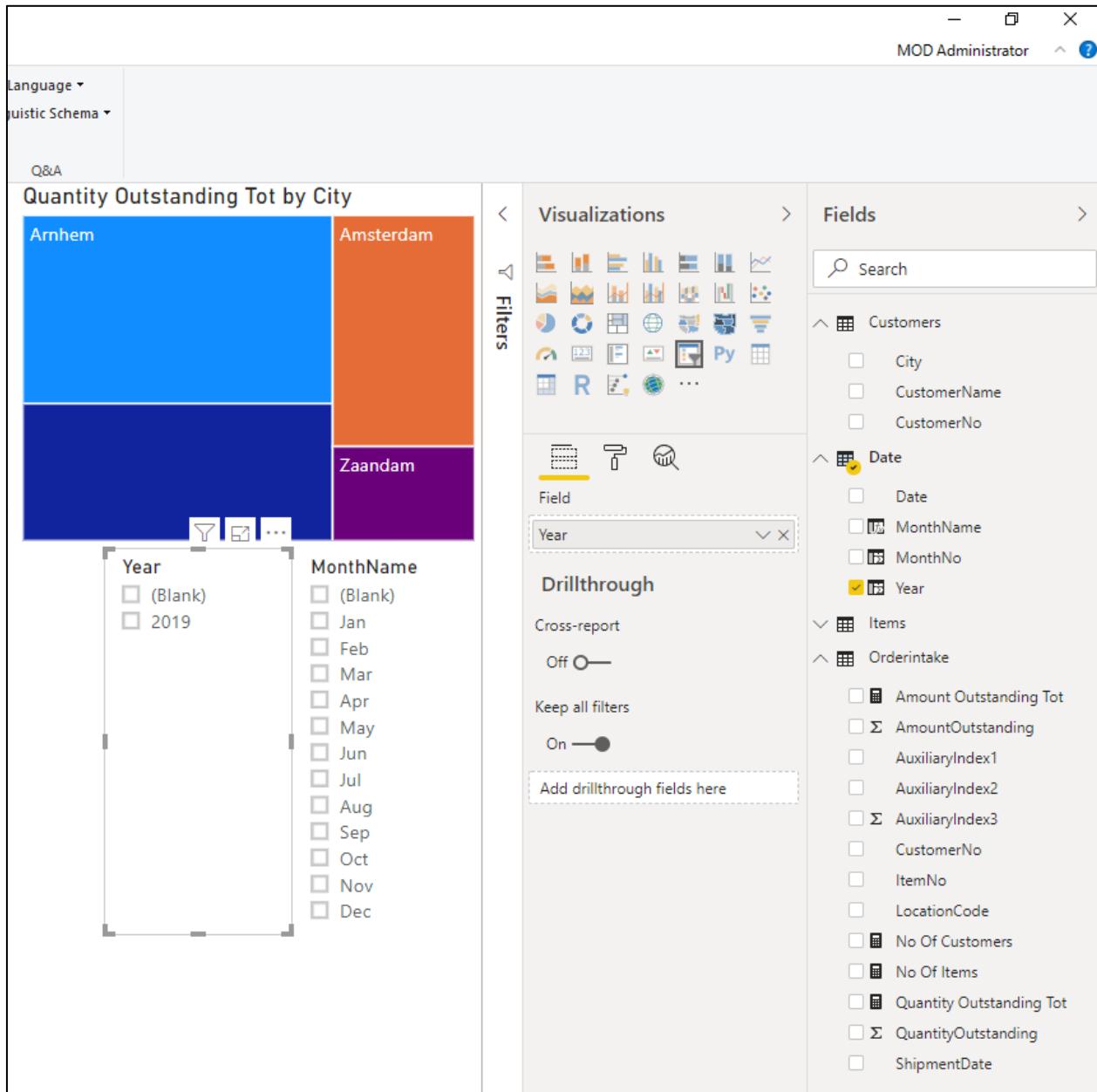
Copy/Paste the previous chart and change it into a Waterfall chart:



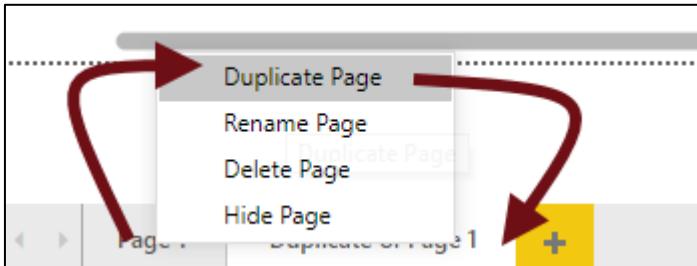
Add a Slicer:



Add another Slicer:



Duplicate the Report page:



On the new page, remove the bottom charts:

CustomerName	Total	2019	Apr	May	Sep	Total
Aldatum Corporation	0,00	0,00	12,00	17,00	19,00	48,00
Alpine Ski House			3,00	10,00	13,00	13,00
Relecloud			27,00	27,00	27,00	27,00
School of Fine Art	8,00		14,00	22,00	22,00	
Trey Research			11,00	11,00	11,00	
Total	0,00	0,00	20,00	20,00	81,00	121,00

Quantity Outstanding Tot by Item

Quantity Outstanding Tot by City

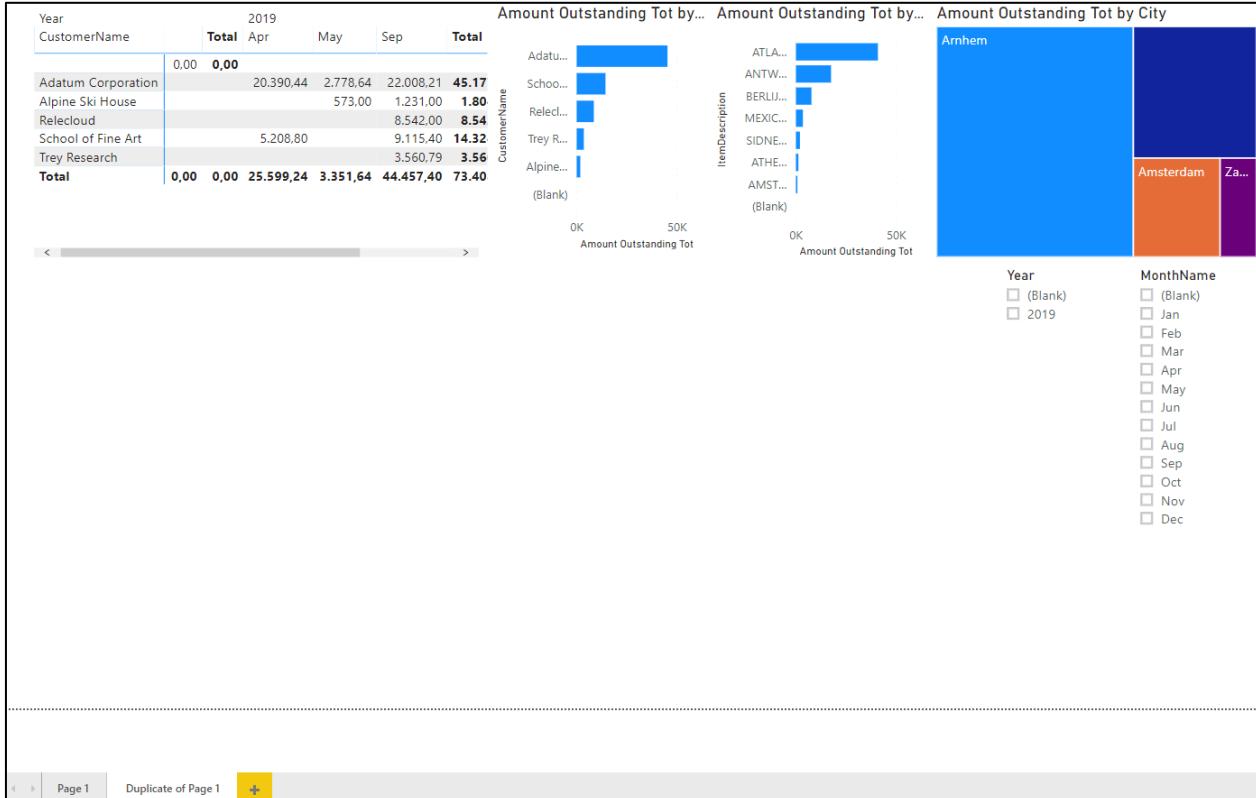
Filters

- Visualizations
- Fields
- Customers
- Date
- Values
- Add data fields here
- Drillthrough
- Cross-report
- Off
- Keep all filters
- On
- Add drillthrough fields here

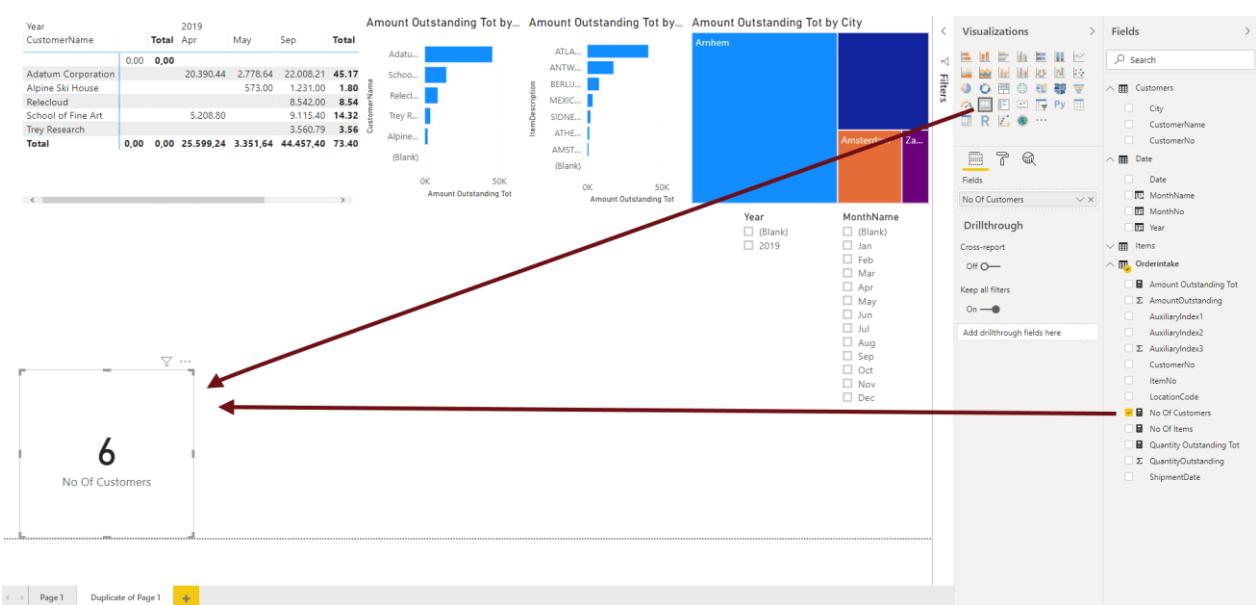
Fields

- Customers
- Date
- Values
- Add data fields here
- Drillthrough
- Cross-report
- Off
- Keep all filters
- On
- Add drillthrough fields here

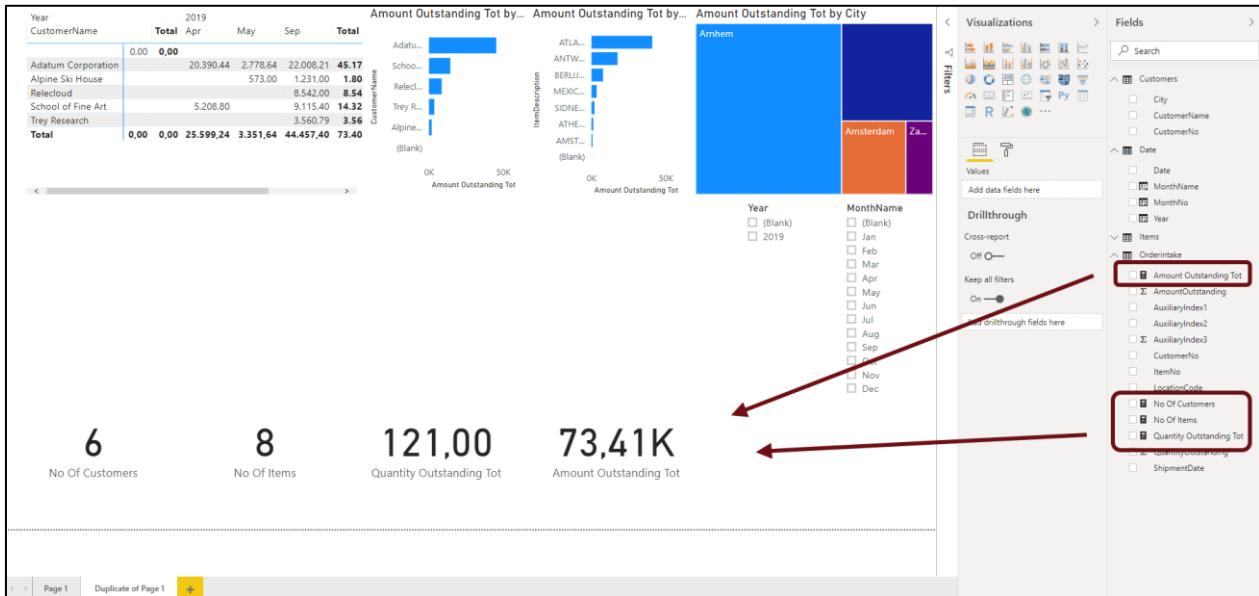
Then go into the visuals on the new page and replace the **Quantity** fields by the corresponding **Amount** fields.



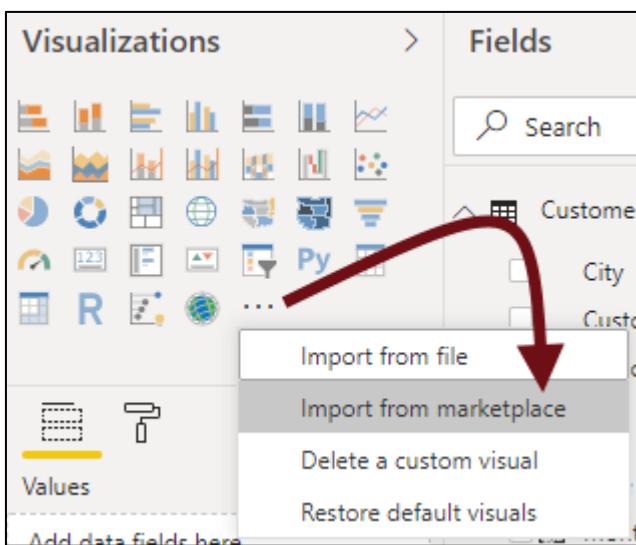
Add a Card:



Copy/Paste the Card and add the following cards:



Add a custom visual:



Look for the Aquarium visual and Add it:

Power BI Visuals

MARKETPLACE | MY ORGANIZATION

Add-ins may access personal and document information. By using an add-in, you agree to its Permissions, License Terms and Privacy Policy.

aquarium 

Category: All

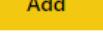
Suggested for you

Enlighten Aquarium  

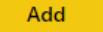
Make dashboards fun with this award-winning Aquarium visualization

Enlighten Data Story  

Use beautifully designed text to tell your story.

Enlighten World Flag Slicer  

Create a visually compelling country slicer to filter your report.

Enlighten Stack Shuffle  

Visually communicate data items in a beautiful, dynamic stack.

Category: All

Advanced Analytics

Data Visualizations

Editor's Picks

Filters

Gauges

Infographics

KPIs

Maps

Power BI Certified

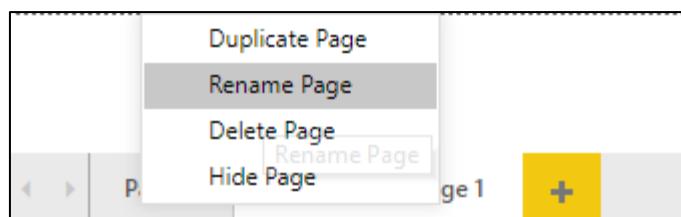
Time

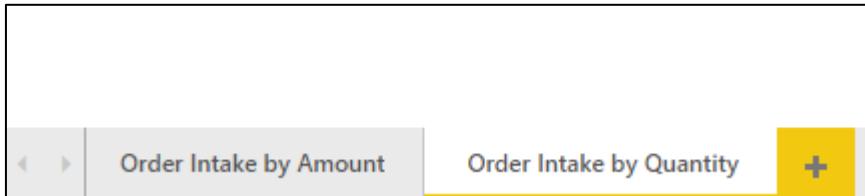
Add it to the report as follows:

The screenshot shows a Power BI report interface. On the left, there is a bar chart titled "Amount Outstanding Tot by ItemDescription" and a treemap chart titled "Amount Outstanding Tot by City". In the center, a message box says "Import custom visual" and "The visual was successfully imported into this report." An "OK" button is at the bottom. On the right, the "Visualizations" pane shows various chart types, and the "Fields" pane lists fields from "Customers", "Date", and "Items" tables. A red arrow points from the "Visualizations" pane towards the imported visual.

The screenshot shows a more complex Power BI dashboard. It includes a table view of customer data, a bar chart for "Amount Outstanding Tot by ItemDescription", a treemap chart for "Amount Outstanding Tot by City", a scatter plot titled "Amount Outstanding Tot and Quantity Outstanding Tot by ItemDescription" featuring cartoon fish, and four KPI cards at the bottom: "No Of Customers" (6), "No Of Items" (8), "Quantity Outstanding Tot" (121.00), and "Amount Outstanding Tot" (73,41K). The "Fields" pane on the right is expanded, showing detailed filter settings for the imported visual, including "Fish", "ItemDescription", "Fish Size", "Amount Outstanding Tot", and "Quantity Outstanding Tot". A red box highlights these specific filter settings. A red arrow points from the "Fields" pane towards the imported visual.

Rename the pages:



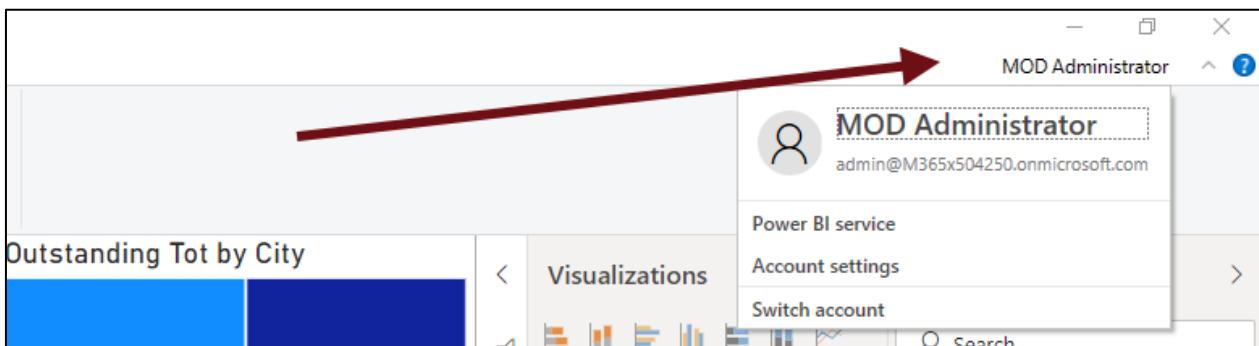


The report is now ready to publish. Please **Save** it now:



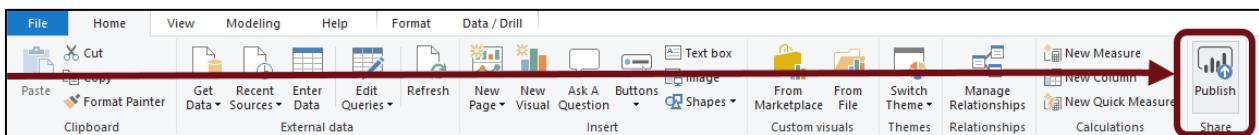
Publish your pbix file to the Power BI Service

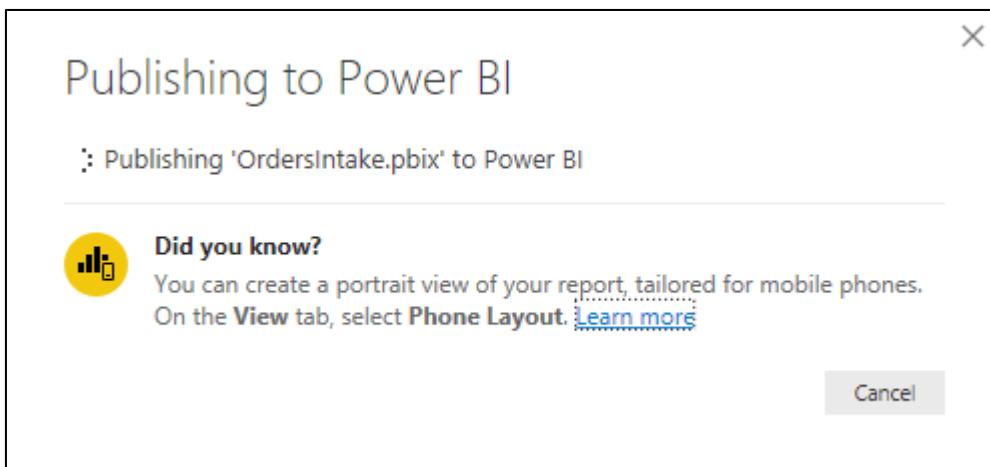
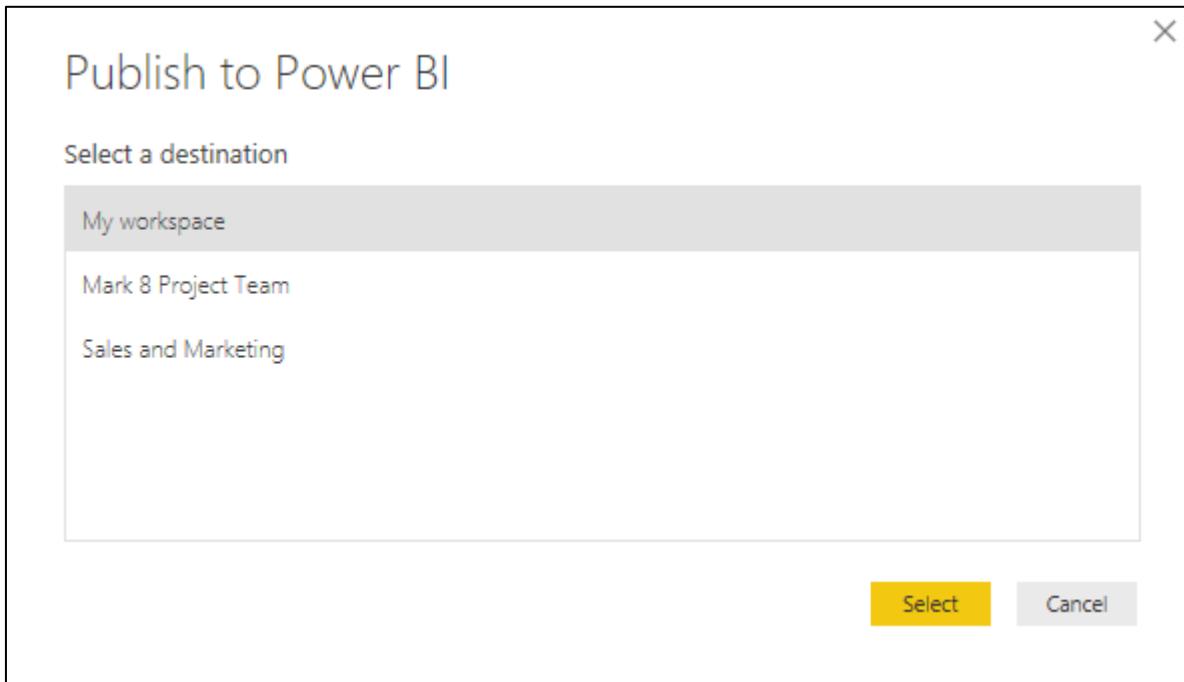
To publish our report, you need to sign in:

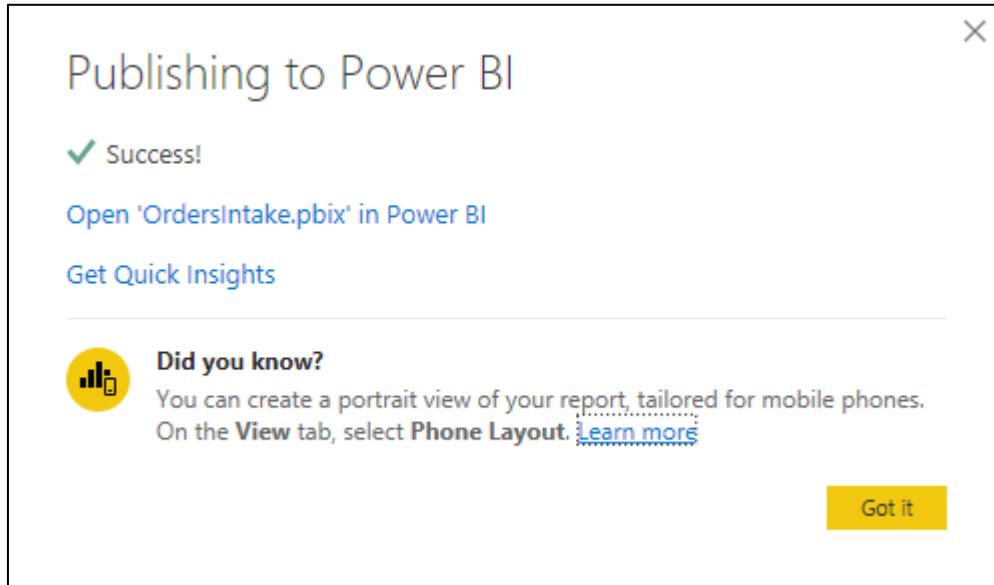


Use your email address that you used to create your Power BI account.

Then **Publish** the report:



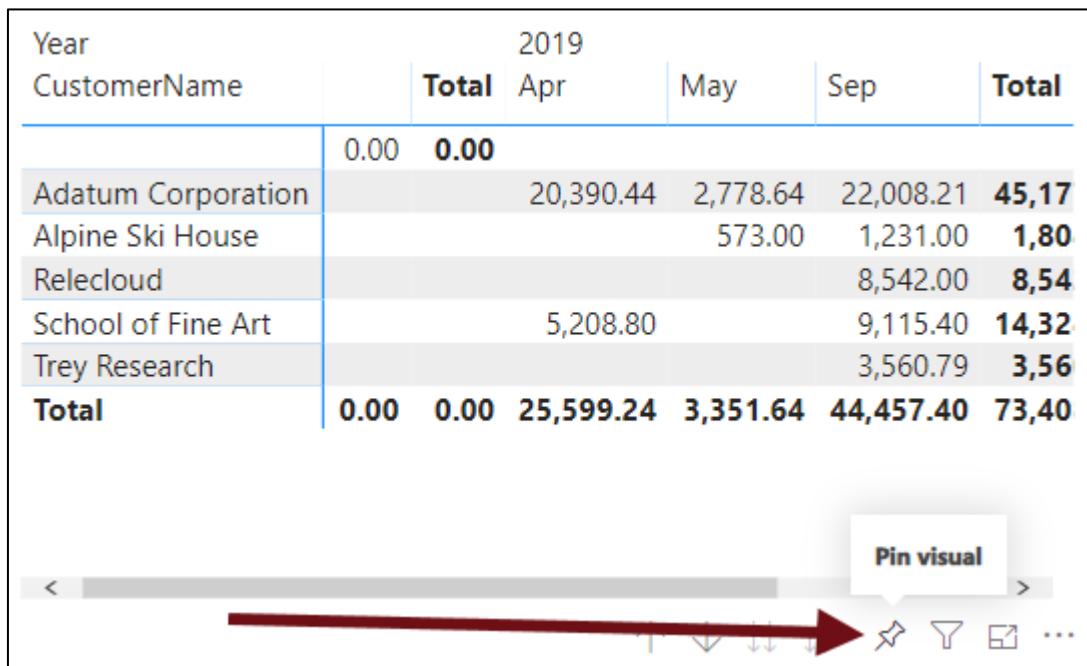




Go to powerbi.microsoft.com or select the link in the popup:

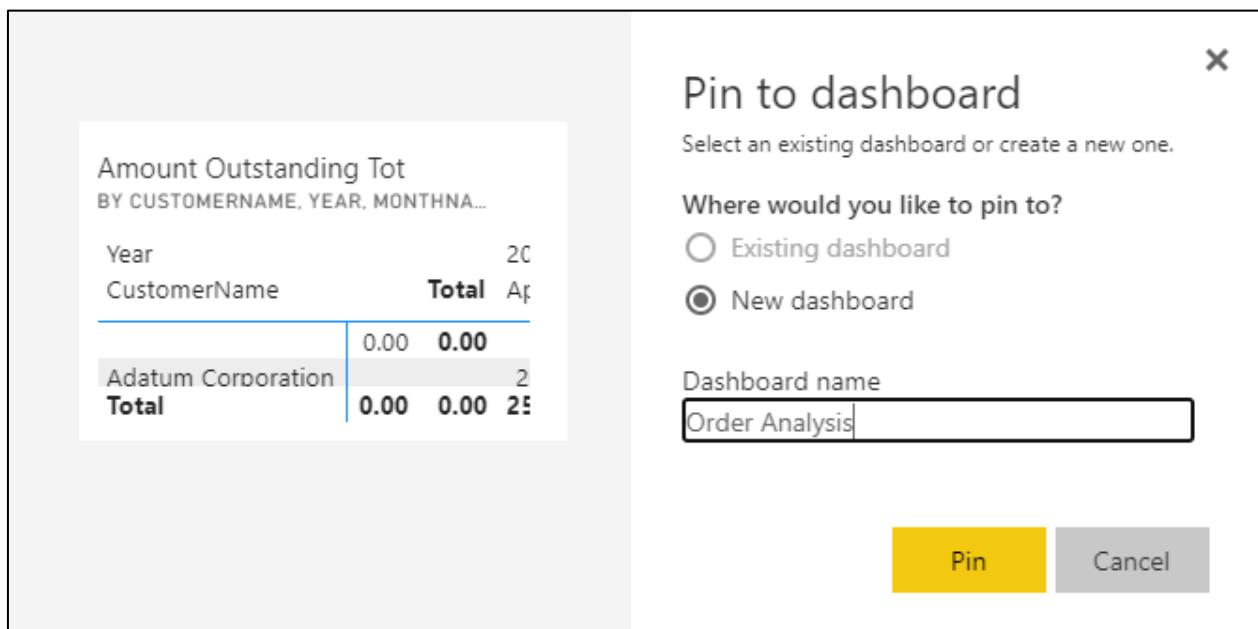
Create a Dashboard

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



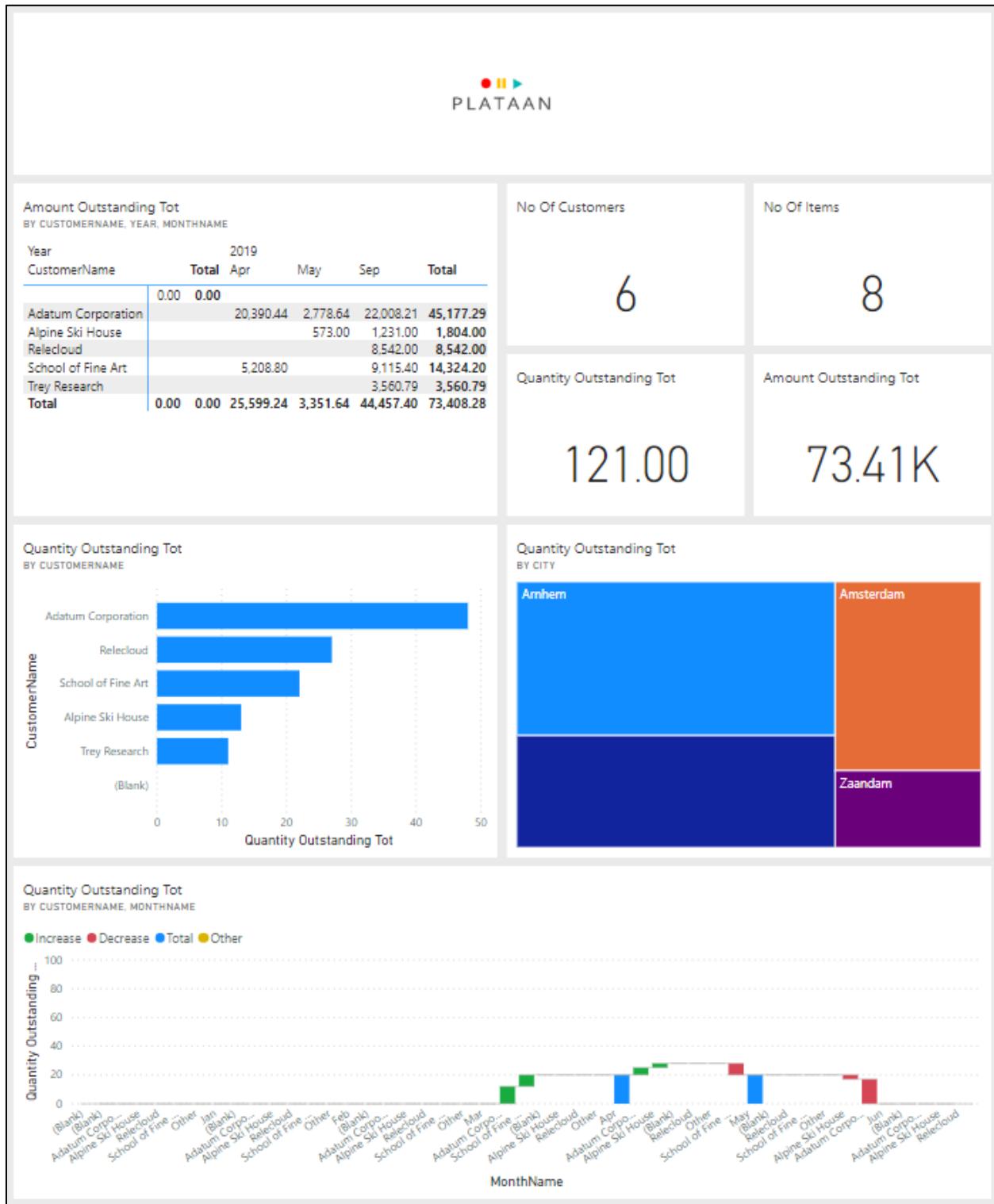
A screenshot of a Power BI report interface. At the top, there's a table titled "Year" with columns for "CustomerName" and "Total". The table shows data for various customers like Adatum Corporation, Alpine Ski House, Relecloud, School of Fine Art, and Trey Research, along with their total amounts. A large red arrow points from the bottom of this table towards a "Pin visual" button. Below the table, there are navigation arrows and other icons.

CustomerName	Total	Apr	May	Sep	Total	
Adatum Corporation	0.00	0.00	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



A screenshot of a "Pin to dashboard" dialog box. On the left, there's a preview of a table visual titled "Amount Outstanding Tot BY CUSTOMERNAME, YEAR, MONTHNA...". The table shows the same data as the one in the previous screenshot. To the right, there's a form to pin the visual to a dashboard. It has a title "Pin to dashboard", a sub-instruction "Select an existing dashboard or create a new one.", and a question "Where would you like to pin to?". There are two radio buttons: "Existing dashboard" (unchecked) and "New dashboard" (checked). Below that is a "Dashboard name" input field containing "Order Analysis". At the bottom are "Pin" and "Cancel" buttons.

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



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