Lab 04: Dataflow Gen 2 in action

Introduction

In this lab, you will use Dataflow Gen2 to execute some advanced Data Preparation activities. You will first consume a complex JSON file with 2 methods, then you will extract and transform data from an Open Data environment based on REST API.

Objectives

After completing this lab, you will be better able to:

1. Import and Transform Data from a complex data structure
2. Understand how to implement advanced M code
3. Import and Transform Data from a REST API call

Estimated time to complete this lab

60 minutes

Contents

Lab Prerequisites

- Workspace: Fabric, Power Premium or Fabric trial
- Individual license: Power Pro or Premium Per User account

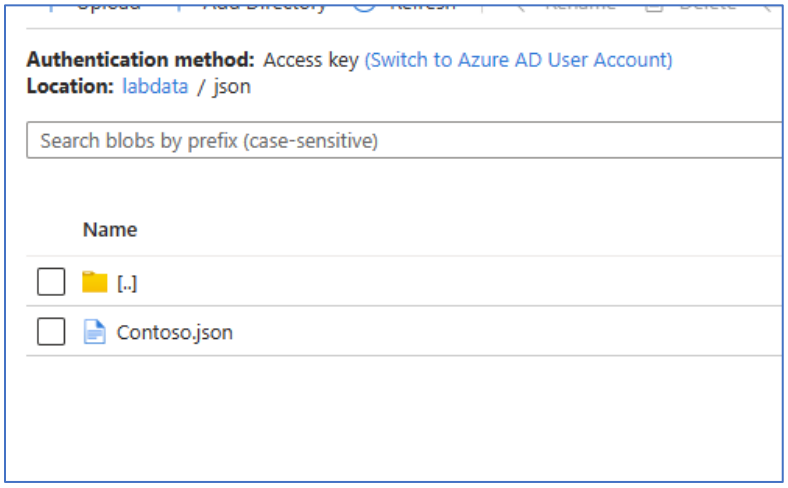
Information provided by your training provider

- Trial tenant (if applicable): login & password, workspace to use for the lab.
- Azure Data Lake Gen2 (containing data sources): account name & shared access signature.

Task 1: Import and Transform Data from a complex data structure

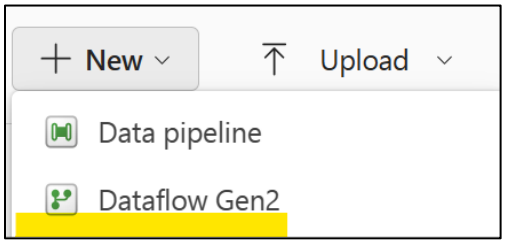
In this task, you will import data from a JSON file then manipulate the structure before loading it to the Lakehouse.

The file to be loaded is name **Contoso.json** and is stored in the labdate/json folder from the Azure Data Lake Store Gen 2 account used in the previous labs.

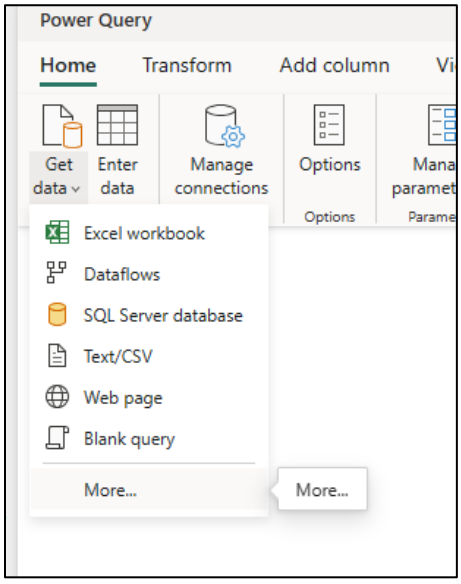


This file contains a list of contacts, with several attributes (age, gender…) and some multiple value attributes (tags, friends).

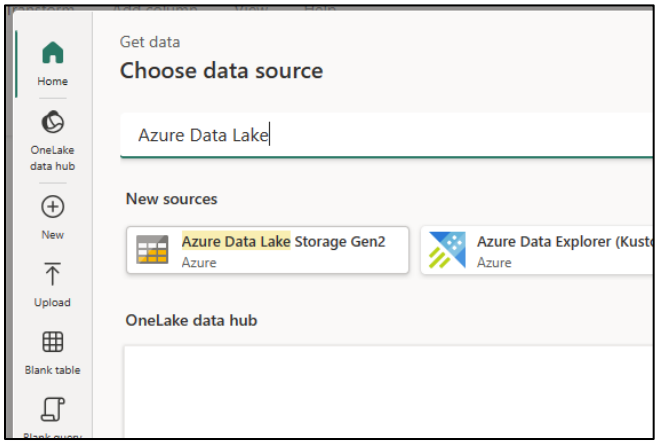
Create a **Dataflow Gen2** artifact in your workspace.



Go to the **Get Data** Menu and click on **More.**



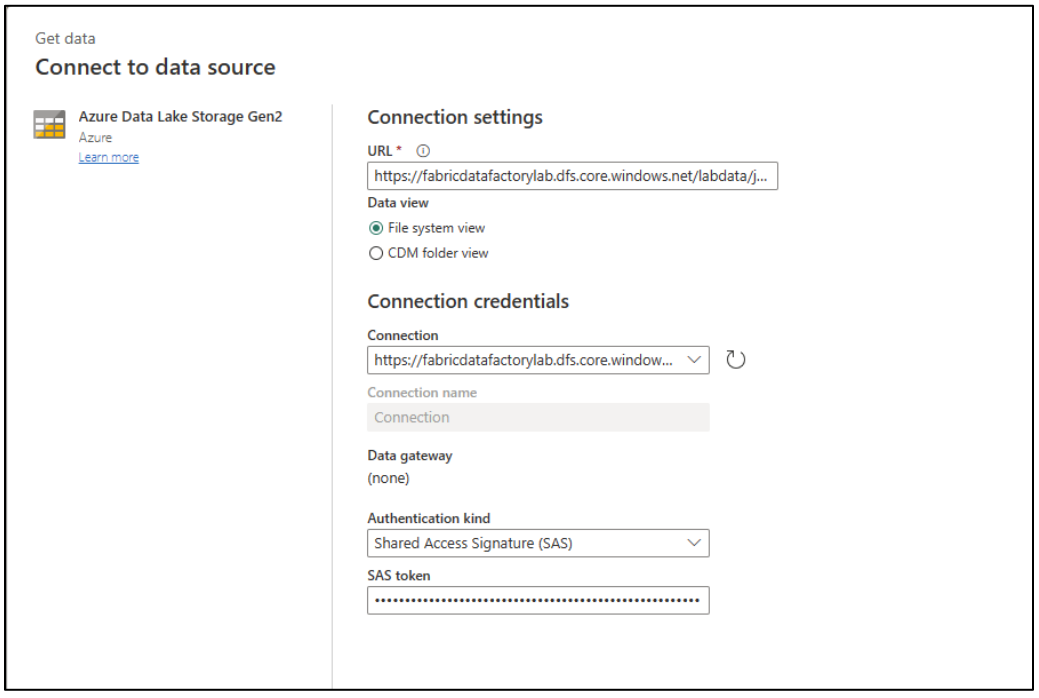
On the Choose data source window, type **Azure Data Lake** and select **Azure Data Lake Storage Gen2.**



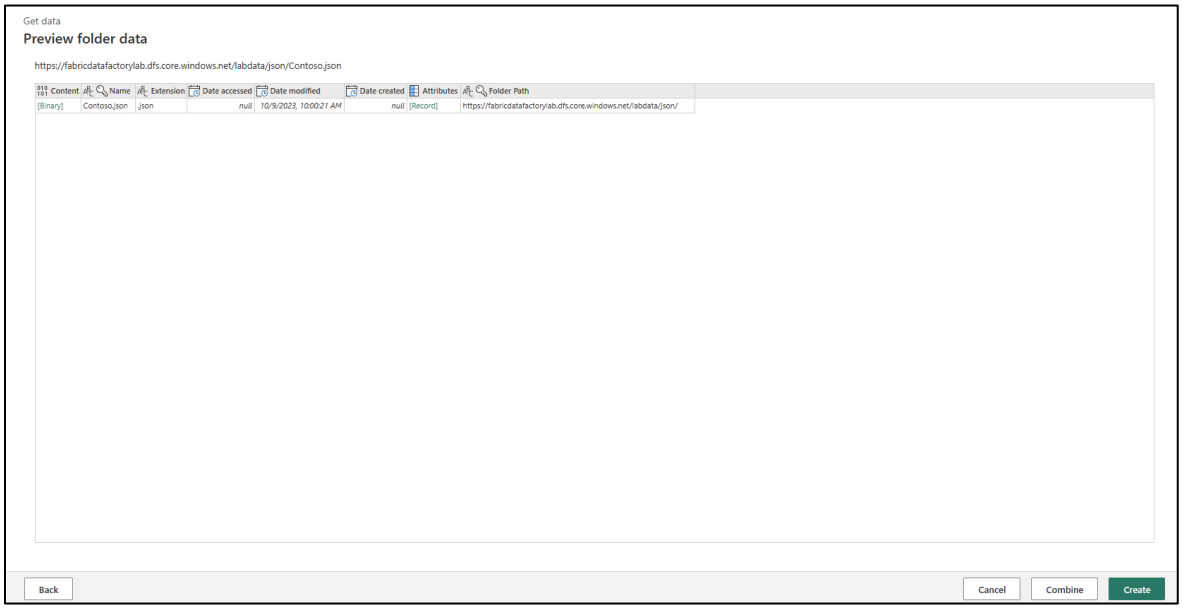
Configure the connection to the Contoso json file stored in the ADLS Gen2 account using a Shared Access Signature.

- **URL**: https://fabricvbdadlspublic.dfs.core.windows.net/labdata/JSON/Contoso.json
- **Authentication kind**: Shared Access Signature (SAS)

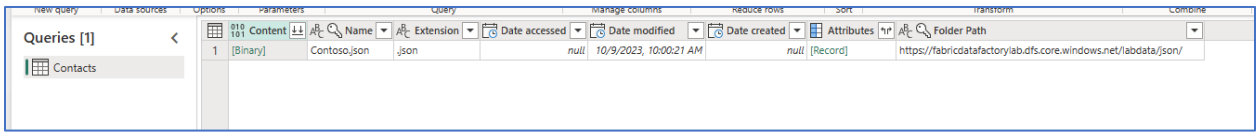
SAS Token: sv=2022-11-02&ss=bf&srt=sco&sp=rlx&se=2024-05-01T02:07:21Z&st=2024-04-10T18:07:21Z&spr=https&sig=Rw%2BE248HSJssu2xV4%2FbLogGXUowVjVHQ98niqWaL0Uo%3D



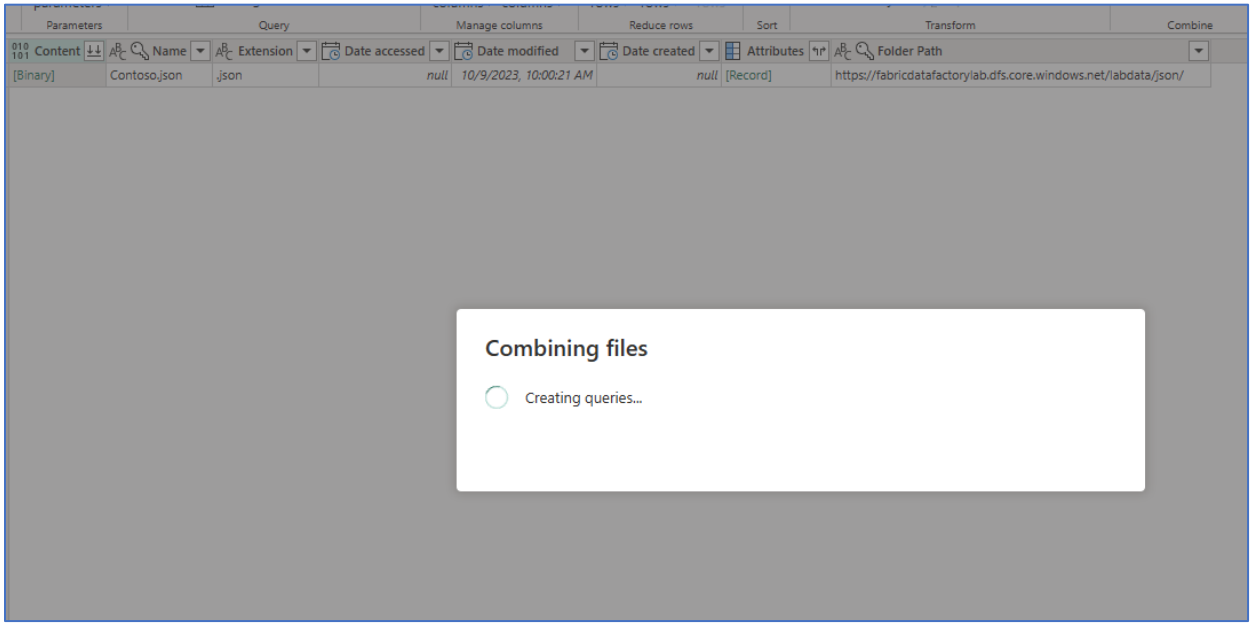
Click on **Create** to edit the query and extract data from the Contoso.json file.



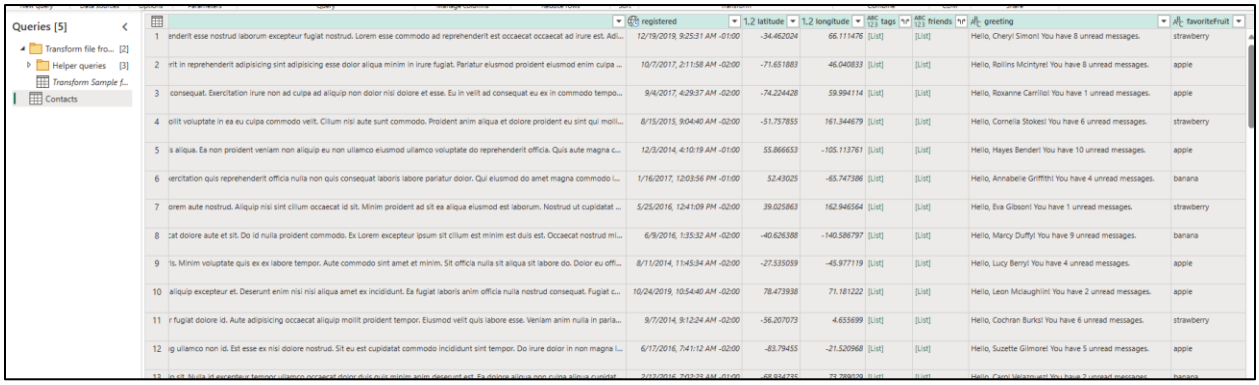
Rename the query named "Query 1" as "Contacts" in the Queries panel, and click on the icon in the **Content** column header to ask the Dataflow Gen2 to explorer and extract the data from JSON file.



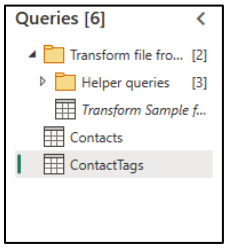
The Dataflow engine will automatically analyze the file structure and display the query result.



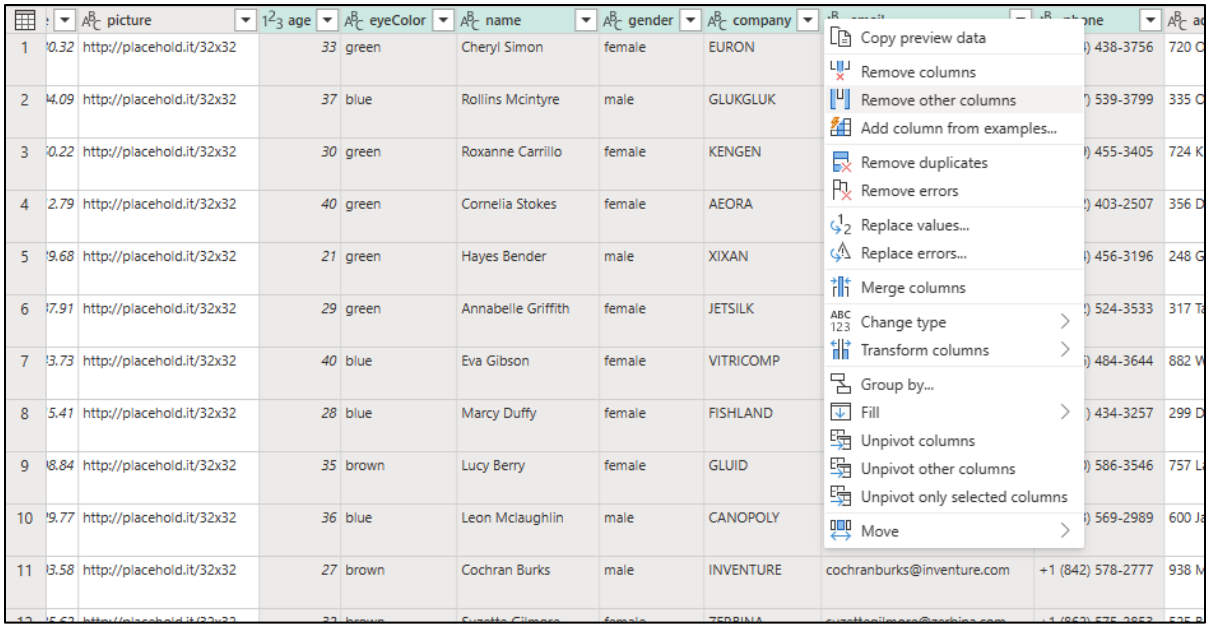
The query result is displayed on the Data Grid, and collection of resources have been automatically added.



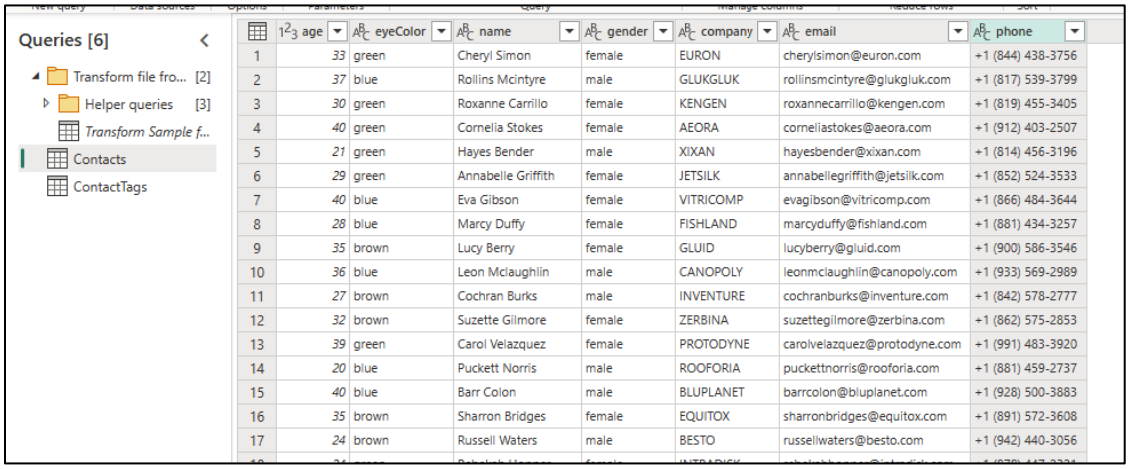
Select the **Contacts** query and clone it to create the new query **ContactTags** (to clone a query, right click, and select "Duplicate").



Select the **Contacts** query, and as indicated in the screenshot hereafter, select the following group of columns: age, eyeColor, name, gender, company, email and phone. Keep the selection active, right on the columns header and select **Remove other columns**.



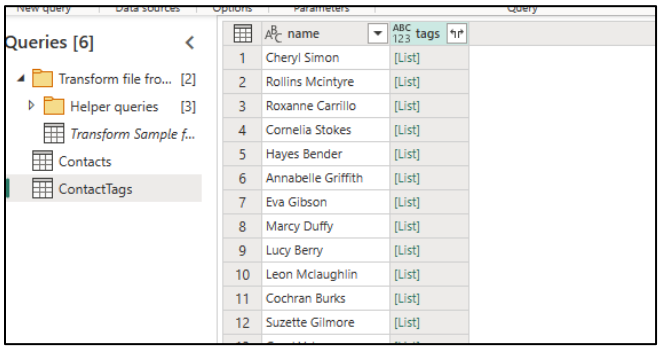
The **Contacts** query should only contain the selected columns.



	age	eyeColor	name	gender	company	email	phone
1	33	green	Cheryl Simon	female	EURON	cherylsimon@euron.com	+1 (844) 438-3756
2	37	blue	Rollins Mcintyre	male	GLUKGLUK	rollinsmcintyre@glukgluk.com	+1 (817) 539-3799
3	30	green	Roxanne Carrillo	female	KENGEN	roxannecarrillo@kengen.com	+1 (819) 455-3405
4	40	green	Cornelia Stokes	female	AEORA	corneliastokes@aeora.com	+1 (912) 403-2507
5	21	green	Hayes Bender	male	XIXAN	hayesbender@xixan.com	+1 (814) 456-3196
6	29	green	Annabelle Griffith	female	JETSILK	annabellegriffith@jetsilk.com	+1 (852) 524-3533
7	40	blue	Eva Gibson	female	VITRICOMP	evagibson@vitricomp.com	+1 (866) 484-3644
8	28	blue	Marcy Duffy	female	FISHLAND	marcyduffy@fishland.com	+1 (881) 434-3257
9	35	brown	Lucy Berry	female	GLUID	lucyberry@gluid.com	+1 (900) 586-3546
10	36	blue	Leon Mclaughlin	male	CANOPOLY	leonmclaughlin@canopoly.com	+1 (933) 569-2989
11	27	brown	Cochran Burks	male	INVENTURE	cochranburks@inventure.com	+1 (842) 578-2777
12	32	brown	Suzette Gilmore	female	ZERBINA	suzettegilmore@zerbina.com	+1 (862) 575-2853
13	39	green	Carol Velazquez	female	PROTODYNE	carolvelazquez@protodyne.com	+1 (991) 483-3920
14	20	blue	Puckett Norris	male	ROOFORIA	puckettnorris@rooforia.com	+1 (881) 459-2737
15	40	blue	Barr Colon	male	BLUPLANET	barrcolon@bluplanet.com	+1 (928) 500-3883
16	35	brown	Sharron Bridges	female	EQUITOX	sharronbridges@equitox.com	+1 (891) 572-3608
17	24	brown	Russell Waters	male	BESTO	russellwaters@besto.com	+1 (942) 440-3056

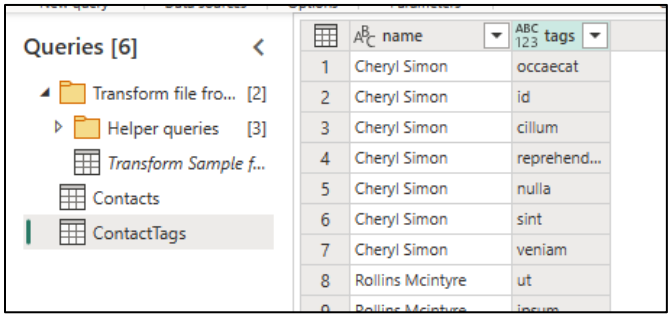
Repeat the same operation for the **ContactTags** query, to keep the following columns: name and tags.

Then click on the small button located on the **tags** column header to expand the column values (which is a list).



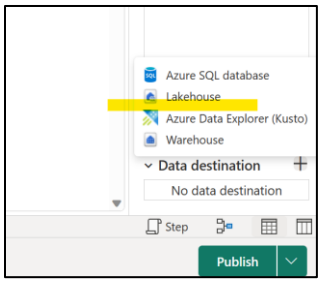
	name	tags
1	Cheryl Simon	[List]
2	Rollins Mcintyre	[List]
3	Roxanne Carrillo	[List]
4	Cornelia Stokes	[List]
5	Hayes Bender	[List]
6	Annabelle Griffith	[List]
7	Eva Gibson	[List]
8	Marcy Duffy	[List]
9	Lucy Berry	[List]
10	Leon Mclaughlin	[List]
11	Cochran Burks	[List]
12	Suzette Gilmore	[List]

As the **ContactTags** query has been expanded, the same name appears who each corresponding tag value.

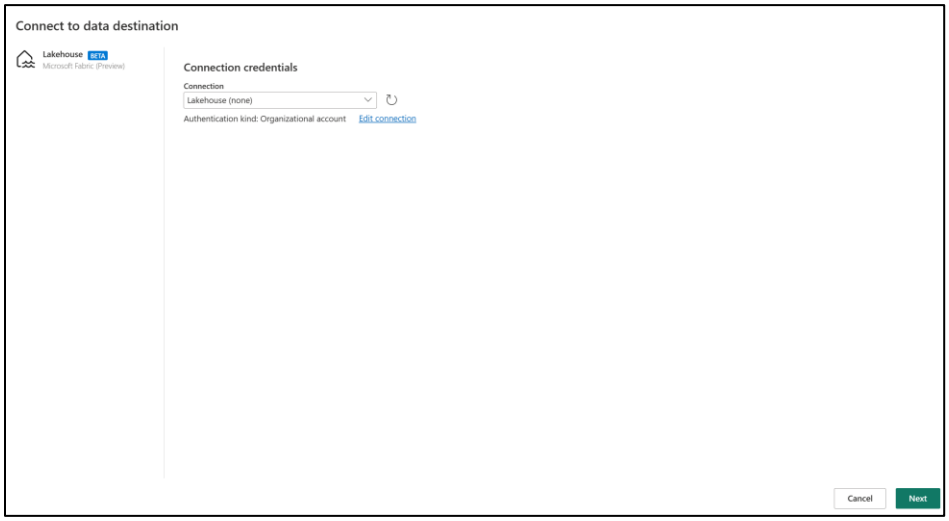


	name	tags
1	Cheryl Simon	occaecat
2	Cheryl Simon	id
3	Cheryl Simon	cillum
4	Cheryl Simon	reprehend...
5	Cheryl Simon	nulla
6	Cheryl Simon	sint
7	Cheryl Simon	veniam
8	Rollins Mcintyre	ut

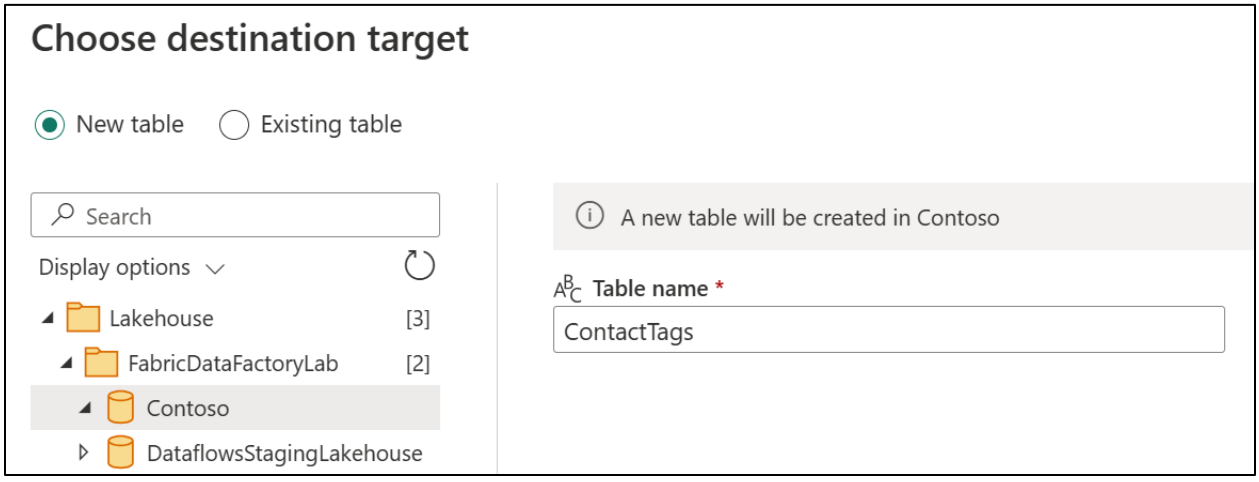
Click on the + icon in the **Data destination** area and select **Lakehouse**.



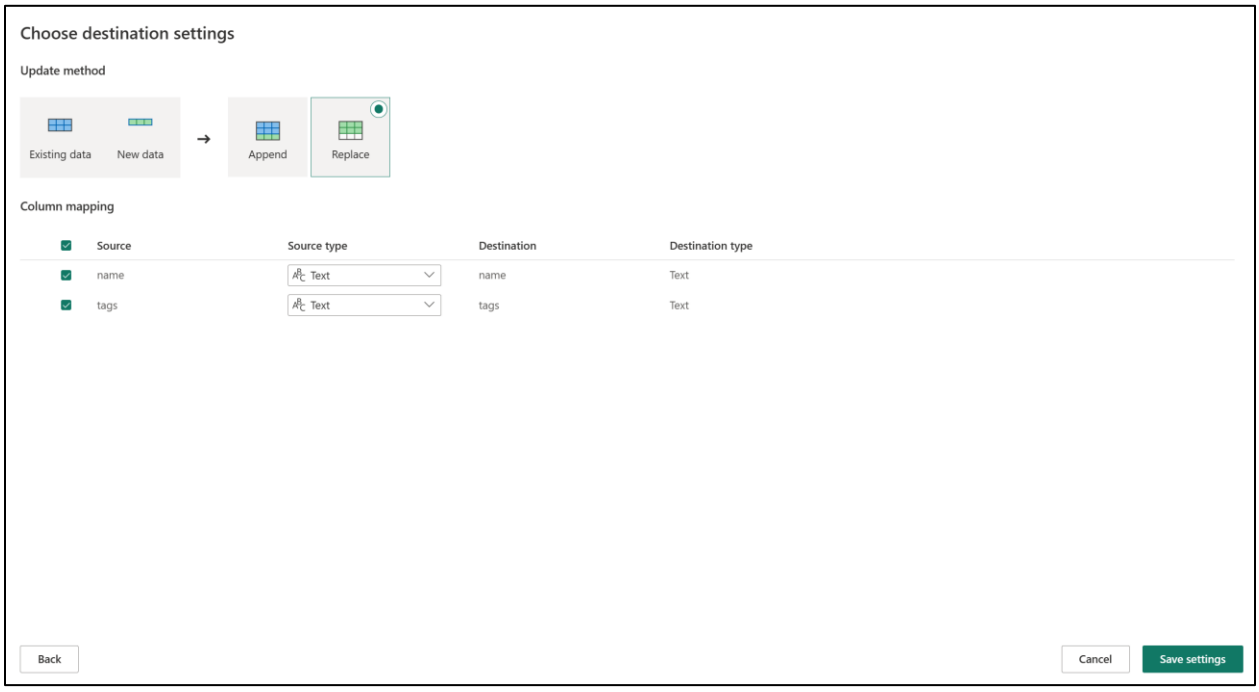
Keep the default configuration and click **Next**.



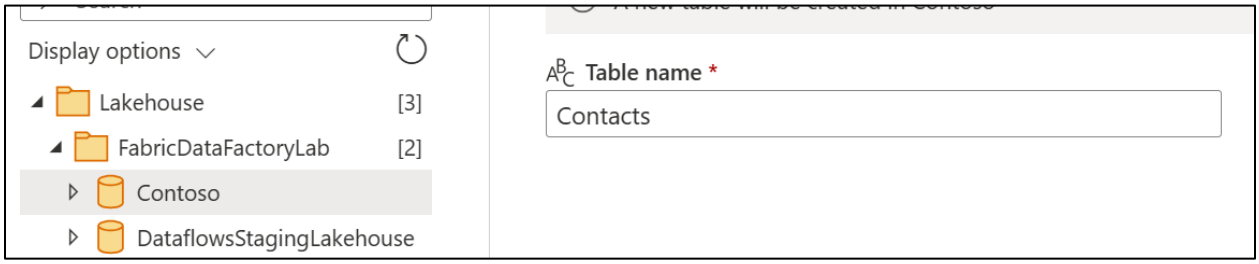
Select the existing **Contoso** lakehouse located in the same workspace, keep the destination table name as **ContactTags**, and click on **Next**.



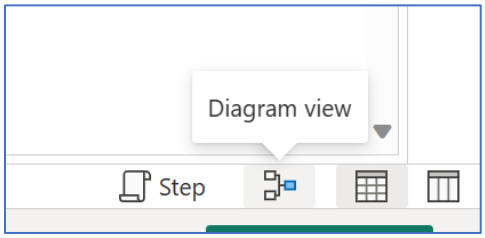
Change Source type value as **Text** for the **tags** column and click on **Save settings**.



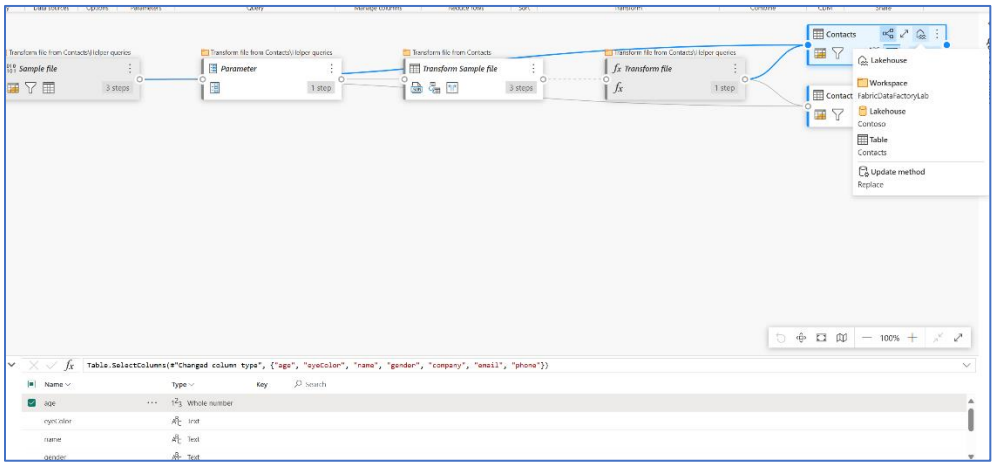
Repeat the same operation for the **Contacts** query to load data in the **Contoso** Lakehouse using **Contacts** as the destination table name.



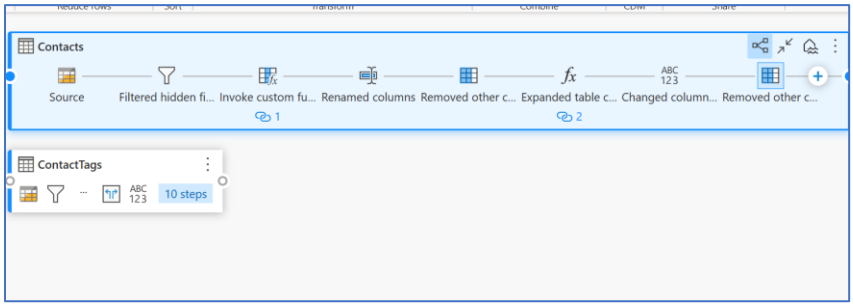
Click on **Diagram view** to see graphically how the 2 queries are structured.



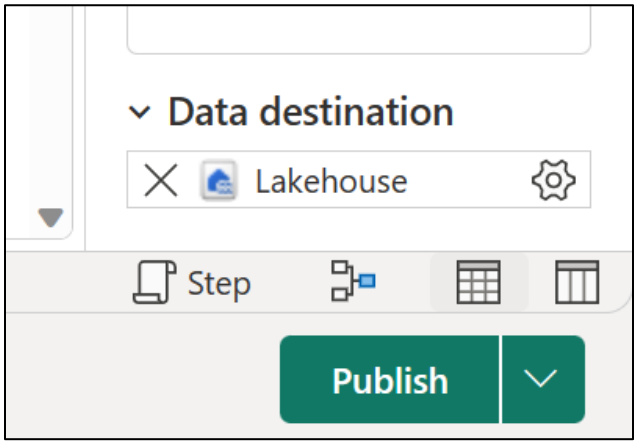
The **Contacts** query display the Lakehouse icon displaying where and how the data will be loaded.



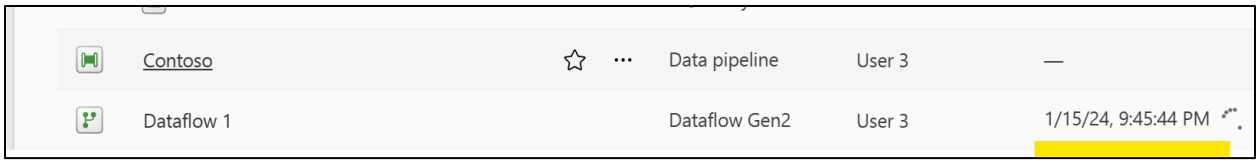
Expand the **Contacts** query to display the sequence of transformation step.



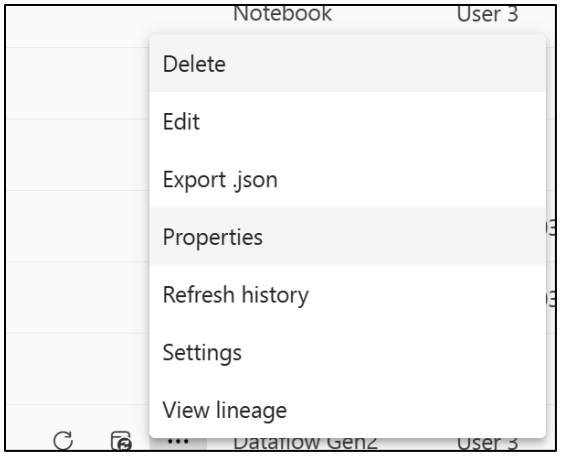
Click on **Publish** to save the Dataflow definition and start its first execution.



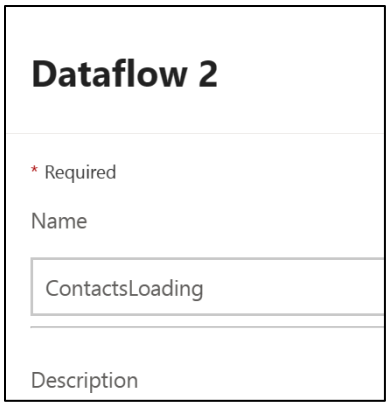
Wait until the end of the first Dataflow execution, by monitoring the column **Refreshed.**



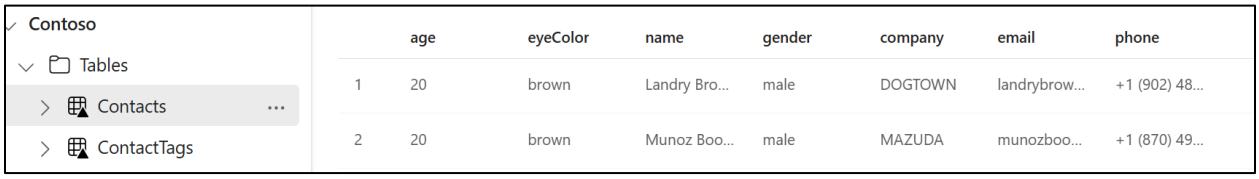
When the dataflow has been successfully executed, click on the Dataflow and select **Properties.**



Rename the dataflow as **ContactsLoading.**



Open the Contoso Lakehouse, and open the 2 tables Contacts and ContactTags, to make sure they contain data loaded from the CSV file.



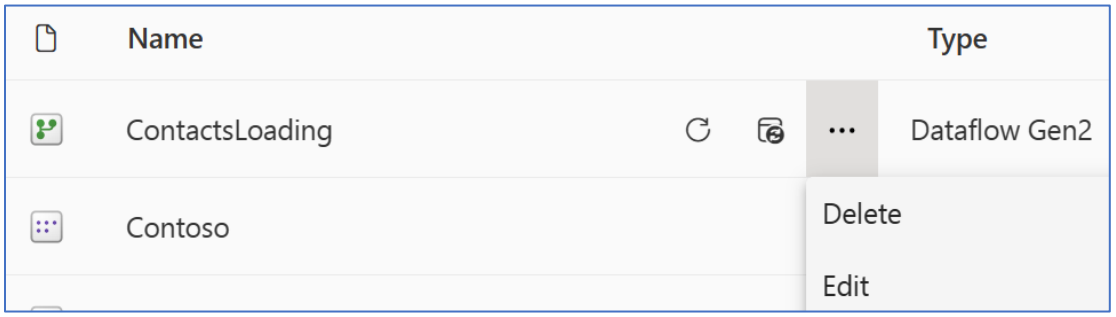
Task 2: Execute advanced M code

In this task, you will process the same **Contoso.json** file using an advanced M (mashup) code snippet which:

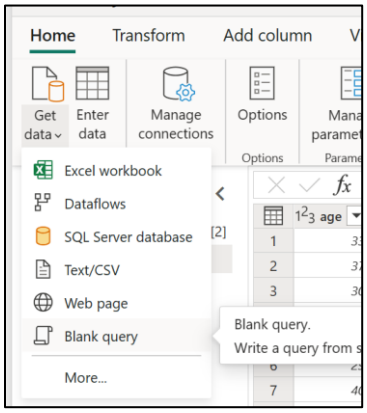
- Directly extracts JSON structure based on a schema definition provided in the code
- Combines 2 instances of the same table object (with a single read of the json file)
- Expose detailed information about the main contact and the friend

```
let

  //Read the JSON file
  Source = AzureStorage.DataLake("https://<storage account>.dfs.core.windows.net/labdata/json/Contoso.json"),

  JsonDoc = Json.Document(Table.Column(Source,"Content"){0}),

  //Display the file content as a list
  TableFromList= Table.FromList(
    JsonDoc,
    Splitter.SplitByNothing(),
    null,
    null,
    ExtraValues.Error
  ),

  //Expand the list to get contact informations (Friends are not yet expanded)
  ExpandedTable = Table.ExpandRecordColumn(TableFromList, "Column1",
                                           {"age", "eyeColor", "name", "gender", "company", "friends", "favoriteFruit"},
                                           {"age", "eyeColor", "name", "gender", "company", "friends", "favoriteFruit"}
                                           ),

  //Remove duplicate contact (this table will be used 2 times : for contact information and friends informations
  TableRemoveDuplicates = Table.Distinct(ExpandedTable,{"name"}),

    //Expand friends for each contact as a list
  TableWithExpandedFriendsColumn = Table.ExpandListColumn(TableRemoveDuplicates, "friends"),

    //Expand friends for each contact as records
  PersonTable = Table.ExpandRecordColumn(TableWithExpandedFriendsColumn, "friends", {"name"}, {"friend"}),

  //Prepare the table of friends by reusing the table "TableRemoveDuplicates"
  //Rename columns with the "friend" prefix
  FriendTable=Table.RenameColumns(
                    //Keep a subset of columns (contact information) from the table "TableRemoveDuplicates"
                    Table.SelectColumns(
                        TableRemoveDuplicates,
```

Edit the dataflow named **ContactsLoading** (right click and select **Edit**).



Click on **GetData** and select **Blank query**



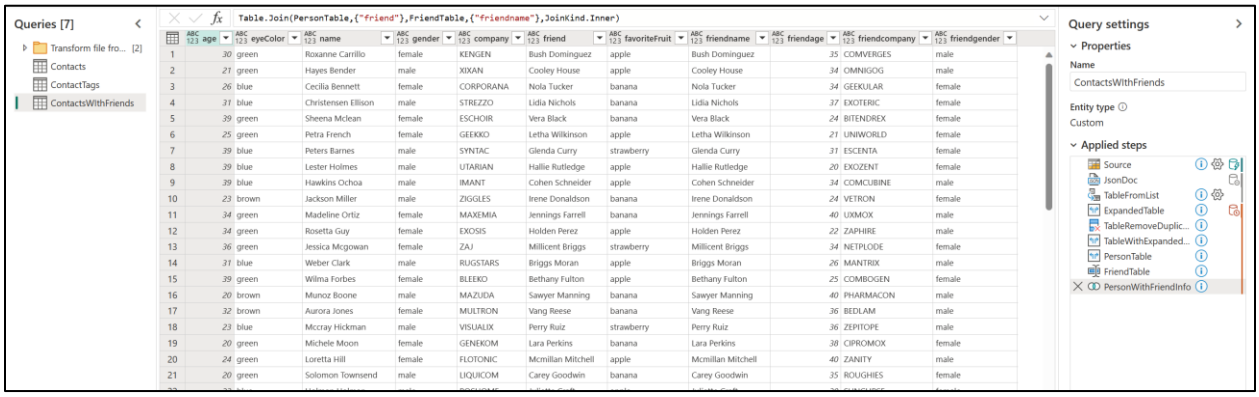
Paste the M code from the M_snippet.txt file, and replace <storage account> with the account name already used in the previous task.

Take the time to analyze the M code which is document with comments.

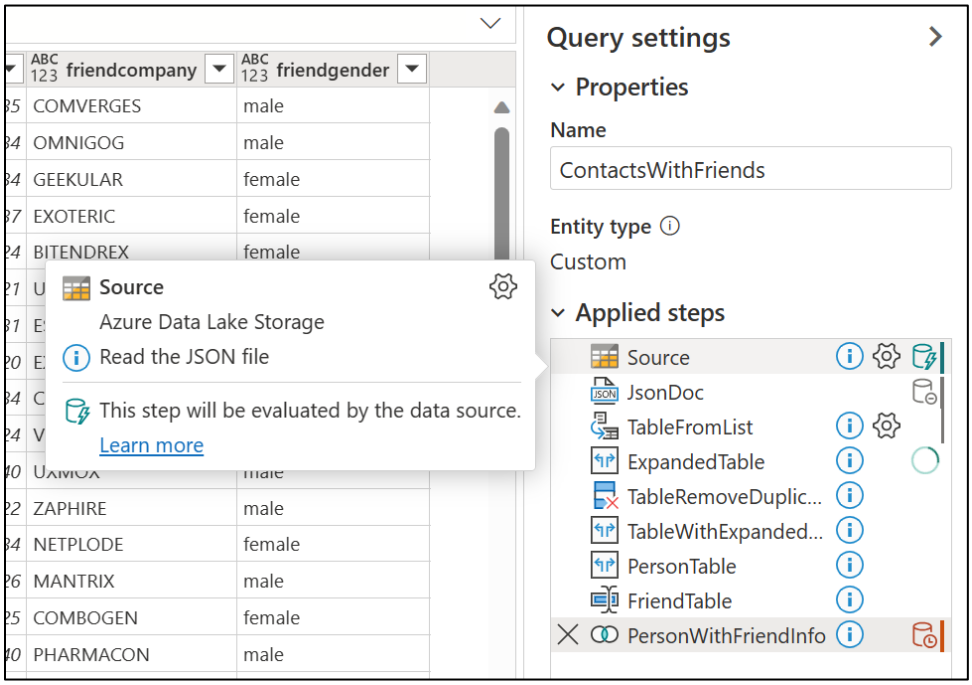
```
//Read the JSON file
Source = AzureStorage.DataLake("https://<storage account>.dfs.core.windows.net/labdata/json/Contoso.json"),

JsonDoc = Json.Document(Table.Column(Source,"Content"){0}),

//Display the file content as a list
```

After few seconds, the query result should appear in the data grid.

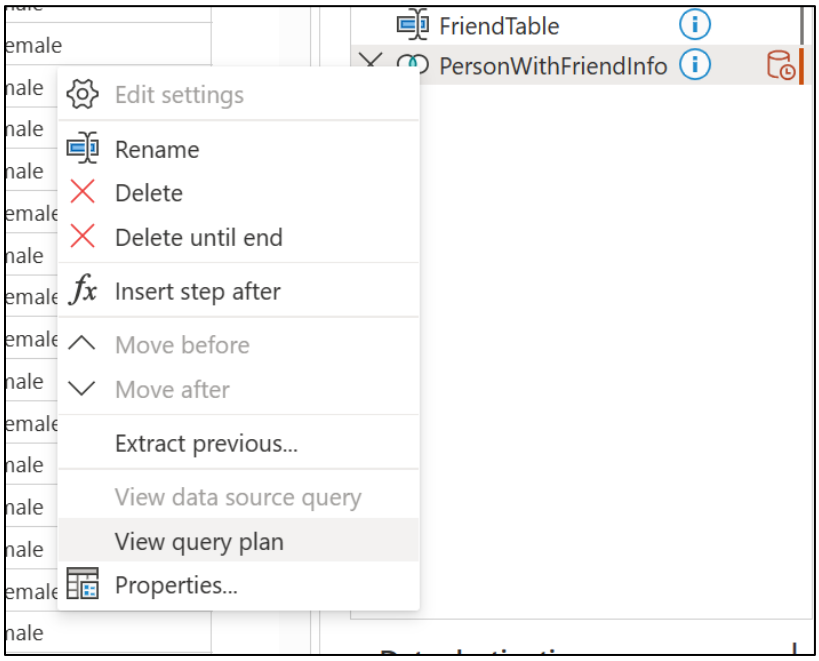
Rename the query as **ContactsWithFriends**.



In the Query settings panel, select the first step named **Source**, and look and the displayed properties : the lightning indicates that the activity is evaluated by the data source.

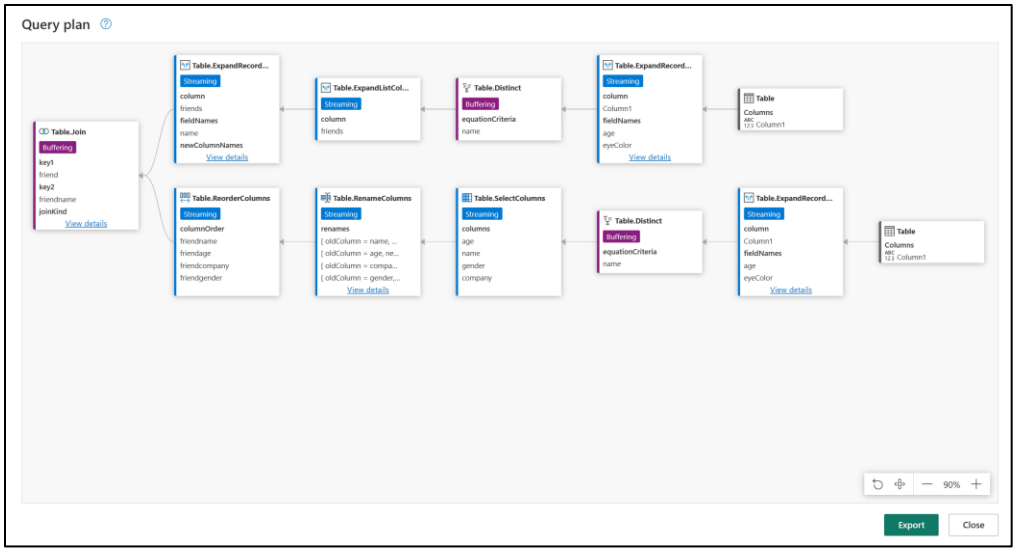


Select the last step **PersonWithFriendInfo**, right click and select **View query plan**.

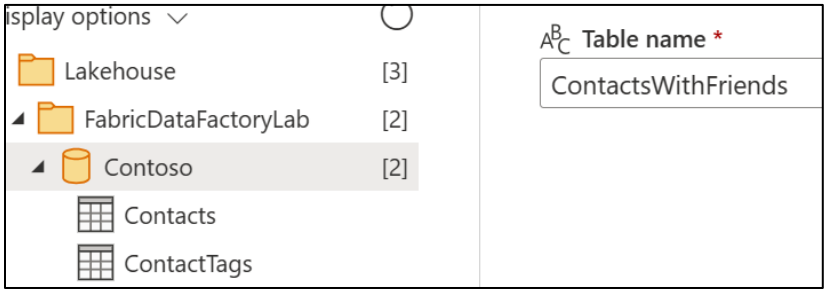


Review the Query Plan displayed with:

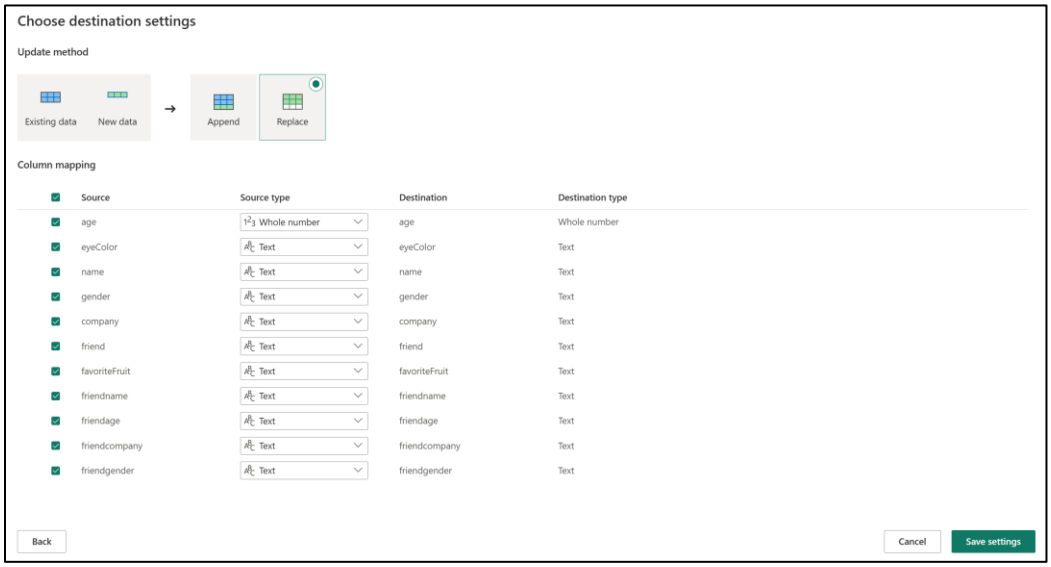
- The buffered tasks
- The streamed tasks



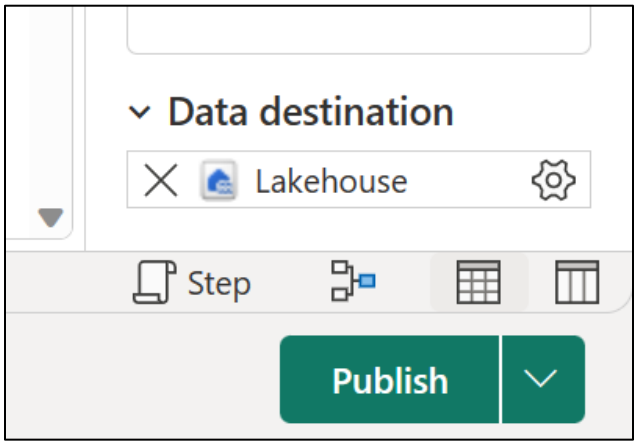
Close the query plan, and load the query to the Lakehouse using **ContactsWithFriends** as the table name.



Configure the source type as defined hereafter:

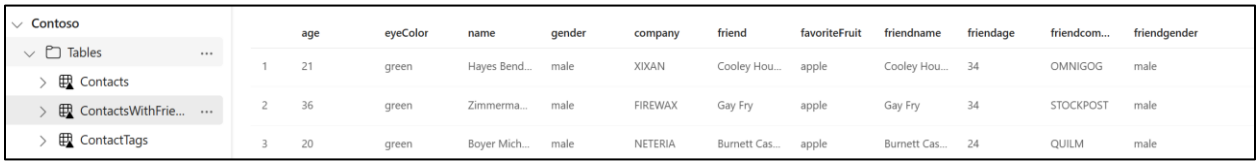


Click on **Publish**



Wait until the end of the dataflow execution.

Control the table content in the Lakehouse.

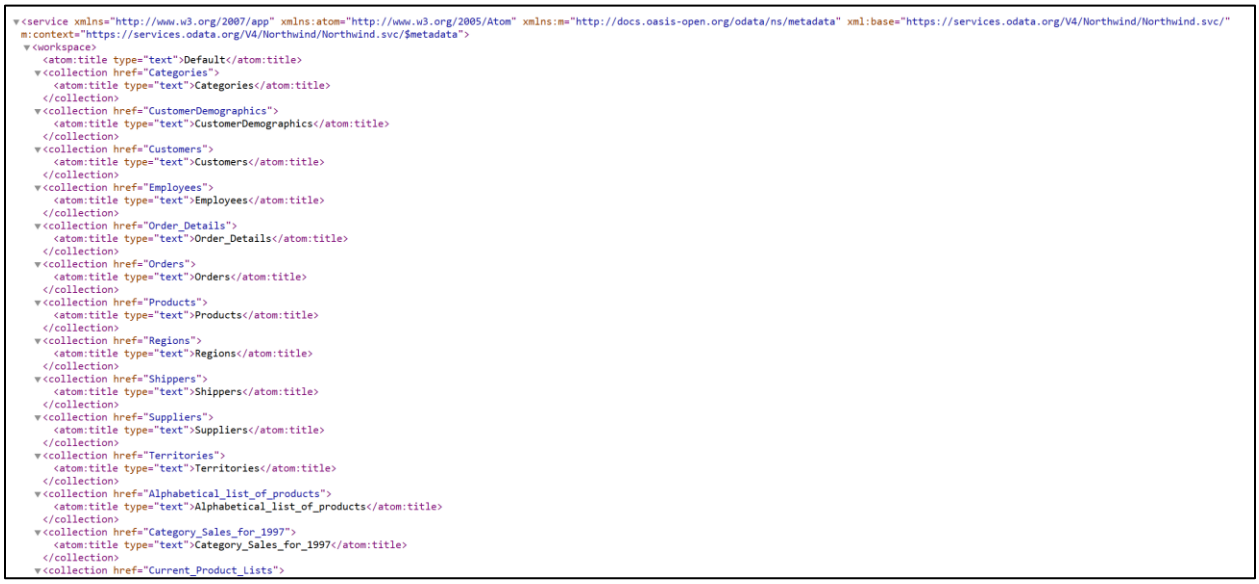


Task 3: Extract data from OData

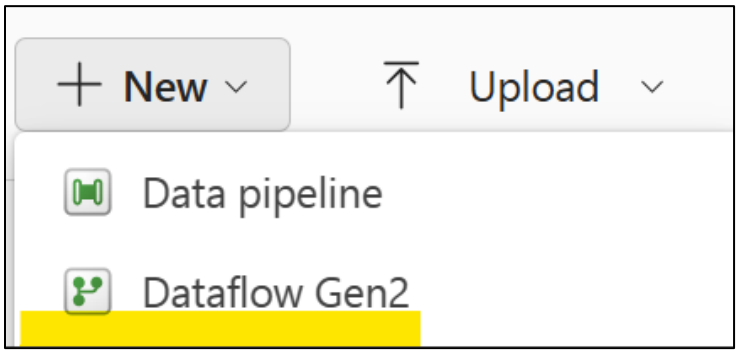
In this task, you will extract data from an OData API endpoint:

~~https://services.odata.org/V4/Northwind/Northwind.svc/~~

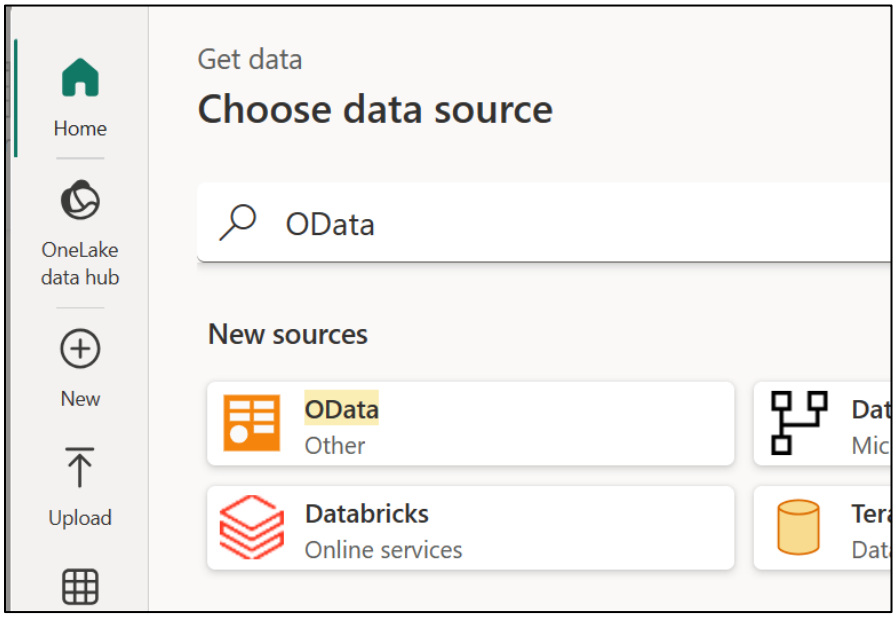
https://services.odata.org/V3/Northwind/Northwind.svc/



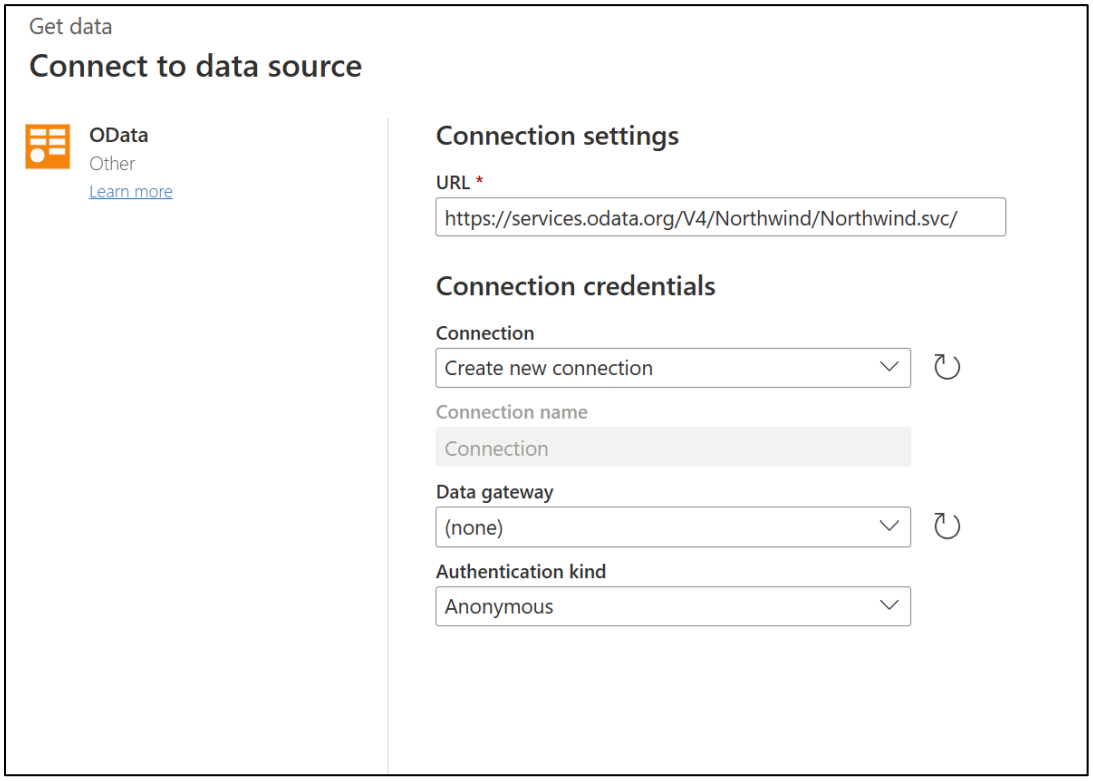
Create a new dataflow Gen2.



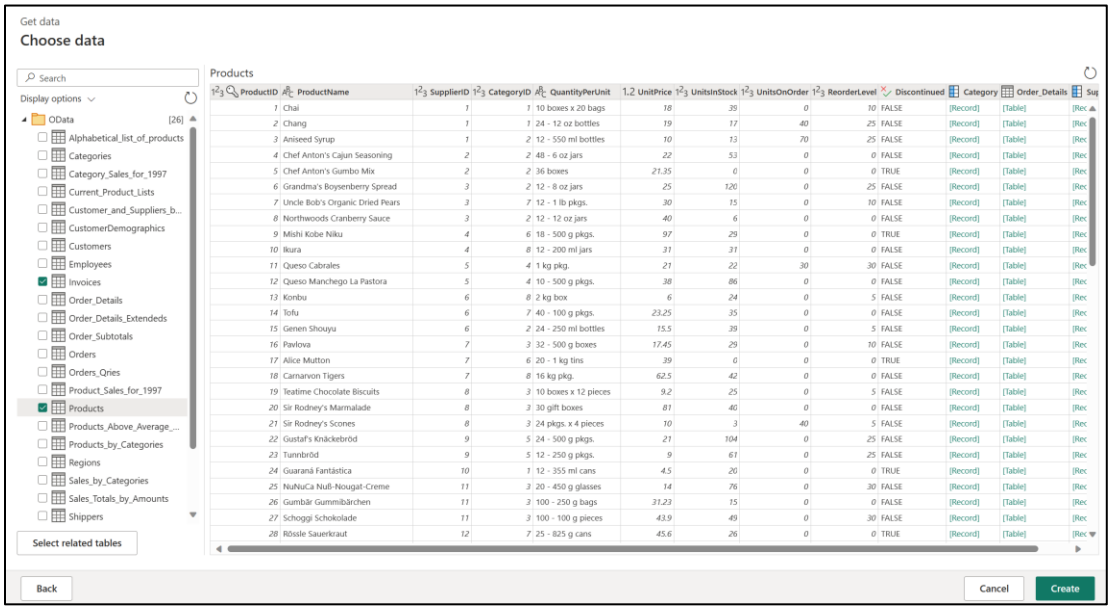
Choose the **OData** connector.



Define the URL and use the **Anonymous** authentication kind.



Select the table **Invoices** and **Products** and click on Create.

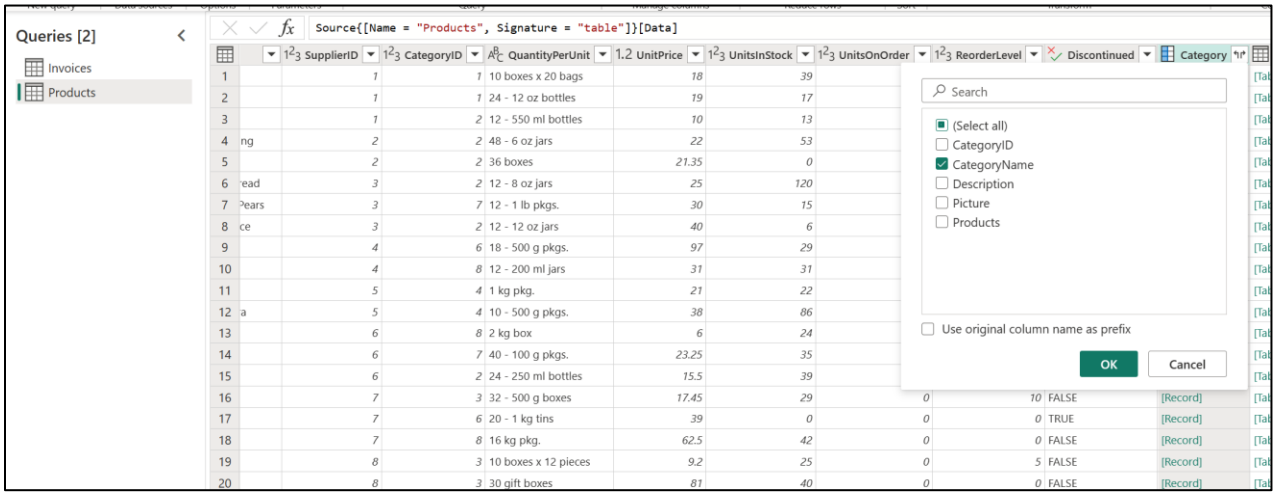


Select the **Invoices** query, expand the **Data** table by selecting the **Data** column.

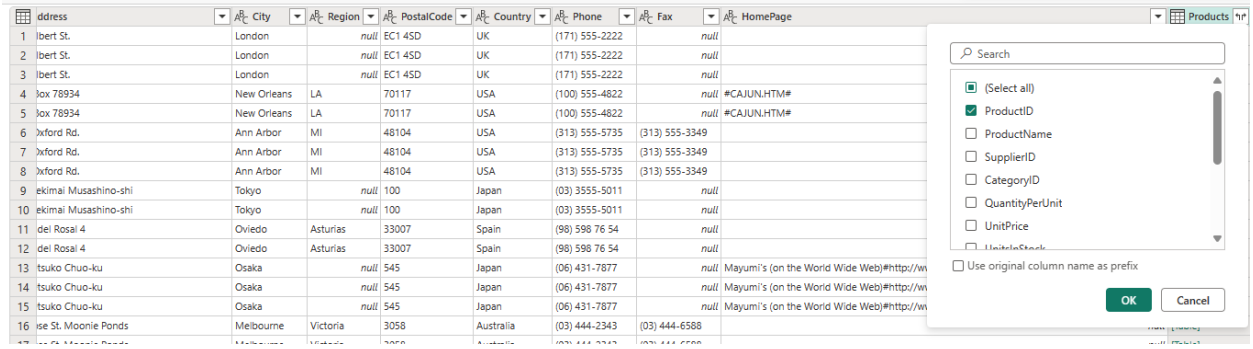
	Name	Data	Signature
1	Alphabetical_list_of_products	[Table]	table
2	Categories	[Table]	table
3	Category_Sales_for_1997	[Table]	table
4	Current_Product_Lists	[Table]	table
5	Customer_and_Suppliers_by_Cities	[Table]	table
6	CustomerDemographics	[Table]	table
7	Customers	[Table]	table
8	Employees	[Table]	table
9	Invoices	[Table]	table
10	Order_Details	[Table]	table
11	Order_Details_Extendeds	[Table]	table
12	Order_Subtotals	[Table]	table
13	Orders	[Table]	table

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued	CategoryName
1	1	Chai	1	1	10 boxes x 20 bags	18	39	0	10	FALSE	Beverages
2	2	Chang	1	1	24 - 12 oz bottles	19	17	40	25	FALSE	Beverages
3	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14	111	0	15	FALSE	Beverages
4	35	Steeleye Stout	16	1	24 - 12 oz bottles	18	20	0	15	FALSE	Beverages
5	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17	0	15	FALSE	Beverages
6	39	Chartreuse verte	18	1	750 cc per bottle	18	69	0	5	FALSE	Beverages
7	43	Ipoh Coffee	20	1	16 - 500 g tins	46	17	10	25	FALSE	Beverages
8	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14	52	0	10	FALSE	Beverages
9	70	Outback Lager	7	1	24 - 355 ml bottles	15	15	10	30	FALSE	Beverages
10	75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	7.75	125	0	25	FALSE	Beverages

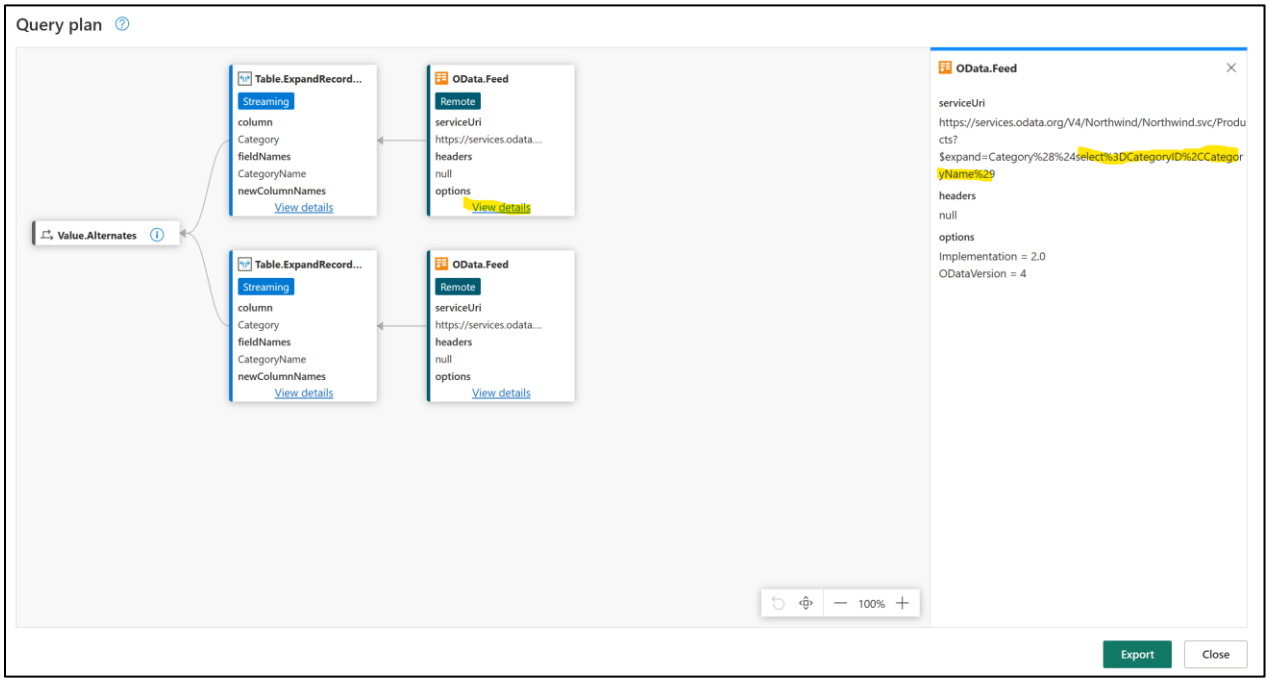
Select the **Products** query, expand the **Category** table by selecting the **CategoryName** column.



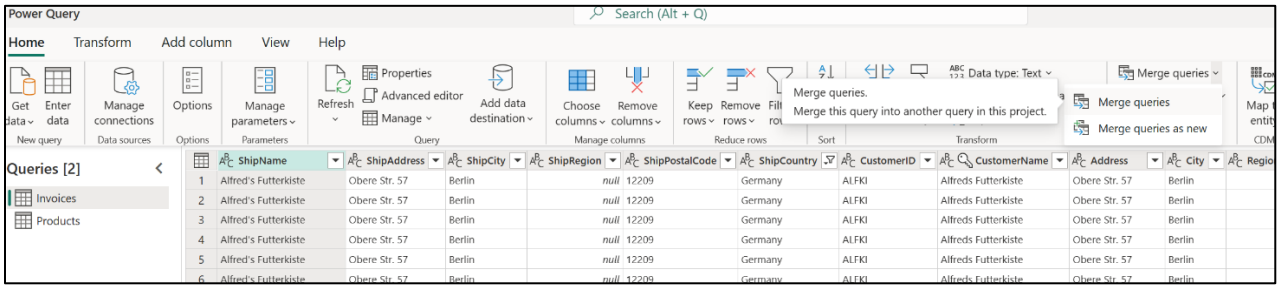
Select the **Products** query, expand the **Products** table by selecting the **ProductID** column.

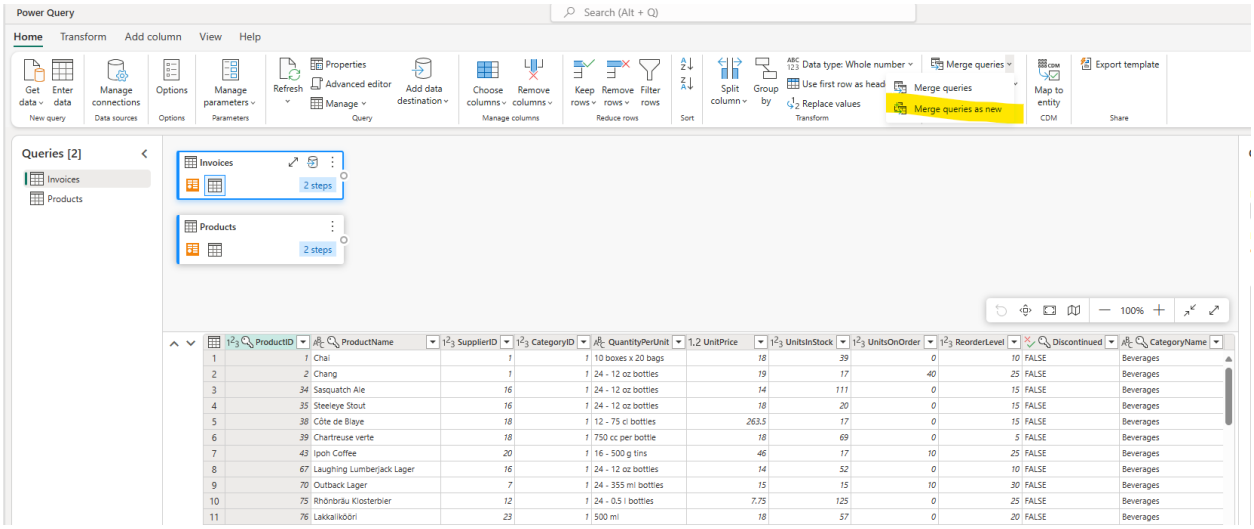


Display the execution plan for the **Products** table which contains 2 possible execution plans, one containing the "Expand" predicate to extract the Category Name from the Products entity.

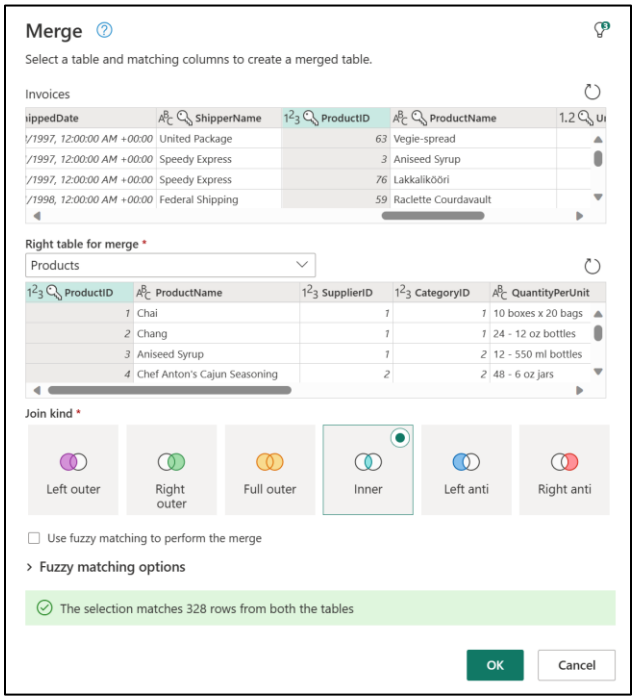


Select the **Invoices** query, then select the Merge queries from the Home ribbon.
https://learn.microsoft.com/en-us/power-query/merge-queries-overview

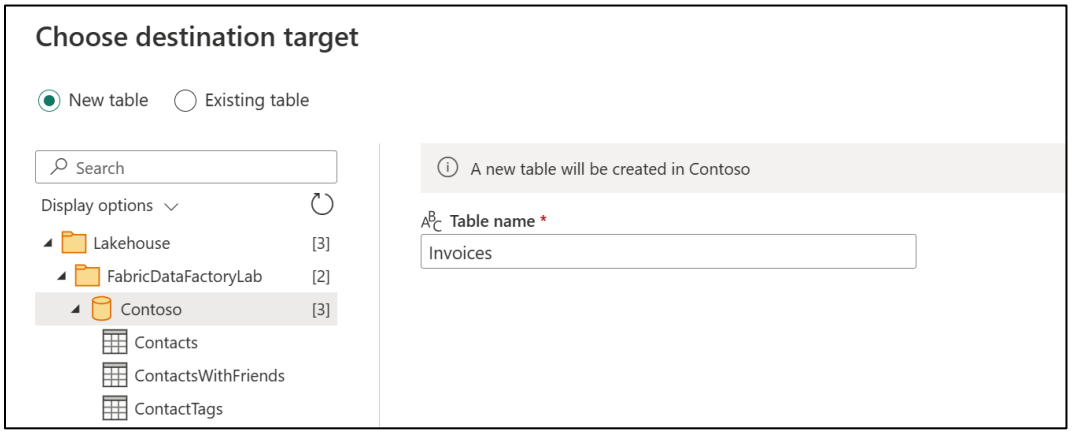



Power Query
Search (Alt + Q)
Home
Transform
Add column
View
Help
Get data
Enter data
Manage connections
Options
Manage parameters
Refresh
Properties
Advanced editor
Manage
Add data destination
Choose columns
Remove columns
Keep rows
Remove rows
Filter rows
Split column
Group by
Data type: Whole number
Use first row as head
Replace values
Merge queries
Merge queries
Merge queries as new
Map to entity
Export template
Queries [2]
Invoices
Products
Invoices
2 steps
Products
2 steps
100%
ProductID
ProductName
SupplierID
CategoryID
QuantityPerUnit
UnitPrice
UnitsInStock
UnitsOnOrder
ReorderLevel
Discontinued
CategoryName
1 1 Chai 1 1 10 boxes x 20 bags 18 39 0 10 FALSE Beverages
2 2 Chang 1 1 24 - 12 oz bottles 19 17 40 25 FALSE Beverages
3 34 Sasquatch Ale 16 1 24 - 12 oz bottles 14 111 0 15 FALSE Beverages
4 35 Steeleye Stout 16 1 24 - 12 oz bottles 18 20 0 15 FALSE Beverages
5 38 Côte de Blaye 18 1 12 - 75 cl bottles 263.5 17 0 15 FALSE Beverages
6 39 Chartreuse verte 18 1 750 cc per bottle 18 69 0 5 FALSE Beverages
7 43 Ipoh Coffee 20 1 16 - 500 g tins 46 17 10 25 FALSE Beverages
8 67 Laughing Lumberjack Lager 16 1 24 - 12 oz bottles 14 52 0 10 FALSE Beverages
9 70 Outback Lager 7 1 24 - 355 ml bottles 15 15 10 30 FALSE Beverages
10 75 Rhönbräu Klosterbier 12 1 24 - 0.5 l bottles 7.75 125 0 25 FALSE Beverages
11 76 Lakkalikööri 23 1 500 ml 18 57 0 20 FALSE Beverages

Define the **Merge** transformation using the **ProductID** for the **Inner** join and validate.

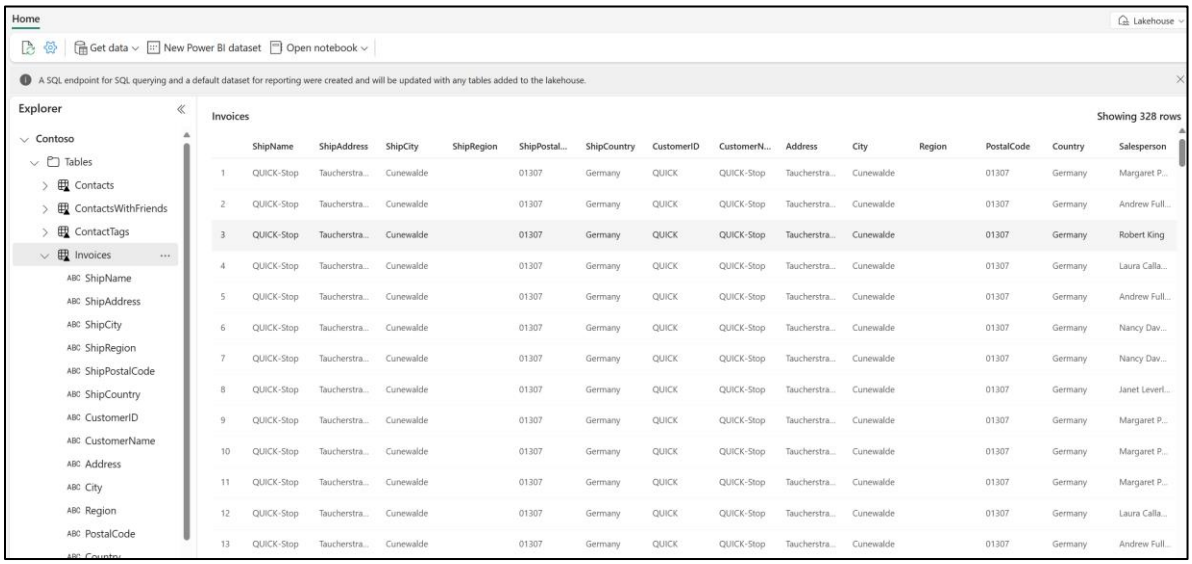


Configure the Lakehouse as the query destination, using **Invoices** as table name.



Publish the Dataflow.

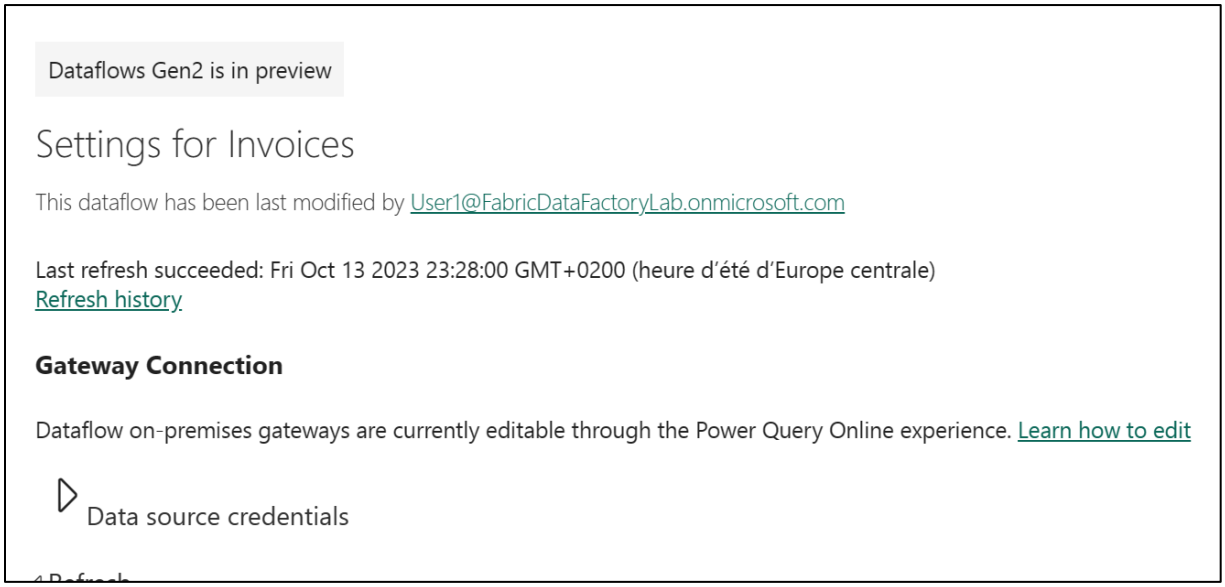
Explore the **Invoices** table in the Lakehouse.



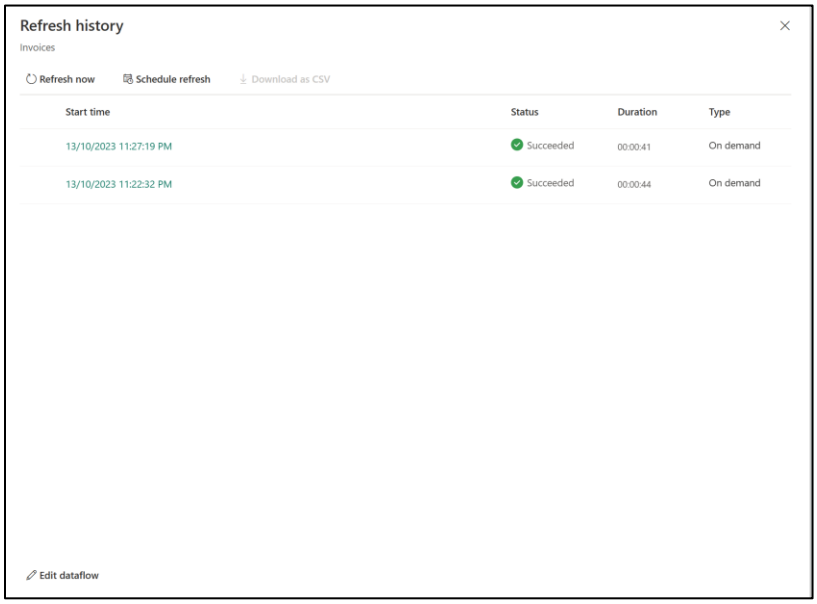
Wait until the end of the dataflow execution and rename it as **Invoices**.

Open the **Refresh Schedule** option for the **Invoices** dataflow.

Select the **Refresh history** link.



Refresh history gives details about the recent dataflow execution.



Configure and **Daily refresh** and click on **Apply**.