Loading and updating very large datasets in Microsoft Fabric

Introduction

How to load a large Dataset into Fabric using Dataflow Gen 2 and set up incremental refresh.

A am frequently requested to show how to load a very large dataset into Fabric and to set up incremental refresh so that only the most recent data is appended to the dataset and historical data is not loaded every time a refresh is done.

Usually this request comes with I have tried to load all of the data but it times out and I can't even get the historical data loaded.

The database I am using as an example is an on-premises SQL Server Database, it is currently 1.265 Terabytes in size and has a little more than 1.8 billion rows of data. I am streaming data from an IoT device and writing it to this database and the application that writes this real time streaming data runs 24x7x365, so the dataset just gets larger every day.

The source data is in an On-Premises SQL Database and I am using a Data Gateway to access it in Fabric, and I am placing this data in a Fabric Lakehouse.

Create a new Dataflow Gen 2.

To load a large dataset into Fabric you can load it in small chunks that won't time out before they complete. In my case I am loading from 300,000,000 to 600,000,000 rows in each chunk, the size you land on will depend on how much data is in each row. I am pulling historical data in one year at a time and I am using Dataflow Gen 2 to accomplish this you will need to create a new Dataflow Gen 2 and issue queries to pull chunks of data at a time. This is set up to use the date field to signify how much data is pulled in. In my case I can pull a years' worth of data at a time without getting a timeout.

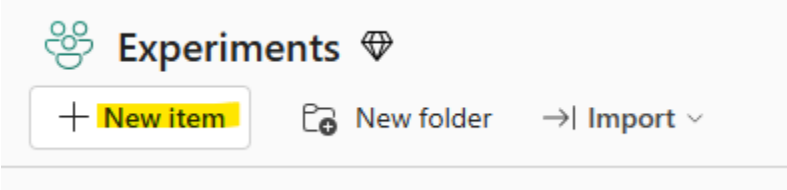
Steps to set up the Dataflow Gen 2 and pull your initial chunk of data.

Go to an existing workspace or create a new workspace.

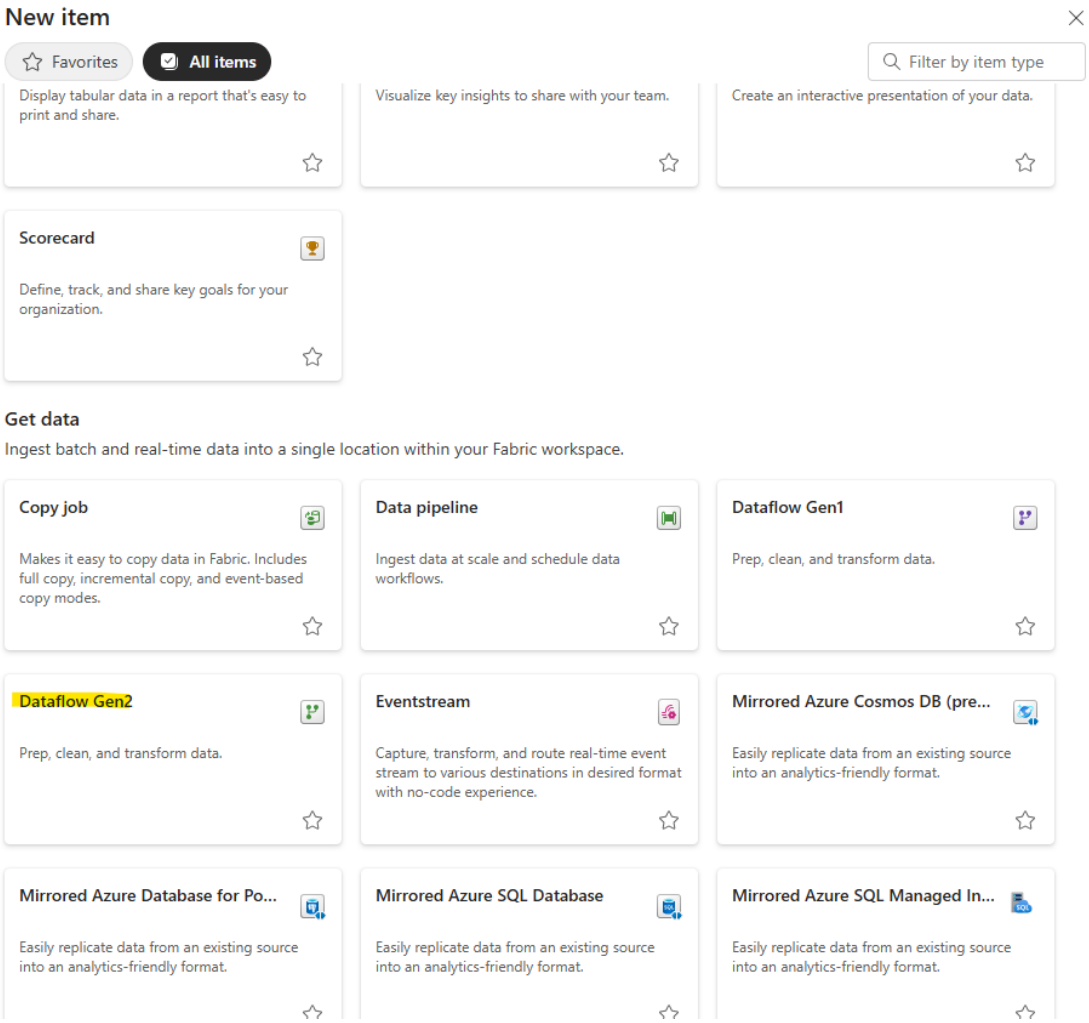
Note: This will only work on data sources that support query folding.

Query Folding is the ability of the Dataflow to push aggregates down to the data source so that you are not pulling all of the data back and filtering the data in the Dataflow, the actual query that is sent to the data source is a query with the required filters applied to just pull back what you have filtered in Power Query.

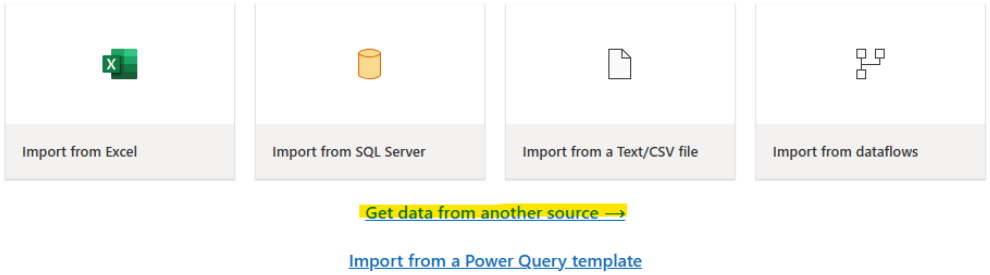
1. Select the + New item button in the top left corner of the screen while in your workspace



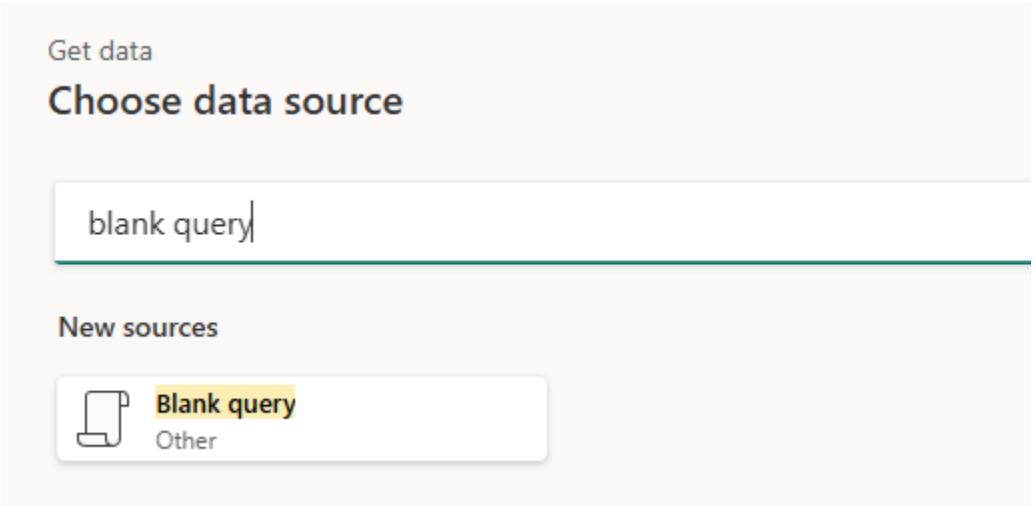
2. Select Dataflow Gen2 from the list of available resources you can create



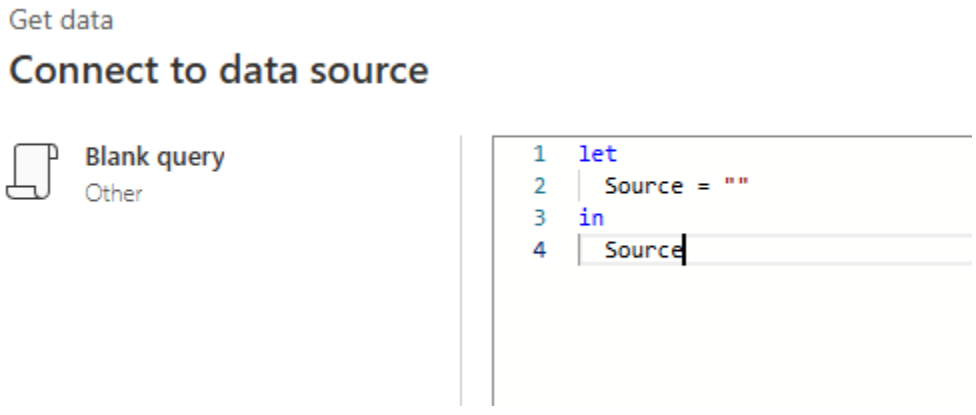
3. Click on Get data from other sources



4. Type in Blank Query in the search bar and select blank query.



5. You are now in the query editor with a blank query.



6. Delete the blank query and paste the following query in the editor.

```
let
        // Define parameters
        //RangeEnd = DateTime.LocalNow(),
        //RangeStart = Date.AddDays(RangeEnd, -1),

        RangeStart = #datetime(2023,1, 1, 0, 0, 0),
        RangeEnd = #datetime(2024, 1, 1, 0, 0, 0),

        // Source data
        Source = Sql.Database("db", "piawaredb"),
        Table = Source{[Schema="dbo", Item="KDFW"]}[Data],

        // Filter data using parameters
        FilteredData = Table.SelectRows(Table, each [dt] >= RangeStart and [dt] <
RangeEnd)
in
        FilteredData
```

Modify the date time parameters to be the dates you want you want as the chunk size. We will modify these times and run the query has loaded all you data.

In my example I was pulling back chunks of data between about 160,000,000 rows all the way up to 550,000,000 rows. My query times ranged between 2 and 4 hours depending on how much data was retrieved in each chunk.
Date format is YYY, MM, DD, HH, MM, SS so May 14th 2024 at 5:00 pm would look like (2024, 5, 14, 17, 0, 0)

To initially load the data pull it in in chunks using dates above.

Click Next in the lower right-hand corner.

Note: If you are using a data source other than SQL you can get the syntax of how to do that from Copilot in the Microsoft Edge browser, open Edge and click on this icon in the top right corner of the screen. 

I used this question:

I need a Power Query M example of how to pull data out of a Postgres database from within a Dataflow Gen 2

I got this response back:

```
let
   // Define the connection string
   ConnectionString = "Host=myserver;Port=5432;Database=mydatabase;User ID=
myuser;Password=mypassword",
   // Create the PostgreSQL connection
   Source = PostgreSQL.Database(ConnectionString, [Query="SELECT * FROM myta
ble"]),
   // Perform any additional transformations if needed
   Result = Table.TransformColumnTypes(Source, {{"column1", type text}, {"column2",
 type number}})
in
   Result
```

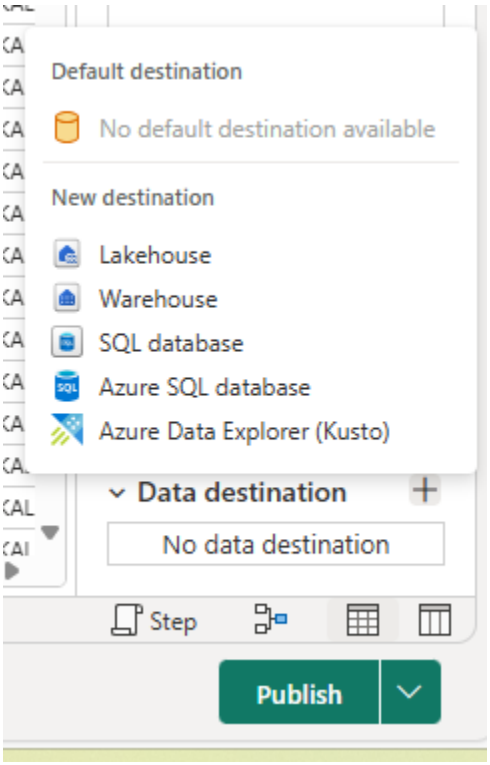
You will need to replace the definition of the Source variable to the Source variable definition in the response from Copilot.

Alternatively, you can create a temporary Dataflow Gen 2, use the built in connector to connect to your data source, Edit the query and extract the Power Query M code to connect to the source from there. Once you have the M code for the source, use that as your source and then delete the Dataflow Gen 2.

Pick a destination for your data and run the Dataflow.

Click on the arrow button next to Publish and choose where you want your data saved. You will need to create this resource if it does not already exist. Make sure the location for your data can hold the amount of data your entire dataset will store there, plus add for growth.

Lakehouse is a good choice for large amounts of data.



Pick an option with a maximum storage limit large enough to load all of your data and still have room for growth.

Lakehouse: Unlimited

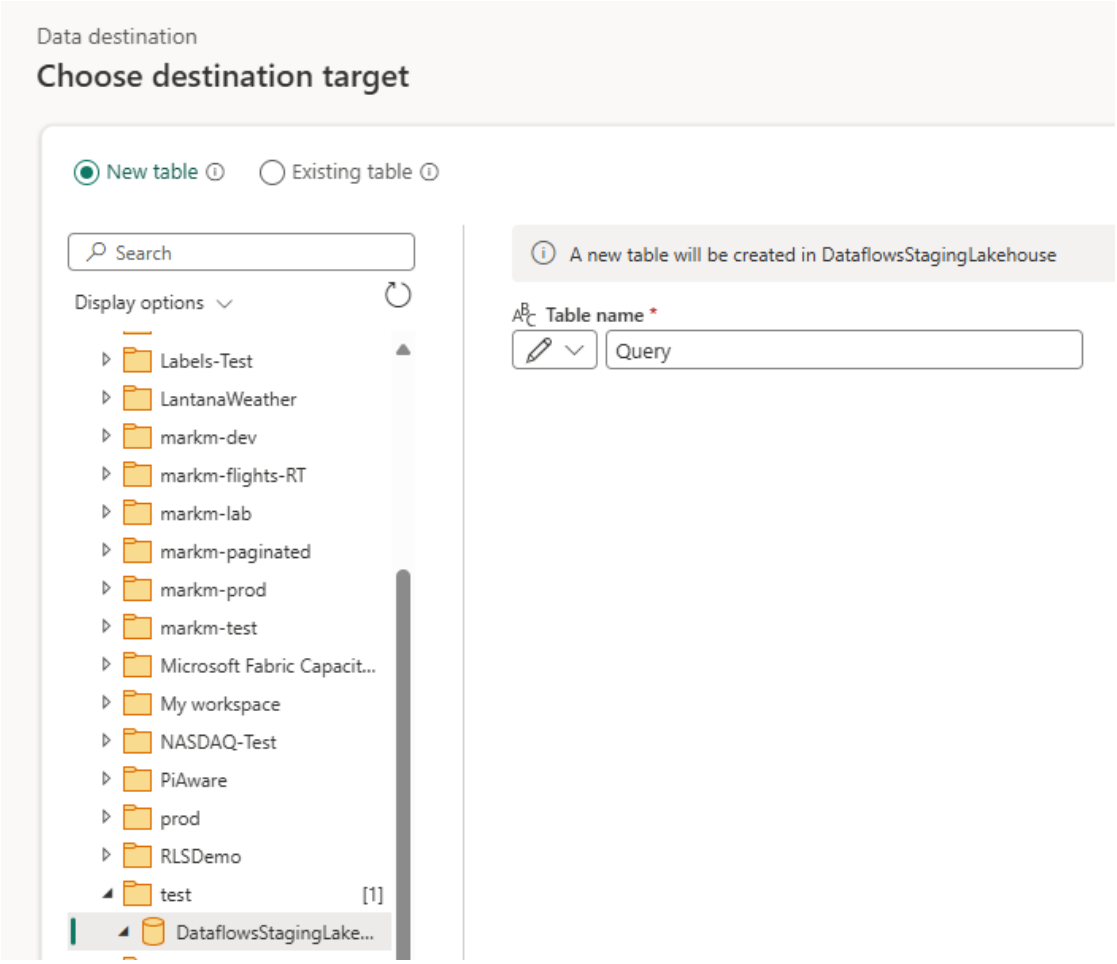
Warehouse: Unlimited

Fabric SQL database: 4 TB

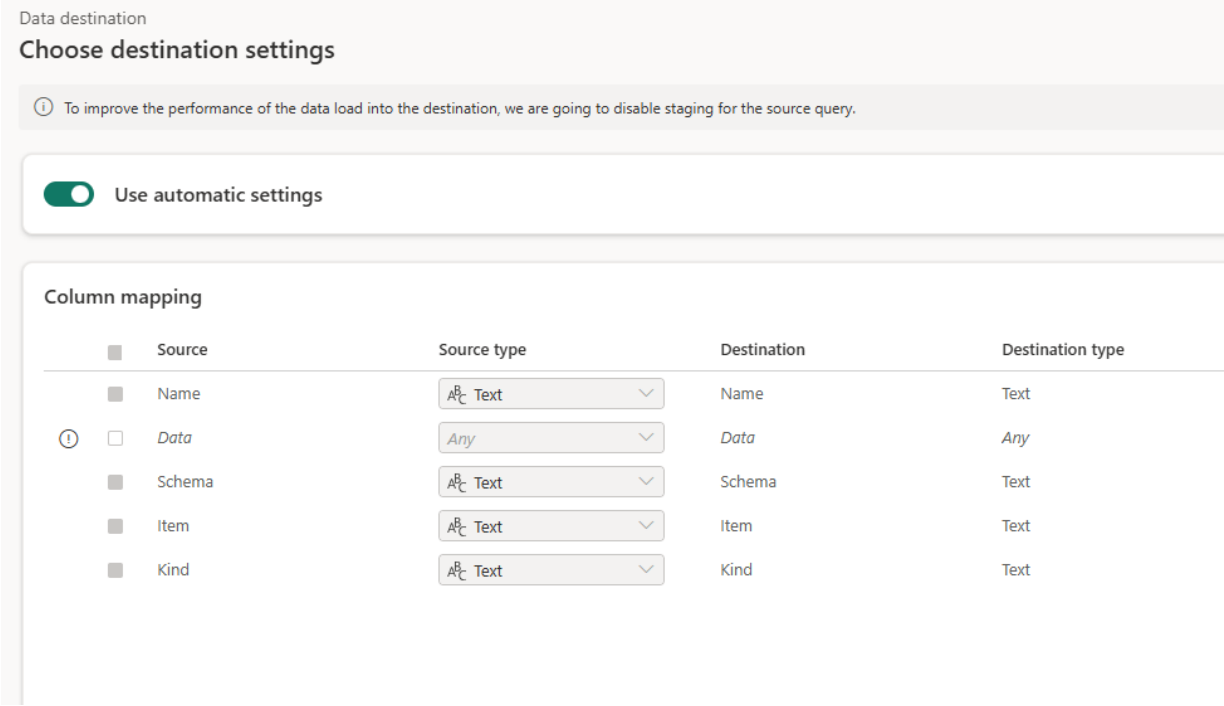
Azure SQL database: 100 TB

Azure Data Explorer (Kusto): Unlimited

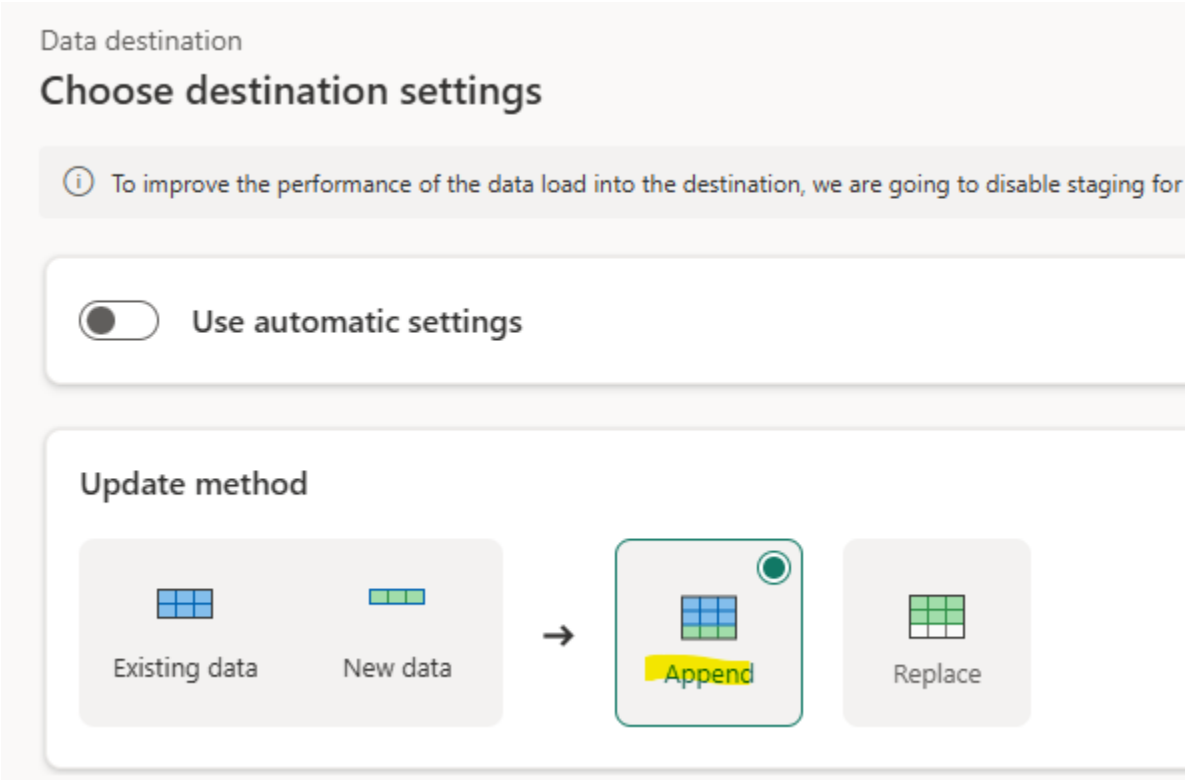
7. You can choose to create a New Table or an Existing Table, The default name of the table will be ‘Query’ , you can use this screen to select another name.



8. The next screen you can use to change the data type of columns if you want something other than what Fabric chooses.



9. On this same screen, turn off Use automatic settings and choose to Append new data to the existing data.



Once you have chosen a destination click on Publish and your data will begin to load. Once that chunk has completed, edit the dataflow dates for the next chunk and run it again. Once you have completed your initial data load you can move on to the Incremental Refresh step.

10. Once you have completed this step you can click Save settings on the lower right-hand corner of the screen.

11. Next click on Publish

12. You can check the status of each batch using the following query

```
SELECT
        YEAR(dt) AS year,
        COUNT(*) AS row_count
FROM
        KDFW
GROUP BY
        YEAR(dt)
ORDER BY
        year;
```

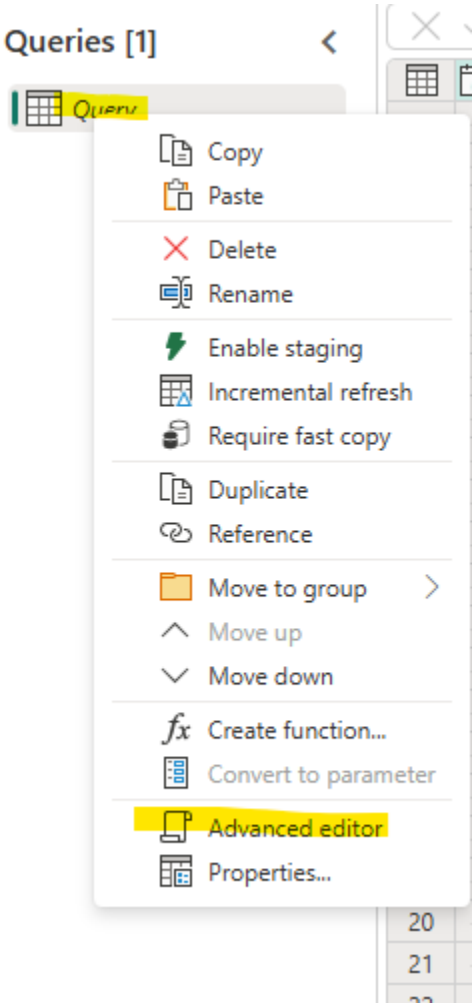
This will show how may rows were loaded for each year. You can check this against your original data source to ensure they match.

Messages | Results

	year	row_count
1	2020	217653591
2	2021	291859897
3	2022	562063094
4	2023	556649873
5	2024	377091818
6	2025	120353458

You will need to modify the query for each chunk of data you load.

To edit an existing query, Open the Dataflow Gen 2 and right click on the Query. Select advanced editor.



Setting up Incremental Refresh

1. After your initial data load, you will need to change the Power Query by commenting out these two lines.

```
//RangeStart = #datetime(2025,1, 1, 0, 0, 0),
//RangeEnd = #datetime(2025, 4, 18, 0, 0, 0),
```

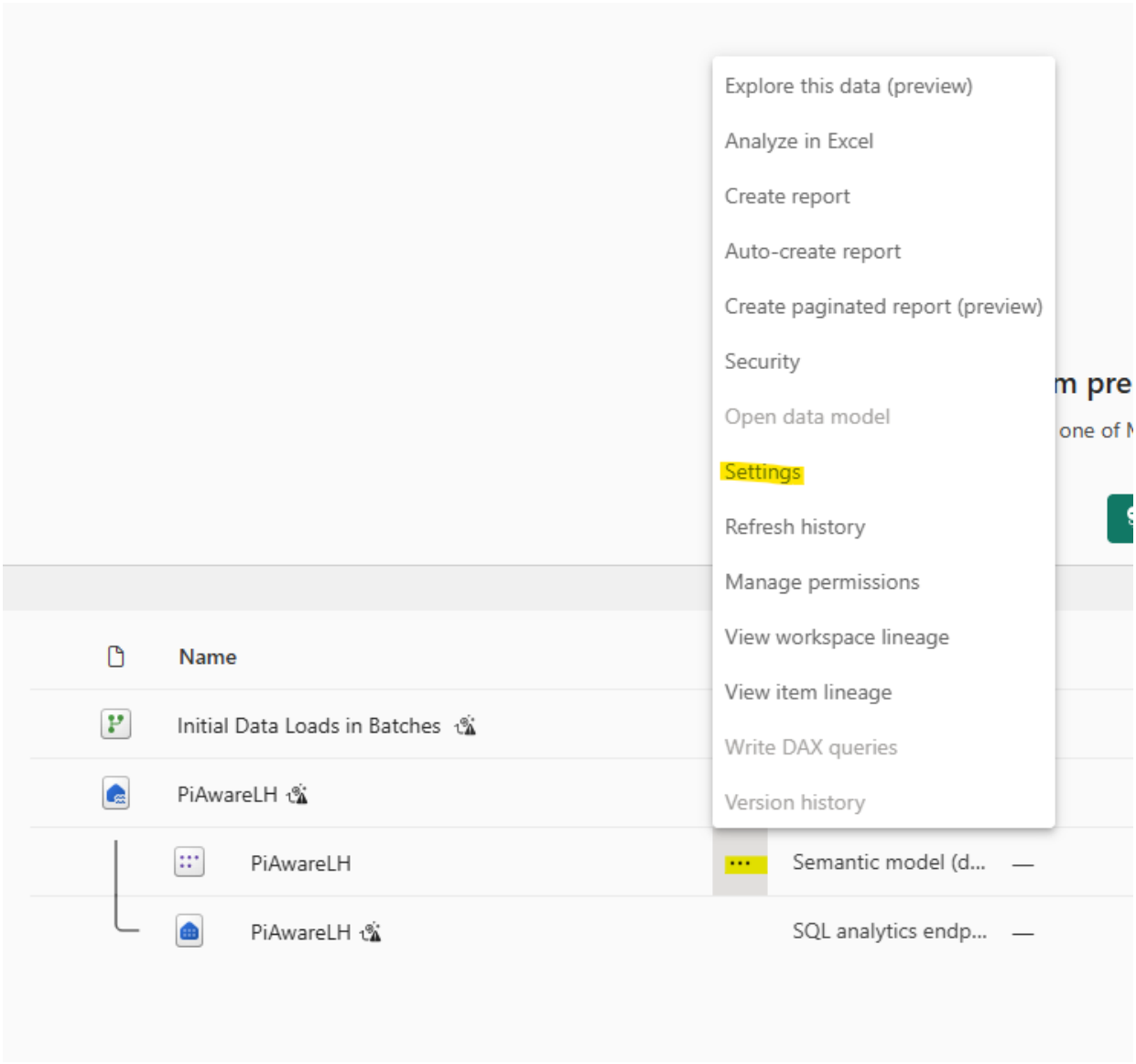
And uncommenting these two lines

```
RangeEnd = DateTime.LocalNow(),
RangeStart = Date.AddDays(RangeEnd, -1),
```

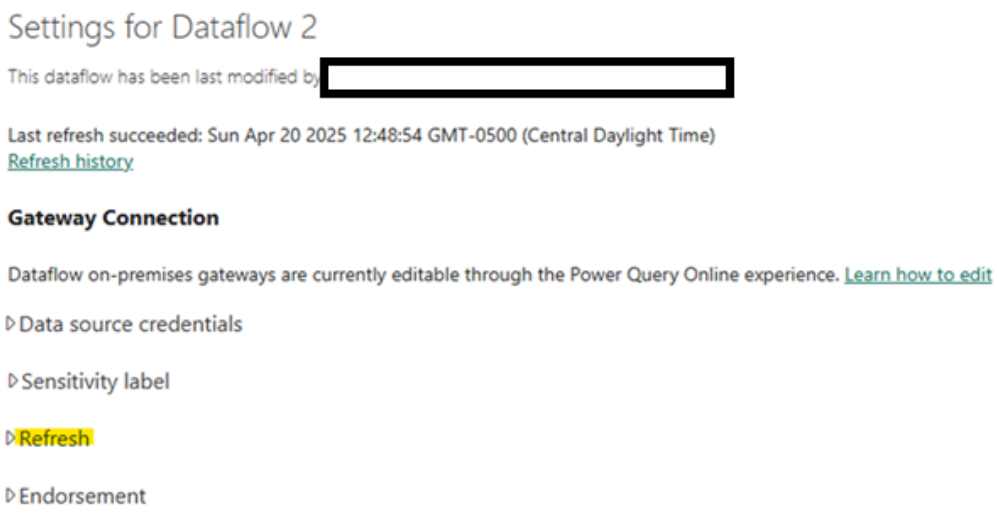
Your query should now look like this:

```
let
	// Define parameters

	RangeEnd = DateTime.LocalNow(),
	RangeStart = Date.AddDays(RangeEnd, -1),

	//RangeStart = #datetime(2025,1, 1, 0, 0, 0),
	//RangeEnd = #datetime(2025, 4, 18, 0, 0, 0),

	// Source data
	Source = Sql.Database("db", "piawaredb"),
	Table = Source{[Schema="dbo", Item="KDFW"]}[Data],

	// Filter data using parameters
	FilteredData = Table.SelectRows(Table, each [dt] >= RangeStart and [dt] <
RangeEnd)
in
	FilteredData
```

2. Click on the three dots next to the Semantic model and click on settings.



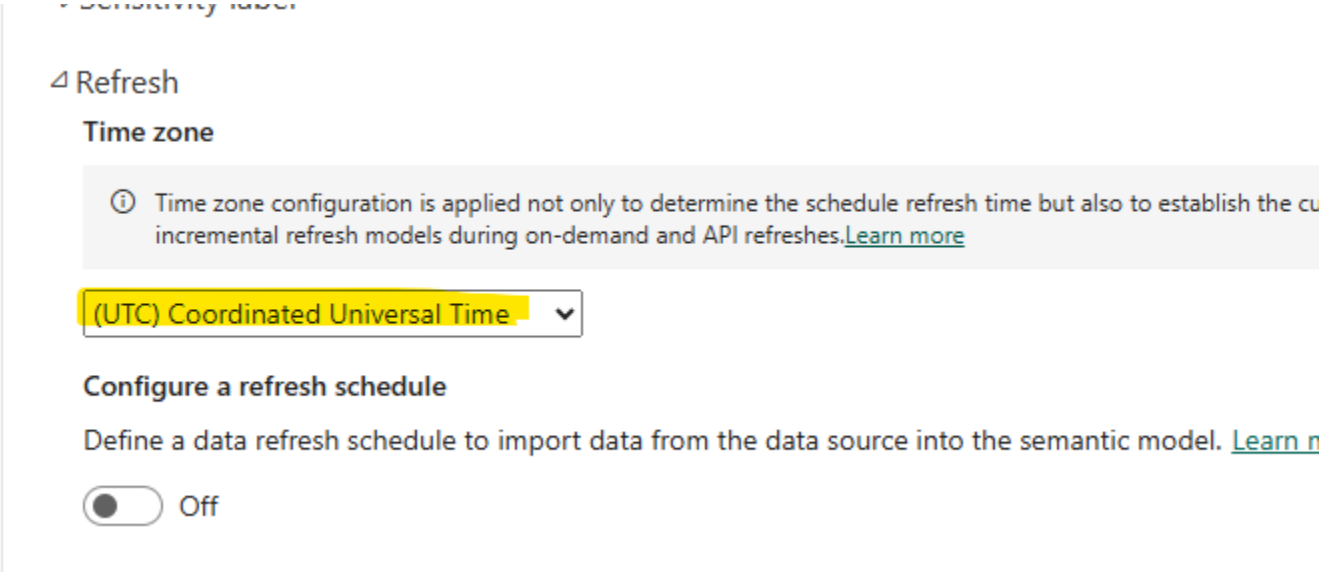
3. You will see refresh as one of your options under settings.



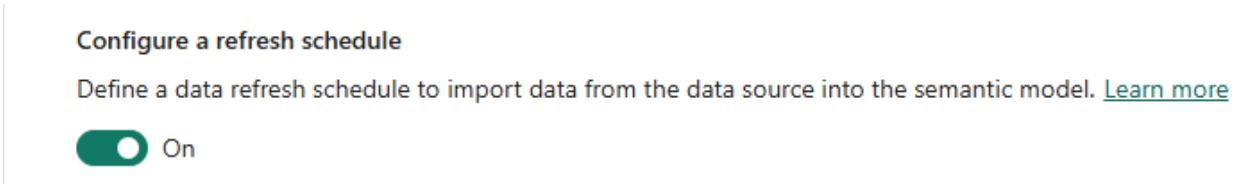
Select a TimeZone

Click on Refresh and set your time zone.

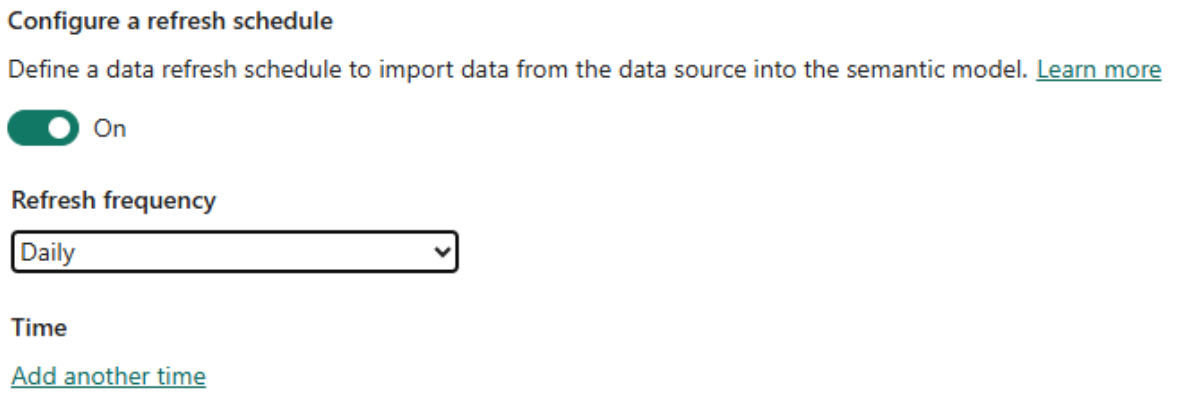
All Times in Azure including Fabric are UTC by default. Make sure to set the time zone of your refresh to match the time zone of your data.



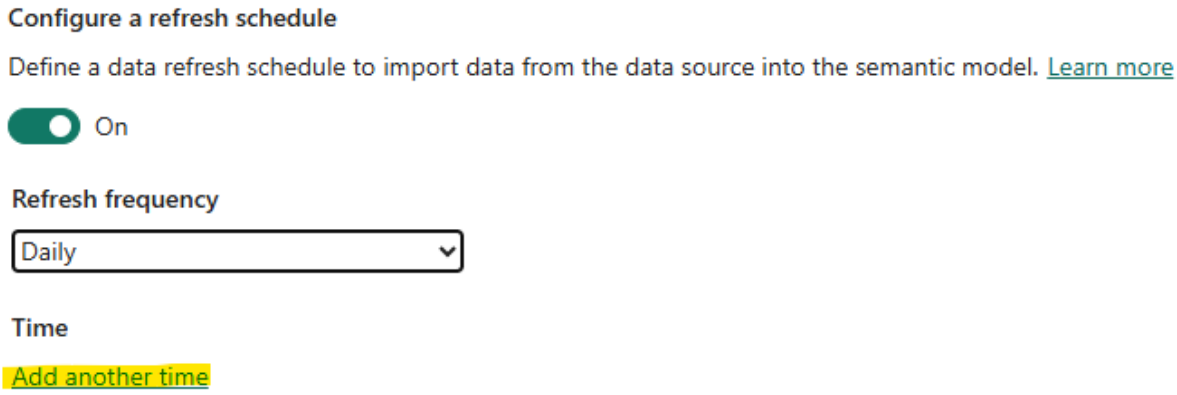
Turn the Define a refresh schedule to on:



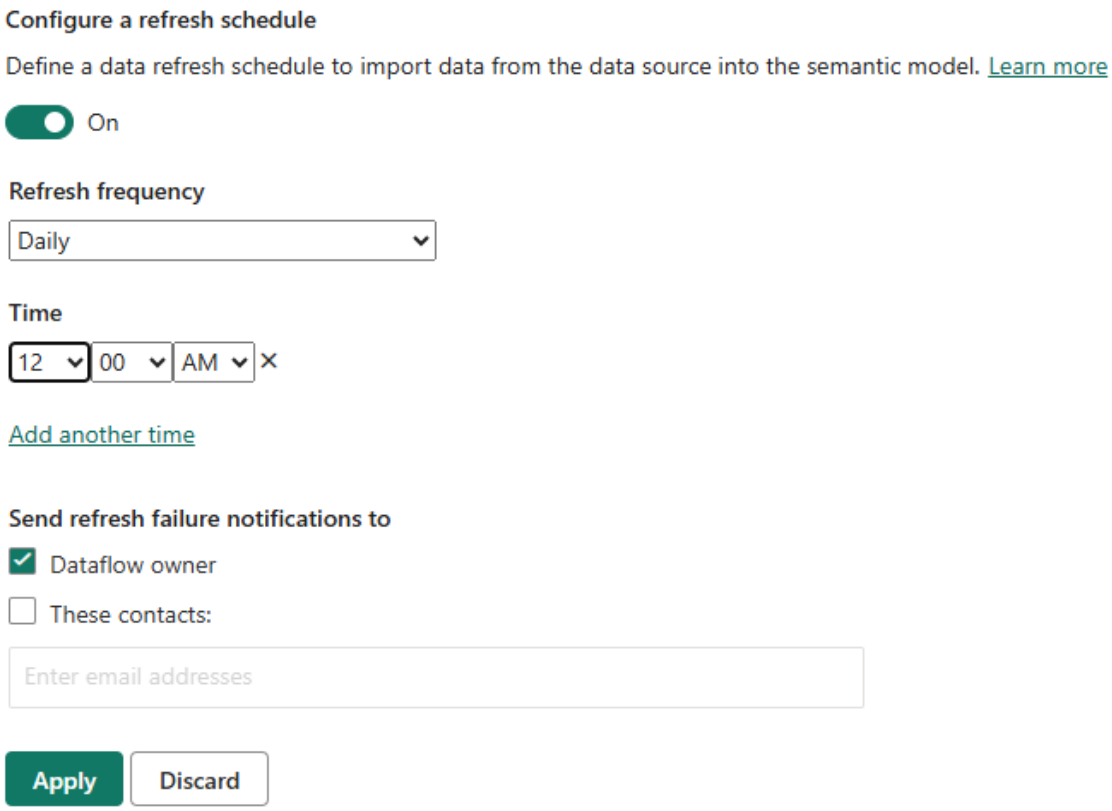
Select a frequency, your choices are Daily and Weekly



Click on Add another time



Select the time of day you would like to refresh the data. For Example: If you are in Central Standard Time zone and your data time stamp is in Central Standard time and you want to pull yesterday's data select 12:00am.



Once you have configured this schedule Fabric will automatically run the dataflow once a day at the time you choose. You can get more granular than once a day. In Fabric, you can schedule up to 48 refreshes per day and the refresh frequency can be every 30 minutes. Just remember to adjust these two lines in your query to match your refresh frequency.

```
RangeEnd = DateTime.LocalNow(),
RangeStart = Date.AddDays(RangeEnd, -1),
```

For example: To pull back just thirty minutes you would change the function from AddDays to AddMinutes like this:

RangeStart = DateTime.AddMinutes(RangeEnd, -30),

Note: M code stops processing statements when it hits a line without a comma ‘,’ at the end of the line. Since there are more statements after this line, make sure to include the comma or you will get an error, or worse the query will save but will not behave as you expected.

That’s it, you are done, you can start building reports from the semantic model and they will have all historical data and current data will be refreshed based on your refresh schedule and query.