



Incremental refresh Power BI

19-12-2024

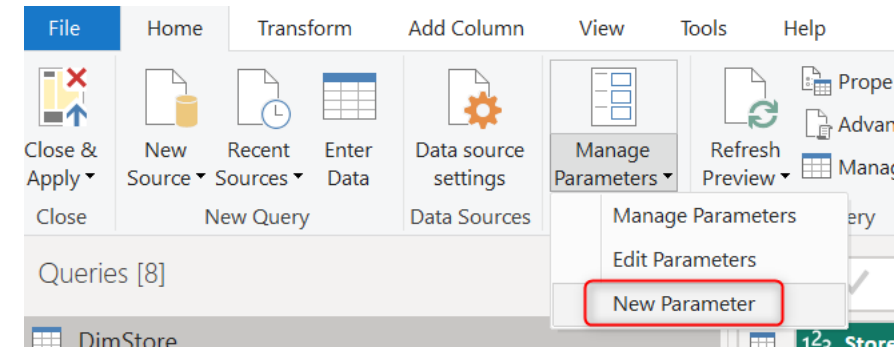


Incremental refresh Power BI

Incremental
refresh

Instead of loading all the data each time you refresh, you can use incremental refresh.

Start by adding two parameters in Power Query.



Parameters

Incremental
refresh

The first one is called RangeStart and the second RangeEnd.

Note: Power Query is case-sensitive.

Select 'Type' and enter 'Date/Time'.

Manage Parameters

Parameters List:

- RangeStart (Selected)
- RangeEnd

RangeStart Configuration:

- Name: RangeStart
- Description:
- ☒ Required
- Type: Date/Time
- Suggested Values: Any value
- Current Value: 1-1-2007 00:00:00

Buttons: OK, Cancel

Queries window

Incremental
refresh

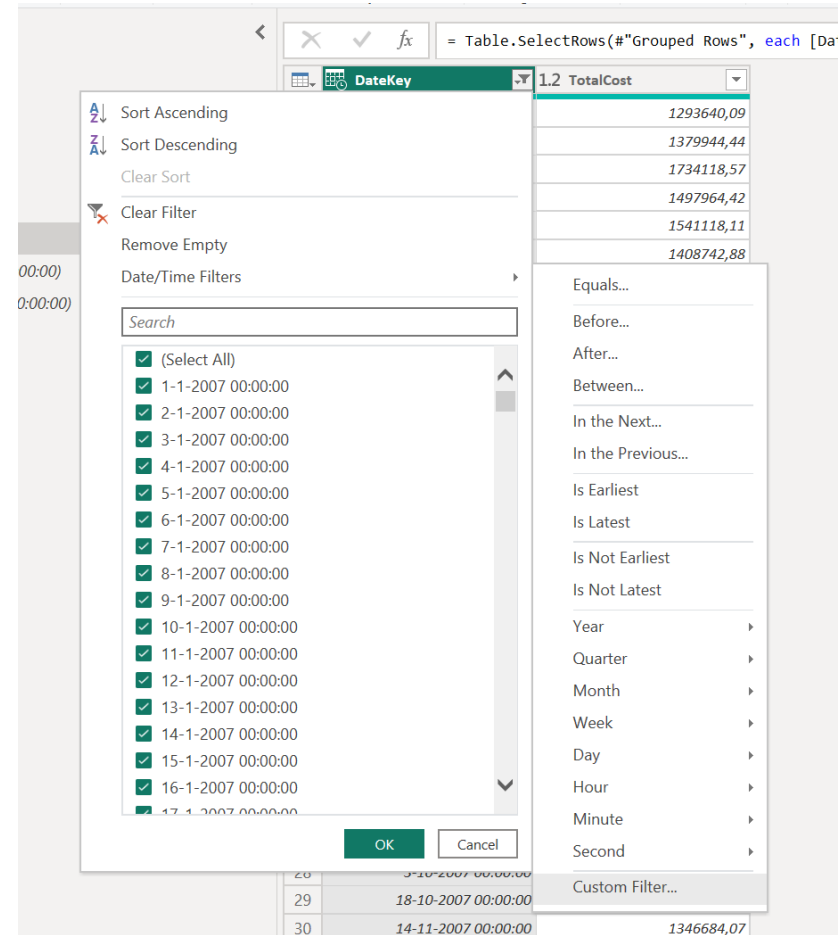
The parameters are in the Queries window.



Customer filter

Incremental
refresh

For the table where you will add incremental refresh, set a custom filter on the date/time column.

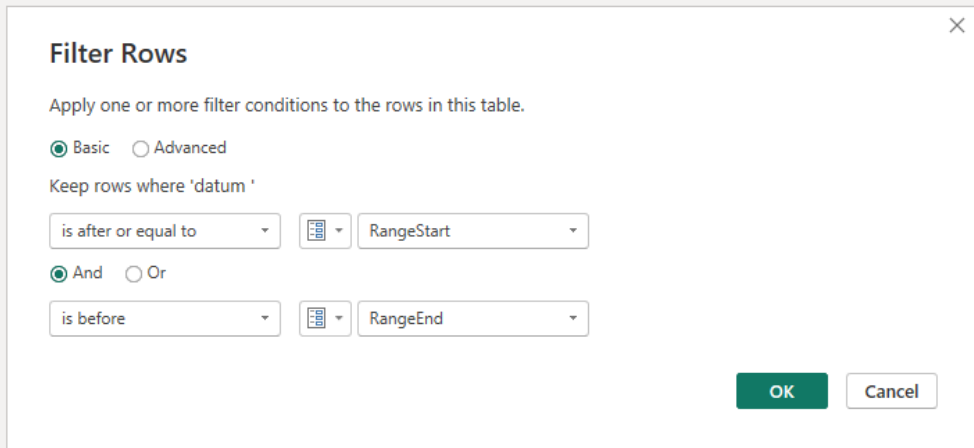


Customer filter

Incremental
refresh

Select the parameters and use 'is after or equal to' for the start date and 'is before' for the end date.

The advantage is that if you have a large dataset, you can use the filter to select, for example, a single day. This allows you to perform operations on the dataset and keep the loading into Power BI Desktop small, making it easier to work with.

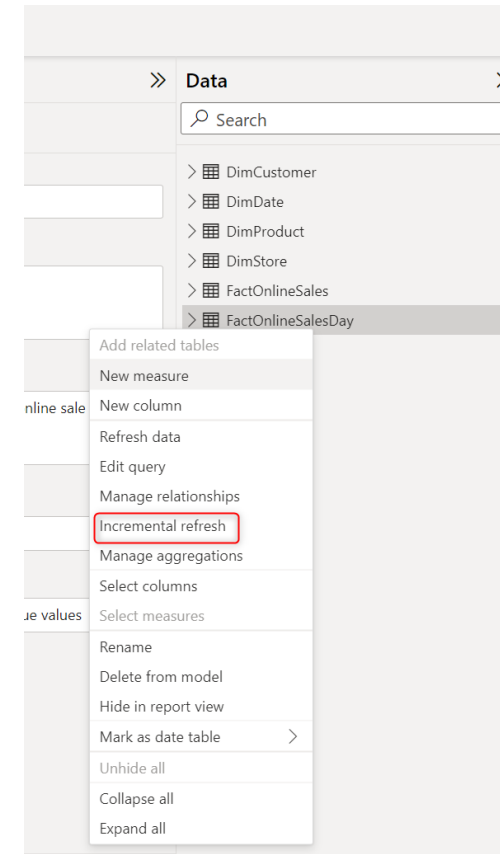


The screenshot shows the 'Filter Rows' dialog box in Power BI Desktop. The dialog has a title bar with a close button (X). Inside, it says 'Filter Rows' and 'Apply one or more filter conditions to the rows in this table.' There are two radio buttons: 'Basic' (selected) and 'Advanced'. Below this, it says 'Keep rows where 'datum '''. There are two filter conditions defined: the first is 'is after or equal to' with a calendar icon and 'RangeStart' as the field; the second is 'is before' with a calendar icon and 'RangeEnd' as the field. Between the two conditions are radio buttons for 'And' (selected) and 'Or'. At the bottom right are 'OK' and 'Cancel' buttons.

Incremental refresh

Incremental
refresh

When you choose Close and Apply to prepare the dataset for Power BI, you can select Incremental Refresh for the desired table.



Incremental refresh

Incremental
refresh

Select the period you want to include in your data model. Also, choose the archive period and refresh period for the data you want to update.

Incremental refresh and real-time data

These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

1. Select table

Date_Basic

2. Set import and refresh ranges

☒ Incrementally refresh this table

Archive data starting 10 Days before refresh date

Data imported from 28/06/2024 to 07/07/2024 (inclusive)

Incrementally refresh data starting 1 Days before refresh date

Data will be incrementally refreshed from 08/07/2024 to 08/07/2024 (inclusive)

3. Choose optional settings

☐ Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

☒ Only refresh complete day [Learn more](#)

☐ Detect data changes [Learn more](#)

4. Review and apply

Archived Incremental Refresh

10 days before refresh date 1 day before refresh date Refresh date

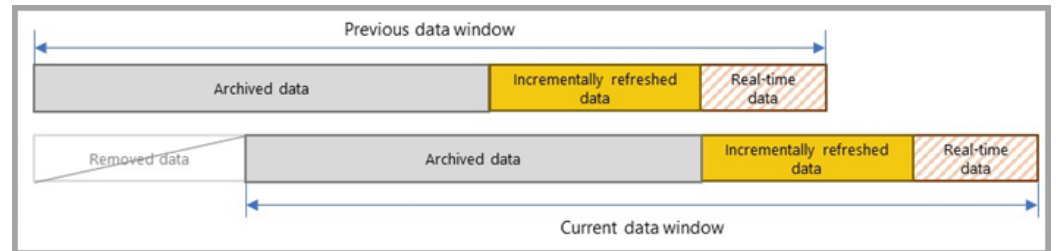
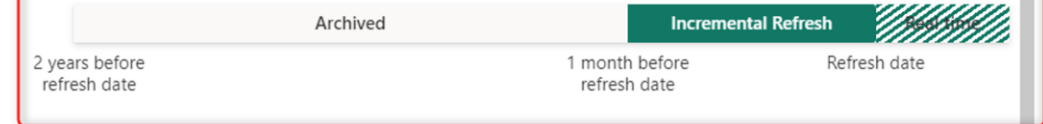
Apply Cancel

Setting up partitions.

Incremental
refresh

The parties work with a sliding window.

4. Review and apply

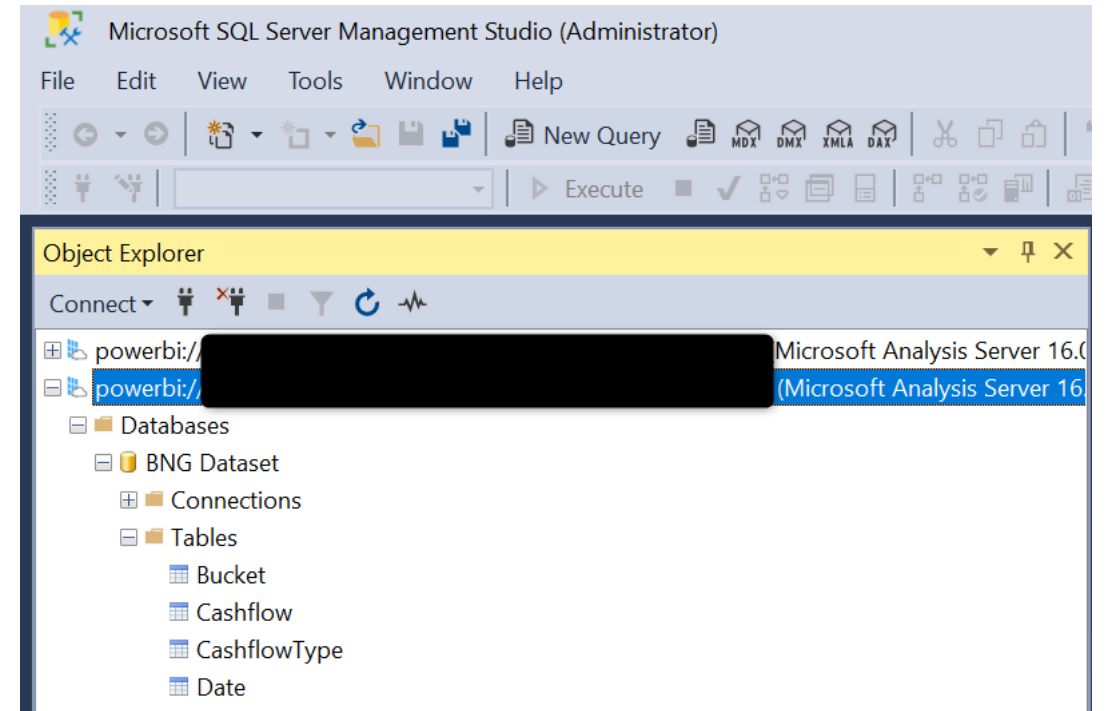


Refreshing partitions via SSMS

Incremental
refresh

We connect to the semantic model via SQL Server Management Studio(SSMS).

Here, we see the tables that are in the semantic model.

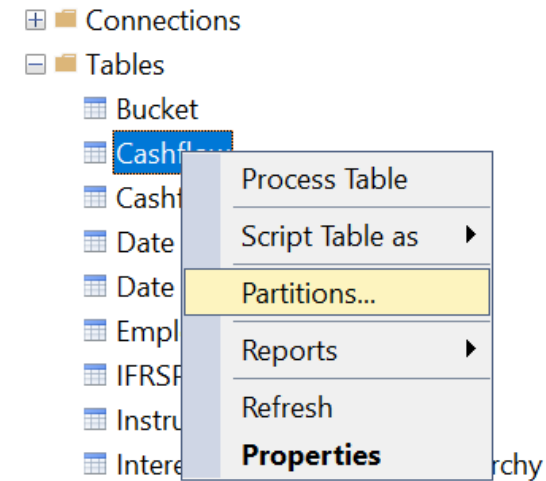


Refreshing partitions via SSMS

Incremental
refresh

If we right-click on a table, we see the option 'Partitions'.

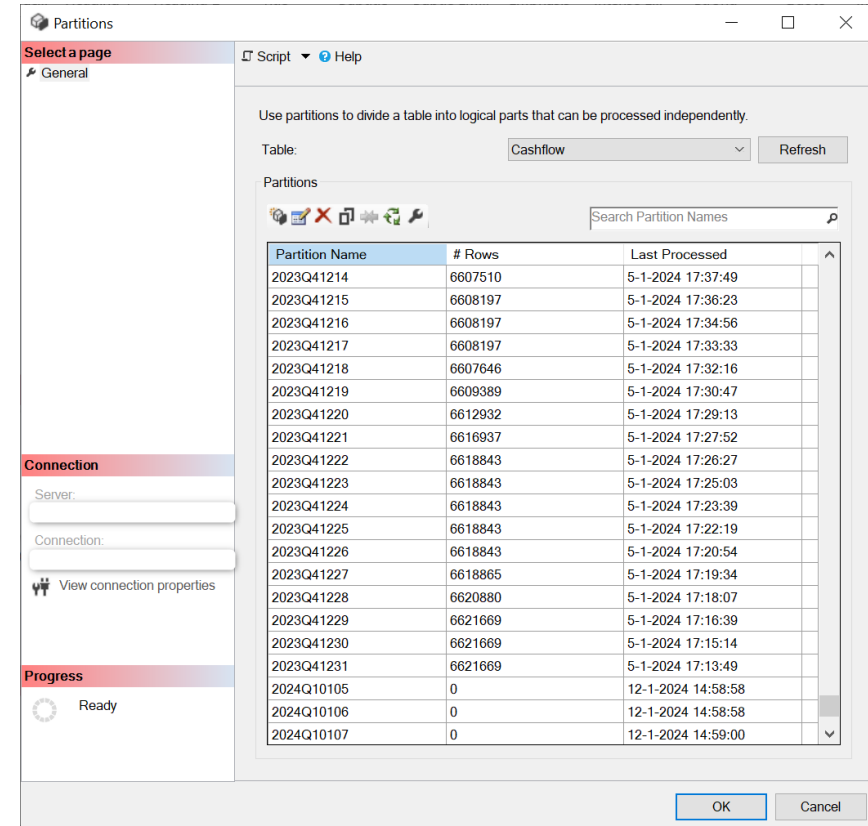
We select this option.



Refreshing partitions via SSMS

Incremental
refresh

Now we see an overview of the existing partitions, including the number of rows and the date of the last refresh.

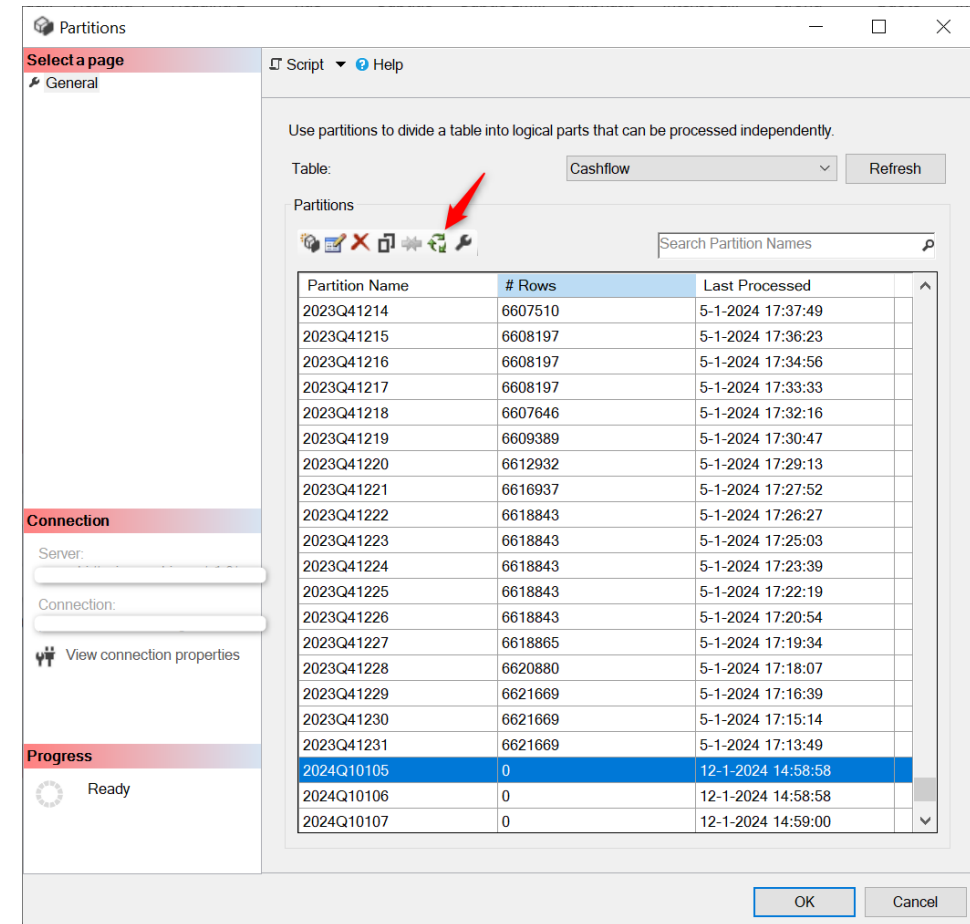


Refreshing partitions via SSMS

Incremental
refresh

We can refresh partitions by clicking on the refresh icon.

We have now selected partition 2024Q10105

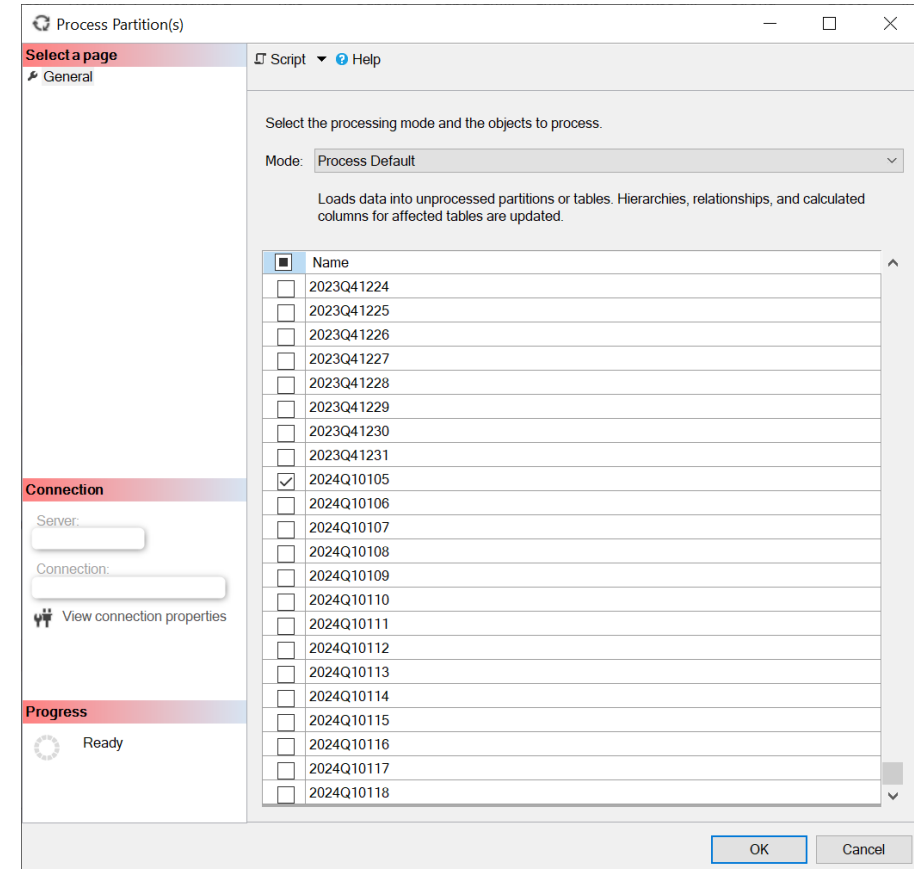


Refreshing partitions via SSMS

Incremental
refresh

Because we had selected partition 2024Q10105, it is now also selected.

We are free to select multiple partitions.



Refreshing partitions via SSMS

Incremental
refresh

Select the processing mode and the objects to process.

Mode: Process Default

- Process Default
- Process Full
- Process Data
- Process Clear
- Process Add

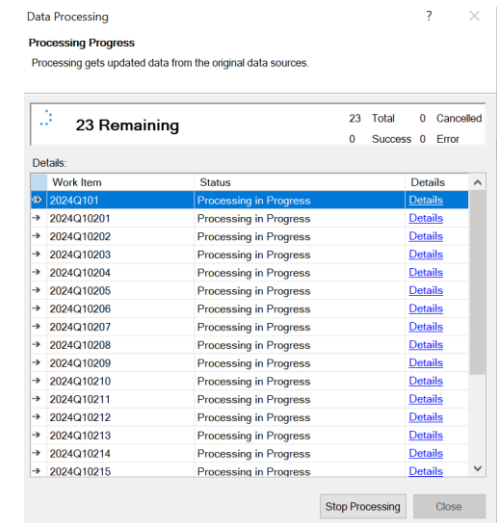
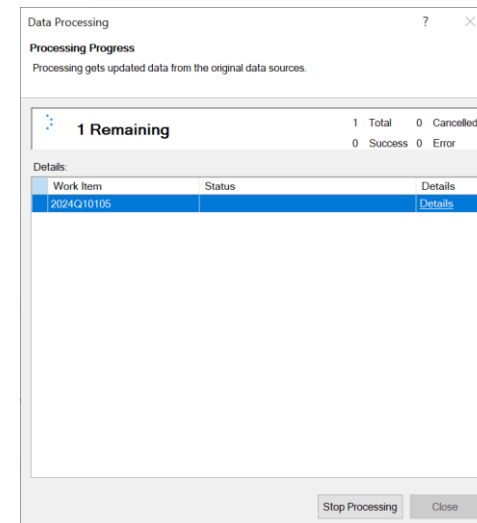
<input checked="" type="checkbox"/>	Name
<input type="checkbox"/>	2023Q41224

Refreshing partitions via SSMS

Incremental
refresh

At the moment we initiate the refresh, we see the following screen.

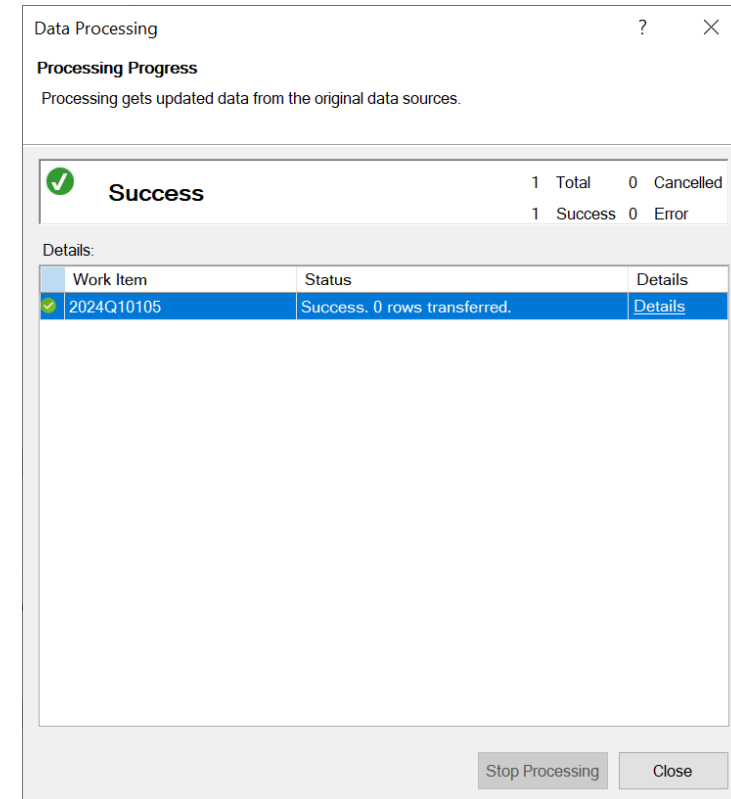
We won't see any changes here until the partitions have been refreshed.



Refreshing partitions via SSMS

Incremental
refresh

The refresh is complete.

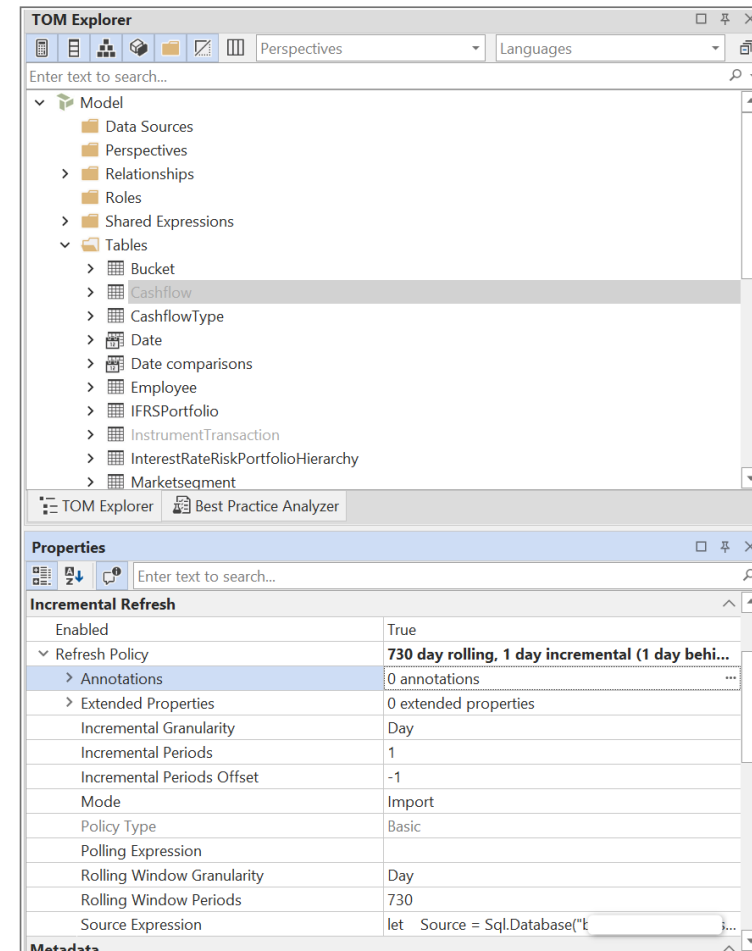


Refreshing partitions via Tabular Editor 3

Incremental
refresh

We can use Tabular Editor 3 for refreshing, but we can also manage the model with it, including controlling and adjusting the Incremental refresh where necessary.

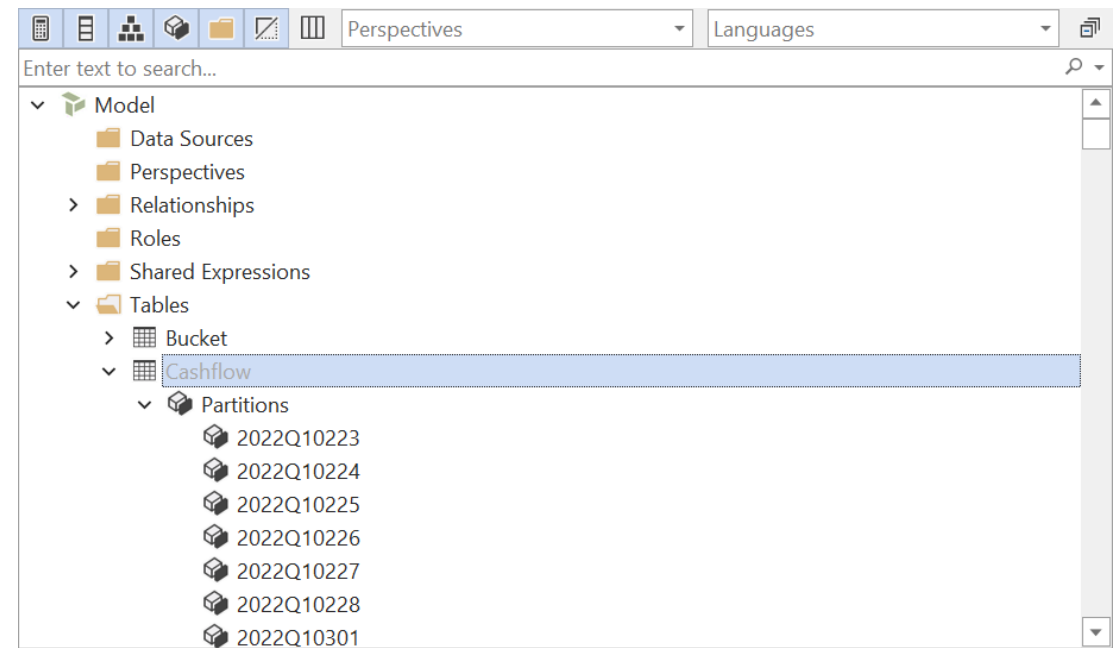
When we select a table, we can view how the incremental refresh is configured in the properties.



Refreshing partitions via Tabular Editor 3

Incremental
refresh

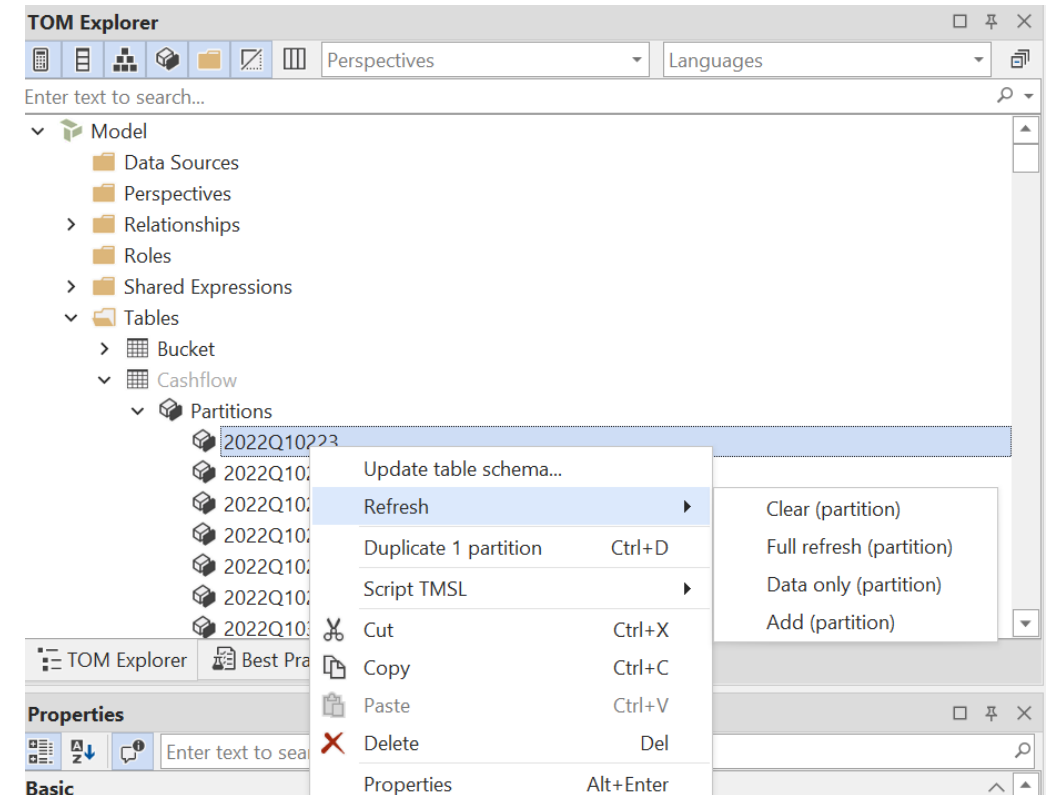
When we open the Partitions for the table, we can see the partitions listed.



Refreshing partitions via Tabular Editor 3

Incremental
refresh

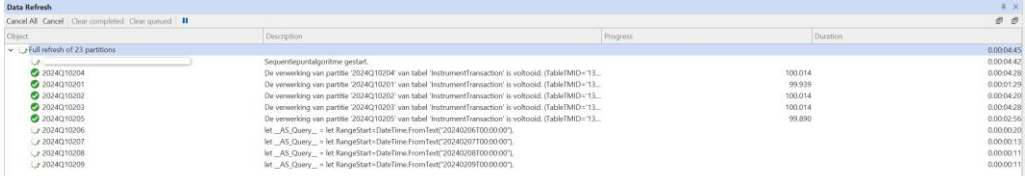
By right-clicking, we see the option 'Refresh', where we can again choose how we want to refresh.



Refreshing partitions via Tabular Editor 3

Incremental
refresh

The advantage compared to SSMS is that you can see what happens during the refresh. You can observe for each partition that rows are being loaded, and once it's done, it shows completion.



The screenshot shows the 'Data Refresh' window in Tabular Editor 3. The window has a title bar with 'Data Refresh' and buttons for 'Cancel All', 'Cancel', 'Clear completed', and 'Clear queued'. Below the title bar is a table with columns 'Object', 'Description', 'Progress', and 'Duration'. The table lists 23 partitions, each with a green checkmark icon in the 'Object' column, indicating successful completion. The 'Description' column shows the specific refresh operation for each partition, including the table name and the range of rows being loaded. The 'Progress' column shows the percentage of completion for each partition, and the 'Duration' column shows the time taken for each refresh operation.

Object	Description	Progress	Duration
Full refresh of 23 partitions	Sequentiaal algoritme gestart.		0.000445
2024Q10204	De verwerking van partitie '2024Q10204' van tabel 'InstrumentTransaction' is voltooid. (TablaTMD-13...	100.014	0.000442
2024Q10201	De verwerking van partitie '2024Q10201' van tabel 'InstrumentTransaction' is voltooid. (TablaTMD-13...	99.939	0.000426
2024Q10202	De verwerking van partitie '2024Q10202' van tabel 'InstrumentTransaction' is voltooid. (TablaTMD-13...	100.014	0.000129
2024Q10203	De verwerking van partitie '2024Q10203' van tabel 'InstrumentTransaction' is voltooid. (TablaTMD-13...	100.014	0.000420
2024Q10205	De verwerking van partitie '2024Q10205' van tabel 'InstrumentTransaction' is voltooid. (TablaTMD-13...	99.890	0.000426
2024Q10206	let _AS_Query_ = let RangeStart=DateTime.FromText("2024Q10206000000")		0.000256
2024Q10207	let _AS_Query_ = let RangeStart=DateTime.FromText("2024Q10207000000")		0.000200
2024Q10208	let _AS_Query_ = let RangeStart=DateTime.FromText("2024Q10208000000")		0.000113
2024Q10209	let _AS_Query_ = let RangeStart=DateTime.FromText("2024Q10209000000")		0.000111



**If you have
questions, please
contact me.**



Peter van den Bos
Business Intelligence Consultant

✉ peter@dutchbigeek.nl

☎ +31 6 13760795

dutchbigeek.nl