

# Incremental refresh Power BI

Peter van den Bos

9-3-2024



# Probleemstelling BNG

Hieronder staan drie tabellen met aantallen die steeds groter worden door de dagelijkse transacties. Dit resulteert tot een grote load als we alles gaan laden per dag.

	Tabel1	Tabel2	Tabel3
Dag 1	100.000	1.000.000	10.000.000
Dag 2	200.000	2.000.000	20.000.000
Dag 3	300.000	3.000.000	30.000.000
Dag 4	400.000	5.000.000	50.000.000
Dag 26	2.600.000	196.418.000.000	1.964.180.000.000
Dag 27	2.700.000	317.811.000.000	3.178.110.000.000
Dag 28	2.800.000	514.229.000.000	5.142.290.000.000
Dag 29	2.900.000	832.040.000.000	8.320.400.000.000
Dag 30	3.000.000	1.346.269.000.000	13.462.690.000.000
Dag 31	3.100.000	2.178.309.000.000	21.783.090.000.000
Jaar 2	6.200.000	4.356.618.000.000	43.566.180.000.000
Jaar 3	9.300.000	6.534.927.000.000	65.349.270.000.000

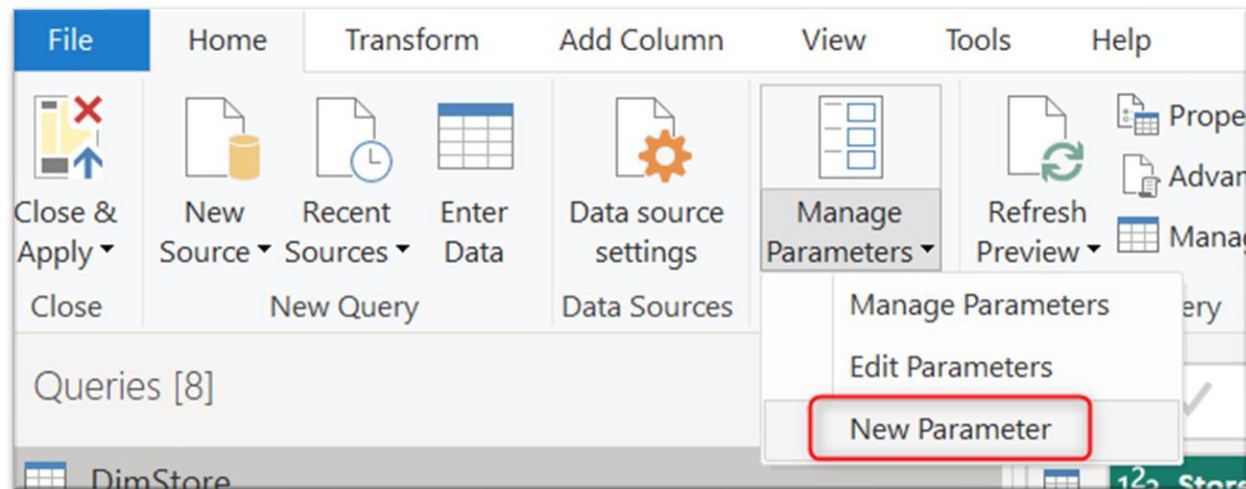


# Incremental refresh Power BI

In plaats van de totale data elke keer bij een refresh te laden, kun je ook incremental refresh gebruiken.

Start hierbij met het toevoegen van twee parameters in Power Query.

Incremental  
refresh

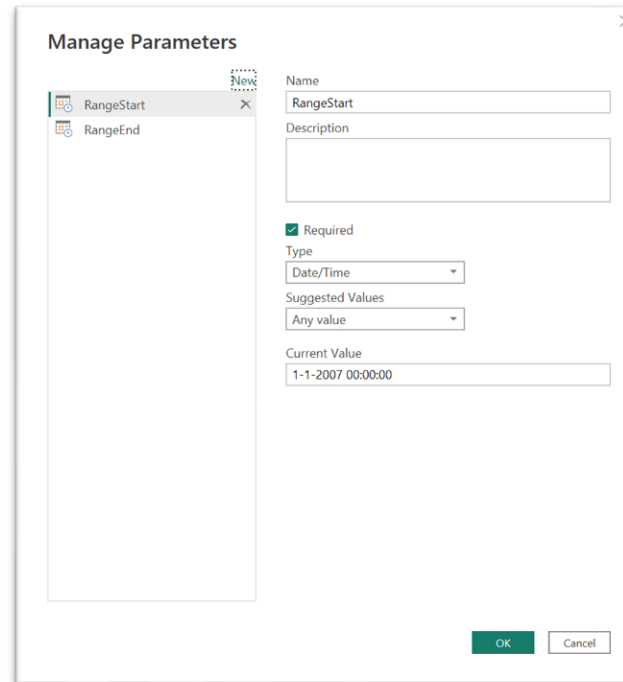


# Parameters

De eerste heet RangeStart en de tweede RangeEnd. Let op: Power Query is hoofdlettergevoelig.

Selecteer 'Type' en vul 'Current Value' in.

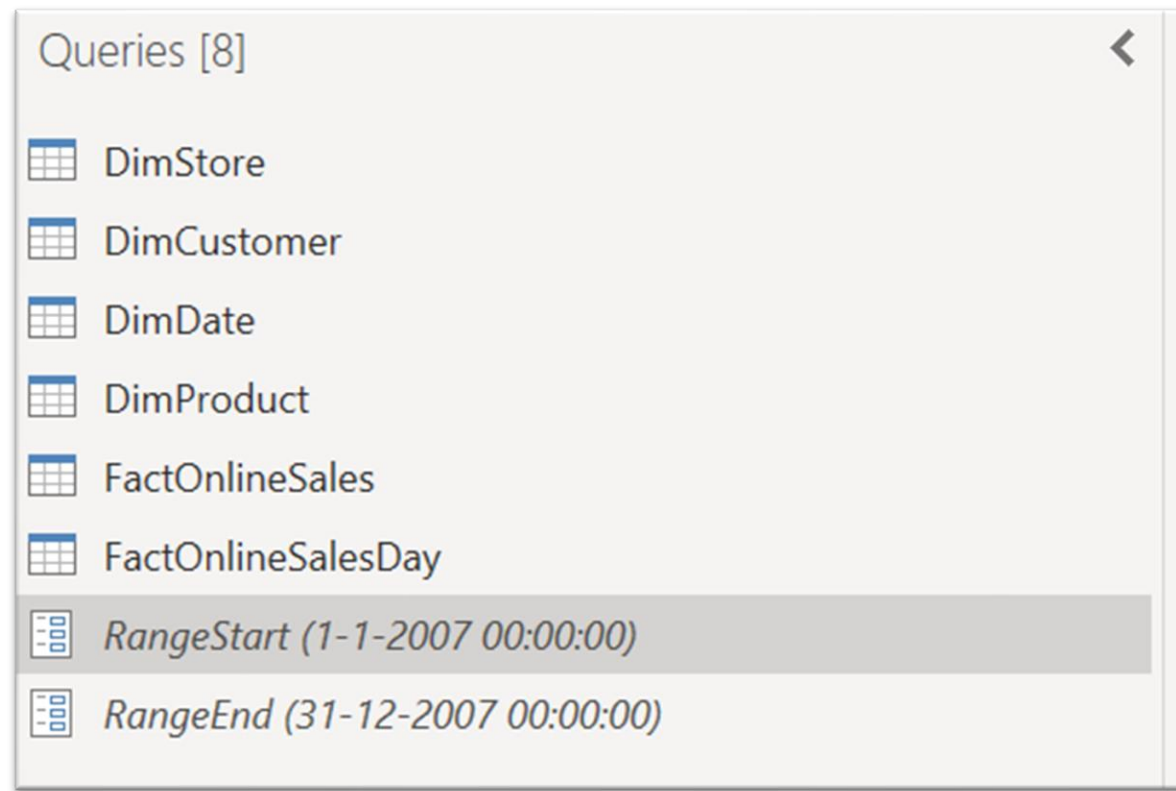
Incremental  
refresh



# Queries window

De parameters staan in Queries window.

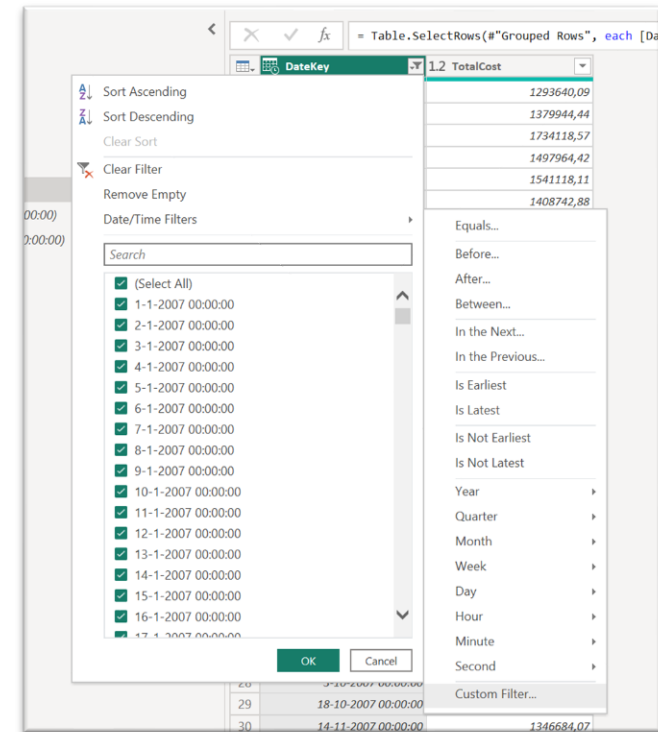
Incremental  
refresh



# Customer filter

Bij het toevoegen van Incremental Refresh aan de tabel stel je een *customerfilter* in op de datumkolom.

Incremental  
refresh

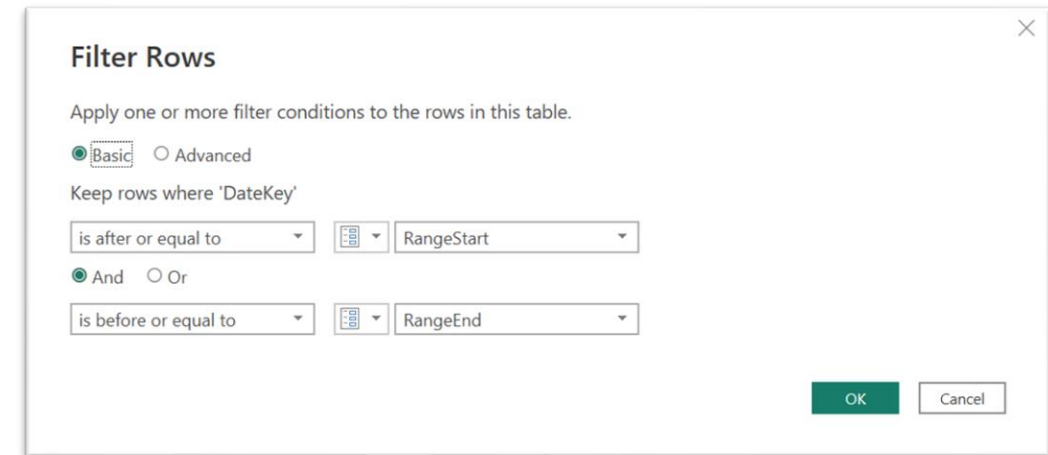


# Customer filter

Selecteer de parameters en gebruik *"is after or equal to"* voor de startdatum en *"is before or equal to"* voor de einddatum.

Het voordeel van het hebben van een grote dataset is dat je, door middel van filters, eenvoudig bijvoorbeeld een selectie van een dag kunt maken. Hierdoor kun je bewerkingen uitvoeren op de dataset en deze vervolgens laden in Power BI Desktop, wat het geheel compact houdt en daardoor ook gemakkelijker maakt om mee te werken.

Incremental  
refresh




The image shows a 'Filter Rows' dialog box from Power BI. It has a title bar with a close button. The main text says 'Apply one or more filter conditions to the rows in this table.' Below this are two radio buttons: 'Basic' (selected) and 'Advanced'. The text 'Keep rows where 'DateKey'' is followed by two filter conditions. The first condition is 'is after or equal to' with a date icon and a dropdown menu set to 'RangeStart'. The second condition is 'is before or equal to' with a date icon and a dropdown menu set to 'RangeEnd'. Between the two conditions are radio buttons for 'And' (selected) and 'Or'. At the bottom right are 'OK' and 'Cancel' buttons.

Filter Rows


Apply one or more filter conditions to the rows in this table.

☒ Basic ☐ Advanced

Keep rows where 'DateKey'

is after or equal to  RangeStart

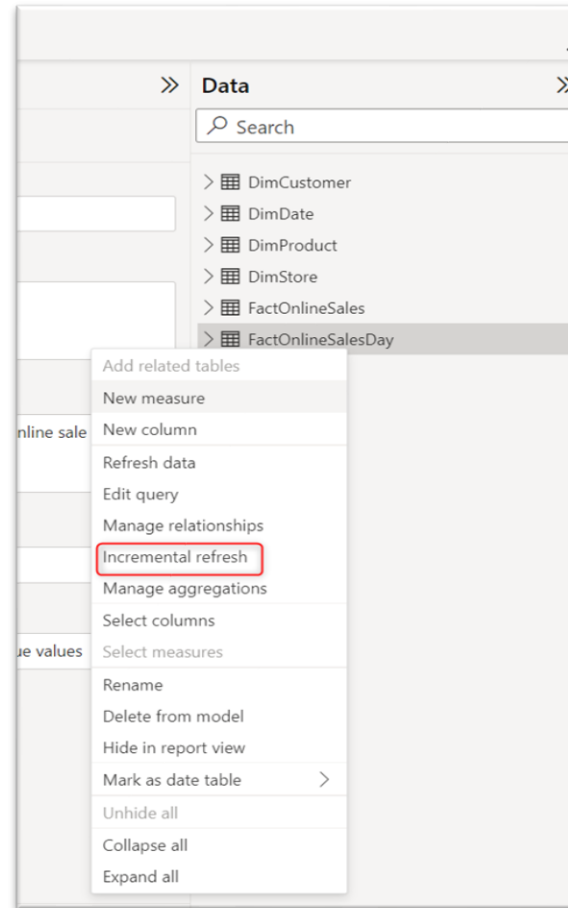
☒ And ☐ Or

is before or equal to  RangeEnd

OK Cancel

# Incremental refresh

Als je Close en Apply kiest om de dataset in orde te maken voor Power BI, kun je voor de gewenste tabel kiezen voor Incremental refresh.



Incremental  
refresh

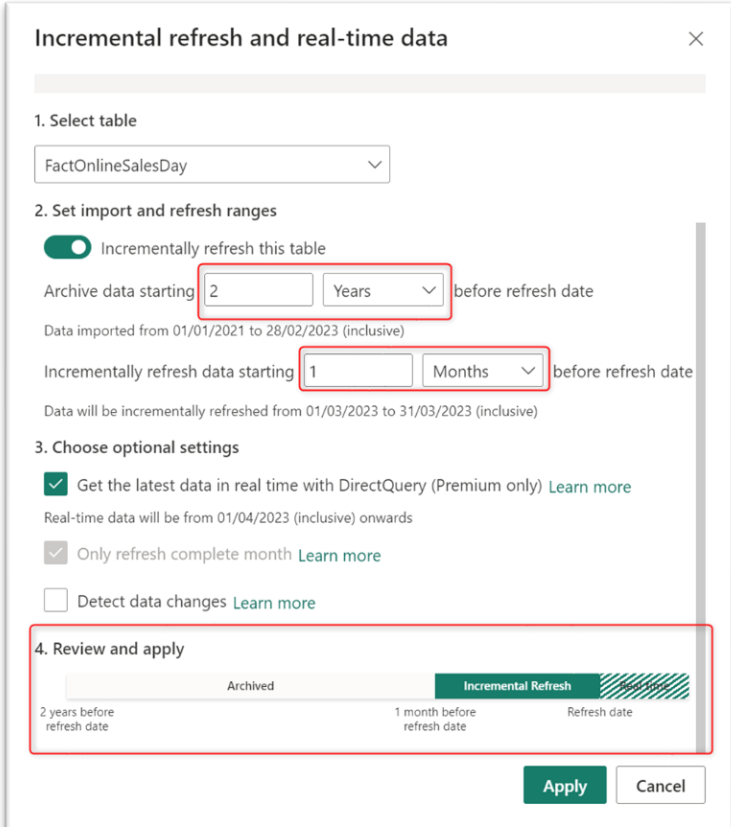


# Incremental refresh

Selecteer de periode die je in je datamodel wilt hebben.

Selecteer ook de periode van het deel van de data die je wilt refreshen.

Incremental  
refresh



**Incremental refresh and real-time data**

1. Select table  
FactOnlineSalesDay

2. Set import and refresh ranges  
☒ Incrementally refresh this table  
Archive data starting 2 Years before refresh date  
Data imported from 01/01/2021 to 28/02/2023 (inclusive)  
Incrementally refresh data starting 1 Months before refresh date  
Data will be incrementally refreshed from 01/03/2023 to 31/03/2023 (inclusive)

3. Choose optional settings  
☒ Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)  
Real-time data will be from 01/04/2023 (inclusive) onwards  
☒ Only refresh complete month [Learn more](#)  
☐ Detect data changes [Learn more](#)

4. Review and apply

Archived Incremental Refresh Real-time

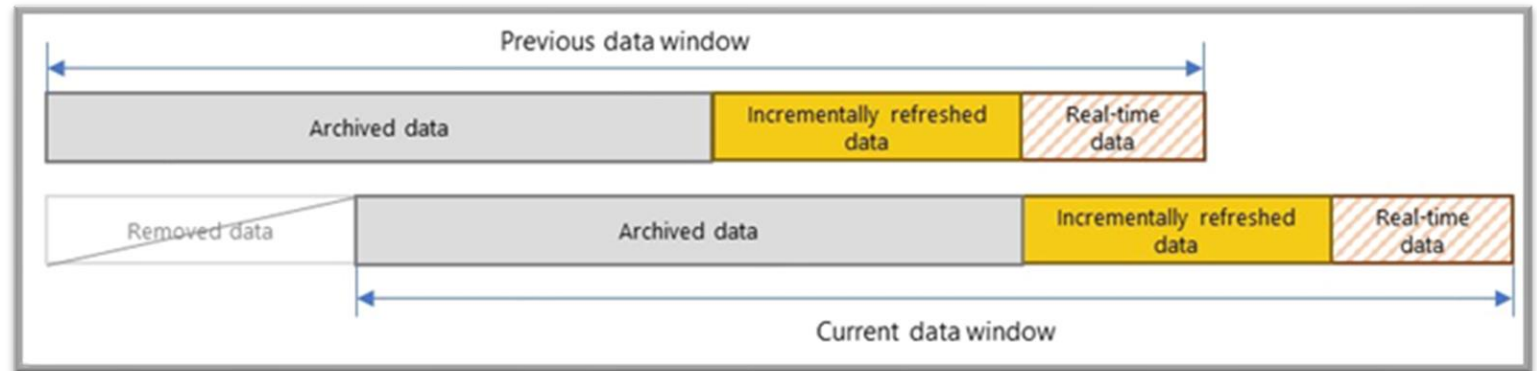
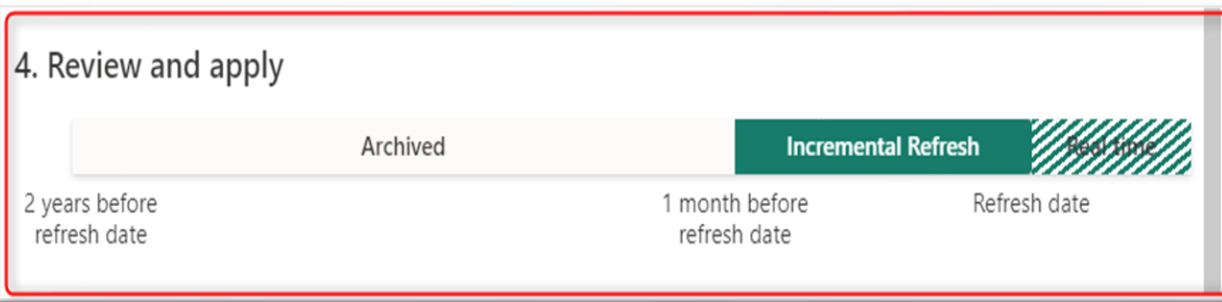
2 years before refresh date 1 month before refresh date Refresh date

Apply Cancel

# Partities opzet

Incremental  
refresh

De partities werken met een verschuivend window.



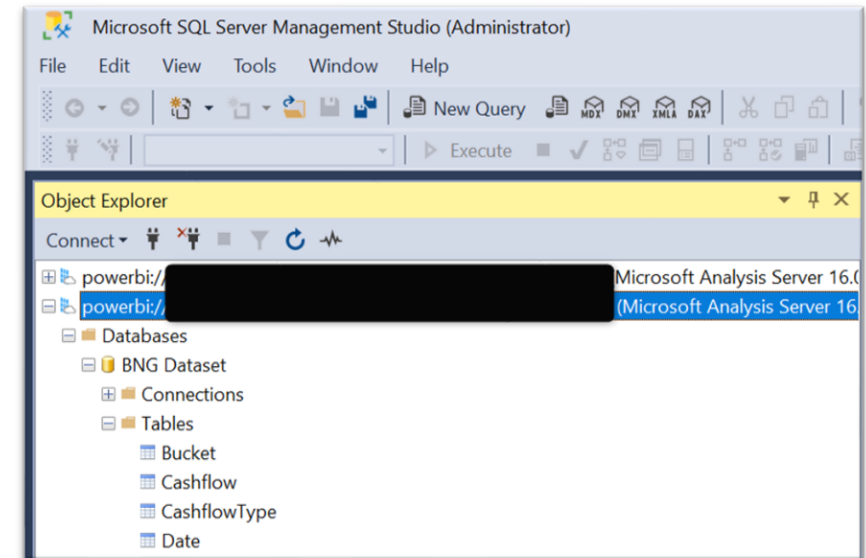
# Partities verversen via SSMS

Incremental  
refresh

We maken verbinding met het semantic model via SQL Server Management Studio.

De volgende modellen zitten in het semantic model:

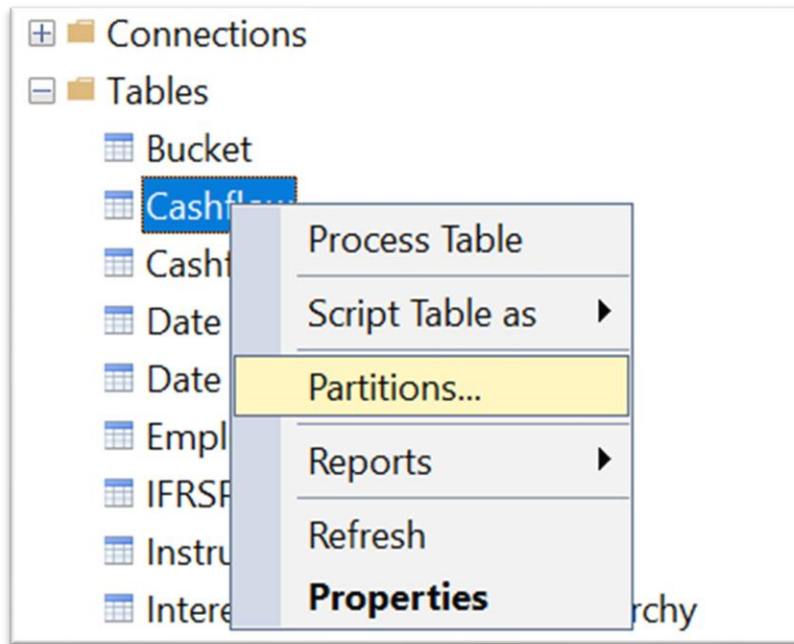
- Bucket
- Cashflow
- Cashflow type
- Date



# Partities verversen via SSMS

Incremental  
refresh

Als we op een tabel rechtermuis toets gebruiken, dan zien we de optie "*Partitions*", deze selecteren we.



# Partities verversen via SSMS

We zien nu een overzicht met de bestaande partities en zien ook het aantal rijen en de datum van de laatste verversing.

Incremental  
refresh

Partitions

Select a page

General

Script Help

Use partitions to divide a table into logical parts that can be processed independently.

Table: Cashflow Refresh

Partitions

Search Partition Names

Partition Name	# Rows	Last Processed
2023Q41214	6607510	5-1-2024 17:37:49
2023Q41215	6608197	5-1-2024 17:36:23
2023Q41216	6608197	5-1-2024 17:34:56
2023Q41217	6608197	5-1-2024 17:33:33
2023Q41218	6607646	5-1-2024 17:32:16
2023Q41219	6609389	5-1-2024 17:30:47
2023Q41220	6612932	5-1-2024 17:29:13
2023Q41221	6616937	5-1-2024 17:27:52
2023Q41222	6618843	5-1-2024 17:26:27
2023Q41223	6618843	5-1-2024 17:25:03
2023Q41224	6618843	5-1-2024 17:23:39
2023Q41225	6618843	5-1-2024 17:22:19
2023Q41226	6618843	5-1-2024 17:20:54
2023Q41227	6618865	5-1-2024 17:19:34
2023Q41228	6620880	5-1-2024 17:18:07
2023Q41229	6621669	5-1-2024 17:16:39
2023Q41230	6621669	5-1-2024 17:15:14
2023Q41231	6621669	5-1-2024 17:13:49
2024Q10105	0	12-1-2024 14:58:58
2024Q10106	0	12-1-2024 14:58:58
2024Q10107	0	12-1-2024 14:59:00

Connection

Server:

Connection:

View connection properties

Progress

Ready

OK Cancel

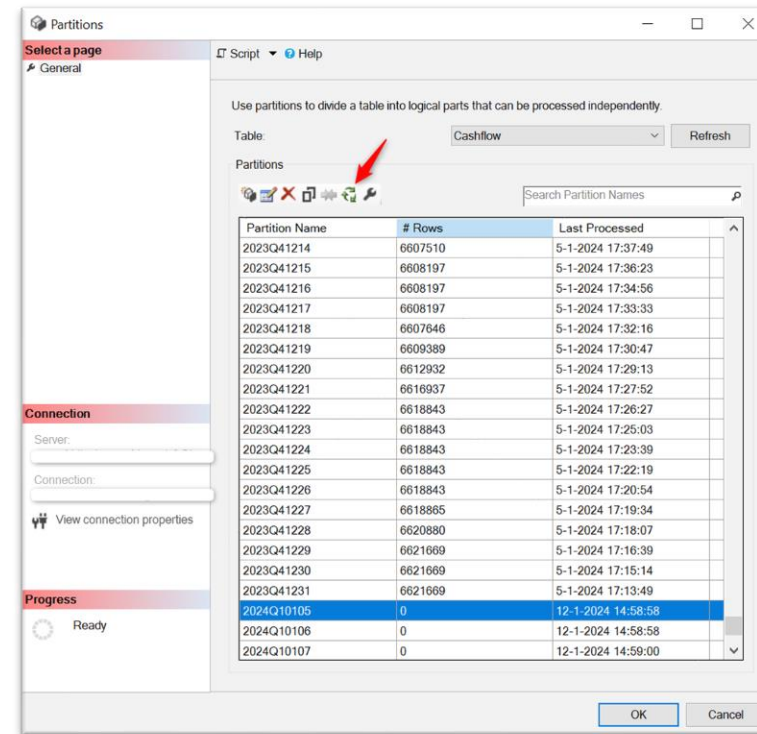


# Partities verversen via SSMS

Incremental  
refresh

We kunnen partities verversen door op het ververs icon te drukken.

We hebben nu partitie 2024Q10105 geselecteerd.

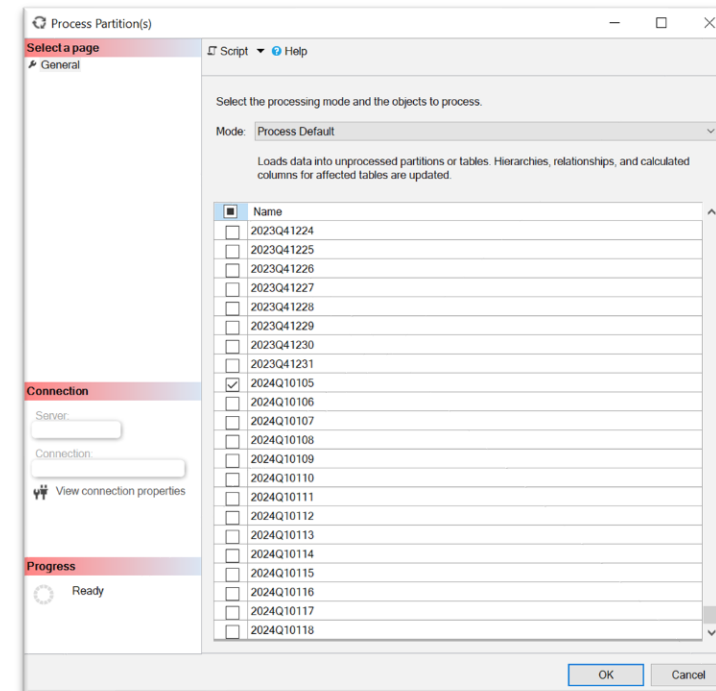


# Partities verversen via SSMS

Incremental  
refresh

Omdat we partitie 2024Q10105 geselecteerd hadden, is deze direct mee geselecteerd.

In deze stap ben je vrij om meerdere partities te selecteren.



# Partities verversen 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

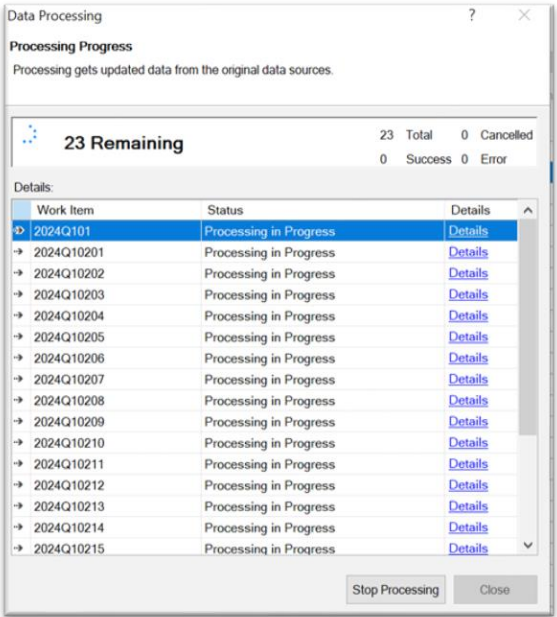
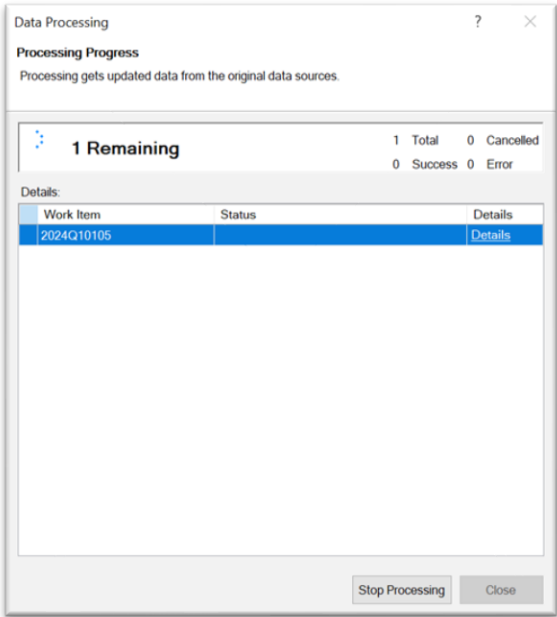


# Partities verversen via SSMS

Incremental refresh

Op het moment van vernieuwen verschijnt het volgende scherm.

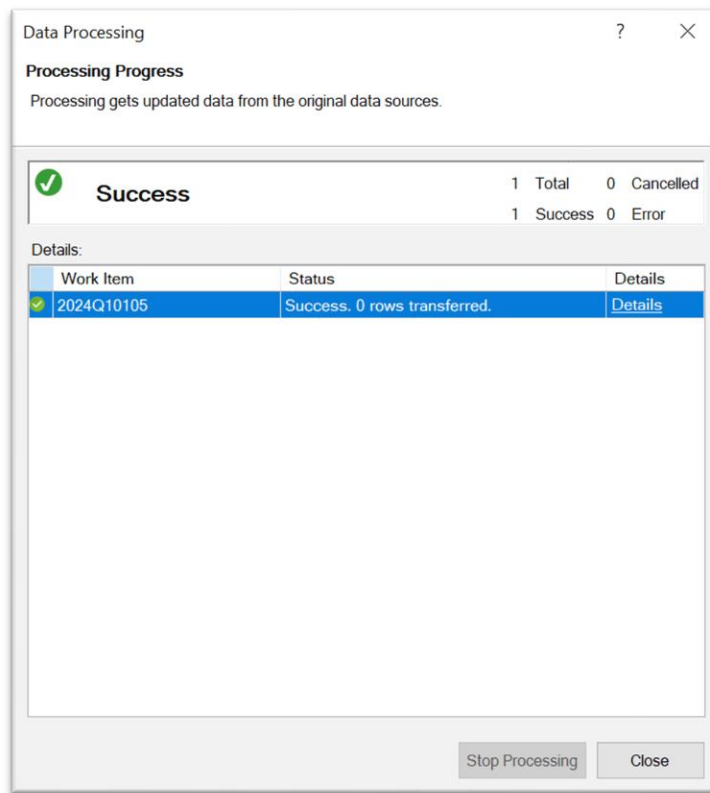
We zien hier geen wijzigingen totdat de partities zijn vernieuwd.



# Partities verversen via SSMS

De verversing is voltooid.

Incremental  
refresh

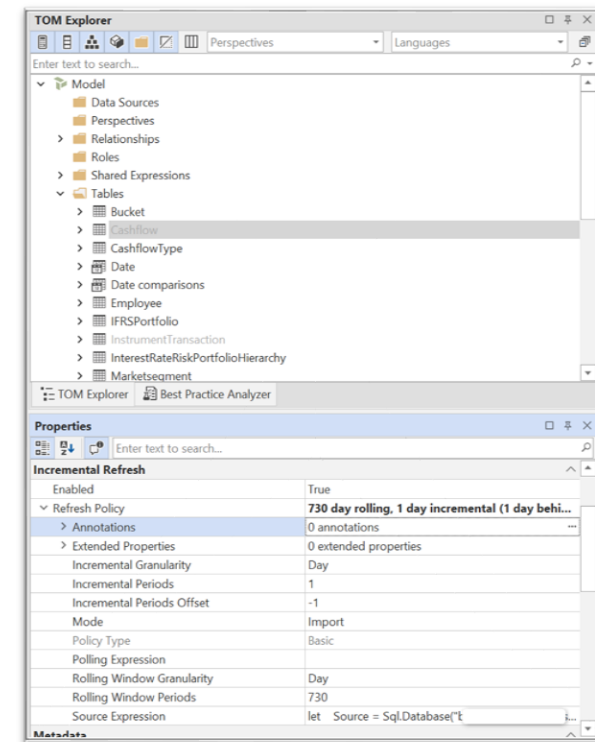


# Partities verversen via Tabular Editor 3

Incremental  
refresh

We kunnen Tabular Editor 3 gebruiken voor het verversen, echter kunnen we hiermee ook het model beheersen en ook de Increment refresh beheersen en aanpassen waar noodzakelijk.

Als we op een tabel gaan staan kunnen we bij de properties zien, hoe de Incremental Refresh is ingeregeld.



# Partities verversen via Tabular Editor 3

Incremental  
refresh

Properties

A

Z

Enter text to search...

Incremental Refresh

Enabled

True

Refresh Policy

730 day rolling, 1 day incremental (1 day behi...

Annotations

0 annotations

Extended Properties

0 extended properties

Incremental Granularity

Day

Incremental Periods

1

Incremental Periods Offset

-1

Mode

Import

Policy Type

Basic

Polling Expression

Rolling Window Granularity

Day

Rolling Window Periods

730

Source Expression

let Source = Sql.Database("...

Metadata

Property Name	Power BI Desktop Equivalent	Description	Expected Value
<b>EnableRefreshPolicy</b>	Incrementally refresh this table	Whether a refresh policy is enabled for the table.  In Tabular Editor, other Refresh Policy properties will only be visible if this value is set to <code>True</code> .	<code>True</code> or <code>False</code> .
<b>IncrementalGranularity</b>	Incremental Refresh Period	The granularity of the incremental window.  Example: "Refresh data in the last 30 days before refresh date."	<code>Day</code> , <code>Month</code> , <code>Quarter</code> or <code>Year</code> . Must be smaller than or equal to the <code>IncrementalGranularity</code> .
<b>IncrementalPeriods</b>	Number of Incremental Refresh Periods	The number of periods for the incremental window.  Example: "Refresh data in the last 30 days before refresh date."	An integer of the number of <code>IncrementalGranularity</code> periods. Must define a total period smaller than the <code>RollingWindowPeriods</code>
<b>IncrementalPeriodsOffset</b>	Only refresh complete days	The offset to be applied to <code>IncrementalPeriods</code> .  Example for: <code>IncrementalPeriodsOffset = -1</code> ; <code>IncrementalPeriods = 30</code> ; <code>IncrementalGranularity = Day</code> ; "Only refresh data in the last 30 days, from the day before refresh date."	An integer of the number of <code>IncrementalGranularity</code> periods to shift the incremental window.
<b>Mode</b>	Get the latest data in real time with DirectQuery	Specifies whether Incremental Refresh is configured with only import partitions or also a DirectQuery partition, to result in a "Hybrid Table".	<code>Import</code> or <code>Hybrid</code> .

# Partities verversen via Tabular Editor 3

Incremental  
refresh

Properties

Enter text to search...

Incremental Refresh

Enabled

True

Refresh Policy

730 day rolling, 1 day incremental (1 day behi...

Annotations

0 annotations

Extended Properties

0 extended properties

Incremental Granularity

Day

Incremental Periods

1

Incremental Periods Offset

-1

Mode

Import

Policy Type

Basic

Polling Expression

Rolling Window Granularity

Day

Rolling Window Periods

730

Source Expression

let Source = Sql.Database("...

Metadata

Property Name	Power BI Desktop Equivalent	Description	Expected Value
<i>PolicyType</i>	N/A	Specifies the type of refresh policy.	Can only contain a single value: <code>Basic</code> .
<i>PollingExpression</i> (Optional)	Detect Data Changes	The M Expression used to detect changes in a specific column such as <code>LastUpdateDate</code> .  In Tabular Editor, the <b>Polling Expression</b> can be viewed and modified from the <b>Expression Editor</b> window by selecting it from the dropdown menu in the top-left.	A valid M Expression that returns a scalar value of the latest date in a column. All records in incremental window hot partitions containing that value for the column will be refreshed.  Records in archived partitions are not refreshed.
<i>RollingWindowGranularity</i>	Archive Data Period	The granularity of the rolling window.  Example: "Archive data starting 3 years before refresh date."	<code>Day</code> , <code>Month</code> , <code>Quarter</code> or <code>Year</code> . Must be larger than or equal to the <code>IncrementalGranularity</code> .
<i>RollingWindowPeriods</i>	Number of Archive Data Periods	The number of periods for the rolling window.  Example: "Archive data starting 3 years before refresh date."	An integer of the number of <code>RollingWindowGranularity</code> periods. Must define a total period larger than the <code>IncrementalPeriods</code>
<i>SourceExpression</i>	Power Query Source Expression	The M Expression for the table data source. This is where the original table M Expression is, and where any existing Power Query transformations would be modified.  In Tabular Editor, the <b>Source Expression</b> can be viewed and modified from the <b>Expression Editor</b> by selecting it from the dropdown menu in the top-left.	A valid M Expression containing a filter step appropriately using <code>RangeStart</code> and <code>RangeEnd</code> .

# Partities verversen via Tabular Editor 3

Incremental  
refresh

FileEditViewExpressionTableModelToolsWindowHelp

expression Editor

Source Expression

on Table 'Cashflow'

```

let
    Source = Sql.Database("...", "dwh", [ConnectionTimeout=#duration(0, 3, 0), CommandTimeout=#duration(0, 3, 0)]),
    bdw_CashFlow = Source[[Schema="bdw",Item="CashFlow"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(bdw_CashFlow,{{"ReportingDate", type datetime}},),
    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each [ReportingDate] >= RangeStart and [ReportingDate] < RangeEnd),
    #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"CashflowId", "SourceSystemCode", "ReportingDate", "TransactionId", "CashflowTypeCode", "LiquidityBucketCode", "CashflowAmountEUR"})
in
    #"Removed Other Columns"
        
```

Model

Data Sources

Perspectives

Relationships

Roles

Shared Expressions

Tables

Bucket

Cashflow

CashflowType

Date

Date comparisons

Employee

IFRSPortfolio

InstrumentTransaction

InterestRateRiskPortfolioHierarchy

Marketsegment

TOM Explorer

Best Practice Analyzer

Properties

Incremental Refresh

Enabled	True
Refresh Policy	730 day rolling, 1 day incremental (1 day behi...
Annotations	0 annotations
Extended Properties	0 extended properties
Incremental Granularity	Day
Incremental Periods	1
Incremental Periods Offset	-1
Mode	Import
Policy Type	Basic
Polling Expression	
Rolling Window Granularity	Day
Rolling Window Periods	730
Source Expression	let Source = Sql.Database("...

Data Refresh

Cancel All

Cancel

Clear completed

Clear queued

Object	Description	Progress	Duration
--------	-------------	----------	----------

# Partities verversen via Tabular Editor 3

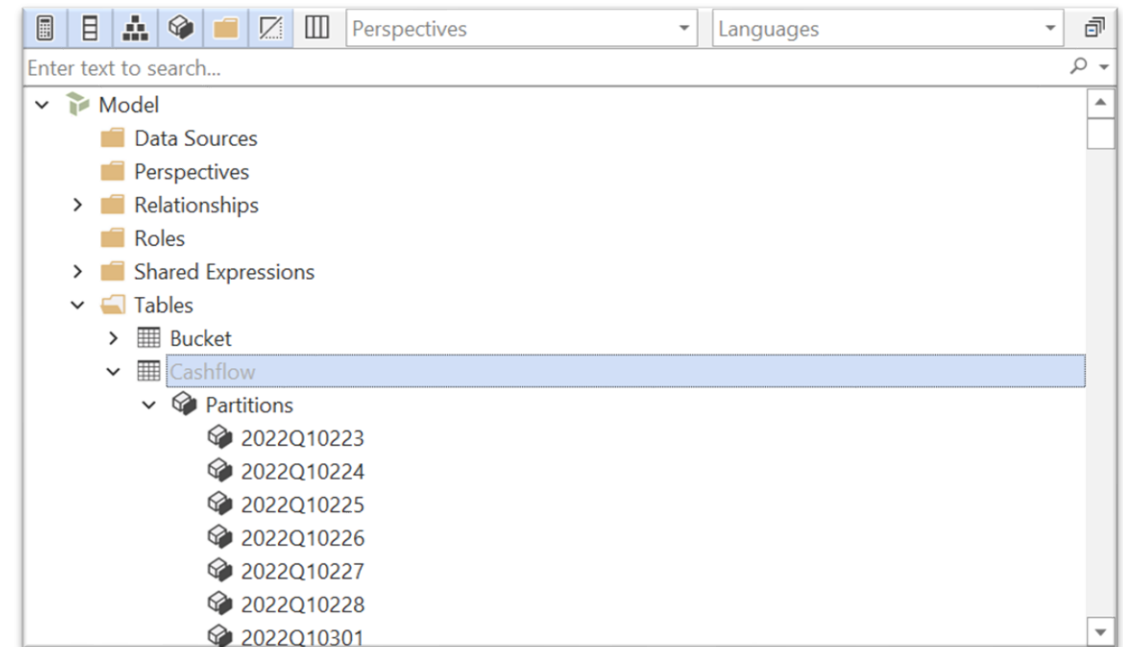
Incremental  
refresh

```
1 let
2     Source = Sql.Database("...", "dwh", [ConnectionTimeout=#duration(0, 3, 0, 0), CommandTimeout=#duration(0, 3, 0, 0)]),
3     bdm_Cashflow = Source[Schema="bdm", Item="Cashflow"]{Data},
4     #"Changed Type" = Table.TransformColumnTypes(bdm_Cashflow,{{"ReportingDate", type datetime}}),
5     #"Filtered Rows" = Table.SelectRows(#"Changed Type", each [ReportingDate] >= RangeStart and [ReportingDate] < RangeEnd),
6     #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"CashflowId", "SourceSystemCode", "ReportingDate", "TransactionId", "CashflowTypeCode", "LiquidityBucketCode", "CashflowAmountEUR"})
7 in
8     #"Removed Other Columns"
```

# Partities verversen via Tabular Editor 3

Incremental  
refresh

Wanneer je de tabel zie je de  
verschillende partities staan.

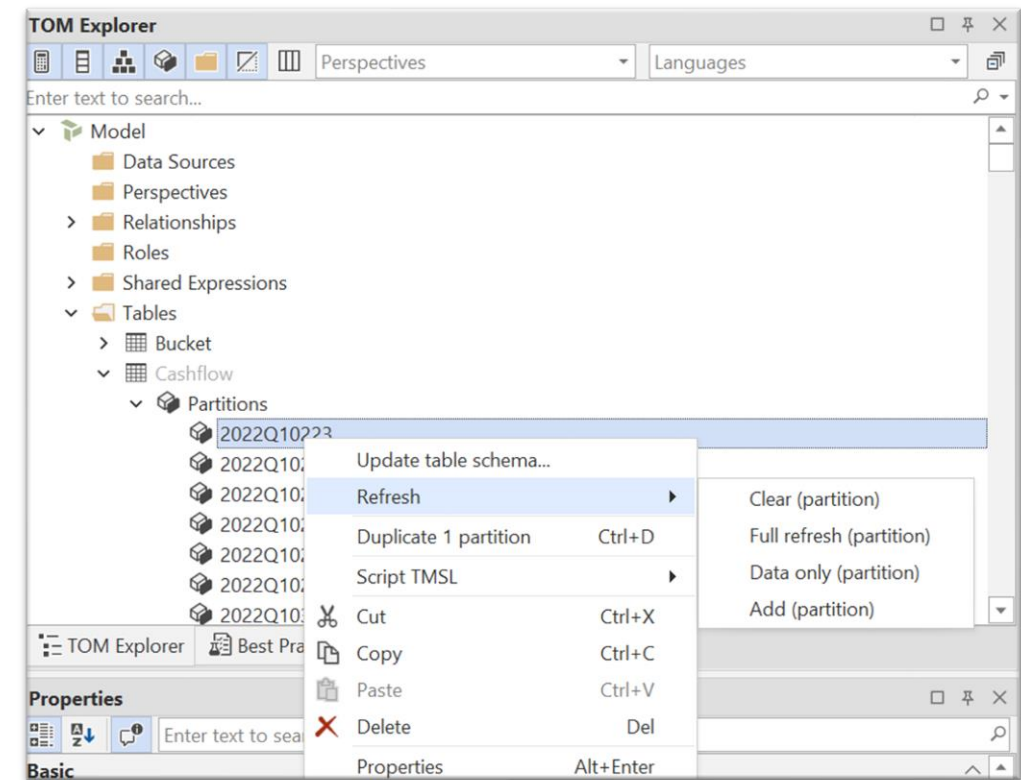




# Partities verversen via Tabular Editor 3

Incremental  
refresh

Wanneer je op de juiste partitie klikt met de rechtermuis knop vind je hier de optie “*Refresh*”. Kies hier de gewenste refresh optie.



# Partities verversen via Tabular Editor 3

Incremental  
refresh

Het voordeel ten opzichte van SSMS is dat je tijdens het vernieuwingsproces kunt zien wat er gebeurt. Je kunt per partitie observeren hoe rijen worden geladen, en zodra het proces is voltooid, wordt dit ook weergegeven.

Data Refresh			
Cancel All Cancel Clear completed Clear queued			
Object	Description	Progress	Duration
Full refresh of 23 partitions			0.00:04:45
Sequentiepuntalgoritme gestart.			0.00:04:42
2024Q10204	De verwerking van partitie '2024Q10204' van tabel 'InstrumentTransaction' is voltooid. (TableTMID='13...	100.014	0.00:04:28
2024Q10201	De verwerking van partitie '2024Q10201' van tabel 'InstrumentTransaction' is voltooid. (TableTMID='13...	99.939	0.00:01:29
2024Q10202	De verwerking van partitie '2024Q10202' van tabel 'InstrumentTransaction' is voltooid. (TableTMID='13...	100.014	0.00:04:20
2024Q10203	De verwerking van partitie '2024Q10203' van tabel 'InstrumentTransaction' is voltooid. (TableTMID='13...	100.014	0.00:04:28
2024Q10205	De verwerking van partitie '2024Q10205' van tabel 'InstrumentTransaction' is voltooid. (TableTMID='13...	99.890	0.00:02:56
2024Q10206	let __AS_Query__ = let RangeStart=DateTime.FromText("20240206T00:00:00"),		0.00:00:20
2024Q10207	let __AS_Query__ = let RangeStart=DateTime.FromText("20240207T00:00:00"),		0.00:00:13
2024Q10208	let __AS_Query__ = let RangeStart=DateTime.FromText("20240208T00:00:00"),		0.00:00:11
2024Q10209	let __AS_Query__ = let RangeStart=DateTime.FromText("20240209T00:00:00"),		0.00:00:11

## Session evaluation



## Event evaluation



Platinum  
partners

**creates.**

 **In Summa**

Goud  
partners

 **Kimura**

 **plainwater**  
de kracht van heldere data

**KASPAROV**  
FINANCE & BI

Zilver  
partners

 **rockfeather**

 **Dynamic**  
People

**GET  
RESPONSIVE**

Brons  
partners

**Hso**

**macaw**

**iqbs**

**VICTA**  
BUSINESS INTELLIGENCE

**Quanto**  
collective analytics

**ilionx**

**valcon**

**VALID**  
STAY AHEAD

Community  
partners

 **broadwick**  
Data & development recruiters

 **THE  
DATA  
COOKS**

 **Tabular Editor**

 **Datamanzi**

**Power BI**  
Connector by DAVISTA

**MINOVA**

 **AZURRO** FINANCE

 **DATA KINGDOM**

**volda;**  
INFORMATIESPECIALISTEN

**DashData.**

**VisionBI**   
Smart Data Experts

 **easydash**



**Heb je vragen of  
interesse?  
Neem contact op.**



**Peter van den Bos**

**Business Intelligence Consultant**



[peter@dutchbigeek.nl](mailto:peter@dutchbigeek.nl)

+ 31 6 13 76 07 95