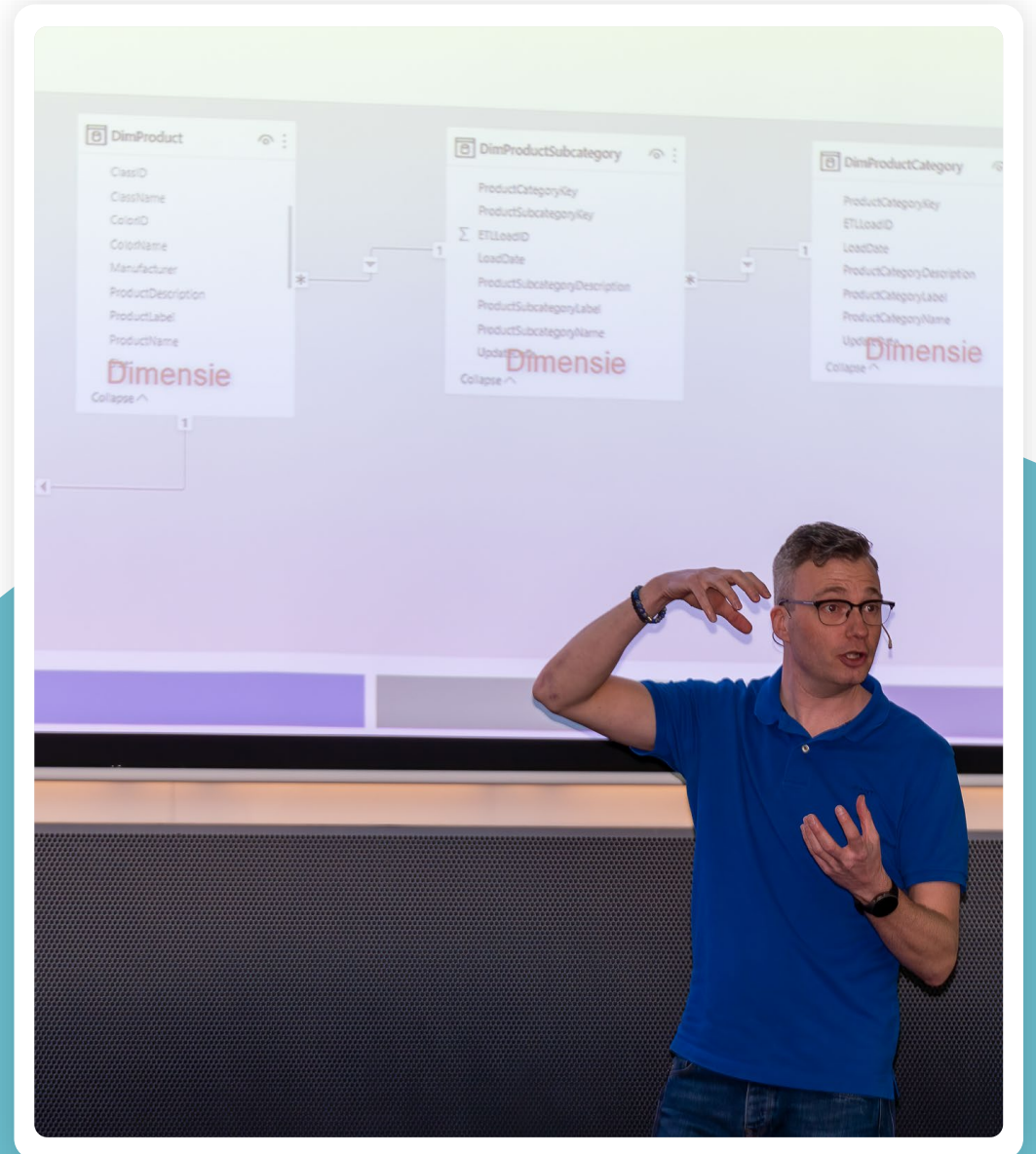




# DAX Advanced



# Praktische zaken

Data modeling

**Duur:**

Start: 9:00 uur, einde: 16:00 uur.

**Pauzes:**

Korte pauzes duren 15 minuten.

Lunch duurt 45 minuten.

**Trainingsstijl:**

Luisteren en kijken en dan vooral zelf doen.

**Telefoons graag geluid uit en voel je vrij om op elk ogenblik een vraag te stellen.**

# Agenda

Advanced

- What is Vertipag?
- Row storage lay-out
- Column storage lay-out
- Querying a columnar database
- VertiPaq compression
- Run Length Encoding (RLE)
- Dictionary encoding
- VertiPaq compression
- Segmentation
- Data memory usage
- Query memory usage
- Materialization
- Materialization in DAX
- Avoiding materialization
- VertiPaq Analyzer
- Relationships
- Reduce dictionary size
- Reduce table size
- DAX Performance
- Formula Engine (FE)
- Storage Engine (SE)
- Optimizing DAX

# What is VertiPaq?

Advanced

- **In-Memory mode:**
  - DAX Formula Engine
  - VertiPaq Storage Engine
  - In-memory columnar database
  - Met alle Tabular opties
- **Direct Query-modus:**
  - DAX naar SQL
  - Optimalisatie vereist tuning van relationele database

# What is VertiPaq?

Advanced

- Het is een in-memory database
- Gebaseerd op de relationele methodologie
- Kolom georiënteerde database

# Row storage lay-out

Advanced

StoreKey	StoreName	ZipCode	StorePhone	AddressLine1
15	Contoso Renton Store	97001	424-555-0189	Renton, Citycenter
17	Contoso Spokane Store	97001	984-555-0185	Logan Ave Shopping mall
20	Contoso Englewood Store	87001	435-555-0113	Cherry Hills Village Shopping mall
22	Contoso Westminster Store	87001	925-555-0114	South Central Shopping mall
25	Contoso Greeley No.1 Store	80631	332-555-0150	W 29th Street, Greeley CityCenter
27	Contoso Lafayette Store	87001	113-555-0173	Lafayette Downtown
30	Contoso Fort Collins Store	87001	656-555-0119	Old Town West ST
32	Contoso Milliken Store	87001	523-555-0175	Milliken Downtown
35	Contoso Denver No.2 Store	87001	952-555-0178	West Highland Borden Ave
37	Contoso Boulder Store	87001	937-555-0137	Transit Village Broadway
40	Contoso Aurora Store	87001	794-555-0159	Willow Park Ave
42	Contoso Green Bay Store	54001	286-555-0189	Green Bay Downtown
45	Contoso Oshkosh Store	54001	786-555-0144	Oshkosh, WI Broadway
47	Contoso Milwaukee No.1 Store	54001	818-555-0192	Havenwoods St

# Column storage lay-out

Advanced

StoreKey	StoreName	ZipCode	StorePhone	AddressLine1
15	Contoso Renton Store	97001	424-555-0189	Renton,Citycenter
17	Contoso Spokane Store	97001	984-555-0185	Logan Ave Shopping mall
20	Contoso Englewood Store	87001	435-555-0113	Cherry Hills Village Shopping mall
22	Contoso Westminster Store	87001	925-555-0114	South Central Shopping mall
25	Contoso Greeley No.1 Store	80631	332-555-0150	W 29th Street, Greeley CityCenter
27	Contoso Lafayette Store	87001	113-555-0173	Lafayette Downtown
30	Contoso Fort Collins Store	87001	656-555-0119	Old Town West ST
32	Contoso Milliken Store	87001	523-555-0175	Milliken Downtown
35	Contoso Denver No.2 Store	87001	952-555-0178	West Highland Borden Ave
37	Contoso Boulder Store	87001	937-555-0137	Transit Village Broadway
40	Contoso Aurora Store	87001	794-555-0159	Willow Park Ave
42	Contoso Green Bay Store	54001	286-555-0189	Green Bay Downtown
45	Contoso Oshkosh Store	54001	786-555-0144	Oshkosh, WI Broadway
47	Contoso Milwaukee No.1 Store	54001	818-555-0192	Havenwoods St

# Querying a columnar database

Advanced

StoreKey	StoreName	ZipCode	StorePhone	AddressLine1
15	Contoso Renton Store	97001	424-555-0189	Renton, Citycenter
17	Contoso Spokane Store	97001	984-555-0185	Logan Ave Shopping mall
20	Contoso Englewood Store	87001	435-555-0113	Cherry Hills Village Shopping mall
22	Contoso Westminster Store	87001	925-555-0114	South Central Shopping mall
25	Contoso Greeley No.1 Store	80631	332-555-0150	W 29th Street, Greeley CityCenter
27	Contoso Lafayette Store	87001	113-555-0173	Lafayette Downtown
30	Contoso Fort Collins Store	87001	656-555-0119	Old Town West ST
32	Contoso Milliken Store	87001	523-555-0175	Milliken Downtown
35	Contoso Denver No.2 Store	87001	952-555-0178	West Highland Borden Ave
37	Contoso Boulder Store	87001	937-555-0137	Transit Village Broadway
40	Contoso Aurora Store	87001	794-555-0159	Willow Park Ave
42	Contoso Green Bay Store	54001	286-555-0189	Green Bay Downtown
45	Contoso Oshkosh Store	54001	786-555-0144	Oshkosh, WI Broadway
47	Contoso Milwaukee No.1 Store	54001	818-555-0192	Havenwoods St



# VertiPaq compression

Advanced

- **Kolom:**
  - Snel toegang tot een enkele kolom
  - Tijd nodig om rijen te materialiseren
  - Trade CPU vs. I/O
- **Rij:**
  - Snel toegang tot een enkele rij
  - Geen materialisatie nodig
  - Trade I/O vs. CPU

# Run Length Encoding (RLE)

Advanced

Quarter	ProdID	Price
Q1	1	523
Q1	1	784
Q1	1	643
Q1	1	184
Q1	1	385
Q1	2	5445
Q1	2	454
Q1	2	665
Q1	1	49
Q4	3	408
Q4	3	612
Q4	3	502
Q4	3	144
Q4	1	300
Q4	1	4247
Q4	1	354
Q4	1	519
Q4	1	38

Quarter	Start	count
Q1	1	9
Q4	10	9

ProdID	Start	count	Price
1	1	5	523
2	6	3	784
1	9	1	643
3	10	4	184
1	14	5	385
			5445
			454
			665
			49
			408
			612
			502
			144
			300
			4247
			354
			519
			38

# Dictionary encoding

Advanced

Quarter
Q1
Q1
Q1
Q1
Q1
Q2
Q2
Q2
Q2
Q2
Q3
Q3
Q3
Q3
Q4
Q4
Q4
Q4

Quarter.id
0
0
0
0
0
1
1
1
1
1
2
2
2
2
3
3
3
3

Quarter.id	Quarter.id
0	Q1
1	Q2
2	Q3
3	Q4

Quarter.id	Start	Count
0	1	5
1	6	5
2	11	5
3	16	5

# VertiPaq compression

Advanced

- **Dictionary codering:**
  - Gebeurt wanneer nodig
  - Grote reeksen hele getallen worden niet gecodeerd
  - Maakt tabellen onafhankelijk van soort gegevens
- **RLE codering:**
  - Alleen als gecomprimeerde gegevens kleiner zijn dan de originele
  - Hangt sterk af van de volgorde van de gegevens
  - SASS kiest automatisch de beste sortering

# VertiPaq compression

Advanced

- **Compressie komt van:**
  - Kolom opslag
  - Dictionary codering
  - RLE codering
- Minder RAM nodig voor in-memory database
- Sneller kolommen scannen
- 10x is een goede gemiddelde compressie ratio, vergeleken met niet-gecomprimeerde SQL database

# Segmentation

Advanced

- **Elke tabel is verdeeld in segmenten:**
  - 8 miljoen rijen voor elk segment in SSAS
  - 1 miljoen rijen in Power Pivot en Power BI
- Dictionary algemeen voor tabel
- Bit-sizing
- DMV beschikbaar om die info op te vragen

# Data memory usage

Advanced

- **Het geheugengebruik hangt af van:**
  - Aantal kolommen
  - Cordialiteit van elke kolom
  - Soort gegevens
  - Aantal rijen
- **Strings:**
  - Gemiddelde grootte is relevant voor grootte dictionary
  - Er is geen gemakkelijke formule

# Query memory usage

Advanced

- **Eenvoudige query's vereisen wat geheugen**
- **Complexe query's vereisen meer geheugen:**
  - Spoolen van tijdelijke waarden
  - Materialiseren van datasets
- **Cache vereist geheugen**
- **Materialisatie is het grote probleem**



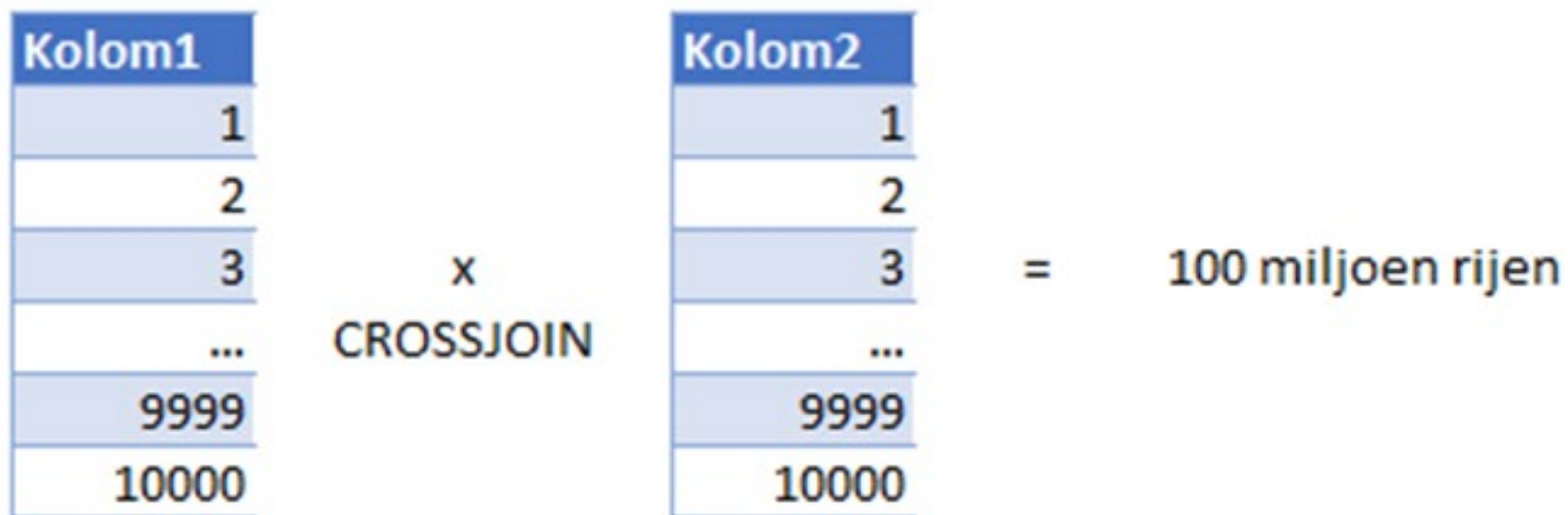
# Materialization

Advanced

- **Materialisatie voor:**
  - Complexe verbindingen
  - Complexe iterators
- **Tijdelijke gegevens gespoold voor verdere verwerking**
- **Geheugeneisen:**
  - Misschien meer dan de hele database
  - Gespoelde gegevens worden niet gecomprimeerd

# Materialization in DAX

Advanced



# Avoiding materialization

DAX

- Complexe conditions, waarbij expressions over meerdere kolommen worden gebruikt, zijn gevaarlijk voor materialisatie
- Hoe eerder ze in het proces berekend worden hoe beter

```
1  bla =  
2  CALCULATE (  
3    SUMX (  
4      FILTER (  
5        FactSales,  
6        OR (  
7          FactSales[CurrencyKey] > FactSales[SalesAmount],  
8          FactSales[channelKey] = 1  
9        )  
10     ),  
11     FactSales[StoreKey] = 77  
12  )  
13 )
```

# VertiPaq Analyzer

DAX

- Beschikbaar op: [www.sqlbi.com/tools/vertipaq-analyzer](http://www.sqlbi.com/tools/vertipaq-analyzer)



# Relationships

Advanced

- Grote dimensies > 2m
- Cardinaliteit filterkolom niet belangrijk; de cardinaliteit van de primaire sleutel is wél belangrijk

# Reduce dictionary size

Advanced

- **Aantal verschillende waarden verminderen:**
  - Date Time: opgesplitst in twee kolommen
  - Date
  - Time
  - Decimaalwaarden: vaste precisie  $10.231 > 10.2$
- **Strings korter maken**
- **Dit moet allemaal gebeuren in brongegevens (bv. SQL-weergaven), niet in berekende kolommen**

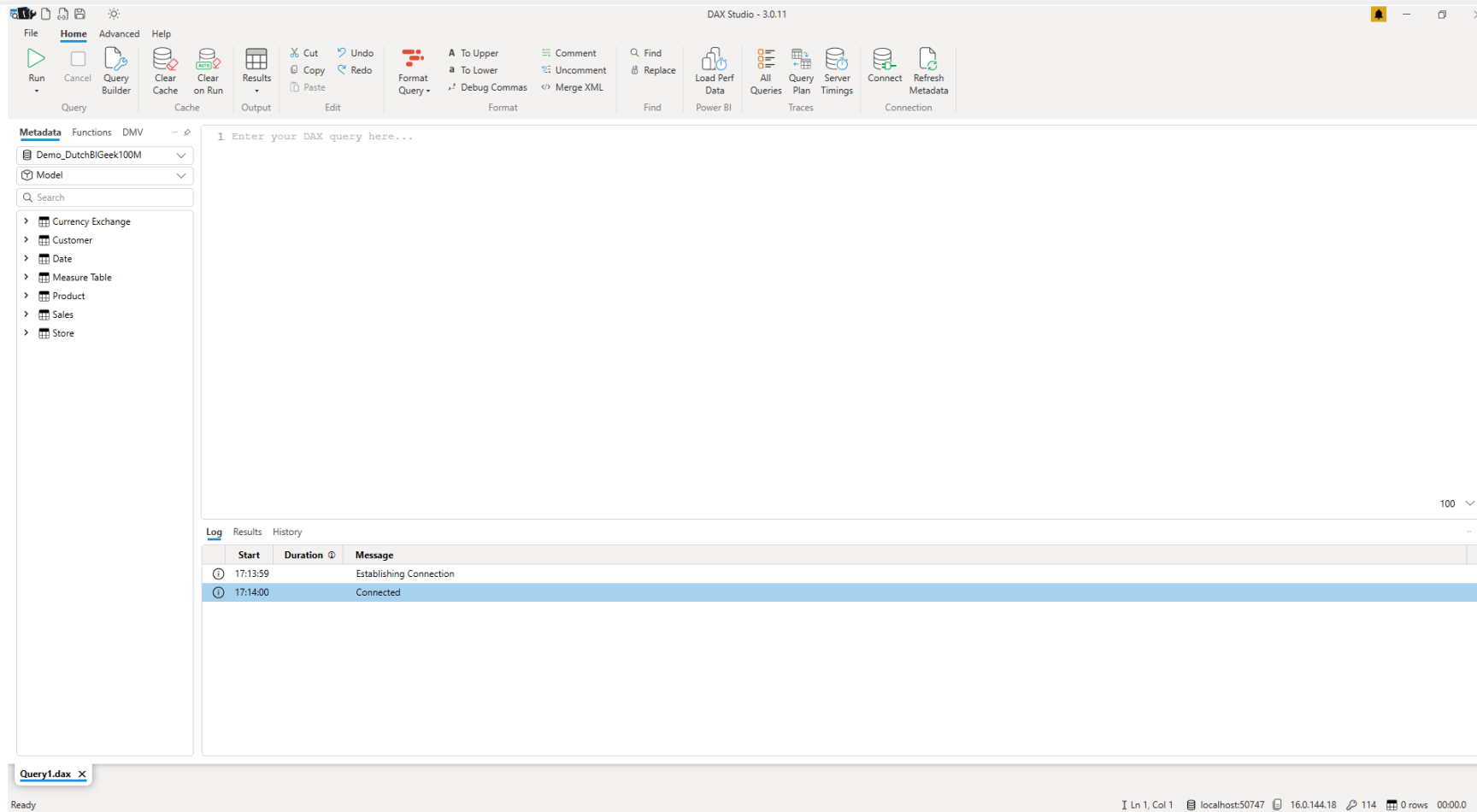
# Reduce table size

Advanced

- Nutteloze kolommen verwijderen

# DAX Studio

DAX studio



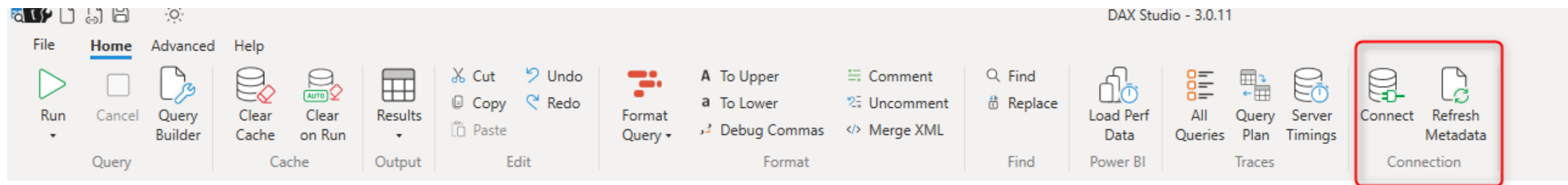
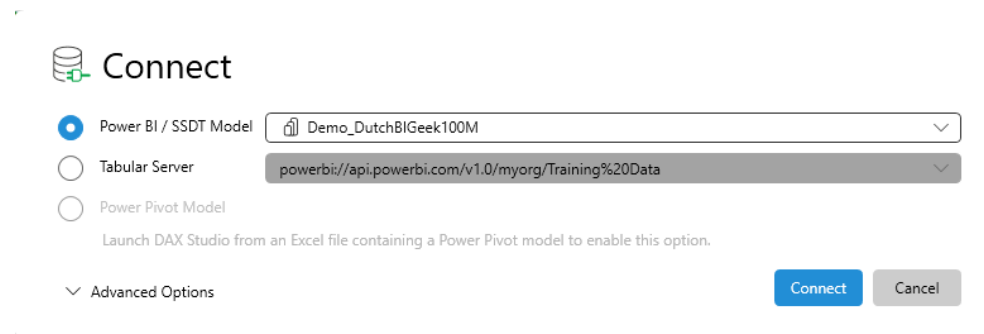


# Maak verbinding met het Datamodel

DAX studio

De eerste stap is het maken van een verbinding met het datamodel.

Dit doe je door op "Connect" te klikken en vervolgens het gewenste model te selecteren, ofwel via Power BI Desktop of door de online link in te vullen.

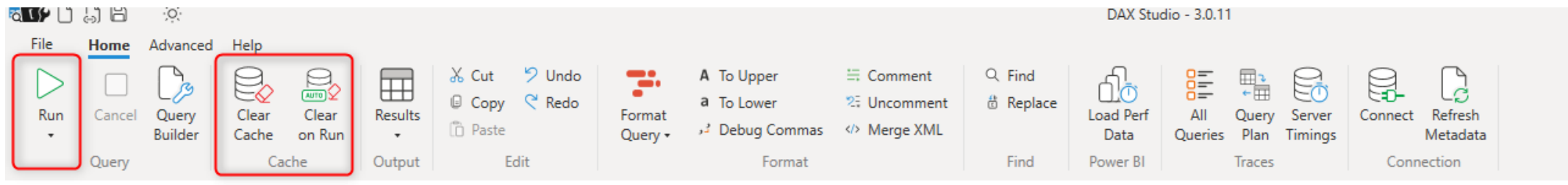


# Uitvoeren van Query's

DAX studio

Klik op **"Run"** om de query of evaluatie van bijvoorbeeld een tabel te starten.

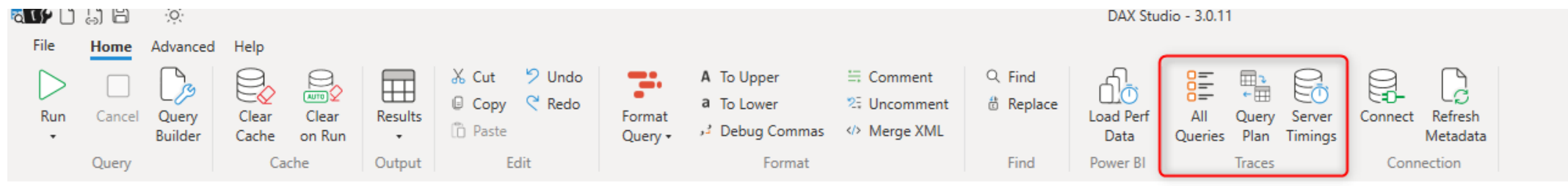
Zorg ervoor dat **"Clear Cache Before Running"** is aangevinkt. Dit garandeert dat de cache wordt verversed bij het uitvoeren van de query.



# Traces

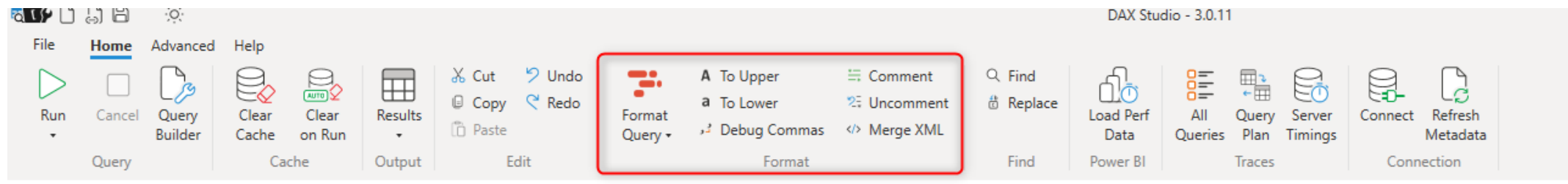
DAX studio

- All Queries: Registreert alle DAX-queries die worden uitgevoerd, voor inzicht in querygedrag en prestaties.
- Query Plans: Toont uitvoeringsplannen van DAX-queries, nuttig voor prestatie-optimalisatie.
- Server Timings: Geeft gedetailleerde tijdmetingen van query-uitvoering, helpt bij het identificeren van bottlenecks.



# Format

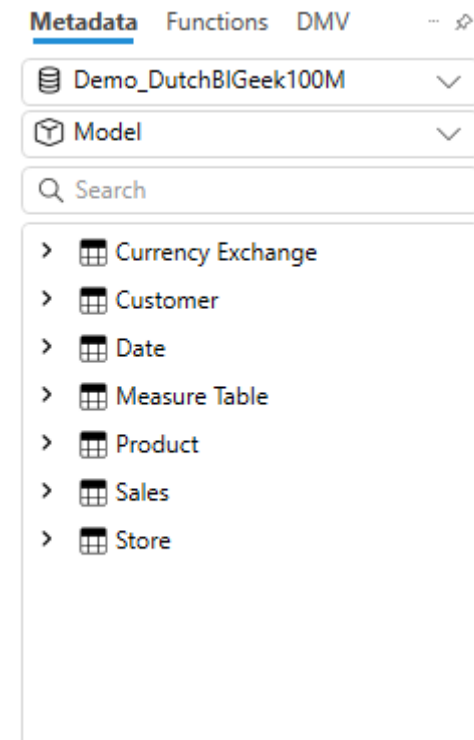
- Format Button: Formateert DAX-code automatisch voor betere leesbaarheid en structuur.
- Comment Button: Zet geselecteerde code om in commentaar, zodat het niet wordt uitgevoerd.
- Uncomment Button: Maakt gecommmentarieerde code weer actief voor uitvoering.



# Metadata

DAX studio

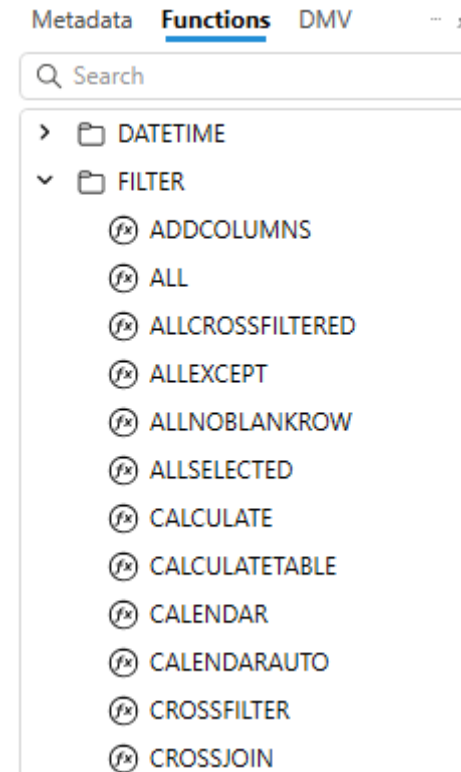
De **Metadata tab** toont het model dat in de verbinding wordt gebruikt en geeft een overzicht van de tabellen in het datamodel.



# Functions

DAX studio

De **Functions tab** geeft een overzicht van alle DAX-functies die je kunt gebruiken.



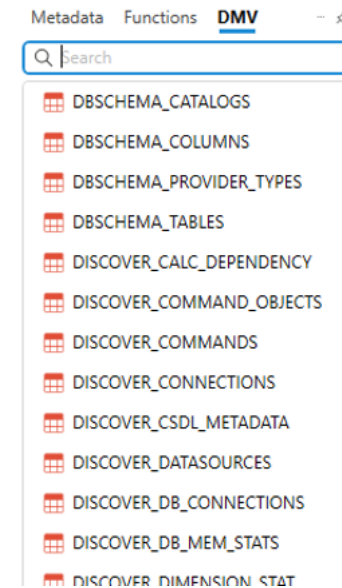
# Data Management Views (DMV's)

DAX studio

De **DMV tab** toont alle Data Management Views (DMV's) die je kunt gebruiken.

DMV's zijn query's die informatie over de interne werking van het data model en de server geven, zoals metadata, prestaties, en instellingen.

Ze zijn nuttig voor het beheren en optimaliseren van je data model.



# Log

DAX studio

Het **Log** in DAX Studio toont een gedetailleerde lijst van gebeurtenissen en acties die tijdens je sessie plaatsvinden.

Dit omvat uitgevoerde queries, fouten, waarschuwingen, en andere diagnostische informatie.

Log	Results	History	All Queries	Query Plan	Server Timings	VertiPaq Analyzer
	Start	Duration ⓘ	Message			
✓	17:38:23	10	Cache Cleared for Database: f44800e0-8ecb-4e4d-8ab3-2363737a646c			
ⓘ	17:38:23		Query Started			
ⓘ	17:38:23		Query 1 Completed (194 rows returned)			
✓	17:38:23	22	Query Batch Completed			
ⓘ	17:53:26		Evaluating Calculation Script for Database: f44800e0-8ecb-4e4d-8ab3-2363737a646c			
✓	17:53:26	12	Cache Cleared for Database: f44800e0-8ecb-4e4d-8ab3-2363737a646c			
ⓘ	17:53:26		Query Started			
ⓘ	17:53:26		Query 1 Completed (2,517 rows returned)			
✓	17:53:26	73	Query Batch Completed			
ⓘ	17:53:39		Evaluating Calculation Script for Database: f44800e0-8ecb-4e4d-8ab3-2363737a646c			
✓	17:53:39	10	Cache Cleared for Database: f44800e0-8ecb-4e4d-8ab3-2363737a646c			
ⓘ	17:53:39		Query Started			



# Result

DAX studio

Het **Result** venster in DAX Studio toont de uitvoer van de DAX-queries die je hebt uitgevoerd.

Dit kunnen tabellen, berekende waarden, of andere resultaten zijn die voortkomen uit je DAX-code.

Het Result-venster is waar je direct de uitkomst van je query's kunt bekijken en analyseren.

Log **Results** History

StoreKey	Store Code	Country	State	Name	Square Meters	Open Date	Close Date	Status
10	1	Australia	Australian Capital Territory	Contoso Store Australian Capital Territory	595	2008-01-01		
20	2	Australia	Northern Territory	Contoso Store Northern Territory	665	2008-01-12	2016-07-07	Closed
30	3	Australia	South Australia	Contoso Store South Australia	2.000	2012-01-07	2015-08-08	Restructured
35	3	Australia	South Australia	Contoso Store South Australia	3.000	2015-12-08		
40	4	Australia	Tasmania	Contoso Store Tasmania	2.000	2010-01-01		
50	5	Australia	Victoria	Contoso Store Victoria	2.000	2015-12-09		
60	6	Australia	Western Australia	Contoso Store Western Australia	2.000	2010-01-01		
70	7	Canada	New Brunswick	Contoso Store New Brunswick	1.105	2007-05-07	2014-03-09	Restructured
72	7	Canada	New Brunswick	Contoso Store New Brunswick	1.500	2015-01-11	2018-02-02	Restructured
74	7	Canada	New Brunswick	Contoso Store New Brunswick	3.500	2018-06-02		

# Historie

DAX studio

De **Historie** tab in DAX Studio toont een overzicht van alle eerder uitgevoerde DAX-queries tijdens je sessie.

Dit maakt het gemakkelijk om eerdere query's te herzien, opnieuw uit te voeren, of aan te passen zonder ze opnieuw te hoeven invoeren.

Het is handig voor het bijhouden van je werk en het terugvinden van eerdere analyses.

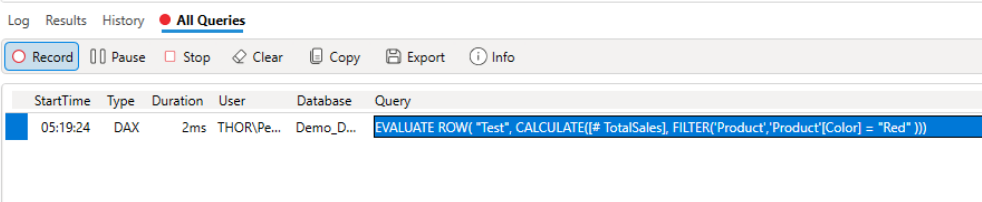
Log Results **History**

Start Time	Rows	Client ms	Server ms	FE	SE	Server	Database	Query
20-8-2024 14:13		-1 ms	1 ms	1 ms	n/a	<power bi>	Demo_DutchBIGeek100M	EVALUATE ROW ( "Test",
20-8-2024 14:13		-1 ms	1 ms	1 ms	n/a	<power bi>	Demo_DutchBIGeek100M	EVALUATE ROW ( "Test",
20-8-2024 14:13	1	116 ms	117 ms	6 ms	111 ms	<power bi>	Demo_DutchBIGeek100M	EVALUATE ROW( "Test", CALCULATE([# TotalSales], FILTER('Prc
20-8-2024 14:12		-1 ms	1 ms	1 ms	n/a	<power bi>	Demo_DutchBIGeek100M	EVALUATE ROW ( "Test",
20-8-2024 14:12	99	6 ms	7 ms	5 ms	2 ms	<power bi>	Demo_DutchBIGeek100M	EVALUATE FILTER (

# All Queries

DAX studio

De **All Queries** functie in DAX Studio registreert en toont alle DAX-queries die tijdens je sessie worden uitgevoerd. Dit geeft je een volledig overzicht van elke query die naar het data model is gestuurd, inclusief de volgorde, de tijdstippen, en eventuele gerelateerde informatie.



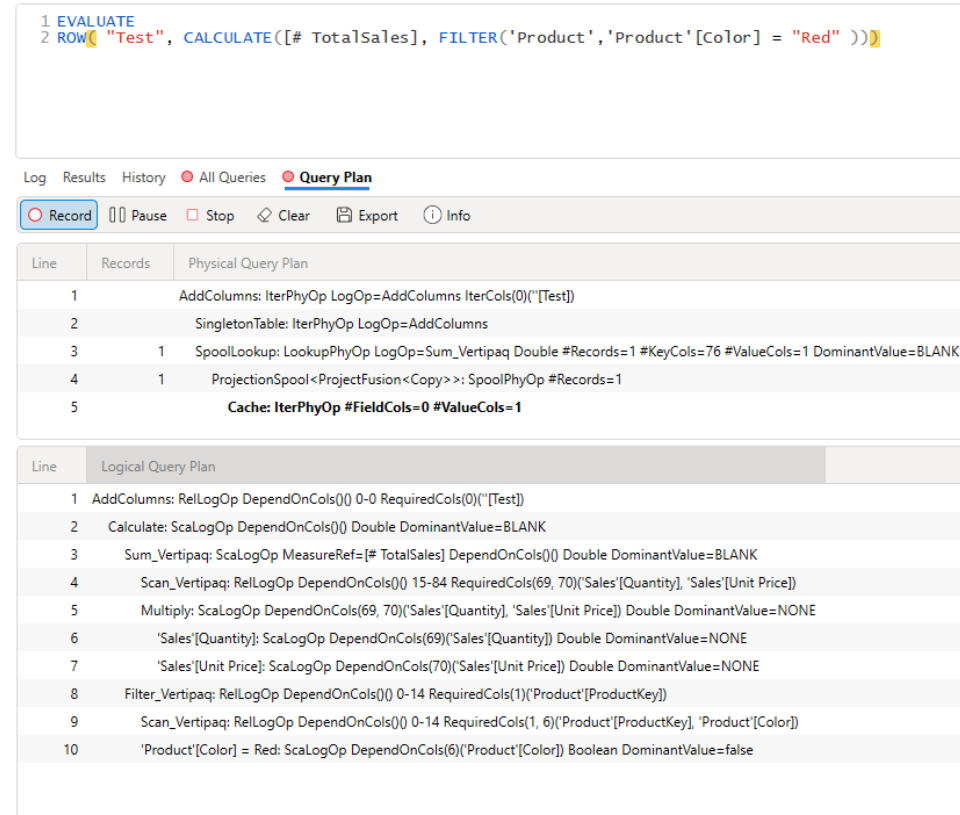
The screenshot shows the 'All Queries' window in DAX Studio. It features a toolbar with buttons for Record, Pause, Stop, Clear, Copy, Export, and Info. Below the toolbar is a table with the following data:

StartTime	Type	Duration	User	Database	Query
05:19:24	DAX	2ms	THOR/Pe...	Demo_D...	EVALUATE ROW( "Test", CALCULATE([# TotalSales], FILTER("Product",Product[Color] = "Red" )))

# Query Plan

DAX studio

De **Query Plan** functie in DAX Studio geeft inzicht in hoe DAX-queries worden uitgevoerd door de engine. Het toont een gedetailleerd uitvoeringsplan dat laat zien welke stappen de formule-engine en de opslag-engine nemen om een query te verwerken. Dit omvat informatie over de gebruikte algoritmen en de volgorde van operaties. Door het queryplan te analyseren, kun je identificeren waar mogelijke prestatieproblemen liggen en beter begrijpen hoe je je DAX-code kunt optimaliseren voor snellere en efficiëntere uitvoering.



The screenshot displays the DAX Studio interface. At the top, a DAX query is entered in the formula bar:

```
1 EVALUATE
2 ROWS("Test", CALCULATE([# TotalSales], FILTER('Product', 'Product'[Color] = "Red" )))
```

Below the formula bar, the 'Query Plan' tab is selected. It contains a toolbar with buttons for Record, Pause, Stop, Clear, Export, and Info. The main area is divided into two sections: 'Physical Query Plan' and 'Logical Query Plan'.

Line	Records	Physical Query Plan
1		AddColumns: IterPhyOp LogOp=AddColumns IterCols(0)('Test')
2		SingletonTable: IterPhyOp LogOp=AddColumns
3	1	SpoolLookup: LookupPhyOp LogOp=Sum_Vertipaq Double #Records=1 #KeyCols=76 #ValueCols=1 DominantValue=BLANK
4	1	ProjectionSpool<ProjectFusion<Copy>>: SpoolPhyOp #Records=1
5		Cache: IterPhyOp #FieldCols=0 #ValueCols=1

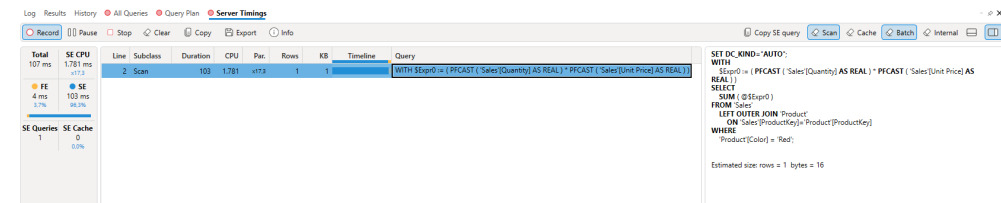
Line	Logical Query Plan
1	AddColumns: RelLogOp DependOnCols(0) 0-0 RequiredCols(0)('Test')
2	Calculate: ScaLogOp DependOnCols(0) Double DominantValue=BLANK
3	Sum_Vertipaq: ScaLogOp MeasureRef=[# TotalSales] DependOnCols(0) Double DominantValue=BLANK
4	Scan_Vertipaq: RelLogOp DependOnCols(0) 15-84 RequiredCols(69, 70)('Sales'[Quantity], 'Sales'[Unit Price])
5	Multiply: ScaLogOp DependOnCols(69, 70)('Sales'[Quantity], 'Sales'[Unit Price]) Double DominantValue=NONE
6	'Sales'[Quantity]: ScaLogOp DependOnCols(69)('Sales'[Quantity]) Double DominantValue=NONE
7	'Sales'[Unit Price]: ScaLogOp DependOnCols(70)('Sales'[Unit Price]) Double DominantValue=NONE
8	Filter_Vertipaq: RelLogOp DependOnCols(0) 0-14 RequiredCols(1)('Product'[ProductKey])
9	Scan_Vertipaq: RelLogOp DependOnCols(0) 0-14 RequiredCols(1, 6)('Product'[ProductKey], 'Product'[Color])
10	'Product'[Color] = Red: ScaLogOp DependOnCols(6)('Product'[Color]) Boolean DominantValue=false

# Server Timings

DAX studio

De **Server Timings** functie in DAX Studio toont gedetailleerde tijdmetingen voor elke fase van een DAX-query, verdeeld over de **Storage Engine** en **Formula Engine**.

Dit helpt je om bottlenecks te identificeren en de prestaties van je queries te optimaliseren.

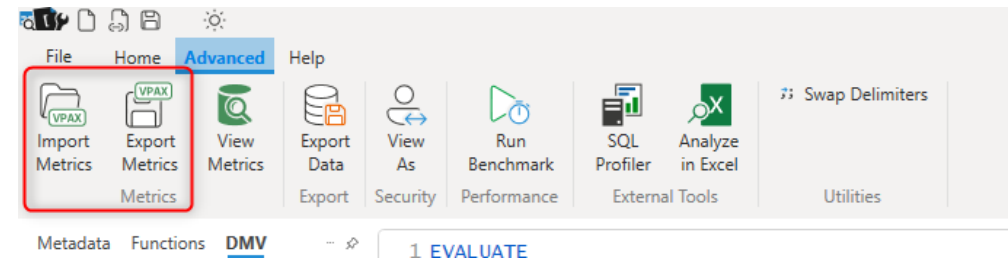


# Metrics

DAX studio

**Import Metrics:** Hiermee kun je eerder opgeslagen prestatiedata inladen in DAX Studio voor verdere analyse.

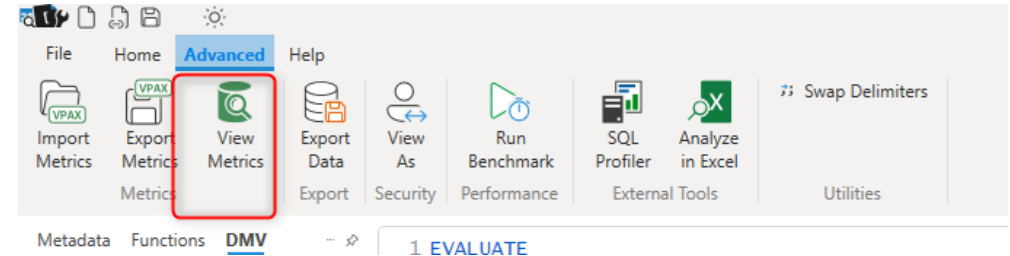
**Export Metrics:** Hiermee kun je de prestatiedata die tijdens je sessie is verzameld opslaan naar een bestand. Dit maakt het mogelijk om deze gegevens later te bekijken of te delen met anderen voor diepgaandere analyse.



# View Metrics

DAX studio

De **View Metrics** functie in DAX Studio toont prestatiestatistieken zoals geheugengebruik en meta-data essentieel voor het optimaliseren van je datamodel.

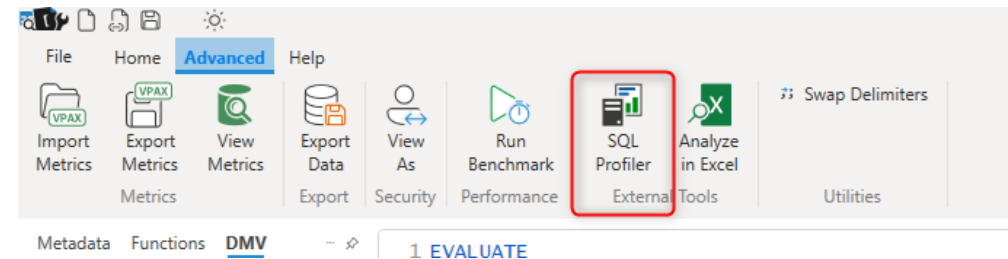


# SQL Profiler

DAX studio

**SQL Profiler** is een tool waarmee je real-time de activiteiten van DAX-queries kunt volgen.

Het registreert gedetailleerde gebeurtenissen en prestaties van queries, wat helpt bij het diagnosticeren en optimaliseren van query-uitvoering in je datamodel.

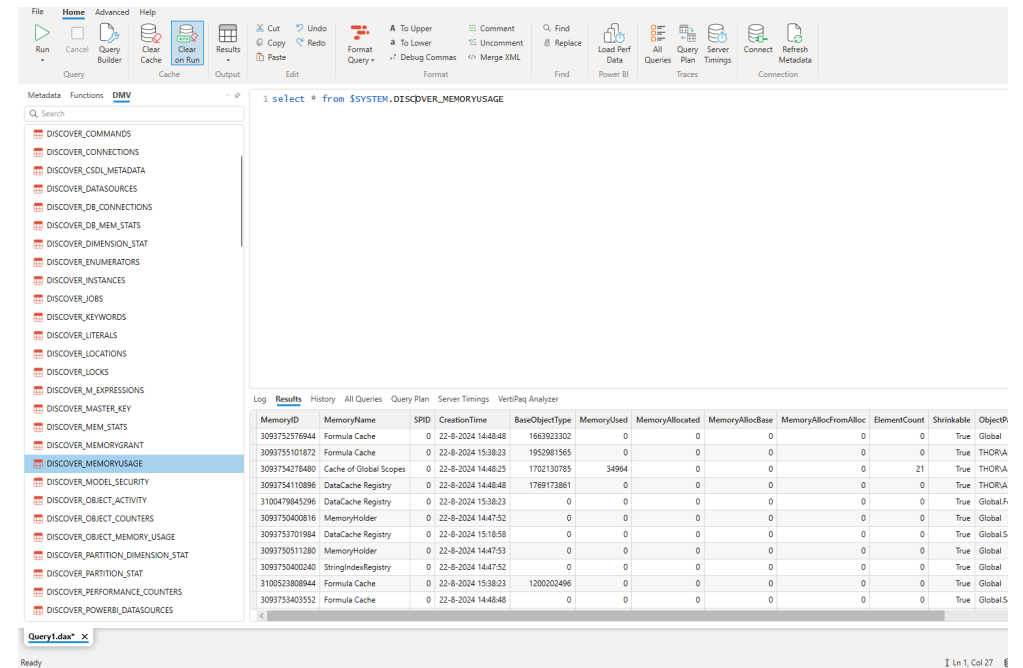




# DMV's

DAX studio

Dynamic Management Views (DMV's) in gebruik om het geheugen inzichtelijk te maken.



The screenshot shows the DAX Studio interface. The query editor contains the following SQL query:

```
1 select * from $SYSTEM.DISCOVER_MEMORYUSAGE
```

The left pane shows the list of Dynamic Management Views (DMVs). The right pane displays the results of the query, which is a table with the following columns: MemoryID, MemoryName, SPID, CreationTime, BaseObjectType, MemoryUsed, MemoryAllocated, MemoryAllocBase, MemoryAllocFromAlloc, ElementCount, Shrinkable, and ObjectID.

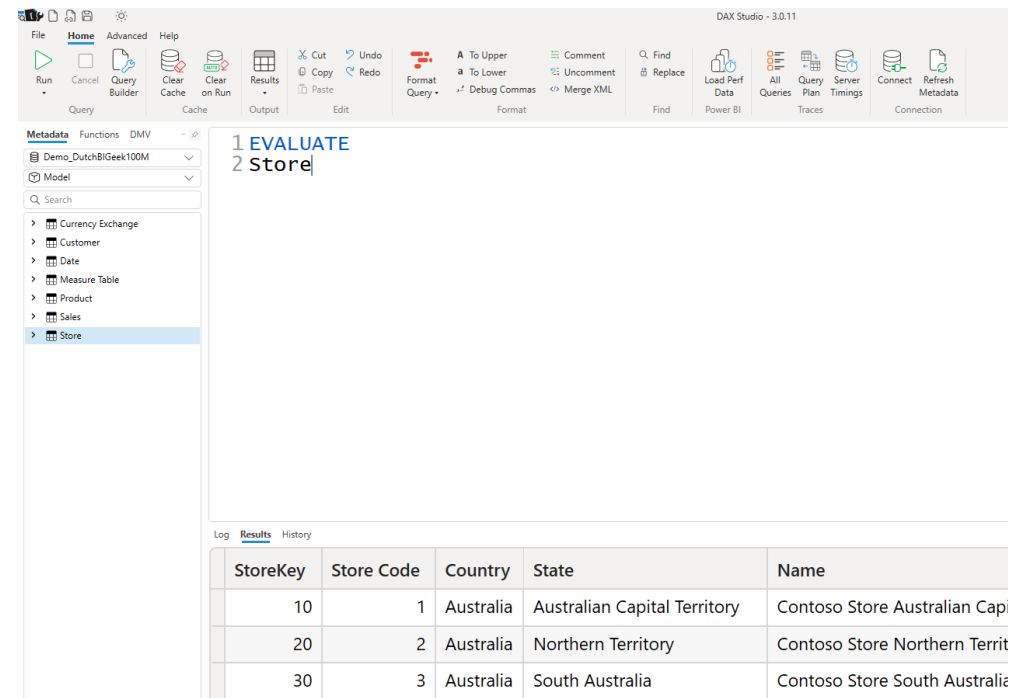
MemoryID	MemoryName	SPID	CreationTime	BaseObjectType	MemoryUsed	MemoryAllocated	MemoryAllocBase	MemoryAllocFromAlloc	ElementCount	Shrinkable	ObjectID
3093752576844	Formula Cache	0	22-8-2024 14:48:48	1663923302	0	0	0	0	0	True	Global
3093755101872	Formula Cache	0	22-8-2024 15:38:23	1952981565	0	0	0	0	0	True	THORA
3093754278480	Cache of Global Scopes	0	22-8-2024 14:48:25	1702130785	34964	0	0	0	21	True	THORA
3093754110896	DataCache Registry	0	22-8-2024 14:48:48	1769173861	0	0	0	0	0	True	THORA
3100478645396	DataCache Registry	0	22-8-2024 15:38:23	0	0	0	0	0	0	True	Global.F
3093750400816	MemoryHolder	0	22-8-2024 14:47:52	0	0	0	0	0	0	True	Global
3093753701984	DataCache Registry	0	22-8-2024 15:18:58	0	0	0	0	0	0	True	Global.S
3093750511380	MemoryHolder	0	22-8-2024 14:47:53	0	0	0	0	0	0	True	Global
3093750400240	StringIndexRegistry	0	22-8-2024 14:47:52	0	0	0	0	0	0	True	Global
3100523809844	Formula Cache	0	22-8-2024 15:38:23	1200202496	0	0	0	0	0	True	Global
3093753403552	Formula Cache	0	22-8-2024 14:48:48	0	0	0	0	0	0	True	Global.S

# Tabellen bekijken in DAX Studio

DAX studio

Om een tabel in DAX Studio te bekijken, moet je altijd gebruik maken van het **EVALUATE** statement.

EVALUATE is een DAX-opdracht die nodig is om een query te starten en de resultaten op te halen.



The screenshot shows the DAX Studio interface. The top menu bar includes File, Home, Advanced, and Help. The Home tab is active, showing various icons for running queries, clearing cache, and viewing results. The left pane shows the 'Metadata' view with a tree structure of tables: Demo\_DutchBIgeek100M, Model, Currency Exchange, Customer, Date, Measure Table, Product, Sales, and Store. The 'Store' table is selected. The main editor area shows a DAX query: `1 EVALUATE`  
`2 Store`. The bottom pane shows the 'Results' view with a table of data.

StoreKey	Store Code	Country	State	Name
10	1	Australia	Australian Capital Territory	Contoso Store Australian Capi
20	2	Australia	Northern Territory	Contoso Store Northern Territ
30	3	Australia	South Australia	Contoso Store South Australia

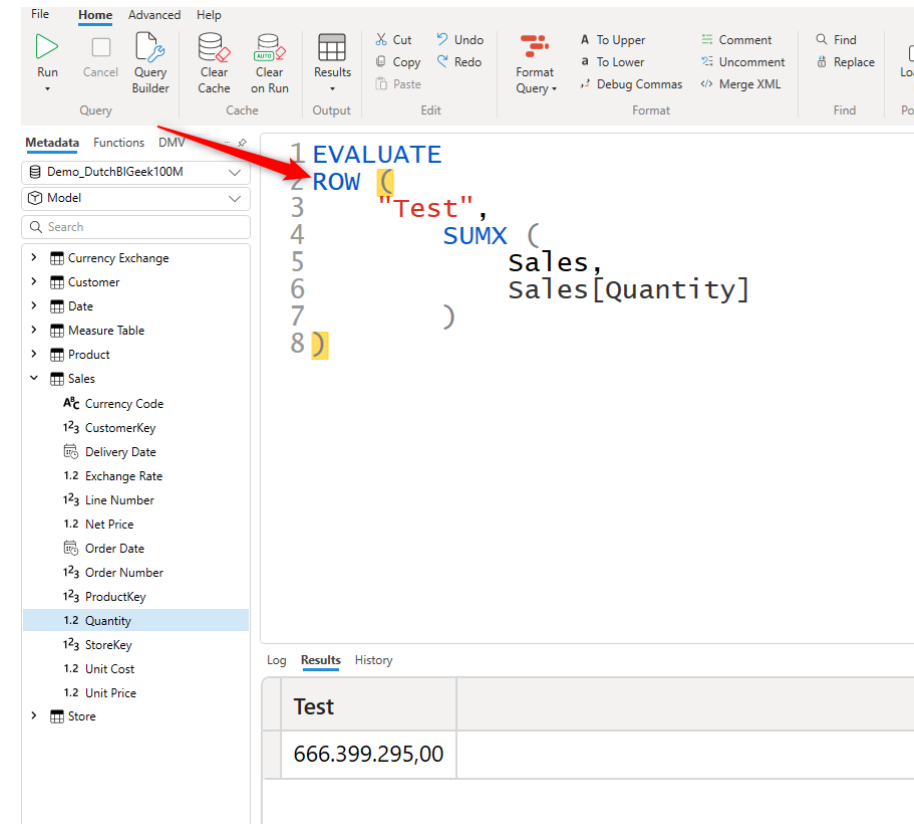
# Tabel Expressions

DAX studio

DAX Studio kan alleen tabel-expressies uitvoeren, wat betekent dat elke measure of bewerking moet resulteren in een tabel, zelfs als het slechts één cel is.

Hiervoor kun je bijvoorbeeld de **ROW** functie gebruiken. Je geeft een naam voor de kolom en na de komma geef je bijvoorbeeld een DAX-measure weer.

Dit is ook handig om de prestaties van een afzonderlijke measure te testen.



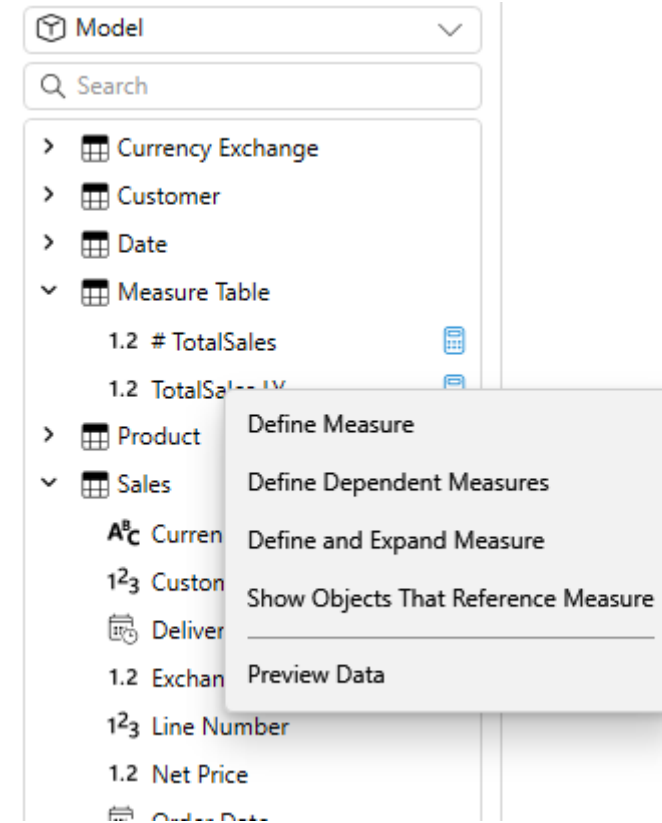
# Een DAX Query bekijken in je Datamodel

DAX studio

Er zijn verschillende manieren om een DAX-query te bekijken in je datamodel.

Als je met de rechtermuisknop op een measure klikt, zie je drie opties die je kunt selecteren:

- **Define Measure**
- **Define Dependent Measure**
- **Define and Expand Dependent Measure**

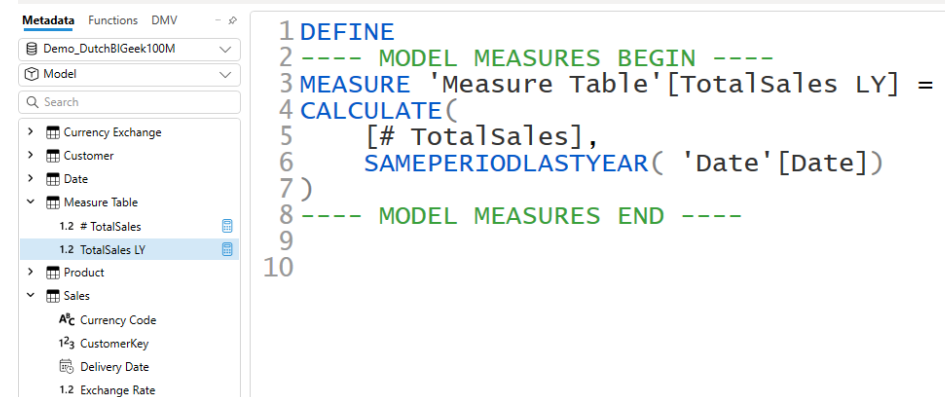


# Define Measure

DAX studio

**Define Measure** in DAX Studio is een optie waarmee je een measure kunt definiëren en testen binnen de DAX Studio-omgeving zonder deze definitief toe te voegen aan je datamodel.

Het stelt je in staat om de measure te bewerken, uit te voeren en de resultaten te analyseren, zodat je kunt experimenteren en optimaliseren voordat je besluit om de measure daadwerkelijk op te nemen in je model.



The screenshot shows the DAX Studio interface. On the left, the 'Metadata' pane displays a tree view of the data model. The 'Measure Table' is expanded, showing '1.2 # TotalSales' and '1.2 TotalSales LY'. The '1.2 TotalSales LY' measure is selected. On the right, the 'Define Measure' dialog is open, showing the DAX code for the selected measure. The code is as follows:

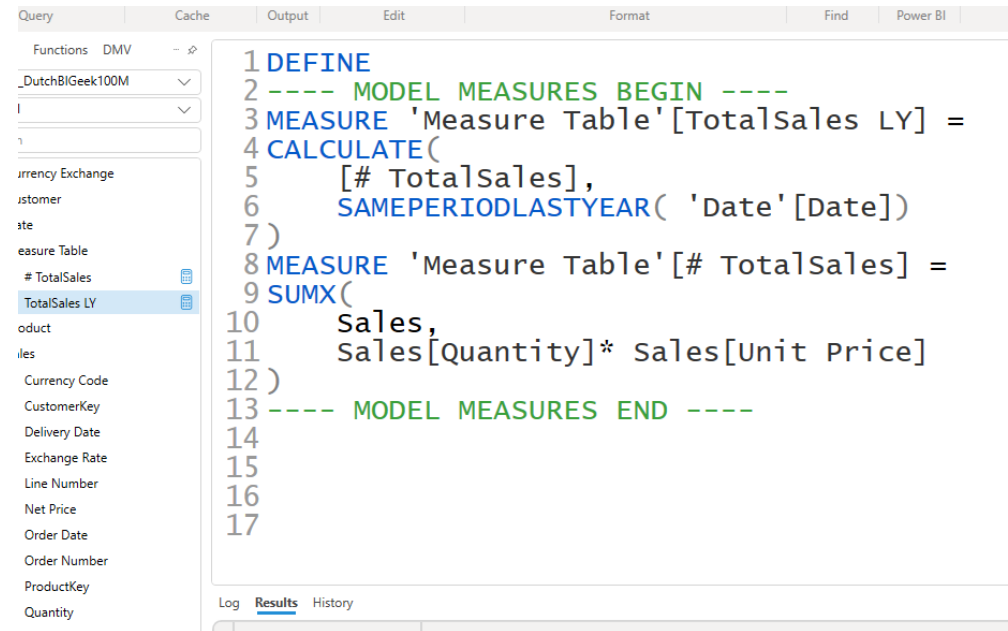
```
1 DEFINE
2 ----- MODEL MEASURES BEGIN -----
3 MEASURE 'Measure Table'[TotalSales LY] =
4 CALCULATE(
5     [# TotalSales],
6     SAMEPERIODLASTYEAR( 'Date'[Date])
7 )
8 ----- MODEL MEASURES END -----
9
10
```

# Define Dependent Measure

DAX studio

**Define Dependent Measure** in DAX Studio definieert een measure samen met alle andere measures waar deze afhankelijk van is.

Dit is handig voor het testen en analyseren van hoe de measure werkt in combinatie met zijn afhankelijkheden.



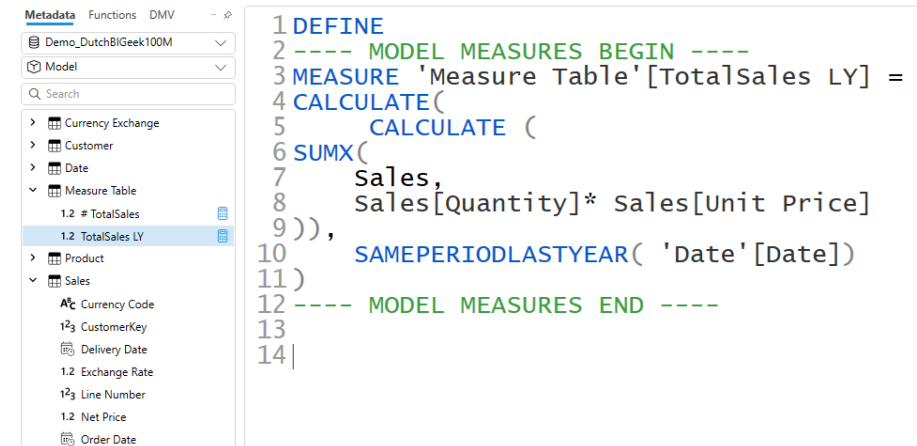
```
1 DEFINE
2 ---- MODEL MEASURES BEGIN ----
3 MEASURE 'Measure Table'[TotalSales LY] =
4 CALCULATE(
5     [# TotalSales],
6     SAMEPERIODLASTYEAR( 'Date'[Date])
7 )
8 MEASURE 'Measure Table'[# TotalSales] =
9 SUMX(
10     Sales,
11     Sales[Quantity]* Sales[Unit Price]
12 )
13 ---- MODEL MEASURES END ----
14
15
16
17
```

# Define and Expand Dependent Measure

DAX studio

**Define and Expand Dependent Measure**  
in DAX Studio definieert een measure én breidt deze uit met alle gerelateerde en afhankelijke measures.

Dit geeft je een compleet overzicht van alle onderliggende berekeningen, wat nuttig is voor grondige analyse en optimalisatie.



The screenshot shows the DAX Studio interface. On the left, the 'Metadata' pane displays a tree view of the data model. The 'Measure Table' is expanded, showing a list of measures including '1.2 # TotalSales' and '1.2 TotalSales LY'. The 'Sales' table is also expanded, showing columns like 'Currency Code', 'CustomerKey', 'Delivery Date', 'Exchange Rate', 'Line Number', 'Net Price', and 'Order Date'. On the right, the 'Functions' pane shows a DAX query. The query is a DAX measure definition for 'TotalSales LY' using the 'CALCULATE' and 'SUMX' functions. The query is as follows:

```
1 DEFINE
2 ---- MODEL MEASURES BEGIN ----
3 MEASURE 'Measure Table'[TotalSales LY] =
4 CALCULATE(
5     CALCULATE (
6 SUMX(
7     Sales,
8     Sales[Quantity]* Sales[Unit Price]
9 )),
10    SAMEPERIODLASTYEAR( 'Date'[Date])
11 )
12 ---- MODEL MEASURES END ----
13
14 |
```

# Voordeel van de drie methoden

DAX studio

Het grote voordeel van de drie methoden (**Define Measure**, **Define Dependent Measure**, en **Define and Expand Dependent Measure**) in DAX Studio is dat je measures kunt aanpassen en testen zonder permanente wijzigingen in je datamodel.

Dit stelt je in staat om experimenten uit te voeren, afhankelijkheden te onderzoeken, en direct de impact van wijzigingen te analyseren, wat handig is voor het optimaliseren van je berekeningen.

```
1 DEFINE
2 ---- MODEL MEASURES BEGIN ----
3 MEASURE 'Measure Table'[TotalSales LY] =
4 CALCULATE(
5     [# TotalSales],
6     SAMEPERIODLASTYEAR( 'Date'[Date])
7 )
8 ---- MODEL MEASURES END ----
9
10 EVALUATE
11 ROW( "Test", 'Measure Table'[TotalSales LY])
```

Log	Results	History
	Test	
	219.725.338.467,22	

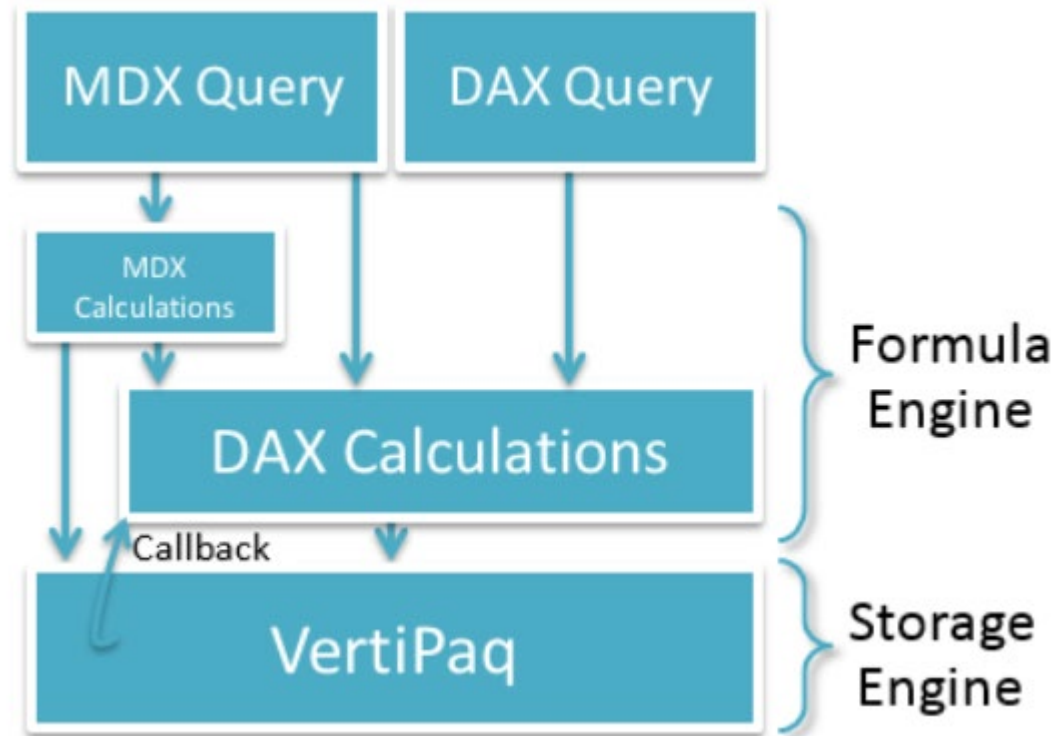
```
1 DEFINE
2 ---- MODEL MEASURES BEGIN ----
3 MEASURE 'Measure Table'[TotalSales LY] =
4 SUM(Sales[Quantity])
5 ---- MODEL MEASURES END ----
6
7 EVALUATE
8 ROW( "Test", 'Measure Table'[TotalSales LY])
```

Log	Results	History
	Test	
	666.399.295,00	



# DAX Performance

Advanced



# Formula Engine (FE)

- **De eerste stop van elke DAX query:**
  - De Formula Engine verwerkt de berekeningen en query's die door de gebruiker zijn opgesteld, zoals DAX (Data Analysis Expressions) formules in Power BI of SSAS-tabular modellen.
  - Wanneer een gebruiker een query indient, bepaalt de Formula Engine welke berekeningen moeten worden uitgevoerd en welke gegevens hiervoor nodig zijn.
  - De Formula Engine maakt gebruik van de VertiPaq Engine om snel toegang te krijgen tot de benodigde gegevens die in het geheugen zijn opgeslagen. Het voert berekeningen uit en retourneert de resultaten aan de gebruiker.
- **De Formula Engine handelt de complexe berekeningen**
- **Is wel single-treaded**

# Storage Engine (SE)

- Kan alleen simpele berekeningen uitvoeren
- Zoals SUM, AVG, MIN, MAX
- Kan query's uitvoeren op de database
- Is ten opzichten van de FE wel multithreaded
- Is ontwikkeld voor de snelheid

# Hoe werkt de SE samen met de FE?

De Storage Engine werkt samen met de Formula Engine om efficiënt gegevens op te halen en query's te verwerken in VertiPaq:

- Verzoek van de Formula Engine: Wanneer de gebruiker een query indient, bepaalt de Formula Engine welke gegevens nodig zijn voor de berekeningen en stuurt een verzoek naar de Storage Engine.
- Gegevensophalen door de Storage Engine: De Storage Engine haalt de gevraagde gegevens op uit het geheugen of de schijf, gebruikt compressie en andere technieken om de gegevens snel beschikbaar te maken.
- Terugkoppeling via Callback: Zodra de benodigde gegevens beschikbaar zijn, roept de Storage Engine een callbackfunctie aan in de Formula Engine. Hierdoor kan de Formula Engine doorgaan met de queryverwerking zonder te hoeven wachten op de dataverwerking.

Deze samenwerking zorgt voor een efficiënte verwerking van grote datasets en verbetert de prestaties van analytische query's.

# Callback in VertiPaq

In VertiPaq wordt een callback gebruikt om de samenwerking tussen de Formula Engine (FE) en de Storage Engine (SE) te optimaliseren tijdens het verwerken van query's:

- **Verwerking van Query:** De Formula Engine bepaalt welke gegevens nodig zijn en stuurt een verzoek naar de Storage Engine.
- **Asynchrone Oproep:** Terwijl de Storage Engine de benodigde gegevens ophaalt, blijft de Formula Engine doorwerken aan andere taken.
- **Callback Actie:** Zodra de Storage Engine de gegevens klaar heeft, roept het de callbackfunctie aan, waardoor de Formula Engine weet dat het kan doorgaan met de queryverwerking.

# Callback voordelen en nadelen

## Voordelen:

- **Efficiëntie:** Geen wachttijd; gelijktijdige verwerking.
- **Snellere Query's:** Snellere respons en betere prestaties door optimalisatie van de datastroom.

## Nadelen van Callbacks in VertiPaq:

- **Afhankelijkheden:** Callbacks creëren afhankelijkheden tussen verwerkingstappen.
- **Prestatie-overhead:** Te veel asynchrone operaties kunnen leiden tot prestatieverlies door extra beheerkosten en context-switching.



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



**Peter van den Bos**  
Business Intelligence Consultant

✉ [peter@dutchbigEEK.nl](mailto:peter@dutchbigEEK.nl)

☎ +31 6 13760795

[dutchbigEEK.nl](https://dutchbigEEK.nl)