sqlbits 2025

18-21 JUNE, LONDON EXCEL

Optimizing Power BI: Enhancing Performance Through Data Modeling

Peter van den Bos

Data Modeling & DAX Specialist
Freelance Trainer & Speaker
Founder @ DutchBiGeek

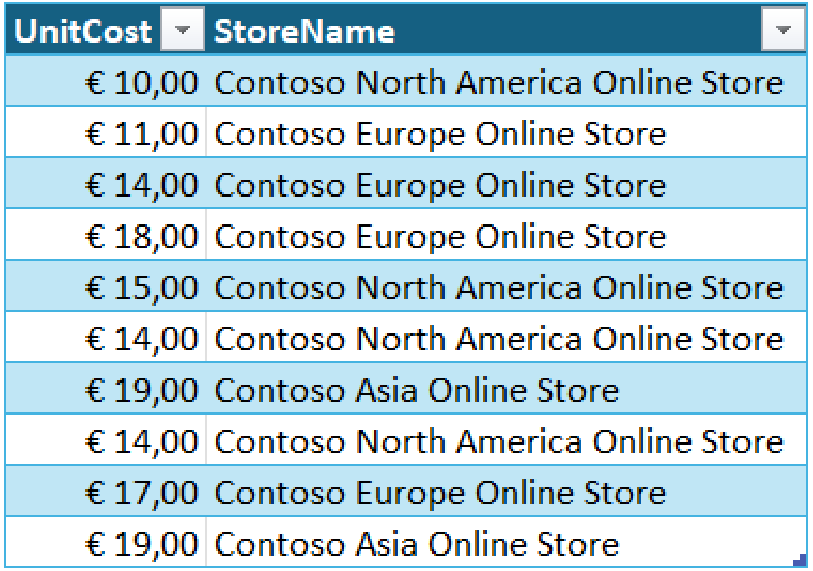
- 🌐 dutchbigeek.nl
- 🔗 linkedin.com/in/petervandenbos
- github.com/PetervandenBos/
- 📅 sessionize.com/peter-van-den-bos
- ▶️ youtube.com/@DutchBIgeek
- ✉️ peter@dutchbiggeek.nl

🧠⚡ **Geek by design**
Because performance isn't optional

Why is the Data Model Important?

- The data model is the foundation of Power BI.
- A well-designed data model improves performance and speeds up queries.
- 80% of performance issues are related to the data model.
- A better data model makes DAX queries easier and more efficient.
- Reduces data redundancy and minimizes errors.

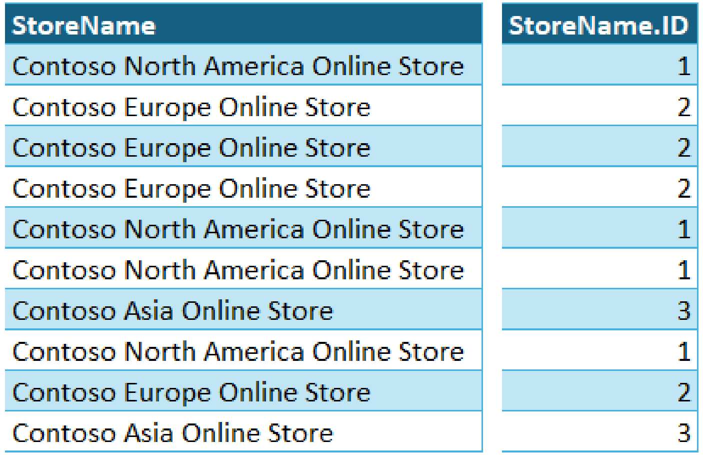
VertiPaq Compression



UnitCost	StoreName
€ 10,00	Contoso North America Online Store
€ 11,00	Contoso Europe Online Store
€ 14,00	Contoso Europe Online Store
€ 18,00	Contoso Europe Online Store
€ 15,00	Contoso North America Online Store
€ 14,00	Contoso North America Online Store
€ 19,00	Contoso Asia Online Store
€ 14,00	Contoso North America Online Store
€ 17,00	Contoso Europe Online Store
€ 19,00	Contoso Asia Online Store

Dictionary Encoding

- Replaces unique text values with numeric keys
- Saves memory: text → integer
- Values stored once in a dictionary, referenced via keys
- Especially effective for low-cardinality columns



StoreName	StoreName.ID
Contoso North America Online Store	1
Contoso Europe Online Store	2
Contoso Europe Online Store	2
Contoso Europe Online Store	2
Contoso North America Online Store	1
Contoso North America Online Store	1
Contoso Asia Online Store	3
Contoso North America Online Store	1
Contoso Europe Online Store	2
Contoso Asia Online Store	3

Run-Length Encoding (RLE)

- Compresses consecutive repetitions of the same value
- Stores as (value + count)
- Only works well when the column is sorted
- Applied only if it actually saves memory

StoreName.ID
1
2
2
2
1
1
3
1
2
3

StoreName
1
1
1
1
2
2
2
2
3
3

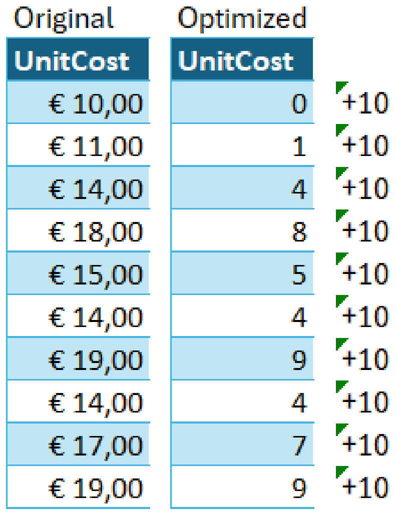
StoreName.ID	Count	Rows
1	0	4
2	4	4
3	7	2

Value Encoding (Bit-Packing)

- VertiPaq chooses the smallest number of bits per column

- Smaller range of values = fewer bits needed

- Works after dictionary encoding or on numeric columns



Original	Optimized	
UnitCost	UnitCost	
€ 10,00	0	+10
€ 11,00	1	+10
€ 14,00	4	+10
€ 18,00	8	+10
€ 15,00	5	+10
€ 14,00	4	+10
€ 19,00	9	+10
€ 14,00	4	+10
€ 17,00	7	+10
€ 19,00	9	+10

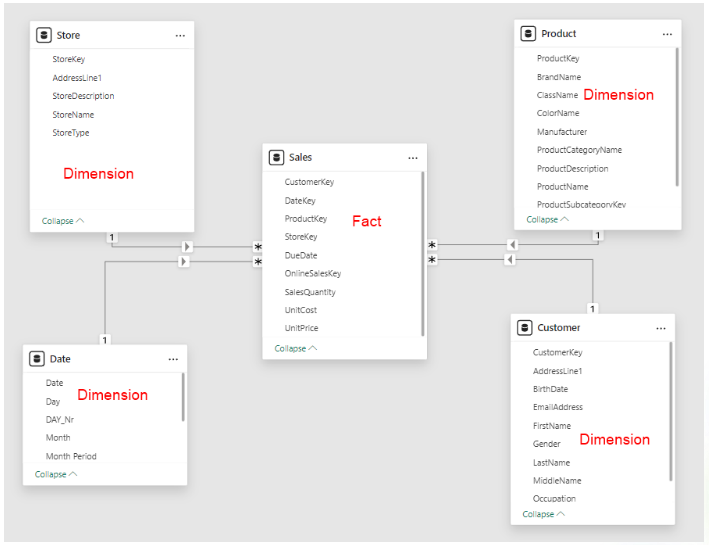
Star Schema

Fact Tables:

- Contain facts, representing an event with dimensions.
- A sale includes a product, a customer, and a date.
- Metrics that can be aggregated to gain insights.

Dimension Tables:

- Descriptive attributes of entities such as a product, customer, employee, or patient.
- Dimensions have attributes like color, category, manufacturer, or price.



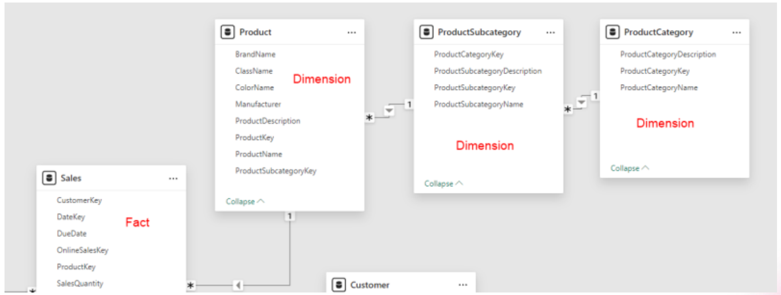
Snowflake-model

The Snowflake Schema is a Variant of the Star Schema

The difference is that dimension tables that are related to each other are connected.

For example:

- **Product**
- **Product Subcategory**
- **Product Category**



What does an Optimized Data Model consist of?

The model consists of facts and dimensions.

- A fact table contains values that you can calculate, such as:
  - Revenue, purchase date, sold products, etc.

- A dimension table contains values you want to filter, such as:
  - Year, month, manufacturer, customer, etc.

Relationships and Keys

To create relationships between tables, keys are used.

Primary Key:

- A unique value that appears only once in a dimension table.

Foreign Key:

- Used in a fact table to indicate how often, for example, a product has been sold.
- The foreign key can appear multiple times in a fact table, as a product is typically sold more than once.

Relationships and Filtering

A relationship must be created between facts and dimensions. Possible relationships include:

- One-to-many
- One-to-one
- Many-to-many

You can also choose the filter direction:

- Single
- Both

Multiple Fact Tables

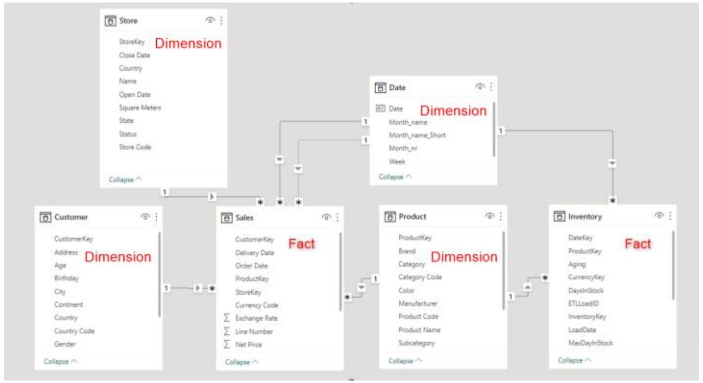
It is possible to use multiple fact tables in your model.

These fact tables may not have much in common with each other

- Inventory and sales.

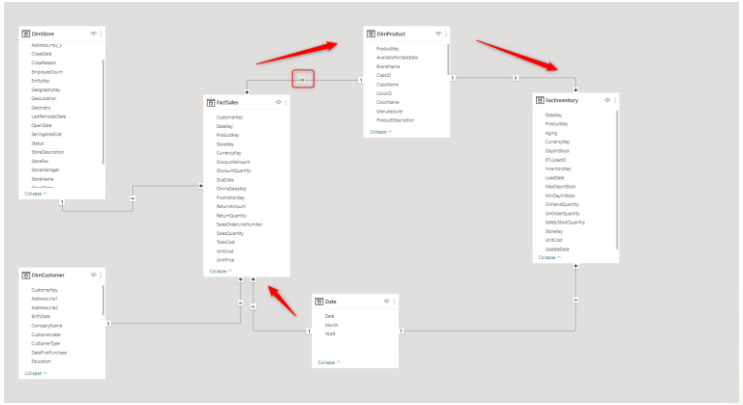
However, it is necessary to have some dimension tables that are connected to the fact tables

- date and product.



Ambiguity

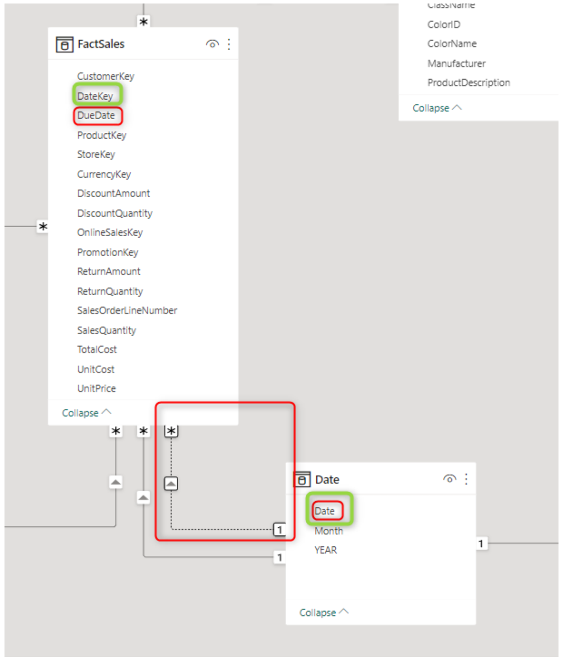
- When a model includes multiple fact tables and uses bidirectional relationships, there is a risk of ambiguity.
- This means the model may not know which path to follow when filtering.
- If it can filter through multiple tables, it may display incorrect values.



Multiple date tables

- When comparisons need to be made between dates, multiple date tables are sometimes used.
- The reason for this is that you can only have one active relationship between tables.
- This adds extra data to your model, which can be inefficient.
- It is better to use the USERELATIONSHIP function in DAX.
- This allows for easier comparisons, and the relationship only becomes active when the measure is used.

Example of multiple date tables



```
€ Total Sales DueDate =
CALCULATE(
    SUMX(
        FactSales,
        FactSales[SalesQuantity] * FactSales[UnitPrice]
    ), USERELATIONSHIP('Date'[Date],FactSales[DueDate])
)
```

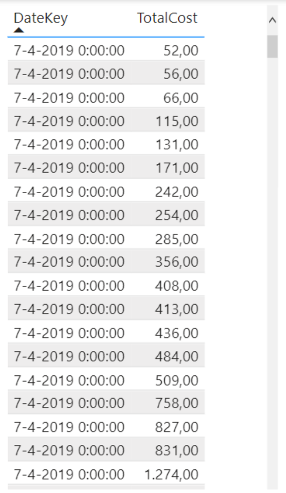
YEAR	Month	€ Total Sales	€ Total Sales DueDate
2020	1	33.009.186,00	37.071.480,00
2020	2	35.875.667,00	32.881.523,00
2020	3	37.972.129,00	39.884.837,00
2020	4	23.246.283,00	24.885.072,00
2020	5	25.203.379,00	25.347.549,00
2020	6	22.731.355,00	22.762.326,00
2020	7	31.128.116,00	29.408.149,00
2020	8	26.639.125,00	28.419.751,00
2020	9	27.177.507,00	25.827.772,00
2020	10	29.276.942,00	30.009.064,00
2020	11	24.816.322,00	25.134.687,00
2020	12	28.318.035,00	27.652.516,00
Total		**345.394.046,00**	**349.284.726,00**

Level of detail in the tables

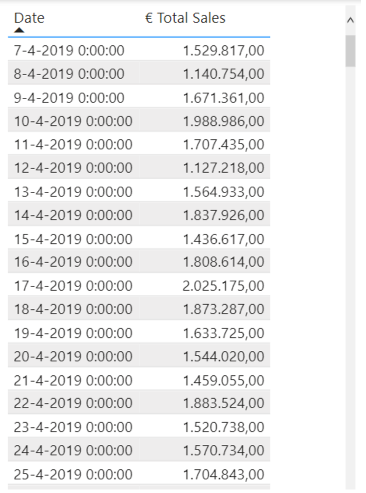
The level of detail at the lowest level in your table.

Consider what is needed at the lowest level of the table, for example:

- **What is the lowest level of each individual purchase?**
- **Is it sufficient to store the revenue per day?**



DateKey	TotalCost
7-4-2019 0:00:00	52,00
7-4-2019 0:00:00	56,00
7-4-2019 0:00:00	66,00
7-4-2019 0:00:00	115,00
7-4-2019 0:00:00	131,00
7-4-2019 0:00:00	171,00
7-4-2019 0:00:00	242,00
7-4-2019 0:00:00	254,00
7-4-2019 0:00:00	285,00
7-4-2019 0:00:00	356,00
7-4-2019 0:00:00	408,00
7-4-2019 0:00:00	413,00
7-4-2019 0:00:00	436,00
7-4-2019 0:00:00	484,00
7-4-2019 0:00:00	509,00
7-4-2019 0:00:00	758,00
7-4-2019 0:00:00	827,00
7-4-2019 0:00:00	831,00
7-4-2019 0:00:00	1.274,00



Date	€ Total Sales
7-4-2019 0:00:00	1.529.817,00
8-4-2019 0:00:00	1.140.754,00
9-4-2019 0:00:00	1.671.361,00
10-4-2019 0:00:00	1.988.986,00
11-4-2019 0:00:00	1.707.435,00
12-4-2019 0:00:00	1.127.218,00
13-4-2019 0:00:00	1.564.933,00
14-4-2019 0:00:00	1.837.926,00
15-4-2019 0:00:00	1.436.617,00
16-4-2019 0:00:00	1.808.614,00
17-4-2019 0:00:00	2.025.175,00
18-4-2019 0:00:00	1.873.287,00
19-4-2019 0:00:00	1.633.725,00
20-4-2019 0:00:00	1.544.020,00
21-4-2019 0:00:00	1.459.055,00
22-4-2019 0:00:00	1.883.524,00
23-4-2019 0:00:00	1.520.738,00
24-4-2019 0:00:00	1.570.734,00
25-4-2019 0:00:00	1.704.843,00

Many-to-many relationships

Many-to-many (M:M) relationships

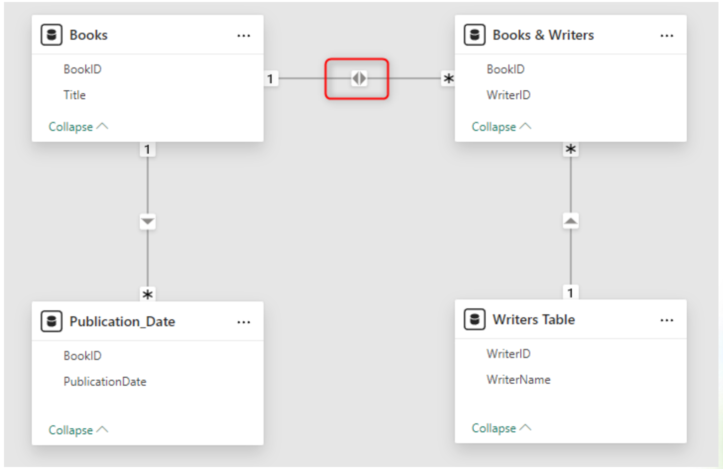
Example: Books and writers

How to solve this?

- Create a bridge table
- Make a table containing all keys

Pay attention to the direction of the cross-filter:

- Filtering only works from the 1 to * side



Importance of the Star Schema for VertiPaq

- Star schema boosts performance:
  - Faster queries by separating dimensions and facts.
  - Simplifies data for efficient VertiPaq processing.

- Proper data preparation is key:
  - Consistent data format ensures better compression and lower RAM usage.

- Small tweaks, big gains:
  - Minor adjustments can greatly improve performance.
  - Next: Key factors for optimizing compression.

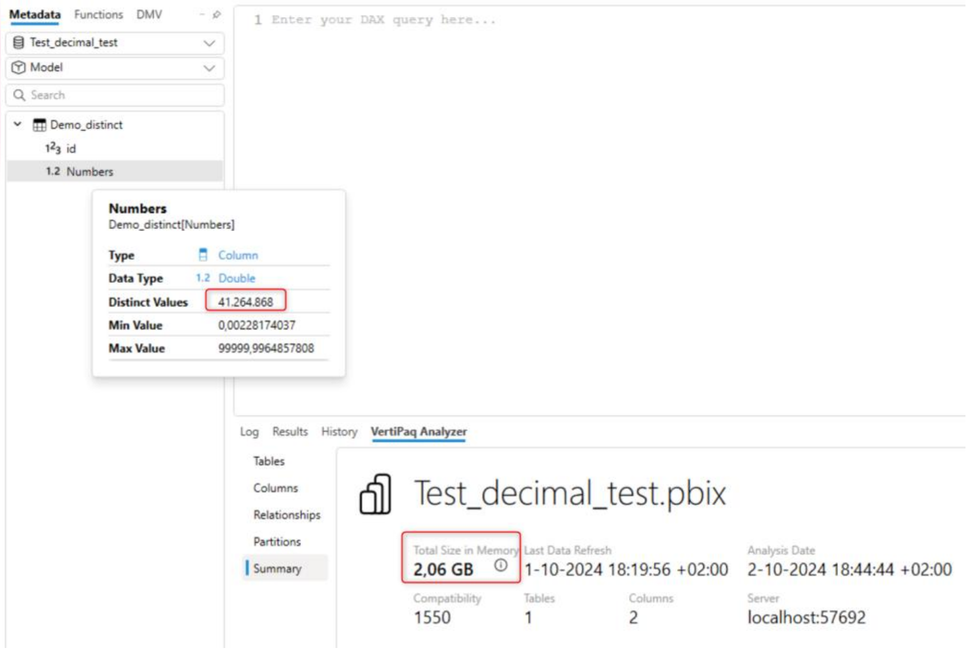
Importance of Choosing the Right Data Types

- Select the correct data types to optimize model performance
    - Understand data requirements and report needs

- Power Query defaults to "Decimal" for numeric columns:
    - May lead to unnecessary digits after the decimal point
    - Evaluate if fewer decimal places or whole numbers are sufficient

- Correct data type choice is crucial for VertiPaq compression

Example Decimal number



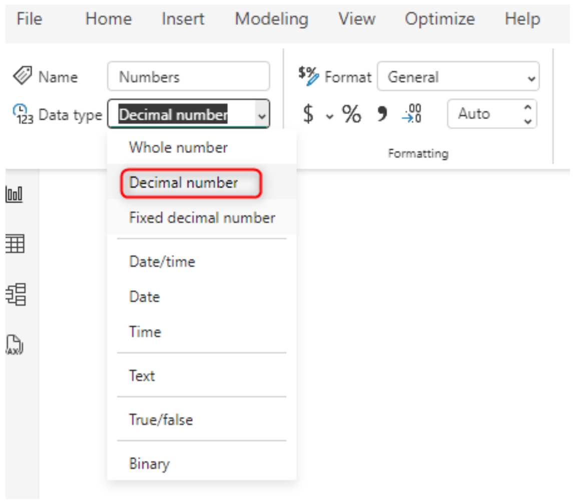
Test_decimal_test
Model
Search
Demo_distinct
id
1.2 Numbers

Numbers
Demo_distinct[Numbers]

Type	Column
Data Type	1.2 Double
Distinct Values	41.264.868
Min Value	0,00228174037
Max Value	99999,9964857808

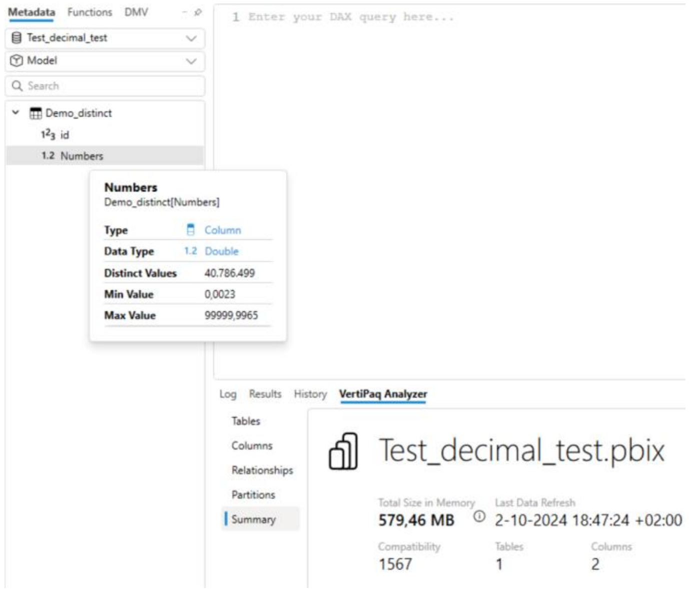
Test_decimal_test.pbix

Total Size in Memory	Last Data Refresh		Analysis Date
2,06 GB	1-10-2024 18:19:56 +02:00		2-10-2024 18:44:44 +02:00
Compatibility	**Tables**	**Columns**	**Server**
1550	1	2	localhost:57692



Name: Numbers
Data type: Decimal number
Format: General
Auto

Example Fixed Decimal number



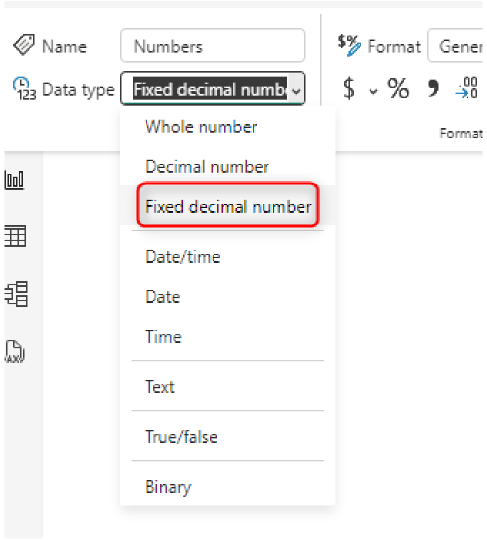
Numbers
Demo_distinct[Numbers]

Type	Column
Data Type	1.2 Double
Distinct Values	40.786.499
Min Value	0,0023
Max Value	99999,9965

Test_decimal_test.pbix

Total Size in Memory	Last Data Refresh
579,46 MB	2-10-2024 18:47:24 +02:00

Compatibility	Tables	Columns
1567	1	2


Name: Numbers
Data type: Fixed decimal number

Optimizing Data Loading for Compression

- Load only necessary data to achieve optimal compression

- Reduce the number of columns per table:
  - Fewer columns lead to more effective compression
  - VertiPaq sort order technique: Stores columns with the lowest cardinality first for better compression
  - More columns, especially with higher cardinality, reduce compression efficiency

- Limit to around 15 columns per table to maintain an efficient model

Calculated Columns: Pros and Cons

- Calculated columns are useful during development:
  - Allow for quick testing and validation without modifying data sources
  - Provide flexibility and speed up development

- Limitations of calculated columns in production:
  - Added after model compression, not compressed efficiently
  - Increase memory usage and reduce performance
  - Can undo optimization efforts by increasing model size

Best practices

- Always use a star schema, or a snowflake schema if necessary.
- A fact table contains values for calculations.
- A dimension table contains values for filtering.
- Avoid using bidirectional relationships.
- Include only the data you actually use.
- Determine the level of detail for your tables in advance.
- Choose the right data type for each column

If you have questions or insights, please contact me !

Peter van den Bos
Business Intelligence Consultant

peter@dutchbigeek.nl
+31 6 13760795
https://www.linkedin.com/in/petervdbos/

dutchbigeek.nl