

Optimize a Model for Performance in Power BI

Introduction to Performance Optimization

Performance optimization, also known as **performance tuning** involves making changes to the current state of the data model so that it runs more efficiently

Poor performance can be characterized by

From the report user's perspective,

report pages take longer to load

visuals take more time to update

resulting in a NEGATIVE user experience

9 TIMES OUT OF 10 – POOR PERFORMANCE is a DIRECT RESULT of:

- Bad data model
- Bad Data Analysis Expressions (DAX)
- or, a mix of the two

Optimizing data early, allows you to mitigate the negative impact of organizational and data growth on the performance of your data model.

A smaller-sized data model uses less resources (memory) and achieves a faster data refresh, calculations, and rendering of visuals in reports

Performance optimization then involves

minimizing the size of the data model

making the most efficient use of the data in the model

which includes:

- Ensuring that the correct data types are used
- Deleting unnecessary columns and rows
- Avoiding repeated values
- Replacing numeric columns with measures
- Reducing cardinalities
- Analysing model metadata
- Summarizing data where possible

Review Performance of Measures, Relationships, and Visuals

If your data model has:

- multiple tables
- complex relationships
- intricate calculations
- multiple visuals
- redundant data

a potential exists for poor performance

In order TO OPTIMIZE PERFORMANCE, you must FIRST:

Identify where the problem is coming from

find out which elements of the report and data model are causing the performance issue

Identify Report Performance Bottlenecks

In order to **achieve optimal performance**, you need to **create an efficient data model** that has **fast running queries and measures**

Hence,

review measures and queries in the data model to **ensure that you are using the most efficient way to get the results that you want**

Identify bottlenecks – when you identify the slowest query in the data model, you can focus on the biggest bottleneck first and establish a priority list to work through the other problems

Analyze Performance

You can use **Performance Analyzer** in Power BI to help you find out **how each of your report elements are performing when users interact with them**

e.g.,

you can find out **how long it takes for a particular visual to refresh when it is initiated by a user interaction**

will **help you identify elements that are contributing to your performance issues**

BEFORE you run PERFORMANCE ANALYZER

to ensure that you **get the most accurate results in your analysis(test):**

- **Clear your visual cache**
 - When you load a visual, you CAN'T CLEAR THIS VISUAL CACHE WITHOUT CLOSING POWER BI
 - **To ensure that you have a clear visual cache**
 - => **Add a blank page to the .pbix file**
 - => **WITH THE PAGE SELECTED**, save and close the file
 - => **Reopen the .pbix file**, it will open on the blank page – **clear visual cache**
- **Clear data engine cache**
 - When a **query is run, the results are cached** – so the **results of your analysis will be misleading**.
 - You need to **CLEAR THE DATA CACHE**
 - => **Restart Power BI Desktop**

When you have **cleared both cache** and **opened the Power BI Desktop file ON THE BLANK PAGE**

=> go to the **View tab**, and select **Performance Analyzer**

To **begin the analysis process**, select **Start recording**

=> select **the page of the report that you want to analyze**

=> and **interact with the elements of the report that you want to measure**

You will see the **results of your interactions display** in the **Performance analyzer pane as you work**

When you are finished, select the **Stop button**

Review Results

You can **review the results** of the performance test in the **Performance analyzer pane**.

In order to **review the tasks in order of duration, longest to shortest**:

- **right-click the Sort icon**, next to the **Duration (ms)** column header
- select **Total time** in **Descending** order

The log information for each visual shows **how much time it took (duration) to complete a category of tasks**:

- **DAX Query**
 - The **time it took for the visual to send the query**, along with **the time it took Analysis Services to return the results**
- **Visual Display**
 - The **time it took for the visual to render on the screen**, including the **time required to retrieve web images or geocoding**
- **Other**

Resolving Issues and Optimizing Performance

This area covers improvements in areas such as **visuals, DAX query, or other elements in your data model**

The following information provides guidance on what to look for and **what changes you can make**

Visuals

If visuals are the **bottleneck leading to poor performance**

Try to **find a way to improve performance with minimal impact to user experience**

In general, **FEWER VISUALS lead to BETTER PERFORMANCE**

Hence, asks yourself if a visual is **TRULY NECESSARY?**

If no, **remove it**

Improve Performance

Consider other ways of providing additional detail, such as:

drill-through pages and **report page tooltips**

Examine the **number of fields in each visual**

Consider **reducing the number of fields in each visual**

Too much data can appear crowded and lose clarity

Ask yourself: **do you really need all this data in a visual?**

Data Analysis Expressions (DAX) Queries

When you examine the results in the **Performance analyzer pane**, you can see:

How long it took the Power BI Desktop engine to evaluate each query (in milliseconds)

You can use **DAX studio** to investigate **DAX queries in more detail**

The **usual suspect**: **You are USING THE WRONG DAX Function**

or

INEFFICIENTLY CALCULATING for the measure

Improve Performance

You could try substituting the current DAX function with OTHER DAX functions that keeps the same functionality

Replace the old DAX function with a NEW DAX FUNCTION that KEEPS THE FUNCTIONALITY and remeasure the performance to determine whether duration time has improved or not

Relationships

Review the relationships between the tables to ensure that you have established correct relationships

Check that relationship cardinality properties are correctly configured

An example:

a one-sided column that contains unique values might be incorrectly configured as a many-side column

Columns

Best practice NOT IMPORT columns that YOU DO NOT NEED

Deal with them at the source when loading data into Power BI Desktop
Hence, BEFORE IMPORTING into Power Query Editor

However if this is not possible, then use Power Query Editor to examine each column
determine whether if each column is needed
and
identify the benefits of adding each column to the data model

If a column adds no value then REMOVE IT from the data model

Consider a column as unnecessary and remove it when you WILL NOT use the column in a particular relationship

Improve Performance

By removing unnecessary columns, you are reducing the size of the data model, this means that you have a smaller file size and a faster refresh time

Report performance also increases as the dataset only includes relevant data

Reducing Cardinality to Improve Performance

Cardinality

A term that is used to describe the uniqueness of the values in a column

Hence, a lot of repeated values (low distinct value count) will have a low level of cardinality
conversely, a lot of unique values (high unique value count) has a high level of cardinality

Also used in the context of relationships between two tables,
where it describes the direction of a relationship

In general, **having**

FEWER COLUMNS in a column leads to **BETTER MODEL PERFORMANCE**

LOWER Cardinality leads to a **more optimized performance**

hence, **important to reduce data to your model, especially for large models**

Summarization

MOST EFFECTIVE tool for **reducing model size** is:

USING A SUMMARY TABLE from the **data source**

an example scenario:

Your organization has two tables – one that **has all the transactional data** and another that **has the summarized aggregation of transactions by date**

PULL IN THE AGGREGATED COLUMN DATA as it is better due to less

Another technique is to **set the Storage Mode property for larger fact-type tables to DirectQuery**

Use Q&A on a Dashboard

With this feature, **users can ask questions about their data BOTH in reports and dashboards and on dashboards**