zebra bi

Power BI Performance Optimization Checklist:

Top 5 things to speed up your data model:

1. Do your data prep
2. Implement a clean star schema
3. Push calculations down the BI stack
4. Choose the right connection type (Import > Live > Composite > DirectQuery)
5. Simplify RLS

How to speed up your reports and visuals through design:

1. Reduce the number of visuals per page
2. Implement an efficient report layout pattern (Overview > Details > More details ...)

Extra: Lazy load visuals with bookmarks and hidden visuals (**more info**)

How to analyze your performance:

Aim for 0.5-1sec total load time per page.

1. Performance analyzer in Power BI Desktop
2. Run the analyzer and copy the query to **DAX Studio**
3. Export logs and analyze the details

More Details

Connection type:

DO's:

- Import
- Live Connection
- (Optional: Composite model)

DON'T's:

- Direct Query

Data model:

DO's: (Will result in fast data model)

- Star schema
- 1-to-many relationships
- Include only necessary elements
- Simple DAX formulas
- Have a separate date table (also mark calendar/date tables as date table)

DON'T's: (Will result in slow data model)

- Have too many tables (multiple fact tables, 10+ dimensions)
- Tons of joins
- Bridge tables
- Complex DAX formulas
- Many-to-many relationships
- Include unused tables & columns
- Use auto-date table

Key reasons for messy data models:

1. Lack of proper data prep/ETL
2. Separated fact tables (e.g. Sales, Plan, Outlook)
3. Lookup tables instead of proper hierarchies/dimensions
4. Including all possible columns instead of only required ones
5. Unnecessarily complex row level security (RLS)

This forces you to use complex DAX expressions, bridge tables and many-to-many relationships.

This is **"DATA KNITTING"** not DATA MODELING!

Do your data prep!

1. **Combine/append** different fact tables into one main fact table
2. **Transform** your hierarchies and attributes into (single) table dimensions
3. **Filter out/remove** any columns that will not be used

Should be done in Power Query or at source.

Types of data models in Power BI:

- Flat table
- **Star schema** = BEST
- Snowflake schema
- Messy model

Note: In theory the flat table can be faster, but DAX is optimized for star schema and in most real case scenarios Star schema will outperform flat tables in terms of performance (and it's much less RAM & storage expensive) (more on **star schema vs. flat table**)

Star schema

Fact Table:

1. Measures
    a) Implicit (hide)
    b) Explicit (use only these)
2. Foreign keys
    a) Integer only!
    b) As short as possible
    c) Hide foreign keys from report view – so the fact table shows only the explicit measures for the users.
3. Reduce granularity!

Dimensions:

- Hide foreign keys (IDs)
- Only necessary columns!
- Calendar key: Date
- Split Date/Time into Date and Time columns
- Avoid calculated columns
- Turn off Power BI's auto-date table
- **Relationships to fact table:** 1-to-many

Push calculations down the BI stack

- Calendar in PowerQuery or source database (not in DAX)
- Only non-additive measures in DAX (use DIVIDE)

Row level security (RLS)

- Can be a performance killer
- Star schema is a must
- Simplify rules, no lookups, no joins (apply to dimensions if possible)
- Consider alternative designs:
    - Split the model and reports for different audiences, publish to separate workspaces

Recap:

- Connection type: Import >= Live > DirectQuery (even with DQ optimized source, 4-5x slower; DQ over Import only if you need real time data); avoid DirectQuery as each visual/filter action/interaction will trigger a query back to the source – slows down the performance a lot.
- Data model: Star schema
- Lean (long not wide) fact tables
- Reduce precision (number of decimals)
- Convert numbers to integer
- If you need calculated columns add them in Power Query (better to avoid them altogether)
- Take only what you need
- Report layout pattern: Landing page > details > more details
- Reduce the number of visuals per page
- Push calculations down the BI stack
- Limit row level security (RLS) logic

Additional materials & information:

1. **DAX Studio download:**
   https://daxstudio.org/

2. **SQLBI:**
   https://www.sqlbi.com/articles/power-bi-star-schema-or-single-table/

3. **SQLBI: Optimizing DAX SWITCH - Part 1:**
   https://www.sqlbi.com/articles/understanding-the-optimization-of-switch/

4. **SQLBI: Optimizing DAX SWITCH - Part 2:**
   https://www.sqlbi.com/articles/optimizing-if-and-switch-expressions-using-variables/

5. **Phil Seamark (MSFT):**
   https://dax.tips/2019/12/24/use-bookmarks-to-lazyload-visuals/

6. **Daniel Otykier - Tabular Editor:**
   https://tabulareditor.com/downloads/

7. **Daniel Otykier - Best Practice Analyzer:**
   https://docs.tabulareditor.com/te2/Best-Practice-Analyzer.html

8. **Michael Kovalsky (MSFT) - Best Practice Analyzer Rules:**
   https://powerbi.microsoft.com/en-au/blog/best-practice-rules-to-improve-your-models-performance/

9. **Michael Kovalsky (MSFT) - Best Practice Analyzer Rules v1.1:**
   https://powerbi.microsoft.com/en-my/blog/best-practice-rules-to-improve-your-models-performance-and-design-v1-1/

Performance
(Best practice analyzer rules by Michael Kovalsky)

- Avoid bi-directional relationships against high-cardinality columns
- Avoid excessive bi-directional or many-to-many relationships
- Avoid snowflake schema architecture
- Do not use floating point data types
- Large tables should be partitioned
- Limit row level security (RLS) logic
- Many-to-many relationships should be single direction
- Minimize Power Query transformations
- Model should have a date table
- Model using Direct Query and no aggregations
- Reduce number of calculated columns
- Reduce usage of calculated columns that use the RELATED function
- Reduce usage of calculated tables
- Reduce usage of long-length columns with high cardinality
- Remove auto-date table
- Remove redundant columns in related tables
- Set IsAvailableInMdx to false on non-attribute columns
- Split date and time
- Date/calendar tables should be marked as a date table
- Unpivot pivoted (month) data