# Power BI Performance Tuning Workflow

PowerBI is cloud based analytics platform that provides self-service analytics at enterprise scale,

Unify data from many sources to create interactive, immersive dashboards and reports that provide actionable insights and drive business results.

One of the common problem statements is “Power BI report is running slow”. Broadly, the optimizing and tuning areas are in the following levels

* The data source(s)

On-premises or Azure. Structured and Unstructured

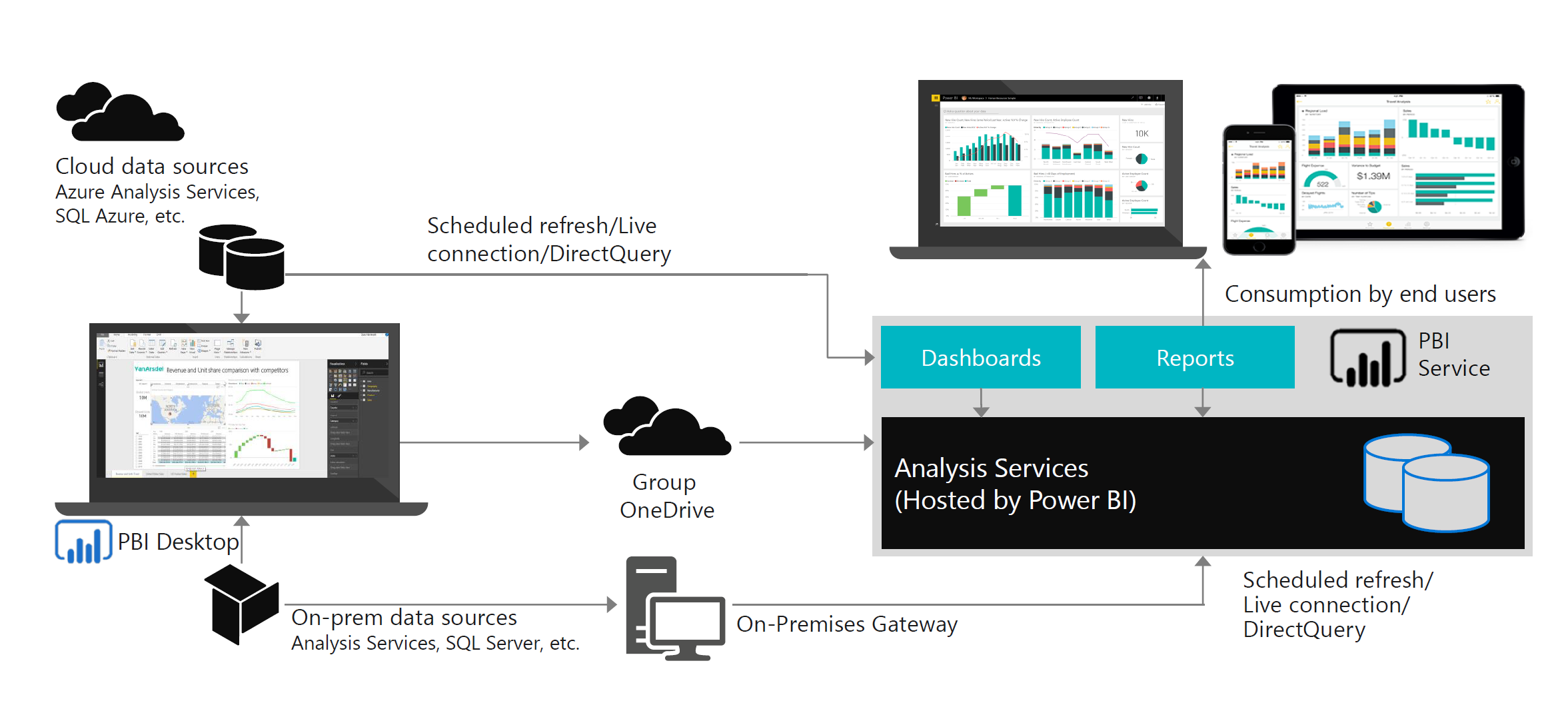
* The data model

Snowflake or Star Normalized or De-normalized.

* Visualizations, including dashboards, Power BI reports

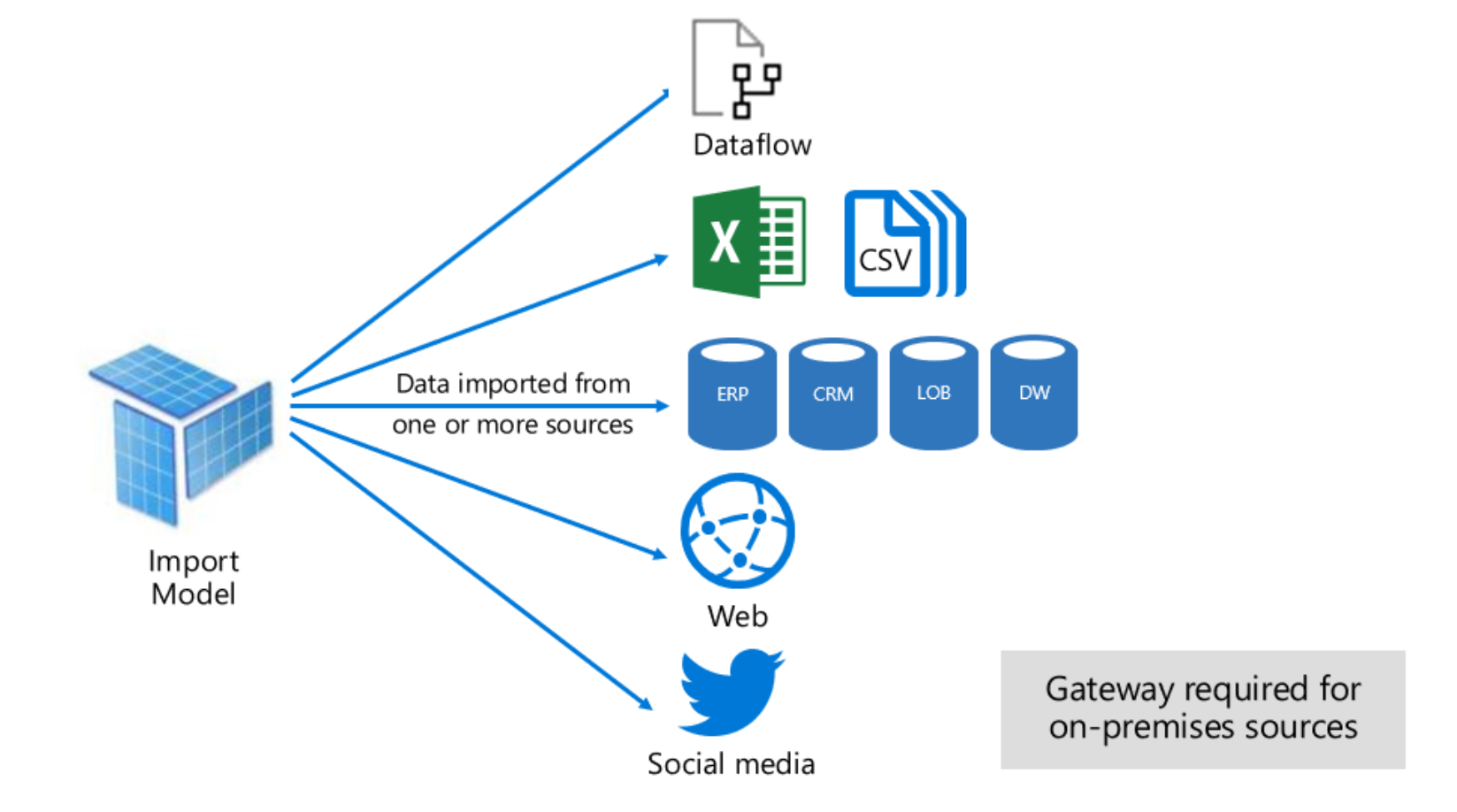
Table Matrix or Graph, Custom or Built-in

* Infrastructure including capacities, data gateways, and the network etc

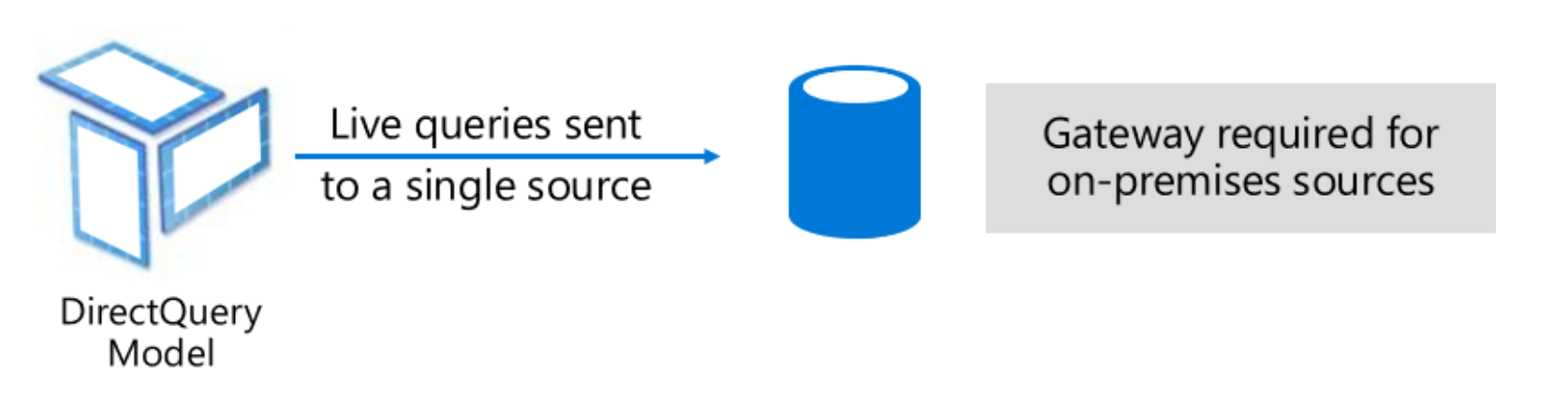


## Inappropriate use of Direct Query \Import

Import mode is the most common mode used to develop models. This mode delivers extremely fast performance thanks to in-memory querying and provides complete DAX support. Because of these strengths, it's the default mode when creating a new Power BI Desktop solution.



*Direct Query* mode is an alternative to Import model. The model in this mode consist only of metadata defining the model structure. When the model is queried, native queries are fired in the background to retrieve data from the underlying data source



### Considerations for Import

PowerBI engine offers around 5-10X compression depending on the data type and values. In shared capacity, the import model can store around 1GB of data but can scale using premium capacity that supports large datasets. However, avoid import model for two reasons

1. The dataset is very large and cannot fit in the capacity.
2. You need real time insights for reporting. Import model works on the refresh model and will not suffice data refresh of few seconds to minutes

### Considerations for Live Connection \ Direct Query

Just quite opposite to Import model. Whenever you need real time data for reporting or analytics or you cannot store the data in the power BI workspace ( shared or premium based on your environment), use direct query.

If your team or organization has already available tabular model, you can continue with AAS or SSAS tabular for live connection scenarios.

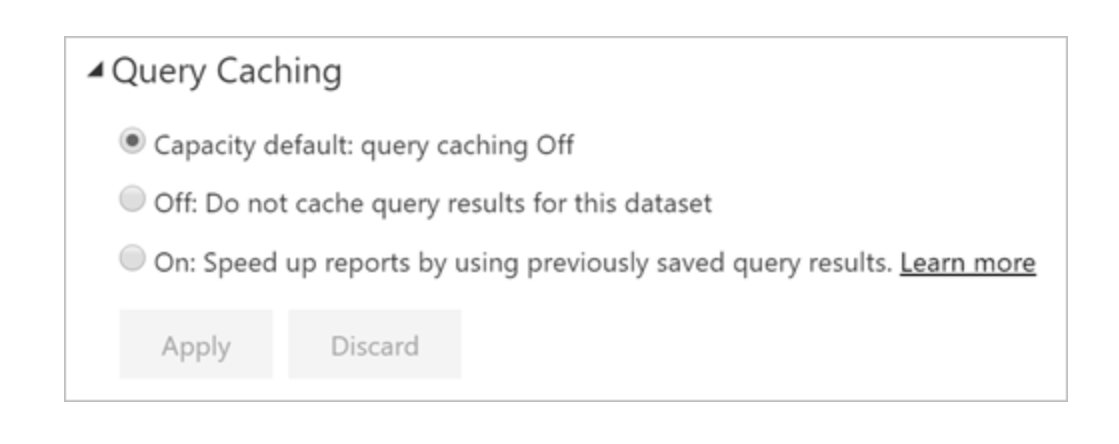
The limitations of direct query model are the extra network roundtrips to retrieve the dataset from the source and DAX coverage is limited and depends on the supportability w.r.t to the various data sources.

### Data Refresh

* Schedule your refreshes for less busy times, especially if your datasets are on Power BI Premium.
* Keep refresh limits in mind. If the source data changes frequently or the data volume is substantial, consider using DirectQuery/LiveConnect mode instead of Import mode if the increased load at the source and the impact on query performance are acceptable.
* Verify that your dataset refresh time does not exceed the maximum refresh duration.
* Use a reliable enterprise data gateway deployment to connect your datasets to on-premises data sources.
* Use separate data gateways for Import datasets and DirectQuery/LiveConnect datasets so that the data imports during scheduled refresh don't impact the performance of reports and dashboards on top of DirectQuery/LiveConnect datasets.
* Ensure that Power BI can send refresh failure notifications to your mailbox
* Configure incremental refresh for datasets are filtered by using Power Query date/time parameters with the reserved, case-sensitive names **RangeStart** and **RangeEnd**. This will minimize the memory footprint and reduce refresh time.

### Use Query Caching

Query caching is a premium feature that provides performance benefits when a dataset is accessed frequently and doesn't need to be refreshed often.

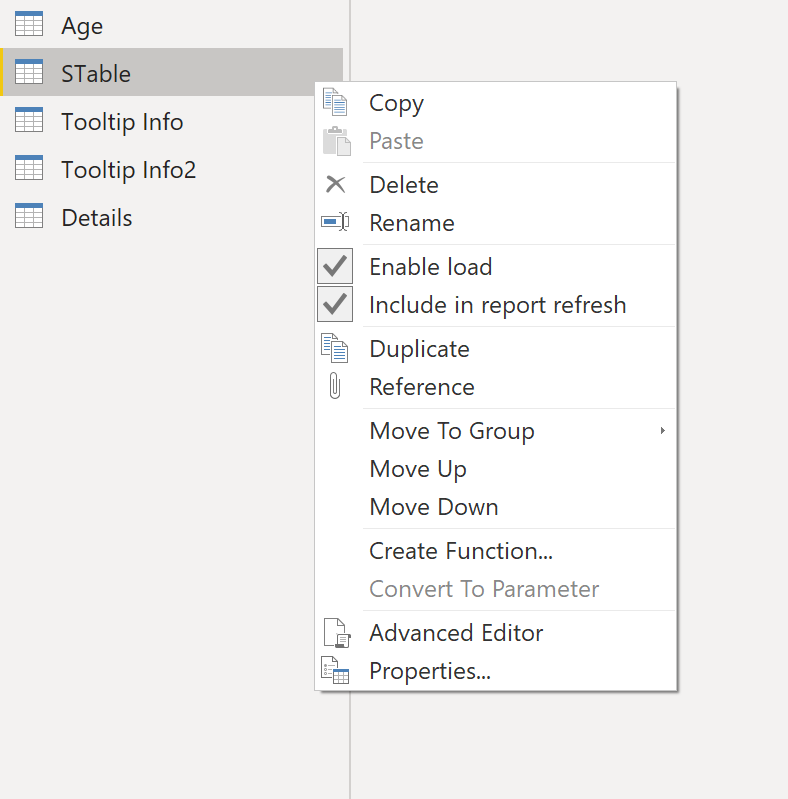


Query caching can also reduce load on your Premium/Embedded capacity by reducing the overall number of queries.

## Data Loading

### Modelling Improvements

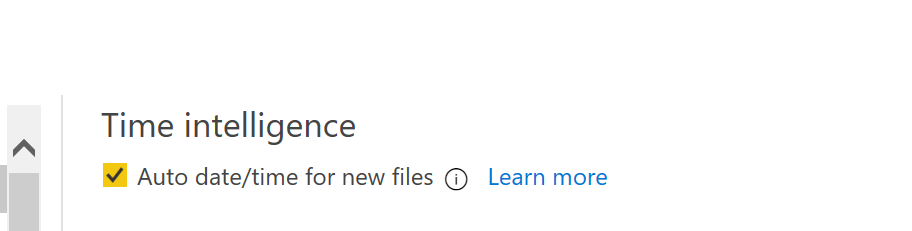
* In Powerbi model, there can be intermediate tables whether used as staging layer or custom queries that author hides to prevent access. The hidden tables consume memory. One way to improve performance is to disable load. Don’t get confused with “Hide in report view” option. This only removes from the view, but the table is loaded in the model and consumes memory.



* Don’t use Auto Date\Time

The Auto date/time is a data load option in Power BI Desktop. The purpose of this option is to support convenient time intelligence reporting based on date columns loaded into a model.

If data model has many datetime fields, then this setting can create several internal tables increasing the memory footprint of small models.



### GroupKind.Local

Groupkind.local performs faster than the default setting. When the data is sorted or in continuous fashion you can speed up your grouping-operations considerably.

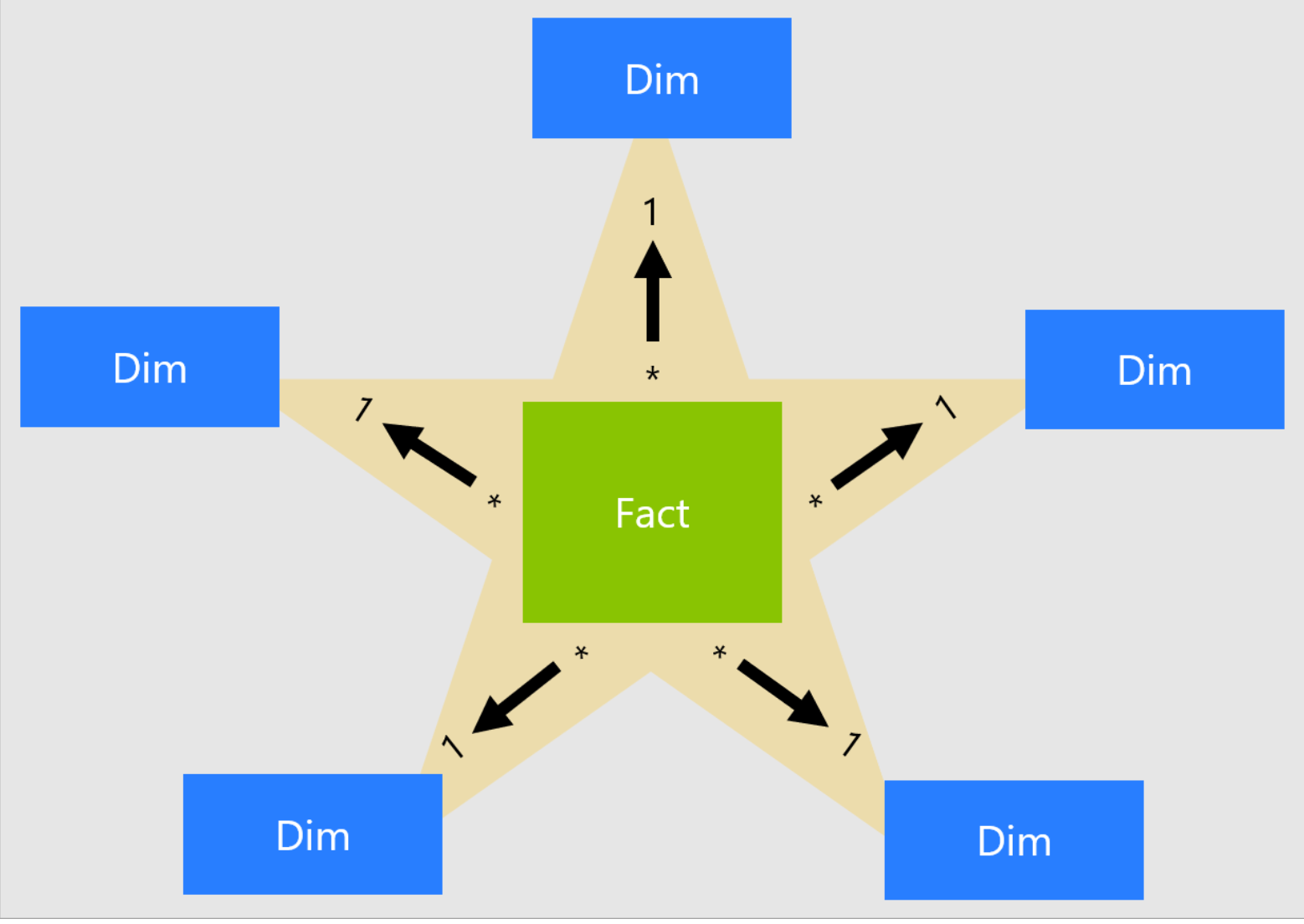
<https://docs.microsoft.com/en-us/powerquery-m/table-group>

## Data models

### Start Schema

Star schema is a mature modelling approach widely adopted by relational data warehouses. It requires modelers to classify their model tables as either dimension or fact.

A well-structured model design should include tables that are either dimension-type tables or fact-type tables. Avoid mixing the two types together for a single table. We also recommend that you should strive to deliver the right number of tables with the right relationships in place. It's also important that fact-type tables always load data at a consistent grain.



For more details, <https://docs.microsoft.com/en-us/power-bi/guidance/star-schema>

### Prefer integers than String

Integer are fix length datatype that uses run length encoding whereas string uses dictionary encoding. Also, if you sort the column data using the integer column, the level of compression will be significant higher. PowerBI segments boundary is 1 million.

### Avoid high precision / cardinality column

If you create models that have higher precision such as numeric or datetime etc, it reduces the compression ratio and increases higher time to load. Wherever possible, find ways to reduce precision without impacting business requirements.

Example: Date has a precision of milliseconds. So, choose the right type such as date if you only need date, Time if you need time only. Also, reduce the precision and round off to values wherever possible.

### Lean Model

The source tables or views might have many columns, but in your data model, the area of interest might be restricted for few rows.

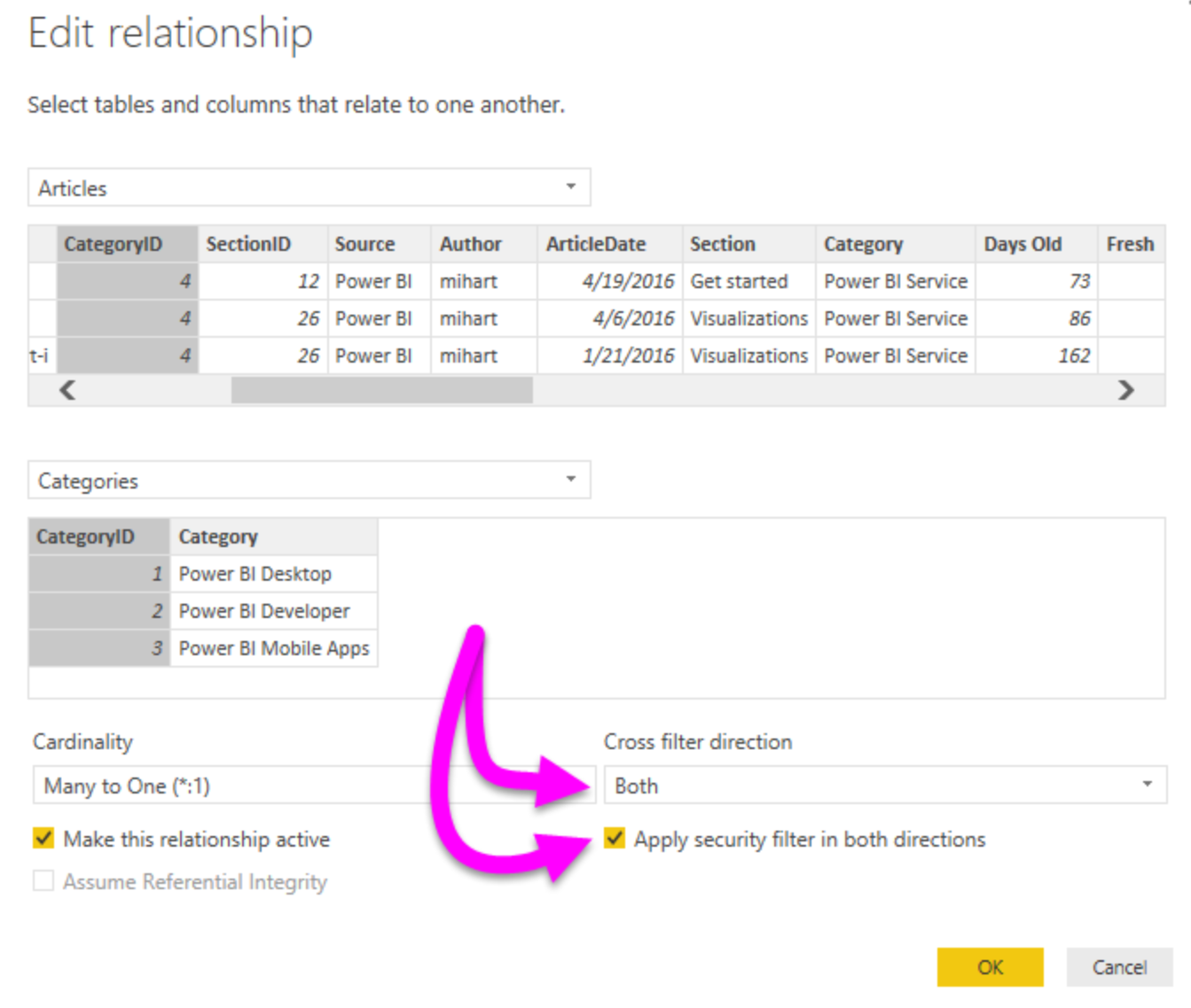
Example: Auditing column such as last modified etc are not useful for analytics purpose. S

Remove unwanted tables or columns in the model. This will reduce the model size and improve the refresh timings.

### Bi-Directional Relationships

The bidirectional cross-filtering feature is very powerful, and it allows us to solve complex models with more ease. However, if you have a model mostly with Bi-directional filters, any slicing or filtering activity might slow down because of the relationship propagation chain.

Also, if not modelled correctly, you might see inconsistent behaviour particularly in a snowflake architecture.



### Default Aggregations

Not all numeric columns are additive in nature such as surrogate columns or primary keys. By default, when any numeric column is placed on visual , Power BI will aggregate that column increasing the report compute time. The suggestion is to change the default aggregations to none for such type of columns

### Composite Model and Aggregations

Whenever possible, it's best to develop a model in Import mode. This mode provides the greatest design flexibility, and best performance. However, challenges related to large data volumes, or reporting on near real-time data, cannot be solved by Import models. In either of these cases, you can consider a DirectQuery model, providing your data is stored in a single data source that's [supported by DirectQuery mode](https://docs.microsoft.com/en-us/power-bi/power-bi-data-sources).

Further, you can consider developing a Composite model in the following situations.

* Your model could be a DirectQuery model, but you want to boost performance. In a Composite model, performance can be improved by configuring appropriate storage for each table. You can also add [aggregations](https://docs.microsoft.com/en-us/power-bi/desktop-aggregations). Both of these optimizations are discussed later in this article.
* You want to combine a DirectQuery model with additional data, which must be imported into the model. Imported data can be loaded from a different data source, or from calculated tables.
* You want to combine two or more DirectQuery data sources into a single model.

For more information, <https://docs.microsoft.com/en-us/power-bi/desktop-composite-models>

Important composite Best practice

* Set the storage mode to **DirectQuery** when a table is a fact-type table storing large data volumes, or it needs to deliver near real-time results
* Set the storage mode to **Dual** when a table is a dimension-type table, and it will be queried together with **DirectQuery** fact-type tables based on the same source

#### Aggregations

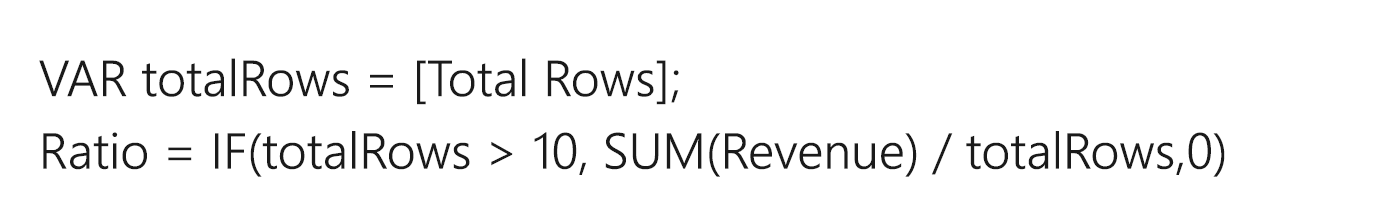
* You can add aggregations to DirectQuery tables in your Composite model. Aggregations are cached in the model, and so they behave as Import tables. Their purpose is to improve performance for "higher grain" queries.
* We recommend that an aggregation table follows a basic rule: Its row count should be at least a factor of 10 smaller than the underlying table.

## Calculations

### Variables vs Repeatable Measures

DAX Measures 

DAX with Variables



In the second scenario, the measure Total Rows in first scenario is executed twice whereas it is executed once using variables. Under such scenarios, variables can improve performance significantly.

### Handling Blanks

Sales (No Blank) =

IF (

ISBLANK([Sales]),

0,

[Sales]

)

Its recommended that measures return BLANK when a meaningful value cannot be returned.

This design approach is efficient, allowing Power BI to render reports faster. Also, returning BLANK is better because report visuals—by default—eliminate groupings when summarizations are BLANK.

### SELECTEDVALUE () vs HASONEVALUE ()

A common pattern is to use HASONEVALUE() to check if there is only one value present or a column after applying slicers and filters and then use VALUES(column name) DAX function to get the single value.

SELECTEDVALUE () performs both the above steps internally and gets the value if there is only one distinct value present or that column or returns blank in case there are multiple values available.

### SELECTEDVALUE () vs VALUES ()

VALUES () returns error in case it encounters multiple values. Normally, users handle it using Error functions which are bad for performance. Instead of using that, SELECTEDVALUE () must be used which is a safer function and returns blank in case of multiple values being encountered.

### DISTINCT () vs VALUES ()

PowerBI adds Blank value to the column in case it finds referential integrity violation. For direct query, Power BI by default adds blank value to the columns as it does not have a way to check for violations.

DISTINCT (): does not return blank which is added due to integrity violation. It includes blank only if it is in part of original data

VALUES (): It includes blank which is added by Power BI due to referential integrity violation

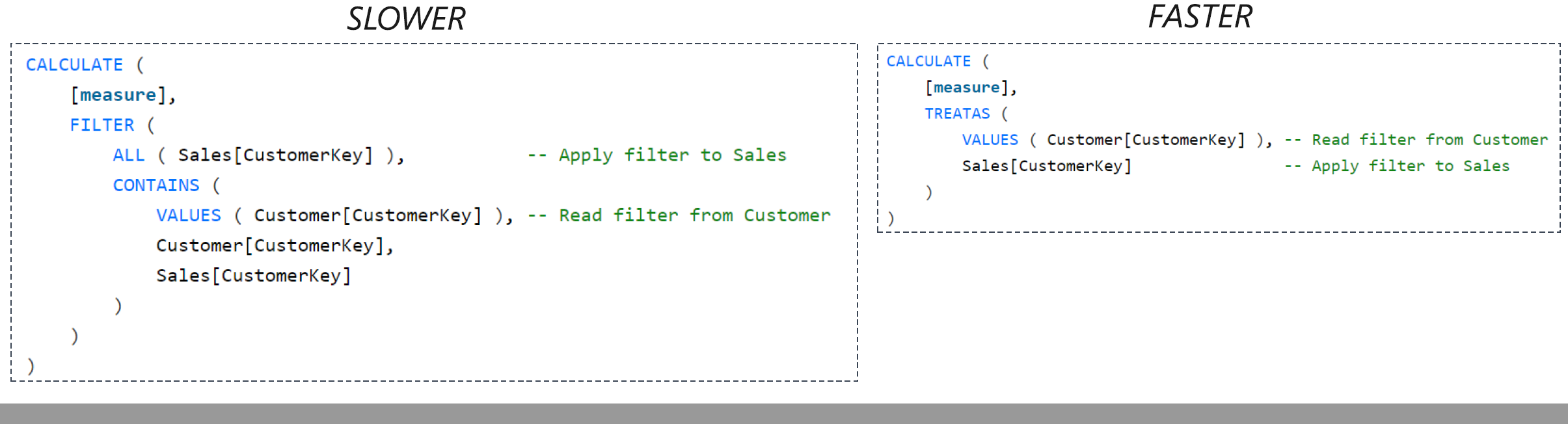
The usage of either of the function should be same throughout the whole report. Use VALUES () in the whole report if possible and blank value is not an issue.

### Avoid FORMAT in measures

Format function is a done in a single threaded formula engine and slows down the calculation for large number of string values

### Optimize Virtual Relationships using TREATAS ()

A virtual relationship is simulated with DAX and column equivalency. FILTER|CONTAINS executes slower than TREATAS.



### ISBLANK () vs Blank ()

Built-in function ISBLANK () to check for blank values is faster than using comparison operator “= Blank ()”

### Ratio Calculation efficiently

Use (a-b)/b with variable instead of (a/b)-1. The performance is same in both case usually. But under edge cases when both a and b are blank values, the former will return blank and filter out the date whereas latter will return -1 and increase the query space.

### DIVIDE () vs /

DIVIDE () function has an extra parameter which is returned in case of denominator value is zero. In case of scenarios that might have zero in denominator, it is suggested to use DIVIDE () as it will internally check id denominator is zero. It also checks for ISBLANK (). However, if there can be guarantee that the denominator will be non-zero, it is better to perform / because DIVIDE() will perform an additional if ( denominator <> 0 ) check.

### Avoid IFERROR () and ISERROR ()

IFERROR () and ISERROR () are sometime used in measure. These function forces engine to perform step by step execution of the row to check for errors. So, wherever possible replace with in-built function for error checking.

Example: DIVIDE () and SELECTEDVALUE () will perform error check internally and return expected results.

### COUNTROWS () vs COUNT ()

When it's your intention to count table rows, we recommend that you always use the COUNTROWS function. It's more efficient and doesn't consider BLANKs contained in any column of the

### SUMMARIZE () Vs SUMMARIZECOLUMNS ()

Summarize is used to provide aggregated result by performing group by action on the columns. It is recommended to use SUMMARIZECOLUMNS () function is optimized version. SUMMAZIE should only be used to get just the grouped elements of a table without any measures/aggregations associated with it.

### FILETER (all (ColumnName)) vs FILTER (VhALUES ()) vs FILTER(Table)

Always better to apply filters at desired column than the whole table. Also, use ALL along with the FILTER function if there is no specific need to keep current context. To calculate measures ignoring all the filters applied on a column, use All (ColumnName) function along with FILTER instead of Table or VALUES ()

### Avoid AddColumns () in measure expression

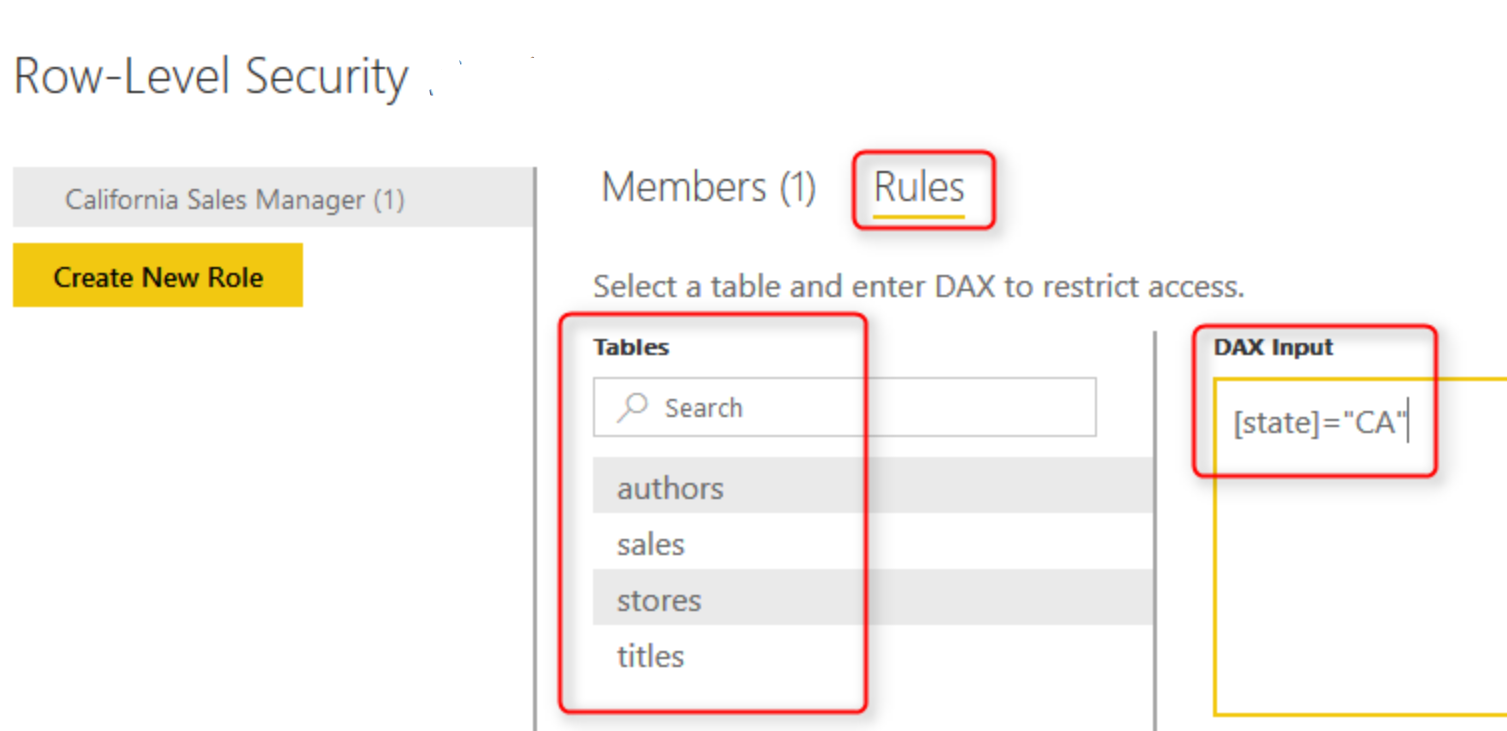
By default, measure is calculated in iterative manner. Adding Addcolumns will convert into nested loop and further slow down the performance.

## Complex Row Level Security

## Keep the security table as small as possible If the security granularity needs to be applied, consider multiple security tables. It is recommended to have relationships between security tables. These will avoid the additional LOOKUP calls for filter operation.

## Avoid Row level security filter directly on Fact tables.

* If you have multiple RLS filter applied on a single fact table, consider a mapping table and implement all the RLS fields on the single table
* Avoid Bi-directional filter. Instead of converting to a single direction in combination of FILTER DAX expression
* Keep filter function simple and avoid regular expression, string manipulation and complex logic.

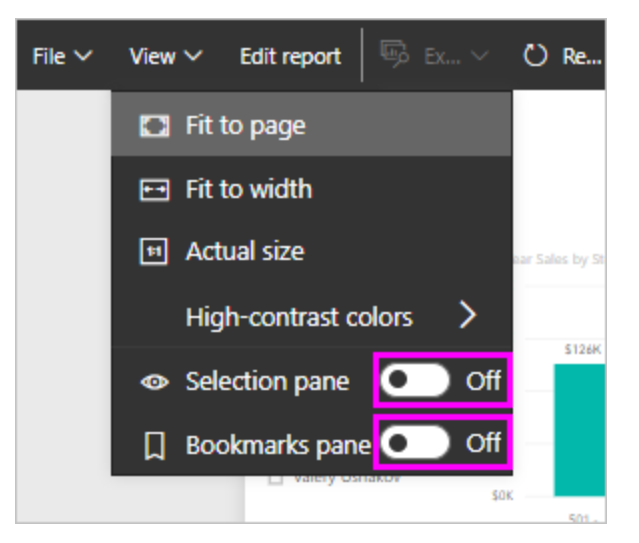


## Reports \ dashboards Design

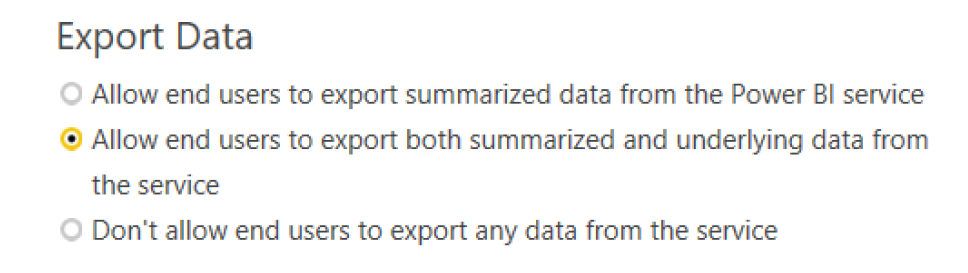
### Limit Number of Visuals

A good starting point is to build a report with few tiles with a maximum limit of 20 visuals. Every visual generates at least one query against its data source. So higher number of visuals can throttle CPU and network.

* The suitable approach is to segregate high level KPI data and granular line item report in separate reports.
* Drill Through to provide granular data
* Set default slicer and filter to limit dataset context
* Use bookmark capability to hide visuals and provide subset of data for analysis



* Test custom visuals performance in isolation. Replace custom visuals with built-in visuals wherever possible. However, if there is a functionality gap, it is recommended to use ones that is validated by Microsoft.
* Avoid export of reports with large granular data. Leaf level queries will consume memory and network bandwidth. Limit data export by selecting either export summarized data or none.



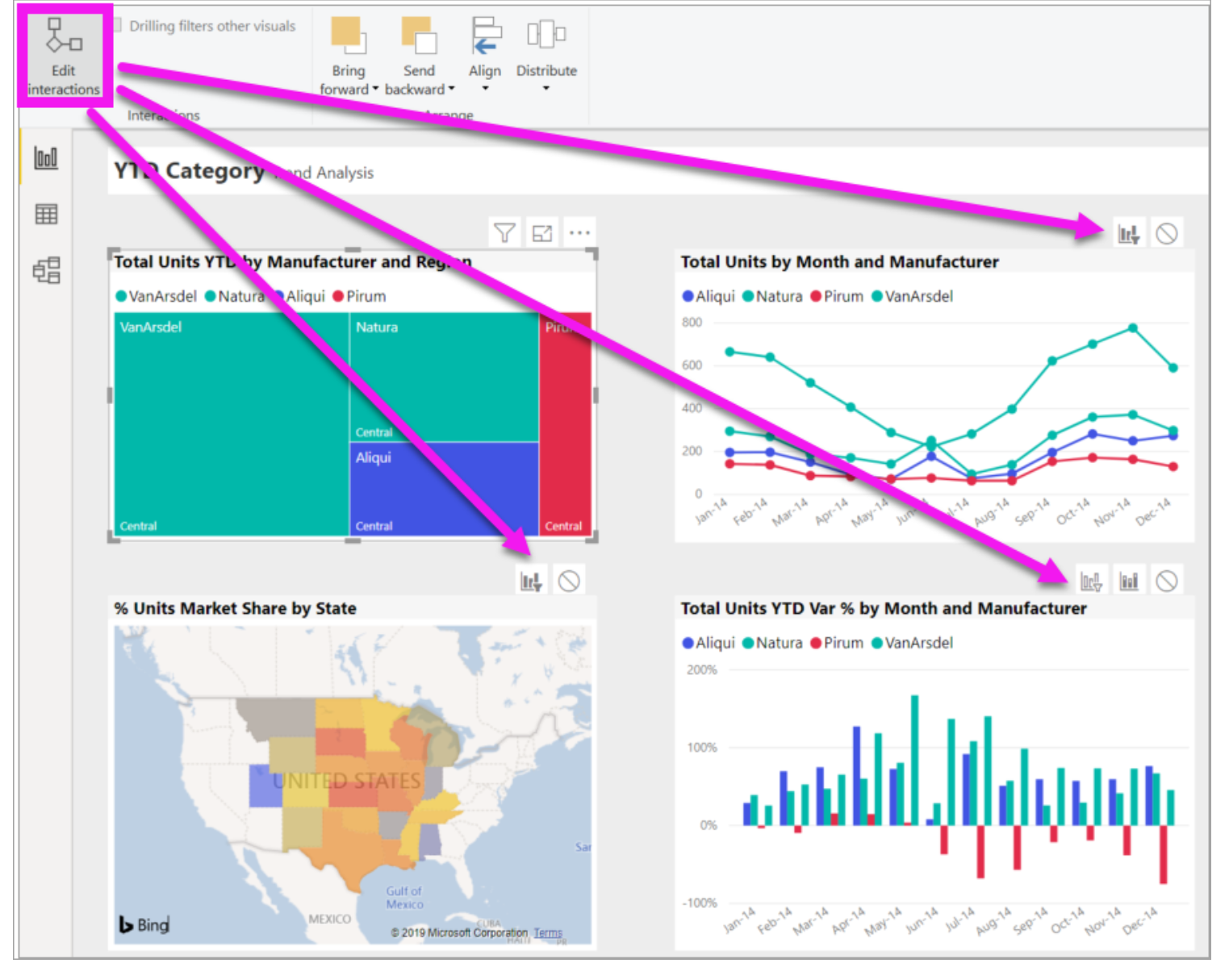
### Simplify Table or Matrix data

Tables\matrix in report with thousands of rows, many columns and measure can be complex and slow. Also, so many rows can be overwhelming for users to gain insight.

* Use TopN filter to limit initial view of table
* Move less critical measures to tooltips so these are only displayed on demand, per row

Slicers \ Filters

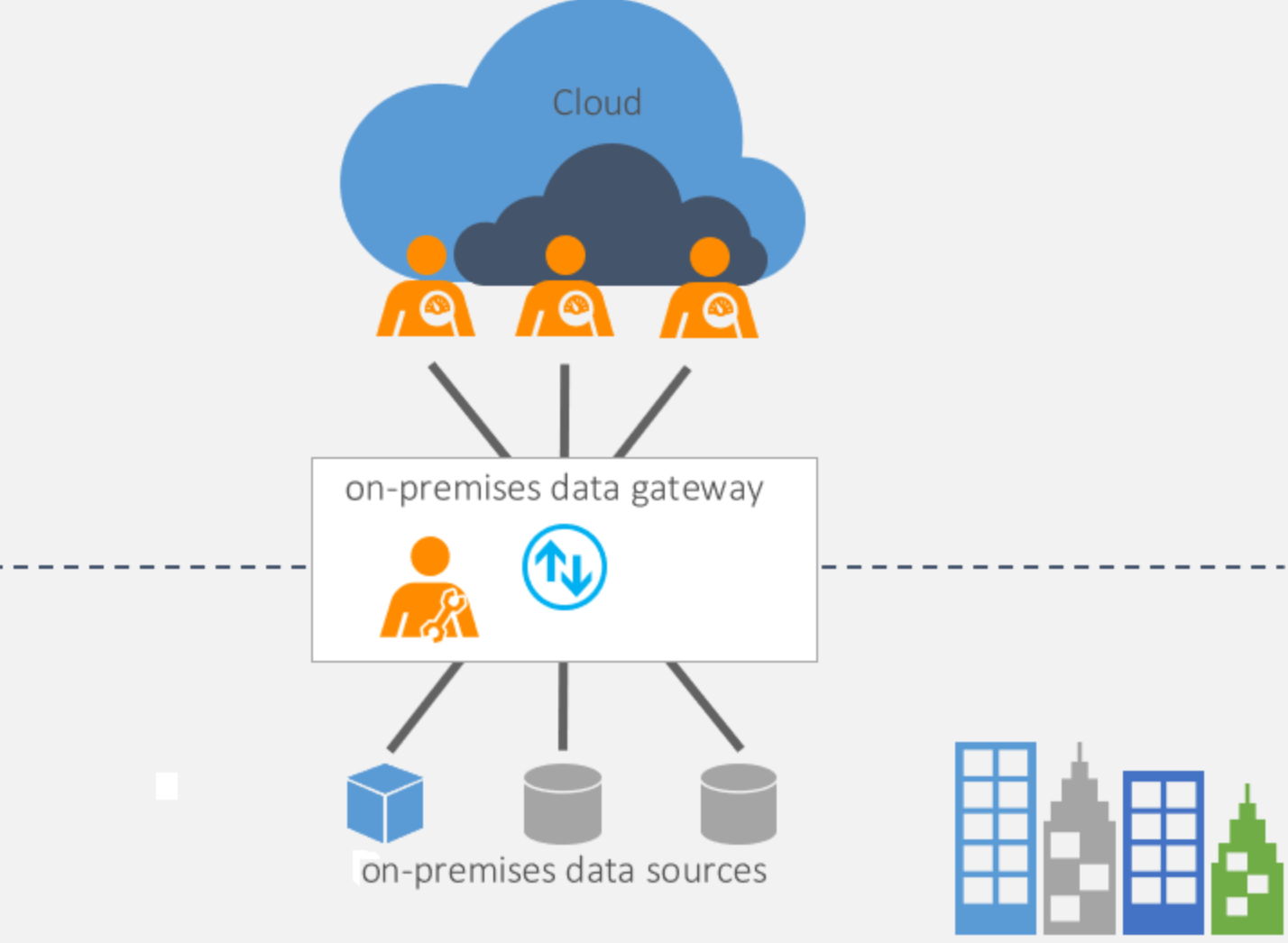
* For high number of visuals, using slicers or cross highlight filters every other visual by default and generated many queries. An option is to edit interactions and disable the meaningless cross filters.



* Avoid slicers with a very large number of values. Slicers has two queries. One is to populate and other is to fetch selection details. Instead of slicer use filter or force context and limit values.
* Also, set default value and set single\multi select property in slicers. It will reduce the context, reduce memory load and fetch less data and process
* Use sync slicers with care.

## Data Source/ Network Latency

The on-premises data gateway ( OPDG) acts as a bridge to provide quick and secure data transfer between on-premises data (data that isn't in the cloud) and several Microsoft cloud services. One of the cloud services is Power BI that uses OPDG to connect to the data sources that is connected to on-premises or sources that is within a private vnet.



### Data Gateway Configuration

* Keep data gateway as close to the data source. The data traffic between gateway and the azure is compressed. This will minimize the raw data movement for shorter distance between data source and gateway and leverage the benefit of the compression in the downstream traffic.
* Use enterprise gateway instead of personal gateway because enterprise gateway enables centralized gateway and data source management and supports various storage model
* Gateway Parallelism

The on-premises data gateway has settings that control resource usage on the machine where the gateway is installed. By default, gateway automatically scale these settings, using more or less resources depending on CPU capacity. In scenarios related to poor refresh, there are few settings that can be considered

MashupDefaultPoolContainerMaxCount: Maximum container count for Power BI refresh, Azure Analysis Services, and others.

MashupDQPoolContainerMaxCount: Maximum container count for Power BI Direct Query. You can set to twice of number of cores in gateway or auto tuning

MashupDQPoolContainerMaxWorkingSetInMB

<https://docs.microsoft.com/en-us/data-integration/gateway/service-gateway-performance-cpu>

### AutoScaling Gateway

Currently the gateway application utilizes resources on your gateway machine based on preset configurations. Using this feature, the application can now scale to use more resources or less depending on the system CPU.

To allow the gateway to scale based on CPU, this configuration “MashupDisableContainerAutoConfig” would need to be “False”. When this is done, the following configurations are adjusted based on the gateway CPU.

<https://powerbi.microsoft.com/en-us/blog/on-premises-data-gateway-june-2019-update-is-now-available/>

### Cluster Load Balancing

* + - Gateway admins can, now, throttle resources of each gateway member to make sure either a gateway member or the entire gateway cluster isn’t overloaded causing system failures.
    - If a gateway cluster with load balancing enabled receives a request from one of the cloud services (like Power BI), it randomly selects a gateway member. If this member is already at or over the throttling limit set for CPU or memory, another member within the cluster is selected. If all members within the cluster are in the same state, the request would fail.
    - CPUUtilizationPercentageThreshold – This configuration allows gateway admins to set a throttling limit for CPU. The permissible range for this configuration is between 0 to 100. A value of 0, which is the default, would indicate that this configuration is disabled.
    - MemoryUtilizationPercentageThreshold – This configuration allows gateway admins to set a throttling limit for memory. The permissible range for this configuration is between 0 to 100. A value of 0, which is the default, would indicate that this configuration is disabled.
    - ResourceUtilizationAggregateionPeriodInMinutes – This configuration is the time in minutes for which CPU and memory system counters of the gateway machine would be aggregated to be compared against the respective threshold limits set using configurations mentioned above. The default value for this configuration in 5.

### Use SSD \ Fast Storage

Power BI gateway refresh and data movement returns large dataset that is temporarily stored on the gateway machine. It is recommended to have SSD storage for spooling layer.