Power BI Best Practices

An initial compilation

ADOLFO J. SOCORRO, PH.D.
eSolutions, Inc.
ajs@eSolutionsPR.com
October 31, 2018

Introduction

This document presents a series of recommendations to assist in developing and organizing Power BI artifacts with the purpose of establishing standard practices that ensure uniformity and good performance. Note, however, that each development effort is different and may present challenges not addressed or resolved by these practices. In such cases, apply trusted principles such as separation of concerns and consistency in your solution. Also, the Power BI suite of analytics tools is always evolving, and you should adjust your practices accordingly and as necessary. Finally, many times we refer to SQL Server as a data source as that is our primary experience.

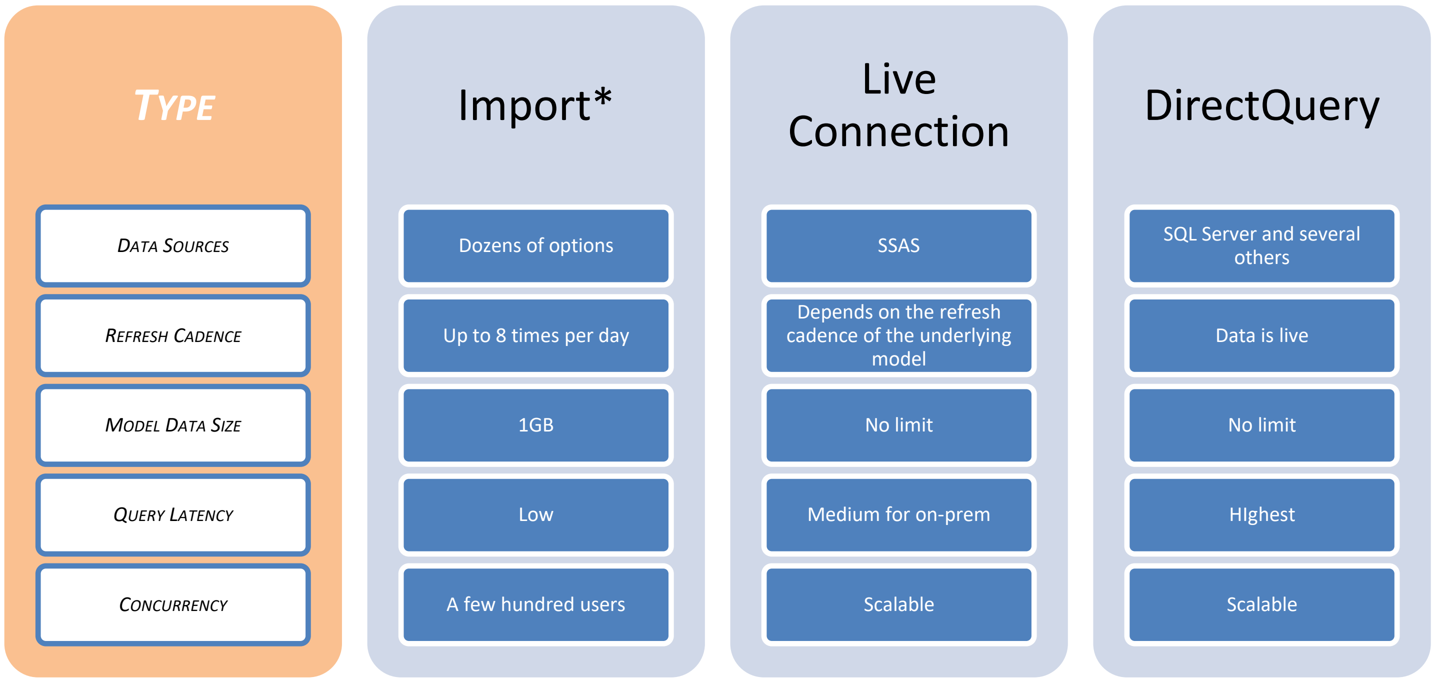
While some recommendations here are the result of my own development experience with Power BI, many (if not most) come from the writings of Power BI luminaries and from the product's documentation. I have cited the corresponding articles in the References section.

1 Data Sources

1.1 Carefully Choose the Connection Method

There are three ways to connect to data: Import, Live Connection, and DirectQuery (note: currently in preview are hybrid methods). The choice of method will depend on various considerations, such as the size of the data, the number of users, and the data refresh needs. Make sure an appropriate assessment is made before the report is built, as switching from one method to another is not always possible or straightforward.

The following chart summarizes the options on several dimensions.

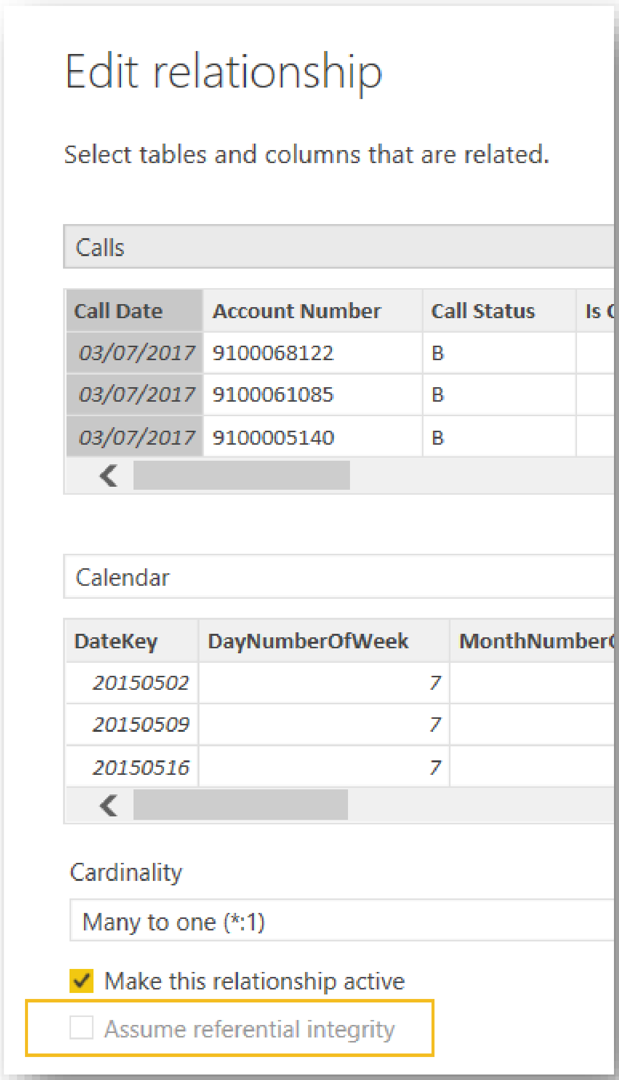


TYPE	Import*	Live Connection	DirectQuery
DATA SOURCES	Dozens of options	SSAS	SQL Server and several others
REFRESH CADENCE	Up to 8 times per day	Depends on the refresh cadence of the underlying model	Data is live
MODEL DATA SIZE	1GB	No limit	No limit
QUERY LATENCY	Low	Medium for on-prem	HIghest
CONCURRENCY	A few hundred users	Scalable	Scalable

**This is for Power BI Pro. Power BI Premium has more flexible refresh and size limits.*

1.2 For DirectQuery: Assume Referential Integrity

If applicable, try setting "Assume Referential Integrity" on relationships. In many cases, this can significantly improve query performance.



Edit relationship

Select tables and columns that are related.

Calls

Call Date	Account Number	Call Status	Is C
03/07/2017	9100068122	B	
03/07/2017	9100061085	B	
03/07/2017	9100005140	B	

Calendar

DateKey	DayNumberOfWeek	MonthNumberO
20150502	7	
20150509	7	
20150516	7	

Cardinality

Many to one (*:1)

☑ Make this relationship active

☐ Assume referential integrity

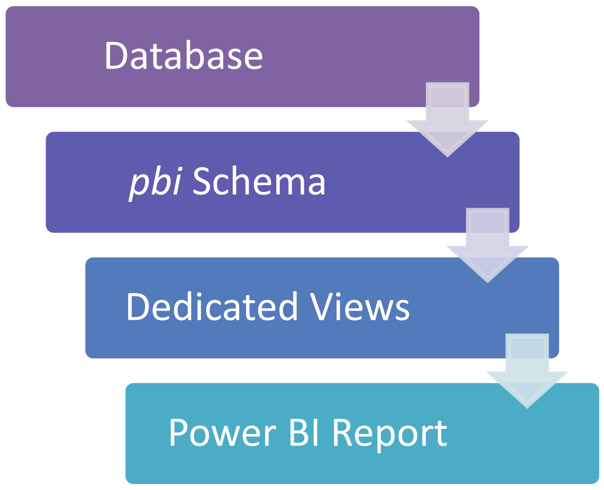
1.3 Import or Query only the Data Needed

Because Power BI's data is entirely loaded in RAM, avoid importing unnecessary columns. For the same reason but also to improve performance, filter the data as close to the source as possible.

As a corollary, never use `SELECT *` in queries.

1.4 Use Dedicated Views and Schema to Fetch Data

If getting data from a relational source such as SQL Server, use views to abstract the report from the physical layout of database tables and never write queries inside reports; from a maintenance and development standpoint, it is simpler and faster to update a view than it is to edit a query inside a report and then republish the report. Also, if available, create a separate schema for the views for grouping and security purposes. This is the recommended architecture:



1.5 Rename Columns in Views

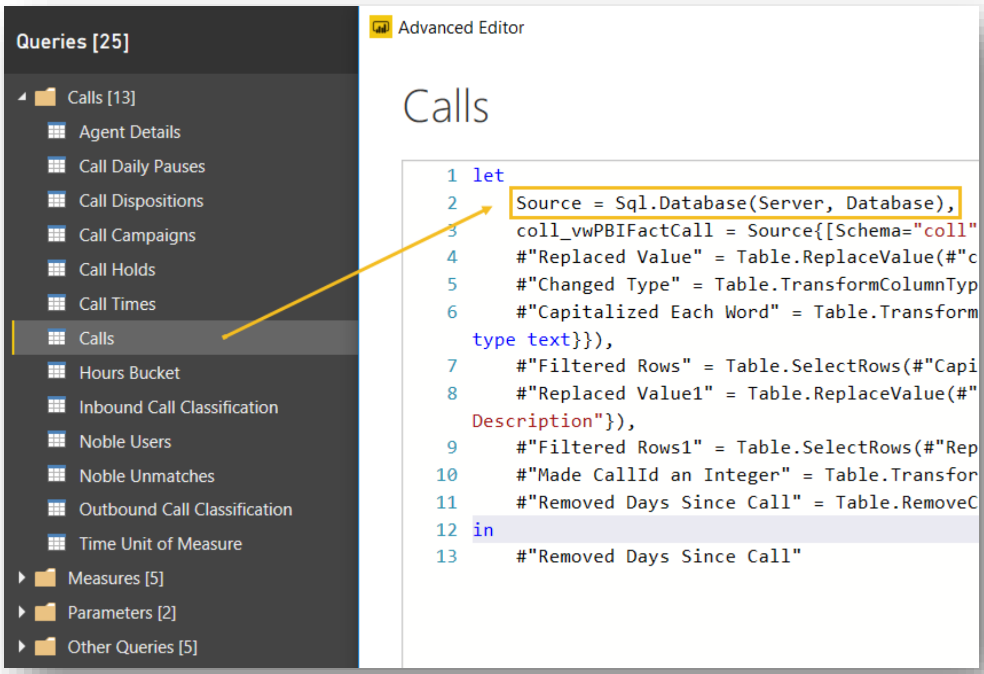
If getting data from a relational source, for example, perform all column renaming in views to simplify technical support, using exactly the names you will expose to users. Also, as it is best to expose calculations through measures rather than through default aggregations, rename all the columns involved so that their names may be reused in the report as the names of measures. For example, rename the column `PaymentAmount` in the view to `[_PaymentAmount]`, so that a measure called `[Payment Amount]` may be created in the report.

1.6 Setup a Corporate Folder for Files to Import

Establish a corporate location for files to import. This location may be organized with subfolders per workspace or report. Don't use personal folders.

1.7 Use Parameters for Connection Information

Don't hard-code data source connection information or file paths in M queries. Use parameters to refer to connection information.

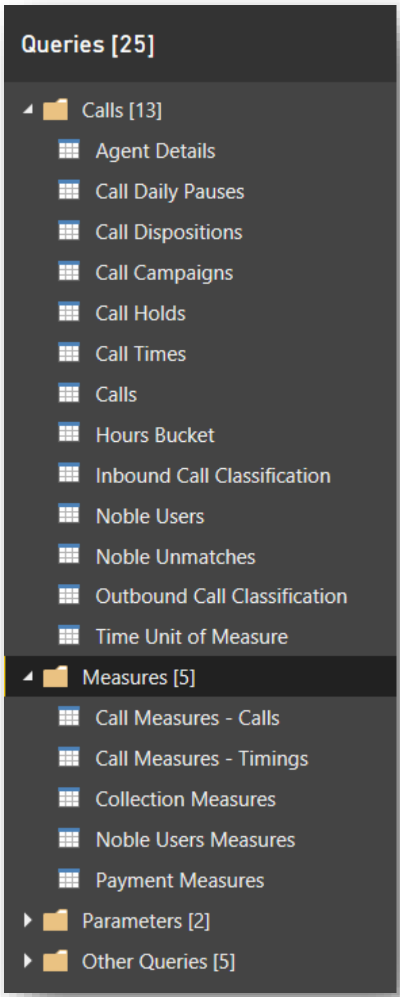


```
let
    Source = Sql.Database(Server, Database),
    coll_vwPBIFactCall = Source{[Schema="coll"
    #"Replaced Value" = Table.ReplaceValue(#"c
    #"Changed Type" = Table.TransformColumnTyp
    #"Capitalized Each Word" = Table.Transform
type text}}),
    #"Filtered Rows" = Table.SelectRows(#"Capi
    #"Replaced Value1" = Table.ReplaceValue(#"
Description"}),
    #"Filtered Rows1" = Table.SelectRows(#"Rep
    #"Made CallId an Integer" = Table.Transfor
    #"Removed Days Since Call" = Table.RemoveC
in
    #"Removed Days Since Call"
```

2 Query Editor and M

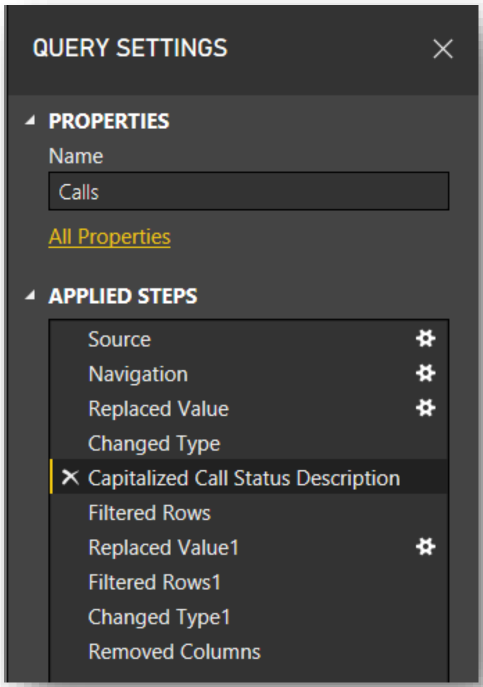
2.1 Use Query Groups

Use query groups to organize queries, and sort them inside each group (as of today, this can only be achieved manually by drag-and-drop).



2.2 Rename Query Steps

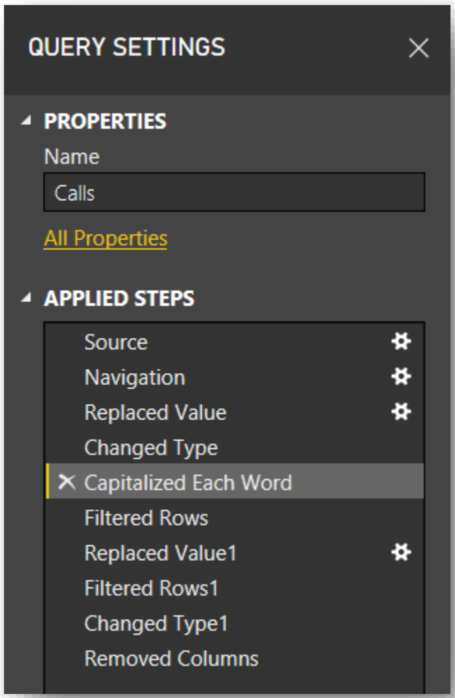
Power BI assigns default titles for steps, but it’s best to give them names related to their specific purpose. For example, “Capitalized Call Status Description”, as here:



Calls

Source
Navigation
Replaced Value
Changed Type
Capitalized Call Status Description
Filtered Rows
Replaced Value1
Filtered Rows1
Changed Type1
Removed Columns

This is self-documenting and better than the default “Capitalized Each Word”:



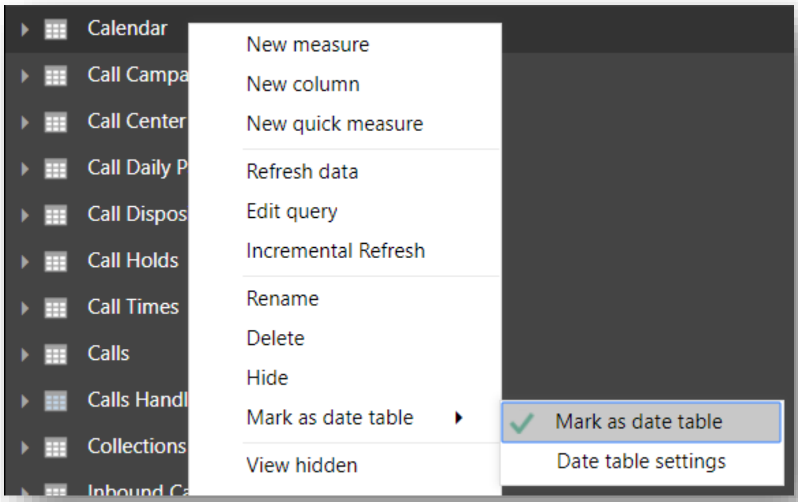
Calls

Source
Navigation
Replaced Value
Changed Type
Capitalized Each Word
Filtered Rows
Replaced Value1
Filtered Rows1
Changed Type1
Removed Columns

3 Data Modeling

3.1 Always use a Date Table

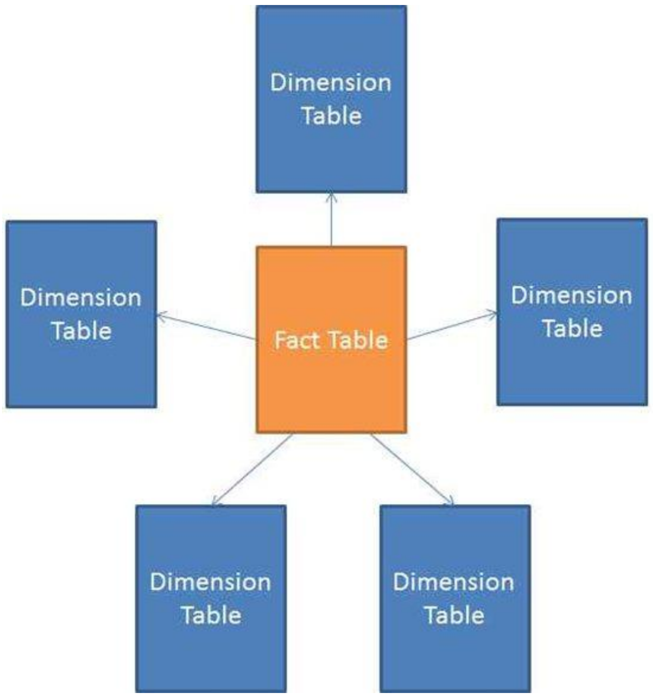
A date table is a major asset for almost any kind of reporting solution. Moreover, in Power BI time intelligence calculations are made simpler by the presence of a table marked as a Date Table.



Many people have published date tables or scripts to generate them. Here's one reference that uses DAX: https://www.sqlbi.com/tools/dax-date-template/.

3.2 Aim for Star Schemas

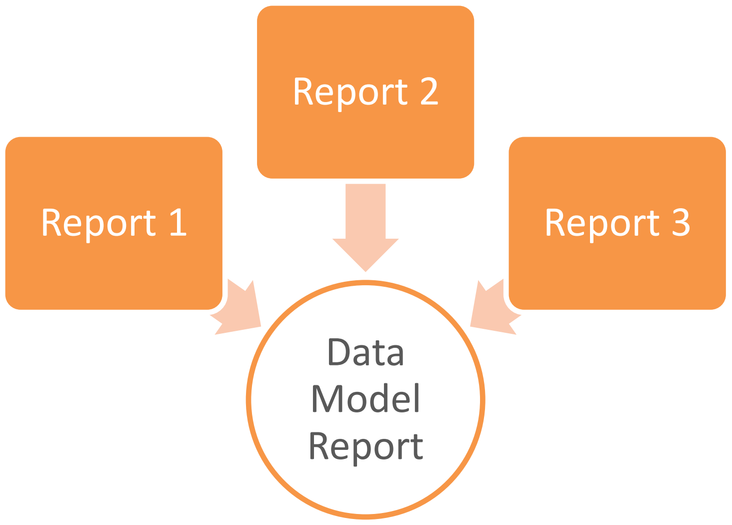
A star schema is a series of table relationships that when arranged in a diagram appear to form a star shape, with one fact table in the middle and dimensions around it. This concept originated as a data warehousing best practice for simplicity and easy querying. For example:



For non-trivial projects, one can expect to have several star schemas.

3.3 Separate Model and Report

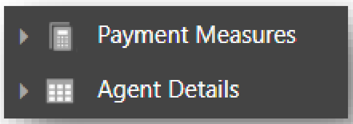
If your data model starts getting to a certain size, either in terms of the number of tables or in terms of the number of measures, it might be an indication that more than one report may be built on top of it. You may also know this beforehand. In such cases, consider creating a report without visuals, only the data model, and create other reports that connect to it via Live Connection. The resulting architecture may look like this:



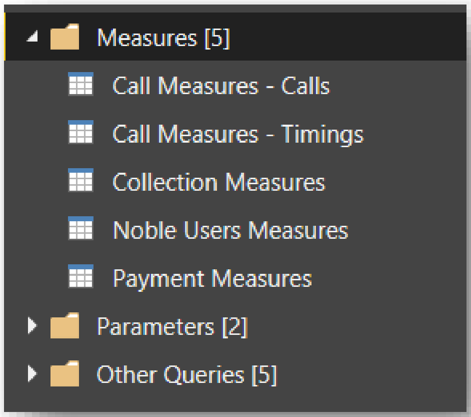
Large-scale projects may instead require an SSAS cube, for example, as the base model.

3.4 Group Measures in Separate Tables

For visual guidance and general organization, group measures into their own "tables." In the purest of models, all fact tables would be hidden, with their data being exposed exclusively through measures. Observe the calculator icon:

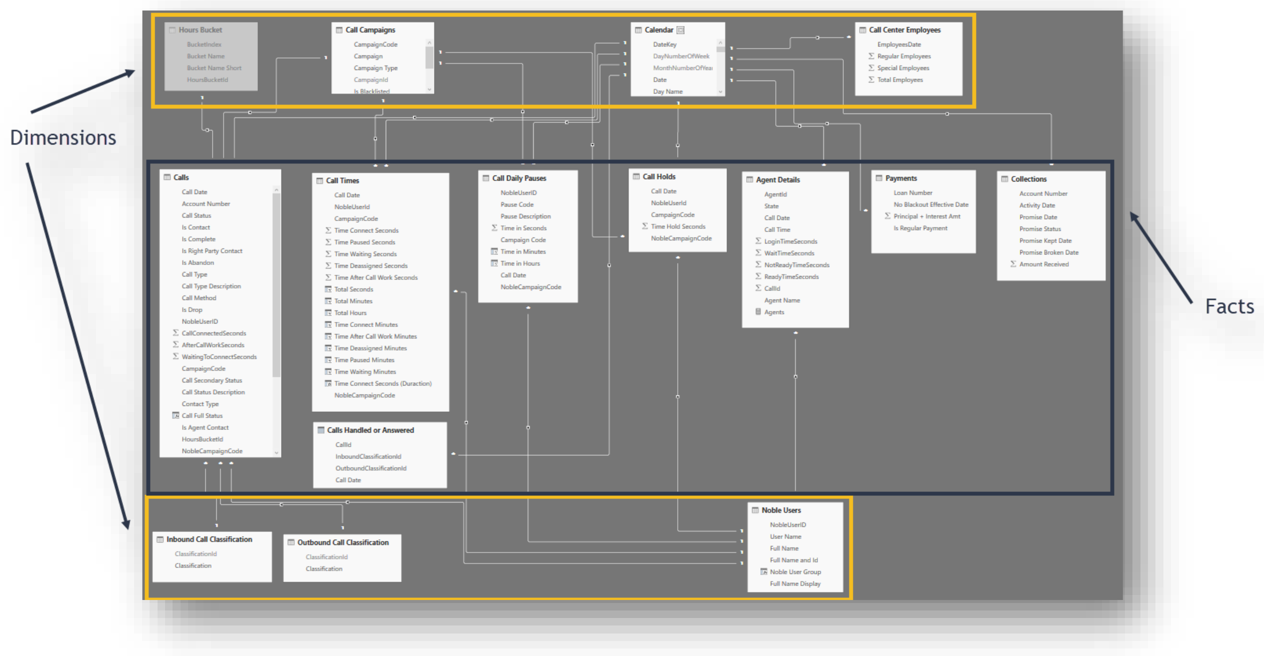


Also note that one fact table may give way to more than one measure table. For example, consider a report about a call center. The measures created for a fact table of calls may be organized in two measure tables: one for measures that count calls (e.g., number of inbound calls, number of outbound calls, number of overflow calls, etc.) and another one for measures that aggregate call timings (e.g., time waiting in queue, time on hold, talk time, etc.).



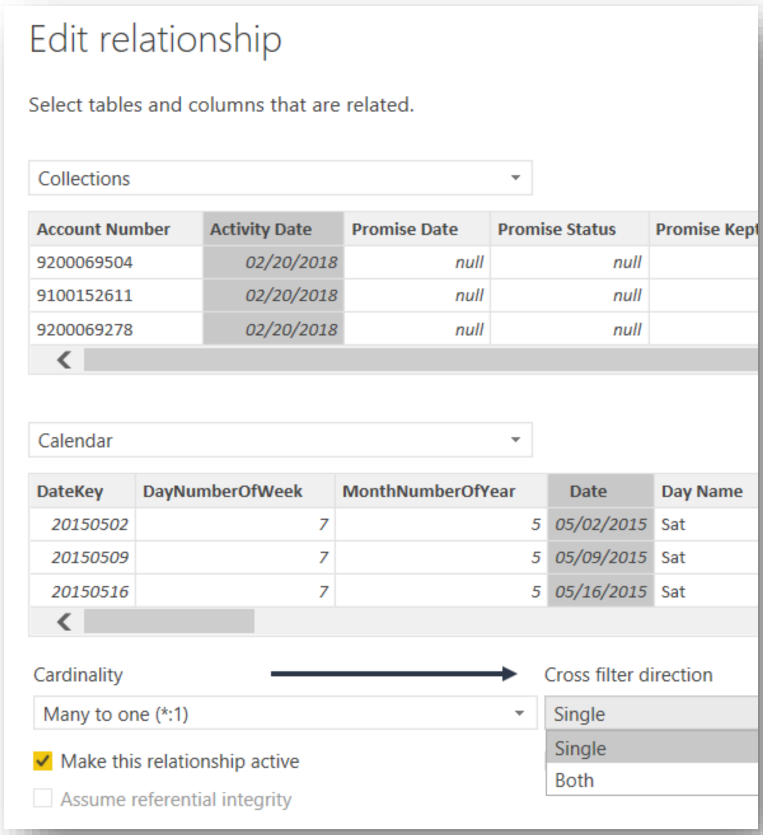
3.5 Use the Collie Method to Organize Diagrams

Author Rob Collie advocates a layout in which dimensions are placed on top and fact tables are placed below them. In the next example, we also lay out dimensions on the bottom as they don't all fit nicely on top.



3.6 Avoid Bi-Directional Filtering

Bi-directional relationships are a powerful modeling construct, but they can also have unexpected side effects. If unavoidable, treat them with care.



Edit relationship

Select tables and columns that are related.

Collections

Account Number	Activity Date	Promise Date	Promise Status	Promise Kept
9200069504	02/20/2018	null	null	
9100152611	02/20/2018	null	null	
9200069278	02/20/2018	null	null	

Calendar

DateKey	DayNumberOfWeek	MonthNumberOfYear	Date	Day Name
20150502	7	5	05/02/2015	Sat
20150509	7	5	05/09/2015	Sat
20150516	7	5	05/16/2015	Sat

Cardinality: Many to one (*:1)

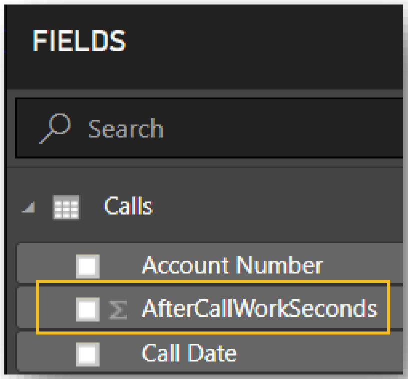
Cross filter direction: Single

✓ Make this relationship active

Assume referential integrity

3.7 Don't use Default Aggregations

Polished models expose functionality through measures. Also, when using *Export to Excel* there is no way to perform aggregations by hand in the resulting pivot table; calculations may only be included via measures.



3.8 Use a Consistent and Convenient Naming Convention

Choose a naming convention and follow it. This results in uniform-looking models that benefit both users and developers—and it makes your work look more professional. Also, try to select a naming convention that minimizes the need for customized titles for visuals and for Q&A synonyms.

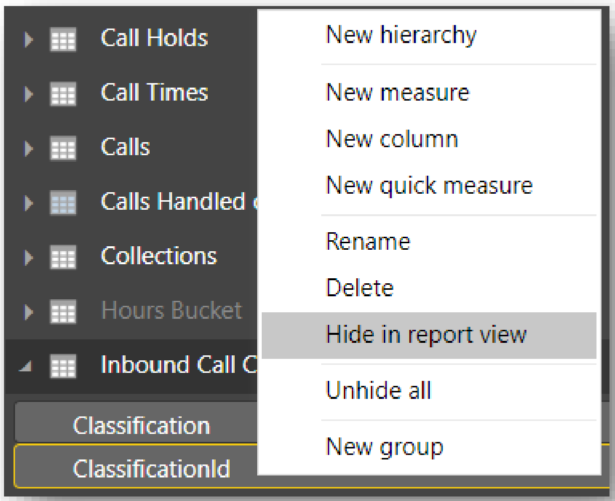
For example, if you have a field named *CalendarYear* and you create a visual that displays sales per calendar year, the default title will be "Sales per CalendarYear." If you had instead named the column *Calendar Year*, you would not need to edit the title to add a space between *Calendar* and *Year*.

3.9 Try not to Repeat Field Names across Tables

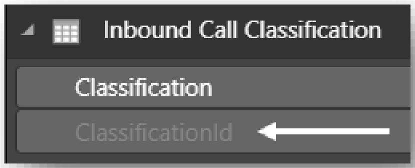
While measure names are enforced to be globally unique, field names are not. This might potentially cause ambiguity for users when examining visuals or using Q&A. For this reason, try to keep names unique across tables. For example, if both the Customer table and the Supplier table have a field called *Name*, rename it to *Customer Name* and *Supplier Name,* respectively.

3.10 Hide Auxiliary Fields

Fields such as keys used only to establish relationships between tables should be hidden if they are not meant for user consumption; this also removes clutter in the Report View.

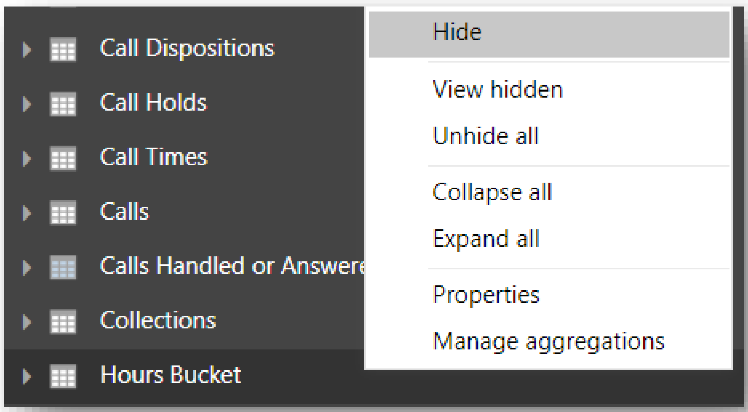


A hidden field looks like this to the report author in the Data View:

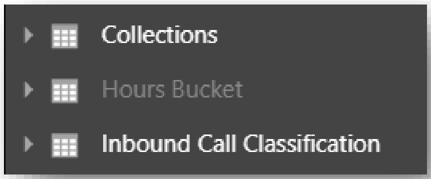


3.11 Hide Auxiliary Tables

Often auxiliary tables are brought in to aid in calculations but are not meant for user consumption. In such cases, it's best to hide them.



Hidden tables look like this in the Data View of a report:



3.12 Display the Date of the Data

Users should be informed of when the last refresh of the data occurred, or if the data is from a live source (i.e., DirectQuery). For the former, one way of obtaining this information is by creating a table in Power BI from a view with code such as this:

```
select getdate() as [Last Refresh Date];
```

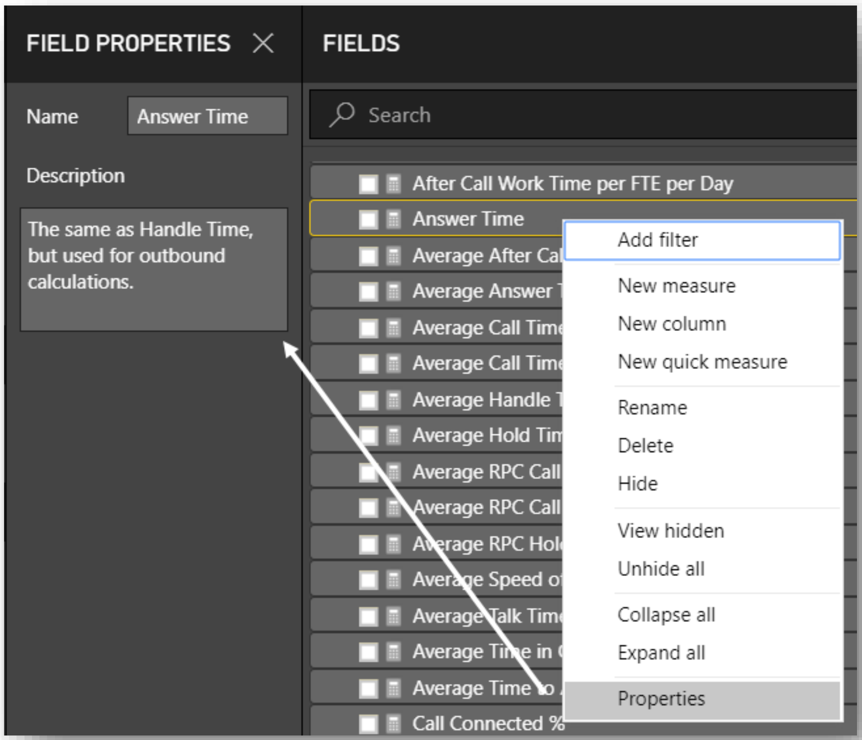
Every time the report is refreshed this table will have a row with the date of the refresh.

If the data comes from a source such as a data warehouse that is loaded at scheduled intervals and the loading date is available, then also show that date in the report.

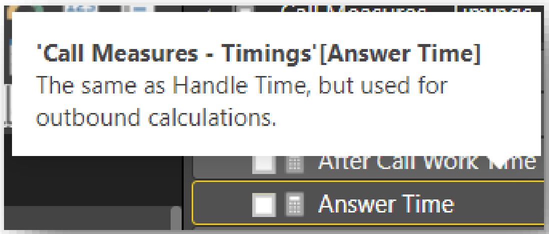
A third indicator of data freshness may come from the dataset itself. For instance, if there is a medical prescriptions table then show the most recent prescription date.

3.13 Write Field Properties

Fields (and measures too) should be documented via the Field Properties panel:



In the same report or in a client of the report which, for example, may be another Power BI report that connects to it via Live Connection, the description appears as a tooltip:


'Call Measures - Timings'[Answer Time]
The same as Handle Time, but used for outbound calculations.
After Call Work Time
Answer Time

4 Reports

4.1 Create Corporate Theme Files

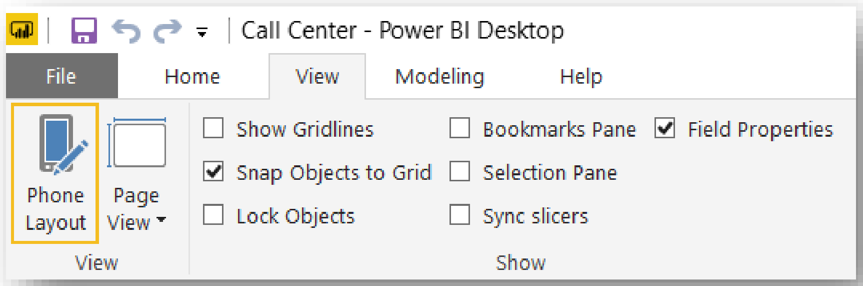
A theme file is a convenient way to ensure pages and visuals have consistent colors and styles; it also reduces the time spent selecting coordinated colors by hand or applying formats. Furthermore, theme files help in keeping a consistent look across reports that may be developed by different individuals.

4.2 Use Template Files

A Power BI template file is like a report file in that it may include pages, visuals, queries, data model artifacts (e.g., relationships and measures). However, it doesn't include any data. A template file may be prepared with some basic settings and components commonly used and distributed throughout an organization or even for personal reuse. For example, such a template file may include data source connection parameters, default pages (such as help pages), commonly-used icons and logos, and a date table, and may come with a corporate theme file already imported.

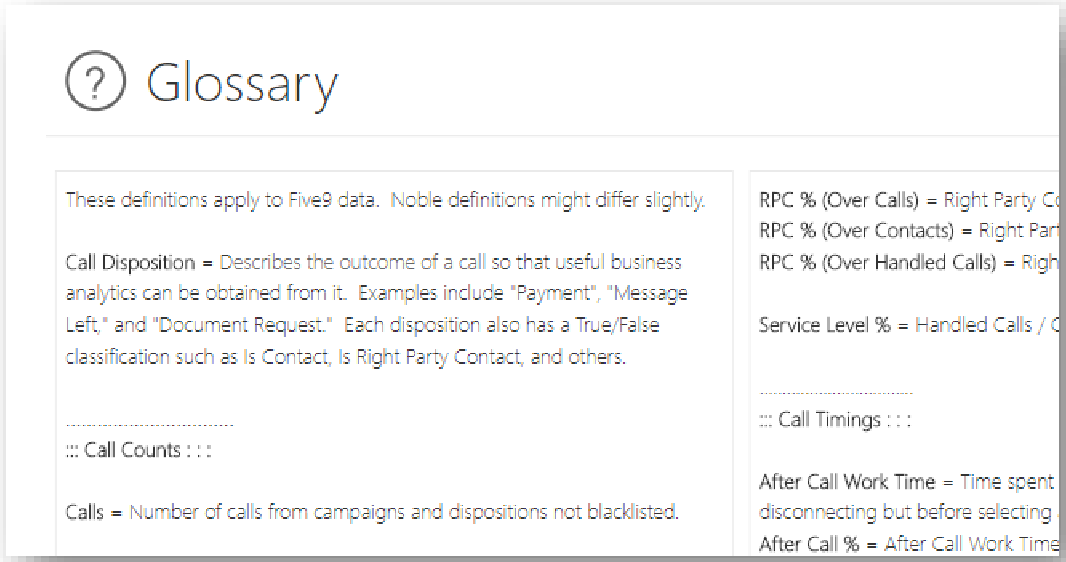
4.3 Configure Phone Layout

Even if you don't expect it to happen, chances are people will log in with their mobile devices. Take a few minutes to configure report pages for mobile screens:



4.4 Supply Help Pages

Consider using the first or the last page of a report as a place to include helpful tips, definitions, assumptions, or an introduction to the purpose of the report. It can also provide links to additional information, identify who to contact with questions, and who the owner of the report is. This is an excerpt of a glossary page in a report with many measures:



Glossary

These definitions apply to Five9 data. Noble definitions might differ slightly.

Call Disposition = Describes the outcome of a call so that useful business analytics can be obtained from it. Examples include "Payment", "Message Left," and "Document Request." Each disposition also has a True/False classification such as Is Contact, Is Right Party Contact, and others.

....
::: Call Counts : : :

Calls = Number of calls from campaigns and dispositions not blacklisted.

RPC % (Over Calls) = Right Party C
RPC % (Over Contacts) = Right Par
RPC % (Over Handled Calls) = Righ

Service Level % = Handled Calls / C

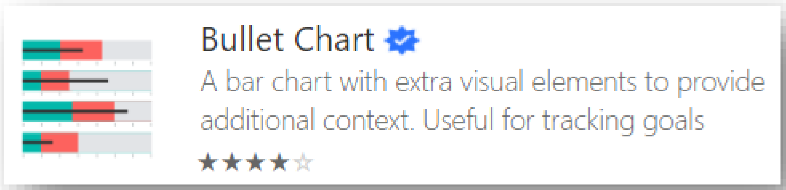
....
::: Call Timings : : :

After Call Work Time = Time spent
disconnecting but before selecting
After Call % = After Call Work Time

Moreover, a hidden page may be used for more technical notes intended for developers.

4.5 Exercise Care when using Custom Visuals

Custom visuals are software artifacts that may have integration, security, and performance issues and may affect your report's or dashboard's performance. Look out for the Microsoft *Certified* seal of approval when choosing custom visuals from the Marketplace.



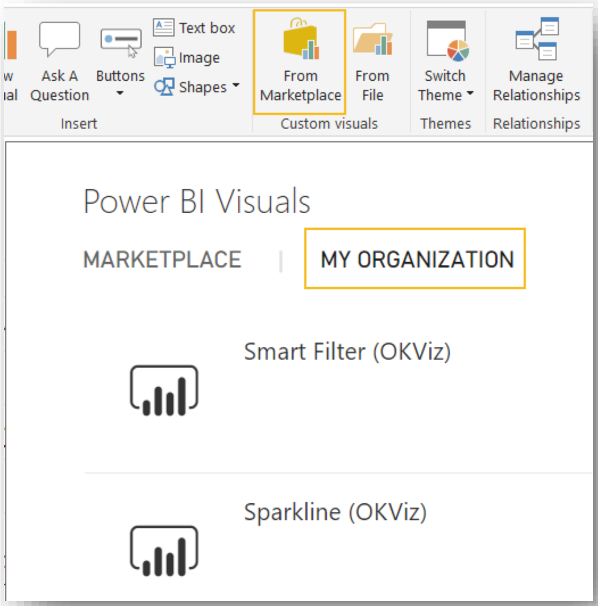
Bullet Chart

A bar chart with extra visual elements to provide additional context. Useful for tracking goals

★★★★☆

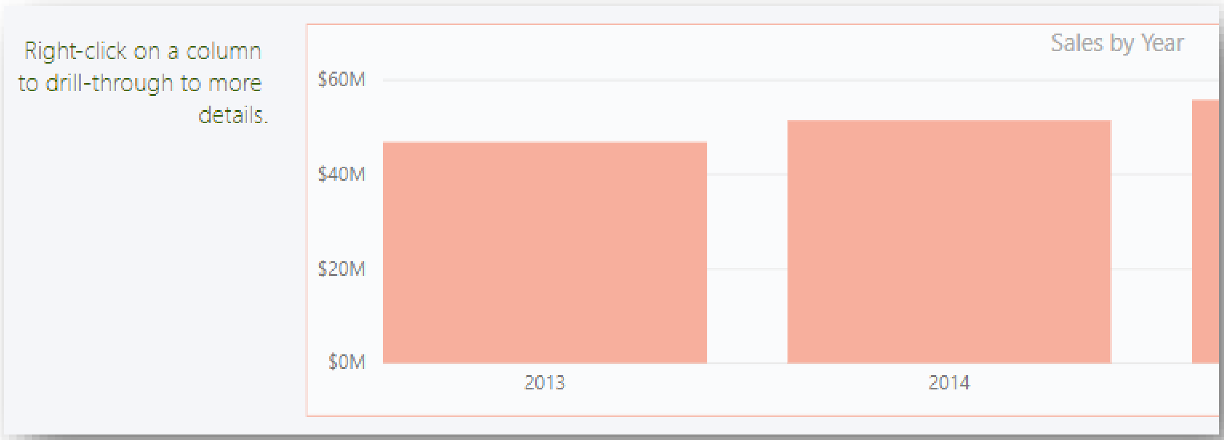
4.6 Make Common Custom Visuals be Organizational Visuals

A visual commonly used throughout the organization can be made to be an "organizational visual" in the Admin Portal. This way an administrator can load new versions of it that automatically update all reports that use it. Organizational visual are available here in reports:

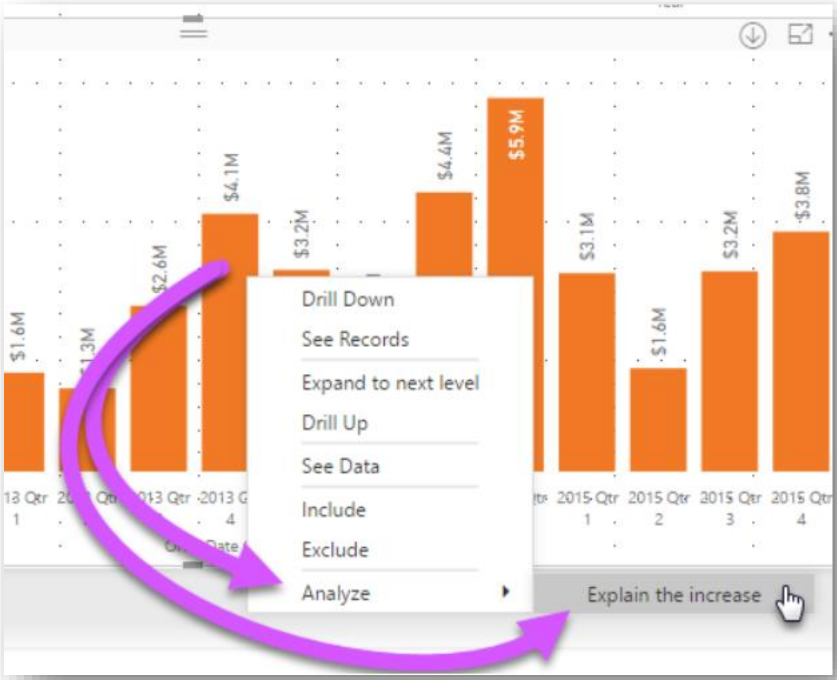


4.7 Guide the User

All the exploratory features of Power BI may not be known to novice users, so help them discover them in your report. For example, if you report has drill-throughs or fancy tooltips, a novice user might completely miss if you don't give them some clue as to their existence. A simple note may be enough guidance. For instance:



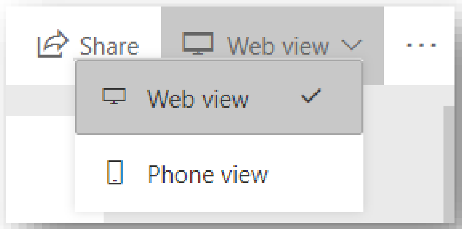
Another great exploratory feature is automatic insights, but its presence is not obvious. Help users find it. This is an image of how to access it, from the product's online documentation:



5 Dashboards

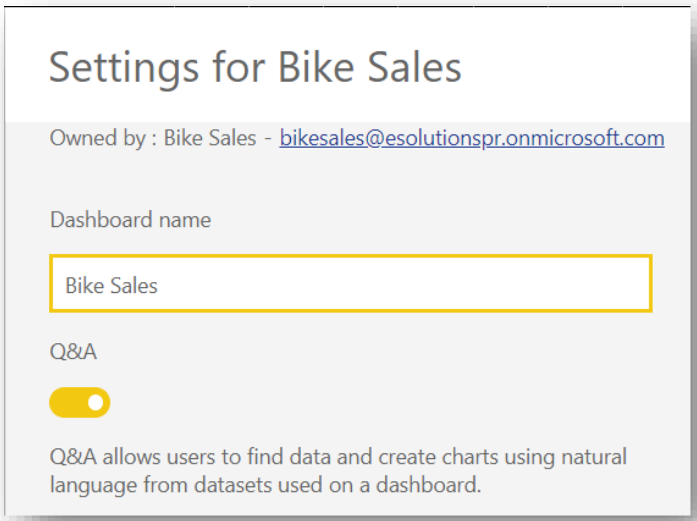
5.1 Configure Phone Layout

It's the same recommendation as for reports. In the case of dashboards, go here:



5.2 Disable Q&A if the Dataset has not been Groomed for it

Leaving it enabled might cause confusion to users when they find it hard to get results. Disable Q&A in dashboard settings:


Settings for Bike Sales

Owned by : Bike Sales - bikesales@esolutionspr.onmicrosoft.com

Dashboard name

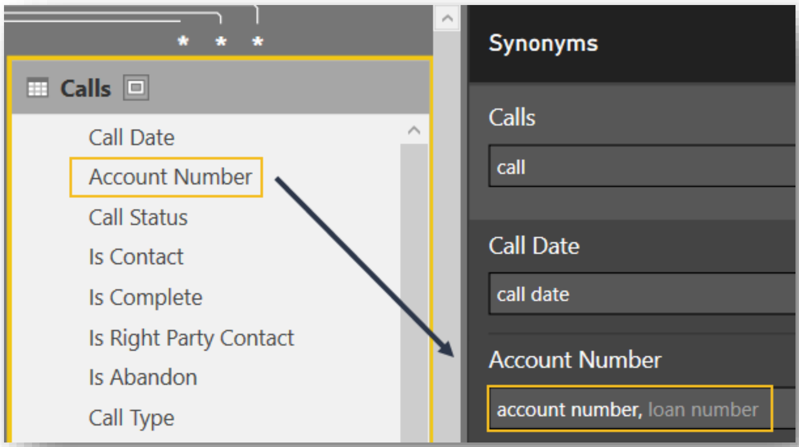
Bike Sales

Q&A

Q&A allows users to find data and create charts using natural language from datasets used on a dashboard.

Some of the best practices presented in this document, such as giving proper names to fields and hiding others, help towards providing users with a pleasant and effective experience with Q&A.

Other techniques are explicit, such as defining synonyms for fields. For example, if "account number" is also commonly referred to as "loan number," add the latter as a synonym in the Relationship View, like this:



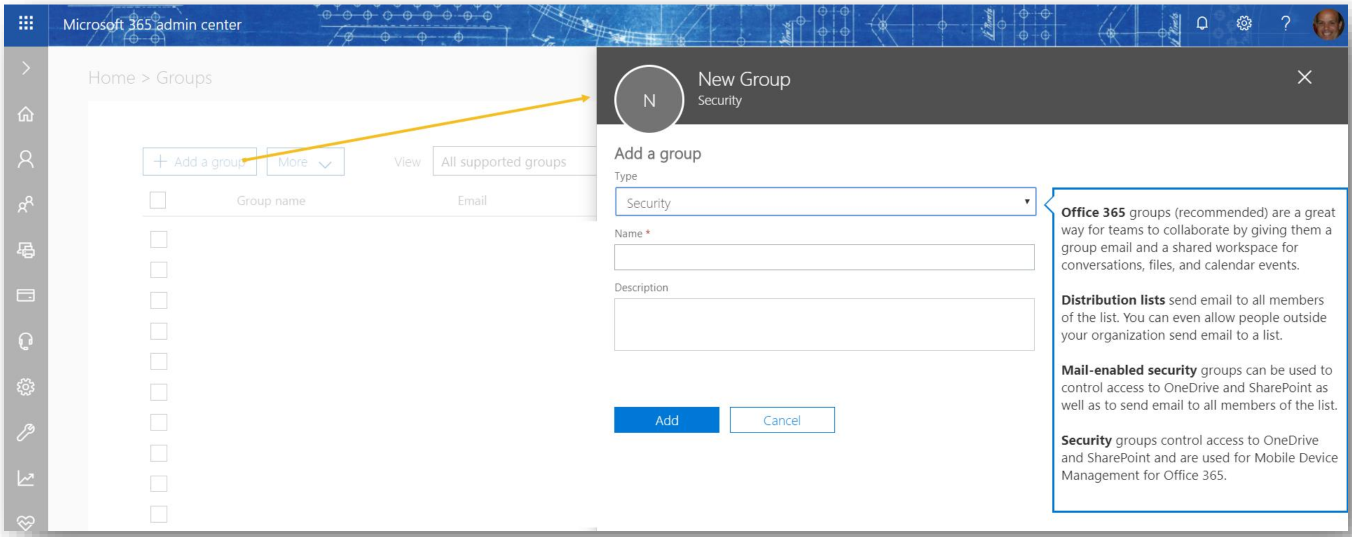
6 Sharing and Security

6.1 Share Reports and Dashboards using Apps

While Power BI allows reports and dashboards to be shared individually, apps allow packaging groups of reports and dashboards for controlled and more convenient distribution and administration.

6.2 Grant Access to Apps with Security Groups

Designate app users with O365 security groups rather than individually. This way, any tenant administrator can manage access and there is no need to republish the app to grant or remove access.

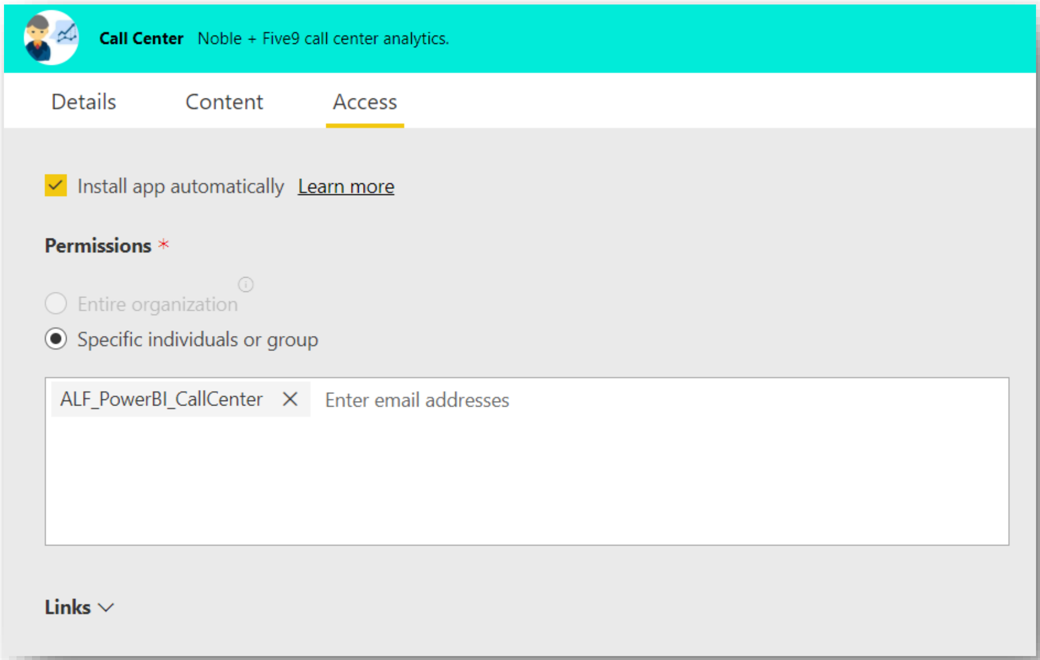


Office 365 groups (recommended) are a great way for teams to collaborate by giving them a group email and a shared workspace for conversations, files, and calendar events.

Distribution lists send email to all members of the list. You can even allow people outside your organization send email to a list.

Mail-enabled security groups can be used to control access to OneDrive and SharePoint as well as to send email to all members of the list.

Security groups control access to OneDrive and SharePoint and are used for Mobile Device Management for Office 365.



✓ Install app automatically Learn more

Permissions *

○ Entire organization
● Specific individuals or group

ALF_PowerBI_CallCenter ✕ Enter email addresses

7 Programming

7.1 Use Proper Code Indentation

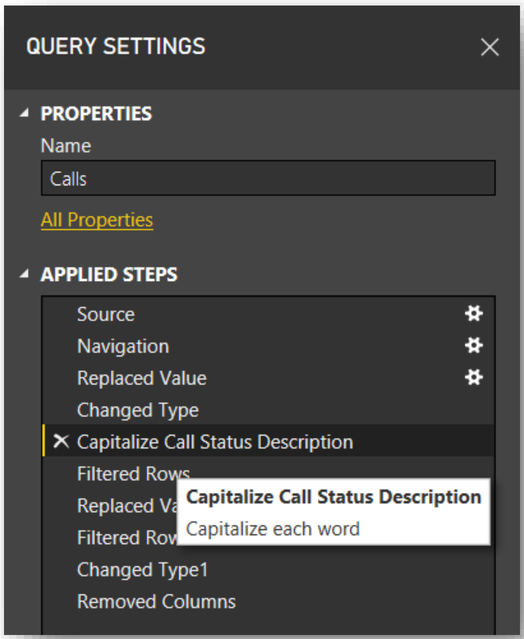
This applies to any kind of programming, of course. The first of the following two versions of the *Average Calls per Day* measure is much harder to read than the second one:

```
Average Calls per Day = CALCULATE(DIVIDE([Calls],COUNTROWS('Calendar')),KEEPFILTERS('Calendar'[Date] <= TODAY()))
```

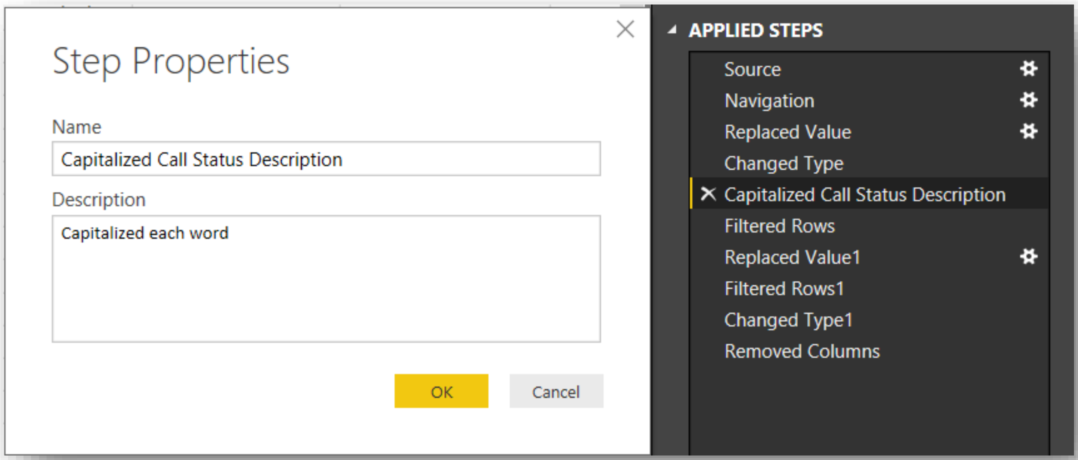
```
Average Calls per Day =
    CALCULATE (
        DIVIDE (
            [Calls],
            COUNTROWS ( 'Calendar' )
        ),
        KEEPFILTERS ( 'Calendar'[Date] <= TODAY() )
    )
```

7.2 Document Code

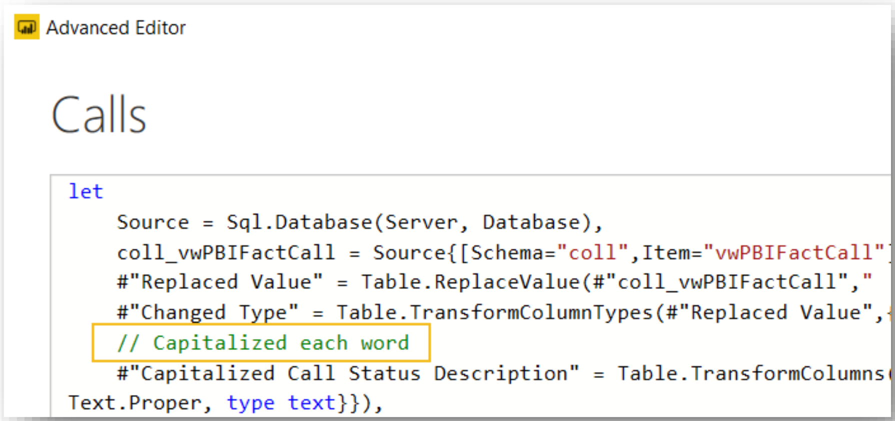
Add documentation to your code, not only for yourself but for your teammates. This too applies to any kind of programming. *Documentation will set you free*. In M, once added, tooltips will show the comment:



There are two ways to achieve this. The first is to place the comment in the properties window of the step:



The second is to place the comment directly in the code:



```
let
    Source = Sql.Database(Server, Database),
    coll_vwPBIFactCall = Source{[Schema="coll",Item="vwPBIFactCall"]
    #"Replaced Value" = Table.ReplaceValue(#"coll_vwPBIFactCall","
    #"Changed Type" = Table.TransformColumnTypes(#"Replaced Value",
    // Capitalized each word
    #"Capitalized Call Status Description" = Table.TransformColumns(
Text.Proper, type text}}),
```

In DAX, a double forward slash may also be used:

```
-- this is a comment
// this is also a comment
```

7.3 Use the DIVIDE function to Perform Arithmetic Division

Rather than having to continuously write code to defend against divisions by 0, let the DAX DIVIDE function take care of it. If it encounters a 0 in the divisor it will return BLANK as its result. For example:

```
Cost Ratio = DIVIDE ( [Total Cost], [Total Revenue] )
```

7.4 Break Measures into Interim Parts for Easier Debugging

Lacking a debugger, a way of testing complex DAX measures is to break them into simpler measures. For example, a measure like this:

```
My Great Formula =
        VAR Var1 = … some complex DAX …
        VAR Var2 = … some other complex DAX …
        RETURN
                DIVIDE ( Var1, Var 2 )
```

may be broken into two measures, one for Var1 and another for Var2, making it easy to display the three values in a table, for example. The resulting main measure would be:

```
My Great Formula =
        DIVIDE ( [Var1AsMeasure], [Var2AsMeasure] )
```

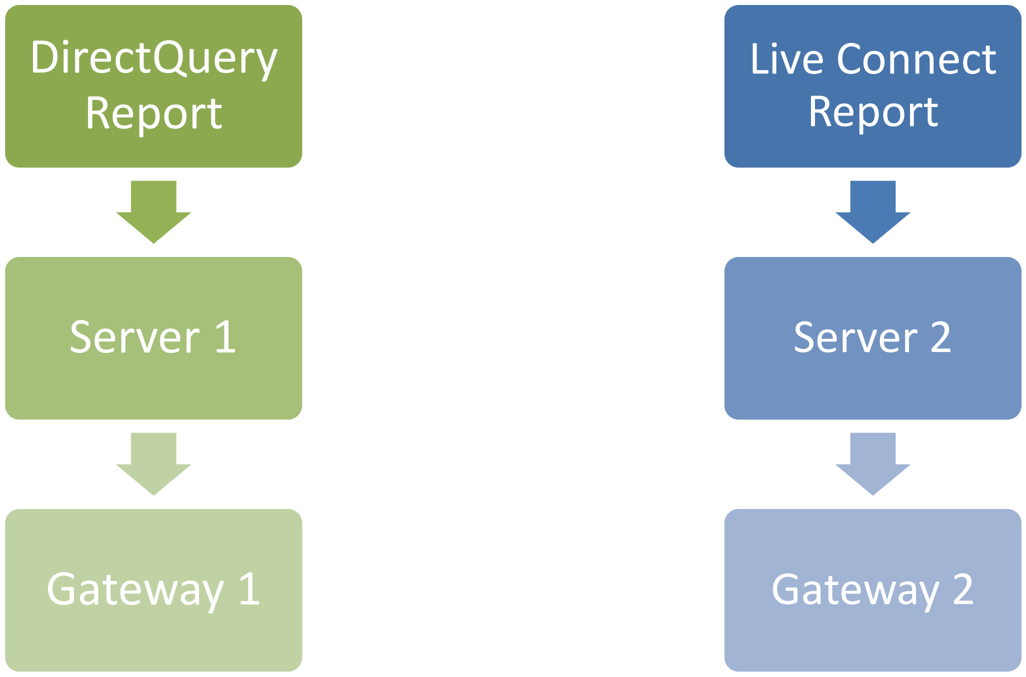
Otherwise, the use of variables to break up the code may be enough, if you modify the RETURN statement to alternatively display the value of each variable while debugging.

The use of variables is also good for development, documentation, and performance, especially in complex DAX code. It allows you to “divide and conquer”, and you may avoid the repeated evaluation of a common subexpression by abstracting it into a variable that is evaluated only once.

8 Performance

8.1 Use Different Power BI Gateways for “Direct Query” and “Scheduled Refresh”

This recommendation is to be tested in case performance degrades. It’s a way balancing load.



8.2 Limit Visuals on Report Pages

Every time a report’s page is rendered queries are run to fetch data and load each visual. If a page renders slowly, consider breaking it into two pages or employing drill-throughs, links, and tooltips.

8.3 Carefully Consider the Use of Report-page Embedding

Dashboard tiles with embedded report pages behave like regular report pages in that every time they cause the execution of queries for each visual. They are not cached in the same way regular dashboard tiles are.

8.4 Break Datetime Fields in Two

Consider breaking a datetime field into separate date and time fields. This will allow for better data compression and thus improved performance, as the number of distinct values will be less (remember, compression is done per column). For large data sets or data sets with many such fields, the benefit will be quite noticeable. Consider this:

Possible unique values for a Datetime field where the date is within one year	**31,536,000** 365 days x 24 hours x 60 minutes x 60 seconds

Possible unique values for a Date field where the date is within one year	365
Possible unique values for a Time field	86,400 24 hours x 60 minutes x 60 seconds
Total	**86,765** 365 + 86,400

As a side note, if your report does not use the time component, eliminate it in the source query or in M.

Alternatively, if you need to do analytics using time consider eliminating the time component and instead use "time buckets" to categorize time values, as in:

Bucket Id	**Time From**	**Time To**
1	6:00	6:59
2	7:00	7:59
…	…	…

9 Other

9.1 Give Report Files User-friendly Names

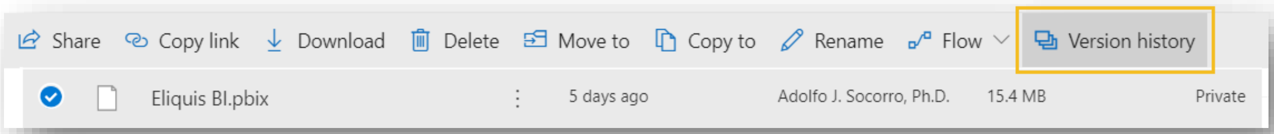
The names of report files are what Power BI uses as the name of reports in the Power BI service. Therefore, make sure they will make sense to end users.

9.2 Do not Edit Reports Online

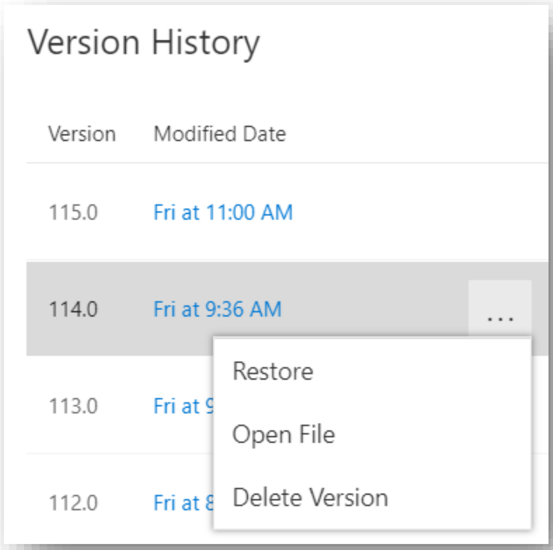
While an edited report may be downloaded to replace a local copy, this is still a feature in preview. Use it only in case of emergency, and make sure you go back to the original report file to reproduce the edits.

9.3 Use a Source Control System

Source code control is a professional safety and collaboration feature. Since a Power BI report is just a single file, an option for source control is OneDrive with its file-versioning feature.



Eliquis BI.pbix | 5 days ago | Adolfo J. Socorro, Ph.D. | 15.4 MB | Private



Version	Modified Date
115.0	Fri at 11:00 AM
114.0	Fri at 9:36 AM
113.0	Fri at 9
112.0	Fri at 8

If multiple artifacts are produced as part of the solution, such as template files, theme files, custom visuals, and design documents, then use a dedicated system such as Git.

10 References

Best design practices for reports and visuals
https://docs.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-best-practices

Power BI Performance Best Practices
https://docs.microsoft.com/en-us/power-bi/power-bi-reports-performance

Data Import Best Practices in Power BI
https://www.sqlbi.com/articles/data-import-best-practices-in-power-bi/

My Top 5 Power BI Visual Design Practices: Transforming Good to GREAT
https://powerpivotpro.com/2017/06/top-5-power-bi-visual-design-practices-transforming-good-great/

Power BI Visualization Best Practices
https://www.sqlbi.com/tv/power-bi-visualization-best-practices/

Best Practices for Power Pivot, Power Query and Power BI
https://exceleratorbi.com.au/best-practices-power-pivot-power-query-power-bi/

Power BI Performance Tips and Techniques
http://blog.pragmaticworks.com/power-bi-performance-tips-and-techniques

Checklist for Finalizing a Data Model in Power BI Desktop
https://www.sqlchick.com/entries/2017/12/23/checklist-for-finalizing-a-data-model-in-power-bi-desktop