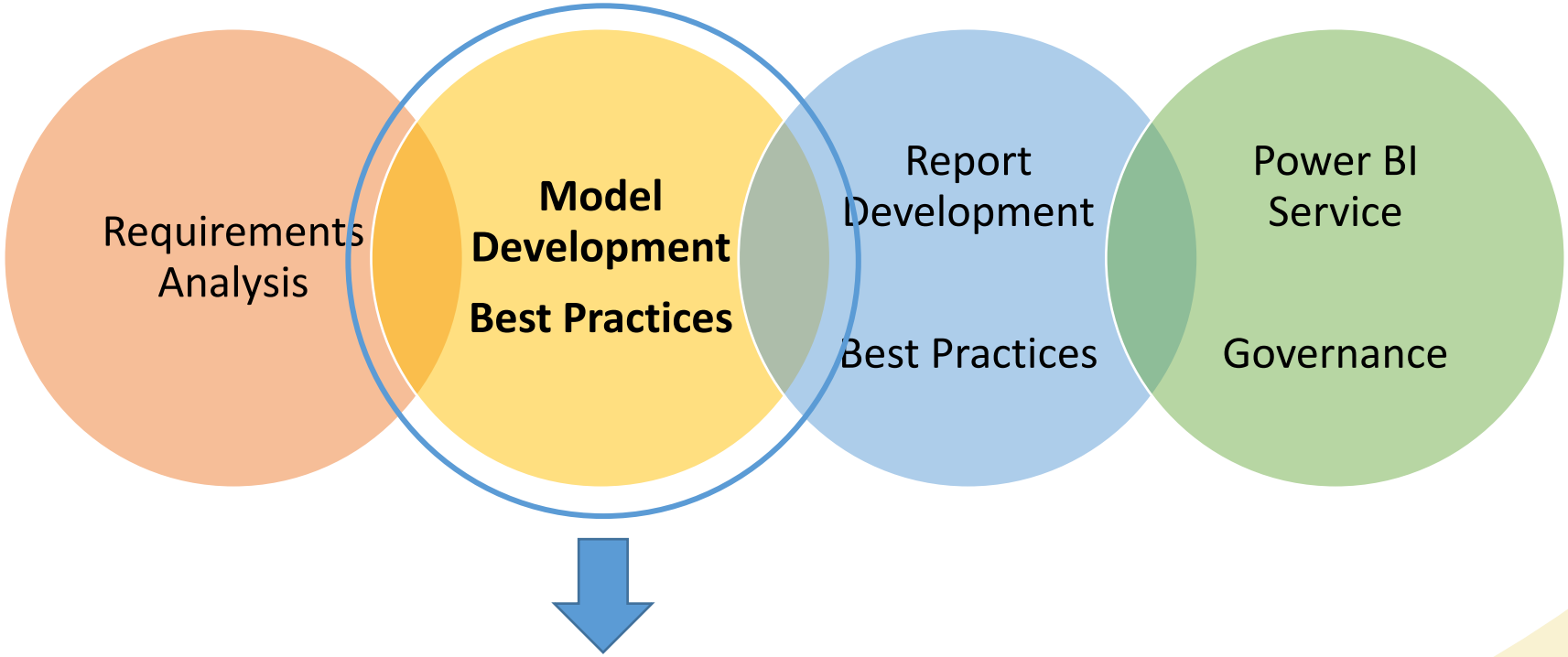
Power BI

Portugal

Developing a great model in Power BI

Introduction

Objective: To **gather transversal Best Practices & Tips** on how to develop a good model



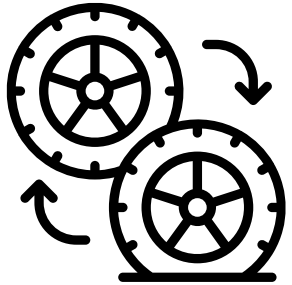
Dimensional modelling *plus* Model finetuning

Some notes

The BP described here **should be carefully thought before being applied to your data models. To some extent, there are no "easy recipes"**, at least on model preparation.

Yet, **using Dimensional modelling is an easy decision.** ☺

Disclaimer: I don't try to "reinvent the wheel, I want to show you to properly build a model". ☺
There are a lot of materials on this, and **I share several important links, including the ones from where I've taken images**.



Please consider reading them in depth.

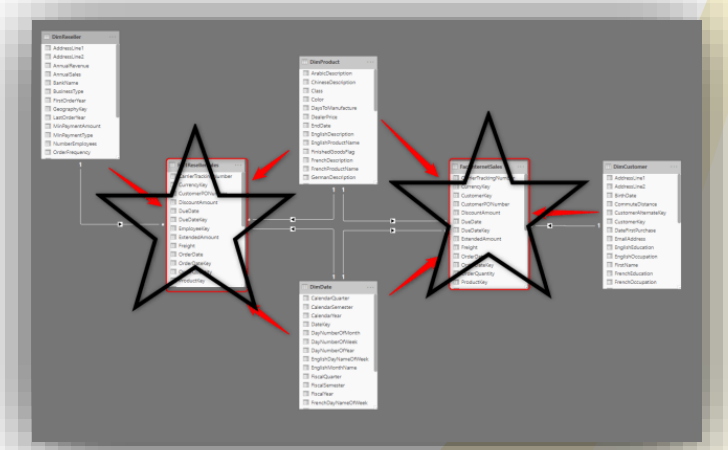
1 – Dimensional modelling

Kimball's Dimensional modelling appeared in 1996 and was promptly accepted in BI projects, note that the concepts of dimensions and facts are older.

Today, tools like **Power BI and Tableau are optimized for Star schemas**.

Don't use OLTP databases or "half baked" DataWarehouses.

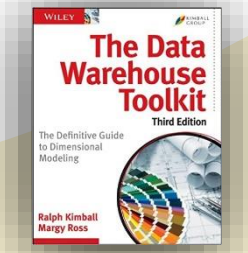
Models ideally should be Star schemas, or eventually Snow flakes, with conformed dimensions connecting them.



Why?

- **Better usability**: Deliver data that's understandable [and consistent] to the business users.

- **Deliver fast query performance**



https://www.sqlbi.com/blog/marco/2017/10/02/why-data-modeling-is-important-in-powerbi/
https://radacad.com/power-bi-basics-of-modeling-star-schema-and-how-to-build-it
https://sqlserverbi.blog/2020/12/25/doing-power-bi-the-right-way-6-data-modeling-essentials-best-practices-1-of-2/
https://www.amazon.com/gp/product/B00DRZX6XS/ref=dbs_a_def_rwt_bibl_vppi_i0

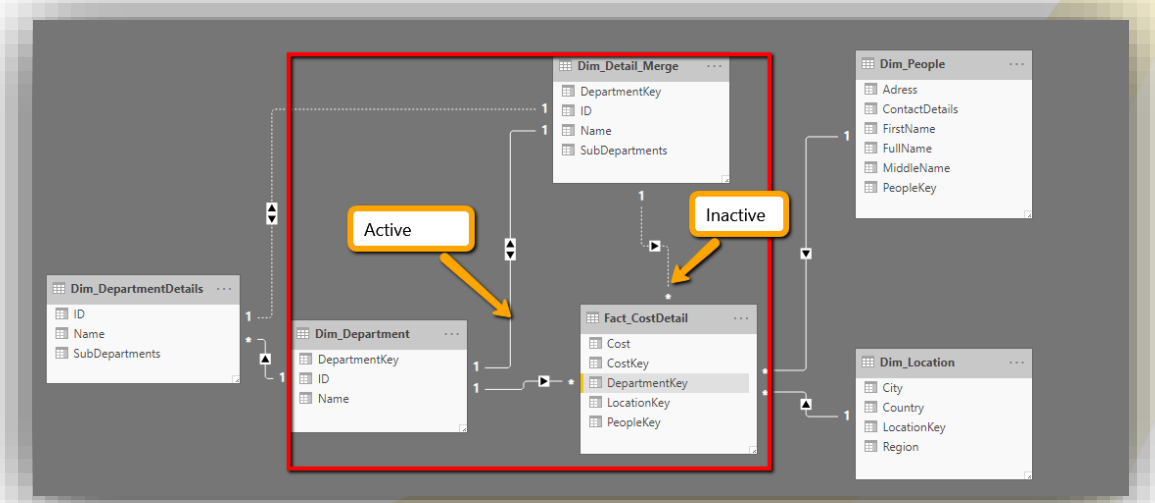
1.1 – Relationship types and Cardinality 1/2

In terms of Dimensional modelling, if you have tables that relate 1:1, must probably you can join them using views.

But **1:n relationships are the type of relationships that we should look for.**

Active vs. Inactive relationships:
Sometimes you just need to do some calculation using a relation between tables that are already connected. In that case create the 2nd relation (that will be inactive by default), and them use *USERELATIONSHIP* to activate it temporarily on your Calculation.

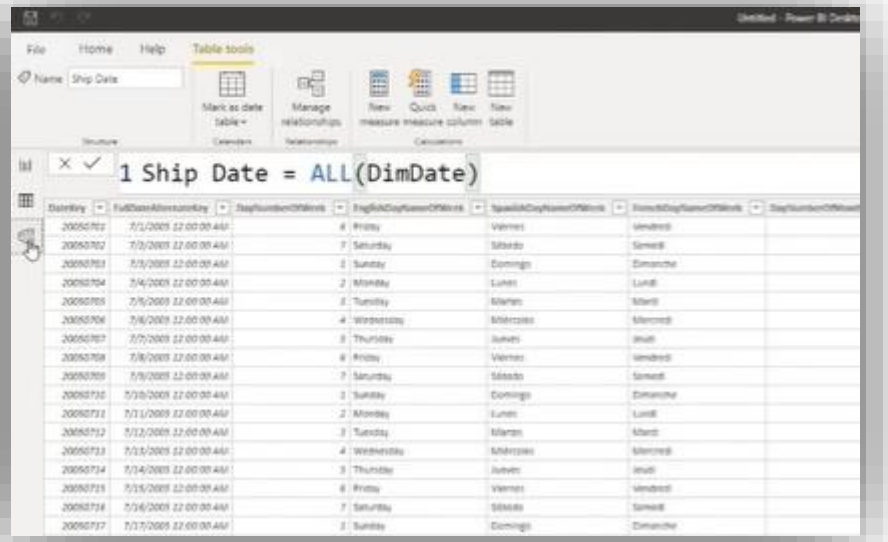
Notice the dash line:

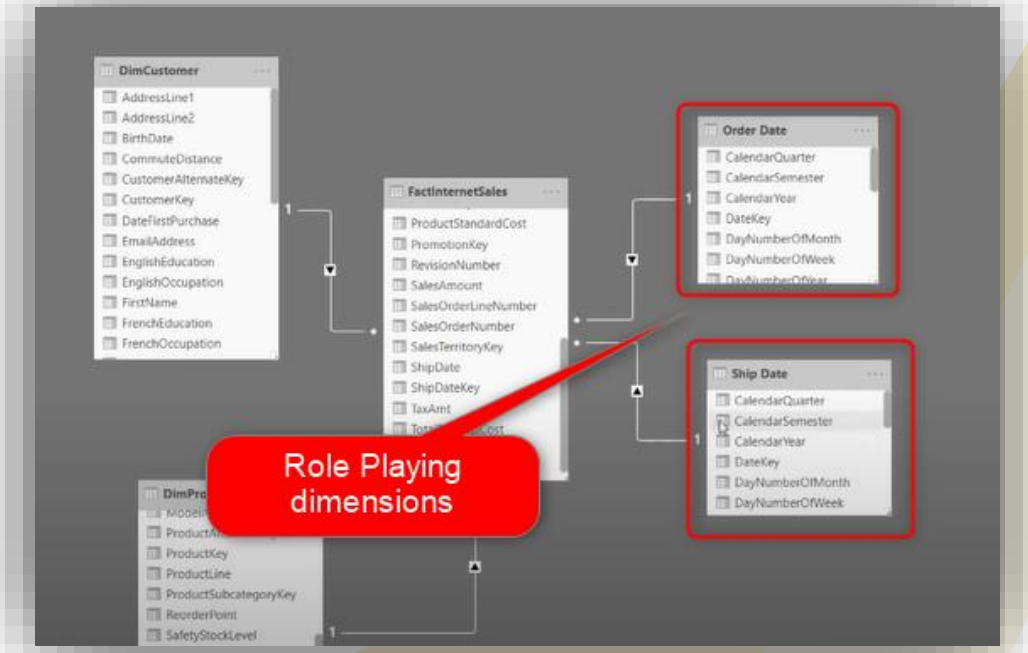


https://radacad.com/power-bi-basics-of-modeling-star-schema-and-how-to-build-it
https://www.youtube.com/watch?v=islhUWaCiJ0&t=10s&ab_channel=MicrosoftPowerBI

1.1 – Relationship types and Cardinality 2/2

In the case that you need to <u>filter</u> by more than one attribute in the same dimension (typical example are several dates), copy the dimension and create a 2nd relation to it (this is the concept of Role playing dimensions). Notice that space waste is minimal.



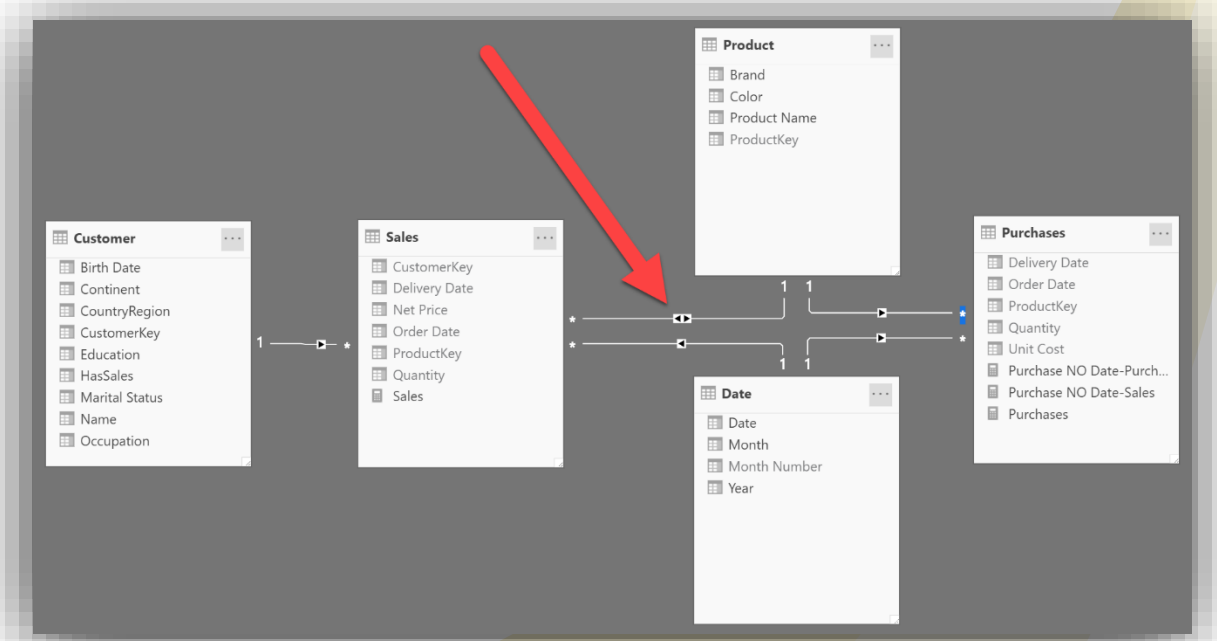


https://radacad.com/power-bi-basics-of-modeling-star-schema-and-how-to-build-it
https://www.youtube.com/watch?v=islhUWaCiJ0&t=10s&ab_channel=MicrosoftPowerBI

1.2 – Ambiguity on bidirectional relations 1/2

Activating bidirectional cross-filter in a Tabular data model might create ambiguous paths in the chain of relationships, resulting in very dangerous models as calculations become unpredictable.

By default, a relationship is created as unidirectional, flowing from the one-side (Dimension) to the many-side (Factual). **It is possible to set it as bidirectional but it should be avoided.** For example, in this model with 5 tables, relationship between Factual Sales and Dimension Product was set as bidirectional.



https://www.sqlbi.com/articles/bidirectional-relationships-and-ambiguity-in-dax/
https://www.youtube.com/watch?v=x3m7qzsVJqQ&ab_channel=SQLBI

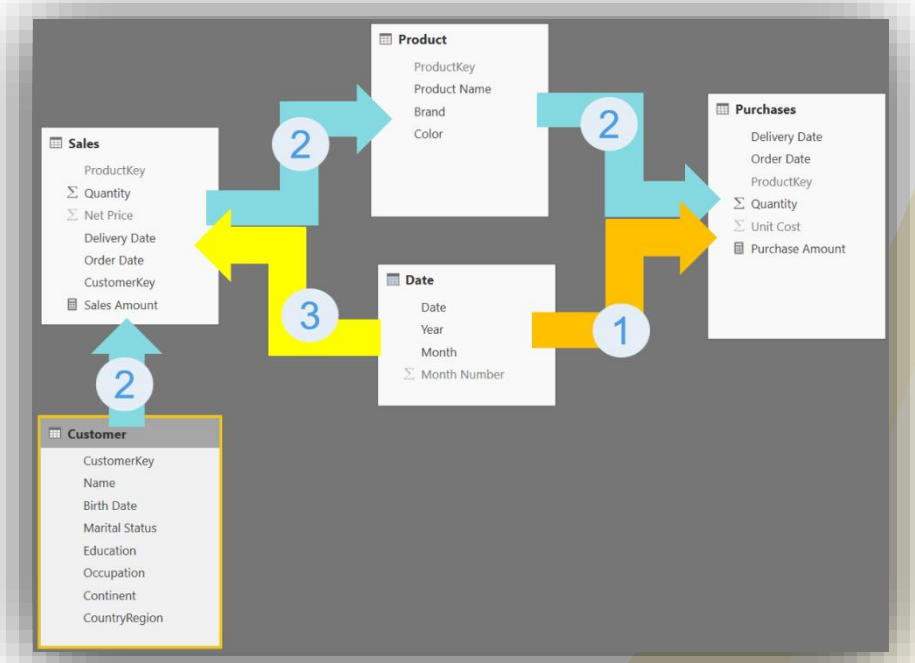
1.2 – Ambiguity on bidirectional relations 2/2

This change creates ambiguity in the model, and several questions are raised.
In several cases, **the engine analyzes multiple paths and can opt for the best path.**

But **If there is a filter on Date and a filter on Customer,** does Date filter Purchases indirectly through Sales (3 & 2), or directly (1)?

Answer is both. But **the conclusion is that the algoritm is too complex to understand and the more tables exist and filters are set, the more ambiguity becomes an issue.**

Long story short: bidirectionality should be used just in particular cases, use *CROSSFILTER* in your calculations to set it temporarily bidirectional.

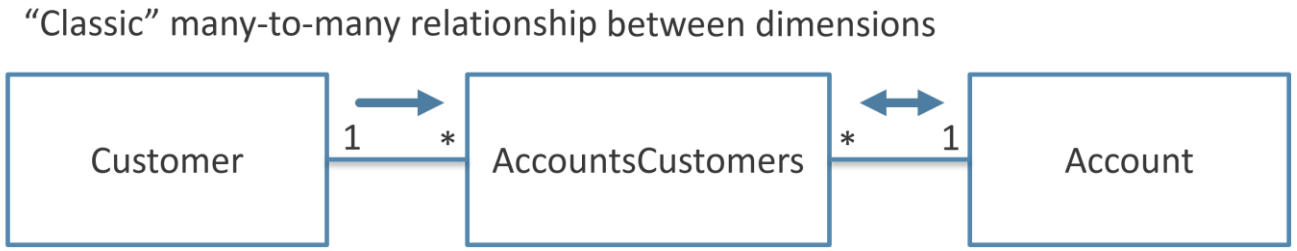


https://www.sqlbi.com/articles/bidirectional-relationships-and-ambiguity-in-dax/

1.3 – Other technical considerations 1/2

You can create **disconnected tables**, they are useful, for example, in What If scenarios.

You can also create **Many-to-many relationships between dimensions**, they require a bridge table containing data coming from the data source. The bridge table holds the information that defines the existing relationships between the entities. Typical example: A Customer has several Accounts, and an Account belongs to more that one Customer.



https://sqlserverbi.blog/2020/12/31/doing-power-bi-the-right-way-6-data-modeling-essentials-best-practices-2-of-2/
https://www.sqlbi.com/articles/relationships-in-power-bi-and-tabular-models/

1.3 – Other technical considerations 2/2

You can create **relationships at different granularities**: these relationships use columns that do not correspond to the identity of the table, so the cardinality is “many” on both ends of the virtual relationship.
There are 2 ways of doing it, in both cases a single direction should always be applied.

Relationship at different granularities

The filter direction of a weak relationship should be “single” (default is “both”)

https://sqlserverbi.blog/2020/12/31/doing-power-bi-the-right-way-6-data-modeling-essentials-best-practices-2-of-2/
https://www.sqlbi.com/articles/relationships-in-power-bi-and-tabular-models/

1.4 – Views, tables & Query folding

If you are getting data from a relational database, such as SQL Server or Oracle, **you should use Views to import the data into your data model**, creating in this way a **layer of abstraction.**

The **views created should include an explicit list of columns,** not using the * (all), and should **include transformation of data**.

The ORDER BY clause should not be required, unless you want to optimize the allocation of data in partitions larger than 8M rows. For smaller tables in SSAS, it doesn't matter.

Consider grouping them by schema in DB, and using clustered indexes.

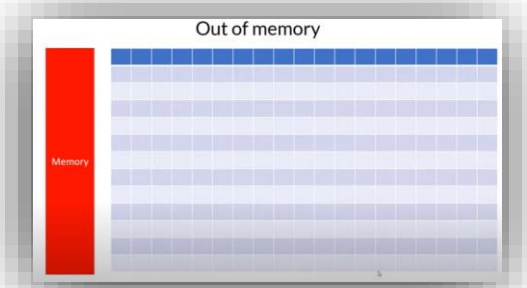
Query Folding is an optimization technique that translates M queries to native queries that can be executed in the database server, so that you don't have to import all data to Power BI. **For it to work you need to import data using views or tables.**

https://www.youtube.com/watch?v=ZSUCmi6h5SY
https://www.cloudmoyo.com/blog/decision-analytics/become-a-power-bi-expert-with-these-best-practices
https://addendanalytics.com/data-modelling-best-practices/

2 – Model finetuning

Still following Kimball's Dimensional modelling, and **to avoid performance issues on Power BI, avoid useless columns on your tables (Factuals and Dimensions).**

Imagine that you process 1M+ entries by day, and that each entry (on each day) has 50 or 100 columns, this absolutely kills the **performance**, as well as the **memory during processing** (if you can even process!…).



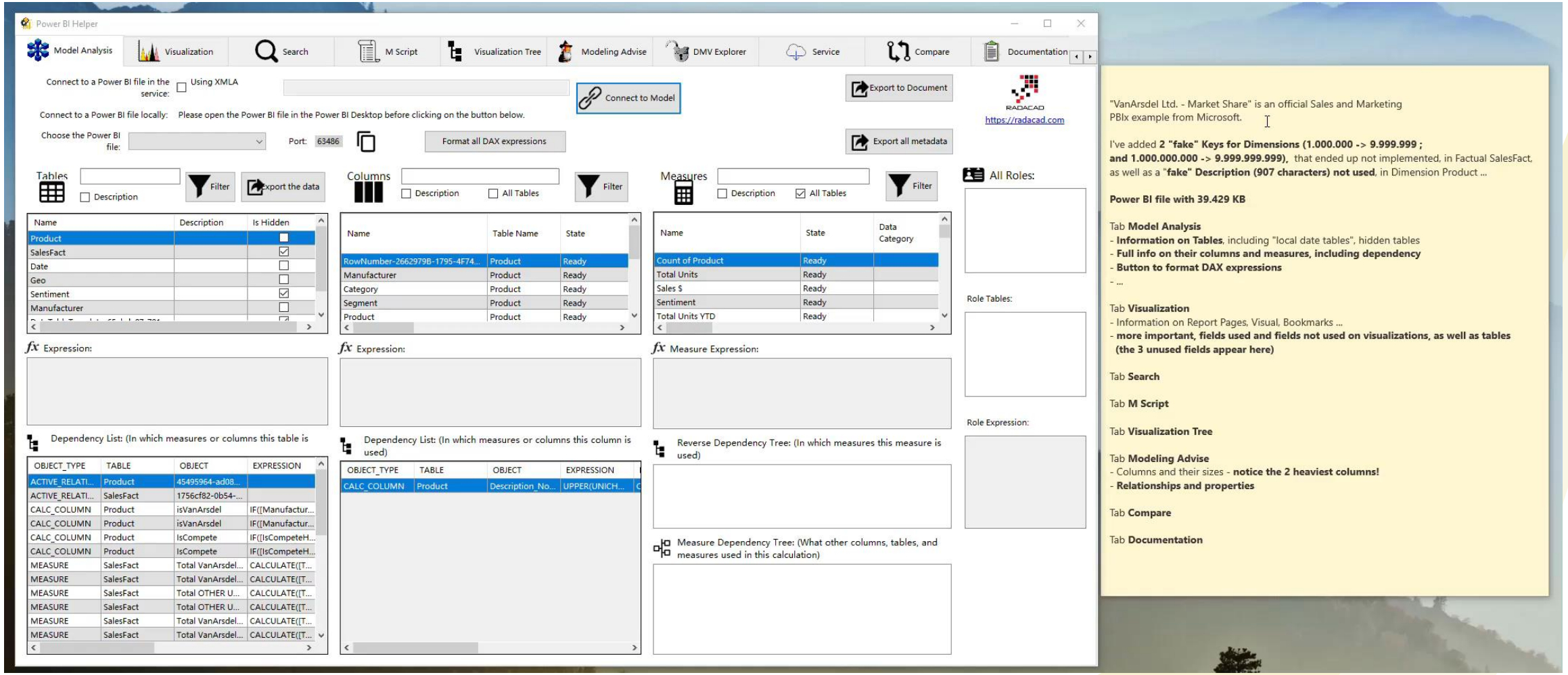
BTW, another thing that happens is **when you search for an atribute and it takes too long to find it on PBI**.

Use Power BI Helper to choose unused columns to remove.
See demo on next slide.

You should also remove the rows that aren't needed.

https://www.youtube.com/watch?v=kiVXI7zjSzY
https://radacad.com/power-bi-helper

Model finetuning (demo 1)



"VanArsdel Ltd. - Market Share" is an official Sales and Marketing PBIx example from Microsoft.

I've added **2 "fake" Keys for Dimensions (1.000.000 -> 9.999.999 ; and 1.000.000.000 -> 9.999.999.999),** that ended up not implemented, in Factual SalesFact, as well as a "**fake" Description (907 characters) not used**, in Dimension Product ...

Power BI file with 39.429 KB

Tab **Model Analysis**
- **Information on Tables**, including "local date tables", hidden tables
- **Full info on their columns and measures, including dependency**
- **Button to format DAX expressions**
- ...

Tab **Visualization**
- Information on Report Pages, Visual, Bookmarks ...
- **more important, fields used and fields not used on visualizations, as well as tables (the 3 unused fields appear here)**

Tab **Search**

Tab **M Script**

Tab **Visualization Tree**

Tab **Modeling Advise**
- Columns and their sizes - **notice the 2 heaviest columns!**
- **Relationships and properties**

Tab **Compare**

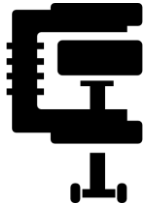
Tab **Documentation**

2.1 – Useless columns & tranformations

By reducing the number of columns, you reduce the amount of data loaded in memory in Power BI, and **more important you avoid high cardinality columns** that are only used for technical reasons (e.g. ID, timestamp).

Ensure the model is as narrow and lean as possible. Power BI works on columnar indexes, longer and leaner tables are preferred (this argument also justifies unpivoting columns)

Try to bring the columns you need calculated from data source (ETL and Views), or at least from PowerQuery, so that they are compressed through Vertipaq.



And only last, develop on DAX, preferentially using measures, in order to avoid calculated Columns.

https://www.sqlbi.com/articles/data-import-best-practices-in-power-bi/
https://exceleratorbi.com.au/which-columns-are-not-used-power-bi-data-model/
https://maqsoftware.com/expertise/powerbi/power-bi-best-practices
https://data-mozart.com/how-to-reduce-your-power-bi-model-size-by-90/
https://maqsoftware.com/expertise/powerbi/power-bi-best-practices
https://prologika.com/high-memory-usage-and-calculated-columns/

2.2 – Columns cardinality

You also should **avoid long text columns (e.g. descriptions).**

For **datetime columns, maintain only at the level client needs** (e.g. Date vs. Date & Hour; remove sec. and msec.).

If date and time are needed, split in 2 columns.

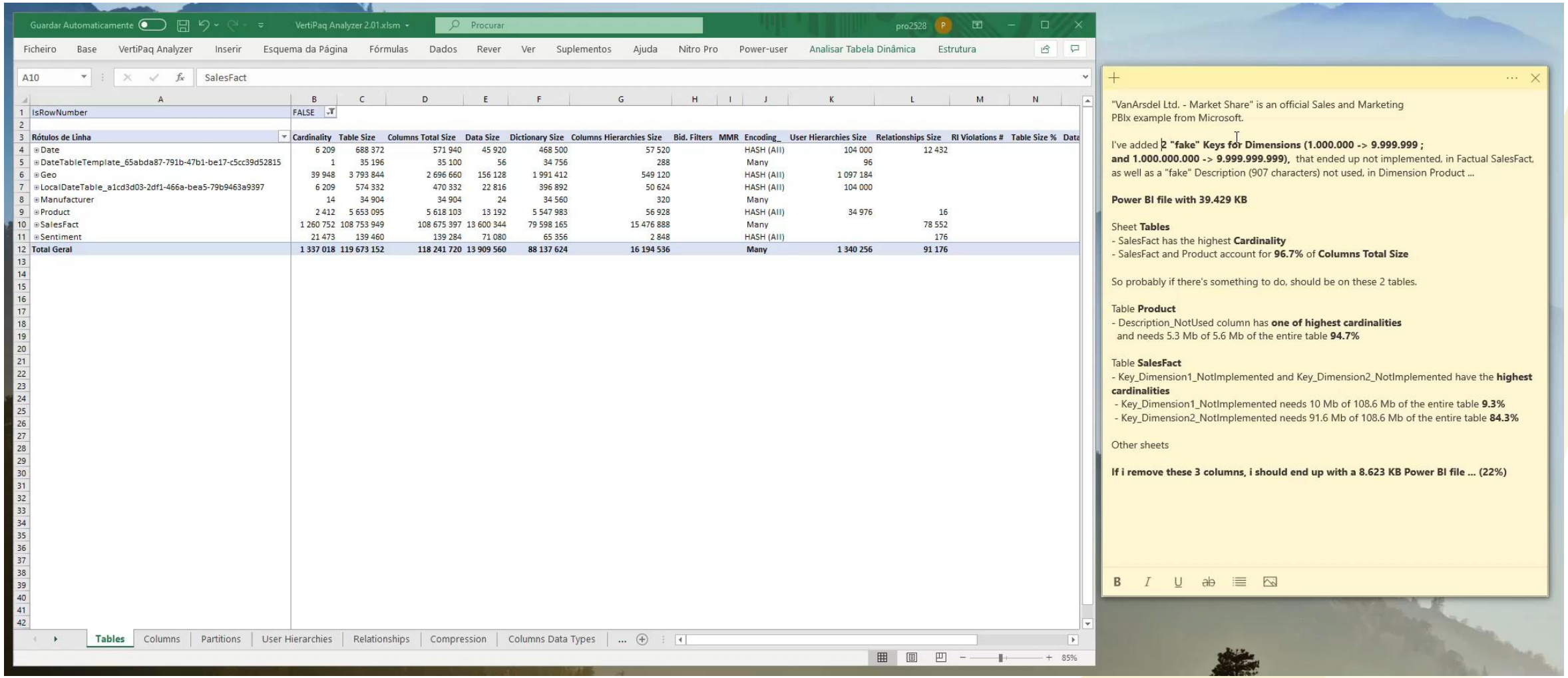
There are another techniques, like using division and modulo operators, and splitting decimals on the source side.

Be careful how you **format your data**.

Use Vertipaq Analyzer to clearly see what are the columns with more more cardinality.
See demo on next slide.

https://www.youtube.com/watch?v=vrgeDfO8RXQ&ab_channel=MicrosoftPowerBI
https://www.sqlbi.com/articles/data-import-best-practices-in-power-bi/
https://www.sqlbi.com/tools/vertipaq-analyzer/
https://data-mozart.com/how-to-reduce-your-power-bi-model-size-by-90/
https://www.youtube.com/watch?v=c-ZqToc85Yc&ab_channel=GuyinaCube

Model finetuning (demo 2)



Guardar Automaticamente — VertiPaq Analyzer 2.01.xlsm — Procurar — pro2528

Ficheiro | Base | VertiPaq Analyzer | Inserir | Esquema da Página | Fórmulas | Dados | Rever | Ver | Suplementos | Ajuda | Nitro Pro | Power-user | Analisar Tabela Dinâmica | Estrutura

A10 | SalesFact

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	IsRowNumber	FALSE												
2														
3	Rótulos de Linha	Cardinality	Table Size	Columns Total Size	Data Size	Dictionary Size	Columns Hierarchies Size	Bid. Filters	MMR	Encoding_	User Hierarchies Size	Relationships Size	RI Violations #	Table Size %	Date
4	Date	6 209	688 372	571 940	45 920	468 500	57 520			HASH (All)	104 000	12 432			
5	DateTableTemplate_65abda87-791b-47b1-be17-c5cc39d52815	1	35 196	35 100	56	34 756	288			Many	96				
6	Geo	39 948	3 793 844	2 696 660	156 128	1 991 412	549 120			HASH (All)	1 097 184				
7	LocalDateTable_a1cd3d03-2df1-466a-bea5-79b9463a9397	6 209	574 332	470 332	22 816	396 892	50 624			HASH (All)	104 000				
8	Manufacturer	14	34 904	34 904	24	34 560	320			Many					
9	Product	2 412	5 653 095	5 618 103	13 192	5 547 983	56 928			HASH (All)	34 976	16			
10	SalesFact	1 260 752	108 753 949	108 675 397	13 600 344	79 598 165	15 476 888			Many		78 552			
11	Sentiment	21 473	139 460	139 284	71 080	65 356	2 848			HASH (All)		176			
12	**Total Geral**	**1 337 018**	**119 673 152**	**118 241 720**	**13 909 560**	**88 137 624**	**16 194 536**			**Many**	**1 340 256**	**91 176**			

Tables | Columns | Partitions | User Hierarchies | Relationships | Compression | Columns Data Types | ... | 85%

"VanArsdel Ltd. - Market Share" is an official Sales and Marketing PBIx example from Microsoft.

I've added **2 "fake" Keys for Dimensions (1.000.000 -> 9.999.999 ; and 1.000.000.000 -> 9.999.999.999),** that ended up not implemented, in Factual SalesFact, as well as a "fake" Description (907 characters) not used, in Dimension Product ...

Power BI file with 39.429 KB

Sheet **Tables**
- SalesFact has the highest **Cardinality**
- SalesFact and Product account for **96.7%** of **Columns Total Size**

So probably if there's something to do, should be on these 2 tables.

Table **Product**
- Description_NotUsed column has **one of highest cardinalities** and needs 5.3 Mb of 5.6 Mb of the entire table **94.7%**

Table **SalesFact**
- Key_Dimension1_NotImplemented and Key_Dimension2_NotImplemented have the **highest cardinalities**
  - Key_Dimension1_NotImplemented needs 10 Mb of 108.6 Mb of the entire table **9.3%**
  - Key_Dimension2_NotImplemented needs 91.6 Mb of 108.6 Mb of the entire table **84.3%**

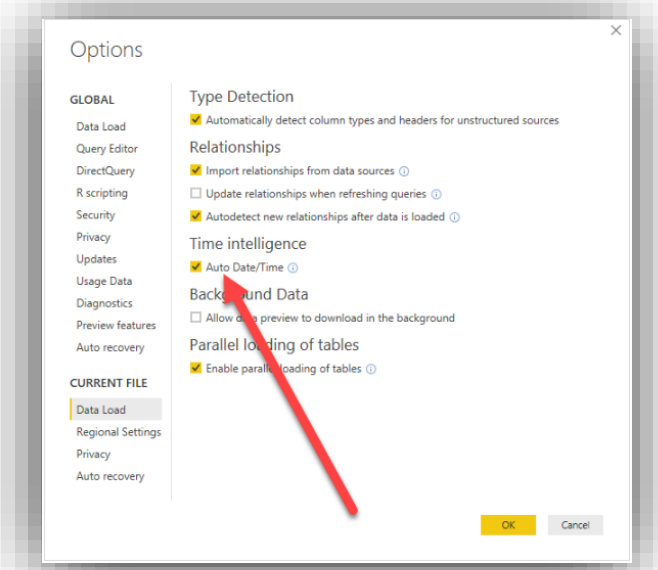
Other sheets

If i remove these 3 columns, i should end up with a 8.623 KB Power BI file ... (22%)

2.3 – Time intelligence

To create time intelligence calculations (like year to date, year over year comparison, and others) in Power BI **there is the need to create dimension Date** on the model, and mark it as date dimension.

Also, avoid using the "internal" Dim Dates, for that deselect option "Auto date/time" in options.



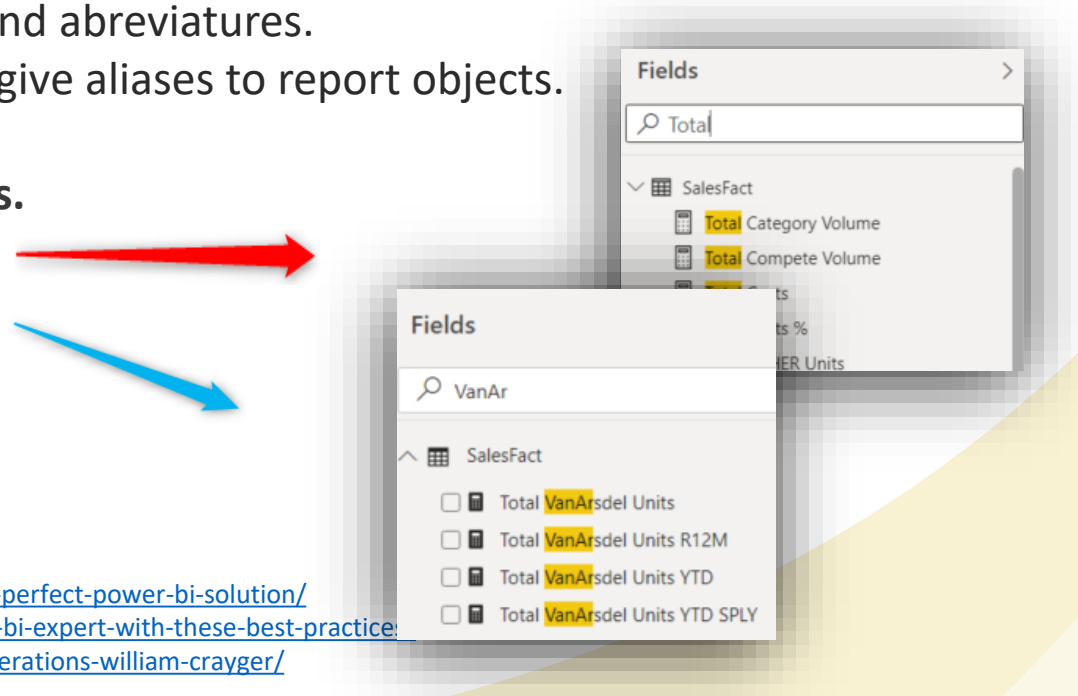
https://www.sqlbi.com/tools/dax-date-template/
https://radacad.com/create-a-date-dimension-in-power-bi-in-4-steps-step-1-calendar-columns
https://radacad.com/all-in-one-script-to-create-date-dimension-in-power-bi-using-power-query
https://www.youtube.com/watch?v=xu3uDEHtCrg
https://www.sqlbi.com/articles/automatic-time-intelligence-in-power-bi/
https://www.sqlbi.com/articles/mark-as-date-table/

2.4 – Formatting columns 1/2

For a matter of user friendliness, organization and compression, you should **carefully define table names and set measure and column names, data types and formatting**. You can also place columns in folders**. If you try to do it after, you will have to correct all the related reports.**

About nomenclature, **use names that are meaningful to your business users or intended audience.**
Otherwise, users might not understand e.g. what “Total Sales Value” is if the generally accepted term is “Net Sales Amount”. Also watch out acronims and abreviatures.
Note that Power BI provides the ability to give aliases to report objects.

This will also help searching the attributes.
Need some Total… you can’t remember?
Searching for VanArsdel… something?



https://sqldusty.com/2015/07/31/three-best-practices-for-building-the-perfect-power-bi-solution/
https://www.cloudmoyo.com/blog/decision-analytics/become-a-power-bi-expert-with-these-best-practices
https://www.linkedin.com/pulse/power-bi-best-practices-design-considerations-william-crayger/
https://maqsoftware.com/expertise/powerbi/power-bi-best-practices
https://sqlserverbi.blog/2019/08/24/power-bi-project-good-and-best-practices/

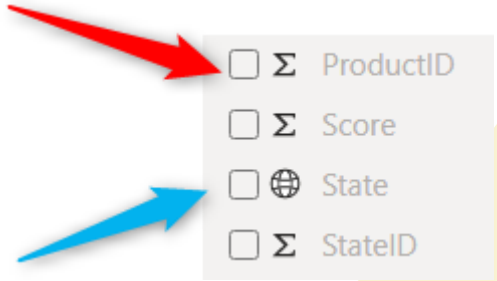
2.4 – Formatting columns 2/2

Avoid ambiguity when naming columns and measures, also hide them when they should not be directly used by users or are technical columns. For example primary and foreign keys, or columns just to specify the order. **You can also hide columns** and intermediate measures **when you create a related explicit measure**.

Format all DAX code for easy readability. Also **format numbers including nr of decimal places and thousands separator**, or scale for thousands or millions.

DAX FORMATTER

Set to **Do Not Summarize** on numeric columns to avoid implicit measures. They probably will just confuse users, or just return inadequate aggregations. Also, if you try to Analyze in Excel, they won't appear.



Don't forget to **set the Data Category** (e.g. City, Postal Code, Country, …) when appropriate.

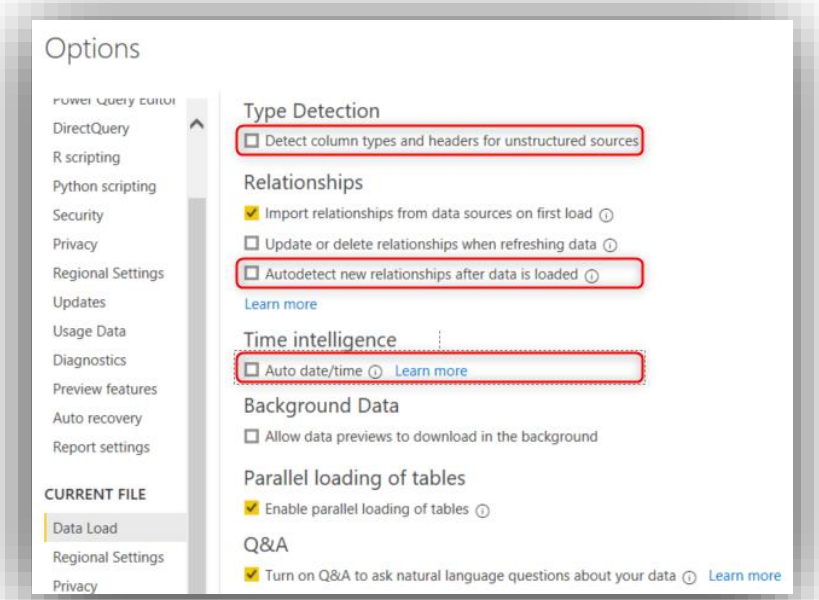
Place notes on the semantic model.

https://sqldusty.com/2015/07/31/three-best-practices-for-building-the-perfect-power-bi-solution/
https://www.cloudmoyo.com/blog/decision-analytics/become-a-power-bi-expert-with-these-best-practices/
https://www.linkedin.com/pulse/power-bi-best-practices-design-considerations-william-crayger/
https://maqsoftware.com/expertise/powerbi/power-bi-best-practices
https://sqlserverbi.blog/2019/08/24/power-bi-project-good-and-best-practices/
https://blog.crossjoin.co.uk/2020/06/28/naming-tables-columns-and-measures-in-power-bi/amp/

2.6 – Power BI Setup

Check setup in Options & settings.

For example you can uncheck column **type detection**, **Relationships** and **Auto date/time**.



You can see some more related tips on Rui Romano session on **Power BI Tips, Tricks and Hacks**.

After that you can also see Pedro Reis session **DAX The way to do it**.

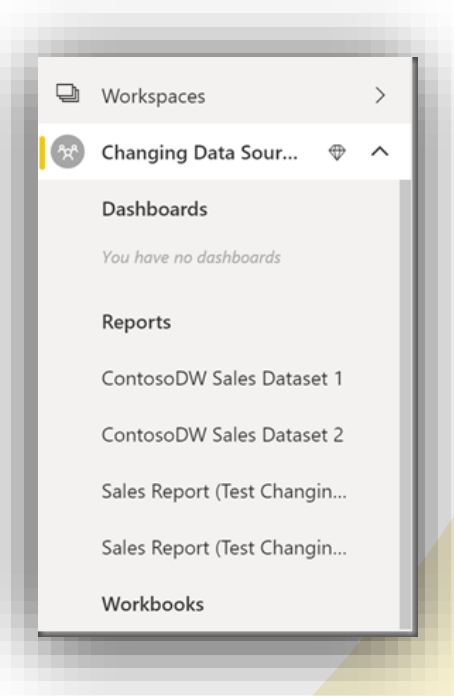
https://www.youtube.com/watch?v=xoR-2NjZCCk
https://www.cloudmoyo.com/blog/decision-analytics/become-a-power-bi-expert-with-these-best-practices/
https://www.youtube.com/watch?v=xoR-2NjZCCk&ab_channel=RADACAD
https://www.youtube.com/watch?v=dK1Snk4FMFU&trk=organization-update-content_share-video-embed_share-article_title&ab_channel=PowerBIPortugal

2.7 – Power BI file names

Give Power BI files an intuitive and friendly name, that will result in them not being confused with similar report projects, as well as associated datasets.

Workspace and report names have limited space in the Power BI Service portal navigation bar. **Try to use names that differentiate like-named items in the first 20 characters.** The full name is visible in the hover-over tooltip or by resizing the panel.

Also, be **careful with the logic you use to rename them**, because if for example you rename some Report to a PBI report V2 version, you also create a new dataset instead of using the existing one.



https://sqlserverbi.blog/2019/08/24/power-bi-project-good-and-best-practices/

// thank you!