Present your back side: 7 ways to tone your Power BI booty

In the battle for the hearts and minds of executive Excel power users, should we be making our back end more presentable? Excel plugins are becoming more commonplace, and pro licencing more widespread, meaning more and more people are accessing our datasets on their own terms.

Patrick LeBlanc from Guy in a Cube did a great video recently, all about getting Excel users to switch to Power BI. The video breaks down some of the cool things you can do in excel when you're connected to a Power BI dataset and it's worth a watch.

While I've personally been fanatical about ultra clean and usable datasets for a long time, this video highlights an emerging industry trend: We can no longer assume that we know how data consumers will access the data our Power BI service. This is a good thing, as it's opening Power BI to an increasing audience. Dataset developers need to not only be aware of this, but perhaps we should embrace it as a way to help our work to reach people where they live.

Here are 7 ways we can ensure that our back sides are clean, mean, and ready to present!

1. Names should be natural

Removing CamelCase and unfriendly_underscored_names in tables and fields is the easiest thing we can change to make life easier for our report consumers. Also, why should we make our end readers care about facts and dimensions? "Sales" and "Store", makes so much more sense to a standard user than "Dim_Store" and "Fact_Sales" (or worse Fact_Sales_Header and Fact_Sales_Lines).

Once we embrace the discipline of naming things in a more natural way a multitude of other questions about how to name our fields start to surface. For example:

- Plural or singular dimension table names? Store or Stores?
- Redundancy: [Store].[Store Name], [Store].[Name], or [Store].[Store]?
- Abbreviations: "Finance Term Number 2" or "Fin Term No 2"?
- Hints: "Date (yyyy-mm-dd)" or just "Date"?

A helpful question we can ask when naming things is how will this be named in an actual report, in a chart or on an axis? How is this most likely to be referred to by the business in a finished, fully polished report? This is how it should be named.

Rule of thumb: Name for reporting. Don't make your data consumers rename anything if it can be avoided. (Bonus: This will also keep your data lineage clearer)

2. Hide your bits and pieces

If you're going to display your rear end, then you really should make sure it's nice and clean. Hide your developer columns. Hide away things that are not a part of the data that will be reported on. Too many data consumers find themselves wading through an inscrutable sea of identifiers, surrogate keys, audit fields and other distractions. This is poor hygiene.

Rule of thumb: If it will never be displayed on a visual, then it should not be visible in your model.

3. Timing is everything – keep your dates clear

Calendar tables (or date tables) are a great way to ensure corporate calendars / financial periods etc are available to sort, slice and otherwise present data. However, with large models often containing many dates, developers will sometimes create multiple calendars. One for order date, one for shipping date, one for returns date etc. This is not the optimal way to go for a solution that will be easy for a data consumer to traverse.

As much as it is possible, we should strive to contain our date logic to a single calendar table.

Multiple inactive relationships from our single calendar table to our facts is a preferable approach for several reasons:

1. Typically, a data consumer will want to see all the orders, deliveries and returns based on a single date selection. Don't force them to filter 3 separate calendars.
2. DAX measures are more than capable of supporting multiple streams of logical filtering based on a single calendar with multiple secondary relationships
3. When we contain our date logic in our measures it means there is less likelihood of a data consumer misunderstanding the calendar system and generating incorrect outputs

Multiple calendars can also be combined in a single calendar table. Have some people working on a US tax year and others on an Australian tax year? A single table with a region field for slicing will make this a no-brainer. The minor hit of many-to-many calendar relationships is usually worth it.

Also consider using a period (time frame) table to assist with slicing and dicing. This small step will bring your report consumers a lot of power that otherwise can typically only be unlocked with a sprawling set of time-indexed measures.

Finally, consider a time-of-day for easy time of day analysis.

Combining each of these approaches almost makes time-based analysis too easy – one measure for actual, one for last year and one for comparison in combination with the above calendarisation can replace a truckload of highly time specific DAX measures.

Of course, it is not practical to connect up every date, time and datetime element in your data. Judgement must be exercised.

Rule of thumb: If your date field has a measure dependent on it (e.g. Order dollars depends on order date, Returns depends on return date) it deserves the full treatment. Otherwise, you might get by with a simple date in the table.

4. Keep your dimensions in check

Do you really need Product table and a Product Category table? Simplify it – merge them down to one table. With VertiPaq compression, the dataset won't mind a few low cardinality columns in your dimensions and your data consumers will thank you for the simplicity of a reduced table count.

Create Hierarchies to simplify the creation of drill-down visuals.

Don't be shy about grouping fields within dimensions, either. We have display folders to keep related fields organised. This is an overlooked feature that can greatly improve dataset usability.

Rule of Thumb: Flatten dimensions and avoid the dreaded snowflake schema as much as possible, short of introducing many-to-many relationships with your facts. Use display folders and hierarchies to organise data within your dimensions, where appropriate.

5. Organise your measurements

Measures don't care where you park them. They float. Measures depend on filter context, meaning they don't conceptually belong to your fact tables. The data consumer defines the filter context, and the measures provide results, often referring to multiple tables at a time.

For this reason, there is nothing to say we can't just organise all the measures in the same place. Consider using a single table with nothing but measures in it. In fact, let's put every single measure in the model in there into that one "all measures" table. You'll see that Power BI anticipates this and will give your measure table a nice measure icon and push it up to the top of your model in the field pane.

While we are at it, let's organise the measures table properly. Group your measures in folders and sub folders. Create a structure that makes sense.

And name them correctly. It's often a great idea to give an indication of the output of a measure in the name. Instead of "Total Sales", maybe "Sales $" would be better. Think about how the measure name will appear in visuals and name it with this in mind.

In fact, while we are considering measure names, let's never use the word "total" again in our measure names! The data consumer will decide if the output of a measure is to be a total, sub-total or a line amount.

Whatever way you choose to organise and name your measures, consistency is vital. If your sales measures are called "Sales $" and "Sales $ LY" then your returns measures should be called "Returns $" and "Returns $ LY".

Finally, it should go without saying, but we should also take time to ensure that we have correctly formatted our measures to render correctly by default. Don't force the data consumer to add brackets for negative amounts, or to round up the decimal places.

Rule of thumb: Well organised, well named, properly formatted measures are a joy to use. Make sure to budget that little bit of extra time to ensure they are well organised, easy to find and identify and read-to-go with no re-naming or re-formatting.

6. Implicit measures are a potentially risky business

By default, a Power BI dataset will allow implicit measures, i.e. if a data consumer uses a number field in a visual, Power BI will automatically sum it (or average it, or count it…). This leads to simplistic data aggregation – e.g., click and drag the sales dollars figure into a chart and it will automatically become a sales sum.

But what if your business rules say that you need to exclude line items for certain SKUs from your sales? What if your data consumer doesn't know this? Perhaps we should force them to use the developed and test approved Sales measure instead. This is known as discouraging implicit measures.

Dogged followers of the explicit measures only approach will often go as far as to hide their fact tables. Whether this is a good or a bad thing is a matter of context. Larger datasets typically benefit more from discouraging implicit measures, whereas smaller, specific use datasets may benefit from the increased freedom afforded by permitting implicit measures.

Rule of Thumb: Developers need to understand who their data consumers are and how they will interact with the data. Explicit measures de-risk reporting, while also restricting flexibility. You will need the right approach for the right situation.

7. Sometimes a little perspective makes all the difference

Perspectives allow us to curate a subset of the dataset fields for particular data consumers. Perhaps store managers don't need access to the GL tables. Perhaps finance data consumers only need to journal lines and related data. Everything else then becomes at best a distraction for these user groups.

The solution can't be to just hide fields, as we still need to see them depending on context. The solution is not to break up your dataset either, as we still want to be able to cross filter between business areas.

Using perspectives to define groups of related fields within your model is a great solution to help drive data consumers to the appropriate areas of your dataset.

Rule of Thumb: Tailoring the system based on use case simplifies interaction and empowers data consumers. With judicious use of perspectives, you can show your data consumers that you know who they are and what they need.

Conclusion

Since we cannot assume we know how our data customers will access the data, it will serve us well to design datasets with the same care and attention to detail that is often afforded only to the reports that connect to them. We need to see our datasets as a critical part of the reporting solution and design them accordingly with usability in mind. This is part of how we will pull in our excel user base and grow Power BI adoption across the business.

To see how we can help your business make their Power BI datasets clean and usable, please contact a Fujitsu Data & AI specialist now.

Contact

Fujitsu Data & AI
+61 3 9924 3000