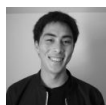


[← BACK TO BLOG](#)

6 tips to make your dashboards more performant

[SHARE](#)

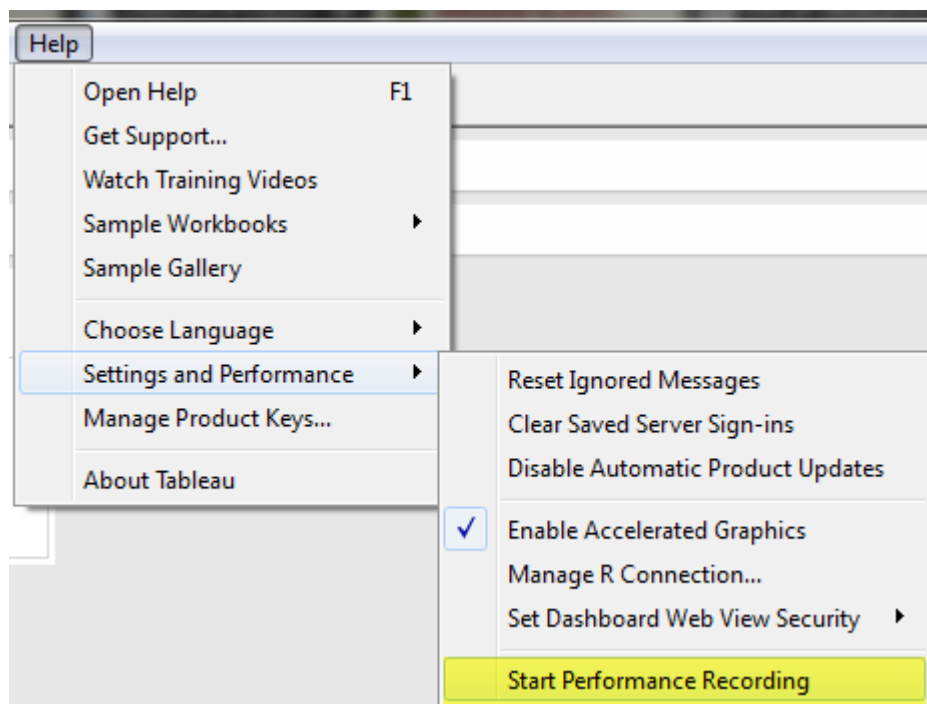
NICHOLAS HARA
JANUARY 15, 2016

Note: Ryan Lempa and Darren McGurran also contributed to this piece.

We here at Tableau are very proud of how easy it is to see and understand data with Tableau. Once you get started, it's intuitive to dive deeper by adding more and more fields, formulae, and calculations to a simple visualization—until it becomes slower and slower to render. In a world where two-second response times can lose an audience, performance is crucial.

So where do I start?

So how can you make your dashboards run faster? Your first step is to identify the problem spots by **running and interpreting your performance recording**. The performance recorder is every Tableau speed demon's ticket to the fast lane. The performance recorder can pinpoint slow worksheets, slow queries, and long render-times on a dashboard. It even shows the query text, allowing you to work with your database team on optimizing at the database level.



Now that you know which views or data connections are slowing you down, below are six tips to make those dashboards more performant. For each tip, we've listed the most common causes of performance degradation as well as some quick solutions.

1. Your data strategy drives performance

Extracts are typically much faster to work with than a live data source, and are especially great for prototyping. The key is to use domain-specific cuts of your data. The Data Engine is not intended to be a replacement for a data warehouse. Rather, it's meant to be a supplement for fast prototyping and data discovery.

Since an extract is a columnar store, the wider the data set, the slower the query time.

- **Minimize the number of fields** based on the analysis being performed. Use the **hide all unused fields** option to remove unused columns from a data source.
- **Minimize the number of records.** Use extract filters to keep only the data you need.
- **Optimize extracts** to speed up future queries by materializing calculations, removing columns and the use of accelerated views.

Keep in mind: Extracts are not always the long-term solution. The typical extent of an extract is between 500 million to one billion rows; mileage will vary. When querying

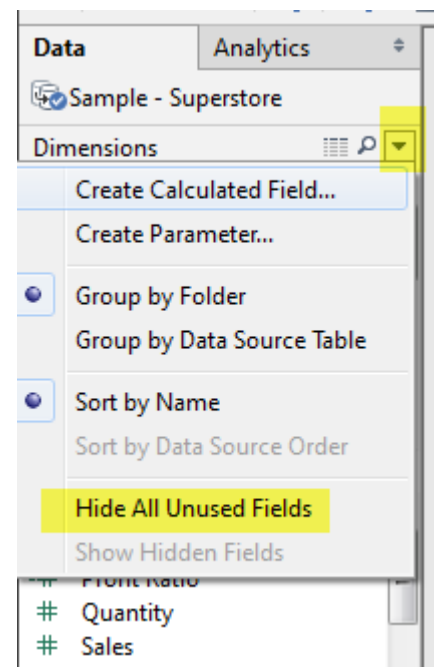
against constantly-refreshing data, a live connection often makes more sense when operationalizing the view.

For more information on data extracts, check out these additional resources:

[Video: Using and Refreshing Extracts](#)

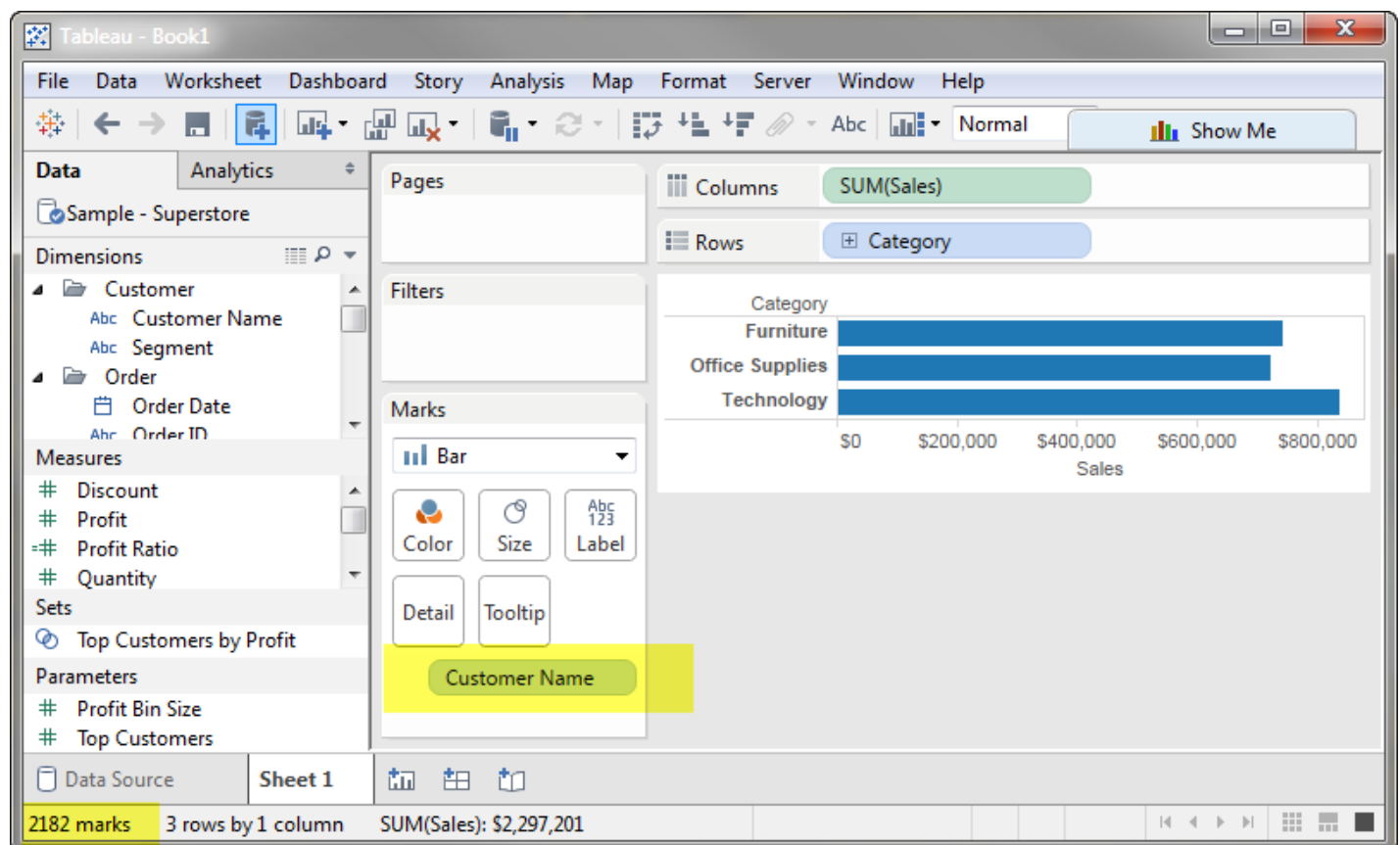
[Online Help for Extracting Data](#)

[Understanding Tableau Data Extracts \(three-part series\)](#)



2. Reduce the marks (data points) in your view

When data is highly granular, Tableau must render and precisely place each element. Each mark represents a batch that Tableau must parse. More marks create more batches; drawing 1,000 points on a graph is more difficult than drawing three bars in a chart.



Large crosstabs with a bevy of quick filters can cause increased load times when you try to view all the rows and dimensions on a Tableau view.

Excessive marks (think: data points) on a view also reduce the visual analytics value. Large, slow, manual table scans can cause information overload and make it harder to see and understand your data.

Here's how you can avoid this trap:

- **Practice guided analytics.** There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly-granular views at the speed of thought.
- **Remove unneeded dimensions** from the detail shelf.
- **Explore.** Try displaying your data in different types of views.

3. Limit your filters by number and type

Filtering in Tableau is extremely powerful and expressive. However, inefficient and excessive filters are one of the most common causes of poorly performing workbooks and dashboards. Note: Showing the filter dialog requires Tableau to load its members and may create extra queries, especially if the filtered dimension is not in the view.

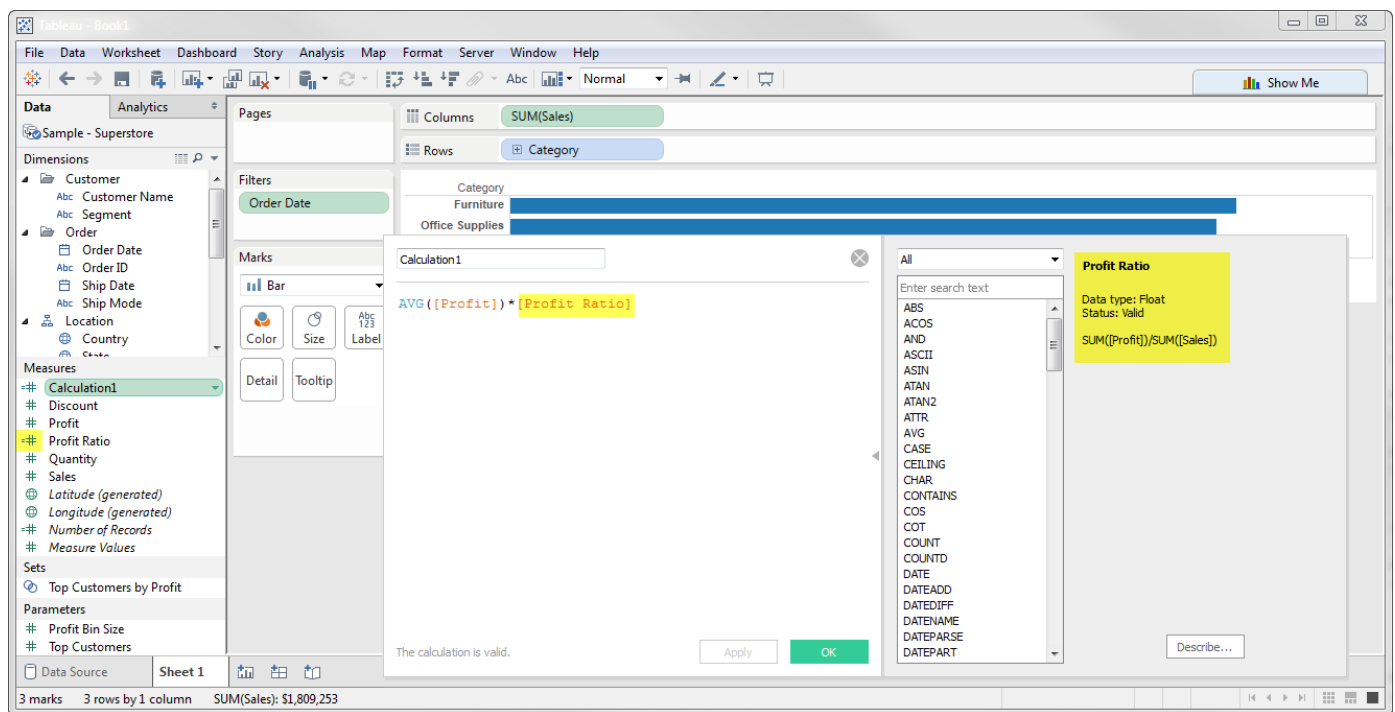
- **Reduce the number of filters in use.** Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.
- **Use an include filter.** Exclude filters load the entire domain of a dimension, while include filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.
- **Use a continuous date filter.** Continuous date filters (relative and range-of-date filters) can take advantage of the indexing properties in your database and are faster than discrete date filters.
- **Use Boolean or numeric filters.** Computers process integers and Booleans (t/f) much faster than strings.
- **Use parameters and action filters.** These reduce the query load (and work across data sources).

Tableau Filter Order



4. Optimize and materialize your calculations

- **Perform calculations in the database.** Wherever possible, especially on production views, perform calculations in the database to reduce overhead in Tableau. Aggregate calculations are great for calculated fields in Tableau. Perform row-level calculations in the database when you can.
- **Reduce the number of nested calculations.** Just like Russian nesting dolls, unpacking a calculation and then building it takes longer for each extra layer.



- **Reduce the granularity of LOD or table calculations in the view.** The more granular the calculation, the longer it takes.
 - **LODs** – Look at the number of unique dimension members in the calculation.
 - **Table Calculations** – the more marks in the view, the longer it will take to calculate.
- **Where possible, use MIN or MAX instead of AVG.** AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG.
- **Make groups with calculations.** Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.
- **Use Booleans or numeric calculations instead of string calculations.** Computers can process integers and Booleans (t/f) much faster than strings.
Boolean>Int>Float>Date>DateTime>String

5. Take advantage of Tableau's query optimization

- **Blend on low-granularity dimensions.** The more members in a blend, the longer it takes. Blending aggregates the data to the level of the relationship. Blends are not meant to replace row-level joins.
- **Minimize joined tables.** Lots of joins take lots of time. If you find yourself creating a data connection with many joined tables, it may be faster to materialize the view in the database.

- **Assume referential integrity** if your database is configured with this option.
- **Remove custom SQL.** Tableau can take advantage of database optimizations when the data connection does not use custom SQL.

6. Clean up your workbooks!

- **Reduce dashboard scope.** Excess worksheets on a dashboard can impact performance.
- **Delete or consolidate unused worksheets and data sources.** A clean workbook is a happy workbook.

For more details on these areas and many more, check out [our whitepaper on designing efficient workbooks](#).

You might also be interested in...

Comments

Submitted by John Jensen (not verified) on January 15, 2016 – 12:23pm

Absolutely brilliant work. While there are many posts and white papers and PS the talk about the subject... This is a perfect balance of content and detail without being a novel!

Nicely done

[Reply](#)

Submitted by KK Molugu (not verified) on January 15, 2016 – 1:25pm

Nice post!

Reply

Submitted by Venkat Kanithi on January 15, 2016 - 2:10pm

Thank you sharing important and handy information.

Reply

Submitted by Baljeet (not verified) on January 16, 2016 - 1:53am

Precise information without lot of verbose.

Reply

Submitted by David Walp on January 16, 2016 - 7:08am

Very useful. Thank You.

Reply

Submitted by Josh Tapley (not verified) on January 16, 2016 - 8:28am

Perfect summary! This looks like a great 1 hour webinar for my user base.

Reply

Submitted by Alex Blakemore on January 16, 2016 - 2:33pm

Can you explain a bit more what you mean by "Exclude filters load the entire domain of a dimension, while include filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members."

When you say "load the entire domain", are you talking about populating a quick filter list?

For instance, an exclude filter could generate efficient SQL in some cases such as '...where status <> "Closed"' that ought to be more efficient than SQL generated by an include filter such as 'where status in ("New", "Review", "Implementation", "Test", "Approval")'. So would you mind explaining a bit more about that recommendation, or

pointing to other material that describes it. On the surface, it seems odd to say that include is always more performant than include. I would think whichever approach required fewer comparisons would be superior.

Reply

Submitted by John Sobczak (not verified) on January 18, 2016 – 6:09am

This is really good! I think this is the first time I have ever seen in writing that extracts are generally faster than live data sources which has been my experience as well.

Reply

Submitted by Nicholas Hara on January 26, 2016 – 1:51pm

Hi John!

Extracts can provide huge speed benefits, but there are some situations that a live connection may be faster. If you are interested in exploring the topic further, we've written some great articles to help guide you on making the right call for your needs.

<http://www.tableau.com/about/blog/2014/7/understanding-tableau-data-extr...> –

This article explains Tableau's Data Extract at its core and provides the baseline for where and how Tableau makes tweaks to improve performance.

<http://www.tableau.com/about/blog/2014/7/why-use-tableau-data-extracts-3...> –

This article looks at optimal use cases for extracts.

<http://www.tableau.com/tableau-data-extracts-part3> – And this one is a simple checklist for Extract use best practices.

Hope these resources help!

Reply

Submitted by Dan Cory on January 22, 2016 – 4:05pm

Alex B. –

Filters that use the custom value list setting generally map directly to the simple SQL.

But filters using the "Select from list" setting, will check the domain of the column to try to find more optimal ways of expressing the filter (such as by using a range query). If it's an exclude filter, that may include reading the domain of the column to figure out the optimal way to express the filter.

If the domain of the column contains null, this further impacts the query Tableau will generate. If you can declare columns "not null" in your database, both Tableau and the database will have an easier time optimizing your query.

I hope that helps!

Dan

[Reply](#)

Submitted by Sasiaknth Reddy on January 28, 2016 - 8:52pm

Critically Acclaimed!

[Reply](#)

Submitted by Joao Cruz on March 14, 2016 - 7:53am

Hi,

Thanks for the post, good overview.

Could you please clarify point six. I assume that it was an afterthought as the URL says '5 tips...'?:) Correct me if I am wrong but Tableau only renders what is on a View/Dashboard so the size of the workbook and number of sheets should not impact performance significantly. (Although I agree we should try our best to keep the workbook clean.)

Thanks

[Reply](#)

Submitted by Cor Bader on May 20, 2016 - 5:27pm

Hi Joao,

A couple of points to be careful with that thought.

1. If you end up publishing a workbook to Tableau Server with 'Show Sheets as tabs' checked, Tableau will need to load the entire workbook (all the XML) even if you are just viewing one sheet. If there are many sheets this could affect performance.
2. In a dashboard, any item that has a container we refer to as a zone. If there is a excessive amount of zones this also will affect performance.

Hope that helps!

[Reply](#)

Submitted by Cor Bader on May 20, 2016 - 5:20pm

Awesome work Nick!

A side note. Crosstab view (text tables) will perform much faster starting in 9.2 using single pane optimization. Make sure you are on a fast upgrade cycle as Tableau is constantly improving performance to keep you in the flow with your data.

"Text Table and Highlight Table Optimization

Text tables and highlight tables are optimized to enhance rendering, resizing, and removing of columns."

<http://www.tableau.com/new-features/9.2>

[Reply](#)

Submitted by Geoffrey (not verified) on July 28, 2017 - 10:17am

This sounds like good info, but I thought I didn't have to be a DBA to get interesting data quickly. As your video suggests?

[Reply](#)

Submitted by Siva Sankara Chagam on August 11, 2017 - 9:54pm

Info is too good. Thank you.

[Reply](#)

Submitted by rushendra sriram on August 20, 2017 – 11:55am

Some of the articles published as no. context filters in the dashboard to improve the performance. I would like to know more on this topic.

Regards

Rushendra SV

[Reply](#)

Submitted by narendra (not verified) on September 6, 2017 – 10:49pm

How to improve the performance Live Tableau dashboards?

I'll try to do most of the tips you gave me... But little bit (1%) of change i think.

Please give me the suggestions

[Reply](#)

Submitted by Kym Dahm (not verified) on September 19, 2017 – 5:23pm

Very helpful post. Definitely sharing with my customers who have performance issues with their dashboards!

[Reply](#)

Submitted by rabson.lungu.2.... (not verified) on October 17, 2017 – 2:02pm

Hi Nicholas, A trillion thanks! This is brilliant work at best. The article is a treasure trove of good tips for the practice. Great KT based on experience.

[Reply](#)

Add new comment

Your name Kristy M.

Comment *

☐ Notify me when new comments are posted

SAVE

Subscribe to our blog

Business E-mail *

Kristy.McGee@tn.gov

SUBSCRIBE

WE RESPECT YOUR PRIVACY

English

[System Status](#) [Blog](#) [Academic](#) [Careers](#) [Contact Us](#)[LEGAL](#) [PRIVACY](#) [UNINSTALL](#)

© 2003-2018 TABLEAU SOFTWARE. ALL RIGHTS RESERVED