

Reduce processing load and maintain dashboard effectiveness

Previously, you learned about the importance of optimizing processing speed for dashboards. **Processing speed** describes how quickly a program can update and load a specified amount of data. Basically, it's how fast your dashboard can deliver answers to users. Processing speed is usually determined by the volume of data, the number of measures, and the number of dimensions. This is another example of a trade-off: you have to balance various factors, often prioritizing one element while sacrificing another, in order to arrive at the best possible result. In this case, there is a trade-off between processing speed and workload. This reading will offer solutions that enable you to reduce processing load while maintaining the effectiveness of your dashboard.

Reduce processing load

One of the primary ways you can work to optimize your processing speed is by reducing the processing load. You can do this by:

- **Pre-aggregating:** This is the process of performing calculations on data while it is still in the database. Pre-aggregating data will transform data into a state that's closer to what you ultimately need because some necessary calculations will happen before the data is sent to the data visualization tool. The trade-off is that your pipeline will involve more steps and your dataset uploaded into the visualization tool will be less flexible, but your users will get the information they need more quickly.
- **Using JOINS:** JOINS are used to combine rows from two or more tables based on a related column. This basically merges tables together before they're ever used in the dashboard. This can save a lot of processing load in the actual dashboard. However, if you are trying to join a full table, it can be more of a burden to the system. This is caused by the dimensionality of the tables. For example, joining a one million row table with a 100 million row table will most likely generate a lot of overhead every time the dash is updated. So it's important to think carefully about how you use JOINS to reduce processing load!
- **Filtering:** Filtering is the process of showing only the data that meets a specified criteria while hiding the rest. Filtering the data early in your dashboard's processing means that it doesn't have to sort through data that isn't actually going to be used. The tradeoff of this is that this means less data is available for your users to view on their own.
- **Linking to external locations:** In cases where you have data in your dashboard that you can provide context for outside of the dashboard and which can help cut down on the processing load, you can link out to that location for users to explore on their own.
- **Avoiding user-defined functions:** Users making requests of your dashboard can add a lot of load to the processing work it's doing. Consider the kinds of questions that users might have when designing the dashboard so that you can address them without the users themselves having to input functions repeatedly.
- **Deciding between data views and tables:** Tables contain actual data. Data views are the result of a stored data query that preserves business logic and can be queried like a database. Data views often require much less processing load because they don't contain actual data, just a view of the data. This makes them less flexible, so you'll want to consider how interactive you need the data in your dashboard to be.

Key takeaways

When you are considering dashboard design, you'll have to consider processing speed and load and decide how to best balance them to deliver the answers your stakeholders need as quickly as possible. This can be challenging, but you can apply the strategies described in this reading to reduce processing load and improve performance.