ISSN No:-2456-2165

# Integrating Diverse Data Sources in Tableau to Elevate Performance

Ayesha Umaima |Data Engineer Business Intelligence

**Abstract:- This paper will focus on the technical perspective of bringing diverse data and Building compelling Tableau dashboards on them with a view towards efficiency, fix mistakes and deliver fast, flexible, and compelling dashboards with an intend to keep the user experience impactful.**

**As the dashboard author, you need to understand the bigger picture of your project and where the user fits. This paper will give you the best way to implement dashboard requirements and focus on what modifications you can make while integrating data sources in tableau to smooth out the path to production.**

## I. INTRODUCTION

Tableau is a fantastic ad-hoc analytical tool, letting the dashboard author ask a question, see the answer quickly through visuals, and then iterate with another question. These reports often turn into quick prototypes that evolve into production dashboards. Features of Tableau that help speed ad-hoc insight, like being able to easily connect to data sources, make new calculations, and add as many filters as you desire, may result in additional workload or complexity if not streamlined for production.

A Tableau Dashboard has four core elements: Data, Calculations, Worksheets, and Dashboard Layout. We start by connecting to our data, enhancing it by creating calculations, using Tableau's interface to query and visualize the data, and finally building out our dashboards. The path is rarely linear and has many iterations, but these are the elements we control as dashboard authors that impact efficiency.

## II. DATA SOURCES IN TABLEAU

Tableau supports a wide range of connections to data on numerous platforms.

- Data sources that use files, including Excel and CSV.
- Relational database data sources, including SQL Server, Snowflake, Oracle, and Teradata
- OLAP data sources, such as Oracle Essbase and Microsoft Analysis Services.
- Hadoop or other No SQL data sources.
- •Data sources that are hosted in the cloud, including Salesforce, Google, etc.
- Native connectors are available from Tableau and are created and optimised for the types of data present in the supported files and databases. If the file or database type you require is one of those listed under Connect, then use this native connector to connect to your data. If your file or database type isn't listed, you might be able to establish your own connection using Other Databases (JDBC), Other Databases (ODBC), a Web Data Connector, or a Connector Plugin made with the Tableau Connector SDK.

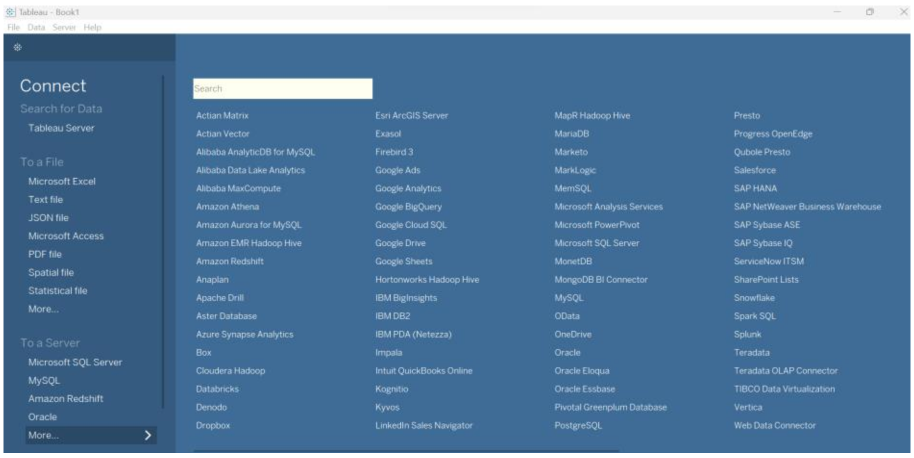


Fig. 1: Connect to your Data

## III. LIVE CONNECTIONS AND HYPER EXTRACTS

For most connections, Tableau can either query the data live or create a Hyper Data Extract (a .twb file). When querying live, that means that Tableau will generate queries to the underlying data source as you build visualizations and dashboards. Since data sources come in all shapes and sizes, slow and fast, the performance of your data source will have a huge impact on your dashboard.

If your data source is slow, the Tableau Hyper Extract is your secret weapon for performance. In my experience, the majority of environments would benefit from using extracts.

### *A. Elevate Live Connections Performance:*

When using a live data connection, you'll be very dependent on the underlying data source's ability to process your requests in a timely manner. In cases where you're connecting to a relational database, your database administrator has a role in helping performance by doing things such as index tuning and proper data modeling.

a) Leave *the complex calculations to a database*:
Consult your data team about adjusting the data source to improve performance. Anything you can do to improve performance before the data even enters Tableau is recommended. It would be wise to incorporate row-level calculations into the data source.

b) *Cast data types in data source***:**
A lot of times values need to be cast from one type to another (string to integer, decimal to integer, etc) so that it's properly displayed or can be used properly in calculations. Doing this in Tableau in every relevant query is a lot of wasted processing. By making sure you have the right data types stored in your data source you can significantly speed up query results.

### *B. Using Extracts to improve performance*

Tableau's Hyper Extracts are a purpose-built analytic data store that allows you to offload processing from your underlying data source to Tableau's Hyper engine. Extract is recommended due to its numerous benefits, The biggest impact we can do is to focus on our Extract to the exact needs of your workbook.

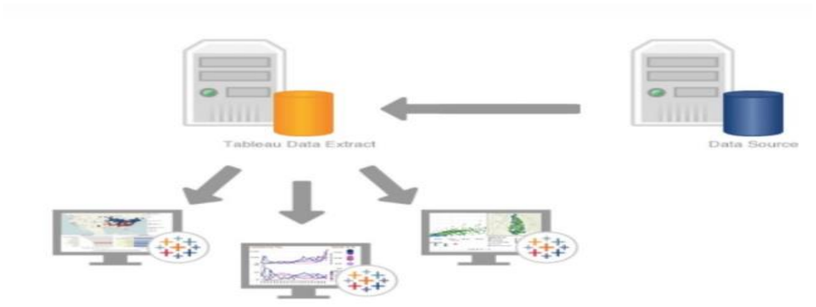
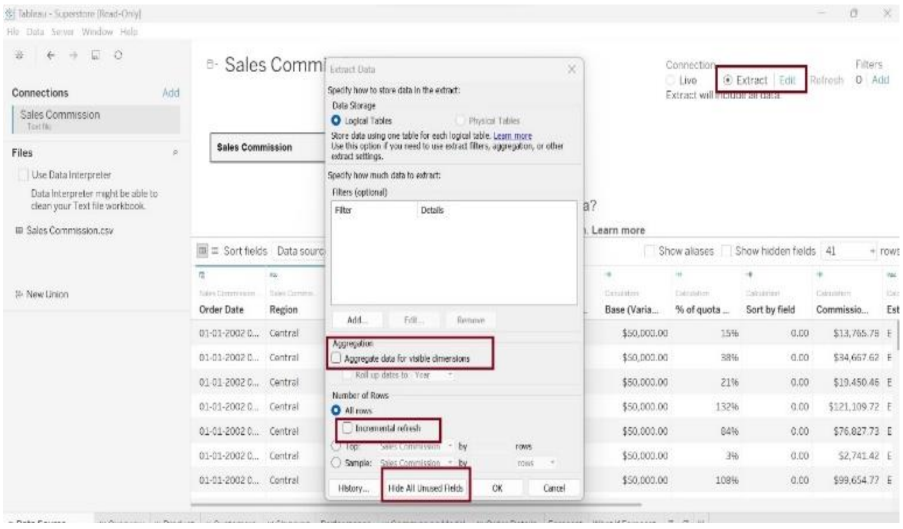


Fig. 2: Tableau Hyper Extract Process Flow.

a) Pre- aggregate data
You may also aggregate your data with Tableau for all visible dimensions. The row-level data is excluded from this kind of extract. Instead, it merely includes only the compiled information. Depending on the kind of study done, it may be the best option for various visualisations. An aggregated extract is more efficient at producing quick performance for your dashboards and spreadsheets because it is smaller than a regular data extract.



Data Interpreter might be able to clean your Text file workbook.

Extract will include all data

Specify how to store data in the extract:
Data Storage
Store data using one table for each logical table. Learn more
Use this option if you need to use extract filters, aggregation, or other extract settings.
Specify how much data to extract:
Filters (optional)

| Filter | Details |
|---|---|

Aggregation
☐ Aggregate data for visible dimensions
Roll up dates to Year
Number of Rows
● All rows
☐ Incremental refresh
○ Top: Sales Commission by rows
○ Sample: Sales Commission by rows

| Order Date | Region | Base (Varia... | % of quota ... | Sort by field | Commissio... | Est |
|---|---|---|---|---|---|---|
| 01-01-2002 0... | Central | $50,000.00 | 15% | 0.00 | $13,765.78 | E |
| 01-01-2002 0... | Central | $50,000.00 | 38% | 0.00 | $34,667.62 | E |
| 01-01-2002 0... | Central | $50,000.00 | 21% | 0.00 | $19,450.46 | E |
| 01-01-2002 0... | Central | $50,000.00 | 132% | 0.00 | $121,109.72 | E |
| 01-01-2002 0... | Central | $50,000.00 | 84% | 0.00 | $76,827.73 | E |
| 01-01-2002 0... | Central | $50,000.00 | 3% | 0.00 | $2,741.42 | E |
| 01-01-2002 0... | Central | $50,000.00 | 108% | 0.00 | $99,654.77 | E |

Fig. 3: Tableau Data Source Filter Options

b) Use Data Source filters
A data source filter can be included in the create extract dialog. This filter will leave out any data from the extract, making it smaller and faster.

c) Hide All Unused Fields
When we initially create the extract, it's recommended to skip this step, since we may need columns, you didn't expect at first and bringing a column back will require the extract to be regenerated. However, in production, this is a quick and easy way to decrease the size of your extract and speed up your queries. When you press this button, Tableau will remove any field in your data that isn't referenced by your Tableau workbook.

d) Incremental refresh:
It's best practice to use incremental when you can and to set the appropriate refresh increment that your users need. Often daily and weekly refreshes are the most common need for the business.

## IV. TRANSFORMING DATA IN THE DATA SOURCE INTERFACE

Preparing data before you start building any visualizations in the sheet can reduce the overhead encountered in the later stage. This will give you much cleaner data to consume. The data source interface provides you many features like interpreter, grouping alike items into a field, hiding unwanted fields, renaming any fields, splitting the field based on delimiter, create calculated fields and pivot two fields.

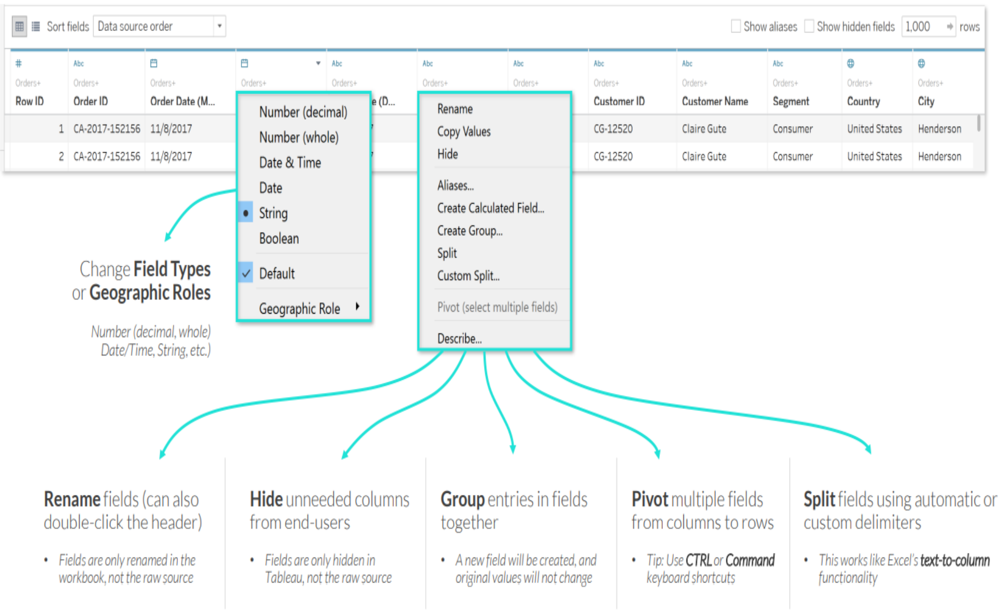

Sort fields
Data source order
Show aliases
Show hidden fields
1,000
rows
Row ID
Order ID
Order Date
Customer ID
Customer Name
Segment
Country
City
Number (decimal)
Number (whole)
Date & Time
Date
String
Boolean
Default
Geographic Role
Rename
Copy Values
Hide
Aliases...
Create Calculated Field...
Create Group...
Split
Custom Split...
Pivot (select multiple fields)
Describe...
Change Field Types or Geographic Roles
Number (decimal, whole) Date/Time, String, etc.)
Rename fields (can also double-click the header)
Fields are only renamed in the workbook, not the raw source
Hide unneeded columns from end-users
Fields are only hidden in Tableau, not the raw source
Group entries in fields together
A new field will be created, and original values will not change
Pivot multiple fields from columns to rows
Tip: Use CTRL or Command keyboard shortcuts
Split fields using automatic or custom delimiters
This works like Excel's text-to-column functionality

Fig. 4: Tableau Data Source Interface

- ***Use Data interpreter:*** The Data Interpreter dynamically cleans poorly formatted Excel/CSV files (extra rows, merged cells, etc.) with a single click. It is only available for Excel workbooks & CSVs.

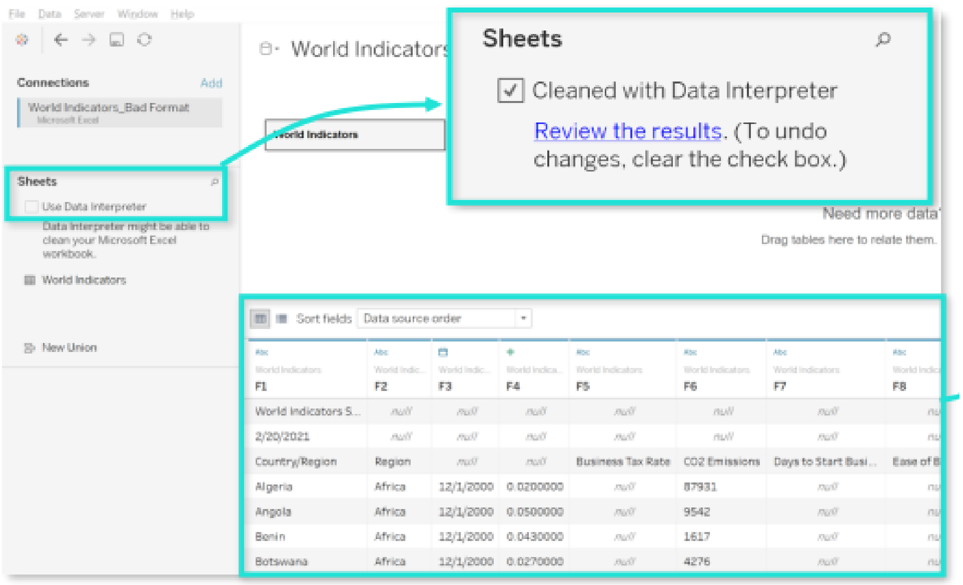

World Indicators
Sheets
Cleaned with Data Interpreter
Review the results. (To undo changes, clear the check box.)
Connections
Add
World Indicators_Bad Format
Use Data Interpreter
Data Interpreter might be able to clean your Microsoft Excel workbook.
New Union
Need more data?
Drag tables here to relate them.
Sort fields
Data source order

Fig. 5: Use Data Interpreter for Flat files

## V. COMBINING DATA IN THE DATA SOURCE INTERFACE

Tableau Desktop provides many options for combining and modeling your data source connections, including relationships, joins, unions, and blending.

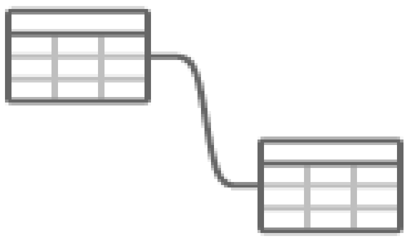


Fig. 6: Relationships

- **Relationships:** are a more flexible alternative to the existing joins and allows Tableau to query the tables independently. The goal of relationships is to provide an easy, fast, and more flexible alternative to using joins. It does not combine the tables together; instead, it concentrates on how two tables link to one another based on shared fields. The logical layers define this relationship.

  Relationships are suggested as your first method of merging your data since they make data preparation and analysis simpler and more comprehensible. Only use joins if you really need to.

Fig. 7: Data Blending

- **Data Blending:** allows you to query two different data sources and “blend” the results in Tableau. The primary driver of blending performance is the cardinality of the fields you’re using to link the data sources.

  Before integrating the results in memory in Tableau, blending queries the data from both data sources, at the level of those linked fields. The more unique values, the larger the query results, the more amount of memory used, and the larger amount of processing it requires.

  Blending should be used minimally and only on low-cardinality dimensions. Relate or join when you can, blend when you must. Blending is an important, but infrequent tool in your toolkit.

## VI. DATA SOURCE BEST PRACTICES

A few recommendations apply broadly to all data source and model types, including doing proper data prep, summarizing data, excluding unused data, and using Hyper extracts when possible.

### A. *Data speed is key:*

Your visualisations can only run as quickly as the underlying data sources, but a tried-and-true method is to keep the data size small. Less data to process and transmit when minimal data. Use only the information required at the grain level for the analysis on the worksheet.

### B. *Use Hyper extracts where possible*

Hyper is a powerful tool for following best practices. Leverage data source filters, aggregating to visible dimensions, and hide unused columns to shrink the size of your extracts and accelerate your dashboards. When your Tableau extract is embedded in your Tableau workbook (vs. being published to the Tableau data server), Tableau also does further optimizations making traditionally slow elements of your dashboard faster, such as Filters set to show only relevant values. With a focused extract, you’ll be well on your way to a fast-loading dashboard.

### C. *Optimize and materialize your calculation*

When a calculation is created and stored in the database, you’re reducing overhead in Tableau. This is especially the case for row-level calculations that do not need to be done by Tableau when a user requests the dashboard. Aggregate calculations are an example of something that often needs to be done specific to a user’s requests and should be calculated fields in Tableau.

### D. *Use initial SQL when possible*

One way to avoid a custom sql query being run multiple times in Tableau is to leverage Initial SQL if it’s available. As the name would indicate, Initial SQL is runs only when the workbook is opened first time and can be used to create a temporary table. That temporary table can then be used like any other table by Tableau and could significantly improve performance. One downside of this method, however, is that the temporary table remains the same during the duration of the user session, so updates to the underlying data will not be reflected in the temporary table until a new session has been opened.

### E. *Custom SQL*

It is the capability Tableau gives you to write your own queries in SQL to your underlying data source. This is a powerful tool that allows you to transform your data, perform more complicated logic, or leverage existing queries to build your dashboards. However, when it comes to production use cases, custom SQL has some performance trade-offs. Tableau leverages the custom SQL as a subquery. In short, Custom SQL creates long and confusing looking queries. For many databases the resulting complex queries will often achieve worse results, despite their best efforts to optimize.

## VII. FEW POINTS TO KEEP IN MIND WHILE DESIGNING A DASHBOARD

- If the data source is slow, it will definitely be slow in Tableau. Tableau extracts are a great solution to this problem. If you must use live data connections, it's best to consult your Database Administration or other IT resources to optimize your data source
- Don't worry about formatting. Everybody loves a beautiful chart but leave the fine-tuning until you know what charts you want to keep and share. Generating charts without worrying about design helps you explore the data rapidly. Tableau's iterative process makes it easy to continue to modify and change charts from one visual to another
- As you create lots of calculations and visualizations, it's easy to allow the default naming to remain and may even seem less efficient to take time to rename your assets. However, quickly giving meaningful names will help you in the long run and make your workbooks legible to you and others. Not only that, but you'll have an easier time leveraging your ad-hoc work in production builds if it follows any naming standards your organization may have
- When working on the prototype, you should spend time defining the core questions to address within your dashboard and finding the best ways to answer those questions.
- One important thing to note is that data volume may be the single most significant factor in performance, and it tends to grow over time as dashboards age. Extracts solve many problems. Changing from a live connection to a Tableau Hyper extract will make most workbooks run faster
- Limit the data. When you are exploring or authoring, you often want to look at all the data, but as you move into production, it's best to only publish with the data you need. Each unneeded row adds additional processing overhead that you just don't need. Consider adding a data source filter, dropping unnecessary columns, or aggregating details to the appropriate level
- Fixed dashboard size. Determine the screen size or sizes that you want to deliver to your end-users and build towards that. Automatic sizing is less efficient than exact dashboard size.

## VIII. CONCLUSION

Performance is a multifaceted problem and goes well beyond technical best practices. Designing the right dashboards, the right way will save you tons of time in troubleshooting. When you find yourself figuring out how to improve an existing workbook, remember there can be many reasons for slow performance. Collect data on performance, identify the most challenging areas and focus on them. You'll get better at identifying issues the more you work on it

Remember that Tableau is continuing to improve and add features all the time. Look to the always creative and helpful Tableau community for the latest ways to improve performance or ways to create new and exciting ways to see and understand your data.

## REFERENCES

[1.] *https://www.tableau.com/resources/reference-materials*
[2.] *https://www.tableau.com/support/help*
[3.] *https://www.tableau.com/learn/training/20224*
[4.] *https://community.tableau.com/s/*

**About the Author**
**Ayesha Umaima**
**Data Engineer Business Intelligence**

A Master of Technology in Computer Science Engineering and with overall 11 yrs. of experience, Ayesha is on her journey of finding meaningful insights in data. Throughout her career, driven by a thirst for knowledge she has worn many hats: Business Analyst, Tableau Developer, Agile Advocate, Cloud Computing Enthusiast, and a Data Engineer. Her projects are mostly centred around Data Science and Machine Learning. Apart from instigating positive changes for clients, she has great affinity to reading and writing books. She enjoys travelling and is particularly fond of Snow Mountains.