



Tableau Metadata Model

Author: Marc Reuter
Senior Director, Strategic Solutions, Tableau Software
March 2012

Most Business Intelligence platforms fall into one of two metadata camps: either model the entire enterprise as a first step, or model nothing. Legacy Business Intelligence solutions require highly skilled individuals with expert knowledge of the business, the data and the platform to model the metadata as part of an extensive exercise. The start-up cost and the change cost are high, but the enterprise gains control and structure. Many 'next generation' Business Intelligence Platforms claim that one of their innovations is to do away with metadata. This is a great solution for quickly getting started with simple data, but rapidly runs into problems when the platform usage spreads beyond a small team of experts and into mainstream, enterprise scale deployments. The start-up costs are low, but control and structure are sacrificed, resulting in increased Total Cost of Ownership.

Tableau has taken a hybrid approach that satisfies the requirement for rapid analysis with true ad hoc reporting and also fulfills the needs for broader metadata management. The metadata system in Tableau has been thoughtfully designed so that IT can add value by providing a rich metadata layer – yet business people are empowered to modify and extend it. Tableau has been so successful in making metadata seamless, approachable and transparent, that customers often believe Tableau does not have a metadata layer.

Tableau has developed products with the following philosophical points about metadata:

- Metadata in existing systems should be leveraged when beneficial
- Analysis must be possible without first undertaking a metadata modeling exercise
- Metadata is a useful abstraction, but should not be contraining
- Metadata are defaults that can be changed at runtime
- Business users don't need to understand metadata to be successful

With these points in mind, Tableau has created a simple, elegant and powerful metadata system known as the Data

Source. Two layers of abstraction and a run time model (VizQL Model) give the business user flexibility while allowing for enterprise-wide metadata. Again, a user can become an expert in Tableau and successfully deploy and operate a complete Business Intelligence platform with no knowledge of how Tableau handles metadata. However, individuals familiar with the concepts of metadata will have a better appreciation for Tableau's approach to metadata after reading this document.

The metadata model

Connection

The first layer of abstraction is the Connection. The Connection stores information about how to access the data and what data to make available to Tableau. The connection attributes for the database, the tables, views and columns to use, and any joins or custom SQL used to access the data. To support true ad hoc analysis, Tableau does not require that all fields, tables or joins be included in the initial connection. This allows users to get started with a simple connection and change it as needed to include more fields or tables. Note that the Connection is not a copy of the data or an import of the data. As such it is not useful to define filters, calculations, aggregations or any other data modeling at this layer. The connection defines the scope of data available to Tableau and this is the layer where database level security is applied.¹ Since it is an abstraction layer, changes can be made at any time.

For example, when moving a set of analysis or dashboards from a test database to a production database, a simply editing the Connection and entering a new server name is all that is required. Tableau takes care of the rest.

¹ For more information on security, see Tableau's Security whitepaper

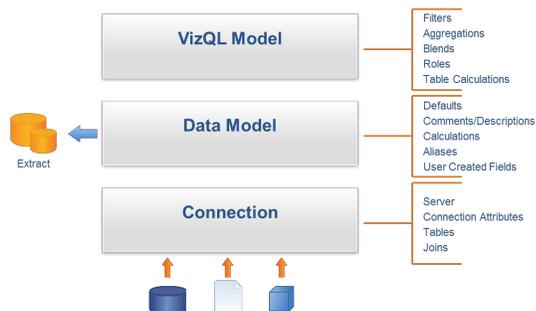


Figure 1: Tableau's metadata model

Tableau's metadata model has 2 layers of abstraction and a run-time model that allow simplicity, flexibility and reusability.

Data model

The second layer of abstraction, the Data Model, is the meat of the data source. Upon connection, Tableau automatically characterizes fields as Dimensions or Measures. When connecting to cubes, this is simply read from the robust metadata in the cube². When connecting to relational data, Tableau chooses whether a field is a dimension or measure based on intelligent heuristics that leverage the database's metadata.

The Data Model is the repository for additional metadata such as data types, roles, defaults, aliases and more. Additionally, it is the repository for user generated fields such as calculations, sets, groups³, etc. The Data Model is independent of the Connection. This means that it is both insensitive to changes to the Connection and its components are reusable in unrelated Connections. This abstraction enables the Data Model to automatically adapt to changes in the underlying data structure without requiring manual changes. This process itself is so elegant that this has eliminated the need for a whole set of features and tools which legacy BI vendors have often called as 'harvesting' or 'change management tools'. This time savings and risk mitigation are passed on to Tableau customers as a natural process of using Tableau.

Tableau further leverages this data model abstraction by allowing the Connection and Data Model to be defined organically. As authors forage for data, explore and

analyze the data, and refine and ask more questions, the Data Model can grow and adapt as needed. Therefore, there is no need to perform any metadata modeling exercises prior to getting started with your analysis or project.

For example, database tables can be added to the Connection at any time and the Data Model will automatically adapt by making the new fields immediately available for analysis. If a field is removed from the database or renamed, it can simply be remapped to a substitute column or automatically removed from all sheets where it is used.

For relational databases, the Data Model also has two connection states. It can use the Connection to dynamically access a live database, or it can be using an extract, or import, of the data in the Connection. This means that a business user can choose at any time whether to use a live connection to the data or to use a local cache of data extracted to Tableau's Fast Data Engine. It is simply a state that can be toggled at any time.

For example, a user has built a workbook using a live connection to a data warehouse. Recently, they have decided to take advantage of Tableau's Fast Data Engine. The user simply chooses the option to create the extract and the entire data source, and all the sheets that use it are now pointed to the extracted data. At any time, the user can choose to toggle back to the live data or even refresh their snapshot of the data.

It is important to note that the creation of the extract is a point and click operation that does not require scripting or SQL skills. The generation and lifecycle of the extract can be an IT administered function so it can be built once and used many times, or it can be a self-service function driven by the needs of business users.

The Connection and the Data Model together compose a Data Source in Tableau.

VizQL model

The final tier is the VizQL Model. The VizQL Model is an implementation unique to Tableau products and an area

² Cube based data sources are Microsoft Analysis Services, Microsoft PowerPivot, and Oracle Essbase

³ Not Available when using cube based data sources

“ Tableau has saved hours and hours of time pulling and manipulating data so we can focus more time on the activities that really matter - ”

“ analyzing the data and serving up discoveries and recommendations to move the business forward. ”

– US Auto Parts

where Tableau has innovated and secured several patents. Because the Data Model is an abstraction layer, the components or fields defined in the model are merely defaults. When the fields are used in an analysis by adding them to a sheet in Tableau, their role and attributes can be adjusted to suit the current needs of the author.

For example, a calculated measure in the Data Model is Employee Age:

```
DATEDIFF('year', [Date of Birth], TODAY())
```

When the user adds it to the sheet, it calculates the average Employee Age (average is the default aggregation for Employee Age).

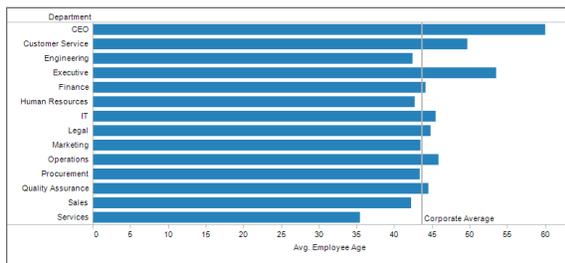


Figure 2: Average employee age by department

The default behavior of a field as defined in the Data Model.

However, the user might also want to find the distribution of age throughout the company. They can click to change the run-time role of Employee Age to be a Dimension. Now other measures, such as the Number of Employees can be aggregated against the Employee Age.

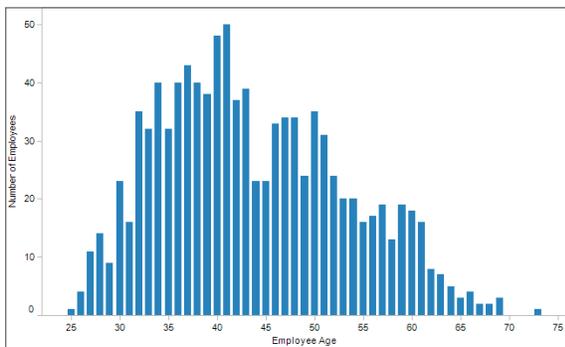


Figure 3: Number of employees by age

A user can easily change the run-time usage of any field in the Data Model to suit their needs.

Furthermore, if they would like to find the Maximum Employee Age or stack Rank the ages, or any other number of scenarios, the user can simply adjust the VizQL model without the need to explicitly model the new measures or dimensions as part of the data source. The result is both a) compactness of the data model because only one representation of a field needs to be modeled, and b) flexibility for analysis regardless of the initial definition of the fields.

The VizQL Model enables this flexibility by allowing users to adjust the role and aggregation of the fields at run time. This enables one user to define the base Connection and Data Model (Data Source) as a collection of fields without needing to know, plan or otherwise account for all the variations of analysis to be performed with the Data Source by other users. Additionally, other users do not need to contact IT or a developer when they want to use fields in unanticipated ways. All users have the flexibility to extend the Data Model to suit their immediate analysis needs. This unique innovation by Tableau is the secret sauce in Tableau's product line which is revolutionizing the data analytics and BI industry.

Another Tableau innovation solves the problem of secondary, or comparison, calculations. Examples of secondary calculations include moving averages, percent of total, and change from previous value. Many calculations, comparisons and transformations that users want to perform are hard to model in the context of a typical data model, but are easy to define in the VizQL Model. Table Calculations are a good example of these calculations and they greatly benefit from the flexibility and reuse afforded to them by the VizQL Model.

For example, in Figure 4, I have Sales plotted over 4 years. I can apply a Table Calculation to find the Growth Rate. If I change the view to show Sales by Month (Figure 5), the Growth Rate adapts to show month over month growth (Jan 2011 over Jan 2010).

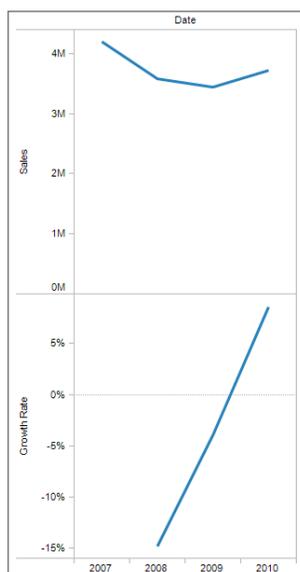


Figure 4: Sales over 4 years

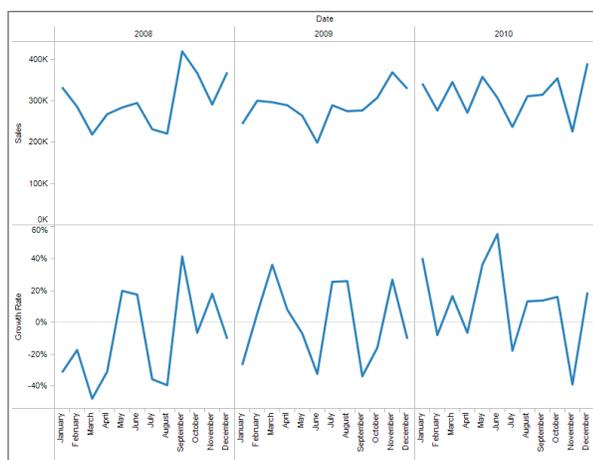


Figure 5: Sales by month

Table Calculations automatically adapt to changes in the visualization therefore reducing the number and complexity of fields needed in the Data Model.

Additional changes such as adding more dimensions (growth rate per product) or calculated month to month growth (Mar 2012 to Feb 2012) are simply adjusting the VizQL Model by adding the appropriate fields or settings. The user does not need to have all permutations of Growth Rate modeled in the data

model. They can simply apply a Growth Rate calculation to the current VizQL Model. As the model changes, the calculation automatically adapts.

Tableau further leverages the VizQL model by enabling users to combine data from different Data Sources into one hybrid model. This is called Data Blending. At run-time, Tableau determines how to merge the query results from two different Data Models into one combined result set. Because this blend happens after the Connection and Data Model have been applied, there are no changes needed to the underlying systems.

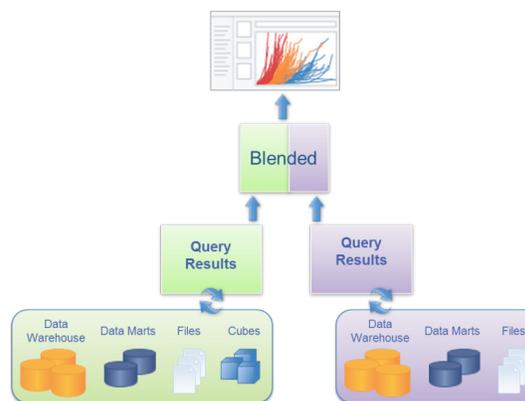


Figure 6: Data blending

Data Blending allows run-time combinations of Data Models without needing to predefine relationships.

Data Blending is another Tableau innovation that has gained tremendous adoption by business users. Data from any source can be combined on the fly without any modeling, scripting or extensive projects. Data does not have to be integrated into a single source of definition in order to perform a simple analysis or to generate a BI application. This capability saves both time and money by eliminating many needs for ETL processes and has enabled true self service analysis to BI users.

Metadata summary

A connection can be used in multiple Data Models, a Data Model can be used in multiple sheets and multiple views can be used in a dashboard. Furthermore, we can exploit these relationships in useful ways. For example, a dashboard can use views that leverage different Data Models and Connections, views can leverage fields from different Data Models, and a Connection can be leveraged by multiple Data Models.

All of this may sound daunting at first, but these concepts are successfully utilized by business users without deep understanding or even extensive planning. They are simply able to combine data that they already have access to into interactive visualizations that help them answer important questions.

Tableau metadata in action

Despite the richness of the metadata model and the power it offers, using Tableau is very simple. For example, consider a new user to Tableau that wants to understand some data in their corporate data warehouse. First, they connect to the database by entering the appropriate information:

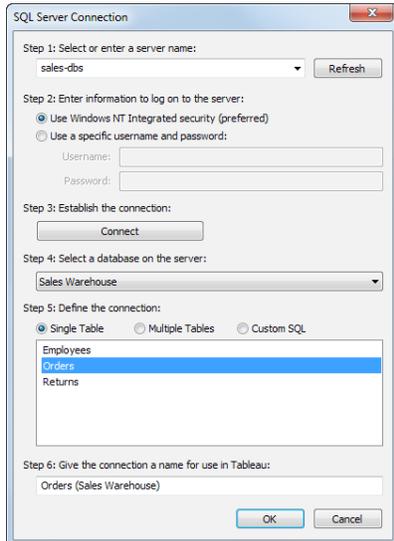


Figure 7: Connect to the database

Creating a Connection is as simple as choosing a server, database and tables.

Then the user double clicks on Sales and Order Date to create a result. That's it. Start to finish in a couple of seconds.

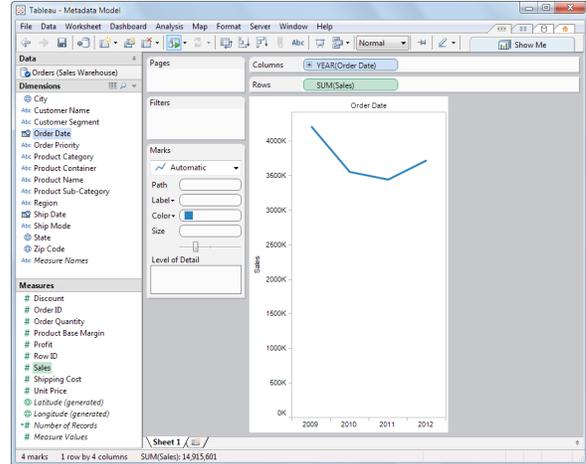


Figure 8: Quick results

Using a new Connection is as simple as double clicking on fields.

Tableau automatically categorized the fields and assigned default roles for them. When the user added the Sales measure, Tableau automatically applied an aggregation. When Order Date was added Tableau automatically built a date hierarchy and summarized the Sales by Year. Additionally, the patented Show Me! technology automatically chose a line chart as the best practice method for visualizing a trend over time.

Sharing metadata

Regardless of how automated the metadata discovery and definition phase is, the metadata represents the result of work. For this reason, Tableau has built in many sharing points into our products to support common workflows and allow customers to leverage their investments.

Published data sources

Remember, the Connection and the Data Model together, compose a Data Source in Tableau.

Publishing Data Sources to the Tableau Server allows an organization to share the Connection and Data Model, and therefore share metadata. A published Data Source can have relationships to many workbooksData Source. This means that changes to a master Data Source can be automatically propagated to workbooks that use the Data

Source. At any stage of creating a Data Source, the metadata can be published to the Tableau Server.

For example, some customers have robust data warehouses that already have friendly field names but contain many tables. IT or power users can publish a Data Source that has the connection information such as the tables and joins or the SQL statement needed to access the data. Other users of Tableau Professional can then use the published Data Source as a starting point for their analysis. They can add calculations, defaults, blend additional data and perform many other data modeling and VizQL model operations.

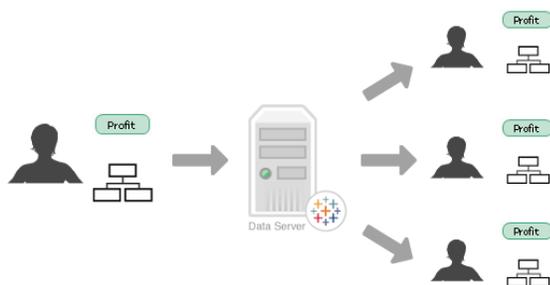


Figure 9: Publishing a data source

Publishing a Data Source provides a way for users to easily share metadata such as definitions and calculations.

Connecting to data sources

The Tableau Data Server takes Data Sources one step further by allowing desktop users to connect directly to a published Data Source⁴. This allows one user to define the connection attributes (tables, joins etc) as well as the base Data Model (calculations, hierarchies, groups etc). Users connecting to this Data Source cannot change the original definitions, but can still augment the Data Model with new fields, VizQL Model changes and Data Blending. The Data Server provides the best of both worlds – ease of metadata definition and management by IT with flexibility and extensibility by the users.

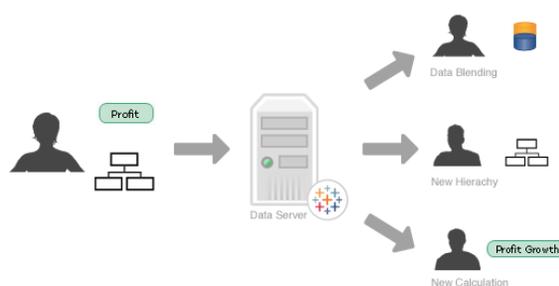


Figure 10: Modify shared metadata

Users can modify and extend the shared metadata—for example, to blend in new data sources or define new calculations.

For example, a user familiar with the tables in a database creates the Connection with several joins. They then add some specific calculations that they want to make standard and maybe some structure to the Data Model such as creating hierarchies. Once they have created their standard Data Model, they publish to the Tableau Server. Another user connects to this Data Source. This user cannot change the joins, nor the connection attributes. However, unlike other metadata approaches, they can add new calculations and otherwise extend the Data Model. For example, the second user could extend the analysis with new geocoding data or a quota spreadsheet he had on his desktop. They could then visualize percent of plan achieved within each store location. All without needing to involve IT or the original Data Source author. When the original user makes changes to the Data Source and republishes it, the second user will receive all the changes yet still retain all of their additions to the Data Source.

⁴ Not supported for cube based data sources

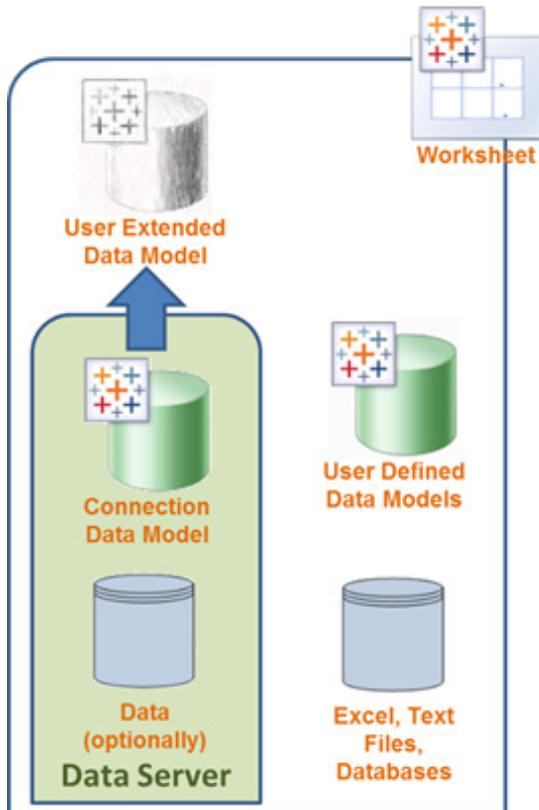


Figure 11: Extending the data model

One user can define the connection and establish standard metadata, another user can extend the connection with more metadata and even more data.

In addition to expressly publishing a Data Source, Tableau provides a variety of means for re-using metadata. Along with publishing to the Tableau Server, Data Sources can be exported and shared as files so they can be used in many different workbooks. Fields in the Data Model, such as calculations and parameters, can be copied and pasted into other Data Models. And even elements in the VizQL Model, from individual fields up to entire sheets can be reused. When suitable, Tableau allows many VizQL Model changes to be persisted into the Data Model, for example Grouping operations and Table Calculations. Because of the layers of abstraction, these abilities to share and re-use the metadata have very few pre-requisites. This means that it is easy to leverage your investment in metadata.

Maintaining metadata

Organizations must be able to maintain their business intelligence independent of the source data in order to minimize disruption and provide greater flexibility. In other words, metadata that is already in use by published Views and Dashboards needs to be adaptable to changes in the data environment, like an updated calculation. Tableau Server automatically manages the metadata to adapt to a number of changes so that end users can continue to consume analysis without disruption. A vast array of additional maintenance can be performed from within the Tableau Server itself, or quickly managed from within Tableau Desktop.

Connections

Tableau Server takes special advantage of the abstraction layers built into Tableau's metadata model. When workbooks are published to the Tableau Server, the Connection information is automatically extracted from the data sources. This separation provides two main benefits. First, the Connections can be shared across workbooks, views and users. This means that there is less overhead in maintaining separate database connections and caches on the Tableau Server and any temp tables or other session scoped information can be leveraged on the database. Second, the Connections can be modified independently of the published workbook. This means that maintenance items such as changing the credentials used to access the database or the database host name can be performed anytime.

For example, IT has provisioned a new database server that has the latest version of Oracle. The schema is the same as the old database, but the server name has changed. Because of Tableau's levels of abstraction, this change is simple to manage. The Oracle server name can be updated on the Tableau Server and all published workbooks will be automatically rerouted to the new database server. There is no need to generate an ancestry report of all objects utilizing a certain server or to manage a project around migrating all of those objects to the new server. A simple update to the Connections solves the problem. This approach can save considerable time and minimize deployment risk.

Data model

The data model can be changed in two ways: explicit operations by users in Tableau Desktop and changes to the underlying connection.

Unlike Connections, explicit changes to the Data Model must be made in Tableau Desktop. The reason is that the scope of changes can be much broader, such as redefining a calculation, and there are many more types of changes that might need to be made. Therefore, a rich, dynamic and responsive editing environment is required. When changes are made, Tableau automatically manages the change across all sheets in the Workbook. For example, changing the alias for 'Calif.' to 'California' automatically results in all sheets, legends, filters etc. displaying 'California'. Likewise with changes to any field definition or default.

For changes to the underlying Connection, Tableau Desktop will either automatically adapt, or allow the author to mitigate the change themselves. For example, with additive changes such as new tables and new columns, no user intervention is required. The fields are automatically available for analysis when the data source is opened or refreshed. For subtractive changes such as missing columns, the user will have the opportunity to replace all occurrences of the original column with a new one. This capability can also be invoked to simply reroute usage of one column to a new one. For example, Order Amount was used in the workbook to determine our sales. But now we need to report on Discounted Order Amount via a calculation. The calculation can be created and Order Amount automatically replaced throughout the workbook.

In addition to Tableau's automatic adaptability to changes to the base systems, users can also explicitly replace the current Data Source with another Data Source.

For example, a user has created their own analysis on their desktop using Tableau Professional and an Access database. However, the access database is a month old, and the current data is in an Oracle data warehouse. The user can simply make a new connection to the Oracle data, and replace their existing connection to Access. The Data Model from both connections is automatically merged and all views are re-run with Oracle data.

VizQL Model

When delivering high value reporting and analytics via a Business Intelligence platform, it is important that the run-time environment can adapt to changes in the database in a robust manner. Because of the layers of abstraction in Tableau metadata, the VizQL model is insensitive to database changes such as renaming a member of a column, adding more members to a column and adding more columns. However, Tableau Server is sensitive to renaming or removing columns that are used in a View. If a required column cannot be found in the database, Tableau Server will remove it from the view at run time. This does not modify the original data source, or View. For example, a View (Figure 11) shows Sales broken down by Region and State. However, during the nightly data warehouse load, the table containing the State column didn't load. The VizQL model will temporarily remove the column from the View and the result will be Sales broken down by Region (Figure 12). Once the data warehouse returns to normal with a complete load of the data, the Tableau view will revert to its published state.

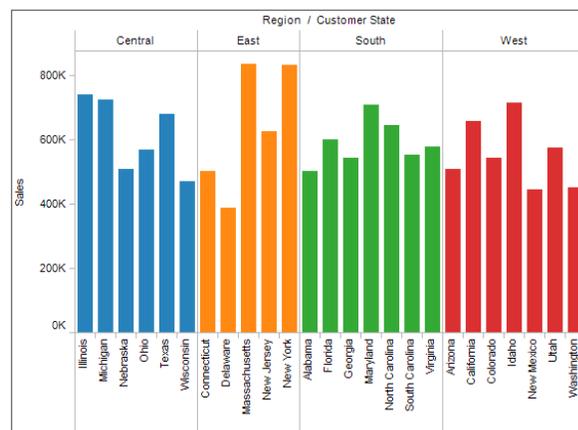


Figure 12: Sales by region and state

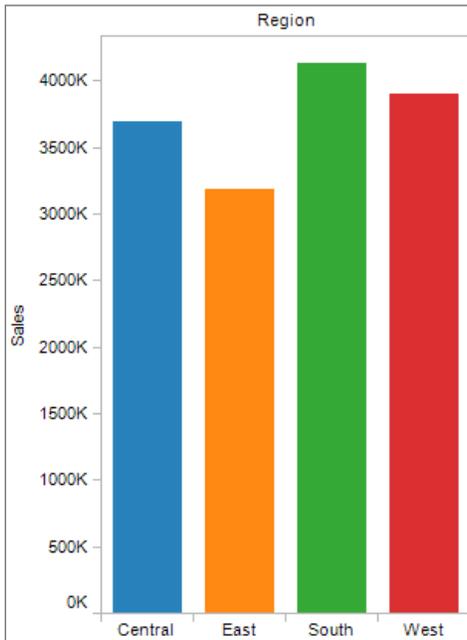


Figure 13: Sales by region and state

Tableau Server automatically adapts to schema changes. Here it temporarily removes missing fields from the view.

For more serious issues, the Tableau Server will present information about the problem.

Summary

Tableau's metadata model is simple, flexible and powerful. It fulfills the Tableau philosophy by enabling people with the right skills to be able to appropriately leverage and impact the model at the right time. People with database expertise can establish, define and maintain connections across the deployment. Authors and power users can extend the data model as needed to satisfy their requirements. End users have the flexibility to use the fields in a variety of roles to answer their specific questions. At each stage of the analytic, development or production cycle, users have the appropriate ability to use or augment the metadata. This comprehensive, user driven approach to metadata, results in broader adoption by business users, with less intervention and maintenance required by IT. Overall, this elegantly designed approach and the numerous breakthrough

innovations which are incorporated in this approach are the key reasons why customers have found tremendous success with Tableau.

About Tableau

Tableau Software helps people see and understand data. Ranked by Gartner in 2011 as the world's fastest growing business intelligence company, Tableau helps individuals quickly and easily analyze, visualize and share information. With more than 6,500 customers worldwide of all sizes and across industries, Tableau is used by individuals throughout an organization, in an office and on-the-go. See the impact Tableau can have on your data by downloading the free trial at www.tableausoftware.com/trial.