



---

## Tableau Metadata Model Version 6.1

**AUTHOR:**

Marc Rueter, Sr. Director Technology  
Consulting & Strategy

**DATE:**

July 2011

Most Business Intelligence platforms fall into one of two metadata camps: either model the entire enterprise as a first step, or model nothing. Traditional 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 and into enterprise scale deployments. The start-up costs are low, but control and structure are sacrificed.

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 be possible without a metadata modeling exercise
- Metadata is a useful abstraction, but should not be constraining
- Metadata are defaults, they 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. It is a 3 tier system with 2 layers of abstraction and a run time model (VizQL Model). 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.

## Tableau Metadata in Action

Despite the richness of the metadata model and the power it offers, using Tableau is very simple. An example is the best way to see this. Consider a new user to Tableau that wants to understand some data in their corporate data warehouse.

First, the user connects to the database by entering the appropriate information: the server name, their credentials and the database and tables to access.

The screenshot shows the 'SQL Server Connection' dialog box with the following steps and fields:

- Step 1: Select or enter a server name:** A dropdown menu shows 'sales-dbs' and a 'Refresh' button is to its right.
- Step 2: Enter information to log on to the server:** Two radio buttons are present: 'Use Windows NT Integrated security (preferred)' (which is selected) and 'Use a specific username and password:'. Below the second option are 'Username:' and 'Password:' text boxes.
- Step 3: Establish the connection:** A 'Connect' button.
- Step 4: Select a database on the server:** A dropdown menu showing 'Sales Warehouse'.
- Step 5: Define the connection:** Three radio buttons: 'Single Table' (selected), 'Multiple Tables', and 'Custom SQL'. Below them is a list box containing 'Employees', 'Orders' (highlighted in blue), and 'Returns'.
- Step 6: Give the connection a name for use in Tableau:** A text box containing 'Orders (Sales Warehouse)'.

At the bottom right are 'OK' and 'Cancel' buttons.

The user then instantly sees a list of all available Dimensions and Measures (also known as fields or variables). By double clicking on Sales and Order data, he creates a result. That's it. Start to finish in a couple of seconds.

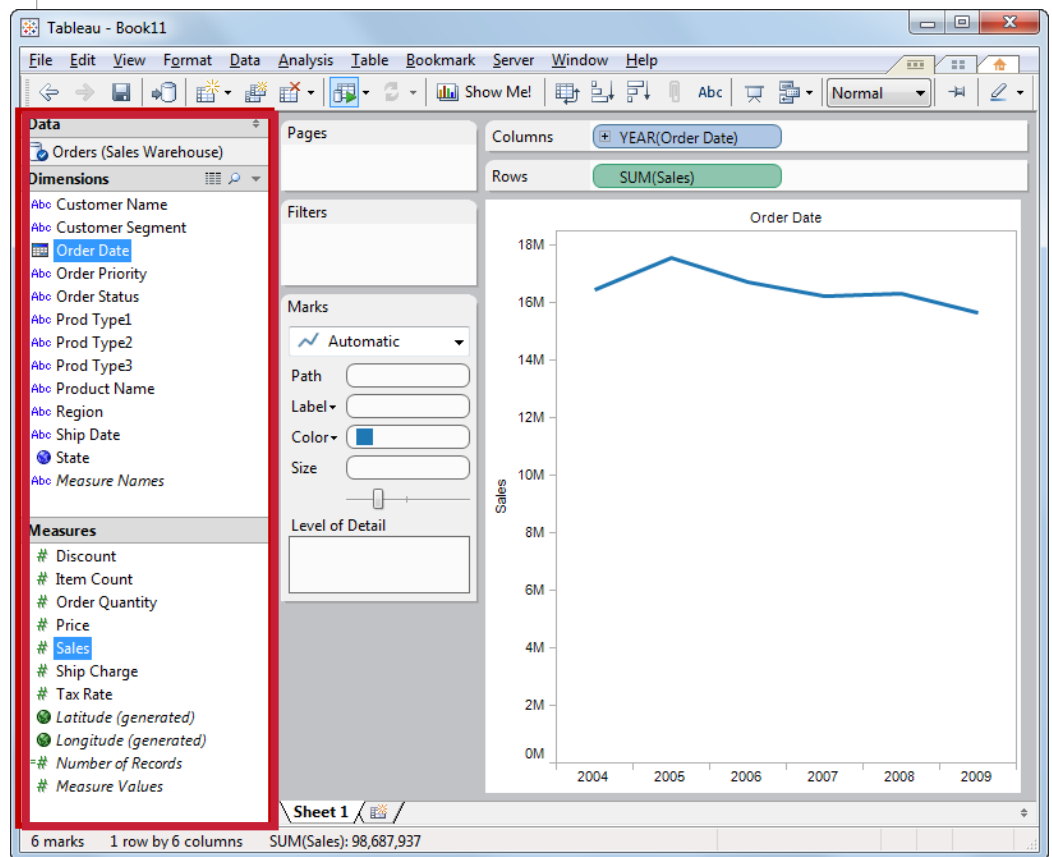
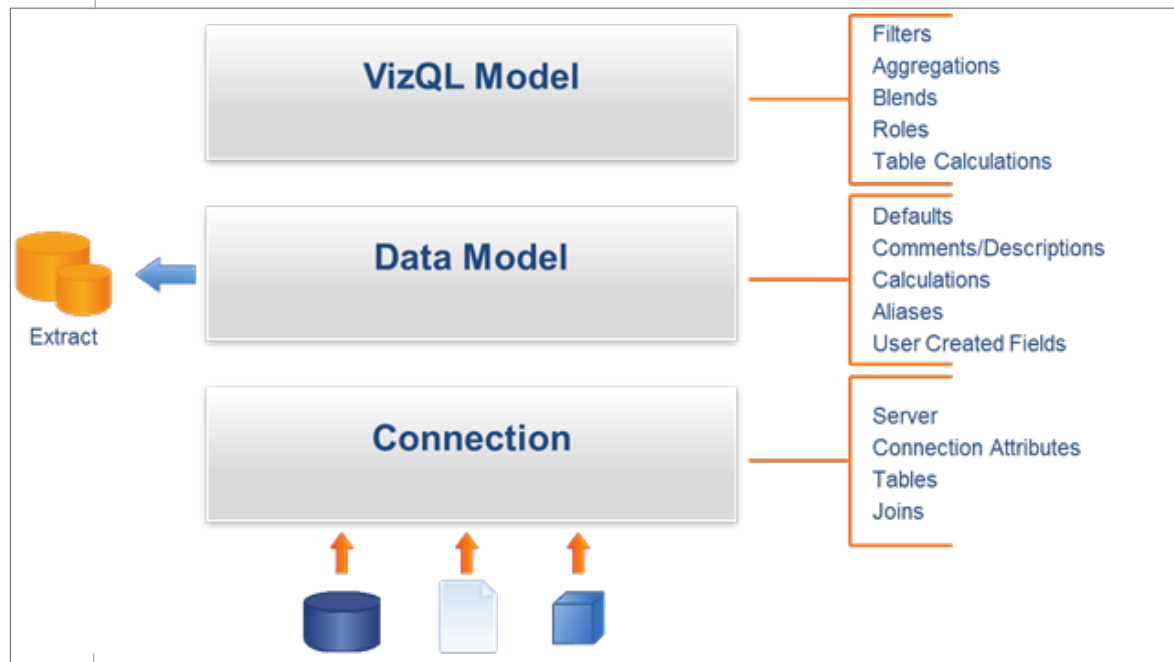


Tableau automatically categorized the fields and assigned default roles for them. When the user added the Sales measure, Tableau automatically applied an aggregation. When Order Date was added Tableau automatically built a date hierarchy and summarized the Sales by Year. Additionally, the patented Show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

## 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. 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. Since it is an abstraction layer, changes can be made at any time.

### Data Model

The second layer of abstraction, the Data Model, is the meat of the data source. Tableau automatically characterizes fields as Dimensions or Measures. When connecting to cubes, this is simply read from the robust metadata in the cube<sup>1</sup>. When connecting to relational data, Tableau chooses whether a field is a dimension or measure based on intelligent heuristics that leverages the

<sup>1</sup> Cube-based data sources are Microsoft Analysis Services, Microsoft PowerPivot and Oracle Essbase

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, groups<sup>2</sup>, 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 Connection 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, for example by renaming fields, adding definitions or creating new calculations. 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 an extract and the entire data source, and all the sheets that use it are now pointed to the extracted data that sits in-memory on the user's machine. 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

<sup>2</sup> Not available when using cube-based data sources

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 where Tableau has innovated and secured several patents. VizQL is Tableau's proprietary technology that lets users quickly build visualizations in Tableau simply by dragging and dropping.

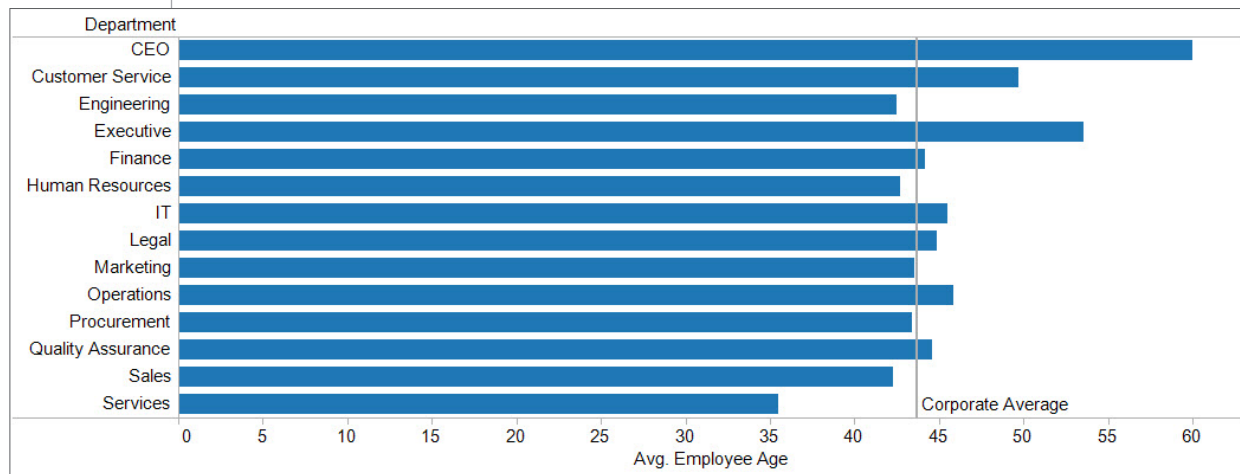
Because the Data Model is an abstraction layer, the components or fields defined in the model are merely defaults. When the fields are used in an analysis by adding them to a sheet in Tableau, their role and attributes can be adjusted to suit the current needs of the author.

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.

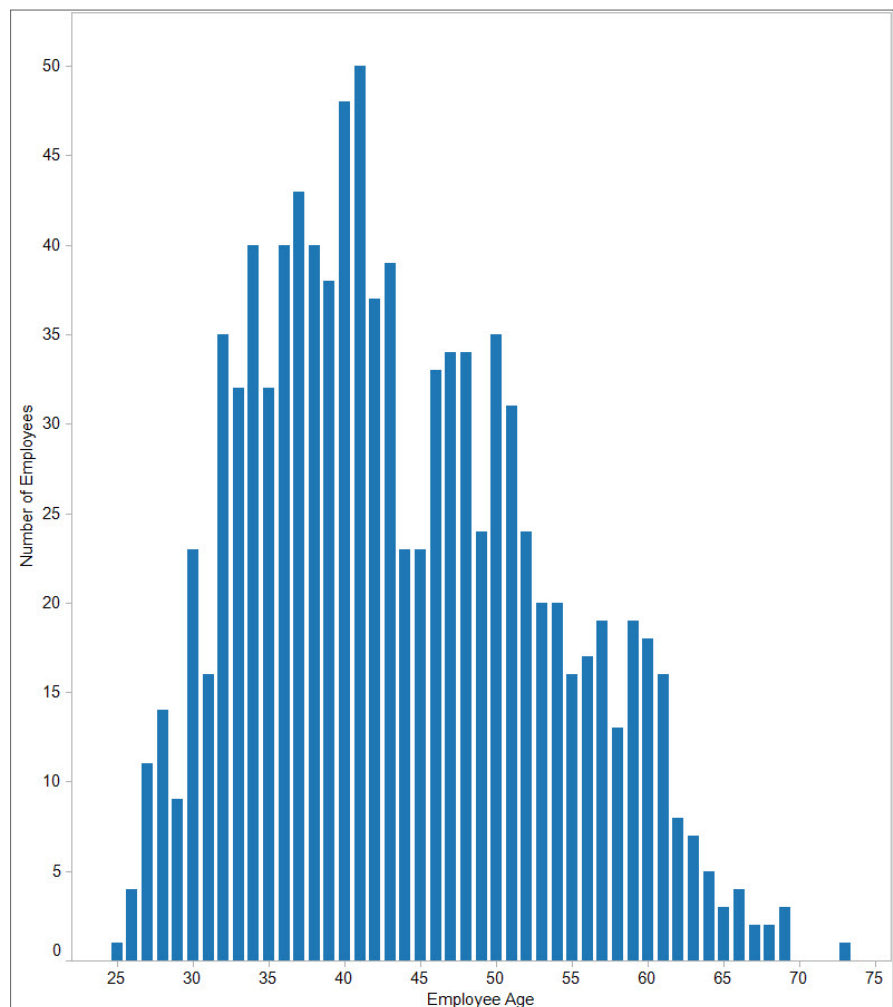
### **Average Employee Age by Department**



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. Furthermore, if they would like to find the Maximum Employee

Age or stack rank the ages, or any other number of scenarios, they 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 compactness of the data model because only one representation of a field needs to be modeled and flexibility for analysis regardless of the initial definition of the fields.

### Number of Employees by Age



The VizQL Model enables this flexibility by allowing users to adjust the role and aggregation of the fields at run time. This unique innovation by Tableau is the secret sauce in Tableau's product line which is revolutionizing the data analytics and BI industry.

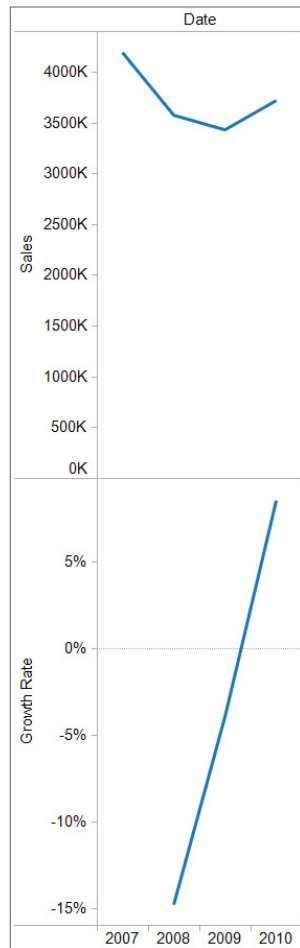
Another Tableau innovation solves the problem of secondary, or comparison, calculations. Examples of secondary calculations include moving averages, percent of total, and change from



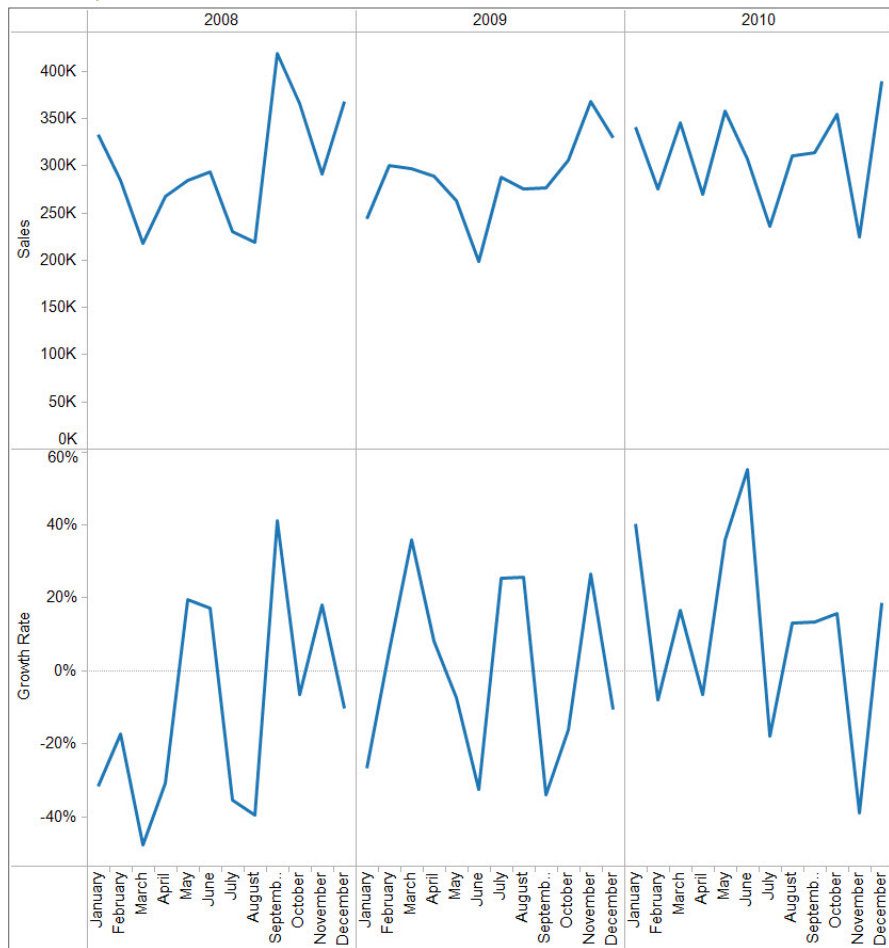
previous value. Many calculations, comparisons and transformations that users want to perform are hard to model in the context of a typical data model, but are easy to define in the VizQL Model. Table Calculations are a good example of these calculations and they greatly benefit from the flexibility and reuse afforded to them by the VizQL Model.

For example, in the image below (left), Sales is plotted over 4 years. A Table Calculation can be applied to find the Growth Rate. If the view is changed to show Sales by Month (below right), the Growth Rate adapts to show month over month growth (Jan 11 over Jan 10).

Annual Growth

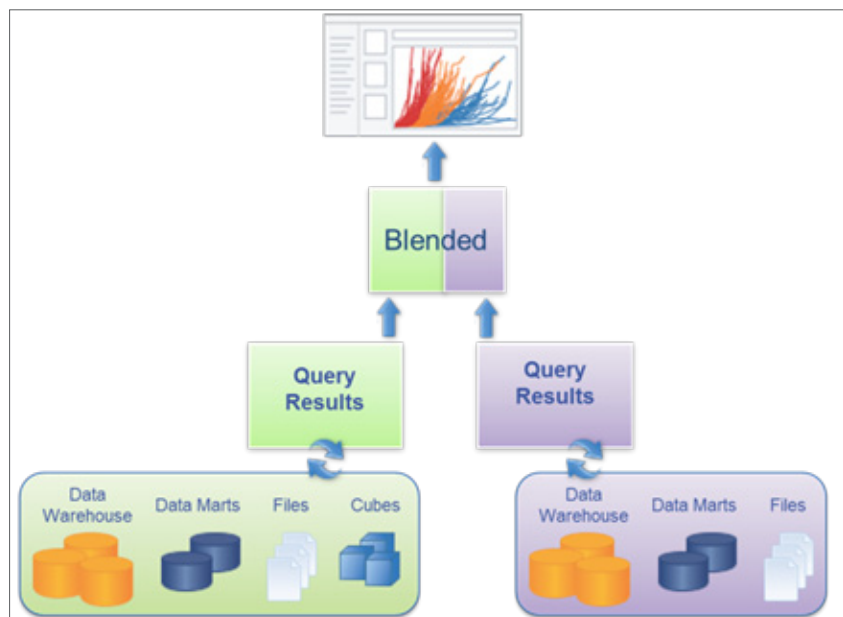


Monthly Growth




Additional changes such as adding more dimensions (growth rate per product) or calculated month to month growth (Mar 11 to Feb 11) 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.



However, since it is often useful to create calculations that combine fields from different data models, Tableau allows Data Models to have dependency on other Data Models.

For example, imagine a case where there are two Data Sources, one using a database containing Sales by Sales Rep, the other an Excel file containing Employee compensation information. The problem is that the database has Sales Rep in two fields, First Name and Last Name; while the Excel file has one field, Last, First. Because the blend occurs after all transformations are applied in the Data Model, a simple Tableau calculation to get the fields in the same format will result in a combined view of the data.



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.

## **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 relationships to be created between workbooks and data sources. 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.

### ***Re-Using Metadata***

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***

An Enterprise class Business Intelligence Platform must be maintainable independently of the source data in order to minimize disruption and provide greater sustainability and 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. 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 2 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.

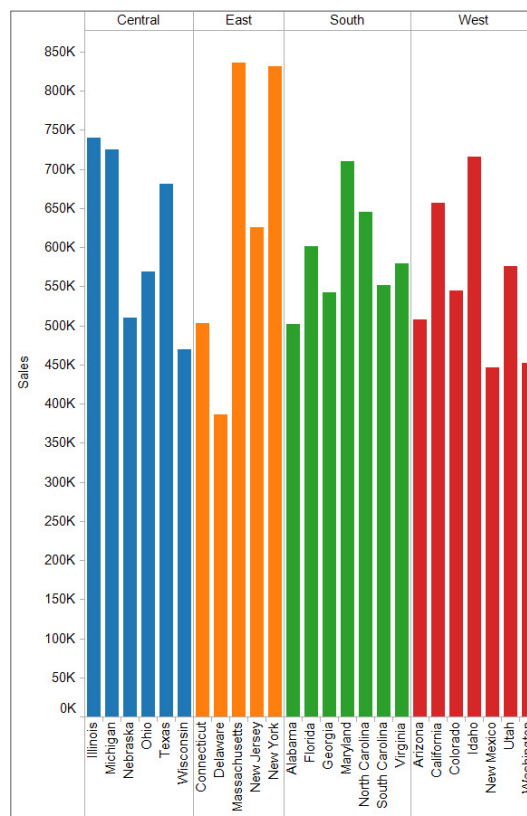
Therefore, a rich, dynamic and responsive editing environment is required. When changes are made, Tableau automatically manages the change across all sheets in the Workbook. For example, changing the alias for 'Calif.' to 'California' automatically results in all sheets, legends, filters etc. displaying '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.

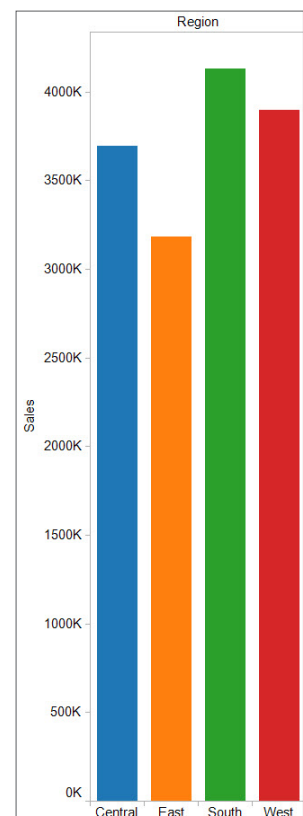
### VizQL Model

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 shows Sales broken down by Region and State (below left). 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 (below right). Once the data warehouse returns to normal with a complete load of the data, the Tableau view will revert to its published state.

Sales by Region and State



Sales by Region



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

## Summary

Tableau's metadata model is simple, flexible and functional. 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.

© Copyright Tableau Software, Inc. 2011

Tableau Software  
837 North 34th Street, Suite 400  
Seattle, WA 98103 U.S.A.

Produced in the United States of America  
July 2011  
All Rights Reserved

© 2011 Tableau Software