About Perspectives

Perspectives are deployable subsets of a model. They are meant to make it easier for analysts to query only the model data that is relevant to their purposes or responsibilities. Rather than provide analysts with the entire model, you can make specific dimensions, metrics, and other objects invisible to them.

For example, suppose that you worked as a data modeler for a retailer. Three business analysts need access to company data to create reports: one report related to inventory, one related to marketing, and one related to orders. Rather than create three separate models, you can create one model and three perspectives: one for each analyst.

So, you create your model and then create the perspectives described in the two following tables.

Table 1. Fact-table columns that are visible in the different perspectives that are based on your model.

Fact-Table Columns	Perspective: Inventory	Perspective: Marketing	Perspective: Orders
Clicks		✓	
Quantity	✓		✓
Sales Amount			✓
Spend		✓	
Tax			✓
Views	✓		
Wholesale Price	✓		

Table 2. Dimensions that are visible in the different perspectives that are based on your model.

Dimensions	Perspective: Inventory	Perspective: Marketing	Perspective: Orders
Color	✓		
Customer		✓	✓
Date	✓	✓	✓
Product	✓	✓	✓
Size	✓		

You deploy the catalog, then grant each analyst permission to query only the perspective they need to build a report. You can control a perspective's permissions from the deployed model's Permissions tab. When the business analysts connect to the model, they can select from the perspectives they have access to.

Note: Perspectives control the visibility of objects in a model, but are **not** meant to be used as a security measure. They are not secure against users querying objects that are not visible. If a user who has permission to query a perspective knows the names of other objects in the model that are hidden from the perspective, they can write an MDX query that runs against the perspective but references those other objects.

If an analyst requests a drill-through that contains objects that are not part of the perspective that the analyst is using, the AtScale engine removes them from the result set.

As the analysts query the perspectives, the AtScale engine creates demand-defined aggregates for the base model. Demand-defined aggregates that are created based on one or more queries on a perspective can also be used to satisfy queries against the perspective's underlying model. Likewise, demand-defined aggregates that are created based on one or more queries on a model can be used to satisfy queries against perspectives on that model.

Moreover, when you deployed your catalog, the AtScale engine created prediction-defined aggregates based on the model, as usual. These aggregates can also be used by the engine to satisfy queries against the model's perspectives. In fact, it is also possible for any user-defined aggregates defined for the model to satisfy queries against perspectives.

When you monitor aggregate tables on the Aggregates page, you notice that the entries always refer to the perspective's base model, not the perspective itself. When you monitor queries on the Queries page, only inbound queries refer to the perspective. On these pages, it isn't possible to search for queries, aggregate definitions, or aggregate instances by perspective.

Later, you edit one of the model's perspectives. However, your changes will not available to users until the following happen:

1. You redeploy the corresponding catalog.
2. Users currently connected to the perspective close the workbook, sheets, or other workspaces in which they are querying the perspective and then reconnect to it.

   For example, Tableau users cannot just download a new `.tds` file and then refresh their connection; they must close their workbook and then reopen it.

Additional Information

- Working with Perspectives
- Deploying a Catalog
- Configuring Permissions on Deployed Models
- Modeling Row Security Objects