About Aggregates

AtScale incorporates the data-warehousing concept of aggregate tables. Such tables most often contain measures from one or more fact datasets and include aggregated values for these measures. (There are dimension-only aggregate tables.) The aggregation of the data is at the level of one or more dimensional attributes or, if no dimensional attributes are included, the aggregated data is a total of the values for the included measures.

With aggregate tables, query engines can improve the performance of queries that request aggregated data such as Sales Amounts by Month or Number of Orders by Country.

Without aggregate tables, query engines need to scan regular tables to return aggregated data, and those tables could have very high row counts. The engines then have to perform the calculations required to aggregate data. Both scanning and calculating are actions that are resource-intensive and time-consuming. Query engines that have no aggregate tables to use must perform these actions for each query on a measure.

Challenges With Implementing And Using Aggregate Tables

Two significant drawbacks to aggregate tables, however, are the time and effort involved in designing, testing, deploying, periodically refreshing the data in them (called rebuilding), and monitoring them. It can take data modelers weeks to design aggregate tables based on requirements gathered from stakeholders or based on query analysis performed on existing OLAP cubes. The tables then need to be deployed and loaded on test systems where modelers can iteratively test and revise them. After all of that comes the meticulous task of deploying the tables on production systems.

Deployed aggregate tables need constant monitoring of their utilization rates, and the response times for the queries that use them also need to be monitored. Monitoring can lead to the redesign of aggregate tables and the creation of new ones. These tasks can involve more than one team (e.g. a team of data modelers and a team extract, transform, and load the data for the test aggregate tables), and therefore require significant time and effort as tables are iteratively designed, tested, and deployed.

Finally, aggregate tables need careful rebuilding at intervals, so that they include the newest data and the most recent updates to an OLAP cube. Rebuilding incorrectly can lead to query results that are inconsistent from one build to the next.

How AtScale Eliminates These Drawbacks

AtScale lets you take advantage of the power of aggregate tables without spending all of that time and effort normally required.

AtScale's adaptive cache technology is able to analyze the queries against your cubes and cube metadata, determine the optimal aggregate tables for improving performance of those queries and possible future queries, and define and create aggregate tables automatically on the fly. Moreover, rather than simply building an aggregate table once and then rebuilding it over time as new data arrives or is updated in your cluster, AtScale continuously optimizes its collection of aggregate tables for changing query workloads.

Because the AtScale engine does this, you don't have to ask yourself questions such as "How do I know when I have 'enough' aggregate tables?" or "How do I know that I have the right aggregate tables?". The engine is always at work creating a sufficient number of aggregate tables that are optimized for your query workloads. All of this optimization happens after a cube is published, when the cube is in production and being queried by business intelligence applications.

This means that you don't have to make an up-front expenditure of time and effort to design, test, and deploy aggregate tables before your cubes start providing business value. After a data modeler designs and publishes a cube in the AtScale Design Center, client applications can start querying it right away. AtScale's adaptive cache continuously learns about the query workloads on that cube and defines and redefines aggregate tables to satisfy them.

- Aggregate Definitions and Aggregate Instances AtScale separates the concept aggregate table into two separate concepts: the definition of an aggregate table and an actual instance (or materialization) of that definition. To summarize, each aggregate table in AtScale is an instance of a definition.
- Types of Aggregate Tables in AtScale There are two types of aggregate table that the AtScale engine can use: system-defined aggregate tables and user-defined aggregate tables. System-defined tables are further divided into demand-defined and prediction-defined. User-defined tables are further divided into manual and hinted.
- When the AtScale Engine Creates Demand-Defined Aggregates Demand-defined aggregates are a type of system-defined aggregate that the AtScale engine creates according to various criteria, such as the history of queries against a cube and statistics about the use of existing demand-defined aggregates.
- When to Define Your Own Aggregate Tables AtScale recommends that you rely on the AtScale engine to generate aggregate tables for you. However, if you require aggregate tables that contain any of the types of dimensional attributes or measures listed below, you must define the aggregate tables manually.
- How Aggregate Tables Are Populated With Data After an aggregate table is defined, an instance of it needs to be created. An instance is an actual table that contains the aggregated data. The process of creating an instance of an aggregate table and populating it is referred to as a build of the table.
- Partitioned System-Defined Aggregate Tables You can specify a level of a dimensional hierarchy to use as a partition key for partitioning instances of system-defined aggregate tables.
- Partitioned User-Defined Aggregates When you define your own aggregate or when you edit an aggregate that you have defined, you can specify a dimensional attribute to use as a partitioning key. If you do so, instances of the aggregate will be partitioned, with one partition created for each key value. Partitioning an aggregate can result in faster query performance.
- About Incremental Rebuilds This type of rebuild of instances of aggregate tables appends new rows and updates

existing rows that fall within a period of time that you can specify.

- Aggregates for Fact Datasets that Use Joins If a fact dataset uses inner joins to join to one more dimensions, the AtScale engine can generate demand-defined aggregates that include data from those dimensions. You can also define your own aggregate-table definitions that use inner joins.
- About Hinted Aggregate Tables A hinted aggregate table includes every join key, measure, and degenerate dimension that is included in a query dataset.
- Life Cycle of Aggregate Tables The following diagram summarizes the stages in the life cycle of an aggregate table.
- Flowchart of the Actions Taken for Each Query Each query that requests aggregated data causes the AtScale engine to find out whether an aggregate table exists that will satisfy the query. There are two possible outcomes of this search: an appropriate aggregate table does not exist and an appropriate aggregate table does exist. This flowchart shows the actions that the engine takes for both outcomes.