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.

Consider that for an aggregate instance to be built there should be an aggregate definition for it. An aggregate instance can be built and used only after the corresponding System-Defined or User-Defined aggregate table definition is created. If you want to see what definitions you have in AtScale you can check the **Definitions** tab in the Aggregate monitor.

For more information about how aggregates are defined, built, and used, see About Aggregates, Aggregate Definitions and Aggregate Instances, Life Cycle of Aggregate Tables, and Defining Aggregates Yourself.

Updating Aggregate Tables

Aggregate-table instances become outdated as a cluster ingests new data and existing data is updated. There are three actions that the engine can take to bring aggregate tables up-to-date.

- Supersede the current instance of an aggregate table with an entirely new instance. The new instance includes more recent data and uses the existing definition for the aggregate table. This is called a full rebuild of an aggregate table.

  The previous instance stays in place as a backup in case the building of the new instance fails. Queries continue to use the previous instance until the new instance is ready.

  The AtScale engine rebuilds aggregate instances by querying the fact tables and dimensions that the definitions of those aggregates are based on. However, if the engine setting AGGREGATES.CREATE.BUILDFROMEXISTING.ENABLED is set to **True** and the engine decides that an aggregate instance A can be rebuilt from data that is in the rebuilt instance B of a different aggregate table, then the engine runs a query on B to populate A. Such a query generally runs much faster than a query on fact tables and dimensions. When rebuilds are triggered for all of the aggregate tables for a cube, the AtScale engine determines which aggregates could be built from other aggregates. The latter are given higher priority in the order of the rebuilds, so that the former can be rebuilt from them.

  

  Instances of incremental aggregates are not rebuilt from the data that is in instances of other aggregate tables.

- Stop using the current system definition for an aggregate table and create no new instance of the table. Instead, create an instance of a newly defined aggregate table that supersedes the previous definition. The new instance includes the most recent data.

- Supersede the current instance of an aggregate table by a) duplicating that instance, and then b) updating the rows in the duplicate and appending new rows to it. This is called incremental rebuild of an aggregate table. For more details about this type or rebuild, see About Incremental Rebuilds.

By default, all of the aggregate tables for a cube are refreshed with full rebuilds. In the AtScale Design Center, you can set a flag on a cube's fact dataset that causes all rebuilds to be incremental rebuilds. If your cube has more than one fact dataset, the aggregate tables will be built incrementally only for those fact datasets on which the flag is set.

Aggregate-table instances are not built or rebuilt individually. Rather, all aggregate-table instances for a cube are built or rebuilt together in a batch-build process. A process builds or rebuilds one aggregate at a time in succession. You can use engine settings to influence the order in which aggregate instances are processed.

How Batch-Build Processes For Cubes Are Started

Many companies have ETL (extract, transform, load) jobs that they routinely run to cleanse raw data and load it into a data warehouse. Before it ends, a job can call one of AtScale's REST APIs to trigger a build process. Alternatively, if a job is configured to write to a log file when it finishes, you can have AtScale monitor a path in a distributed file system (in a path in a bucket, if your data warehouse is Google BigQuery; in HDFS, if your data warehouse is Hadoop) for this signal that the job is complete. AtScale can then start a build process.

If no routine ETL job runs on your cluster or you do not want to wait for a routine job to run, you can click a button in the AtScale Design Center to start a build process.

Automatic Restarts Of Interrupted Batch-Build Processes

As an aggregate instance for a cube is built for the first time or rebuilt, the build process assigns it a status. If a build process is interrupted by the AtScale engine stopping, restarting the engine restarts the build process automatically. The status of an aggregate instance at the time of the interruption determines what the process does with that instance when the process restarts.

- If the status of an instance is **New**, the instance has not yet been created and the process creates it.
- If the status of an instance is **In Progress**, the process restarts work on the instance.
- If the status of an instance is **Done**, the process ignores the instance.
- If the status of an instance is **Failed**, the process attempts again to create the instance, if the value of the engine setting AGGREGATES.BATCH.RETRY.MAXATTEMPTSPERAGGREGATE has not yet been reached for that instance.

If the build process completes successfully, the status of all aggregate instances that the process created is set to **Active**.

More Information

- Rebuilding Aggregates Using the REST API
- Triggering Aggregate Rebuilds
- Rebuilding Aggregates Manually
- Settings for Enabling and Prioritizing Builds of Aggregate Instances