

# 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.

Joins can be extremely expensive when tables have a large number of rows. For this reason, the AtScale engine will build aggregates for fact datasets that join to one or more dimensions only if the compression ratio (the number of rows in the fact dataset to the number of rows in a proposed aggregate) is higher than the compression ratio for an aggregate that does not use a join.

For example, suppose that a fact dataset has a billion rows and contains an attribute named Customer Key. This key is used joins with the Customer dimension, which has one million rows. A query against the fact dataset requests Order Quantity by Customer Name. An aggregate table based on this query would contain one million rows versus the billion rows that are in the fact dataset. Therefore, because the compression is 1000x, the AtScale engine would create an aggregate.

Now, suppose that the Customer dimension contains an attribute named Gender Key that is used for joins with the secondary dimension Gender. The Gender dimension contains only two rows, and so there are only two different values in the Gender Key column in the Customer dimension.

A query requests order quantity by gender. Because it is possible for the AtScale engine to join the fact dataset to the Customer dimension when generating an aggregate, the Gender Key attribute is available for the aggregate to group Order Quantity by. The resulting aggregate definition would generate aggregate instances of only two rows (one for each gender key), and the compression ratio would be much higher than the compression ratio for an aggregate that did not use the join. In this case, it is very likely that the AtScale engine would create an aggregate-table definition that uses the join.

For the AtScale engine to be able to define aggregates that use joins, the engine setting `AGGREGATES.CREATE.JOINS.ENABLED` must be set to the value `True`, which is the default value. For a description of this setting and other settings that are related to when and how the engine defines aggregates that use joins, see [Settings for Specifying Whether and How to Use Joins](#).

## Rebuilding Aggregate Instances That Contain Data From Inner Joins

Whenever a new aggregate instance is created, the AtScale engine creates it with a full build. Every subsequent rebuild of an instance can be either a full rebuild or (if the `incremental` property is set in the fact dataset's [underlying SML](#)) an incremental build. Guidelines for managing aggregates that contain data from inner joins differs according to whether the aggregates are rebuilt fully or rebuilt incrementally.

## Full rebuilds

A full rebuild creates a new copy of an instance with the result set from a query on the fact dataset and joined dimensions. Therefore, if the data in joined dimensions changes, a full rebuild includes the changed data.

## Incremental rebuilds

You must be sure that the joined dimensions rarely, if ever, change outside of the grace period. When the AtScale engine performs an incremental rebuild, it does not search through existing rows in the aggregate to replace old values or include newly appended values. As more changes are made to a dimension outside of the grace period, the aggregate instance becomes less accurate. If changes are made outside of the grace period, you should trigger a full rebuild of the aggregate table.

However, if changes are made to joined dimensions within the grace period, there is no need for a full rebuild of the aggregate.