

Other Aggregate Settings

You can view and modify the following cube-level aggregate settings.

Aggregate Partition For BigQuery

- ▲ `aggregate.partition.bigquery.range.end` : The upper bound for integer value partitioning on BigQuery. The default value is 10000.
- ▲ `aggregate.partition.bigquery.range.interval` : The interval for integer value partitioning on BigQuery. The default value is 10.
- ▲ `aggregate.partition.bigquery.range.start` : The lower bound for integer value partitioning on BigQuery. The default value is 0.

Aggregate Batch

- ▲ `aggregates.batch.buildFromExisting.enabled` : Build aggs from aggs for aggregate batch rebuilds. The default value is true.
- ▲ `aggregates.batch.buildFromExisting.threshold` : Use the optimized algorithm only when the number of aggregates in a batch are less than this threshold. The default value is 100.
- ▲ `aggregates.batch.cube.attributes.threshold` : Use the optimized algorithm only when the number of attributes in a cube are less than this threshold. The default value is 250.
- ▲ `aggregates.batch.cubeDataRequests.parallelism` : Number of cube data requests that can be made in parallel. The default value is 2.
- ▲ `aggregates.batch.retry.maxAttemptsPerAggregate` : The maximum number of reattempts to build a single aggregate during a single batch build. This number cannot exceed the value of `aggregates.batch.retry.maxattemptsperbatch`. The default value is 3.
- ▲ `aggregates.batch.retry.maxAttemptsPerBatch` : The maximum number of reattempts to build aggregates during a single batch build. The default value is 5.
- ▲ `aggregates.batch.reuseOrderingGraph.enabled` : Use the existing ordering graph if available for rebuilding aggregates. The default value is true.
- ▲ `aggregates.batch.max.failures` : Maximum number of failures for a batch build, before the whole batch fails. The default value is 0.

Setting a value higher than 0 enables failure tolerance for aggregate batch builds. For example, assume you have set this number to 5, and 3 instances in a batch build fail. The rest of the instances would be built successfully. The 3 failed aggregate instances would be moved to a list. At the end of aggregate batch, the system would come back to the list of failed aggregates and would retry to build them, as configured in the retry settings (see above).

Large Aggregate Tables

- ▶ `aggregates.largeTableOptimization.distributionKeyColumn.minimumCardinality` : The minimum cardinality required for the highest cardinality dimensional attribute to be used as a table distribution or clustering key. The default value is 30.
- ▶ `aggregates.largeTableOptimization.enabled` : Enables consideration of optimizations for large aggregate tables, such as column-based clustering or distribution. The default value is False.
- ▶ `aggregates.largeTableOptimization.minimumEstimatedRows` : The minimum estimated row count required for the aggregate system to consider applying optimizations, such as clustering or distribution. The default value is 100000.

Aggregate Maintenance

- ▶ `aggregates.maintenance.deactivateUnused.enabled` : Whether to deactivate system-defined aggregates that have been unused for the required number of days (as indicated by `aggregates.maintenance.zeroUtilizationTTL`). The default value is True.
- ▶ `aggregates.maintenance.zeroUtilizationTTL` : The minimum time with no usages that must pass before an aggregate will be deactivated. The default value is 45.days.

The format is `{number}{separator}{time}`. `{number}` is a positive number. `{separator}` is either blank, space, or a dot. `{time}` is one of the following: ms, milli, millisecond, s, sec, seconds, min, minutes, h, hours, d, days. For example: `1000.ms` or `2 seconds`.

Aggregate Creation Schedules

- ▶ `aggregates.new.build.scheduled` : When true, new aggregate instance builds are postponed until the next scheduled batch build for the cube. If false (default), new aggregate instances may be queued for building at any time.
- ▶ `aggregates.uda.build.scheduled` : When false (default), new user-defined aggregate build instances are queued for building following the project publish event. If true, UDA builds are postponed until the next scheduled batch build for the cube.
- ▶ `aggregates.predictionDefined.build.scheduled` : When false (default), new prediction-defined aggregate build instances are queued for building following the project publish event. If true, PDA builds are postponed until the next scheduled batch build for the cube.

For more information, see [Scheduler for Aggregate Creation](#).

Other Aggregate Settings

- ▶ `aggregates.dataWarehouseCacheTableRequests.enabled` : Whether to cache data warehouse table requests. The default value is True.
- ▶ `aggregates.dataWarehouseCacheTableRequests.maximumRowCount` : The maximum number of rows for a data warehouse table request to be cacheable, if caching is enabled. The default value is 50000.

- ▶ `aggregates.dimensional.allowJoinsToSecondaries.enabled` : Whether or not to allow joins to secondary attributes to be filtered out for completeness testing. The default value is True.
- ▶ `aggregates.dimensional.build` : Set to True (default) to allow the engine to create aggregates that contain dimensional attributes only. Such aggregates can be useful in Tableau for queries against fact tables that contain degenerate dimensions.
- ▶ `aggregates.slowBuild.cutoff` : The duration cutoff for a completed aggregate build query to emit a `SlowAggEvent`. The default value is 4 seconds.
- ▶ `aggregates.smallTableReplication.enabled` : Whether to consider applying data replication to system-defined aggregate tables. The default value is False.
- ▶ `aggregates.smallTableReplication.factBasedAggs.enabled` : Whether to allow aggregate table replication on fact-based aggregates. The default value is False.
- ▶ `aggregates.smallTableReplication.maximumEstimatedRows` : The maximum estimated number of rows that an aggregate table can have to be considered for replication. The default value is 1000.
- ▶ `aggregates.systemGenerated.activeInstance.extraAllowance` : The maximum number of additional system-defined aggregates that are temporarily permitted to be active when a cube has reached its retention limit. The default value is 2.
- ▶ `aggregates.systemGenerated.activeInstance.retentionLimit` : The maximum number of active system-defined aggregates to retain per cube. This is also known as the Retention Limit. The default value is 20.
- ▶ `aggregates.systemGenerated.withDistinctCountMeasure.retentionPercentage` : The maximum number of active aggregates with distinct count measures per cube is calculated as this percentage of the value of `aggregates.systemGenerated.activeInstance.retentionLimit` for each cube. This pool of system-defined aggregate tables is sized and counted separately from the regular system-defined aggregate table pool. The default value is 40.
- ▶ `aggregates.withDistinctCounts.widening.enabled` : Allows widening of distinct count aggregates with other measures, including distinct counts. This setting requires `aggregates.create.widening.enabled` to be set to True, for details see [Settings that are Related to Narrowing and Widening](#).

More Information

- ▶ [Cube Level Configuration Settings](#)
- ▶ [Settings for Large Table Optimization](#)