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.

Supported Incremental Indicator Types

AtScale only supported Long and Integer Incremental Indicator column types in versions older than 2020.3.0. This excluded certain Data Warehouses, such as Snowflake, from the ability to set Incremental Indicators and trigger Incremental Aggregate rebuilds. AtScale version 2020.3.0 and later introduces support for fixed-point Decimal(38,0 - Snowflake only), Timestamp and DateTime Incremental Indicator types. The addition of fixed-point Decimal support enables Incremental Aggregate Builds if using the Snowflake Data Warehouse. The addition of general support for Timestamp and DateTime Incremental Indicator types increases the flexibility of AtScale to work with more schema designs without modification. Secondarily, for Google BigQuery users, the benefit of Timestamp Incremental Indicator support is that it allows users to lower their costs by leveraging Timestamp-based partition pruning.

The full list of supported Incremental Indicator Types as of AtScale version 2020.3.0 is:

- Long
- Integer
- Decimal (38,0) (Snowflake only) - Added in AtScale version 2020.3.0
- Timestamp - Added in AtScale version 2020.3.0
- DateTime - Added in AtScale version 2020.3.0

Note: If using an IBM Db2 Data Warehouse, data loss can occur when reading database output from a Timestamp(12) column. The fully supported precision for TIMESTAMP is up to TIMESTAMP(9) or nanoseconds. When reading a TIMESTAMP(12) value, the last 3 digits are truncated.

Incremental Indicators

To specify that you want aggregate tables for a cube to be rebuilt incrementally, you simply specify a column in the fact dataset to use as an **Incremental Indicator**. This column must have values that increase monotonically, such as a numeric UNIX timestamp showing seconds since epoch, or a Timestamp/DateTime. The values in this column enable the AtScale engine both to append rows to an aggregate table and update rows during an incremental rebuild.

Incremental aggregates are database Views that consolidate one or more incrementally-built tables. Periodically, these tables are consolidated to maintain optimal query performance. You can specify how many tables to allow before consolidation takes place. The distribution of data in the underlying aggregate tables is based on the incremental indicator values. Segmenting of tables is based on ranges of values in the incremental indicator. Incremental indicator values are used as an internal tracking mechanism and are not made for querying in the incrementally-built aggregate.

Appending Rows

During an incremental aggregate build, AtScale retrieves rows from the fact dataset that have an incremental indicator value greater than the last value recorded in the incremental aggregate.

Example

Suppose that the last incremental indicator key value used in the incremental aggregate is 1475191168000 (the UNIX timestamp in seconds since epoch for Thu, 29 Sep 2016 23:19:28 GMT). When an incremental rebuild of the aggregate begins, the engine will look in the fact dataset for rows with an incremental indicator value greater than this value, perform the necessary data aggregation, and append rows to the aggregate.

Note: When the incremental indicator column type is as a timestamp, the Grace Period field is always interpreted as Seconds.

Updating Rows

You also specify a grace period when you select a fact dataset's column to use as an `incremental indicator`. When the AtScale engine starts an incremental build, the grace period determines how far back in time the engine will look for updates to rows in the fact dataset, based on the units that are in the indicator. The engine then updates rows in the aggregate table where the incremental indicator matches those of rows in the fact dataset.

Example

As in the previous example, suppose that 1475191168000 is the last incremental indicator key value in an aggregate table. The grace period is 86400 seconds (one 24-hour period). The engine looks back in the fact dataset to the first row where the indicator is 86400 seconds earlier than the moment that the incremental build began. The engine also does this in the aggregate table.

Next, the engine compares the data in the fact set with the aggregated data in the aggregate table, moving forward in time in the data until the engine reaches the point in time when the incremental build began. As it compares data, the engine updates the aggregated data when it finds changes in the fact dataset.

At the start of an incremental rebuild, the AtScale engine runs a query against an aggregate to find the latest incremental indicator to figure out new data that has been added since the last update. The query uses a WHERE filter to only scan data since the last update which enables table pruning against partitioned fact datasets.

Moreover, if more than one aggregate configured for incremental rebuilds is based on the same fact datasets, the engine runs the query to see what new data has been added since the last update and uses the result for the remaining aggregates. For example, suppose that six aggregates configured for incremental rebuilds are based on the table `factinternetsales`. The query is run on the first of these six aggregates and the result is used for all six. This process saves time and resources versus the alternative of running the query for each aggregate.

Incremental Rebuilds Of Aggregates That Use Joins

You can create aggregate tables that use joins to one or more dimensional datasets, and such aggregate tables can be rebuilt incrementally. See Aggregates for Fact Datasets that Use Joins.

More Information

Setting Properties to Allow Incremental Rebuilds of Aggregates