Calculated Column FAQs

This section answers the most frequently asked questions (FAQs) about creating and using calculated columns in an AtScale cube.

What Can I Do With A Calculated Column?

Calculated columns are useful for deriving values from the original dataset columns (such as calculating someone's age based on their birth date), doing data cleansing and pre-processing (such as combining column values or substituting one value for another), or for computing new data values based on a number of input variables (such as calculating a profit margin value based on revenue and costs).

Can I Refer To A Calculated Column In Another Calculated Column Formula?

No. Calculated column formulas can operate only on the original dataset columns.

What Formulas Can I Use To Define A Calculated Column?

The formulas that describe a calculated column must be valid SQL expressions that:

- Would be valid as an item in the column list of a SELECT statement.
- Use SQL functions and expression syntax supported by the engine you are using to query your data warehouse.
- Do not require an input or output type that AtScale doesn't support. (For example, SQL functions that operate on map or array data types.)

Can I Use Aggregate Functions In A Calculated Column Formula?

No. Although it is valid SQL, it is not the correct way to add aggregations to an AtScale cube. Instead, define a measure or a calculated member using the aggregate calculations that AtScale supports.

What SQL Operators And Functions Can I Use In A Calculated Column?

If you are using Google BigQuery as your data warehouse, see Functions & Operators in the BigQuery documentation.

If you are using a Hadoop cluster as your data warehouse, the answer depends on the underlying SQL-on-Hadoop engine that you are using. Here are links to the documentation for the SQL-on-Hadoop engines that AtScale supports:

Engine	Link to SQL Docs
	SQL Functions:

Hive	https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-Built-inFunctions Operators: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-Built-inOperators
Impala	SQL Functions: http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v2-0-x/topics/impala_functions.html Operators: http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v2-0-x/topics/impala_operators.html
SparkSQL	Spark SQL is designed to be compatible with the Hive Metastore, SerDes and UDFs. Currently Spark SQL is based on Hive 0.12.0 and 0.13.1. https://spark.apache.org/docs/latest/sql-programming-guide.html#compatibility-with-apache-hive

Is The AS Statement Supported?

Yes, except when using InterSystems IRIS as a data warehouse.

The AS (alias) statement is not supported by InterSystems IRIS, and you should not use it in the formula for a calculated column. Using it with such a data warehouse would lead to failed queries with "Unsupported CAST target specified" message.