Table and Column Statistics

The SQL engines you use do a certain amount of optimizing of the queries on their own—they look for the best way to proceed with your query, when possible. When the query uses joins, the optimizers can do a better job when they have *table statistics* and *column statistics*. For the table as a whole, these statistics include the number of rows, the number of files used to store the data, and the total size of the data. The column statistics includes the *approximate* number of distinct values and the maximum and average sizes of the values (*not* the maximum or average value, but rather the size used in storage). The optimizers use this information when deciding how to perform the join tasks. Statistics also help your system prevent issues due to memory usage and resource limitations.

These statistics are not automatically calculated—you have to manually trigger it using a SQL command (see below). Once statistics are computed, both Hive and Impala can use them, though if you compute them in Hive, you need to refresh Impala's metadata cache. If you make any changes to the table, such as adding or deleting data, you'll need to recompute the statistics.

Both Hive and Impala can use the statistics, even when calculated by the other machine. However, when you have both Impala and Hive available, Cloudera recommends using Impala's COMPUTE STATS command to calculate and view the statistics. The method for Hive (see below) is a bit more difficult to use. If you *do* use Hive, you must refresh Impala's metadata cache for the table if you want Impala to use the statistics.

Statistics in Impala

Impala's syntax for calculating statistics for a table (including statistics for all columns) is COMPUTE STATS *dbname.tablename*; If the table is in the active database, you can omit *dbname.* from the command.

To see the statistics in Impala, run SHOW TABLE STATS *dbname.tablename*; or SHOW COLUMN STATS *dbname.tablename*;

Note: If the statistics have not yet been computed, #Rows for the table shows -1. The #Nulls statistics for each column will always be -1; old versions of Impala would calculate this statistic, but it is not used for optimization, so newer versions skip it.

Statistics in Hive

Hive's syntax for calculating statistics for a table is ANALYZE TABLE *dbname.tablename* COMPUTE STATISTICS; If the table is in the active database, you can omit *dbname.* from the command. To calculate column statistics, add FOR COLUMNS at the end of the command.

To see the table statistics in Hive, run DESCRIBE FORMATTED *dbname.tablename*; The Table Parameters section will include numFIles, numRows, rawDataSize, and totalSize. To see the statistics for a column, include the column name at the end: DESCRIBE FORMATTED *dbname.tablename columnname*; You can only display column statistics one column at a time.

Try It!

1. The planes table in the fly database has not had any statistics computed for it. In Impala, run SHOW TABLE STATS fly.planes; Notice that #Rows says -1. Then run SHOW COLUMN STATS for the same table. Most of the statistics there are also -1. (The INT types have max and average size of 4, because all integer types have a fixed size.)
2. Use COMPUTE STATS fly.planes; to compute the table and column statistics. Check the table and column statistics for these, and note that there is information where there wasn't before. (The #Null column will still be all -1 though, as noted above.)
3. Compare the number of rows from the table statistics to the #Distinct Values statistics for the tailnum column. Most likely it appears that there are more distinct values in that column than there are rows in the table! This isn't unusual—remember, #Distinct Values is an *approximation* of the number of distinct values in the column, not an actual count.