Importance of Statistics in SQL Server

Notes extracted from Source: https://sqlespresso.com/2019/06/26/

1. What Are Statistics?

In SQL Server, **statistics** are metadata objects that describe the **distribution of data values** in one or more columns of a table or indexed view.

The **Query Optimizer** uses statistics to estimate how many rows a query will return — this process is called **cardinality estimation**.

Each statistics object consists of three parts:

1. **Header** – shows **when stats were last updated** and how many rows were sampled.
2. **Density Vector** – describes **uniqueness** of column values or combinations of columns.
3. **Histogram** – shows **data distribution and frequency** for the first key column.
    - The histogram helps estimate how many rows have values in a certain range and how many equal certain values.

You can view these details using:

```
DBCC SHOW_STATISTICS (table_name, stats_name);
```


2. Why Statistics Matter

Statistics are critical because the Query Optimizer depends on them to create efficient execution plans:

✓ **Accurate Row Estimates**

- Based on statistics, the optimizer estimates how many rows a query will return.
- Better estimates help it choose fast operations (like index seeks) over slow ones (like full scans).

✓ **Optimal Query Plans**

- Good statistics improve execution plan quality.
- Poor or stale stats can lead the optimizer to pick inefficient plans, harming performance.

3. What Happens When Statistics Are Stale

As data changes through INSERT, UPDATE, or DELETE, the distribution of values changes.

If the statistics are not updated to reflect these changes, SQL Server may make bad estimates.

✓ Over-estimated row counts → suboptimal plans

✓ Under-estimated row counts → poor join/scan choices

✓ Longer query times and higher resource usage

4. How Statistics Are Created & Maintained

Automatic Creation

- SQL Server automatically creates stats for indexed columns and for columns in predicates

AUTO_CREATE_STATISTICS = ON

- Auto-created stats usually start with `_WA_` in their name.

Automatic Updates

- AUTO_UPDATE_STATISTICS ensures stats get updated when SQL Server determines they are stale.

- However, the default threshold (20% change) may be too high for large tables, causing stats to remain stale for a long time.

Manual Updates

You can manually update stats using:

```
UPDATE STATISTICS table_name stats_name WITH FULLSCAN;
```

Or schedule regular updates as part of maintenance jobs.

https://www.databasejournal.com/ms-sql/importance-of-statistics-and-how-it-works-in-sql-server-part-2/

5. Checking & Interpreting Statistics

Using DBCC SHOW_STATISTICS, you can examine:

- **Header**: last updated, sample size
- **Density**: uniqueness and selectivity
- **Histogram**: value distribution and frequency

This helps you understand how well statistics reflect the actual data.

6. Best Practices for Statistics

✓ **Keep AUTO_UPDATE_STATISTICS ON** so SQL Server can maintain stats automatically.

✓ **Schedule manual updates** on large tables where automatic updates may lag.

✓ **Monitor Estimated vs Actual Rows** in execution plans — large discrepancies indicate stale stats.

✓ Consider **Trace Flag 2371** (default in SQL Server 2016+) to lower the stats update threshold and adapt to data changes more quickly.

7. Summary

- Statistics represent how data is distributed in a table.
- They are essential for the Query Optimizer to choose the best execution plan.
  https://medium.com/%40rahulgosavi.94/understanding-statistics-in-sql-server-a0647e8d9833
- Stale or missing statistics lead to bad estimates and slower queries.
- Regular maintenance and understanding what stats store helps keep SQL Server performance optimal.