## 12 Steps to SQL Server Query Tuning

**REVIEW FILTER AND** Explain plans **EXAMINE THE** ACCESS PREDICATES Examine cost, use estimated **EXECUTION PLAN** Know how query row count and statistics is interpreted (e.g., time START bind variables) Know when Actual plans predicate is applied Get actual plan use actual stats (should be earlier Find the (use DBMS\_XPLAN, at runtime rather than later) DISPLAY\_CURSOR) expensive operators Review data types to avoid implicit **EVALUATE** Review table conversions **EXISTING** definitions INDEXES Get sizes and ANALYZE 3 row counts **GATHER TABLE** Index **COLUMNS IN** INFORMATION definition WHERE CLAUSE Look for select \* or If it's a view, get scalar functions (some the underlying table Get current are less optimal) 5 definitions statistics Know the selectivity Know if histograms or Foreign key constraints baselines are being can help the optimizer create used; get sample bind Know what objects Know the selectivity and better execution plans **REVIEW EXISTING** variable values exist (avoid duplicating cardinality of the columns **KEYS AND** efforts later on) (is there data skew?) CONSTRAINTS What is primary **RUN THE QUERY AND** key definition? **RECORD BASELINE** Know the selectivity SARG-able METRICS (need index?) TUNE Gather average **THE OUERY** Look for execution times high-cost steps **RE-RUN** Nested loop/ merge/hash joins **THE QUERY** Focus on logical I/Os (number of logical reads) Wait events TIP Focus on Seeks vs scans—which most expensive 9 Record the is more expensive in this operations first Out-of-date time, logical I/O scenario? statistics can impact and wait events, performance **RF-RUN** and compare with initial THE QUERY Consider using a functionbaseline **CONSIDER ADJUSTING INDEXES** based index to exclude rows **ENGINEER OUT** in very large tables **THE STUPID** Make small 10 changes Reduce logical I/O Abuse of wildcards (\*)pulling back too many Nested views that go rows Cursors and row-by-row across database links processing Look for frequently encountered performance inhibitors Scalar functions

solarwinds

Find and fix query performance faster with SolarWinds Database Performance Analyzer. Free 14-day trial at: www.solarwinds.com/dpa-download

Join/query/table hints