

SQL Case Studies

by Tony Bokas

Case Study 1 – Trace and Tune Stored Procedure

Problem

Post procedure to SQL Server database started timing out after a few minutes. Development team increased the timeout threshold to 5 minutes to compensate. Procedure still timed out at 5 minutes.

Investigation

Conducted tracing and tuning of application processes while posting XML documents to server. Isolated the stored procedure that caused the timeout. Tuning with clustered indexes did not improve performance. Inspected the code and discovered dozens of JOIN statements in one INSERT statement.

Research and Hypothesis

Evaluated the procedure multiple times with varied XML sizes. Posting time was inconsistent. Sometimes, it took only a few seconds, and other times, it timed out. Reviewed public boards and Microsoft documentation. Suspected the execution plan cache was the culprit.

Solution

Modified the stored procedure. Removed the JOIN statements from the INSERT. Added multiple SELECT statements for variable assignments. Used variables with the INSERT instead. Post procedure reverted to expected times: a few seconds for normal reports and less than 20 seconds for large reports with many images and attachments.

Testing and Implementation

Evaluated the updated stored procedure on a test database with real reports. First, tested it in debug environment, then, tested the live application. Validated findings. Submitted code for review. Merged code into main branch for inclusion in next release.

Statistics

Team members: 2

Approximate timeline: 1 week

Lines of code: under 2,000

Case Study 2 – Migrate CRM Database

Problem

Organization needed to convert Act! database to Zoho CRM database. Act! database was over 15 years old. It did not accommodate advertising campaigns, help desk, correspondence, and other modern CRM features. Existing Act! setup and user routines created redundant and inaccurate customer profiles.

Investigation

Examined Act! export tools and wizards. Deemed the tools inadequate, as they did not export all screens and tables. Zoho CRM provided an import tool that required CSV formats. Import tool allowed mass migration or custom migration. Due to our ETL needs (update data, check for errors, and change data types) we chose custom migration. Obtained the raw Act! database backup.

Research and Hypothesis

Collected internal requirements from accounting, help desk, marketing, and development staff. Researched the apps in Zoho suite. Conducted meetings with peers and superiors about impactful design choices. Created detailed lists of data conversion requirements. Created multiple SQL scripts to analyze and convert Act! data to future Zoho data types and categories. Conducted initial test conversions in Zoho. Obtained final approval for full conversion and began the implementation.

Solution

Set a target date for the full migration. Wrote extensive SQL code to export, transform, and import data in multiple passes. Checked data conversions and determined realistic conversion success rates with the affected departments. Resolved technical and non-technical disagreements and needs among stakeholders.

Testing and Implementation

Began test migrations. Conducted final migration by the target date. Began using Zoho CRM for customer interactions. Wrote and ran additional SQL scripts post-migration to correct unforeseen conditions. Documented the results and held briefings with internal staff to apprise them of changes. Conducted additional research into Zoho suite. Trained staff on the new CRM technologies and expanded company use of available apps to streamline activities.

Statistics

Team members: 9

Approximate timeline: 3 months

Lines of code: under 5,000

Case Study 3 – Map Sales by Geolocation

Problem

Client wanted to use advanced features in our appraisal application suite to analyze comparable sales. However, client had thousands of sales with empty geographic fields (state, county, region, area, and zone). Client requested we populate the empty fields according to the sale latitude and longitude. This would map most fields and allow them to utilize our advanced analytical tools.

Investigation

Script design was theoretically simple. Query each sale and identify those with missing fields. Then, use the latitude and longitude to place each sale in polygons that correspond to a state, county, region, area, and zone. Needed to review the data quality and research available SQL Server libraries.

Research and Hypothesis

Selected the Geography library in SQL Server after reviewing Microsoft documentation. Did initial tests with coordinates to identify points inside and outside a simple polygon. Reviewed sample sales from the client. Client provided KML files with custom polygons. Imported KML files into Google Earth and manually reviewed their integrity. Analysis revealed some polygons were inverted. This caused points inside a polygon to appear as outside the polygon.

Solution

Approximately 100 polygons outlined sales regions in multiple states. Created a validation procedure to check for inverted polygons. Used regular expressions to correct any inverted polygons. Created a coordinate placing procedure. Added statistics and error messages to the script for internal and external use. Implemented polygon hierarchy for overlapping polygons with procedural code. Conducted multiple tests with sample sales for every polygon-delineated area.

Testing and Implementation

Forwarded the initial script to the client QA team for testing. Collaborated with client QA to improve the SQL implementation as needed. Re-ran local tests and reviewed the results. Examined updated sales for accuracy. Sent the final script to client QA for re-testing. After approval, the client Production team ran the script on their live database with our assistance. Client validated the accuracy of the mapped state, county, region, area, and zone fields. Over the coming months, the client forwarded any anomalies to us. Re-evaluated script to ensure the cause was a user-error rather than a script error.

Statistics

Team members: 4

Approximate timeline: 2 months

Lines of code: under 10,000