

Case Studies
For
MS SQL Server

Performance Optimization

Version - SQL Server 2012

Client based in – India

The Challenge

One of existing our clients whom we provide support for BI and SQL Server had a stored procedure giving poor performance and we were asked to analyse it for optimization. The limitation was to just re-structure the queries without making any changes to the underlying DB structure.

The Solution

We found that unnecessary use of temporary tables was made, where the result could be managed without them. These temporary tables accounted for a lot of memory usage. Keeping in mind the final output required, we eliminated certain temporary tables and wherever possible, combined multiple results into a single result set. Also, there was heavy and incorrect use JOINS in the SELECT statement giving the final output. We eliminated certain JOINS from the this query and shifted them to separate functions as per requirements. After optimization, the query that previously took 1.75 hrs to run for a certain set of inputs finished within 6 seconds.

Database Design

Version - SQL Server 2008

Client based in – India

The Challenge

For our client, handling and managing the business processes involved a lot of manual work which was tedious, error- prone and time consuming. The client suffered delays in making deliveries to the customers, which were danger for the client's image. The client wanted us to develop an easy to use ERP that would automate their business processes and help eliminate the cons of manual work. Giving the much obvious benefits of SQL Server, we decided to use it in for the back-end.

The Solution

Primary aspect that comes in mind while working with data is security. We used SQL Server's role based security to prevent unauthorized access to the database. We analysed client's business processes, and required permissions and roles were decided.

Next thing was to decide for the tables to be used, we created appropriate normalized tables so as to eliminate data redundancy. An important aspect that demanded our attention here was – on which tables to create index and when to create them. Indexes when used optimally provide a significant performance boost. Other minute details like primary keys and foreign keys were taken care of to ensure integrity of data.

After that, there was the turn to design stored procedures which would be used to access /modify the data. All the business operations of the client were grouped logically and relevant stored procedures were designed to reduce server round trips.

Transactions were used to ensure that data was not corrupted in cases of failures like power off or I/O failures.

A data purging mechanism was also designed so as to save resources occupied by unwanted historical data.

The system is running smoothly and the client has not complained about the performance issues so far.

Migrating from SSIS to T-SQL

Version - SQL Server 2014

Client based in – US

The Challenge

The client provided us some SSIS packages, which extracted required information from their servers having identical databases. For some servers, it was not possible to use SSIS and the client wanted to have T-SQL Queries developed, which would return the same results as those of the SSIS packages.

The Solution

This SSIS packages involved complex business logic. It was not as simple as picking up a SELECT query from the SSIS package and create a new stored procedure that would output the results of the query.

There were many derived column transformations, and we used CASE statements to achieve the desired result.

The SSIS packages used lookup transformations and filtered data. We used JOINS to achieve the results using T-SQL.

Condition based inserts and updates were managed in SSIS using conditional split. Instead of writing complex and lengthy INSERT and UPDATE statements, we found it beneficial to make use of the MERGE statement.

A couple of packages also involved PIVOT transformations. The pivoting was done on multiple columns. We choose to get the final result using a **GROUP BY statement** instead on applying multiple pivots on the data.

Apart from these, we also optimized several T-SQL statements originally used in the SSIS packages.

We completed the task accurately and well before time. This was a good learning curve for us.