CASE STUDY

SQUARESHIFT

BigQuery Optimization for Global Cloud-Based Learning and Talent Management Software

Client

The client is a leading global provider of cloud-based learning and talent management software. The company offers comprehensive solutions designed to help organizations recruit, train, manage, and engage their employees effectively.

With a focus on innovation and user experience, the company provides tools for learning management, performance management, succession planning, and more

TECHNOLOGY STACK

Google BigQuery

SQL

Outcome

The implementation of these strategies led to a dramatic reduction in the volume of data processed, thereby decreasing the associated costs significantly by **28%**. Additionally, the optimization reduced the demand on computing slots, contributing to lower operational costs and enhanced query performance.

PROJECT OBJECTIVES

A critical challenge that the client faced is managing large-scale data analytics controlling costs without compromising the efficiency and speed of data retrieval and analysis.The original architecture in their BigQuery setup involved dropping and recreating tables every time an ETL (Extract, Transform, Load) job was run. This process not only incurred high computational costs due to the constant processing of large datasets but also led to inefficient use of resources and time. They needed SquareShift's expertise to make strategic changes to their BigQuery environment and reduce the processing cost.

SOLUTION DELIVERY

- **Historical Data Preservation**: Initially, each ETL job led to dropping and recreating the full dataset in the tables. We shifted to a model where all existing data was loaded into a designated historical data table. This table now serves as a persistent data repository that does not require recreation with each job.
- **Incremental Data Processing:** Rather than reprocessing the entire dataset, the new approach involves fetching only the most recent data from the raw tables. This data is then processed and transformed incrementally before being inserted into the historical data table. This method significantly reduces the volume of data processed in each run.
- **Column Management:** SquareShift team streamlined the dataset by removing unused columns, thereby reducing the amount of data loaded and processed.
- **Join Optimization:** SQL joins were optimized to reduce complexity and execution time.
- **Table Clustering and Partitioning:** Implementing clustering and partitioning improved query performance and further reduced costs by optimizing data storage and retrieval processes.
- **Use of Common Table Expressions (CTEs):** CTEs were utilized to minimize redundancy, reducing the need to process the same tables repeatedly.
- **Selective Optimization:** Of the 31 tables managed, Only events tables underwent this optimization process.