Case Study: LLM-Driven SQL Optimization at Scale

Client: Northbridge Marketplace (company name anonymized). Figures as reported by the author.

Project Summary

At Northbridge Marketplace, a global online talent marketplace, I led a large-scale initiative to optimize **over 4,000 legacy SQL queries** using a large language model wired into an in-house prompt-orchestration layer. The project delivered **six-figure monthly savings** by finding inefficient query design, reducing data scanned, and cutting **Snowflake credit usage**.

Problem

The data engineering team managed thousands of legacy SQL files across marketing and analytics pipelines. The queries were:

- Poorly optimized for Snowflake, driving high compute cost
- Repetitive or redundant across teams
- Not built to scale as query volume grew
- Opaque: nobody could see which queries drove cost

The result was a steady rise in infrastructure spend and a data platform that was hard to scale.

Goal

- Analyze 4,000+ SQL files
- Measure each query's cost, data scanned and Snowflake credit consumption
- Use an LLM to generate optimized versions that keep the same results
- Quantify the savings
- Give engineers and analysts clear, prioritized recommendations

Solution and Technical Approach

- **Query ingestion and parsing:** a Python service that walked 4,000+ SQL files and loaded each query for analysis.
- **Cost modeling:** for each query, pulled execution time, data scanned (GB/TB) and Snowflake credits from query history.
- **LLM optimization layer:** prompts grounded in the query, its table metadata and its cost profile produced optimized SQL that preserved functional intent.
- **Cost simulation:** compared before and after metrics to estimate savings per query.
- **Recommendation engine:** one report per file with optimization suggestions, line-by-line diffs, estimated monthly savings, and a priority level (low, medium, high).

Results

- **Analyzed:** 4,000+ SQL queries
- **Optimized:** over 80% of files had actionable improvements
- **Savings:** six-figure monthly cost reduction
- **Efficiency:** engineering review time cut by 70% through automation
- **Scalability:** the system extends to new pipelines and is part of the query review process

Impact

- Analytics and data engineering teams cut query cost and complexity at scale
- Leadership could track cost savings per team and per pipeline
- Set the pattern for later LLM-assisted engineering tooling

Tools and Technologies

- LLM with an in-house prompt-orchestration layer
- Snowflake, SQL, Python
- Airflow for orchestration
- GitHub Actions for automation
- Looker for reporting and stakeholder dashboards

Lessons Learned

- LLMs improve engineering workflows most when they are grounded in domain metadata and hard constraints
- Cost visibility works best when it is proactive, not a monthly surprise
- Optimization is not only about compute: it also improves performance, reliability and developer experience

---

Artur Kochetkov · linkedin.com/in/artur-kochetkov · github.com/artur115