Accelerating dbt Model Migration from Snowflake to Databricks Using LLM-Powered Automation

Delivering the fastest dbt conversion through intelligent automation and scalable frameworks.

Executive Summary

This white paper details how Computomic successfully executed a large-scale migration of over **900 dbt models** from Snowflake to Databricks in **record time** using a **Large Language Model (LLM)-powered automation framework**. The project achieved **over 90% automation success** and reduced manual conversion effort from several weeks to mere days, setting a new benchmark for intelligent data platform modernization.

1. Introduction

Enterprises today are increasingly migrating from legacy cloud data platforms to **open, scalable architectures** built on Databricks. However, migrating hundreds of **dbt models**—often complex, tightly coupled, and Snowflake-specific—presents a significant challenge. Traditional methods involve weeks of manual rewriting and validation. Computomic developed an **LLM-driven intelligent automation framework** that fundamentally redefines how dbt migrations are performed.

2. Migration Goals and Objectives

- **Accelerate Migration:** Transition more than 900 dbt models to Databricks rapidly and efficiently.
- **Ensure Accuracy & Reusability:** Achieve greater than 90% syntax conversion success, minimizing manual intervention.
- **Preserve Functional Parity:** Maintain SQL logic integrity and performance equivalence between Snowflake and Databricks.
- **Reduce Cost & Time:** Cut estimated effort by over 70%, enabling faster time-to-value.

- **Establish a Scalable Framework:** Create a reusable, automated pipeline to handle future cross-platform dbt migrations.

3. The Starting Point: Snowflake-Centric dbt Environment

The initial state comprised a mature dbt project built for Snowflake, complete with SQL models, macros, and seeds stored in Git. The primary challenge was translating certain Snowflake-specific syntax, such as IFF, ARRAY_AGG, and FLATTEN, into Databricks-compatible SQL expressions without breaking dependency chains or logic consistency.

4. Target Architecture: LLM-Powered dbt on Databricks

Component	Description
Source	dbt project hosted in Git (Snowflake SQL, dbt macros, and seeds).
LLM Engine	Claude
Databricks Runtime	Python notebooks orchestrating model batches and API calls
Output	Translated dbt SQL models stored in Databricks repo with metadata tagging

This pipeline transformed model SQL code using prompt-engineered LLM requests, achieving high accuracy through structured prompts and iterative validation.

5. The Conversion Lifecycle

a. Extraction

Parsed dbt model SQL files and configuration files (dbt_project.yml) to create a structured migration input.

b. Prompt Engineering

Designed precise prompts for the LLM to ensure correct translation of Snowflake-specific syntax to Databricks SQL (Delta).
Example prompt:

> "Convert Snowflake SQL to Databricks SQL using Delta syntax, Spark functions, and compatible macros."

c. API Integration

Utilized Databricks REST API endpoints to batch-process dbt models via the Claude Sonnet 4 engine.

d. Validation

Performed syntax validation using dbt parse and Databricks SQL checks to confirm structural integrity.

e. Refinement

Applied iterative prompt tuning and test validation cycles to continuously improve accuracy and reduce manual effort.

6. Results and Quantifiable Impact

Metric	Outcome
Models converted	900
Automation success rate	88 - 93%
Manual effort reduction	From 10–14 weeks → ~4 days
Common function replacements	IFF() → IF(), ARRAY_AGG → COLLECT_LIST, FLATTEN → EXPLODE
Cost	$100-300 in LLM API usage
Scalability	Framework reusable for other SQL platforms

The framework's cost efficiency and automation accuracy demonstrate its effectiveness as a next-generation migration accelerator.

7. Best Practices and Lessons Learned

- **Leverage LLMs for Modernization:** Intelligent models can drastically reduce rewrite efforts when guided by domain-specific prompts.
- **Prompt Tuning is Critical:** Structured prompt design and inclusion of context variables from dbt_project.yml significantly improve accuracy.

- **Cross-validation is Essential:** Comparing original and converted SQL in tools such as *Beyond Compare* ensures syntactic and logical parity.

- **Iterative Refinement Yields Excellence:** Incremental improvements in prompt design and validation quickly compound into large efficiency gains.

- **Automation + Human Review:** Combining automated translation with targeted manual QA ensures quality without compromising speed.

8. Conclusion

Computomic's intelligent automation framework redefines how dbt migrations are performed, delivering **record-breaking speed, accuracy, and scalability**. By blending **LLM-driven translation**, **Databricks-native orchestration**, and **automated validation**, this approach sets a new industry standard for data platform modernization. The framework is **extensible to other platforms** and can serve as the foundation for future cross-cloud migrations.

Authors:

Sam Chourey, Resident Solutions Architect
Pavani Vure, Project Manager
Nagendra Beerangi, Divya Kale, Technical Leads