

# Migrating Legacy Data Warehouses to Snowflake

Ujjawal Nayak

Software Development Manager

## Abstract

The transition from legacy data warehouses to modern cloud-based platforms such as Snowflake has become a strategic priority for enterprises seeking scalability, flexibility, and improved analytical performance. However, migrating large-scale legacy systems presents significant challenges, including data compatibility issues, security concerns, performance bottlenecks, and business continuity risks. This article examines the key obstacles encountered in legacy-to-Snowflake migrations and outlines best practices for ensuring a smooth and efficient transition. Organizations can optimize their data warehousing capabilities by leveraging structured migration frameworks, automated ETL processes, and robust validation strategies while minimizing risks.

**Keywords:** Data Warehouse Migration, Snowflake, Cloud Data Platform, ETL, ELT, Schema Transformation, Data Security, Performance Optimization, Data Governance, Cloud Computing

## I. Introduction

Traditionally built on-premises, legacy data warehouses often struggle to keep up with modern businesses' evolving needs. Companies today require real-time analytics, scalability, and cost-efficient solutions, which these traditional systems fail to provide.

Snowflake, a cloud-native data warehousing solution, offers a highly elastic, fully managed platform where storage and computing are separated, ensuring superior performance and cost optimization. However, moving from legacy on-premises setups to Snowflake's cloud ecosystem is not a simple lift-and-shift process—it requires strategic planning, technical expertise, and careful execution.

This article breaks down the significant challenges businesses encounter during legacy-to-snowflake migration and provides best practices for a smooth transition.

## II. Challenges in Migrating Legacy Data Warehouses to Snowflake

### 1. Data Model and Schema Compatibility

Legacy data warehouses often follow rigid schema designs optimized for specific workloads, whereas Snowflake provides a more flexible, semi-structured data model. Differences in data types, indexing methods, and partitioning strategies can cause compatibility issues that must be addressed during migration.

## **2. Performance Optimization Differences**

Traditional databases rely on manual performance tuning techniques like indexing, partitioning, and query optimization. However, Snowflake's architecture leverages automatic scaling and micro-partitioning, meaning that copying old optimization techniques may not result in the expected performance gains.

## **3. Complexity of ETL and Data Transformation**

Legacy Extract, Transform, Load (ETL) processes are typically built for on-premise execution using custom scripts or ETL tools like Informatica, Talend, and SSIS. Migrating to Snowflake often requires re-engineering these workflows to fit a cloud-native Extract, Load, Transform (ELT) model, which involves loading data and transforming it inside.

## **4. Security, Governance, and Compliance Risks**

Migrating sensitive data to a cloud environment introduces concerns about data security, encryption, access control, and regulatory compliance (such as GDPR, HIPAA, and SOC 2). To avoid legal or compliance risks, businesses must ensure that Snowflake's security policies align with internal governance frameworks.

## **5. Downtime and Business Continuity Risks**

Without careful planning, data warehouse migration can cause significant service disruptions. Ensuring minimal downtime, seamless data transfer, and complete data integrity is crucial to preventing business operations from being impacted.

## **6. Cost Considerations**

Snowflake follows a pay-as-you-go pricing model, which can be cost-efficient if appropriately managed. However, inefficient migration strategies—such as excessive data transfers, unoptimized queries, and improper warehouse sizing—can lead to unexpected cost overruns.

# **III. Best Practices for a Successful Snowflake Migration**

## **1. Pre-Migration Planning and Assessment**

- a. Conduct a comprehensive assessment of your existing data warehouse, including data volume, schema complexity, ETL workflows, and performance benchmarks.
- b. Identify dependencies between data assets to determine whether a phased migration or whole cutover approach is better.
- c. Develop a detailed migration roadmap with clear timelines, key milestones, and risk mitigation strategies.

**2. Selecting the Right Migration Strategy**

- a. **Lift-and-Shift (Rehosting):** Moves the dataset with minimal modifications; best for simple migrations.
- b. **Replatforming:** Adjusts ETL pipelines and queries while keeping the overall data structure intact.
- c. **Refactoring:** Redesign data models and ETL processes to fully leverage Snowflake's cloud-native capabilities for better efficiency and scalability.

**3. Optimizing Data Ingestion and Transformation**

- a. Transition from **ETL to ELT** using **Snowflake's powerful SQL functions** and external tools like **DBT (Data Build Tool)**.
- b. Utilize **Snowpipe** for **real-time data ingestion** instead of batch processing.
- c. Convert **legacy stored procedures** into **Snowflake-compatible implementations** using **JavaScript-based User-Defined Functions (UDFs)** or external computing resources.

**4. Strengthening Security and Governance**

- a. Implement **Role-Based Access Control (RBAC)** to manage **user permissions** and **data access policies**.
- b. Enable **data encryption** (both in transit and at rest) using Snowflake's **built-in security features**.
- c. Define **data retention, auditing, and masking policies** to meet **industry regulations**.

**5. Performance Tuning and Cost Optimization**

- a. **Optimize storage** using **Snowflake's automatic compression and micro-partitioning**.
- b. Monitor and adjust **virtual warehouse sizes** to avoid unnecessary computing costs.
- c. Use **Snowflake's Query Profile tool** to identify **inefficient queries** and improve execution plans.

**6. Testing and Validation**

- a. Implement **robust data validation** procedures to ensure **data consistency and integrity** post-migration.
- b. Run **parallel execution** to compare **query performance** before and after migration.
- c. Conduct **User Acceptance Testing (UAT)** with **business stakeholders** to validate reports and dashboards.

**7. Phased Rollout and Post-Migration Monitoring**

- a. Start with a **pilot migration** to test workflows and **evaluate performance**.
- b. Gradually transition workloads while maintaining **dual environments** as a **fallback option**.
- c. Continuously **monitor data pipelines, query performance, and cost metrics** using **Snowflake's built-in tools**.

#### IV. Case Study: Migrating a Financial Services Data Warehouse to Snowflake

To illustrate these best practices, let's explore how a **large financial services company** successfully transitioned from an **on-premises Teradata warehouse to Snowflake**.

##### Challenges Faced:

- **Complex ETL workflows** with multiple transformation jobs.
- **Strict compliance requirements** due to **GDPR and financial data security regulations**.
- Inefficient indexing strategies cause **performance issues**.

##### Migration Approach:

- Used a **phased migration strategy**, moving **non-business-critical workloads first**.
- Replaced **traditional ETL processes** with an **ELT pipeline**, leveraging **Snowflake's SQL-based transformations**.
- Implemented **fine-grained RBAC policies and encryption** to secure sensitive data.

##### Results and Benefits:

- **Query execution times improved significantly** due to Snowflake's auto-optimization.
- **Lower storage and compute costs** through **more efficient resource allocation**.
- **Enhanced security and compliance** with **centralized governance and encryption**.
- **Faster data access and improved analytics** for better decision-making.

#### 5. Conclusion

Migrating from a legacy data warehouse to Snowflake presents technical and strategic challenges. However, with proper planning, optimized ETL processes, strong security measures, and a phased migration approach, businesses can fully unlock the potential of Snowflake's cloud-native architecture.

A well-executed migration strategy is key to reducing risks, lowering costs, and enabling faster, more reliable data insights—helping organizations stay ahead in today's data-driven world.

#### 6. References

- [1] Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.)*. John Wiley & Sons.
- [2] Inmon, W. H. (2005). *Building the Data Warehouse (4th ed.)*. Wiley
- [3] Microsoft SQL Server to Snowflake Migration: <https://www.snowflake.com/resource/microsoft-sql-server-to-snowflake-migration-reference-manual/>
- [4] Oracle to Snowflake migration: [https://medium.com/@amangupta\\_data/navigating-the-cloud-migration-from-oracle-exadata-to-snowflake-388be55c96da](https://medium.com/@amangupta_data/navigating-the-cloud-migration-from-oracle-exadata-to-snowflake-388be55c96da)
- [5] Teradata Corporation. (2023). *From Teradata to Snowflake: A Migration Strategy*. <https://www.teradata.com>

- [6] Gartner, Inc. (2023). *Magic Quadrant for Cloud Database Management Systems*. <https://www.gartner.com>
- [7] Stedman, D. (2021). *Cloud Data Warehouse Performance Benchmarks: Comparing AWS Redshift, Google BigQuery, and Snowflake*. *Journal of Cloud Computing*, 10(3), 45-62.
- [8] Hemsoth, N. (2021). *The Future of Cloud Data Warehousing: Why Enterprises Are Moving to Snowflake*. *The Next Platform*. <https://www.nextplatform.com>
- [9] Dresner Advisory Services. (2023). *Cloud Data Warehouse Market Trends: Adoption, Benefits, and Challenges*. <https://www.dresneradvisory.com>
- [10] Databricks vs. Snowflake Performance Report. (2023). *Cloud Data Warehouse Comparison: Performance, Cost, and Usability*. <https://www.databricks.com>
- [11] Madsen, M., & Boicea, A. (2020). *The Role of ELT in Modern Data Architectures: A Case Study on Snowflake Adoption*. *International Journal of Big Data*, 9(2), 78-93.
- [12] Ahmad, S., & Patel, R. (2022). *Optimizing Snowflake Costs: Techniques for Query Optimization and Warehouse Scaling*. *International Journal of Cloud Computing*, 12(4), 91-108.
- [13] Kapoor, A. (2023). *Reducing Snowflake Costs: Auto-Suspend, Resource Monitoring, and Query Optimization*. *Data Engineering Journal*, 15(1), 33-50.
- [14] Schmitz, L. (2021). *Maximizing Efficiency in Snowflake: Best Practices for Reducing Storage and Compute Costs*. *Cloud Data Strategies Journal*, 8(3), 77-89.
- [15] Riley, J. (2023). *Cost-Saving Strategies in Snowflake: Analyzing Query Execution and Resource Utilization*. *Proceedings of the Cloud Computing Conference*, 29(3), 112-129.