interworks

Migrating Analytics Workloads from SQL Server to Snowflake

This white paper walks through processes and planning involved when migrating large scale analytics from SQL Server to Snowflake.

Specifically we cover:

- Project planning
- Development and testing of automated migration processes Activities around the migration itself
- Post-migration cleanup

InterWorks have delivered several such migration projects including:

NHS Greater Manchester

In 2023/24, we migrated NHS Greater Manchester fully from SQL Server to Snowflake following best practices to rebuild many existing data flows. This included SSIS orchestrated processes that used bespoke Python to ingest Excel files, which we migrated to Snowflake using Azure Function Apps as the orchestrator.

Optio Group

Optio are a specialist insurance company based in London. We migrated their existing SQL Server based analytics to Snowflake, including key financial data from Total Objects, accounting data from Xero and a proprietary system," Optio Connect." Once the key data was in Snowflake, we used Matillion to set up efficient data transformation pipelines, with the output delivered to end users via Tableau. We have a case study describing our work with Optio here.

NHS South East London

For NHS South East London Integrated Care Board, we migrated 35 databases, more than 3,000 tables and around 7 TB in managed MSSQL to Snowflake extending their ADF framework for daily reloads and deltas.

Why SQL Server?

SQL Server is a great database tool. It's one of the most widely used in the world. Very many people are familiar with it. It has a solid and well thought out user interface and a well-deserved reputation for reliability and ease of use.

Like most databases, it is optimised for storage and retrieval of data a few rows at a time. If you want a database for logging sales, processing invoices or recording scientific data, it can be a great choice.

Data analytics, data science and AI consume data in a different way to the way transactions are recorded. Instead of focusing on lots of detail about individual transactions, data analysis involves analysing a large number of records but with only a small fraction of the information recorded in each record being used.

Why Move Analytics Loads to Snowflake?

Snowflake is better suited to data analytics workloads than SQL Server.

Vastly Better Scalability

With SQL Server, you are limited by the hardware hosting your server instances. It is possible to upgrade the hardware, but this requires capex purchases, downtime and sometimes reconfiguration so many organisations prefer to estimate hardware requirements in advance, then live within those limits. In practice, the amount of power your analytics solution needs varies substantially over time, with usage gradually increasing as the solution is rolled out, and with particular peaks around company month and year ends. To provide adequate performance during these peak times means over- provisioning hardware for much of the rest of the time.

By contrast, with Snowflake, computational power is rented by the minute, with compute resource that can be powered up and down automatically according to demand. You only pay for what you use. This can save substantial amounts of money, particularly in the early stages of a project where usage is just ramping up.

Snowflake has access to a substantial portion of the compute power within an Azure, AWS or GCP data centre. So while there will be some theoretical limit on the amount of compute resource available, it's not a limit that even very large organisations are going to be close to hitting.

Support for Modern Enterprise Features

Like most technology products, SQL Server went through a phase of rapid improvement in its early days but has since reached a stage of maturity where new features become more difficult to implement. For SQL Server, the pace of improvement has plateaued in recent years. As of writing, SQL Server 2022 is three years old and is little different from its predecessor, SQL Server 2019. The newest version (SQL Server 2025) is still in preview. The lack of innovation in recent years, combined with the cost and difficulty of updating to newer versions means that many organisations are still using much older versions of SQL Server today.

Snowflake is still adding features rapidly, and includes support for modern enterprise features such as iceberg tables, clean rooms, semi-structured data handling and extensive AI integration. In Snowflake, upgrades and patches happen weekly. They are automatic and transparent with no downtime, so there's no risk of being stuck on an obsolete version.

Feature	SQL Server 2022	Snowflake
Iceberg Table Support	No native support. Requires external pipelines or third-party tools to sync data to Iceberg tables.[1]	Native support. Iceberg tables are generally available and fully integrated, supporting open lakehouse and direct analytics on Iceberg/Parquet data.[2][3][4]
Data Clean Rooms	No native feature for data clean rooms.	Native, fully managed solution. Secure, privacy preserving collaboration with built-in controls and differential privacy.[5][6]
Semi-Structured Data	Enhanced support for JSON and graph data. Can store and query semi-structured data directly in SQL tables, but with more limited flexibility and native types than Snowflake.[7]	Native, first-class support for JSON, Avro, Parquet, ORC, XML. Uses VARIANT, ARRAY and OBJECT types. Can query and join semi-structured data natively, including from external sources without ingestion.[8][9]

Attract The Best Talent

Many data professionals prefer working with modern, scalable cloud platforms like Snowflake that offer advanced features such as automatic scaling, seamless data sharing, and integration with diverse tools, which SQL Server, especially on- premise versions, lacks or supports less efficiently.

Candidates often seek opportunities to develop expertise in cutting-edge cloud technologies with strong growth potential. Organizations that do not adopt Snowflake risk appearing less attractive to these candidates, who may favour employers investing in the latest cloud data platforms and offering career advancement in high-demand skills.

Why Now?

The Snowflake platform continues to add new features at a faster rate than SQL Server. The capability gap between the two platforms is only getting wider.

At the 2025 Snowflake summit in June, Snowflake announced some improvements to tooling that now make the move to Snowflake simpler and cheaper than it has been in the past.

1. Automated, AI-Powered Migration with SnowConvert AI

Snowflake's "SnowConvert AI" is now completely free. It automates and accelerates migrations from SQL Server and other legacy platforms to Snowflake.

It uses advanced AI to:

- Analyse your existing SQL Server codebase and data models.
- Automate code conversion (including tables, views, stored procedures, and ETL/BI logic).
- Validate converted code and migrated data, reducing manual effort and risk.

Snowflake claim that conversion and testing phases are now 2–3 times faster than manual methods (ref)

2. Data Loading from SQL Server to Snowflake without third-party tools using OpenFlow

OpenFlow (now generally available on AWS, coming soon to Azure) is a open, extensible, managed data ingestion tool built into Snowflake. The list of currently supported data sources is quite small, but it includes SQL Server, so you can load data directly from SQL server into Snowflake with no need for a third- party ETL tool.

Preparation

Establish the Core Project Team

The first activity with any project is to identify the people who will fulfil three key roles. These roles together form a project board. The project board meets periodically to oversee progress and may be called upon to agree corrective action if the project goes substantially off course, but it is not itself concerned with day to day management of the project.

The roles on the project board are:

- **Project Sponsor:** The person who provides funding and is ultimately accountable for the business benefits delivered by the project.
- **The Lead User:** This person represents those people who will be responsible for using the products created by the project to deliver business benefit. It is the responsibility of the lead user to liaise with and represent other users, ensuring that the project will allow them to deliver those benefits.
- **The Lead Supplier:** This person represents those people who will be creating the project's products. Where InterWorks are delivering this project, we appoint a project lead who can take on the role of lead supplier.

In addition to the project board, there is one more key person that needs to be appointed at the start of the project.

Project Manager

This is the person who is ultimately responsible for delivering the project, and reports to the project board. They manage the project plan and monitor progress and outcomes day to day.

The project manager may be assisted by a project management office, but ultimately any project should only have one project manager. A project manager can be internal or external to your organisation, but they should be experienced and/or qualified in project management.

Establish a Document Store and Code Base

Document Store

Right at the start of the migration project establish a place to store documentation (typically a Confluence space or a Sharepoint Library).

Here you can store project documentation (plans, risks, progress and meeting notes etc), as well as technical and operational documentation (training courses, procedures for changing data sources in analytics tools etc).

Git Repository

The migration project will involve scripts and code rather than manual processes where practical, so you should establish a git repository or equivalent to store this safely.

Pre-Mortem

When planning a project, it's easy for project teams to fall into roles, where some members feel they have to be relentlessly optimistic in order to "sell" the project internally, while others feel they need to balance this optimism so that risks can be properly identified and addressed. If it's always the same people in these roles, it's easy for communication to break down between different groups.

Sometimes it can feel unsafe for people to speak out about their concerns within the project planning phase, lest they be perceived as negative or unhelpful. Project teams can then fall into a kind of groupthink, a consensus that may be unrealistically optimistic, but where nobody really feels responsible for the view of the group as a whole.

Most people are used to the idea of a project post-mortem where people look back on a project and discuss what went well, what went wrong and how it might be done differently next time. The concept of a pre-mortem comes from organisational psychologist Gary Klein who first wrote about it in the Harvard Business Review in 2007.

A pre-mortem meeting during the early planning phases allows a group to discuss all the possible ways that the overall project might fail without judgement.

- It breaks overconfidence by imagining failure and critically analysing causes.
- It overcomes planning fallacy by anticipating obstacles and risks that might otherwise be overlooked.
- It encourages honest dialogue and diverse perspectives, flipping the group dynamic by rewarding those who identify potential problems in advance.

Document the Existing State

Most production SQL Server setups are only partially documented. There are often missing pieces that can cause headaches during migration. To avoid surprises, the project team needs to document the current environment thoroughly.

It's crucial to identify external dependencies like ETL tools, reporting systems, machine learning processes and custom apps that interact with SQL Server. Capturing these ensures nothing breaks when moving to Snowflake. Security roles, user permissions and provisioning processes should also be documented to replicate or improve access controls.

Creating a diagram of the current architecture helps communicate the environment and dependencies clearly. Where documentation is missing or outdated, fill gaps by interviewing stakeholders, using discovery tools and examining system metadata. Solid documentation upfront reduces risks and smooths the migration process.

Migrate or Leave Alone

Not everything in SQL Server can or should be migrated to Snowflake. Both platforms have distinct strengths suited to different workloads. When creating the project plan, It's important to separate elements that are to be migrated from those that are to be left in place.

In particular:

- Where an analytics data pipeline is to be recreated in Snowflake, we may reproduce the logic that exists in SQL Server. If data is still accessible in the data source, we can load it directly from there, rather than copy transient data from SQL Server.
- Where there is historical data in SQL Server that was once loaded from a data source, but is no longer accessible from the data source, we need to migrate that data from the existing SQL Server system.
- SQL Server's row based architecture means that it is most likely still the better choice for transactional workloads (leave alone).

Snowflake Data Architecture

Many existing SQL Server environments have evolved organically over many years, often without a clearly defined or consistent data architecture. This can lead to complexity, technical debt and challenges in managing and scaling the platform.

Migrating to Snowflake presents an opportunity to introduce modern data architecture best practices that will make the platform easier to manage and extend in the future.

A formal data architecture might include:

- **Declarative Logic:** Snowflake encourages the use of declarative SQL and set-based operations rather than procedural or imperative code. This simplifies

maintenance and can improve query efficiency and performance. Outside of Snowflake itself, tools like dbt can be used to enforce declarative logic within development teams.

- **Layered Medallion Architecture:** Adopting a layered architecture, often called the Medallion Architecture, which organizes data into Bronze (raw), Silver (data at various stages of integration and transformation) and Gold (curated, business-ready) layers. This structure simplifies governance and makes for easier troubleshooting.
- **Simplified Data Management:** Snowflake's architecture eliminates the need for manual partitioning, indexing and tuning, allowing teams to focus on logical data modeling and pipeline design.
- **Scalability and Separation of Compute:** Snowflake's multi-cluster, shared-data architecture enables independent scaling of compute resources for different workloads, allowing for consistent performance without having to over-provision hardware.

Replatform, Refactor or Reengineer

One of the most dangerous phrases in project management is "while we're at it, why don't we..." When migrating from SQL Server to Snowflake there are certainly opportunities to improve upon the way things worked in the legacy system.

Choosing the extent to which the migration project should simply lift and shift versus try to improve what previously existed should be a conscious high-level decision for the project board, as there are time, cost and risk implications that will impact the whole project.

We can characterise three possible approaches here as **replatforming, refactoring** and **reengineering.**

Replatforming

This involves a direct, like for like migration of objects and data. Internal and external behaviours remain the same.

Advantages:

- This is the fastest to deliver as most of the conversion work can be done automatically.
- The migration process can be made pretty much invisible, with the end user experience being unchanged.
- This is potentially the lowest risk option in the short term as the conversion process is largely automatic.

Disadvantages:

- This can perpetuate poorly written, inefficient code or duplicated code on the new system.
- Fails to take advantage of the many features in Snowflake that are not available in SQL Server that may improve reliability or performance.

This kind of approach is most appropriate where the project needs to be delivered to tight timelines, perhaps to meet a deadline related to support services or to move off an unsupported platform.

Refactoring

Refactoring focuses on small, internal code changes to improve readability and maintainability without altering external functionality.

Advantages:

- Can make code easier to understand, maintain and extend. Duplicate code can be removed.
- Can lead to some improvements in reliability and performance.
- As with replatforming, the process can be made seamless for the end user

Disadvantages:

- Can take longer than simple replatforming as it will involve some more manual rewriting of existing code.

Reengineering

This involves restructuring or reorganising the existing system at a more fundamental level, potentially altering external behaviour. This might include converting an existing procedural data pipeline to run declaratively, or replacing a large number of existing output tables that are not widely used with a leaner set of outputs focusing on updated business requirements.

Advantages:

- Allows the greatest scope for improvement to performance and reliability, following modern design principles such as a data product oriented medallion architecture.
- New functionality can take advantage of features in Snowflake (e.g. python integration, dynamic tables or AI functions) that do not exist in SQL Server.
- The output reflects what the business needs today, not what was needed some years ago.

Disadvantages:

- This approach will take the longest to complete due to the increased time taken to redesign the overall system and implement new functionality.
- Where existing external functionality is not preserved, this may force end users to change their existing processes, adding friction.
- Where existing output formats are not preserved, data validation becomes more complicated.

Which to Choose

The above approaches are not mutually exclusive. A solid approach can be to split the overall migration project into two phases:

- Migration phase, follows the replatforming approach to get off the legacy platform ASAP
- Refactoring/Re-engineering phase: Once the system is migrated, go back and target particular areas for a separate exercise in refactoring or reengineering.

This multi-phase approach allows migration process to complete quickly resulting in immediate benefits from reduced administrative overhead and greater scalability of Snowflake. Refactoring and reengineering efforts can then focus on particular areas where the effort involved will have the greatest positive impact.

Incremental Delivery or Big Bang

There are fundamentally two approaches to data migrations: incremental delivery or big bang.

Incremental Delivery

We can break down a large migration into more manageable chunks, starting by splitting the eventual audience into smaller groups and by splitting the deliverables into smaller feature sets.

- By user groups (choose small local group first, then expand)
- By functionality (MVP first, then add more features)

Delivering incrementally this way reduces risk and reduces the time gap between the project start and delivering the first tangible benefits.

This process can be more flexible as additional new features that become available during the project can be incorporated into later delivery stages.

Big Bang

The alternative, big bang approach delivers the whole project in one go.

This has its own advantages:

- Everyone is focused on a single cutover date.
- The process to cutover from the old to the new system is simplified as it's only done once, and there is no need to manage parallel running of two systems.
- The overall time between project start and final delivery is likely to be shorter.

There are disadvantages that mirror the advantages of incremental delivery:

- There is a higher risk of failure as a single cutover event becomes a single point of failure.
- Decisions made early on in the planning phase may be difficult to change later.

Declutter

When you move house, it makes sense to get rid of unused clutter rather than go to the expense of packing it up and moving it. Likewise, when migrating to a new data platform, it helps if you can identify objects that are no longer used so you can exclude them from the project.

Your SQL Server DBA may be able to manually sift through the current databases and schemas in SQL Server and identify items that are not being actively used. Beyond that, you may be able to automatically find items no longer in use, by checking audit logs. Within SQL Server, there are features such as "SQL Server Audit," "Extended Events" and "Query Store" that allow for long term tracking of object usage, but these features are not available in all versions and editions of SQL Server and are not enabled by default.

Snowflake has a built in query history, recording all access to Snowflake objects for one year. This can be extended by scheduling a periodic extraction of the query history to permanent tables; so once migration is complete, any unused objects can be automatically identified and flagged for deletion.

Enablement

Enablement is not simply a euphemism for training. It may include training, but also includes access to resources, tools and support. The goal is to remove barriers that prevent a user being able to make effective use of the analytics platform.

Snowflake Training

Training requirements will vary by stakeholder group. Where your migration project is a replatforming or refactoring exercise, then the experience for end users using analytics tools like Power BI or Tableau, should essentially be the same. For people who will be working directly with the Snowflake platform, Snowflake offer a range of instructor led training, on demand courses and free hands-on workshops.

Data Discoverability and Natural Language Queries

One of the toughest problems that many users face is in understanding the data that is available to them. Data marts may contain dozens of tables with similar sounding column names, each of which represent something slightly different. Snowflake has features that can help with this. Descriptions can be added to tables and columns to provide more contextual information and more recently Snowflake has introduced the concept of semantic models which contain metadata describing tables and columns including synonyms and the relations between them. These semantic models can then be used by Cortex Analyst, an LLM tool within Snowflake that allows natural language queries over your data.

Dedicated Data Governance and Catalog Tools

A trend that we've seen recently with many customers is the adoption of dedicated data governance tools such as Atlan. These tools work across your data stack and provide a unified view of the data assets available, including a business glossary and information about data ownership and SLAs. Most modern data governance tools include end-to-end data lineage tracking, so any data issues can easily be traced back to the source and end users can be automatically alerted of any issues or outages that have occurred upstream.

Creating the Project Plan

Once you've established the core project team, performed a premortem meeting to identify risks, documented the existing SQL Server installation, identified those areas that are to be migrated, refactored or left running on SQL Server, and thought about enablement, then you are in a position to create a more detailed project plan for the migration.

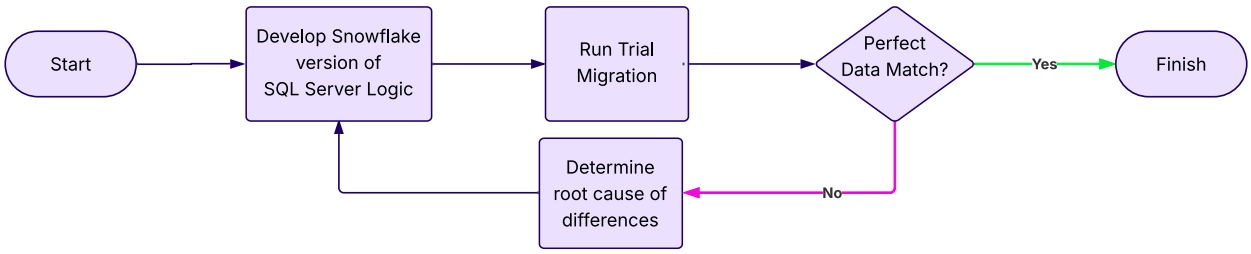
The project manager is responsible for creation and maintaining the project plan. The initial plan and subsequent changes (outside of predefined tolerances) should be approved by the project board.

Beyond the specifics of a SQL Server to Snowflake migration, there's a lot that could be written about generic project planning. There are well established and detailed frameworks for project planning (e.g. PMBOK, PRINCE2), so the project manager doesn't have to reinvent this process themselves.

Development & Test Cycles

> "I have not failed 10,000 times. I have succeeded in proving 10,000 ways that will not work." - *Thomas Edison*

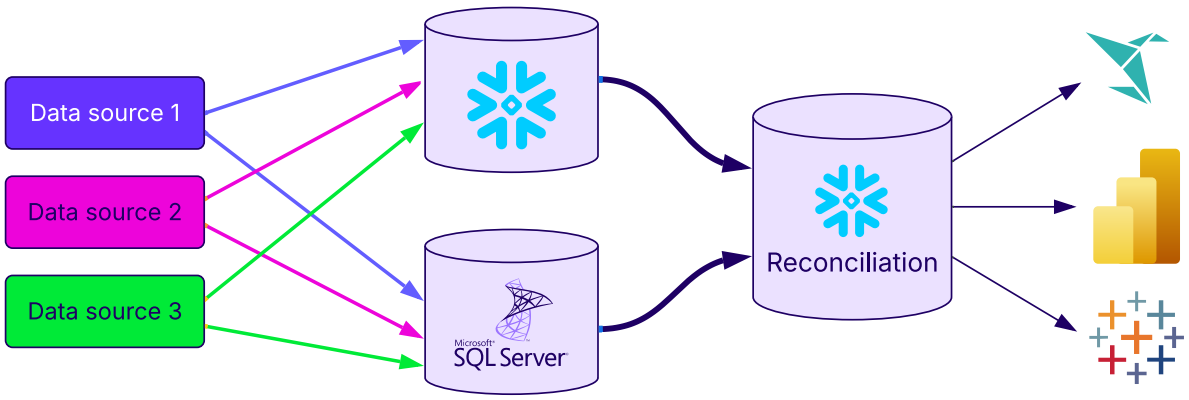
Database migration is essentially an iterative process. The process works by creating code that will copy objects from the old system to the new, and business logic in the new system that replicates the business logic of the old system. Depending on the complexity of your existing SQL server installation, this could involve migration or conversion of hundreds or thousands of individual objects.



Tools such as Snowflake's SnowConvert AI can help a great deal with the mechanical process of converting T-SQL to Snowflake SQL, but we still need to test the converted data pipeline to make sure the data output from Snowflake exactly matches the data output from SQL Server.

We test each converted object by completing a series of dry runs, at the end of which the output from Snowflake is compared to that from SQL Server. The migration process will most likely fail at first; but after repeated attempts with good testing and code control, we will have an automated migration process that will work reliably.

Instead of comparing output data manually, we can create a separate reconciliation database within Snowflake and then check for differences using a set of reconciliation tests in the form of custom SQL queries. The output from these reconciliation tests can be recorded and viewed within Snowflake itself, or displayed using a third-party analytics package.

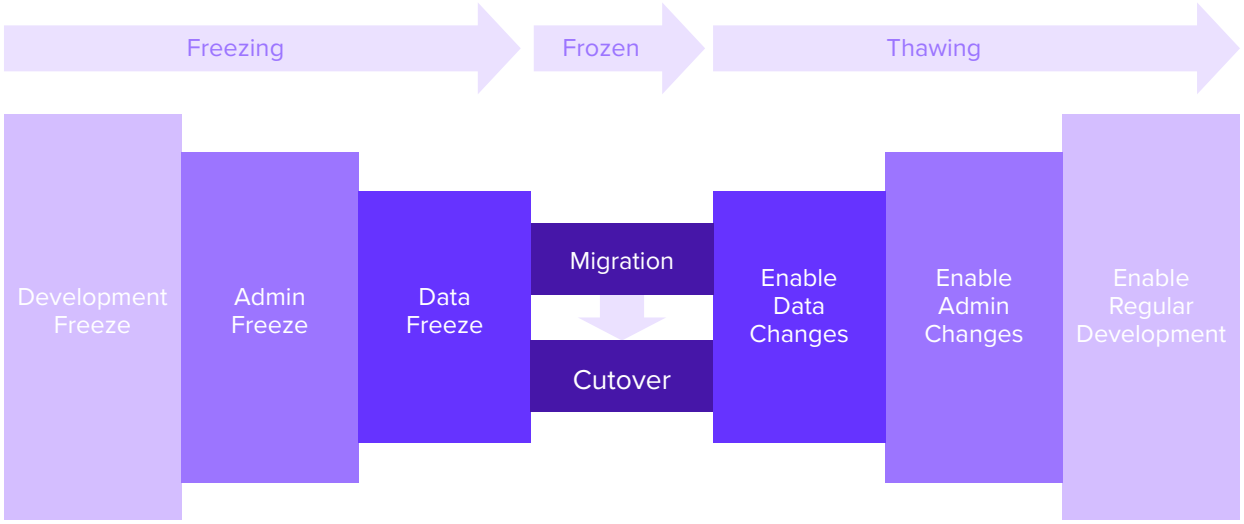


Having this kind of test framework means that the migration process can be developed and refined over dozens of iterations, until you are at the point where you can be confident that during the actual cutover, the migration processes will execute flawlessly.

Going Live

The Freeze / Thaw Cycle

As the date for the migration approaches there should be a series of earlier dates where changes on the source databases are restricted. This ensures that any migration scripts can be updated and tested before the migration data to capture any changes to the SQL server installation that have occurred up to the start of the frozen periods.



Development Freeze: 3-4 Weeks Before Migration

Any new development on the existing SQL Server installation ceases. This would include any new functionality or code changes to existing objects.

Admin Freeze: 1 week Before Migration

At this point any administrative changes such as new users or permission changes are halted. The only changes that are still allowed are disabling of users that have left the company and installation of emergency security patches.

Data Freeze: A Few Hours Before Migration

At this point all data processing on the SQL Server ceases. ETL processes are halted. Users are informed that the system will be unavailable while migration takes place. All user access is disabled.

Migration Event

Once all data is frozen on the source system, the migration process can be executed and reconciliation results reviewed. At this point, the project manager decides whether to proceed with the cutover to Snowflake or abort.

If cutover is approved, then any client applications pointing to SQL Server are redirected to point to Snowflake.

Enable Data Changes

User access is re-enabled so users can start logging in and viewing data in Snowflake. ETL processes are started and new data starts to be loaded into Snowflake. The data refresh processes are started within Snowflake.

Enable Admin Changes

New users and changes to permissions can now be performed within Snowflake.

Enable Regular Development

All regular development activities recommence on Snowflake.

Post-Migration Cleanup

Post Mortems

Following a migration event, it's helpful to perform a post mortem with key stakeholders to look at what went well and what could be improved.

Where the overall project is being implemented as a series of incremental deliveries, these post mortems can help improve future migration events.

It can be worth waiting a week or two after the migration is complete before conducting the post mortem exercise so that there is time for any issues relating to the migration to emerge. This post mortem would typically take the form of a meeting with the project manager and the project board.

Decommissioning

Once migration of all databases on a SQL Server instances has been completed, it may be possible to decommission the old SQL Server instance.

If the SQL Server instance is being used for ongoing OLTP services, then it will still need to be maintained, but the individual databases or schemas that have been fully migrated to Snowflake may be removed.

Follow the steps below to decommission a SQL Server in a safe way:

- Get confirmation from the project team that migration is successful and complete.
- Document a warranty period (typically 30-90 days) where the old server remains offline but recoverable if issues arise. Take the SQL Server offline.
- Run a final backup of the SQL Server if required for compliance.
- Uninstall the SQL Server software.
- Use secure data destruction methods to wipe any data drives.
- Remove the decommissioned server from any monitoring systems, network documentation and disaster recovery plans.
- Cancel software licences and support contracts

Leveraging Snowflake

The project to migrate to Snowflake may be over, but the real opportunity is in how the organisation now uses Snowflake to unlock new value.

Snowflake brings simplicity, flexibility and speed. Instead of worrying about infrastructure or capacity, teams can focus on insights and innovation. The ability to bring together all kinds of data, share it securely across the business and make it available in real time creates a platform for smarter, faster decisions.

This is where the real journey begins: Turning data into a strategic asset. With Snowflake, the organisation can experiment more easily, explore new use cases and collaborate more effectively. By embracing Snowflake's broader capabilities, organisations can continue to build momentum and deliver long-lasting competitive advantage.

About InterWorks

InterWorks is a people-focused tech consultancy delivering premier service and expertise in collaboration with strategic partners. Our clients trust us to guide them through their unique challenges, bringing together the right people and technologies to help everyone succeed. Covering data strategy, architecture, management, visualisation, AI, ML and more, we support you in every aspect of your BI strategy. We value laying a strong foundation, and our relational approach allows us to customize solutions that empower you to do data your way.

Tell Us Your Data Needs