



Netezza to Snowflake Migration Guide

MIGRATION STRATEGIES AND BEST PRACTICES

What's inside:

- 3 Why migrate?
- 5 Strategy—thinking about your migration
- 7 Migrating your existing Netezza warehouse
- 11 Need help migrating?
- 12 Appendix A—Migration tools
- 13 Appendix B—Migration preparation checklist
- 14 Appendix C—Data type conversion table
- 15 Appendix D—SQL considerations



Why migrate?

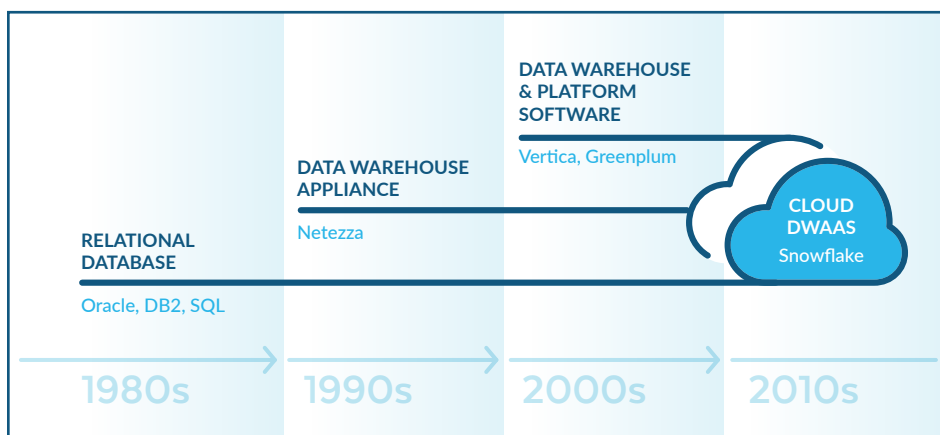
Decades ago, Netezza identified the need to manage and analyze large volumes of data. But just as the volume, velocity and variety of data have since changed, the cloud has enabled what's possible today with modern data analytics. For example, by separating compute from storage, Snowflake has developed a modern cloud data warehouse solution that automatically and instantly scales in a way not possible with Netezza, whether the current Netezza system is on-premises or in the cloud. Snowflake accomplishes this with its multi-cluster, shared data architecture.

YOUR MOTIVATION TO MIGRATE

Some of the key reasons customers migrate off of Netezza:

1. **Legacy platform**
Traditional technology fails to meet the needs of today's business users such as the increasing requirement for unlimited concurrency and performance.
2. **Cloud**
A strategy to move from on-premises to cloud implies a move away from traditional IT delivery models to a more 'on-demand', 'as-a-service' model with minimal management intervention.
3. **New data sources and workloads**
Key data sources for the modern enterprise are already in the cloud. The cloud also allows for new types of analytics to be assessed and refined without a long-term commitment to infrastructure or specific tools.
4. **Cost**
Snowflake allows true pay-as-you-go cloud scalability without the need for complex reconfiguration as your data or workloads grow.

THE EVOLUTION OF DATA PLATFORMS



WHY SNOWFLAKE?

Snowflake's innovations break down the technology and architecture barriers that organizations still experience with other data warehouse vendors. Only Snowflake has achieved all of the defining qualities of a data warehouse built for the cloud.



ZERO MANAGEMENT

Snowflake reduces complexity with built-in performance so there's no infrastructure to tweak, no knobs to turn and no tuning required.



ALL OF YOUR DATA

Create a single source of truth to easily store, integrate and extract critical insight from petabytes of structured and semi-structured data (JSON, XML, AVRO).



ALL OF YOUR USERS

Provide access to an architecturally unlimited number of concurrent users and applications without eroding performance.



PAY BY THE SECOND

Snowflake's built-for-the-cloud solution scales storage separate from compute, up and down, transparently and automatically.



DATA SHARING

Snowflake extends the data warehouse to the Data Sharehouse™, with direct, governed and secure data sharing within seconds, so enterprises can easily forge one-to-one, one-to-many and many-to-many data sharing relationships.



COMPLETE SQL DATABASE

Snowflake supports the tools millions of business users already know how to use today.



Snowflake enables the data-driven enterprise with instant elasticity, secure data sharing and per-second pricing. Snowflake's technology combines the power of data warehousing, the flexibility of big-data platforms, the elasticity of the cloud and live data sharing at a fraction of the cost of traditional solutions.

Strategy—thinking about your migration

WHAT SHOULD YOU CONSIDER?

There are several things to contemplate when choosing your migration path. It's usually desirable to pilot the migration on a subset of the data and processes. Organizations often prefer to migrate in stages, reducing risk and showing value sooner. However, you must balance this against the need to keep the program momentum and minimize the period of dual-running. In addition, your approach may also be constrained by the inter-relationships within the data, such as data marts that rely on references to data populated via separate process in another schema.

Questions to ask about your workloads and data

- What workloads and processes can you migrate with minimal effort?
- Which processes have issues today and would benefit from re-engineering?
- What workloads are outdated and require a complete overhaul?
- What new workloads would you like to add that would deploy easier in Snowflake?

'Lift and shift' vs a staged approach

The decision whether to move data and processes in one bulk operation or deploy a staged approach will depend on several factors. They include the nature of your current data analytics platform, the types and number of data sources, and your future ambitions and timescales.

Factors that lead to a lift-and-shift approach may focus on what you already have:

- You have highly integrated data across the existing warehouse.
- Or, you are migrating a single independent, standalone data mart.
- Or, your current system uses well-designed data and processes using standard ANSI SQL.
- Or, you have timescale pressures to move off legacy equipment.

Factors that may lead to a staged approach:

- Your warehouse platform consists of many independent data marts and other data applications, which can be moved independently over time.

- Or, you have critical data and processes within your data warehouse that no longer perform well and require re-engineering.
- Or, you've chosen to focus on new business requirements rather than reworking legacy processes.
- Or, you want to change your data ecosystem, such as adding a new ELT or other data ingestion tool, or move to new business intelligence (BI) and visualization tools.



WHAT YOU DON'T NEED TO WORRY ABOUT

When migrating to Snowflake from Netezza, you can forget about a number of concerns not relevant to your Snowflake environment.

Data distribution

In Snowflake, data distribution and/or data skewing are of no concern. Since compute is separate from storage in Snowflake's architecture, the data is not pre-distributed to the MPP compute nodes. In Snowflake, we have MPP compute nodes that do not rely on the data being distributed ahead of time.

Since Snowflake's data is not pre-distributed, it can scale to more parallel compute nodes instantly. With Netezza, customers would have two options:

1. The outright purchase of a new appliance, followed by migration to the new system with everything that entails.
2. Netezza customers who have purchased a "Capacity on Demand" system would need to request a capacity increase and wait for IBM's assistance to implement.

Both are processes that require significant planning, resources and delays, all of which are unnecessary with Snowflake.

Workload management

Workload management is unnecessary in a Snowflake environment due to its multi-cluster architecture, which enables you to create separate virtual warehouses for your disparate workloads so as to avoid resource contention completely.

Statistics generation

Snowflake automatically captures statistics, relieving DBAs from having to set up jobs to collect statistics for performance tuning. It's automatic in Snowflake so you no longer have to remember to add new tables to the process when your data warehouse grows.

Capacity planning

With Snowflake, you pay for only what you use. Snowflake is a SaaS product but further enhanced for efficiency with per-second, usage-based pricing. Under this model, Snowflake also offers the option for further cost reductions for customers who want to pre-purchase usage. On the flip side, with capacity planning for an on-premises Netezza system, you run a risk of over- or under-configuring your system. If you need more capacity, you must buy in predefined increments. With Snowflake's elastic storage and compute architecture, you never have this risk, so you can save money and avoid the time previously spent on extensive planning.

Disaster recovery

Netezza has several disaster recovery scenarios. Many of them require the purchase of another appliance. With Snowflake, this is unnecessary. Snowflake leverages many of the built-in features of the cloud, such as the automatic replication of data built into AWS. Snowflake is automatically synced across three AWS availability zones by design. There is no work on your part to establish this.

Separate dev/test environment

With Netezza, if you want to do development and testing, you will need additional appliances, which means an additional cash outlay for them plus the time to configure them. Not so with Snowflake as you can simply create another database in your account and set it up for any purpose you need, such as dev or test. In addition, with Snowflake's zero-copy clone feature, you can instantly populate those databases with complete copies of production data for no additional cost. With Netezza, you would have to endure the painstaking process of exporting your production data from one system to import it to your dev or test server.



Migrating your existing Netezza warehouse

To successfully migrate your enterprise data warehouse to Snowflake, you need to develop and follow a logical plan that includes the items in this section.

MOVING YOUR DATA MODEL

As a starting point for your migration, you'll need to move your database objects from Netezza to Snowflake. This includes the databases, tables, views and sequences in your existing data warehouse that you want to move over to your new Snowflake data warehouse. In addition, you may want to include all of your user account names, roles and objects grants. At a minimum, the user who owns the Netezza database must be created on the target Snowflake system before migrating data.

Which objects you decide to move depend highly on the scope of your initial migration. Regardless, there are several options to make this happen. The following sections outline three approaches for moving your data model from Netezza to Snowflake.

Using a data modeling tool

If you have stored your data warehouse design in a data modeling tool, you can generate the DDL needed to rebuild these objects. Since Snowflake uses standard

SQL, you simply need to pick ANSI SQL instead of Netezza as the output scripting dialect. Keep in mind, Snowflake is self-tuning and has a unique architecture. You don't need to generate code for any distribution keys. You only need basic DDL, such as CREATE TABLE, CREATE VIEW and CREATE SEQUENCE. Once you have these scripts, you can log into your Snowflake account to execute them.

If you have a data modeling tool, but the model is not current, we recommend you reverse engineer the current design into your tool, then follow the approach outlined above.

Using existing DDL scripts

You can begin with your existing DDL scripts if you don't have a data modeling tool. But you'll need the most recent version of the DDL scripts (in a version control system). You'll also want to edit these scripts to remove code for extraneous features and options not needed in Snowflake, such as distribution keys. Depending on the data types you used in Netezza, you may also need to do a search and replace in the scripts to change some of the data types to Snowflake optimized types. For a list of these data types, please see [Appendix C](#).

Creating new DDL scripts

If you don't have a data modeling tool or current DDL scripts for your data warehouse, you'll need to extract the metadata needed from Netezza in order to generate these scripts using a utility such as `nzdumpschema` or `NZ_DDL`. But for Snowflake, this task is simpler since you won't need to utilize distribution keys.

As mentioned above, depending on the data types in your Netezza design, you may also need to change some of the data types to Snowflake optimized types. You will likely need to write a SQL extract script of some sort to build the DDL scripts. Rather than do a search and replace after the script is generated, you can code these data type conversions directly into the metadata extract script. The benefit is that you have automated the extract process so you can do the move iteratively. Plus, you will save time editing the script after the fact. Additionally, coding the conversions into the script is less error-prone than any manual clean-up process, especially if you are migrating hundreds or even thousands of tables.

MOVING YOUR EXISTING DATA SET

Once you have built your objects in Snowflake, you'll want to move the historical data already loaded in your Netezza system over to Snowflake. To do this, you can use a third-party migration tool (see [Appendix A](#)) or an ETL tool, or you can use a manual process to move the historical data. When choosing an option, consider how much data you have to move. For example, to move 10s or 100s of terabytes, or even a few petabytes of data, a practical approach may be to extract the data to files and move it via a service such as AWS Snowball or Azure Data Box. If you have to move 100s of petabytes or even exabytes of data, AWS Snowmobile or Azure Data Box are likely the more appropriate options.

If you choose to move your data manually, you will need to extract the data for each table to one or more delimited flat files in text format either using External Tables to get single files or `NZ_UNLOAD` to create several files in parallel. You will then upload these files using the `PUT` command into a cloud storage staging bucket, either internal or external. We recommend that these files be between 100MB and 1GB to take advantage of Snowflake's parallel bulk loading.

After you have extracted the data and moved it to the storage service of your chosen cloud infrastructure provider, you can begin loading the data into your table in Snowflake using the `COPY` command. You can check out more details about our `COPY` command in our [online documentation](#).



MIGRATING YOUR QUERIES AND WORKLOADS

Data query migration

Since Snowflake uses ANSI-compliant SQL, most of your existing queries will execute on Snowflake without requiring change. However, Netezza does use some Netezza-specific extensions, so there are a few constructs to watch out for. Some examples include the use of LIKE ANY and QUALIFY. See [Appendix D](#) for details and suggested translations.

Migrating BI tools

Many of your queries and reports are likely to use an existing business intelligence (BI) tool. Therefore, you'll need to account for migrating those connections from Netezza to Snowflake. You'll also have to test those queries and reports to be sure you're getting the expected results.

This should not be too difficult since Snowflake supports standard ODBC and JDBC connectivity, which most modern BI tools use. Many of the mainstream tools have native connectors to Snowflake. Check our [website](#) to see if your tools are part of our ecosystem. Don't worry if your tool of choice is not listed. You should be able to establish a connection using either ODBC or JDBC. If you have questions about a specific tool, your Snowflake contact will be happy to help.

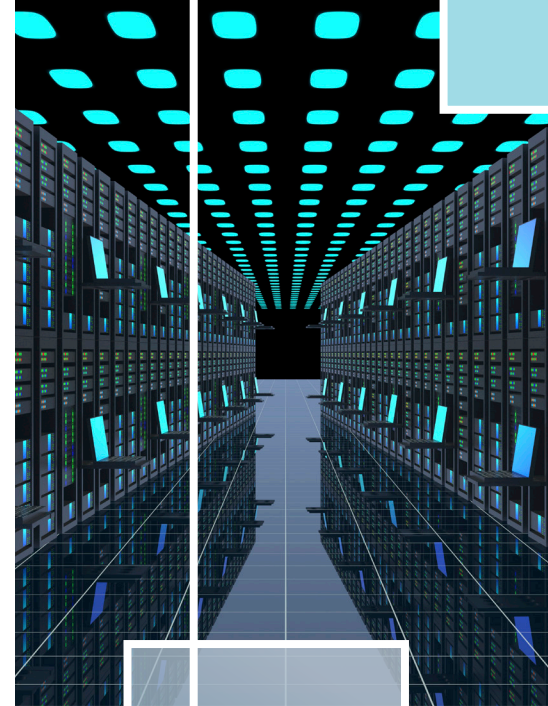
Handling workload management

As stated earlier, the workload management required in Netezza is unnecessary with Snowflake. The multi-cluster architecture of Snowflake enables you to create separate virtual

warehouses (compute clusters) for your disparate workloads to avoid resource contention completely. Your workload management settings in Netezza (WLM) will give you a good idea of how to set up Snowflake virtual warehouses. However, you'll need to consider the optimal way to distribute these in Snowflake.

As a starting point, create a separate virtual warehouse for each workload. You will need to size the virtual warehouse according to resources required to meet the SLA for that workload. To do so, consider the following:

- Is there a specific time period in which this workload must complete? Between certain hours? You can easily schedule any Snowflake virtual warehouse to turn on and off or to auto suspend and automatically resume when needed.
- How much compute will you need to meet that window? Use that answer to determine the virtual warehouse size.
- How many concurrent connections will this workload need? If you normally experience bottlenecks, you may want to use the [Snowflake multi-cluster warehouse](#) for those use cases to allow automatic scale-out during peak workloads.
- Think about dedicating at least one large virtual warehouse for tactical, high-SLA workloads.
- If you discover a new workload, you can easily add it, on demand, with Snowflake's ability to instantly provision a new virtual warehouse.





MOVING THE DATA PIPELINE AND ETL PROCESSES

Snowflake is optimized for an ELT (extract, load, transform) approach. However, we do support a number of traditional ETL (extract, transform, load) and data integration solutions. We recommend a basic migration of all existing data pipelines and ETL processes in order to minimize the impact to your project, unless you are planning to significantly enhance or modify them. Given the fact that testing and data validation are key elements of any changes to the data pipeline, maintaining these processes, as is, will reduce the need for extensive validation.

Snowflake has worked diligently to ensure that the migration of processes running on traditional ETL platforms is as painless as possible. Native connectors for tools such as Talend and Informatica make the process quick and easy.

We recommend running the data pipeline in both Snowflake and Netezza during the initial migration. This way, you can simplify

the validation process by enabling a quick comparison of the results from the two systems. Once you're sure queries running against Snowflake are producing identical results as queries from Netezza, you can be confident that the migration did not affect data quality. But you should see a dramatic improvement in performance.

For data pipelines that require re-engineering, you can leverage Snowflake's scalable compute and bulk-loading capabilities to modernize your processes and increase efficiency. You may consider taking advantage of our new [Snowpipe](#) tool for loading data continuously as it arrives to your cloud storage provider of choice without any resource contention or negative impact to performance thanks to Snowflake's cloud-built architecture. Snowflake makes it easy to bring in large datasets and perform transformations at any scale.

CUT OVER

Once you migrate your data model, your data, your loads and your reporting over to Snowflake, you must plan your switch from Netezza to Snowflake. Here are the fundamental steps:

1. Execute a historic, one-time load to move all the existing data.
2. Set up ongoing, incremental loads to collect new data.
3. Communicate the cut-over to all Netezza users, so they know what's changing and what they should expect.
4. Ensure all development code is checked in / backed up, which is a good development practice.
5. Point production BI reports to pull data from Snowflake.
6. Run Snowflake and Netezza in parallel for a few days and perform verifications.
7. Turn off the data pipeline and access to Netezza for the affected users and BI tools.



Need help migrating?

Snowflake is available to accelerate your migration, structure and optimize your planning and implementation activities; and apply customer best practices to meet your technology and business objectives. Snowflake's Engagement, Delivery and Advisory Services Team deploys a powerful combination of data architecture expertise and advanced technical knowledge of the platform to deliver high performing data strategies, proof of concepts and migration projects.

Our global and regional [solution partners](#) also have extensive experience performing proof of concepts and platform migrations. They offer services ranging from high-level architectural recommendations to manual code conversions. A number of Snowflake partners have also built tools to automate and accelerate the migration process.

Whether your organization is fully staffed for a platform migration or you need additional expertise, Snowflake and our solution partners have the skills and tools to accelerate your journey to cloud-built data analytics, so you can reap the full benefits of Snowflake quickly. To find out more, please contact the Snowflake sales team or visit Snowflake's [Customer Community Lodge](#).

Appendix A—Migration tools

Several of our Snowflake ecosystem partners have offerings that may help with your migration from Netezza to Snowflake. For more information, or to engage these partners, [contact](#) your Snowflake representative. These are just a few:

WIPRO

Wipro has developed a tool to assist in these migration efforts. CDRS Self Service Cloud Migration Tool (patent pending) provides an end-to-end, self-service data migration solution for migrating your on-premises data warehouse to Snowflake. This includes Snowflake-specific optimizations.

WHEREscape

Existing WhereScape customers who have built their Netezza warehouse using WhereScape RED can leverage all the metadata in RED to re-platform to Snowflake. Since WhereScape has native, optimized functionality for targeting a Snowflake database, it's relatively simple to redeploy the data warehouse objects and ELT code with these steps:

- Change the target platform from Netezza to Snowflake.
- Generate all the required objects (tables, views, sequences and file formats).
- Deploy and execute the DDL in Snowflake.
- Regenerate all the ELT code for the new target.
- Deploy and execute to continue loading your data in Snowflake.

ANALYTIX DS MAPPING MANAGER

If you need to migrate from one ETL tool or method to a new one, you might consider Mapping Manager. This tool uses a metadata-driven approach to migrate ETL/ELT code from one tool vendor to another.

Appendix B—Migration preparation checklist

DOCUMENT THE EXISTING SOLUTION

- List of databases that need to be migrated
- List of database objects that need to be migrated
- List of processes and tools that populate and pull data from the existing solution
- List of security roles, users and permissions
- List of Snowflake accounts that exist or need to be created
- Frequency of security provisioning processes
- Document the existing solution into an as-is architecture diagram

DETERMINE A MIGRATION APPROACH

- List of processes to migrate as is
- List of processes that need re-engineering
- List of processes that need to be fixed
- Draft of migration deliverables
- To-be architecture diagram

CAPTURE THE DEVELOPMENT AND DEPLOYMENT PROCESSES

- List of tools that will be introduced with the migration
- List of tools that will be deprecated after the migration
- List of development environments needed for the migration
- List of deployment processes used for the migration

PRIORITIZE DATASETS FOR MIGRATION

- List of data sets to migrate first
- Method for identifying process dependencies for data sets
- Documentation of process dependencies for data sets

IDENTIFY THE MIGRATION TEAM

- List of migration team members and roles
- Contact information for all team members

DEFINE THE MIGRATION DEADLINES AND BUDGET

- List of business expectations for the migration deadline
- Documented budget allocated for the migration project
- Completed estimation template for Snowflake virtual warehouses

DETERMINE THE MIGRATION OUTCOMES

- List of high-level desired outcomes and assumptions at the completion of the migration
- Documented plan for communicating the migration project wins to stakeholders

Appendix C—Data type conversion table

Sample data type mappings needed when moving from Netezza to Snowflake.

NETEZZA DATA TYPE	NOTES	SNOWFLAKE DATA TYPE	NOTES
BOOLEAN		BOOLEAN	
CHAR	Max 64K	CHAR	Max 16MB
VARCHAR	Max 64K	VARCHAR	Max 16MB
NCHAR	Max 64K	CHAR	Max 16MB
NVARCHAR	Max 64K	VARCHAR	Max 16MB
DATE		DATE	
TIMESTAMP		TIMESTAMP	
TIME		TIME	
TIME WITH TIME ZONE		TIMESTAMP_TZ	
NUMERIC(p,s)		NUMERIC	
REAL		REAL (FLOAT)	
DOUBLE PRECISION		DOUBLE PRECISION (FLOAT)	
INTEGER		INTEGER	
BYTEINT		BYTEINT	
SMALLINT		SMALLINT	
BIGINT		BIGINT	

Appendix D—SQL considerations

Representative list of changes, so Netezza SQL queries run correctly in Snowflake.

AGE

This function retrieves the interval between two timestamps/dates and provides precision to the same degree as the data. This functionality can be achieved in Snowflake via the DATEDIFF function:

NETEZZA	SNOWFLAKE
Select age('10-22-2003', '7-6-2002');	SELECT DATEDIFF(day, '2003-10-22', '2002-7-6');

NOW

This returns the current timestamp. This can be achieved via the CURRENT_TIMESTAMP function in Snowflake:

NETEZZA	SNOWFLAKE
SELECT NOW();	SELECT CURRENT_TIMESTAMP();

STRPOS(S,B)

Retrieves the starting position of string “b” in string “s”. Can be achieved via the POSITION function in Snowflake: :

NETEZZA	SNOWFLAKE
SELECT STRPOS("fuzzy wuzzy","uzzy");	SELECT POSITION("uzzy","fuzzy wuzzy");

CURRENT_PATH

Retrieves the PATH session variables for the current Netezza session. Similar functionality can be achieved in Snowflake by taking substrings from CURRENT_SCHEMAS.

NETEZZA	SNOWFLAKE
SELECT CURRENT_PATH;	SELECT substr(current_schemas(), 3, length(current_schemas()) - 4) as CURRENT_PATH;