snowflake®

TERADATA TO SNOWFLAKE MIGRATION GUIDE

Don’t let your past determine your future

CHAMPION GUIDES

TABLE OF CONTENTS

WHY MIGRATE?

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

YOUR MOTIVATION TO MIGRATE

Some of the key reasons enterprises migrate off of Teradata:

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 storage and compute scalability without the need for complex reconfiguration as your data or workloads grow.

WHY SNOWFLAKE?

Snowflake's innovations break down the technology and architecture barriers that organizations still experience with traditional data warehouse vendors. Only Snowflake has achieved all six of the defining qualities of a true cloud data platform, as displayed in the chart below.

THE CORE OF SNOWFLAKE

Snowflake delivers the performance, concurrency, and simplicity needed to store and analyze all data available to an organization in one location. Snowflake's technology combines the power of data warehousing, the flexibility of big data platforms, scalable compute and storage, and live data sharing at a fraction of the cost of traditional solutions.

FASTER ANALYST ACCESS TO DATA

Snowflake's elastic, near-unlimited scale and speed means analysts have fast access to all current and historical data at any time to make quicker, more accurate decisions.

NEAR-ZERO MAINTENANCE

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

ALL OF YOUR DATA

With Snowflake, you can 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).

DATA SHARING

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

ALL OF YOUR USERS

Snowflake allows a virtually unlimited number of concurrent users and applications without eroding performance.

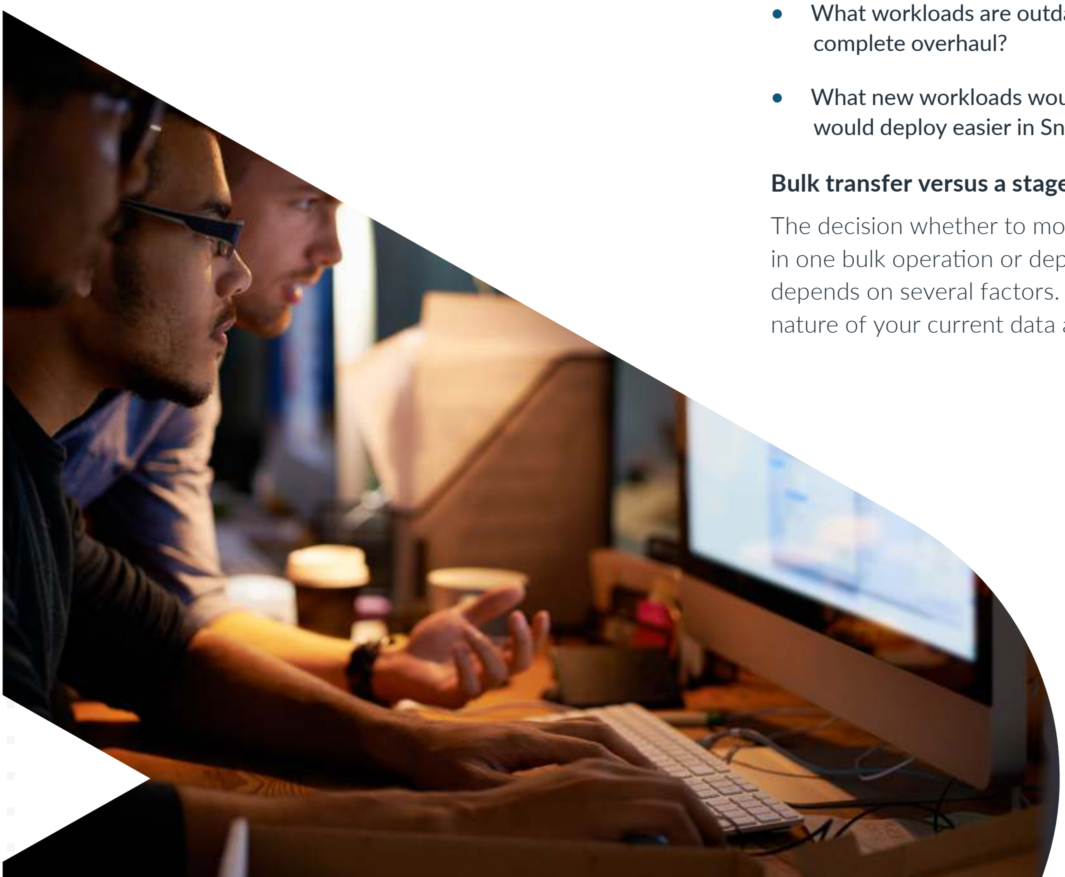
COMPLETE SQL DATABASE

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

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 a separate process in another schema.



Questions to ask about your workloads and data:

- What workloads and processes can be migrated 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?

Bulk transfer versus a staged approach

The decision whether to move data and processes in one bulk operation or deploy a staged approach depends 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 time frames.

Consider moving data in one bulk transfer if you have:

- Highly integrated data across the existing warehouse
- An independent, standalone data mart
- Well-designed data and processes using standard ANSI SQL
- A need to move off of legacy equipment quickly

Consider using a staged approach, if you have any of the following:

- A warehouse platform with many independent data marts and other data applications that can be moved independently
- Critical data and processes within your data warehouse that no longer perform well and need re-engineering
- New business requirements that can't be met by reworking legacy processes
- Changes to your data ecosystem, such as new data ingestion, BI, or visualization tools

WHAT YOU DON'T NEED TO WORRY ABOUT

When you migrate to Snowflake from Teradata, you can ignore the following factors because they are no longer relevant:

Data distribution and primary indexes

Snowflake does not need primary indexes. Since compute is separate from storage in Snowflake's architecture, the data is not pre-distributed to the MPP compute nodes. Snowflake has 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 Teradata, the data would have to run a reconfig, with new AMPs/nodes added and new hashmaps created before the data from the physical table could be redistributed to the new AMPs. This is a process that requires significant planning and resources, which impacts performance but is not necessary with Snowflake.

Indexing and query optimization

Snowflake has an optimizer built from the ground up and architected for MPP and the cloud. Snowflake understands parallel execution plans and automatically optimizes them, relieving customers of this task. This means there are no indexes in Snowflake so you don't have to migrate your USIs or NUSIs.

Workload management

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

Statistics collection

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 that is further enhanced for efficiency with 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 Teradata system, you run a risk of over- or under-configuring your system. Even with Teradata Vantage, you have the similar capacity planning risk as compute and storage are fixed per instance. If you need more capacity, you must buy it 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

Teradata has several disaster recovery scenarios. Many of them require the purchase of another system, and purchasing software such as Unity to implement these scenarios. With Snowflake, none of this is necessary. Snowflake leverages many of the built-in features of the cloud, such as the automatic replication of data built into AWS. Snowflake is implemented in multiple regions on AWS, Azure, and Google Cloud and supports cross-cloud replication for disaster recovery to your cloud provider and region of choice. There is no work on your part to establish this.

Separate dev/test environment

With Teradata, to perform development and testing, you need additional servers, which means an additional cash outlay for the hardware plus the time to configure the hardware. But with Snowflake, 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 Teradata, you 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 TERADATA 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 Teradata to Snowflake. This includes the databases, tables, views, and sequences in your existing data warehouse that you want to move to Snowflake Cloud Data Platform. In addition, you may want to include all of your user account names, roles, and objects grants. At a minimum, create the user who owns the Teradata database on the target Snowflake system before migrating data.

Which objects you decide to move will be highly dependent on the scope of your initial migration. There are several options for making this happen. The following sections outline three possible approaches for moving your data model from Teradata 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 as the output scripting dialect rather than Teradata. Keep in mind, Snowflake is self-tuning and has a unique architecture. You won't need to generate code for any indexes, partitions, or storage clauses of any kind that you may have needed in Teradata. 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.

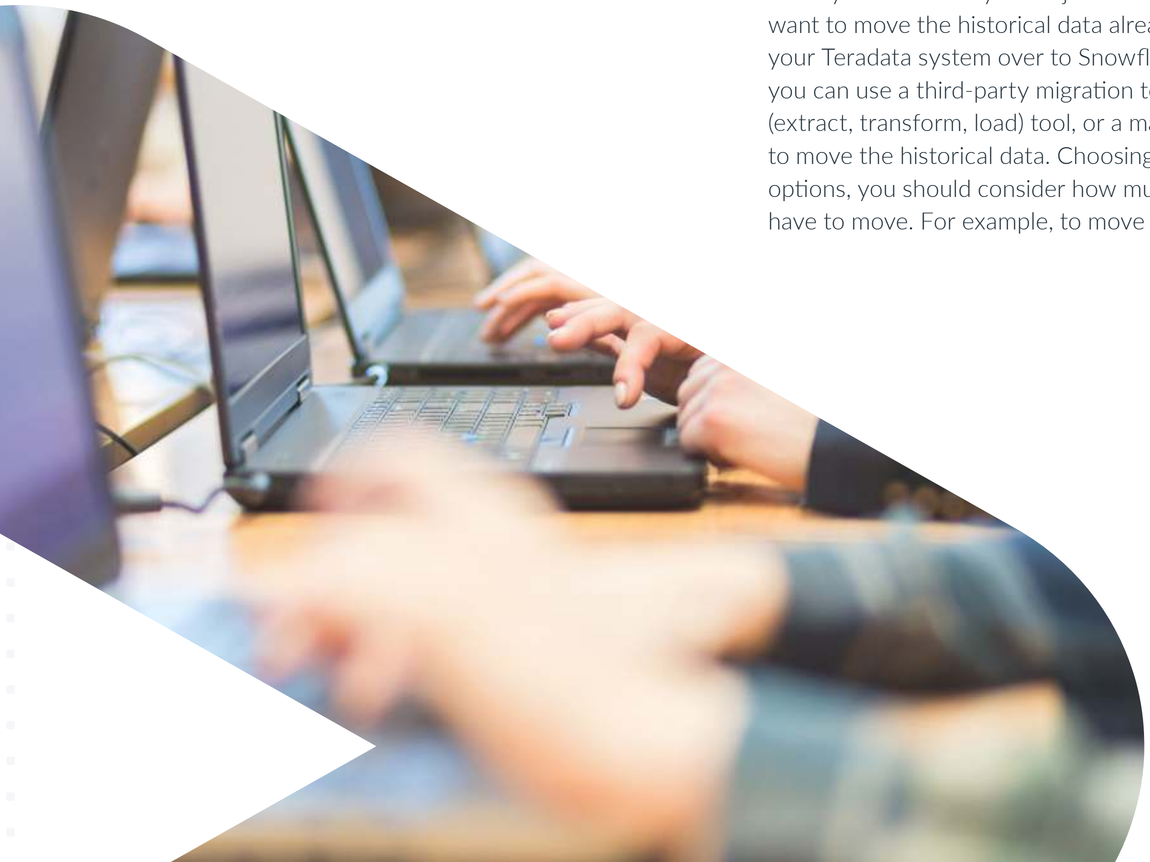
Using existing DDL scripts

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 primary indexes and other storage or distribution related clauses. Depending on the data types you used in Teradata, 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, see Appendix A.

Creating new DDL scripts

If you don't have current DDL scripts or a data modeling tool, you will need to extract the metadata needed from the Teradata data dictionary to generate these scripts. But for Snowflake, this task is simpler since you won't need to extract metadata for indexes and storage clauses.

As mentioned above, depending on the data types in your Teradata 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 cleanup 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 Teradata system over to Snowflake. To do this, you can use a third-party migration tool, an ETL (extract, transform, load) tool, or a manual process to move the historical data. Choosing among these options, you should 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, Azure Data Box, or Google Transfer Appliance. If you have to move 100s of petabytes or even exabytes of data, AWS Snowmobile is likely the more appropriate option.

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 using Teradata Parallel Transporter (TPT). Then upload these files using the PUT command into your cloud provider's blob storage. We recommend these files be between 100 MB and 1 GB to take advantage of Snowflake's parallel bulk loading.

After you have extracted the data and moved it to your cloud provider's blob storage, you can begin loading the data into your table in Snowflake using the COPY command. You can check out more details about the COPY command in 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, Teradata does use some Teradata-specific extensions, so there are a few constructs to watch out for. See Appendix B for details.

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 Teradata 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. Don't worry if your tool of choice is not available. 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.

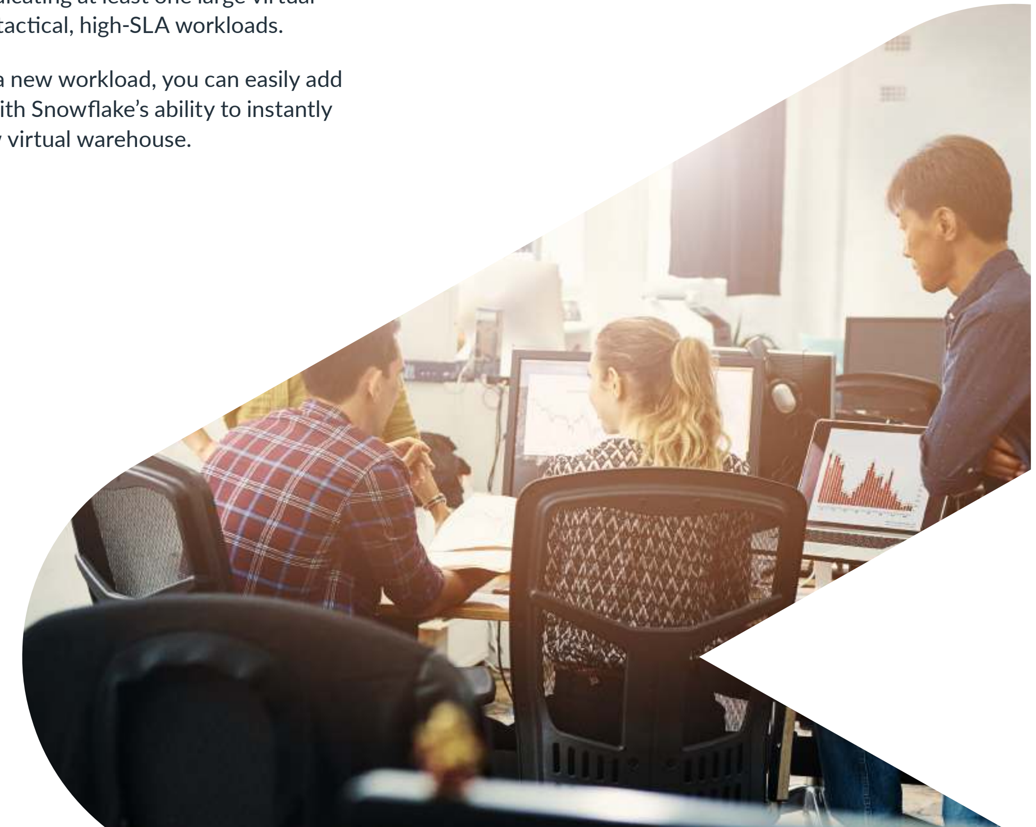
Handling workload management

As stated earlier, the workload management required in Teradata is unnecessary with Snowflake. The multi-cluster architecture of Snowflake allows you to create separate virtual warehouses for your disparate workloads to avoid resource contention completely. Your workload management settings in Teradata (TASM or TIWM) will give you a good idea of how you'll want 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 needs to complete? Between certain hours? You can easily schedule any Snowflake virtual warehouse to turn on and off, or just auto suspend and automatically resume when needed.
- How much compute will you need to meet that window? Use that to determine the appropriately sized virtual warehouse.
- 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 workloads 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, Snowflake supports many traditional ETL (extract, transform, load) and data integration solutions. We recommend a basic migration of all existing data pipelines and ETL processes 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.

Run the data pipeline in both Snowflake and Teradata 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 Teradata, you can be confident that the migration did not affect data quality. But you should see a dramatic improvement in the 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 Snowpipe for loading data continuously as it arrives to your cloud storage provider of choice, without any resource contention or impact to performance. 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 Teradata 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 Teradata 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 Teradata in parallel for a few days and perform verifications.
7. Turn off the data pipeline and access to Teradata for the affected users and BI tools.

NEED HELP MIGRATING?

Snowflake's solution partners and Snowflake's Professional Services team offer several services to accelerate your migration and ensure a successful implementation. The Snowflake Alliances team is working with top-tier system integrators that have experience performing platform migrations.

Snowflake solution partners and Snowflake's Professional Services team understand the benefits of Snowflake and apply their experience and knowledge to the specific challenges that your organization may face during the migration process. They offer services ranging from high-level architecture recommendations to manual code conversion. Additionally, many Snowflake partners have built tools to automate and accelerate the migration process.

Whether your organization is fully staffed for a platform migration or you need additional staffing, Snowflake's solution partners and Snowflake's Professional Services team have the skills and tools to make this process easier, so you can reap the full benefits of Snowflake.

To find out more, contact Snowflake's solutions partner team or the Snowflake sales team.

To understand the business benefits of migrating from Teradata to Snowflake, click here.

APPENDIX A: DATA TYPE CONVERSION TABLE

This appendix contains a sample of some of the data type mappings you need to know when moving from Teradata to Snowflake. Many are the same, but you will need to change a few.

TERADATA DATA TYPE	NOTES	SNOWFLAKE DATA TYPE	NOTES
BYTEINT		BYTEINT	
SMALLINT		SMALLINT	
INTEGER		INTEGER	
BIGINT		BIGINT	
DECIMAL		DECIMAL	
FLOAT		FLOAT	
NUMERIC		NUMERIC	
CHAR	Up to 64K	CHAR	Up to 16MB
VARCHAR	Up to 64K	VARCHAR	Up to 16MB
LONG VARCHAR	Up to 64K	VARCHAR	Up to 16MB
CHAR VARYING(n)		CHAR VARYING(n)	
REAL		REAL	
DATE		DATE	
TIME		TIME	
TIMESTAMP		TIMESTAMP	
BLOB		BINARY	Up to 8MB
CLOB		VARCHAR	Up to 16MB
BYTE		BINARY	
VARBYTE		VARBINARY	
GRAPHIC		VARBINARY	
JSON		VARIANT	
ARRAY		ARRAY	

APPENDIX B: SQL CONSIDERATIONS

Below are examples of some changes you may need to make to your Teradata SQL queries so they will run correctly in Snowflake. Note that this is not an all-inclusive list.

UPDATING DATA THROUGH A VIEW

Teradata allows inserts, updates, and deletes to be executed against a view, which will then update the underlying table. In Snowflake, inserts, updates, and deletes must be executed against a table and can't be executed against a view. Again, load processes may need to be re-engineered to account for this.

UPDATE SYNTAX

Teradata allows the FROM in an UPDATE statement to come before the SET statement. In Snowflake, the UPDATE syntax requires that the FROM comes after the SET statement.

DELETE ALL SYNTAX

Teradata supports adding ALL to the end of a DELETE statement. In Snowflake, ALL at the end of a DELETE statement isn't supported and needs to be removed.

TERADATA-SPECIFIC SYNTAX

Teradata has SQL syntax for creating tables (DDL) that isn't used in Snowflake:

- SET/MULTISET
- FALLBACK
- PRIMARY INDEX
- PARTITION BY
- COMPRESS
- FORMAT
- INDEXES

Teradata has SQL syntax with views (DDL) that isn't used in Snowflake:

- LOCKING ROW FOR ACCESS
- SEL (must be spelled out as SELECT)
- DEL (must be spelled out as DELETE)

Also REPLACE VIEW syntax in Teradata should be updated to CREATE OR REPLACE VIEW in Snowflake.

ABOUT SNOWFLAKE

Snowflake Cloud Data Platform shatters the barriers that prevent organizations from unleashing the true value from their data. Thousands of customers deploy Snowflake to advance their businesses beyond what was once possible by deriving all the insights from all their data by all their business users. Snowflake equips organizations with a single, integrated platform that offers the only data warehouse built for any cloud; instant, secure, and governed access to their entire network of data; and a core architecture to enable many other types of data workloads, including a single platform for developing modern data applications. Snowflake: Data without limits. Find out more at **snowflake.com**.