snowflake®

ORACLE TO SNOWFLAKE MIGRATION GUIDE

Migration strategies and best practices

CHAMPION GUIDES

TABLE OF CONTENTS

WHY MIGRATE?

Oracle has had a role in relational databases and data warehouses for over 30 years. With the introduction of engineered systems such as Exadata, Exalytics, Exalogic, SuperCluster, and the 12c Database, the tight integration of storage and compute enabled faster processing of larger amounts of data with on-premises infrastructure. However, the volume, velocity, and variety of data has increased dramatically, and the cloud has enabled greater possibilities with modern data analytics. For example, by separating compute from storage, Snowflake has developed a modern cloud data platform that automatically and instantly scales compute and storage in a way not possible with Oracle, whether the current Oracle system is on-premises or hosted in the cloud. Snowflake Cloud Data Platform accomplishes this with its multi-cluster, shared data architecture.

YOUR MOTIVATION TO MIGRATE

Some of the key reasons customers migrate off of Oracle include the following:

1. **The legacy platform is inadequate:** Traditional technology fails to meet today's business user's needs, such as unlimited concurrency and performance.
2. **The cloud offers a no-management solution:** Moving from on-premises to the cloud means moving away from traditional IT delivery models to on-demand, as-a-service models with minimal management or intervention. Oracle's recent database cloud options are not quite ready for prime time. At a minimum, they are little more than existing server technologies hosted in Oracle data centers.
3. **New data sources and workloads 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. **The cost is affordable and predictable:** Snowflake allows for true pay-as-you-go 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 other data warehouse vendors. Only Snowflake has achieved all six of the defining qualities of an effective cloud data platform:

Snowflake delivers the performance, concurrency, and simplicity needed to store and analyze all data available to an organization, in one location and at a fraction of the cost of traditional solutions.

NEAR-ZERO MAINTENANCE

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

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.

ALL OF YOUR DATA

With Snowflake, you can create a single source of truth to easily store, integrate, and extract critical insights from petabytes of structured and semi-structured data (JSON, Avro, ORC, Parquet, or XML).

DATA SHARING

Snowflake enables direct, governed, and secure data sharing in near-real time, 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 performance degradation.

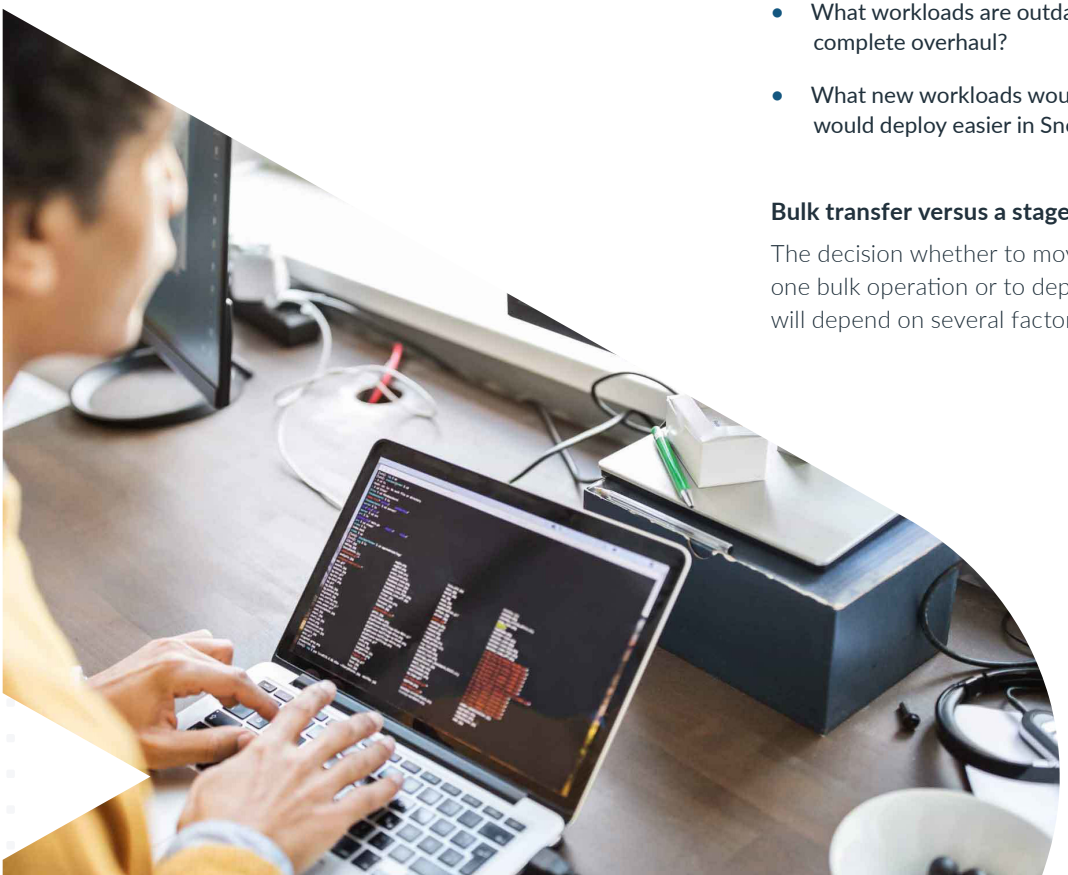
COMPLETE SQL DATABASE

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. Many organizations pilot the migration on a subset of the data and processes. Then they migrate in stages, reducing risk and showing value sooner. However, you must balance risk mitigation against the need to maintain program momentum and minimize the period of running systems in parallel. In addition, your approach may be constrained by interrelationships 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 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?

Bulk transfer versus a staged migration

The decision whether to move data and processes in one bulk operation or to 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 time frames.

Consider moving data in one bulk transfer if you have any of the following:

- Highly integrated data across the existing warehouse
- A single independent, standalone data mart
- Well-designed data and processes using standard ANSI SQL
- A need to move off 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 over time
- Critical data and processes within your data warehouse that no longer perform well and require 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 migrating to Snowflake from Oracle, you can ignore the following factors because they are no longer relevant:

Data distribution and primary indexes

In Snowflake, you don't need to worry about primary key indexes, data distribution, or data skewing. Snowflake also eliminates the need to manage table partitions or sub-partitions. Because compute is separate from storage in Snowflake's architecture, massively parallel processing (MPP) compute nodes do not rely on the data being distributed ahead of time.

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 you of this task. Because Snowflake does not use indexes, you don't have to migrate your indexes and partitions.

Data file management

Snowflake manages all database objects behind the scenes without the use of data files. Your storage grows and shrinks instantly as you add or delete data. This also means Snowflake has no concept of tablespace, so you can execute your table-create DDL statements without any storage clause.

Workload management

Workload management is unnecessary in a Snowflake environment due to its multi-cluster architecture, which allows you to create separate compute clusters for your disparate workloads to avoid resource contention completely. Additionally, Snowflake scales compute up or down automatically based on demand.

Statistics collection

Snowflake automatically captures statistics, relieving DBAs from the mundane tasks of setting up jobs to collect statistics for performance tuning. Snowflake performs these tasks in real time as data is loaded, so the metadata is always up to date. You no longer have to add new tables to the process as your data grows.

Capacity planning

With Snowflake, you pay for only what you use. Snowflake is a SaaS product that is further enhanced for efficiency with per-second, usage-based pricing. Under this model, Snowflake also offers further cost reductions for customers who want to pre-purchase usage. On the flip side, with capacity planning for on-premises Oracle engineered systems (Exadata and Supercluster), you run the risk of over- or under-configuring your system, especially with high-availability (HA) configurations such as Oracle Real Application Clusters (RAC), where multiple servers are required. Even with Oracle Database Cloud Services, you have a similar capacity planning risk because compute and storage are fixed per instance. If you need more capacity within an Oracle cloud service, you must buy in predefined increments. Snowflake's elastic storage and compute architecture eliminates this risk, so you can save money and avoid the time previously spent on extensive planning.

Disaster recovery

The Oracle database has several disaster recovery scenarios, such as Active Data Guard and the Zero Data Loss Recovery Appliance (ZDLRA). Many of them require you to buy additional hardware, software licenses, and network infrastructure. But Snowflake leverages the built-in features of its cloud infrastructure providers. By design, Snowflake is automatically synced across multiple availability zones. No work is required on your part to establish this.

Separate dev/test environment

With Oracle, to perform development and testing, you need additional servers, which means an additional capital outlay for the hardware plus the time to configure it. But with Snowflake, you can simply create another database in your account and configure it for any purpose, 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 at no additional cost. Using Snowflake's standard role-based access control, you issue simple grant statements to determine who can access each of these separate databases. With Oracle, you would have to endure the painstaking process of exporting your production data from one system and then importing it to your dev or test environment.

Snowflake has developed a modern cloud data platform that automatically and instantly scales compute and storage in a way not possible with Oracle, whether the current Oracle system is on-premises or hosted in the cloud.

MIGRATING YOUR EXISTING ORACLE WAREHOUSE

To successfully migrate your enterprise data warehouse to Snowflake Cloud Data Platform, develop and follow a plan that includes the steps presented in this section.

MOVING YOUR DATA MODEL

As a starting point for your migration, you need to move your database objects, including databases, tables, views, and sequences, from Oracle to Snowflake. 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 Oracle database or schema on the target Snowflake system before you migrate data. Your choice of which objects to move depends on the scope of your initial migration.

After deciding which objects to move, choose a method for moving your data model from Oracle to Snowflake. The following sections outline three different methods.

Using a data modeling tool

If your data warehouse design is stored in a data modeling tool such as Oracle SQL Developer Data Modeler, you can generate the DDL needed to rebuild your existing database objects. Since Snowflake uses standard ANSI SQL, you simply need to pick ANSI SQL rather than Oracle as the output scripting dialect. The majority of your Oracle DDL will execute in Snowflake without change. Keep in mind that Snowflake is self-tuning and has a unique architecture. You won't need to generate code for any indexes, partitions, or storage clauses that you may have needed in an Oracle database. You need only basic DDL, such as CREATE TABLE, CREATE VIEW, and CREATE SEQUENCE. After you have these scripts, you can log into your Snowflake account to execute them through the UI or the command-line tool SnowSQL.

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. Follow this link to learn how to connect Oracle SQL Developer Data Modeler to Snowflake.

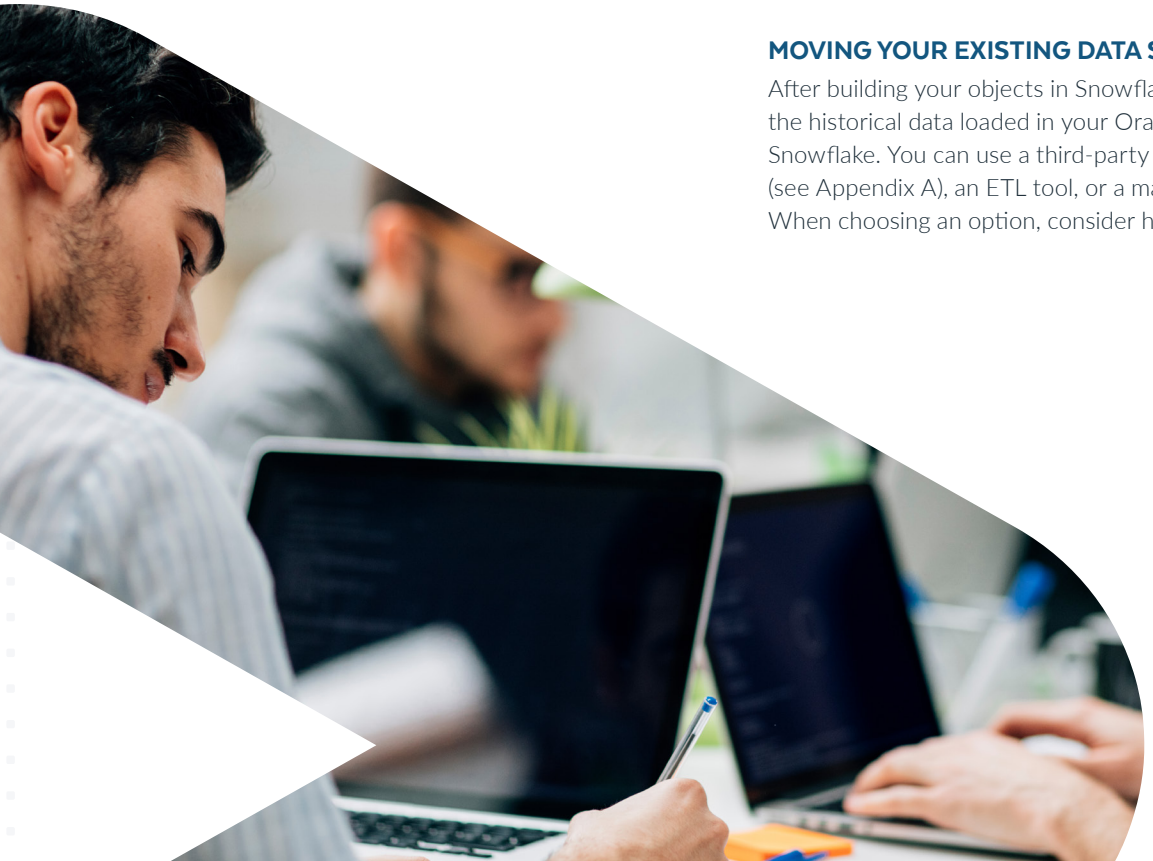
Using existing DDL scripts

If you don't have a data modeling tool, you can begin with the most recent version of your existing DDL scripts (in a version control system). Edit these scripts to remove code for extraneous features and options not needed in Snowflake, such as indexes, table-space assignments, and other storage or distribution-related clauses. Depending on the data types you used in Oracle, 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 B.

Creating new DDL scripts

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



Depending on the data types in your Oracle 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 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, which lets you automate the extract process and execute the move iteratively. Plus, you will save time editing the script later. 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

After building your objects in Snowflake, move the historical data loaded in your Oracle system to Snowflake. You can use a third-party migration tool (see Appendix A), an ETL tool, or a manual process. When choosing an option, consider how much data you have to move. For example, to move tens or hundreds of terabytes up to a few petabytes of data, a practical approach is 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 hundreds of petabytes or even exabytes of data, AWS Snowmobile or Azure Data Box are available 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. Use one of the many methods available to the Oracle database such as PL/SQL routines using utl_file or SQLcl to pump the data out to the desired format. Then upload these files using the PUT command into an internal or external Amazon S3 staging bucket. These files should 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 S3, you can begin loading the data into your table in Snowflake using the COPY command. See more details about the COPY command in the Snowflake 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, Oracle uses some Oracle-specific extensions, so you need to watch out for a few constructs. Some examples include the use of FETCH FIRST x ROWS ONLY. See Appendix C for details and suggested translations.

Another common change relates to formatting of date constants used for comparisons in predicates. For example:

In Oracle it looks like this:

where my_date_datatype > '01-JAN-17';

Or

where to_char(my_date_datatype, 'YYYY-MM- DD') > '2017-01-01';

Or

where my_date_datatype > to_date('2017-01-01', 'YY-MM-DD');

In Snowflake it looks like this:

where my_date_datatype > cast('2017-01-01' as date)

Alternatively in Snowflake you can also use this form:

where my_date_datatype > '2017-01-01'::date

Migrating BI tools

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

This should be simple 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 the Snowflake website to see if your tools are supported. 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 Oracle 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 configuration from the Oracle Database Resource Manager, the I/O Manager, and any Instance Caging settings will give you a good idea of how to set up Snowflake compute resources. However, you'll need to consider the optimal way to distribute these in Snowflake. As a starting point, create a separate compute resource for each workload, then size the resource according to resources required to meet the SLA for that workload. 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 to automatically suspend and resume when needed.
- How much compute will you need to meet that window? Use that estimate to determine the appropriate compute resource size.
- How many concurrent connections will this workload need? If you normally experience

bottlenecks, you may want to use the Snowflake multi-cluster resource for those use cases to enable automatic scale out during peak workloads.

- Think about dedicating at least one large compute resource 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 compute resource.

MOVING THE DATA PIPELINE AND ETL PROCESSES

Snowflake is optimized for an ELT approach. However, Snowflake supports many traditional ETL and data integration solutions. We recommend a basic migration of all existing data pipelines and ETL processes to minimize the impact on your project unless you are planning to significantly enhance or modify them. Because testing and data validation are key elements of any changes to the data pipeline, maintaining these processes will reduce the need for extensive validation.

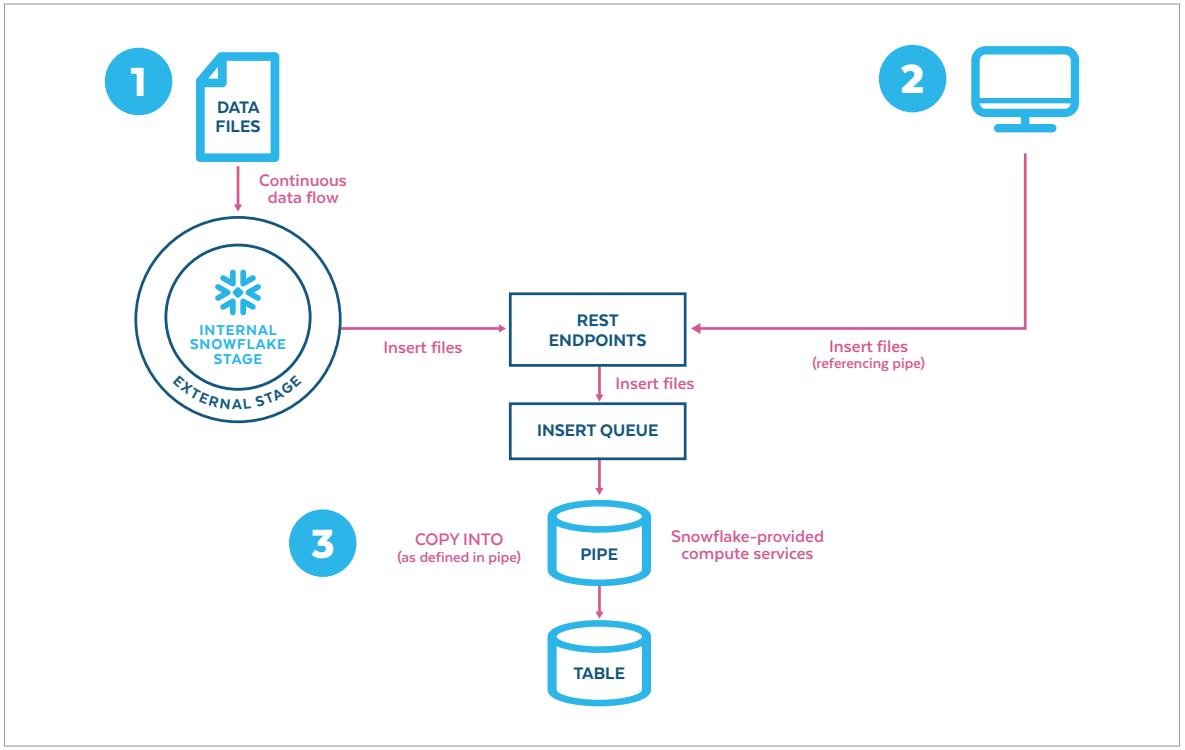
Snowflake has worked diligently to ensure that the migration of processes running on traditional ETL platforms is as painless as possible. Native connectors for tools such as Talend and Informatica make the process quick and easy.

Run the data pipeline in both Snowflake and Oracle during the initial migration to simplify the validation process by enabling a quick comparison of the results from the two systems. When you're sure queries running against Snowflake are producing identical results as queries from Oracle, you can be confident that the migration maintained data quality, and 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 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 data sets and perform transformations at any scale.

Snowpipe enables loading data from files as soon as they're available in a stage. This means you can load data from files in micro-batches, making it available to users within minutes instead of manually executing COPY statements on a schedule to load larger batches.

A pipe is a named, first-class, Snowflake object that contains a COPY statement used by the Snowpipe REST service. The COPY statement identifies the source location of the data files (a named stage) and a target table. Snowflake supports structured and semi-structured data, including semi-structured data types such as JSON and Avro.



CUT OVER

After you migrate your data model, data, loads, and reporting to Snowflake, plan your switch from Oracle to Snowflake. Here are the steps:

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

The chart to the right outlines high-level considerations for migrating to Snowflake.

HOW TO MIGRATE TO SNOWFLAKE
Maintain existing schema
Set up role-based security
Automate processes to accelerate migration • Automated schema migration • Automated tagging and metadata collection • User and role sync with Directory Service • Change data capture to reduce daily data volumes

WHY TO MIGRATE TO SNOWFLAKE
Easy schema migration
Pay-as-you-go model
Bulk loading to improve data loading
Automated query optimization
Scalable compute to power data transformation
Role-based security

REDUCE EFFORT BY ELIMINATING THE NEED FOR	
Up-front capacity planning	Separate systems for disaster recovery
Data distribution and sort keys	Distinct test/dev environments
Database indexes	Data marts and data silos
Workload management	Database statistics management

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, proofs of concept, and migration projects.

Our global and regional solution partners also have extensive experience performing proofs of concept and platform migrations. They offer services ranging from high-level architectural recommendations to manual code conversions. Many 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 Snowflake Cloud Data Platform, so you can reap the full benefits quickly. To find out more, contact the Snowflake sales team or visit Snowflake's Customer Community Lodge.

APPENDIX A: MIGRATION TOOLS

Snowflake ecosystem partners may have offerings that can help with your migration from Oracle to Snowflake. For more information, or to engage these partners, contact your Snowflake representative.

RED PILL ANALYTICS

The Avalanche Program from Red Pill Analytics leverages Snowflake replication and analytics technology partners to deliver a rapid migration to Snowflake.

WHERESCAPE® AUTOMATION FOR SNOWFLAKE

WhereScape automation for Snowflake accelerates your ability to start using Snowflake for new and ongoing cloud data infrastructure projects. WhereScape can help you migrate data warehouses, data vaults, data lakes, and data marts by:

- Automating workflow
- Providing built-in best practices for common methodologies such as 3NF and Data Vault 2.0
- Supporting continuous integration
- Generating SQL code
- Supporting iterative design and development
- Creating technical level documentation

WIPRO

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

TATA CONSULTANCY SERVICES (TCS)

TCS has also developed accelerators and utilities to accelerate the migration from Oracle to Snowflake. They have trained experts and experience with large migrations.

ANALYTIX DS

Mapping Manager

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

JumpStart Automation Services

- Data vault integration
- Code generation automation
- ETL platform migration

APPENDIX B:
DATA TYPE CONVERSION TABLE

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

ORACLE DATA TYPE	NOTES	SNOWFLAKE DATA TYPE	NOTES
NUMBER		BYTEINT	
NUMBER		SMALLINT	
NUMBER		INTEGER	
NUMBER		BIGINT	
NUMBER		DECIMAL	
FLOAT		FLOAT	
NUMBER		NUMERIC	
FLOAT		CHAR	Up to 16 MB
VARCHAR2/NVARCHAR2		VARCHAR	Up to 16 MB
CHAR(n)		CHAR VARYING(n)	
FLOAT		REAL	
DATE		DATE	
TIMESTAMP	Only HH:MI:SS	TIME	
TIMESTAMP with TIMEZONE		TIMESTAMP_LTZ	aliases: TIMESTAMPLTZ TIMESTAMP WITH LOCAL TIME ZONE
BINARY_FLOAT/BINARY_ DOUBLE		BINARY	Up to 8 MB
BINARY_FLOAT/BINARY_ DOUBLE		VARBINARY	
CLOB / VARCHAR2		VARIANT	
CLOB / VARCHAR2		ARRAY	

APPENDIX C: SQL CONSIDERATIONS

Below are examples of some of the changes you may need to make to your Oracle SQL queries to have them run correctly in Snowflake. Note that this is not an exhaustive list.

TOP-N

Selects the first n rows of the result set. This non-ANSI syntax is usually used with an ORDER BY clause. (Sometimes it is used without an ORDER BY clause to get a data sample.) Snowflake supports the ANSI FETCH keyword syntax and the non-ANSI LIMIT keyword in place of FETCH:

ORACLE	SNOWFLAKE
SELECT sales_name FROM sales_commission WHERE campaign = 'C1' ORDER BY commission_value DESC FETCH FIRST 10 ROWS ONLY;	SELECT sales_name FROM sales_commission WHERE campaign = 'C1' ORDER BY commission_value DESC FETCH 10;

JSON -
JAVASCRIPT OBJECT NOTATION
TABLE NAME: CI2
COLUMN NAME: BITCOIN_JSON

```
{
   "CODE": "USD",
   "NAME": "US DOLLAR",
   "RATE": 5645.879004
}
```

ORACLE	SNOWFLAKE
SELECT JSON_VALUE (a.bitcoin_ json, $.code) FROM CI2 a;	SELECT bitcoin_json:code

UDF — USER DEFINED FUNCTION

ORACLE	SNOWFLAKE

CREATE or REPLACE FUNCTION
add5 (n in number)
  RETURN number
IS
  sum_number number;
  BEGIN
  SELECT 5 + n
  INTO sum_number
  FROM dual;
RETURN (sum_number);
end;
	CREATE or REPLACE FUNCTION
	add5 (n number)
	RETURNS number
	as 'n + 5';

Native non-ANSI, Oracle-specific function	CREATE or REPLACE FUNCTION instr(mainStr string, srchStr string) RETURNS int as $$charindex(SRCHSTR, MAINSTR)$$;
Native non-ANSI, Oracle-specific function	CREATE or REPLACE FUNCTION months_between(date1 date, date2 date) RETURNS float as $$cast(datediff(month, date2::timestamp, date1::timestamp) as float)$$;

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**.