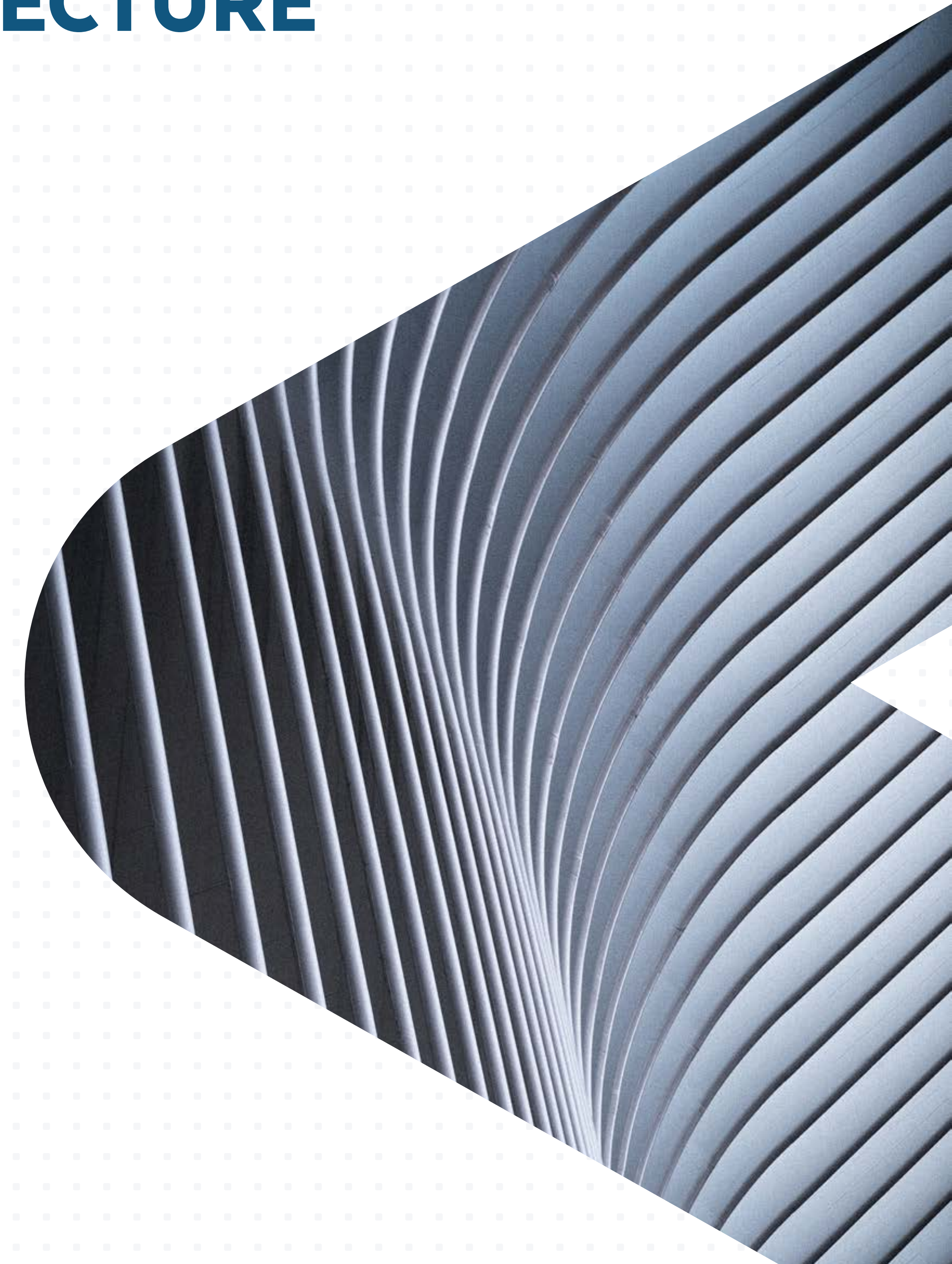
ARCHITECTURE
GUIDE

snowflake®

FUNDAMENTAL DATA WAREHOUSE ARCHITECTURE

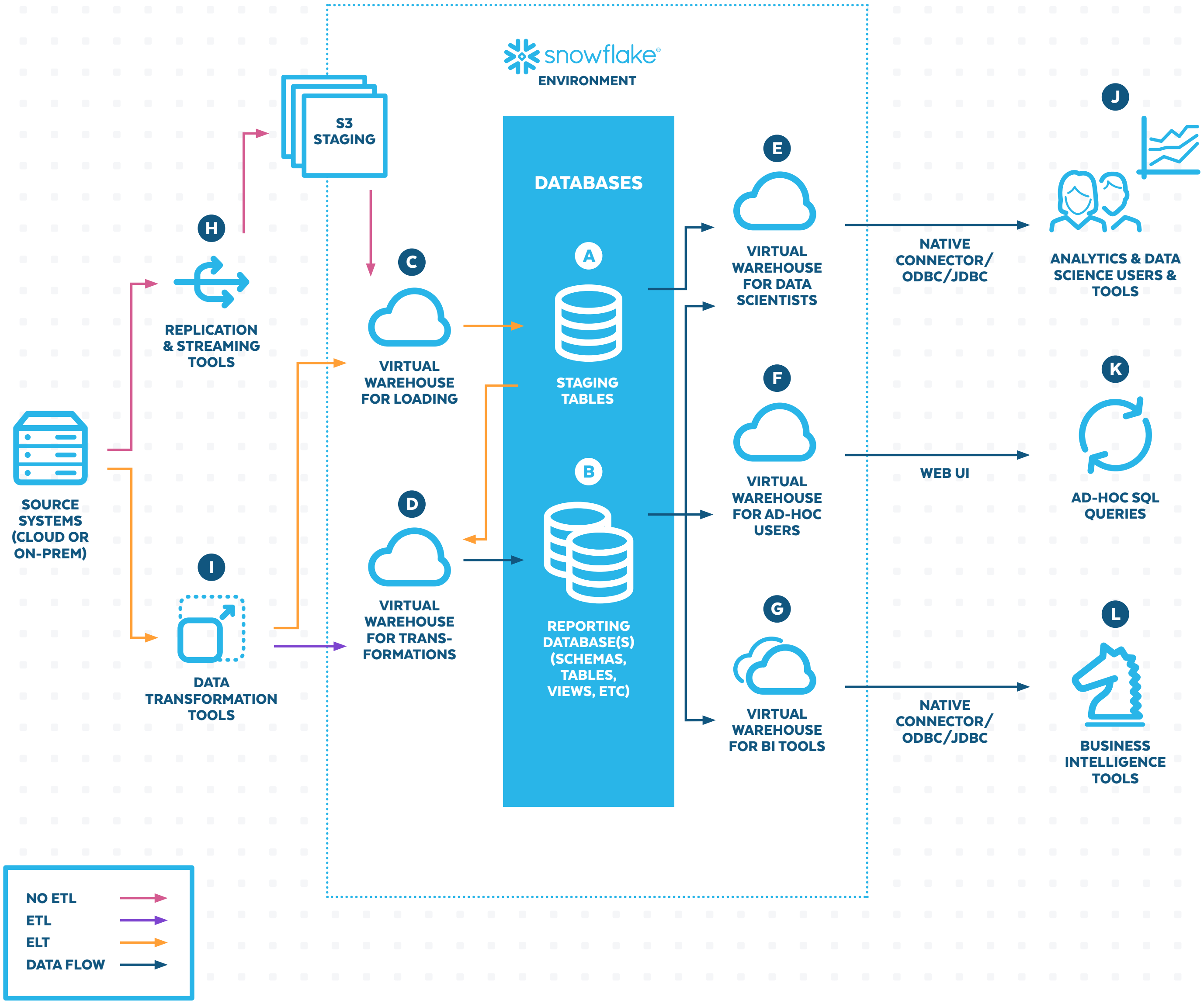


CHAMPION
GUIDES

INTRODUCTION

This reference guide details one of the architectures customers have successfully implemented with Snowflake. If you do not already have a preferred architectural approach to your data warehouse, we recommend you consider this approach or other Snowflake customer configurations as a starting point.

A FUNDAMENTAL SNOWFLAKE DATA WAREHOUSE ARCHITECTURE


snowflake
ENVIRONMENT
S3 STAGING
DATABASES
A
STAGING TABLES
B
REPORTING DATABASE(S) (SCHEMAS, TABLES, VIEWS, ETC)
H
REPLICATION & STREAMING TOOLS
SOURCE SYSTEMS (CLOUD OR ON-PREM)
I
DATA TRANSFORMATION TOOLS
C
VIRTUAL WAREHOUSE FOR LOADING
D
VIRTUAL WAREHOUSE FOR TRANS-FORMATIONS
E
VIRTUAL WAREHOUSE FOR DATA SCIENTISTS
F
VIRTUAL WAREHOUSE FOR AD-HOC USERS
G
VIRTUAL WAREHOUSE FOR BI TOOLS
NATIVE CONNECTOR/ ODBC/JDBC
WEB UI
NATIVE CONNECTOR/ ODBC/JDBC
J
ANALYTICS & DATA SCIENCE USERS & TOOLS
K
AD-HOC SQL QUERIES
L
BUSINESS INTELLIGENCE TOOLS
NO ETL
ETL
ELT
DATA FLOW

CREATING YOUR DATABASES

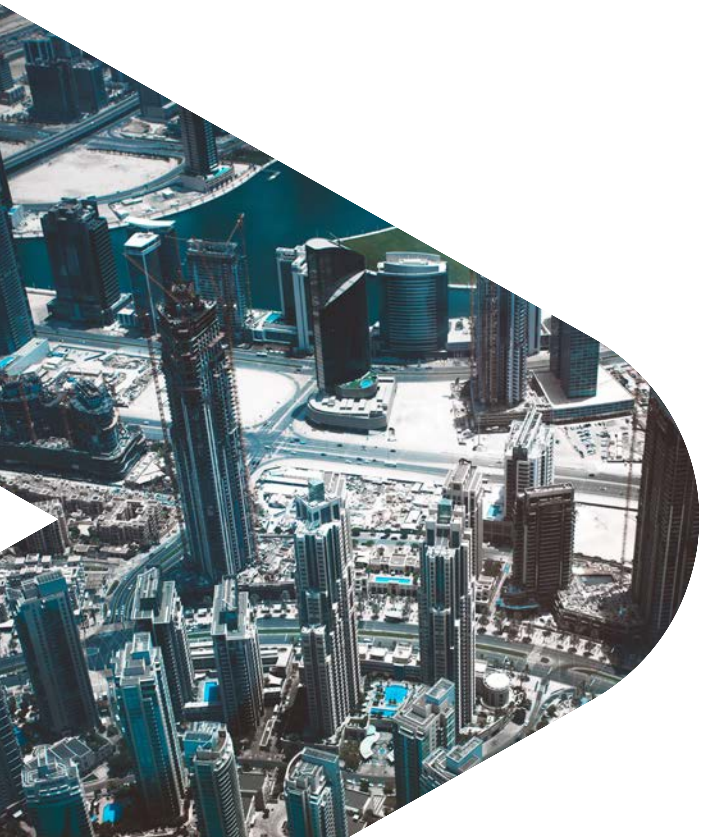
DATABASES

For this architectural approach, you'll need to create two or more databases within your Snowflake account. The next sections outline what those databases are and how you can use them.

A STAGING DATABASE

This database will hold staging tables into which you'll load all your source data. These tables allow you to rapidly load raw data into Snowflake so you can examine, profile or query it before pushing it further downstream to business users and various enterprise BI tools.

At this point, you'll have a few options to consider based on your specific use case. First, you can store the staging tables in one schema or multiple schemas. Multiple schemas make sense if you want to separate your staging tables by source system. Another reason to use multiple schemas would be to segment the inbound data based on security considerations such as data privacy.



Depending on your preferred methodology, you can define these staging tables as transient or persistent (see Snowflake's documentation for details). Transient staging tables are appropriate if you want to move all your incoming data into permanent data warehouse tables for reporting. Then, you can purge the staging tables. On the flip side, persistent tables make more sense if you want to take a data lake approach and keep your data in raw format for data discovery or data science purposes. Persistent tables also make sense if you're capturing new data but are not ready to store the data in your permanent data warehouse.

B REPORTING DATABASE(S)

You'll need at least one other defined database to hold the permanent tables for your data warehouse. This layer of the architecture provides users and query tools with a single, integrated, historicized, semantically consistent and analysis-friendly set of tables and views used to explore all your enterprise data in one place. Snowflake is schema agnostic. Therefore, you can design these tables using traditional, historicized, third normal form (3NF), star schema, data vault or any hybrid schema.

Because Snowflake allows cross-database joins, you also have the option to split your data warehouse tables into multiple databases. There are several scenarios where you might consider doing so:

1. You may want to separate data by logical subject areas if you have groups within your organization that deal with distinctly separate data subjects.
2. If some of your data is highly sensitive, such as PHI or credit card data, which requires rigorous security measures, you may want to segregate that data into a separate database.

VIRTUAL WAREHOUSES

In Snowflake, a “virtual warehouse” is a compute cluster used to process queries. A virtual warehouse can access any data stored in Snowflake because Snowflake’s architecture decouples data storage from compute. All virtual warehouses operate independently of each other, eliminating resource contention or performance impact across all user groups and other data warehouse activities. We recommend creating several virtual warehouses to take full advantage of this aspect of Snowflake’s unique architecture.

C VIRTUAL WAREHOUSE FOR LOADING

This warehouse provides the compute resources for loading your source data into your staging tables. Having a separate virtual warehouse for loading will allow you to run load processes anytime, or even continuously, in isolation from the resources your data warehouse end users will need. You can load this data directly from your S3 buckets using Snowflake’s COPY command or via a process controlled by your ETL or ELT tool of choice.

D VIRTUAL WAREHOUSE FOR TRANSFORMATIONS

The transformation warehouse provides the compute resources for formatting, transforming and aggregating the raw data in your staging tables as you move the data into your permanent data warehouse tables. You can execute these processes from either a standard ETL/ELT tool or a SQL script.

E VIRTUAL WAREHOUSE FOR DATA SCIENTISTS

The data science warehouse enables your data scientists to perform their exploration and test their theories without impacting other data warehouse users.

F VIRTUAL WAREHOUSE FOR AD-HOC USERS

A virtual warehouse for ad-hoc users provides compute resources needed to perform special purpose or other ad-hoc queries with SQL to better understand the data. This enables users to query the data warehouse without impacting other data warehouse users.

G VIRTUAL WAREHOUSE(S) FOR BI TOOLS

For business intelligence (BI) activity, this warehouse provides the compute resources to support all the work done via standard BI, visualization or dashboard tools. If there are many users and tools, you may want to set up several virtual warehouses to support the different groups and workloads. Again, this helps isolate these queries, so they do not impact performance for other users.

DATA LOADING & TRANSFORMATION

There are several options for moving your data into Snowflake. Which approach you choose is dependent on the type of data, nature of the source systems and the existing tools already in your environment. You also have the option of using multiple methods or tools, depending on your requirements.

H REPLICATION & STREAMING TOOLS

The simplest way to load data into Snowflake is by replicating your raw data to flat files on S3. Then, use the COPY command to load those flat files into tables in your staging database. You can enable this from your private S3 bucket, outside Snowflake or from the S3 staging area provided within your Snowflake account. You can also move the data into S3 using various streaming platforms such as Kafka.

I DATA TRANSFORMATION TOOLS

Another alternative is to use standard ETL or ELT tools to move, and possibly transform, data from your source systems into your Snowflake staging database. From there, you can use these same tools to move data into your data warehouse database. Because Snowflake offers both JDBC and ODBC connectors, you can use nearly any off-the-shelf tool. Some vendors also provide optimized, native connectors for Snowflake.

QUERY AND ACCESS TOOLS

Once your data is in Snowflake and ready to use, you'll need to make it available to your users to perform their analysis. Again, because of Snowflake's open interface and native SQL capabilities, you have a wide choice of tools and techniques to access the data in your Snowflake data warehouse.

J DATA SCIENCE AND ADVANCED ANALYTICS

You can use R, Python, SPARK or any number of tools to query, explore and analyze your data. We offer open source connectors for all of these tools.

K AD-HOC SQL QUERIES

For those conversant with SQL, you always have the option of viewing and analyzing your data with SQL via our web user interface. Snowflake offers a fully functional SQL interface, including many analytic functions. In addition, you have the ability to query semi-structured data using our SQL extensions.

L BUSINESS INTELLIGENCE TOOLS

Snowflake supports accessing all your data with industry-standard business intelligence and visualization tools via ODBC and JDBC. As with the ETL/ELT tools, many Snowflake partners offer native connectors to Snowflake for a more optimized experience. Again, the choice is yours.

ABOUT SNOWFLAKE

Snowflake is the only data warehouse built for the cloud, enabling the data-driven enterprise with instant elasticity, secure data sharing and per-second pricing, across multiple clouds. Snowflake combines the power of data warehousing, the flexibility of big data platforms and the elasticity of the cloud at a fraction of the cost of traditional solutions. Snowflake: Your data, no limits. Find out more at **snowflake.com**