snowflake®

DATA MANAGEMENT AND THE DATA LAKE:
ADVANTAGES OF A SINGLE PLATFORM APPROACH

Now you don't have to choose between a data warehouse and data lake

CHAMPION GUIDES

TABLE OF CONTENTS

INTRODUCTION

Many enterprises struggle to formulate a data architecture that supports their current and future analytics needs. Should they build a data lake or a data warehouse—or some combination of the two? What are the advantages and disadvantages of these two architectural paradigms, and which types of analytics applications are best served by each of them?

This ebook discusses the unique qualities of data warehouses and data lakes, reviews their key differences, and explains how you can establish a single cloud data platform that supports both approaches, simplifying your overall data environment.

THE CHALLENGE

Data is one of the most critical assets of most organizations. Almost every organization builds a data architecture to store, prepare, manage, and analyze its data, generally with distinct options for data warehouses and data lakes. Technology experts like the structure provided by a data warehouse for relational data, but they also like the flexibility of a data lake, which can include semi-structured and unstructured data types. Embracing both data management paradigms maximizes flexibility, but it forces the organization to choose one or the other architecture method as the central data repository.

Snowflake is challenging this conventional thinking by enabling organizations to create data lakes and data warehouses based on one unified data platform.

ARCHITECTURAL DISTINCTIONS

A *data lake* is a centralized repository where you can store structured, semi-structured, and unstructured data on any scale, at a lower cost compared to most traditional data warehouse and RDBMS platforms. These versatile data repositories address the "three Vs" of big data: volume, variety, and velocity. You can store raw data "as is" without having to structure the data before it is loaded. Data lakes offer flexibility on the front end because they can quickly ingest many types of data—including binary and semi-structured data. However, consumers can't immediately run analytics on this raw data. The data first has to go through a data engineering cycle of transformation, cleansing, and standardization. In addition, the business must create a data model to govern its use and orchestrate data queries.

A *data warehouse* is a highly modeled repository, designed primarily for structured data. Typically, the data stored in a warehouse is indexed, loaded into tables, and carefully related to all of the other data in the warehouse. Data warehouses tend to be highly standardized, and the data is cleansed before it is loaded.

IT organizations typically govern data warehouses, which often causes friction with the business teams that depend on the data for operational analytics. Business consumers want flexible, self-service access to the warehouse for business intelligence, data science, and data visualization projects. They don't want to be saddled with elaborate data management practices to use, maintain, update, or govern the data in the warehouse.

As an adjunct to the data warehouse, data marts meet the demand for various subject-specific workloads, as well as the consequent scalability and concurrency demands of end consumers.

Based on these three popular analytic paradigms, for the last 30 years a typical enterprise data architecture has looked something like *Figure 1* on the following page.

Traditional data architecture

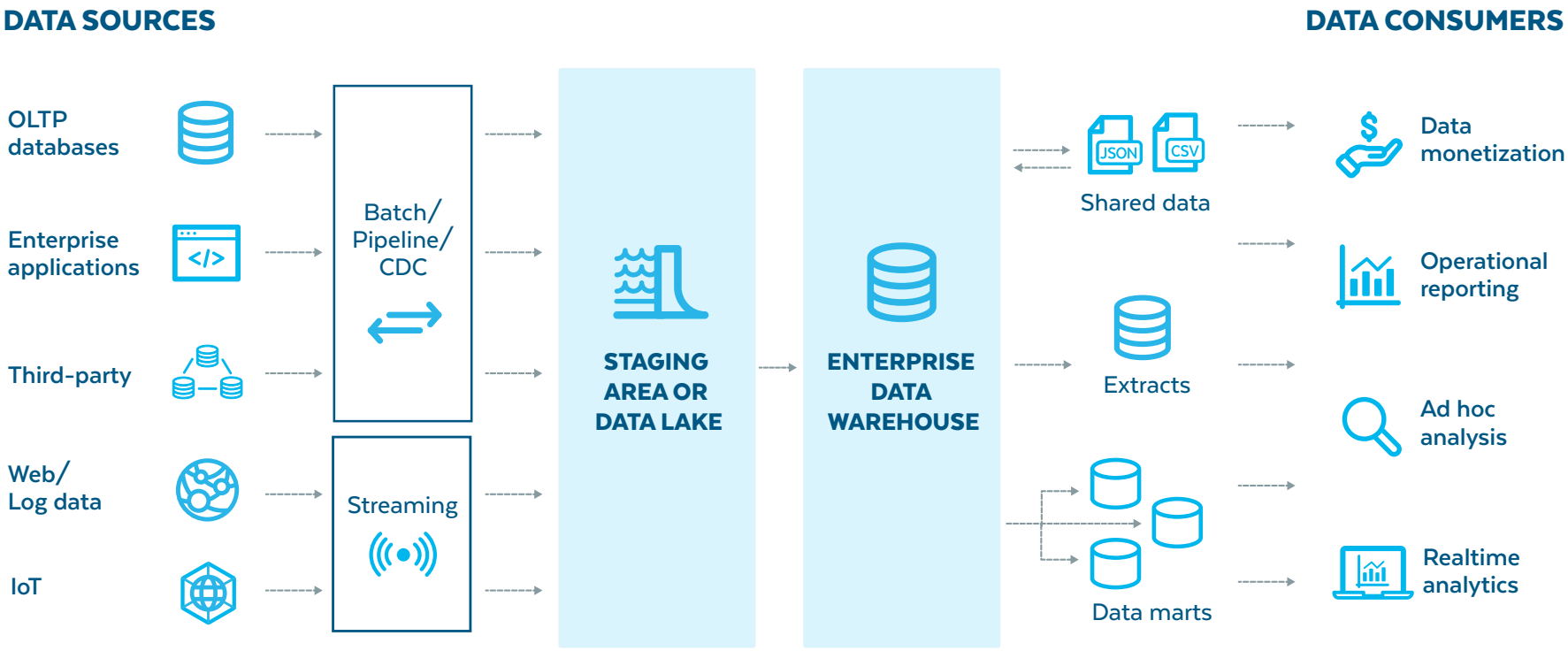


Figure 1: A typical data lake and data lake architecture reference diagram.

HISTORICAL PRECEDENTS

Traditional data warehouses were not designed to store large volumes of data, particularly when the data is a mix of tables and JSON. To support a broad range of use cases, you need a way to store and analyze a mix of data varieties. While data lakes play a role in the data landscape, it might not be necessary to implement them separately from data warehouse systems.

Consider an architecture in which enterprise data is grouped into four logical data zones, as shown in *Figure 2* on the following page.

The left side of the diagram shows the types of data sources that produce data at a very high velocity, volume, and variety. This raw data is staged through four distinct processes, or zones:

- **Data in the raw zone must be prepared and cleansed before it can be consumed. This requires transformation logic or scripts to capture the raw data, clean up inconsistencies, and standardize the format.**
- **The cleansed, standardized data is then stored in the conformed zone, where a set of procedures transforms the data and integrates it with other data sets.**
- **The resulting data is the reference database, including business mappings, business logic, and data hierarchies.**
- **The end result is clean, modeled data, ready for consumption by business users, analysts, data scientists, and any other authorized user who wishes to run reports, visualize the data through dashboards, issue ad hoc queries, or obtain analytic insights.**

INTRODUCING A DATA PLATFORM FOR THE FUTURE

It's hard to gain value from your data when each new analytic application creates a data silo. Silos increase cost, reduce performance, and lead to never-ending data quality problems. To accelerate the process of obtaining business-ready analytics, many organizations seek a common data platform that can support both types of systems—an architecture that simplifies the development and deployment of both data lakes and data warehouses.

Snowflake Cloud Data Platform is engineered to work natively with both structured and semi-structured data at unlimited scale. It includes data warehouses, data lakes, and many types of data engineering and data science applications, providing a foundation for running any workload from a single service.

Logical data zones

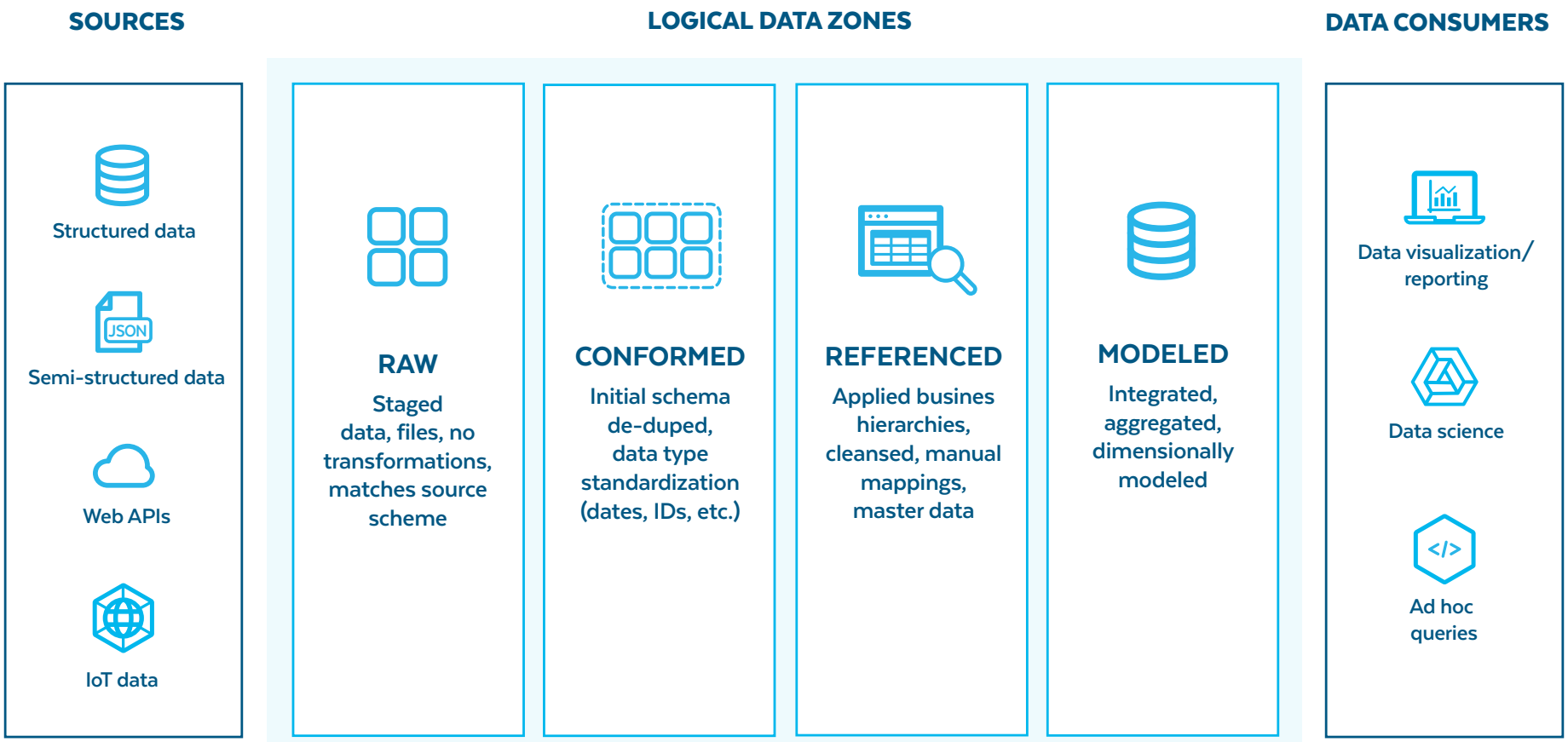


Figure 2: Logical data zones for enterprise data.

SOLVING COMMON BUSINESS AND TECHNOLOGY CHALLENGES

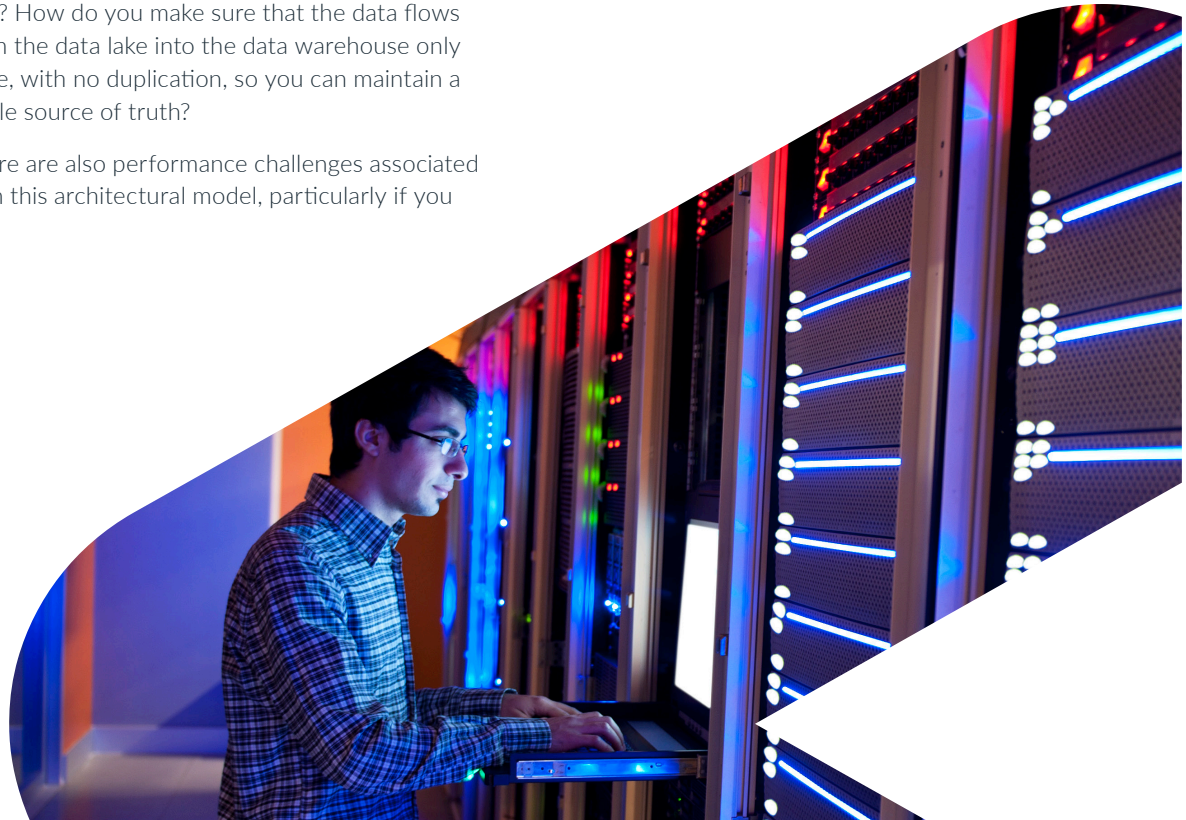
Figure 2 depicts a popular architecture for a modern data repository. However, many organizations have questions about the optimal way to move their data through these four logical data zones, as well as how to prepare data for consumption via a data warehouse or data lake. For example, how do you create a data pipeline to efficiently and reliably transmit data through these various zones? How do you make sure that the data flows only once and isn't duplicated? How do you enforce reliable data quality and data correction procedures?

Figure 3, on the following page, offers more detail about how Snowflake Cloud Data Platform moves data from the raw data zone to the conformed data zone and, finally, into the modeled data zone. Data can be generated via Kafka or a similar messaging pipeline and persisted into a cloud bucket.

From the cloud bucket, Apache Spark or a similar transformation engine converts the data into an optimized columnar format, such as Parquet, and persists the data into the conformed data zone. From there, other scripts create models and push the data into the modeled data zone. Some organizations may push the modeled data back into the lake with Apache Hive, or use a SQL query engine such as Presto or Athena to read the data directly from the data lake itself. In other instances, the modeled data can be loaded into a data warehouse. Either way, the data is ready for consumption by data analysts, data scientists, and other users.

Although these diagrams make the process look relatively straightforward, in practice it isn't easy to operationalize these business processes in a secure, managed, and governed way. For example, if you push the modeled data into a data warehouse, how do you sync that data with the source data in a data lake? How do you make sure that the data flows from the data lake into the data warehouse only once, with no duplication, so you can maintain a single source of truth?

There are also performance challenges associated with this architectural model, particularly if you use Hive or some other technology to persist data in and out of the data lake. Other challenges arise as the surrounding data architecture evolves. For example, if your data arises from IoT devices, how do you add data when new sensors are added? How do you get new data or a new schema of that data into the data lake? How do you modify your data pipeline scripts to push data into the data warehouse with complete integrity?



CREATING AN EXTENSIBLE DATA ARCHITECTURE WITH SNOWFLAKE

Astute organizations constantly evaluate their data architectures and examine new technologies to solve these difficult problems. They need a modern data platform born in the cloud—a platform that delivers the best of modern data warehousing, the best of data lakes, and much more. With Snowflake, you no longer have to decide whether to create a data lake or a data warehouse. You can establish a single platform that supports both.

Figure 3 (to the right) illustrates this architecture. On the left, data is pulled in from multiple sources, then transformed through the raw, conformed, and modeled data zones into a cohesive architecture within the primary data lake. Consumers can easily access the data from the modeled zone using their choice of business intelligence tools. You can also build a rich application layer—particularly if your cloud data platform includes standard connectors to Python, Spark, JDBC, ODBC, and others.

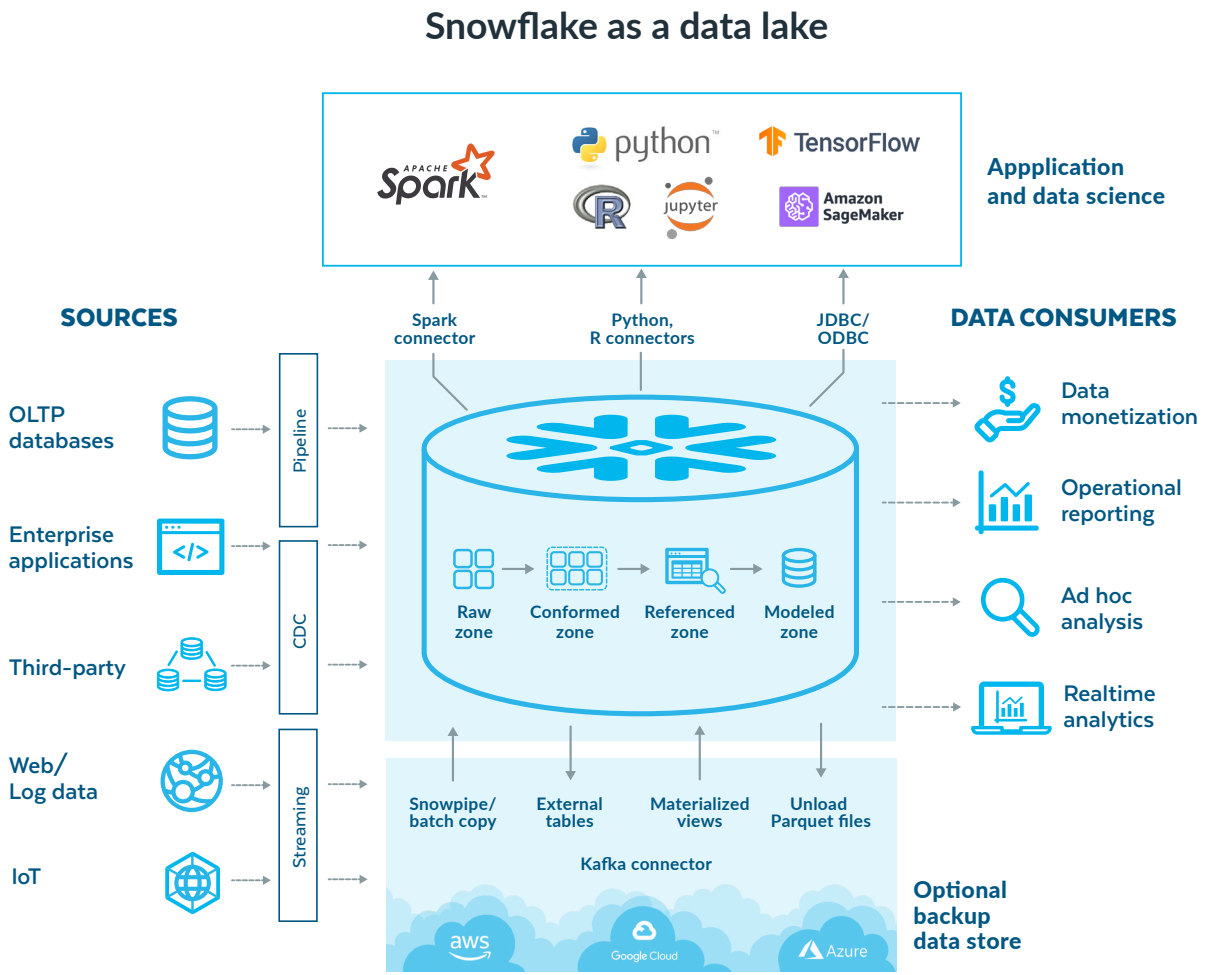


Figure 3: Snowflake offers a single platform that supports a rich and extensible data lake.

You can feed data into the data lake using multiple methods including the Snowflake Snowpipe pipeline, batch copy procedures, a Kafka connector, or a utility provided by one of Snowflake's ETL partners. You can store the data in a raw data file or convert it into a variant data type and place it in a Snowflake table. Within this unique data lake platform, even raw data can be indexed, including semi-structured data and deeply nested JSON files. This makes the process of searching, querying, analyzing, and visualizing the data in the data lake much easier and much faster.

AUTOMATING OPERATIONS WITH STREAMS AND TASKS

Snowflake includes a serverless ingestion service that can asynchronously load data into your cloud storage environment. Standard connectors and adapters allow you to easily ingest event streams from Kafka and other messaging systems, while streams and tasks make it easy to schedule data loads for SQL jobs. You can ingest data into the repository and share it with a global base of employees, customers, and partners without setting up ETL pipelines or you can exchange data among regions. The platform automatically transforms data into the type and structure required for each target table. For example, Snowflake's Apache Kafka connector lets you continuously stream JSON records for storage and analysis.

Snowflake uses table streams to monitor raw data sets as they are added, updated, and deleted. Transformation logic can sort error data into a separate table. Standard transformations cleanse the remaining data and store it in enriched tables. Another set of streams and tasks can store the data as model tables and views.

IMPROVING PERFORMANCE FOR EXTERNAL TABLES WITH MATERIALIZED VIEWS

Snowflake materialized views allow you to calculate aggregates out of the conformed data. This can be fully automated as data is ingested into a cloud bucket via Snowpipe. Snowflake lets you set up these procedures as automated services that run in the background, so as soon as the data comes into the modeled zone, authorized users can consume the data.

To reduce costs, you can store data that is accessed infrequently in an external cloud bucket such as Amazon S3, Azure Blob Storage, or Google Cloud Platform, rather than ingesting it into a Snowflake table. You can use SQL tools to query data in these external tables, eliminating the need to build an ETL layer or orchestration pipeline.

USING SNOWFLAKE TO AUGMENT AN EXISTING DATA LAKE

So far, this ebook has focused on creating new data lakes from scratch. However, if you have an existing data lake, you can take advantage of Snowflake's unique architecture to improve it. Snowflake complements and extends your existing data lake architecture in two ways:

- **As an ETL engine to process raw data and generate conformed data**
- **As a query engine to query modeled data**

USING SNOWFLAKE AS AN ETL ENGINE

Most of the cost and effort of creating data lakes is associated with transforming the data and maintaining the data pipelines. Apache Spark handles most transformations in existing data lakes. However, Spark engineers are in short supply, which makes it difficult to build, test, and debug new data pipelines, spin up Spark clusters, and maintain the transformations. Snowflake simplifies the creation of ETL pipelines, as well as the transformation of data from raw to conformed.

The Snowflake ETL engine converts data from the raw zone into the conformed zone. External tables can point directly to the files in the data lake, and you can build streams and tasks over these external tables to automate operations and improve performance. These streams keep track of changes in the data lake.

For example, when new files are added or deleted, a stream can run the necessary transformations to cleanse the data. You can store conformed data in an enriched internal table and export it back into Parquet files in your data lake using Snowflake's partition and load functionality. If your existing data lake is your source of truth and you can't ingest data into Snowflake, you can use an external table with materialized views to improve query performance. These services are trouble-free; they work automatically in the background. (See Figure 4).

USING SNOWFLAKE AS A QUERY ENGINE FOR YOUR DATA LAKE

In addition, you can run the Snowflake query engine (*see Figure 5*) against these external tables and join them with internal tables. Almost all of the functionality that works on internal tables in Snowflake also works on external tables. Creating materialized views on these external tables makes the performance of SQL queries almost comparable to queries on Snowflake internal tables. This allows you to establish your data lake as your single source of truth and use Snowflake as a query engine. Three primary capabilities within Snowflake make this possible: external tables, streams over external tables, and materialized views.

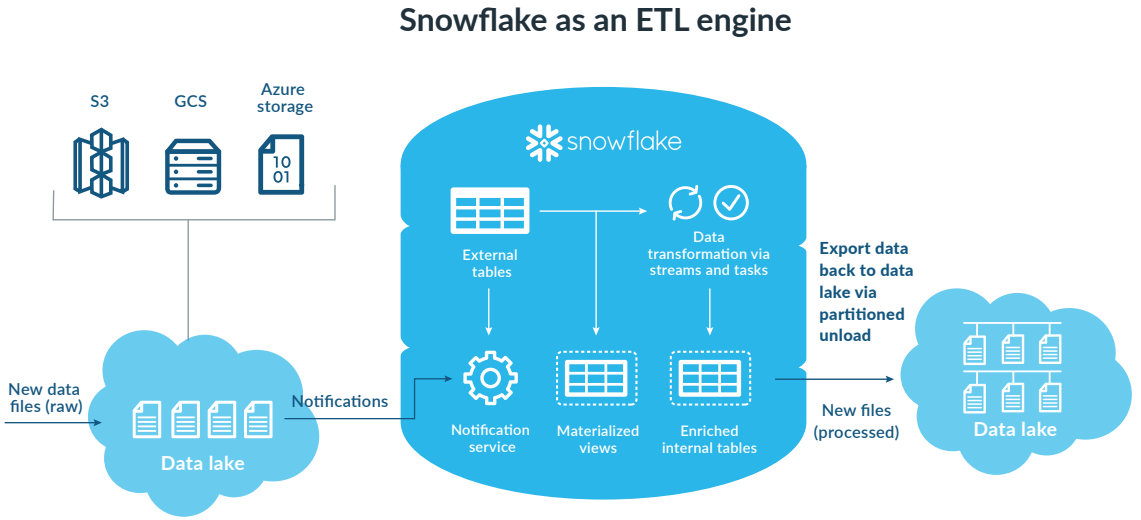


Figure 4: Snowflake offers a better way to load and manage data via external blob storage or internal tables in the data lake.

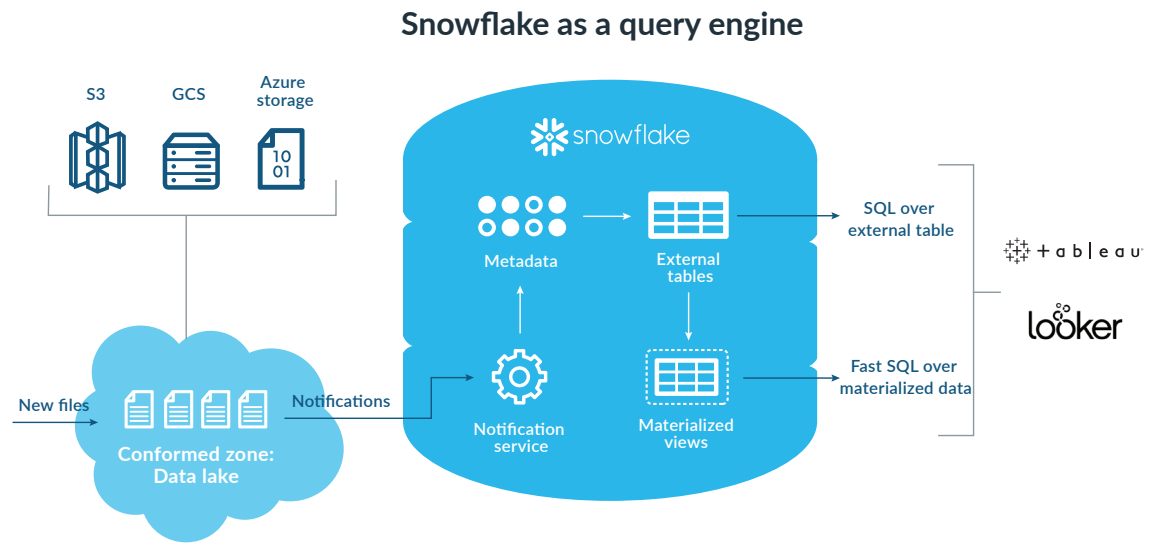


Figure 5: Improve query performance with Snowflake Cloud Data Platform.

THE VALUE OF A UNIVERSAL DATA PLATFORM

It's time to stop thinking about your data in terms of existing types of systems, such as legacy data warehouses, data marts, and data lakes. Snowflake has dramatically changed the data engineering landscape by eliminating the need to develop, deploy, and maintain these distinct data systems. For the first time, one data platform can support all three of these analytic paradigms, making it easier to manage structured and semi-structured data, such as tables and JSON, in a cohesive way.

Snowflake is the only data platform built for the cloud. Most other "cloud" data warehouses were designed more than 20 years ago and have not been architected to leverage the scalability of the cloud. Only Snowflake offers a complete enterprise data platform that includes data governance, ACID-compliant transactions, and live data sharing across multiple types of clouds. (See Figure 6).

Figure 6: Snowflake provides you one cohesive, enterprise cloud data platform for all your data.

CONCLUSION

In the past, companies created data warehouses and separate physical data marts that allowed multiple groups of stakeholders and data consumers to store and analyze data from enterprise applications. Next came data lakes, driven by steady advancements in data science and a desire to store and examine non-relational data types. Today, organizations commonly mix data processing technologies and analytics techniques, but each of these methods provides limited insight from a unique slice of data.

For the first time, you can manage all of your enterprise data in one highly scalable and fully elastic platform. Snowflake Cloud Data Platform offers a near-zero management foundation for running any workload, including data warehouses, data lakes, and many types of data engineering and data science applications. It includes a unified repository powered by a comprehensive layer of services for security, governance, data sharing, metadata management, and transaction management, bringing consistency to all types of analytic projects.

ABOUT SNOWFLAKE

The 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 types of data workloads, including a single platform for developing modern data applications. Snowflake: Data without limits. Find out more at **snowflake.com**.