PROCESSING MODERN DATA PIPELINES

Achieve Performance, Scalability, and Efficiency for Data Engineering and Data Science Workloads

snowflake

WHITE PAPER

INTRODUCTION

Data is central to how we run our businesses, establish our institutions, and manage our personal and professional lives. Nearly every interaction generates data—whether from software applications, social media connections, mobile communications, or many types of digital services. Multiply those interactions by a growing number of connected people, devices, and interaction points, and the scale is overwhelming—and growing rapidly every day.

While all this data holds tremendous potential, it is often difficult to mobilize for specific purposes. Until recently, businesses had to be selective about which data they collected and stored. Compute and storage resources were expensive and sometimes difficult to scale. Today, the rise of affordable and elastic cloud services has enabled new data management options—and necessitated new requirements for building data pipelines to capture all this data and put it to work. You can accumulate years of historical data and gradually uncover patterns and insights. You can stream data continuously to power up-to-the-minute analytics.

However, not all data pipelines can satisfy today's business demands, so choose carefully when you design your architecture and select data platform and processing capabilities. Many pipelines add unnecessary complexity to business intelligence (BI) and data science activities due to limitations within the underlying systems used to store and process data. For example, in some cases you may have to convert raw data to Parquet simply because that's the format your system requires. Or perhaps your processing systems cannot handle semi-structured data such as JSON in its native format.

This white paper describes the technical challenges that arise when building modern data pipelines and explains how these challenges can be solved with Snowflake by automating performance with near-zero maintenance. The Snowflake platform offers native support for multiple data types and can accommodate a wide range of data engineering workloads to build continuous data pipelines, support data transformation for different data workers, operationalize machine learning, share curated data sets, and other tasks. Snowflake customers benefit from a powerful data processing engine that is architecturally decoupled from the storage layer, yet deeply integrated with it for optimized performance and pipeline execution.

REVIEWING BASIC DATA PIPELINE CAPABILITIES

Data pipelines automate the transfer of data from place to place and transform that data into specific formats for certain types of analysis. They make this possible by performing a few basic tasks:

- Capturing data in its raw or native form
- Ingesting data into a data warehouse, data lake, or other type of data store, housed on premises or in the cloud
- Transforming data into a business-ready format that is accessible to users and applications
- Augmenting data to make it more valuable for the organization

As shown in Figure 1, in a modern data pipeline, data moves through several stages as it is transformed from a raw state to a modeled state. To augment the value of data, data must be deduplicated, standardized, mapped, integrated, and cleansed to prepare it for immediate use.

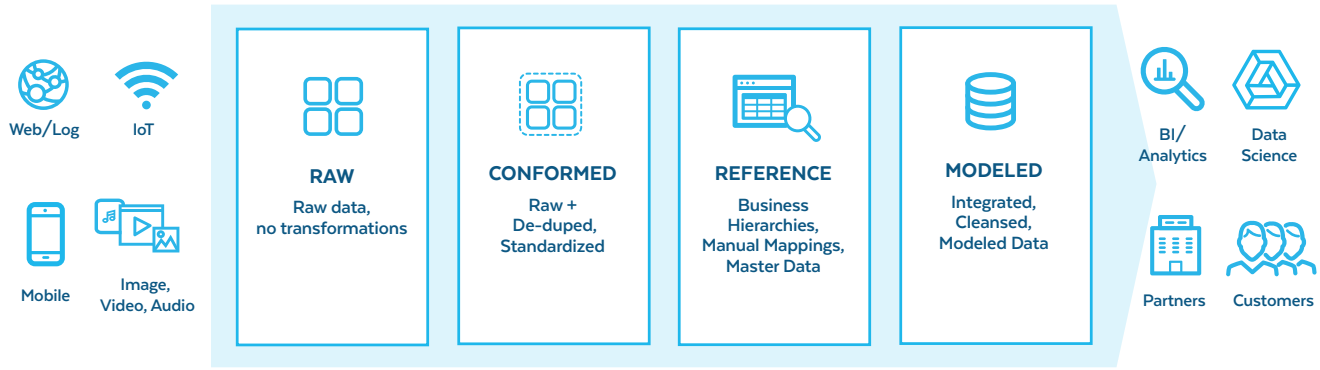

Figure 1: Data pipelines connect raw data from original source systems and transform it to a business-ready state for data consumers and the workloads they depend on.

Data sources typically produce raw data in row- or document-oriented formats such as JSON or CSV. If your downstream apps require a different format such as Parquet, ORC, or Avro, you may need to duplicate the data before you can use it, which increases costs and complicates data governance activities. Additionally, due to the nature of today's exploratory analytics endeavors, you may spend money to convert data that you never even use.

Be wary of data storage architectures that require you to store data in a unique physical representation for performance reasons. If your data platform can natively ingest these popular data formats, you don't need a pipeline that exists only to handle conversions.

THE RISE OF MODERN DATA PIPELINES

Today's modern data pipelines have arisen in response to three important paradigm shifts in the software industry.

Paradigm shift #1: On-premises ETL to cloud-driven ELT

As described in *Cloud Data Engineering for Dummies*, yesterday's data pipelines were designed to accommodate predictable, slow-moving, and easily categorized data from business applications such as enterprise resource planning (ERP), supply chain management (SCM), and customer relationship management (CRM) systems. The destination was a pre-modeled data warehouse, where data was loaded into highly structured tables from which it could be readily accessed via Structured Query Language (SQL) tools.

These pipelines depended on extract, transform, and load (ETL) workflows, where data was processed and transformed outside of the "target" or destination system. These traditional ETL operations used a separate processing engine, which involved unnecessary data movement and tended to be slow. Furthermore, these engines weren't designed to accommodate schemaless, semi-structured formats—a death blow in today's world of continuous and streaming data sources.

To accommodate these newer forms of data and enable more timely analytics, modern data integration workloads leverage the processing power of target data platforms in the cloud. In these more modern architectures, data pipelines are designed to extract and load the data first, and then transform it once the data reaches its destination (ELT rather than ETL). Cloud analytics platforms are great at handling these transformations, causing a decisive shift from ETL to ELT.

UNDERSTANDING VALUE-ADDED TRANSFORMATIONS

Data pipeline transformations are often very similar to transformations performed when executing analytic queries. Common operations in these transformations include the following:

- Filters narrow your processing to the relevant data, as well as remove irrelevant or incorrect data.
- Joins and unions correlate one data set to another.
- Projections apply business calculations to fields in your data or provide additional structure to make it semantically meaningful to business users.
- Groupings consolidate fine-grained data to create coarser-grained business metrics.
- Aggregations roll up data according to various dimensions such as by time period, location, and product type.

These value-added transformations are so common that it often makes sense to execute them within the data pipeline workflow, rather than at runtime when users issue analytics queries. A good processing engine can handle these transformations just as easily in the "upstream" data pipeline as it can during "downstream" analytics, since essentially the same relational algebra operations are performed.

With traditional ETL architectures on premises, ETL jobs contend for resources with other workloads running on the same infrastructure. In contrast, modern ELT architectures move these transformation workloads to the cloud, enabling superior scalability and elasticity. Not every cloud solution is created equally though; some may hide an on-premises nature with a cloud veneer. The scalable and elastic ones have the capability to separate each workload and scale among readily available compute and storage resources. Modern data pipelines use the limitless processing resources of the cloud to perform rich transformations on the data platform underlying the data pipelines. As a result, you don't need to prepare data before you load it. You can load the raw data and transform it later, once the requirements are understood. Finally, a fully managed cloud service relieves you of many of the management, maintenance, and provisioning challenges associated with on-premises infrastructure.

Keeping your raw data and your transformed data in the same system gives you more options later on. For example, you can join the raw data with the transformed data and perform a backfill operation to update the target tables, for instance, when you add new fields.

Paradigm shift #2: Batch processing to continuous processing

Many businesses produce data continuously, but they may only make updates available for analytics at periodic intervals, such as weekly, daily, or hourly, typically via bulk or batch data-update processes. This ensures good compression and optimal file sizes, but it also introduces latency—the time between when data is born and when it is available for analysis. Latency delays time to insight, leading to lost value and missed opportunities.

A more natural way to ingest new data into your technology stack is to pull it in continuously, as data is born. Continuous ingestion requires a streaming data pipeline that makes new data available almost immediately, such as a few seconds or minutes after the data is generated.

Until recently, enabling these continuous analysis scenarios was expensive. Because they were constrained by a fixed set of on-premises resources, IT pros had to strike the right balance between maintaining fresh data for analytics, and not overspending on hardware and software infrastructure. They also had to work around other processing jobs that used those same computing resources, which often meant running data pipeline jobs during off hours, commonly called "maintenance windows."

Waiting to upload batch data during scheduled maintenance windows delays business results. Data becomes stale, and in some cases it is no longer relevant. Loading data continuously, by contrast, not only enables more timely insights but entirely new business models. For example, vehicles emit data from hundreds of sensors every time the driver starts the ignition, turns the wheel, brakes, and accelerates—along with passively generated data on the conditions and performance of onboard equipment. If this data can be captured and put to work immediately, it can be used to monitor fleet performance, predict maintenance requirements, and satisfy service level agreements. Rather than waiting for these vehicles to come into a depot to download the information, a streaming ingestion service can collect it all the time, allowing for quicker decision-making.

Paradigm shift #3: Consolidation of systems for structured, semi-structured, and unstructured data

Database schemas are constantly evolving. Modifications at the application level, such as the introduction of semi-structured JSON data to augment structured relational data, necessitate changes to the underlying database schema. Application developers may need to constantly interface with database administrators to accommodate these changes, which delays the release of new features.

To overcome these limitations, modern data platforms include built-in pipelines that can seamlessly ingest and consolidate all types of data so it can be housed in one centralized repository without altering the schema. However, legacy systems might stand in the way of this goal. Data warehouses typically ingest and store structured data, defined by a relational database schema. Data lakes store many types of data in raw and native forms, including semi-structured data and some types of unstructured data. It may be challenging to integrate end-to-end data processing across these different systems and services.

Even cloud-based systems can impose limitations, since each public cloud provider has unique services for data storage, data transformations, and analytics.

Cloud providers may also require separate services to store unstructured and structured data. Implementing end-to-end data pipelines requires you to understand and rationalize the differences among all of these cloud environments, including different semantics and unique management procedures. If you can't rationalize these differences you will be forced to maintain separate data silos, which complicates data governance.

FULFILLING TODAY'S EXPECTATIONS FOR THE NEW DATA PIPELINE PARADIGMS

As organizations attempt to take advantage of these paradigm shifts, fulfilling their expectations has become progressively more difficult. Common obstacles include the following:

- Data of every shape and size is being generated, and it often ends up sequestered in siloed databases, data lakes, and data marts—both in the cloud and on premises. Different applications have different expectations for the delivery of data, forcing data engineers to master new types of languages and tools.
- Organizations with legacy data pipeline architectures operate with a fixed set of hardware and software resources, which invariably leads to reliability and performance issues. As data pipeline workloads increase, system administrators may need to juggle the priorities of other workloads contending for those same resources.
- Complex data integration and orchestration tools may be necessary to create new data pipelines and hand-coded interfaces. Many IT teams find that they spend too much time managing these pipelines and the associated infrastructure. Rather than focusing on advising the business on how to get the most value out of organizational data, these highly paid technology professionals are mired in technical issues related to capacity planning, performance tuning, and concurrency handling.

Organizations are growing impatient with these obstacles. As data becomes progressively more important to day-to-day business operations, their expectations are rising. Yet analysts and decision-makers often have to contend with incomplete or stale data, and there could be a general lack of faith in analytics outputs.

Modern enterprises need to cope with different integration styles—from batch to streaming to everything in between. Many of these organizations require continuous data processing capabilities to deliver rapid business insights. At a minimum, customers demand scalable and elastic data processing software that can accommodate all types of data, accessible through usage-based business models.

A DATA PLATFORM FOR THE NEW PARADIGMS

The Snowflake platform includes data pipeline capabilities as part of the basic service, minimizing the need for complex infrastructure tasks. It accommodates batch and continuous data loading equally well. In addition, you can land all your data irrespective of its shape—structured, semi-structured, or unstructured (in preview)—in one central repository and leverage various languages—including SQL, Scala, Java, Python (in preview), and Javascript—to transform, prepare, and query the data to support different requirements.

Snowflake also includes capabilities to consolidate, cleanse, transform, and access your data, as well as to securely share data with internal and external data consumers. It natively supports popular data formats, not just structured relational data, but also semi-structured data (such as JSON, Avro, ORC, and Parquet) and unstructured data in the form of files—all accessible via familiar SQL constructs.

One benefit of being able to access semi-structured data via SQL is the ability to create data pipelines without having to first define the data schema. This is especially advantageous for JSON and XML data, which may change often in cloud application sources. Snowflake maintains the original shape of the raw data but also transparently applies highly optimized storage techniques to the data so that analytics and data transformations perform exceptionally well.

Because you can ingest raw data directly into Snowflake without losing the fidelity of the raw representation, there is no need to create a pipeline merely to transform data into a different format. Snowflake performs these transformations automatically, putting your data in an optimal form for high-performance analytics while remembering its original shape. This can lead to significant storage and compute cost savings since you avoid data duplication and the overhead required to maintain the various copies of the data.

In contrast to conventional data lake solutions, Snowflake gives you a consistent way to work with all your data, even when you are amassing various types of files of different sizes and formats, each placed in different folders and file structures. With Snowflake, you don't need to think about your file formats, files sizes, folder structures, and partitioning strategies. You don't need to use a unique file management approach, you are not bound to a single way for how you can partition your data, and you don't have to compromise between the requirements of your storage system and a separate processing engine. Snowflake lets you interact with unstructured and semi-structured data just as you interact with structured data in database tables, and it enforces consistent security and governance across all data sets. This provides the benefit that you can run data pipelines end to end on the same platform—starting with unstructured data stored as files in Snowflake and then promoting semi-structured and structured properties from the unstructured file in your Snowflake data pipelines, which lead into the downstream data assets that serve your business users. Running and managing your data pipelines end to end in the same system provides significant governance benefits with consistent concepts and semantics all across.

Understanding Snowflake's unique architecture and powerful processing engine

Snowflake's data processing engine is built for speed and scalability with an architecture that takes full advantage of public cloud resources. Its high-performance query processor is optimized for analytics, such as when you need to visualize data and generate reports, as well as for processing and transforming data during pipeline operations. The exceptional performance that users enjoy for analytics can be pushed into the pipeline to create high-value data and aggregated data sets, yielding ultrafast response times when users issue queries.

Pre-computing results makes sense when common queries are issued repeatedly, such as for popular dashboards and reports. With Snowflake, data engineers can easily make the choice of how much of this pre-computation work to perform in the data pipelines initially, versus how much to perform at analytics runtime.

Thanks to its unique multi-cluster shared data architecture, in which compute and storage resources are decoupled but tightly integrated, Snowflake can easily handle data transformations at scale. Each data engineering workload receives dedicated resources, eliminating contention when multiple workloads run at the same time. If data volume suddenly spikes, and you need more compute and storage capacity to absorb the load, Snowflake scales to instantly supply the necessary resources to your data pipeline. You can set it to scale automatically or choose to manually add compute resources with a few clicks. You don't have to anticipate data volumes in advance. There is no need to spend time on capacity planning, cluster management, and other low-level maintenance tasks. On top of that, you are paying only for what you consume.

Snowflake mitigates resource contention when multiple data pipelines run in tandem. Large processing jobs don't take away resources from other processing jobs running on the same data—not just data engineering workloads, but also BI, data science, data apps, and other workloads that use Snowflake. Each workload is isolated from the others, so there is no need to schedule data engineering jobs during off hours.

Snowflake makes it easy to accommodate real-time and streaming sources. It handles structured, semi-structured, and unstructured data in a cohesive way. Just as importantly, Snowflake decouples the database schema from the data that these workloads generate, allowing new types of data to fit into existing Snowflake tables. This design accommodates evolving needs and simplifies the management of databases and files.

Storing all your data in Snowflake also unlocks the benefits of the data economy for your business. Data in Snowflake can easily participate in the Data Cloud, making it easy for you to collaborate over shared data assets across departments in your organization or across organizations with your business partners and customers, regardless of what cloud provider they use or region they are based in.

DRILLING DOWN INTO SNOWFLAKE'S UNIQUE CAPABILITIES

The Snowflake data processing engine offers unique capabilities for aggregating and transforming data, both during data pipeline operations and during query processing operations for analytics. Some of the highlights are described below.

Micro-partitioning

Many databases improve performance by placing data into partitions—contiguous units of storage that can be manipulated independently. Traditional data warehouses rely on static partitioning of large tables to enable better scaling. As shown in Figure 2, Snowflake maps data into micro-partitions: groups of rows in tables, organized in a columnar fashion. To ensure efficient query processing, each micro-partition also includes metadata such as the range of values for each of the columns, the number of distinct values, and additional properties. (Micro-partitions also serve as the basis for pruning, as discussed below.)

Traditional database management systems, by contrast, require you to define a partitioning scheme upfront—often before you load your data. This requires you to anticipate early in the design phase how your system will be used—which is not always conducive to data science and other advanced analytics. While these legacy database management systems rely on database administrators (DBAs) to formulate table partitions, clustering keys, and other details, Snowflake creates these partitions automatically, organizing data for optimal performance. You don't have to think about what indexes to create, how to partition data, or which clustering keys to use. As users submit queries, Snowflake uses selective query predicates to home in on only the relevant partitions, yielding a huge performance advantage.

Pruning

The metadata that Snowflake maintains enables precise pruning to eliminate unnecessary micro-partitions. This technology works by isolating the parts

DATE	PRODUCT	CUSTOMER	AMOUNT
Feb 14	Boots	Frank	$150
Feb 14	Boots	Benoit	$150
Feb 14	Skis	Thierry	$300
Feb 14	Snowboard	Mike	$250
Feb 15	Boots	Chris D	$150
Feb 15	Skis	Denise	$600
Feb 15	Snowboard	Shelly	$250
Feb 16	Boots	Rob	$150
Feb 16	Skis	Sunny	$600
Feb 16	Snowboard	Chris K	$250
Feb 16	Snowboard	Greg	$750
Feb 16	Snowboard	Matt	$750

Figure 2: Snowflake maps data into micro-partitions for efficient query processing.

of the data set that must be scanned to return each query result. Snowflake touches only the partitions necessary to return the answer set. For example, if a large table contains one year of historical data, sorted into date and hour columns, a query targeting a particular hour might need to scan only 1/8,760th of the micro-partitions in the table, since there are 8,760 hours in a year. Snowflake won't look at all the other hours that are not relevant.

Further, pruning is applied to the resulting set of qualifying micro-partitions to eliminate the data of columns that are not required to process the query. This performance improvement is crucial for analytics scenarios that often operate only on a strict subset of the columns present in a table, and that's why Snowflake's micro-partitions use a columnar storage representation internally.

Materialized views

The Snowflake Materialized Views feature allows data engineers to express pertinent data transformations within their data pipelines to define how data is physically stored in materialized views: pre-computed data sets stored for later use. Pre-aggregating data into materialized views can improve the performance of queries that frequently use the same subquery results, such as within commonly used dashboards and reports. Because the data is pre-computed, querying a materialized view is faster than executing a query against the base table of the view. Materialized views can dramatically speed up aggregation, projection, and selection operations for common query patterns. All materialized views are automatically and transparently maintained by Snowflake.

Other database management systems run sophisticated view maintenance procedures to keep materialized views in sync with the base tables from which they are derived. View maintenance is performed either as part of the same process that also updates the base table or out of band later. Running the view maintenance as part of the base table processing keeps the materialized view up to date and business results accurate. However, it makes the base table update more expensive. Due to that processing overhead, the alternative approach runs view maintenance only at periodic intervals, leading to stale data and outdated business results. The Snowflake Materialized Views feature contains up-to-date results, thanks to a query processor that maintains materialized views out of band but fetches residual data from the base tables as necessary.

Here's how it works: Snowflake runs a background process that incrementally updates the materialized views as base tables are updated. If queries are submitted in the interim, Snowflake figures out which data in the base table hasn't been propagated into the materialized view representation. It retrieves data from the materialized view, determines which updates are missing, grabs those missing bits from the base table, and stitches it all together to return an accurate query result. Snowflake maintains a fine-grained sequence of versions for every table to accelerate these query activities.

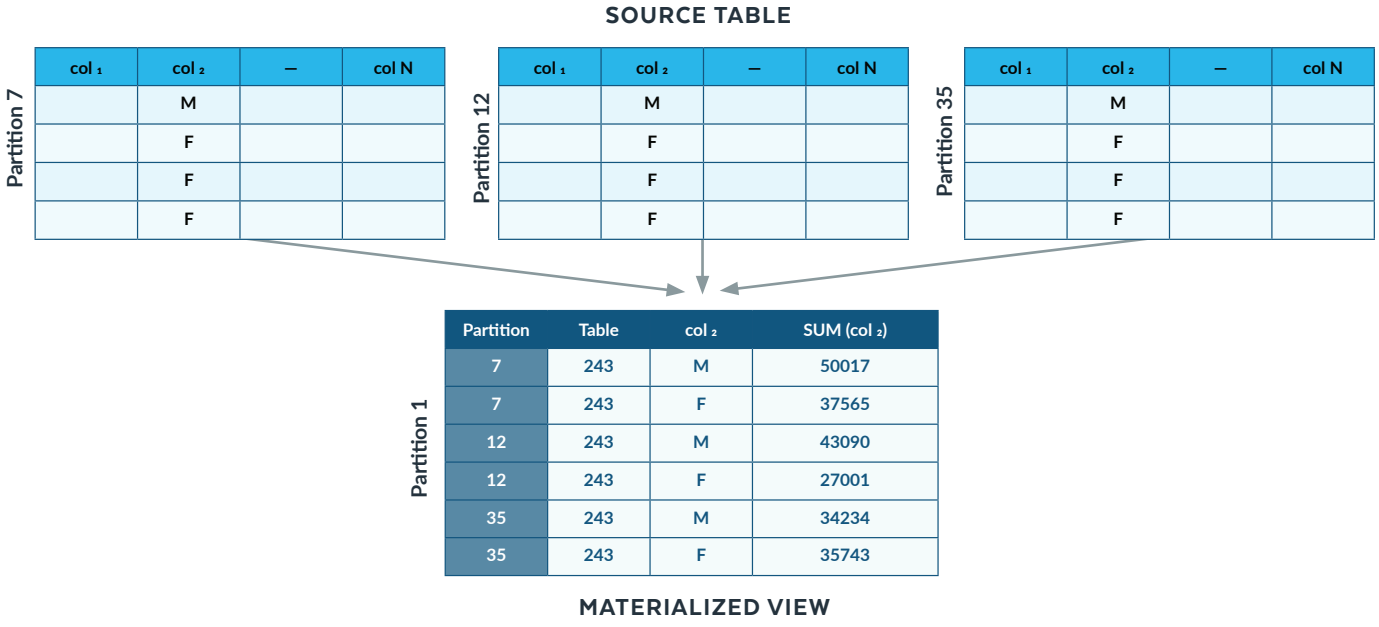

Figure 3: Snowflake enables you to store frequently used projections and aggregations in materialized views for faster queries.

Serverless ingestion

Today's cloud service providers provision, scale, and manage the infrastructure required to host your data and run your business applications, but they typically require customers to manually provision and manage cloud resources such as virtual machines. However, cloud services are called *serverless* when customers are insulated from which servers are needed for each operation and the manual management of infrastructure and resources. Serverless cloud services scale dynamically and automatically, increasing or decreasing capacity instantly as the load on the service fluctuates.

Snowpipe, Snowflake's serverless ingestion service, automatically manages capacity for your data pipeline as data ingestion loads change over time. Similarly, Snowflake uses a serverless approach for maintaining materialized views. Behind the scenes, Snowflake optimizes its serverless services to maximize performance while minimizing costs. Underutilized resources are released right away.

These elastic pipeline capabilities are particularly important when you are managing batch and micro-batch uploads, as well as for accommodating streaming data. For example, as soon as an item is scanned at a retail point-of-sale location, the data is available to inform restocking decisions. As soon as a bank suspects a fraudulent credit card transaction, the cardholder is queried to authorize or deny the charge.

Transactional guarantees

Data integrity is the bedrock of data management systems, but with some of today's complex data engineering scenarios, transactional guarantees can be difficult to achieve. For example, a data pipeline may consume data from multiple staging tables, apply transformations to the data, and then write the results to multiple target tables in the cloud. If there is a mishap with any of the associated servers or network connections, some parts of these operations may not be completed, resulting in data corruption that can be difficult to reconcile. To protect against these failure scenarios, data engineers need to write and maintain complex error detection and compensation logic in their data pipelines or manually stitch up and correct data after a failure. In some cases, this might not be possible and the business is stuck with wrong data and wrong business results after such a failure. None of these outcomes are acceptable for businesses today.

Snowflake instead is a fully ACID-compliant system that fully supports transactional guarantees no matter whether a transaction touches a single table or updates multiple tables with several terabytes of data stored in them. A particularly powerful example in the context of data pipelines is Snowflake's proprietary streams technology (also referred to as "table streams"), which combines highly efficient change detection with transactional guarantees—even when multiple tables are involved in the same transaction. These "all or nothing" properties ensure that if any part of a transaction is interrupted, all data will be rolled back to its previous state—even when those transactions involve bulk uploads and lots of data.

Schema-on-Read

With Snowflake, you can load semi-structured data such as JSON, XML, Parquet, ORC, and Avro directly into a relational table, and then query the data with a SQL statement and join it to other structured data—without worrying about the schema of that data. In contrast, most conventional data warehouses and big data environments require you to first load this type of data to a Hadoop or NoSQL platform, shred it, and then load it into columns in a relational database. Until you perform these extra steps, you can't run SQL queries or use a BI tool against that data.

Snowflake keeps track of self-describing schemas using the VARIANT data type, which allows you to load semi-structured data as is into a column in a relational table. Then, Snowflake Schema-on-Read lets you create view definitions over this raw data so you can query it right away. Data engineers can create data pipelines that load the data directly into a relational table, so business users can instantly derive meaning from the schema in the view definition. They can query the components and join them to columns in other tables—with no coding or shredding required to prepare the data.

Change data capture

Change data capture (CDC) technology simplifies data pipelines by recognizing the changes that have occurred since the last data load, and then incrementally processing or ingesting the new data.

This is a highly efficient way to maintain analytics databases as new transactions occur in the source systems. An initial bulk upload of the entire data set can be supplemented by periodic or continuous updates as new transactions occur.

However, detecting and processing changes from data sources in an incremental way can be challenging for traditional data warehouse systems. Snowflake's table streams capability can incrementally consume changes as they occur in source tables in Snowflake such as staging tables with raw data. This is an important capability because it avoids having to reprocess large data sets whenever new data arrives. Snowflake retains version history for tables, allowing it to identify incremental changes quickly and to drive a data pipeline in a more or less continuous fashion. Its built-in data ingestion service can asynchronously load data from a cloud storage environment, automatically detecting the differences as data is changed. This increases performance and reduces compute capacity needs by allowing you to focus processing on the changed data or newly added data instead of bringing in the whole data set again.

Near-zero maintenance

With Snowflake's fully managed software as a service (SaaS) offering, there is no need to manage infrastructure or optimize low-level resources. Snowflake optimizes each query and each pipeline operation without requiring users to understand the processing engine. There is no need to engage in manual housekeeping tasks such as vacuuming or physical data file optimization. A global metadata layer manages permissions and transactions. By contrast, in order to use Spark or Hadoop effectively, you need to understand these environments at a deep level. Snowflake encourages data professionals to focus on higher value activities.

Snowpark

Snowpark is a new developer experience for Snowflake. Snowflake has always delivered performance and ease-of-use for users familiar with SQL. With Snowpark, data engineers, data scientists, and developers who prefer other languages can take advantage of the same benefits.

Snowpark enables you to write code directly with Snowflake in a way that is deeply integrated into the languages you use most, using familiar concepts such as DataFrames. But the most important aspect of Snowpark is that it has been designed and optimized to leverage the best of the Snowflake engine: performance, reliability, and scalability with near-zero maintenance. Think of the power of a declarative SQL statement available through a well-known API pattern in Scala, Java, or Python. Snowpark for Scala is available now with more language support coming soon.

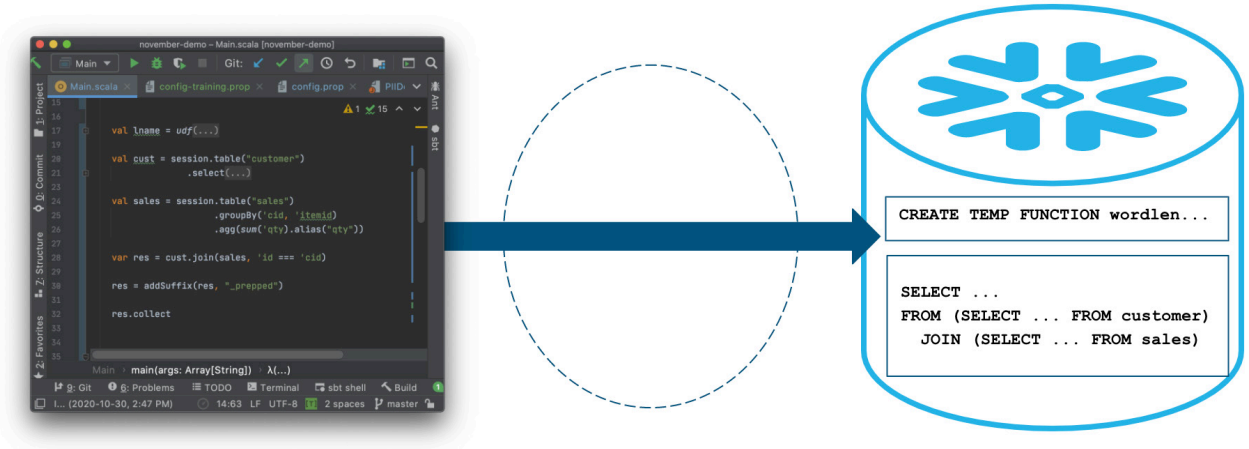

Figure 4: Snowpark pushes all of its operations directly to Snowflake without Spark or any other intermediary.

LEVERAGING THE SNOWFLAKE DATA CLOUD

The Snowflake Data Cloud is a global data network that spans multiple public clouds. Inside the Data Cloud, you can unite siloed data, easily share governed data, and execute data-driven workloads such as data warehousing, data lakes, data engineering, data science, data applications, and secure data sharing. Because all data is consolidated in one place, you can eliminate data silos—and the associated administrative procedures that go along with maintaining those silos.

The Data Cloud enables data engineers to create data pipelines that streamline data ingestion and transformation tasks by removing unnecessary data pipelines and redundant data processing. For example, curated, live, accurate data sets can be easily shared through a direct share between two accounts, a data exchange set up for a designated group, or in Snowflake Data Marketplace, where third-party data is available from more than 140 data providers as of April 30, 2021. In all cases, data sets are query-ready, so data engineers no longer need to build complex pipelines or APIs just to load or copy data again, and data scientists can get instant and secure access to all the data relevant to their models.

Data in the Data Cloud doesn't require ETL: authorized users access it directly from the source, subject to a universal data governance model that includes features such as role-based access control (RBAC) and Snowflake's Dynamic Data Masking. The Data Cloud not only facilitates internal collaboration but also makes it easy to share governed data with partners and customers, while leveraging a cohesive set of data services, including manipulating data formats, scaling data systems, and enforcing data quality and security.

GUIDING PRINCIPLES OF DATA PIPELINE MODERNIZATION

Whether data arises from an enterprise application, a website, or a series of IoT sensors, data engineers must figure out how to capture data from those data sources, ingest it into a versatile data repository, and put it in a useful form for the user community. Traditional data pipeline architectures often require you to land and transfer data into many systems, which can lead to having many different copies of the data in many different places. It's much simpler to manage your data when you can consolidate it all in one location—as a single source of truth.

How do you get there? Follow these guidelines to get started:

1. **Take a critical look at all your data pipelines.** Do some of them exist merely to optimize the physical layout of your data, without adding business value? If so, ask yourself if there is a better, simpler way to process and manage your data.

2. **Think about your evolving data needs.** Once you have assessed your current and future needs, compare those needs to the reality of what your existing architecture and data processing engine can deliver. Look for opportunities to simplify, and don't be bound by legacy technology.

3. **Root out hidden complexity.** How many different services are you running in your data stack? How easy is it to access data across these services? Do your data pipelines have to work around boundaries between different data silos? Do you have to duplicate efforts or run multiple data management utilities to ensure good data protection, security, and governance? Complexity is the enemy of scale.

4. **Take a hard look at costs.** Do your core data pipeline services leverage a usage-based business model? Is it difficult to develop new pipelines from scratch, and are special skills required? How much time does your technology team spend manually optimizing these systems? Make sure you include the cost to manage and govern your data and your data pipelines with the cost of running and maintaining your data architecture.

5. **Create value-added pipelines.** Whether a specific data transformation happens within a data pipeline or a query operation, the logic required to join, group, aggregate, and filter that data is essentially the same. Moving these computations "upstream" into the pipeline accelerates performance and amortizes processing cost when users issue the same or similar queries repeatedly.

Your data is an important product for your organization. Consider not only the value of that data internally but also the additional value you can create with data pipelines that stretch across companies, regions, continents, and cloud providers. When you liberate your data, you can collaborate with others more effectively.

ABOUT SNOWFLAKE

Snowflake delivers the Data Cloud—a global network where thousands of organizations mobilize data with near-unlimited scale, concurrency, and performance. Inside the Data Cloud, organizations unite their siloed data, easily discover and securely share governed data, and execute diverse analytic work-loads. Wherever data or users live, Snowflake delivers a single and seamless experience across multiple public clouds. Snowflake's platform is the engine that powers and provides access to the Data Cloud, creating a solution for data warehousing, data lakes, data engineering, data science, data application development, and data sharing. Join Snowflake customers, partners, and data providers already taking their businesses to new frontiers in the Data Cloud. **Snowflake.com**.

This white paper contains express and implied forwarding-looking statements, including statements regarding (i) our business strategy, (ii) our products, services, and technology offerings, including those that are under development, (iii) market growth, trends, and competitive considerations, and (iv) the integration, interoperability, and availability of our products with and on third-party platforms. These forward-looking statements are subject to a number of risks, uncertain-ties and assumptions, including those described under the heading "Risk Factors" and else-where in the Quarterly Report on Form 10-Q for the fiscal quarter ended April 30, 2021 that Snowflake has filed with the Securities and Exchange Commission. In light of these risks, uncertain-ties, and assumptions, actual results could differ materially and adversely from those anticipated or implied in the forward-looking statements. As a result, you should not rely on any for-warding-looking statements as predictions of future events.