Inside the Snowflake Elastic Data Warehouse™

THE NEED FOR CHANGE

Data and the way that data is used have changed, but data warehousing has not. Today's premises-based data warehouses are based on technology that is, at its core, decades old. To meet the demands and opportunities of today, data warehouses have to fundamentally change.

* ***Data has changed.*** It used to be that the data you needed to analyze came primarily from internal sources (e.g. transactional, ERP, and CRM systems) in well-defined, structured forms at a reasonably predictable rate and volume. Today data comes not only from those sources, but increasingly from constantly evolving sources of machine-generated data outside your direct control such as application logs, web interactions, mobile devices, sensors, and more. That data frequently arrives in flexible semi-structured formats such as JSON or Avro and arrives in highly variable rates and volumes.
* ***The ways in which data is used have changed.*** Data used to flow through complex ETL pipelines into a data warehouse, where reporting queries ran periodically to update a known set of dashboards and reports. That process often took days. Today a wide array of analysts need to explore and experiment with data as quickly as possible, even without knowing in advance where there might be value in it. Not only analysts, but also a growing number of applications need immediate access to data to make decisions.
* ***Technology has evolved.*** There are technologies available today that were not even on the radar when conventional data warehouses were designed. For example, cloud applications and cloud infrastructure have emerged to play a critical role in the IT strategy of all types and sizes of organizations.

Although data warehousing technology has ably served organizations over many years, the accumulated baggage of several decades has taken its toll. Data warehouses have increasingly failed to adapt to these changes: they struggle to handle rapidly arriving and constantly evolving data, to provide the flexibility to scale rapidly to meet ever-changing demands, and to do that cost-effectively.

At the same time, newer "big data" offerings such as those based on Hadoop are not the answer either. They are useful tools for advanced data science, but were simply not designed for data warehousing: they require difficult-to-find new skills, are not fully compatible with the existing ecosystem of SQL-based tools, and fail to deliver interactive performance.

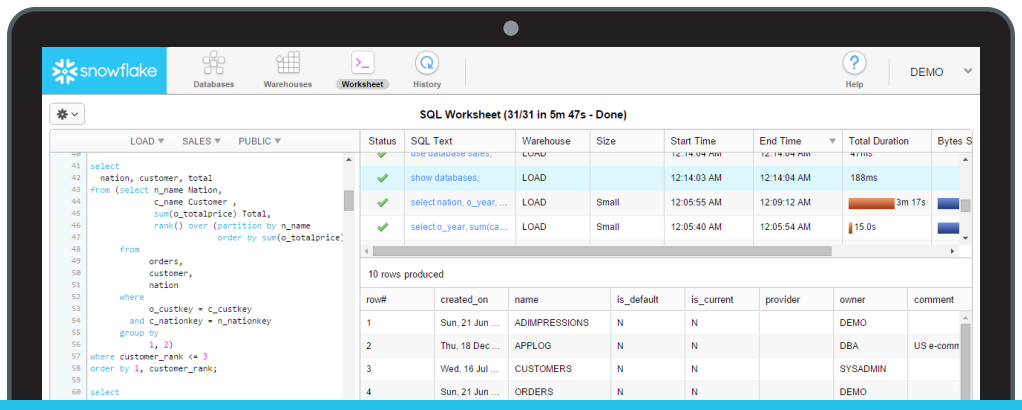
> *Today's data warehouses are based on technology that is decades old. To meet the demands and opportunities of today, data warehouses have to fundamentally change.*

REIMAGINING THE DATA WAREHOUSE

Addressing these limitations is not just a matter of adding a few more features to existing architectures—there are fundamental assumptions baked into current architectures that are no longer true. A redesign of the data warehouse is necessary.

If we were to start over, unencumbered by the accumulated baggage of data warehousing history, what would we build? The ideal data warehouse would combine the strengths of data warehousing—

FIG 1 Snowflake can store any scale of diverse data at a low cost.



performance, security, and a broad ecosystem, with the flexibility and scalability of "big data" systems. Such a data warehouse would be:

* ***A true service:*** The data warehouse would take care of all infrastructure and management automatically so that users could focus on getting value from their data.
* ***Completely elastic:*** Able to operate at any scale, and to scale up and down at any time without disruption.
* ***Able to store any type of business data:*** Natively handle diverse types of data at a very low cost and at any scale without requiring complex transformations before loading that data into the data warehouse.
* ***Seamless fit with existing skills and tools:*** There has been a significant focus on tools for the small number of data scientists, without addressing the huge community of people and tools that understand standard SQL. Full support for standard SQL makes it possible to offer a better engine for those users--no new expertise, no new programming paradigms, and no new training required.

As we considered these goals, we realized that the cloud is the perfect infrastructure for the ideal data warehouse, and the only infrastructure that can make it possible—the cloud offers near-infinite resources in a wide array of configurations, available on-demand, at a cost based only on use. Public cloud offerings have matured such that they current support a large and growing set of enterprise computing needs, often delivering higher data durability and overall availability than private datacenters, without the upfront capital costs.

Although a small number of data warehouse options today are marketed as "cloud", none of them was designed for the cloud. These offerings are either managed service offerings of existing products or simply an installation of existing software in a public cloud infrastructure.

SNOWFLAKE: ELASTIC DATA WAREHOUSE AS A SERVICE

Snowflake was founded by a team with deep experience in the current generation of data warehousing products. Our team set out to build a completely new data warehouse, one designed to deliver dynamic infrastructure, performance, and flexibility at a fraction of the cost. We did that not by starting with an existing software code base such as Postgres or Hadoop, but rather with a compiler and a willingness to write every line of code in the core data warehouse engine.

Snowflake's solution: a complete SQL data warehouse designed from the ground up as a software service that can take full advantage of cloud infrastructure. Snowflake's data warehouse is, at its core, a massively parallel processing (MPP) database that is fully relational, ACID compliant, and processes standard SQL natively, without translation or simulation.

We built our data warehouse service with a new architecture, one designed with the goal to deliver the best of data warehousing and big data solutions, but without the limitations of current architectures.

> *"Snowflake is the first analytic database that really leverages the power of the cloud."*
>
> Jeff Shukis, *VP Engineering and Tech Ops, VoiceBase*

The Limits of Current Architectures

Current data warehouse architectures were not designed to provide dynamic elasticity. Data warehouse appliances, with their fixed configurations, are certainly the most limited, but even software-only products cannot be truly elastic. That is because of limitations of the two dominant database architectures—shared-disk and shared-nothing.

In a shared-disk system, all data is stored on a storage device that is accessible from all of the nodes in the database cluster. All changes to data are written to the shared disk to ensure that all nodes see a consistent version of the data. Shared-disk databases keep data management simple–all processing nodes in the database cluster have direct access to all data, and that data is consistent because all modifications to data are written to the shared disk. However, the scalability of this architecture is limited because as the number of processing nodes increases, the storage device and the network to it quickly become bottlenecks, forcing

processing to slow down to wait for data to be returned from the shared disk.

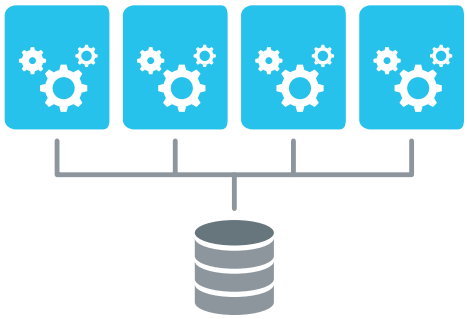


FIG 2.1 Shared disk architecture

Shared-nothing databases use a different architecture that distributes data across all of the processing nodes in the system, with each node holding a subset of the data in the database. Shared-nothing architectures eliminate the bottleneck of communication with a shared disk, but their elasticity is still limited. As the database is scaled to more and more nodes, shuffling data between nodes can become costly—performance is heavily dependent on how data is distributed across the nodes in the system. Distribution of data across the processing nodes is typically done through static assignment. Data is distributed at the time it is loaded by either a user-specified distribution key or by a default algorithm, and changing that typically requires completely redistributing data across the cluster, a slow and disruptive operation. Further, shared-nothing architectures require careful sizing of the hardware infrastructure in order to deliver the right balance of processing, memory, I/O bandwidth, and storage for the expected workloads because the balance in each node is fixed.

FIG 2.2 Shared nothing architecture

A NEW ARCHITECTURE: MULTI-CLUSTER, SHARED DATA

Snowflake started with a fundamentally new architecture for data warehousing in order to deliver this elasticity, an architecture that physically separates but logically integrates storage and compute. Snowflake's multi-cluster, shared data architecture consists of these independent components:

* ***Database storage:*** the persistent storage layer for data in the Snowflake data warehouse service.
* ***Processing:*** a collection of compute resources that execute data processing tasks required for queries.
* ***Cloud services:*** a collection of services that manage global aspects of the Snowflake deployment, including metadata, infrastructure management, security, and access control.

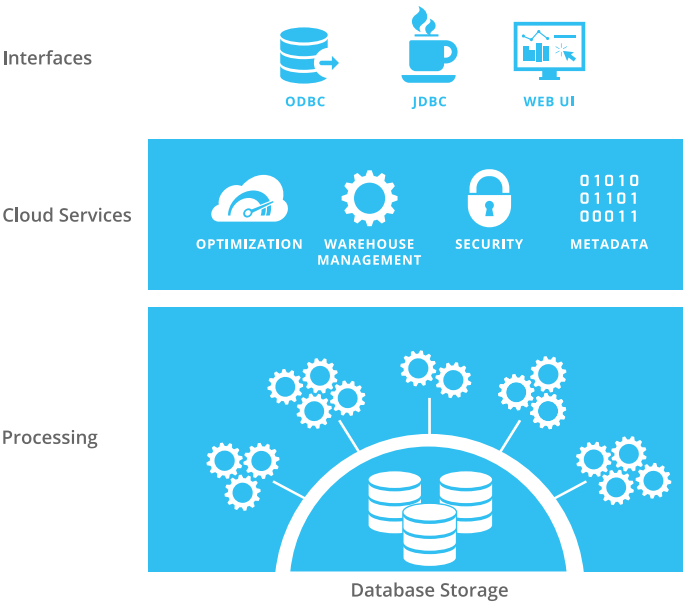


FIG 3 Built from the ground up for the cloud, Snowflake's unique architecture physically separates and logically integrates compute and storage

In a traditional data warehouse, the different layers are tightly coupled due to the way that the platform is built: storage and compute are typically tied together in a fixed configuration, either due to the configuration of the physical nodes used by the system (whether commodity servers or cloud compute instances) or due to being part of a data warehouse appliance. Even "big data" platforms tie storage and compute tightly together because they store data on the compute nodes in the system. These platforms can scale compute and storage to some degree, but even they have limited ability to scale as the number of workloads and users increase.

Snowflake dynamically brings together these layers, delivering exactly the resources needed, when they are needed. The database storage layer resides in a scalable cloud storage service, such as Amazon Web Services S3 service, which ensures data replication and availability without requiring management by users. Snowflake stores data that has been loaded in an optimized, columnar format in the database storage layer, organized into databases as specified by the user.

To allocate compute resources, users create "virtual warehouses", which are MPP compute clusters for processing queries. These virtual warehouses have the ability to access any of the databases in the database storage layer to which they have been granted access, and they can be created, resized up or down, and deleted dynamically as resource needs change. As virtual warehouses execute queries, they transparently cache data from the database storage layer. This hybrid architecture provides the unified storage characteristic of a shared-disk architecture, but with the performance benefits of a shared-nothing architecture.

The cloud services layer consists of a set of services that manage the Snowflake system—metadata, security, access control, and infrastructure. The services in this layer provide communicate with client applications (including the Snowflake web user interface, JDBC, and ODBC clients) to coordinate query processing and returning results to applications. The services in this layer retain metadata about the data stored in Snowflake and how that data has been used, making it possible for new virtual warehouses to immediately have the benefit of that metadata for optimization.

Unlike a traditional data warehouse, Snowflake can dynamically bring together the optimal resources to handle each particular usage scenario, with the right balance of IO, memory, CPU, etc. This flexibility is what makes it possible to support data warehouse workloads with different query and data access patterns in a single service.

The Snowflake architecture enables the following key capabilities:

* Multidimensional elasticity
* Diverse data together in a single system, without compromise
* A self-managing service, not infrastructure

MULTIDIMENSIONAL ELASTICITY

The ideal data warehouse would be able to size up and down on demand to provide exactly the capacity and performance needed, exactly when it is needed. However, current products are difficult and often costly to scale up, and almost impossible to scale down. That forces an upfront capacity planning exercise that typically results in oversizing the data warehouse, sizing for the peak workload but running underutilized at all other times.

Cloud infrastructure uniquely enables full elasticity—resources can be added and discarded at any time. That makes it possible to have exactly the resources you need for all users and workloads—but only with an architecture designed to take full advantage of that.

Snowflake's separation of storage, compute, and system services makes it possible to dynamically combine and modify the configuration of the system–resources can be sized and scaled independently and transparently, on-the-fly. This makes it possible for Snowflake to deliver full elasticity across multiple dimensions:

* ***Data:*** The amount of data stored can be increased or decreased at any time. Unlike shared-nothing architectures where the ratio of storage to compute is fixed, the compute configuration is unaffected by changes to the volume of data in the system. This architecture also makes it possible to store data at a very low cost because no compute resources are required for storing data in the database.

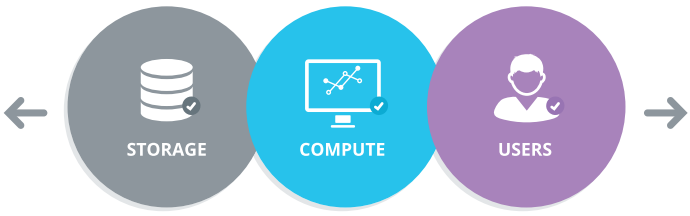


FIG 3 Snowflake's unique architecture enables it to elastically support any scale of data, processing, and workloads

* ***Compute:*** The compute resources being used for query processing can also be scaled up or down at any time as the intensity of the workload on the system changes. Because storage and compute are decoupled and the data is dynamically distributed,

changing the compute resources does not require a reshuffling of data. Compute resources can be change on-the-fly without disruption.

* ***Users:*** With traditional architectures it is not possible to independently scale the concurrency of queries to support more workloads and users without also scaling compute and storage—as more users and workloads are added, the system simply gets slower and slower. Regardless of how large the cluster becomes, eventually the system cannot support additional concurrency and the only option is to purchase an additional, new system. This brings with it the extra management burden of replicating or migrating data across systems. Snowflake can scale to support more users and workloads without performance impact because multiple virtual warehouses can be deployed on-demand, all with access to the same data.

DIVERSE DATA, WITHOUT COMPROMISE

Snowflake designed a data warehouse that allows you to store all of your business data in a single system. That is a sharp contrast from current products, which are typically optimal for only a single type of data and which force you to create data silos for different data and different uses of data.

Native Support for Semi-Structured Data

Traditional database architectures were designed for storing and processing data that strictly adhered to structured forms that could be stored in relational rows and columns. These architectures built their processing models and optimizations around the assumption that this data consistently contained the set of columns defined by the database schema. This assumption made possible performance and storage optimizations such as indexes and pruning, but at the cost of demanding a static, costly-to-change data model.

Structured relational data remains a critical type of data for reporting and analysis. But a significant share of data today is machine-generated data delivered in semi-structured data formats such as JSON, Avro, and XML.

Such data is commonly hierarchical and often does not adhere to a pre-defined schema—data elements may exist in some records but not others, while new elements may appear at any time in any record. Correlating and analyzing the information in this semi-structured data together with structured data is important to extract the information within it.

Using semi-structured data in a traditional relational database today requires compromising flexibility or performance. One approach is to transform that data into relational form by extracting fields and flattening hierarchies so that it can be loaded into a relational database schema. This approach effectively puts the constraints of a fixed schema on that semi-structured data, sacrificing information and flexibility—fields not specified for extraction are lost including new fields that appear in the data because adding new fields requires redesigning the data pipeline and updating all data previously loaded to include the new fields. The alternative to this approach, which some databases have implemented, is a special datatype for storing semi-structured data as a complex object.

Although this approach preserves the information and flexibility in the semi-structured data, it sacrifices performance because the relational database engine does not know how to optimize processing for such datatypes. For example, accessing a single element in an object commonly requires a full-text scan of the entire object in order to locate the element.

Because traditional data warehouses do not support the capabilities needed to effectively store and process semi-structured data, many customers have turned to alternative approaches such as Hadoop for processing this information. While Hadoop systems can load semi-structured data without requiring definition of a schema and transformation of the data, they are inefficient at processing structured data and require specialized skills that are not broadly available. Some customers are using a combination of both a traditional data warehouse and a Hadoop cluster to bring together structured and semi-structured data not because it is simple or efficient, but because it is the only solution available to them.

The Need for a Fresh Approach

Snowflake took a new and different approach, designing a data warehouse that can store and process semi-structured and structured data together in a single

service without compromising flexibility or performance. Snowflake's patent-pending approach provides native storage of semi-structured data together with the relational model and robust optimizations that relational databases provide.

> *"I can't say enough about how fantastic the native JSON support is. Snowflake lets us load our JSON data as is, flatten it all out, load it into the event tables, and then parse that into views. My analysts are really happy about this."*
>
> Josh McDonald, *Director of Analytics Engineering, KIXEYE*

Snowflake started by making it possible to flexibly store semi-structured records inside a relational table. Snowflake provides a custom datatype that allows schema-less storage of multiple variants of hierarchical data including semi-structured forms such as JSON and Avro. This makes it possible to load semi-structured data directly into Snowflake without preprocessing, without losing information, and without defining a schema for the semi-structured data. You simply create a table containing a column with Snowflake's VARIANT datatype and then issue a load command to load files containing semi-structured data into that table.

When Snowflake loads that semi-structured data, it optimizes how it stores that data internally by automatically discovering the attributes and structure that exist in the data and using that knowledge to optimize how the data is stored. As Snowflake loads semi-structured data, it looks for repeated attributes across records and then organizes and stores those repeated attributes separately, enabling better compression and fast access similarly to the way that a columnar database optimizes storage of columns of data. Statistics about these pseudo-columns are also calculated and stored in Snowflake's metadata repository for use in optimizing queries. This storage optimization is completely transparent to the user.

In addition to optimizing storage of semi-structured data, Snowflake provides querying capabilities that can take full advantage of the optimization capabilities of a relational database. Snowflake enables querying of semi-structured data through extensions to SQL, making it simple to use relational queries that can combine access to structured and semi-structured data in a single query. Because of Snowflake's approach to storing semi-structured data, the Snowflake query optimizer has metadata information about the semi-structured data that allows it to optimize access to that data—for example, statistics in the metadata allow the optimizer to use pruning to minimize the amount of data that needs to be accessed.

Single System for All Business Data

Current architectures create isolated silos of data. Structured data is processed in a data warehouse. Semi-structured data is processed with Hadoop. Complex, multi-step operations are required to bring this data together. Scalability limits force organizations to separate workloads and data into separate data warehouses and datamarts, each of which is an island of data that has limited visibility and access to data in other database clusters.

All of these silos make it possible to configure a data warehouse, datamart, or Hadoop cluster that is tuned for a particular workload, but at the cost and overhead of needing to manage multiple systems and implement processes to replicate multiple copies of data. Even with all of this infrastructure, it is often difficult to gain the desired business insight because the system in use doesn't contain all of the relevant data and is unable to effectively process a query to provide the desired results.

Snowflake's decoupling of database storage from compute makes it possible to keep all business data in a single system without sacrificing performance, and without an explosion in cost. Different workloads and groups can be provided independent virtual warehouses sized specifically for their needs while still having access to all data in the system to which it has the appropriate access permissions.

SELF-MANAGING SERVICE

Conventional data warehouses and "big data" platforms require significant care and feeding. They rely on skilled database administrators and operations experts spending significant time and effort managing and tuning the data warehouse or data platform: choosing data distribution schemes, creating and maintaining indexes, updating metadata, cleaning up files, and more.

Manual optimization was feasible in an environment where queries were predictable and the number of different workloads was few, but it simply does not scale when there are a large number of ever-changing workloads. The time and effort required to optimize the system for all those different workloads quickly gets in the way of actually analyzing data.

In contrast, Snowflake set out to build a data warehouse as a software service where users focus on analyzing data rather than spending time managing and tuning hardware and software. That required Snowflake to design a data warehouse that would:

* ***Eliminate management of hardware and software infrastructure.*** The data warehouse should not require users to think about how to deploy and configure physical hardware. Similarly, users should not need to worry about installing, configuring, patching, and updating software.
* ***Enable the system to learn and adapt.*** Rather than require users to invest time configuring and tuning (and retuning) a wide array of parameters, Snowflake designed a data warehouse that sees how it is being used and dynamically adapts based on that information.

Eliminating Infrastructure Management

The Snowflake data warehouse was designed to remove the pains of managing hardware and software infrastructure. It is built on cloud infrastructure, which it transparently manages for the user. Users simply log in to the Snowflake service and it is immediately available, without complex setup required.

Ongoing management of software infrastructure is also managed by Snowflake. Users do not need to manage patches, upgrades, and system security. The Snowflake service takes care of all of these automatically.

Capacity planning, a painful requirement for deployment of a conventional on-premises data warehouse, is all but eliminated because Snowflake makes it possible to add and subtract resources on the fly. Because it is easy to scale up and down based on need, you are not forced into a huge upfront cost in order to ensure sufficient capacity for future needs.

> *"Snowflake is faster, more flexible, and more scalable than the alternatives on the market. The fact that we don't need to do any configuration or tuning is great because we can focus on analyzing data instead of on managing and tuning a data warehouse."*
>
> *Craig Lancaster, CTO, Jana*

DELIVERING AN ADAPTIVE DATA WAREHOUSE

Snowflake realized that the only way to deliver a service that could handle all users and all workloads was to design a system that would automatically adapt and optimize itself based on usage. Making that possible required two things:

* A feedback loop that would automatically observe usage and identify optimizations based on that information.
* The ability to dynamically adapt to take advantage of the optimizations identified by that feedback loop.

Snowflake's unique patent-pending technology delivers both this feedback loop and the ability to take advantage of the optimizations it identifies. Because Snowflake stores metadata separately from database storage and virtual warehouses, metadata that is needed to optimize queries and data access is immediately available to any virtual warehouse that needs it, even virtual warehouses that have just been created.

Examples of how Snowflake adaptively optimizes the data warehouse based on usage include:

* ***Data distribution*** is managed automatically by Snowflake based on usage. Rather than use a static partitioning scheme based on a distribution algorithm or key chosen by the user at load time, Snowflake automatically manages how data is distributed in the virtual warehouse. Data is automatically redistributed based on usage to minimize data shuffling and maximize performance.
* ***Loading data*** is dramatically simpler because complex ETL data pipelines are no longer needed to prepare data for loading. Snowflake natively supports and optimizes diverse data, both structured and semi-structured, while making that data accessible via SQL.
* ***Dynamic query optimization*** ensures that Snowflake operates as efficiently as possible by looking at the state of the system when a query is dispatched for execution, not just when it is first compiled. That adaptability is crucial to enable a system that can be scaled up and down on the fly.

THE IMPACT OF REINVENTION

By reimagining and reinventing the data warehouse, Snowflake has addressed key limitations of today's technology. Doing so required a new architecture, one that was not tied to the decades of data warehouse architectural history.

As a result, users can focus on getting value out of their data without needing to spend significant time worrying about all of the tasks that are required with current data warehousing solutions. Rather than being bottlenecked waiting for the availability of overstretched IT and data scientist resources, analysts get rapid access to data in a service that can operate at any scale of data, users, and workloads.

To learn more about Snowflake, visit us on the web at at www.snowflake.net or contact us at customer@snowflake.net to schedule a product demonstration by a Snowflake specialist.

About Snowflake

Snowflake Computing, the cloud data warehousing company, has reinvented the data warehouse for the cloud and today's data. The Snowflake Elastic Data Warehouse is built from the cloud up with a patent-pending new architecture that delivers 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. The company is backed by leading investors including Altimeter Capital, Redpoint Ventures, Sutter Hill Ventures and Wing Ventures. Snowflake is headquartered in Silicon Valley and can be found online at snowflake.net.

snowflake

www.snowflake.net | @SnowflakeDB