Author: Marc Staimer

TECHNICAL PAPER

Why Snowflake **Melts** When Compared to the Oracle Autonomous Data Warehouse

Why Snowflake **Melts** When Compared to the Oracle Autonomous Data Warehouse

Introduction

Database cloud services have grown rapidly over the past few years. Kenneth Research pegs the worldwide database cloud services market at $10.37 Billion USD at the end of 2019 with a CAGR of 15.7% through 2026. That puts the revenue at more than $28.78 Billion USD by the end of 2026 making database cloud services a very large market. The Covid-19 pandemic appears to have accelerated that growth.

Like many fast-growing markets, it has spawned several database cloud services including OLTP, data warehousing, big data analytics, AI-machine learning (AI-ML), and more. Database cloud services include structured, semi-structured, and unstructured databases. One area seeing significant market demand growth is analytics. Two of the more distinguished data warehouse cloud services are Oracle Autonomous Data Warehouse and **Snowflake**.

Snowflake has received much trade press about its data cloud vision, simplicity, scalability, flexibility, data sharing, and multi-cloud capabilities across AWS, Azure, and Google Cloud Platform. Combined with its highly successful IPO in 2020 raising $3.4B USD, **Snowflake** has generated significant hype. Autonomous Data Warehouse has been lauded for its unique autonomy, unrivaled capabilities such as extremes of elasticity and non-disruptive online patching, and superb price/performance.

These two competitive cloud services would appear superficially to be similar offerings. A double click below the surface reveals that they are quite different.

Up until now, these two celebrated competitive database cloud services have not been compared and contrasted. They are both more frequently compared to much less siloed database cloud services such as AWS Redshift, Microsoft Azure SQL Data Warehouse, and Google BigQuery. But not directly to each other.

This raises the questions of how are they comparable? How are they different? Which workloads work better in which service? This technical research paper goes deep to answer these questions. And the results are both unexpected and to some, a bit surprising. It becomes empirically obvious that Oracle Autonomous Data Warehouse has extensive advantages over Snowflake including:

- A more complete vision.
- Better execution of that vision.
    - Simpler setup, queries, scaling, data protection, flexibility, and overall usefulness.
    - Much more automation.
    - Greater reliability.
    - Considerably better performance.
    - Better elasticity both upwards and downwards.
    - Tighter, more complete security
    - Superior flexibility.
    - Significantly lower costs.
    - Far better cost/performance.

Snowflake on the other hand, has one advantage over Oracle Autonomous Data Warehouse. That being its currently unique ability to share data between different Snowflake users or customers without copying or moving data.

A deep look illustrates why Oracle Autonomous Data Warehouse is the better pick for the majority of global organizations[1].

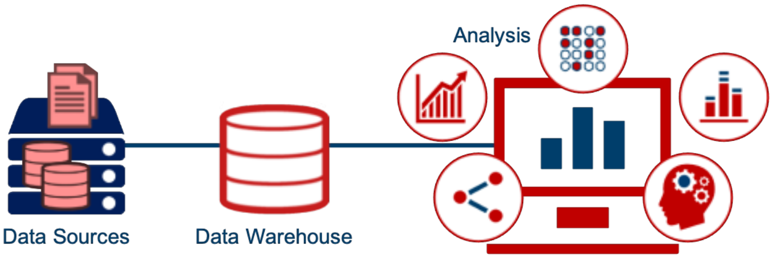
[1] The conclusions and opinions stated in this paper are solely those of the author.

Table of Contents

Level Setting

What is a Data Warehouse?

A data warehouse is a database system used for reporting and data analytics. It is a core business intelligence (BI) component. Data warehouses are centralized repositories of integrated data from one or more separate connected and disconnected sources. The purpose of a data warehouse is to provide actionable insight into the organization's performance. It does this by comparing data consolidated from multiple sources. The data warehouse runs query and analysis on historical data derived from transactional sources. It drives business intelligence (BI) while enabling analytical reports for the entire organization.



Data in the data warehouse is generally thought to be historical data and does not change. Nor should it be alterable. This was historically by design since the analytics are on events that have already occurred. The objective was to discover changes and/or patterns in data that have occurred over time. It was essential for the data warehouse data be stored securely, reliably, quickly retrievable, and easily manageable. However, data warehouses have changed and most today have a data-refresh/data-restate process. This is especially prevalent when loading sales and financial related data. This makes WORM storage models problematic.

The data stored in a data warehouse does not originate there. It's uploaded or pipelined from transactional databases. Before the data warehouse can analyze, provide insights, or reports on that data it may require cleansing, and transformation. The two most common methods for this are extract, transform, and load (ETL) or extract, load, and transform (ELT).

Extraction involves gathering large amounts of data from multiple source points. The data is then transformed. It starts with the cleansing after it has been compiled. Cleansing scrutinizes the data for errors, correcting or excluding any errors found. The cleansed data is then converted from its transactional database format to the data warehouse format and stored. The data is also sorted, consolidated, and summarized, and coordinated, thereby making is easier to use. Data warehouses are meant to grow over time as more data is added from disparate sources.

Many IT organizations leverage data warehouses for data mining. Data mining is a process that converts raw data into useful or actionable information. It looks for information patterns that deliver insights and help improve internal business processes.

What is a Data Warehouse Cloud Service?

A data warehouse cloud service in its basic essence is a data warehouse in a public cloud. However, to be an attractive cloud service offering requires more. The key is for the service to eliminate the mundane, time-consuming, labor-intensive data warehouse administrative tasks. Tasks such as configuring, provisioning, performance tuning, patching, upgrading, troubleshooting, root cause analysis, data protection, backup, data recoveries, disaster recoveries, and business continuity.

Eliminating these tasks from the data warehouse administrators empowers them to focus on more strategic and revenue-generating operations. Operations that focus on actionable insights, time-to-market, new projects, project completion, all done faster.

Data warehouse cloud services eliminate much of the risk associated with data analytics and business intelligence. There is no requirement for upfront hardware or data center investment, software license costs (although many services enable licenses already purchased or subscribed to run in their public cloud), implementation costs, maintenance costs, long-term contracts, etc. Customers only pay for the computing and storage resources that they use. If they are not making queries or actually doing analytics at a given time, they are not paying for the data warehouse service. They always pay for the storage their data consumes. However, they are still only paying for the resources they use.

Those resources tend to be dynamically elastic, scaling both up and down. The elastic granularity varies by service provider. Compute resources are a major component of a data warehouse cloud service's cost. This means performance has an oversized impact on what customers pay. Better performance translates into less time and fewer resources for the queries to complete. Less time and fewer resources utilized converts into lower cost. For data warehouse cloud services, time truly is money.

Data Warehouse Cloud Service Provider Vision

The data warehouse cloud service provider's vision is what drives their service offering. That vision underpins what, where, how, and when services are provided. It determines the customers and buyers they're targeting, messaging, features, functions, and roadmaps. The service provider's vision is the underlying core philosophy behind their data warehouse cloud service. It is not the data warehouse cloud service itself. Each update to that service should bring it closer to fulfilling the vision although the service provider may never fully achieve that vision. How well a given service provider is executing on its vision is validation or non-validation of that vision. Visions change over time.

Snowflake Vision

Snowflake is quite clear in its vision. They call it a "data cloud." In a nutshell, the data cloud is designed to be a very simple, easy to utilize giant data warehouse/data lake repository where all customers can share data securely without having to copy it. Or as their tagline says: "One Copy of the Data, Many Workloads."

By the end of 2020, the Snowflake data cloud is making progress on fulfilling that vision. Their data cloud enables any Snowflake customer to share their data with any other Snowflake customer without replicating the data.

Oracle Autonomous Data Warehouse Vision

Oracle's vision for Autonomous Data Warehouse is very different from Snowflake's. In Oracle's vision, databases and data warehouses are software infrastructure that should be completely automated into a transparent utility. Similar to electricity, water, or data center hardware infrastructure. Customers just turn it on and use it. They shouldn't have to set it up, provision it, update it, patch it, tune it, troubleshoot it, repair it, protect it, recover it, or manage it in any way. Oracle's vision is that databases and data warehouses should just be there transparently, always easy to use. Almost as if the database or data warehouse could just read the customer's mind as to what they want to accomplish.

By the end of 2020, the Oracle Autonomous Database has gained thousands of happy and satisfied customers. Oracle's Autonomous Database cloud services now include Autonomous Transaction Processing Database, Autonomous Data Warehouse, Autonomous JSON Database, and soon Autonomous Graphic Database.

Data Warehouse Cloud Service Provider Execution

There are many aspects to delivering an effective data warehouse cloud service. As previously described, performance is very important for several obvious reasons (faster is always better) including the impact on costs. Costs that are not just out-of-pocket, but opportunity costs as well. Other essential aspects of an effective data warehouse cloud service include simplicity/automation, scalability, security, data pipelining/ETL/ELT, and of course, TCO.

Simplicity/Automation

Simplicity/automation is a major reason an organization decides to take advantage of a data warehouse cloud service. Another word for this is convenience. The amount of simplicity/automation in a data warehouse cloud service generally correlates to the amount of time DBAs have to spend on more strategic efforts. Efforts that advance the organization's goals and objectives. Less automation consumes DBA time in manual non-productive labor-intensive efforts. It also increases human errors, causing project interruptions and delaying time-to-market. Simplicity/automation in data warehouse cloud services is essential. However, the amount of that simplicity/automation varies by service provider

Snowflake Simplicity/Automation

Snowflake is a relatively simple/automated data warehouse cloud service. Snowflake developed its data warehouse database from the ground up to specifically run within public cloud ecosystems such as AWS, Azure, and Google Cloud Platform (GCP). Snowflake asserts that its data cloud was born in and architected for the cloud[2]. Similar to the term cloud native, it generally means that its software runs in the public cloud, is optimized for public cloud infrastructure, and leverages public cloud services. Cloud native software commonly lacks the ability to also run on-prem, which is currently the case for Snowflake.

Snowflake promotes itself as cloud native with a very simple easy-to-use data warehouse. On the automation front, it generally provides auto patching, maintenance, upgrades, and data protection as a managed service. Snowflake also provides automated security with access control, single sign-on (SSO), authentication, encryption, multi-factor authentication (MFA), and secure link. Scaling and encryption rekeying within a cluster or multi-cluster is automated for Enterprise and higher priced editions. Snowflake data protection and DR is automated via continuous data protection features (CDP) Time Travel[3] and Fail-safe. For the most part, Snowflake is indeed quite simple and easy-to-use as a data warehouse, after the data has been loaded. A typical customer experience provisioning a new Snowflake account, loading pre-formatted data from local cloud storage and running queries takes about a few minutes.

Surprisingly, Snowflake comes up short on the automation front in several areas. Snowflake stresses their provisioning is automated. The caveat is that's only within a selected cluster or multi-cluster virtual data warehouse. The DBA has provisioning work to do. And yes, it's pretty easy to add nodes within a cluster or even add a cluster. But first, the DBA must choose the type of cluster to setup. Then they have to pick the correct storage options. To do those two things, they need to match their workloads to the kind of cluster that meets each workload needs. A cluster can have different fixed hardware sizes of 1 (X-Small), 2 (Small), 4 (Medium), 8 (Large), 16 (X-Large), 32 (2X-Large), 64 (3X-Large), or 128 (4X-Large) nodes.

This requires the DBA to experiment with different cluster sizes to select the one that meets their workload and performance needs. There is also no guidance or automation for Materialized Views. Snowflake assumes the DBA has lots of experience in this area. Materialized Views are critically important and utilized by DBAs to optimize performance. In other words, Snowflake does not actually automate optimized performance.

There are several other automation areas missing or deficient. Although data protection is automated for the most part, data warehouse replication, failovers, failbacks, restores, etc. are not. They require DBA intervention. Manual processes take time and effort. They also have to be tested periodically because they are subject to human error, which can be catastrophic during an emergency. Snowflake's automation or autonomy is far from complete.

Snowflake's focus on database simplicity means leaving out some capabilities and making tradeoffs. For example, Snowflake chose to forgo indexes and the enforcement of constraints in its database. Why does that mater when Snowflake is a data warehouse and doesn't purport to be anything else? Because it adds complexity and limits Snowflake to certain types of data warehouse workloads. Without indexes, every query needs to scan rows or columns in the entire table to find what it's looking for. This is true even for simple queries that simply want to look up only a few records (a common query type for operational data warehouses).

Snowflake's solution to such performance challenges is to support on-demand additional cloud infrastructure resources to be thrown at the queries. Counter-intuitively, the automated way Snowflake increase resources has a nominal effect on slow queries. This is because that automation requires a spinning up of a new cluster the exact size of the cluster with slow queries. Spinning up clusters does not add resources to slow queries. Speeding up queries requires the administrator manually intervene to increase the fixed shape size to a larger fixed shape size.

Snowflake's simplifying the data warehouse architecture has other serious tradeoffs. For constraints: Snowflake supports the following constraint types from ANSI SQL standard:

[2] Born in the cloud is not an architecture but rather marketecture (marketing description that pretends to be architecture). It means it runs in the cloud service provider's virtual machines, containers, or bare metal and utilizes cloud storage and networking.

[3] Snowflake Time Travel enables access to historical data that has been changed or deleted at any point-in-time in the defined period. It allows querying, cloning, and restoring data related objects, tables, schemas, and databases for up to 90 days that may have been accidentally or deliberately deleted. DR of Snowflake historical data is provided through Fail-safe.

- UNIQUE KEY
- PRIMARY KEY
- FOREIGN KEY
- NOT NULL

However, from Snowflake documentation: "Snowflake supports defining and maintaining constraints, but does not enforce them, except for NOT NULL constraints, which are always enforced."

A key area where Snowflake complicates matters is the loading of data. Snowflake lacks specific relational capabilities such as indexing, data normalization, and ER models that reduce data redundancy. Why does that matter when Snowflake is a data warehouse and doesn't purport to be anything else? Because it adds significant complexity to every Snowflake data warehouse. Data warehouses analyze the data that is generated elsewhere and, quite frequently, in transactional databases. That requires the transactional data to somehow be pipelined or ELT to make it useful for any Snowflake analysis. Snowflake has limited tools and only a few pre-packaged connectors, such as for Salesforce.

Without enforced constraints, Snowflake customers have no assurances of the data quality inside their data warehouse. This shortcoming puts responsibility on the applications or developers to discover and correct those errors. In this case simplifying the database made the applications' work more complicated.

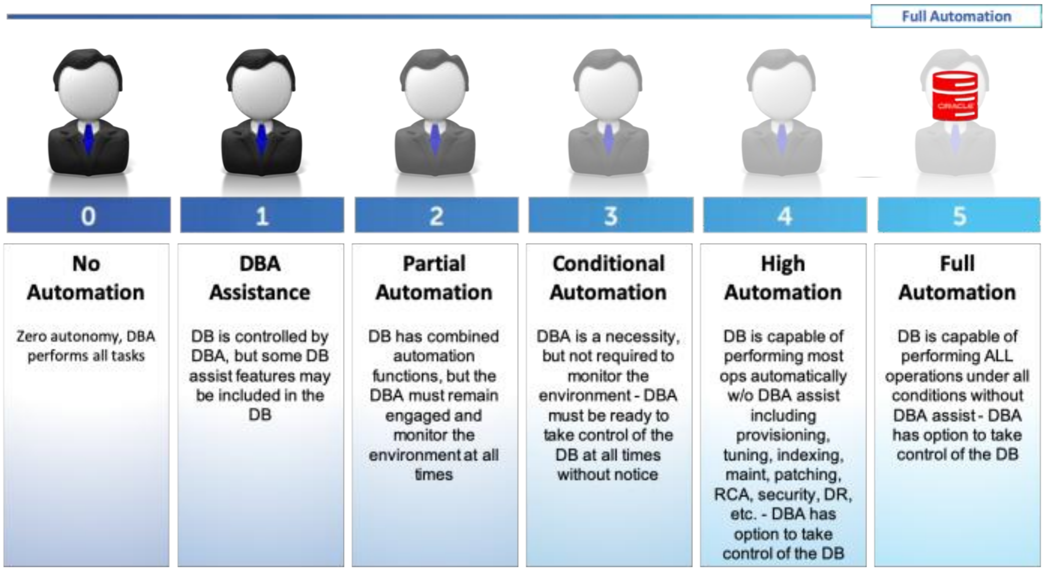
Snowflake's lack of core features like indexes and constraints has one other major consequence: any customer moving their data warehouse from Oracle or other well-established data warehouse technology will have to completely re-engineer their data transformation and load scripts work with a Snowflake data warehouse. This process consumes unexpected large amounts of time, resources, and cost. These costs are currently unavoidable on the Snowflake platform.

Snowflake is definitely a very simple system to use. They have no management knobs to fiddle with. But it is an immature service requiring more elbow grease as well as third-party tools and services. More tools and automation are needed.

Oracle Autonomous Data Warehouse Simplicity/Automation

Oracle has set the standard for simplicity and automation with Autonomous Data Warehouse. Customers need only a few minutes to create the instance, load the data, and run queries. Autonomous Data Warehouse eliminates just about all manual labor-intensive administrative overhead when operating a data warehouse. That includes seamless data access from transactional databases, securing that data, developing data-driven applications, protecting the data, DR, business continuity, and much more.

There's a reason "Autonomous" is included in the name. The definition of autonomous is self-governing to a significant degree. In technology it means a high degree of automation. There are six levels of database autonomy established for autonomous systems ranging from 0 to 5 as seen in the diagram below, Oracle Autonomous Data Warehouse is at the full automation level. There are no overhead tasks DBAs need to perform.



0	1	2	3	4	5
No Automation	**DBA Assistance**	**Partial Automation**	**Conditional Automation**	**High Automation**	**Full Automation**
Zero autonomy, DBA performs all tasks	DB is controlled by DBA, but some DB assist features may be included in the DB	DB has combined automation functions, but the DBA must remain engaged and monitor the environment at all times	DBA is a necessity, but not required to monitor the environment - DBA must be ready to take control of the DB at all times without notice	DB is capable of performing most ops automatically w/o DBA assist including provisioning, tuning, indexing, maint, patching, RCA, security, DR, etc. - DBA has option to take control of the DB	DB is capable of performing ALL operations under all conditions without DBA assist - DBA has option to take control of the DB

Oracle Autonomous Data Warehouse automates provisioning, configuring, securing, tuning, scaling, patching, backing up, troubleshooting, repairing, and failover-failback. It does not stop there. It automates much more including:

Oracle Autonomous Data Warehouse Automates

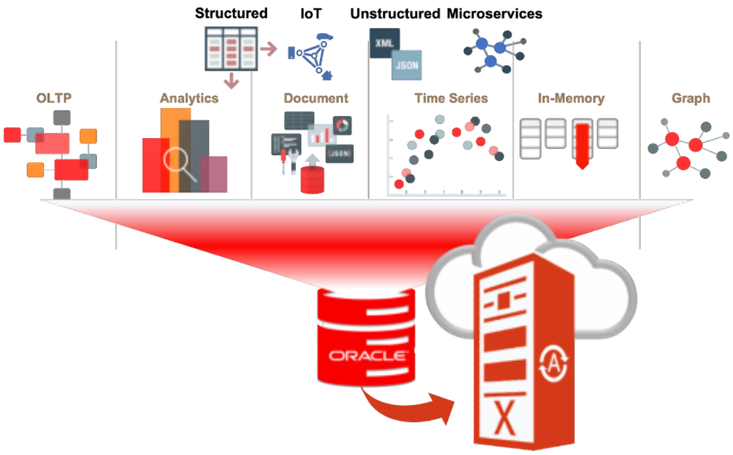
Materialized Views	Clone Refresh	Segment Space Mgmt
Indexes & Partitions	SQL Tuning	Statistics Gathering
Columnar Flash	Workload Capture/Replay	Storage Management
In-Memory population	SQL Plan Management	Workload Repository
Application Continuity	SQL Monitor Capture	Diagnostics Monitor
Health Framework	Data Optimization	Query Rewrite
Diagnostic Framework	Memory Management	Undo Management

Take the contentious issue of patching. Not only is it handled automatically and completely transparently to the customer, but it occurs generally eight times more frequently than other data warehouse cloud services, including Snowflake. Automated non-disruptive patching ensures higher levels of security. Upgrades are always enabled, fully automated, with zero customer actions, while eliminating performance regressions during upgrades with SQL plan management. This is simplicity and automation at work.

Upgrades are also customer and application transparent. Autonomous Data Warehouse even prevents performance regression during upgrades with SQL Plan Management requiring "zero" customer actions.

From the customer point of view, Oracle Autonomous Data Warehouse is completely self-driving, elastically self-scaling, self-tuning, self-optimizing, self-securing, and self-healing. Whereas Oracle Autonomous Data Warehouse is at autonomy level 5, Snowflake is at best between autonomy levels 2-3.

Three primary underlying technologies enable this autonomy level in Autonomous Data Warehouse. The **first** is Oracle Database is ranked by far and away the top data warehouse in the



market per Gartner[4] and Forrester[5]. It's a comprehensive converged database that is continually updated and improved every year. Oracle Database provides built-in multi-model data and multi-workloads such as analytical SQL, data warehouses, OLTP, document/JSON, object/XML, Spatial/Graph, blockchain, and AI-Machine Learning. It also inherits all the enterprise data warehouse features in Oracle Database and Exadata including an extensive set of built-in analytics tools such as OLAP, Advanced Analytics, Oracle ML SQL Notebooks, and APEX for low-code Dev, and SQL Developer for database Dev.

The mature Oracle Database engine provides the foundation for many of the automation features described above. Oracle has been working for 15 years on automating capabilities like memory management and query tuning.

The breadth of Oracle Autonomous Data Warehouse also greatly simplifies many analytic problems. For example, rather than relying on an external service for machine learning, Autonomous Data Warehouse has an extensive collection of pre-built machine learning algorithms and includes a notebook designed for data scientists. It includes support for Python and R, the programming languages of choice for data scientists. More importantly, the Oracle Autonomous Data Warehouse includes Automated Machine Learning (AutoML). AutoML simplifies and accelerates Machine Learning implementations with an intuitive "Click-to-Build." Customers can easily use the built-in Machine Learning capabilities by selecting a data set and then choosing a target attribute to predict. AutoML will choose the optimal algorithm by comparing the results from multiple algorithms and automatically tuning the specific subset of data and parameter settings needed for that algorithm. AutoML makes creating machine learning models intuitive to create and deploy with stress-free application integration via REST or SQL without the need for knowledgeable, experienced, and skilled professional data scientists. Autonomous Data Warehouse similarly handles spatial analytics and graph analytics natively within its service, without requiring any additional services, as well as providing complete support for JSON data.

Unlike Snowflake, Autonomous Data Warehouse includes self-service tools designed for data analysts and data scientists. Data analysts now have tools for drag and drop loading; declarative data transformations plus data cleaning; auto-creation of powerful business models; interaction discovery based on spatial and geographic relationships; guided discovery of hidden patterns and anomalies.

Customers can simply drag-and-drop simple Excel files directly into the Autonomous Data Warehouse. It automatically loads different Excel sheets into different tables. It works just as well for larger data sets. The same user interface (UI) leverages the scalable data loading procedures when loading from object storage. And it allows customers to easily load data from other databases (not just Oracle,) they have access to via database links.

Moreover, Autonomous Data Warehouse has embedded all of the highly useful production-proven, easy-to-use, quality capabilities of the Oracle Data Integrator (ODI) with support for many data sources including Fusion, SalesForce, SAP, and many more. These are data sources in high use and high demand. In fact, Autonomous Data Warehouse has more connectors to more sources than any other data warehouse cloud service…period.

These built-in, self-service tools bring the simplicity of Autonomous Data Warehouse to the data analyst so that they can manage their own data. Autonomous Data Warehouse also remains open so that customers can use the tools of their choice. Just about every OLAP, Business Intelligence, and Analytics software (Tableau, Qlik, Looker, Microsoft, SAP Business Objects, TIBC Spotfire, MicroStrategy, and of course Oracle Analytics Cloud) works with Oracle Autonomous Data Warehouse, including all OCI and on-premises applications. Autonomous Data Warehouse data integration works with Oracle GoldenGate, Data Integrator, OCI Data Integration, Informatica, Talend, IBM, Data Virtuality. Snowflake has a much smaller group of compatible applications.

[4] Sources: 2020 Gartner Critical Capabilities for Cloud DBMS Report and Gartner MQ for Data Management Solutions for Analytics
[5] Source: 2020 Forrester Wave: Data Management for Analytics

The **second** unique underlying technology that enables autonomy in Autonomous Data Warehouse is the Oracle Exadata Database Machine. Whereas Snowflake abstracts its data warehouse from the underlying cloud x86 servers, hypervisor, and object storage infrastructure, Oracle took an outside-the-box approach to optimizing performance by providing custom-built database-optimized hardware (Exadata) in its public cloud. Autonomous Data Warehouse runs on Exadata in every Oracle Cloud Infrastructure data center, Exadata Cloud@Customer, and Oracle Dedicated Region Cloud@Customer.

Autonomous Data Warehouse squeezes every possible performance advantage from its Exadata hardware to deliver unprecedented performance, and performance will be further discussed in detail in the performance section below. Exadata is additionally a self-healing infrastructure built for enterprise-class highly availability: triple-mirrored drives for drive failures; automatic transparent recovery for compute node failures; and application transparent online patching with zero database downtime.

The **third** Oracle unique underlying automation technology is also unmatched. It comes from Oracle's more than four decades of refining best practices, AI-machine learning, and policy automation.

A self-healing data warehouse is unique to Oracle. Autonomous Data Warehouse delivers that unmatched capability by continuously monitoring the entire cloud data warehouse environment for 12,000+ metrics and 1,500+ alarms. It automatically files service requests for any detected issues. Any issues identified by monitoring or filed by customers are immediately addressed with an integrated team of cloud, support and development. No customer actions are required. Oracle Autonomous Data Warehouse has averaged four times faster closure of issues than on-prem and has been proven to detect and correct over 87.7% (7 out of 8) issues before the customer becomes aware of them. Those issues are resolved quickly because log files and diagnostics are automatically gathered and immediately accessible. Frequent patching enables bugfixes to be swiftly deployed into production and minimizes any long-term disruptions. Autonomous Data Warehouse patches on average every 12 days, transparently, online, without customer disruption.

This unprecedented automation level is all 'behind the scenes.' It's what distinguishes Autonomous Data Warehouse from simply running the Oracle Database on-premises or in another public cloud. That automation enables high-performance, highly available, and highly secure data warehouses with radically reduced administrative costs. It turns non-experts into experts without requiring knowledge, skills, or experience to create or access a data warehouse. Whereas the Autonomous Data Warehouse autonomy scale is easily classified as a 5, Snowflake is at best, between autonomy levels 2-3 as illustrated in the autonomy definitions previously discussed.

Remember that Snowflake can't enforce UNIQUE key, PRIMARY key, and FOREIGN key constraints, thereby forcing application developers to build this into their applications. Autonomous Data Warehouse does not have this issue.

One more thing, Autonomous Data Warehouse includes APEX. APEX is a very popular low-code application development tool that greatly simplifies application development by orders of magnitude, typically averaging 20X faster development times than conventional approaches.

Performance

Performance is a crucial reason why so many IT organizations move to a data warehouse cloud service. Their underlying assumption is that data warehouse cloud services are performance elastic, on-demand, and scalable beyond what they might conceivably need, and faster when it comes to restores from outages.

Data warehouse cloud service performance based on configurations is a directly metered cost. Computational resources are charged per unit of time consumed and the infrastructure being utilized. Both Snowflake and Oracle charge by the second. The longer it takes for a query to complete, the more it costs. Greater underlying hardware infrastructure speeds queries while adding to the per unit of time metering cost. Finding the right balance between performance and infrastructure can be time-consuming and frustrating. These are known as "below-the-line" costs.

Performance has a direct impact on "above-the-line" costs that can be much greater than the "below-the-line" costs. Above-the-line costs are those related to productivity and revenues. The Doherty Threshold revealed in IBM's "Economic Value of Rapid Response Time[6]" and discussed in the AMC television series, "Halt and Catch Fire," shows that when a computer, applications, and users interact at a pace where neither has to wait on the other, productivity soars and the quality of work tends to improve.

In the case of a data warehouse cloud service, the Doherty Threshold plays out: higher performance enables faster actionable insights. That leads to faster actions and a lower monthly cloud bill. Faster actions in turn lead to faster-time-to-market, resulting in competitive advantage and greater revenues. In summary, data warehouse cloud service performance has an enormous impact on both below-the-line and above-the-line costs.

Snowflake Performance

Snowflake delivers instant performance scaling without disruption or the requirement to redistribute data as long as it's on the more costly Enterprise, Enterprise for Sensitive Data, and Virtual Private Snowflake editions. Auto-concurrency enables DBAs to set min/max cluster sizes. Clusters in the Enterprise or greater Snowflake editions scale elastically and automatically over this range, should there be unexpected demands.

The problem, as previously discussed, is how the customer determines the right infrastructure nodal size and correct min/max range. The reality is that the DBA must try different virtual data warehouse sizes and choose between a single or multi-cluster configuration. As previously noted, Snowflake's virtual data warehouses are deployed on clusters of fixed shape sizes of 1, 2, 4, 8, 16, 32, 64, or 128 nodes. Each node is equivalent to 4 cores per Snowflake virtual data warehouse. Each fixed shape size is twice as large as the previous size and also twice the cost. Choosing the right size takes a bit of time, testing, and guestimates for each virtual data warehouse. Virtual data warehouses can be expanded to the next hardware cluster size easily, on the fly, in milliseconds, and non-disruptively. Each increase in cluster size appreciably speeds up a query or multiple queries within the virtual data warehouse.

However, multi-clusters do not speed up any specific query. Individual queries only have access to the resources within a specific cluster shape, not a multi-cluster. Clusters are virtual compute resources. A specific query's performance cannot exceed that of the cluster size in which it is running. Additional clusters do not increase the resources within a cluster. They increase the resources to the entire virtual data warehouse. Per Snowflake, multi-cluster warehouses are best utilized for scaling resources to improve concurrency for users and queries. They provide no performance improvement for slow-running queries or data loading.

In addition, to meet increasing data warehouse performance demands, multi-cluster data warehouses are weighted to start additional clusters of the same size by default, generally adding to the overall cloud cost. To avoid this scenario, DBAs can over-provision on cluster size (node counts) or build a new virtual data warehouse.

Also, keep in mind that Snowflake's performance is limited by the generic hardware infrastructure available in the public cloud that it runs on. Generic, general-purpose hardware is not optimized for data warehouses (or any other workload). Abstracting Snowflake's virtual data warehouse software from the public cloud hardware infrastructure does not overcome this limitation.

Then there's Snowflake's performance issues with rapid data ingestion, ongoing queries, and reporting. Snowflake best practices recommends splitting Snowflake deployments into two distinct separate virtual data warehouses. The first one is configured as a large size hardware node to speed data ingestion. The second can be a much smaller configuration focused on queries and reporting. This provides an effective workaround while adding significant compute by requiring two distinct separate instances.

Another performance issue is data access from transactional or other sources. Snowflake is a pure data warehouse. It cannot access directly the data in transactional relational databases that represents 80% of the database market. It has to access this data from non-Snowflake sources. Snowflake has extremely

[6] Economic Value of Rapid Response Time

limited native tools to do so. In addition, developing, operating, managing, patching, upgrading, and documenting the ETL scripts takes time and adds significant cost. Most Snowflake users utilize third-party tools or services such as Talend or Hevo that add yet more complexity and cost. In all cases, it takes additional time to implement, meaning that data accessed is no longer in real-time, making it stale and insights out-of-date.

Snowflake asserts that no table partitions and no indexes are a good thing because there is nothing for the DBA to tune. Unfortunately, this can and does reduce the virtual data warehouse's query performance. A single table "n" year of events will take a while to scan. Snowflake's recommended work-around is called clustered tables. It can sort the table data by a field such as the transaction date. Doing this enables Snowflake to use min/max stats of micro-partitions to eliminate the transactions that don't have the correct date range. This accelerates queries that filter on transaction date. Clustered tables improve query performance but increase the cost of both compute and storage resources.

No indexes eliminates having to tune those indexes, simplifying the DBA's life. But it means queries are inefficient and take longer even if clustered tables are already being utilized. Longer queries increase compute time costs. Snowflake recommends "materialized views[7]" that define a copy of the base table clustered by a different key. This enables a faster query. However, the reduction in compute time costs are more than exceeded by additional compute and storage resources required to maintain a full data copy and shadow each DML operation for the base table.

To summarize, Snowflake's customers have to comprehend how their data is used and laid out to get reasonable query performance. This is not trivial. It consumes considerable and ongoing amounts of DBA time. To add insult to injury, Snowflake doesn't provide monitoring reports to enable their customers to optimize their virtual data warehouses. They have to utilize third-party tools. Again, more complexity and cost. Although that additional cost likely pales in comparison to the cost of sub-optimal tables and inefficient queries, those costs have been known to scale into many thousands of dollars (USD).

There is one more consideration with Snowflake's virtual warehouse performance—bulk data load. Modifying just a single field in a single row will cause the whole data block to be copied. That's a big issue. To optimize operations, Snowflake demands bulk inserts. Bigger batches improve the data processing. They also improve the final data layout. Small batch updates can be highly problematic because they require a full table scan for each and every update. The Snowflake engine then creates a copy of each micro-partition, assuming the data layout matches.

Most customers are ultimately most concerned about query performance. Snowflake concurrent queries or sessions are recommended to be no more than eight. Indeed, performance can and does degrade when concurrent queries exceed eight. Snowflake's workaround is for customers to spin up one or more new clusters. Although that will not speed up a query already in progress, it will provide their best performance for multiple concurrent queries above eight. And Snowflake makes it very easy to spin up new clusters. One small detail is that it also doubles the cost for each cluster that is spun up.

Another major Snowflake query performance issue is when the query is poorly written. Poorly written queries have a major negative impact on query performance and reduce concurrency. Poorly written queries tend to be quite common among Snowflake customers. As previously noted, spinning up new clusters does not speed up a query. Poorly written queries in any data warehouse will noticeably degrade performance. This is highly evident with Snowflake. Auto-scaling only nominally helps.

One more thing. Snowflake currently does not support common data warehouse features such as joins, stored procedures, and triggers. These functions accelerate queries. Not having them slows down queries. Important to note that Snowflake does not currently address complex joins. When complex joins are required, they are currently recommending the customer spin up a separate PostgreSQL database. This defeats the purpose of utilizing a managed data warehouse cloud service.

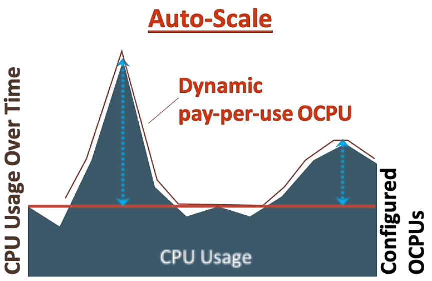
To summarize, Snowflake's performance may appear to be simple because there are no indexes or partitions to tune. Appearances are deceiving. Snowflake's performance is far from optimal and requires noteworthy virtual warehouse manipulation to obtain reasonable levels of performance. Getting that

[7] Material Views are typically used to manage aggregate summaries.

acceptable performance requires too many work-arounds and DBA interventions that take time and add cost. Snowflake's performance is not a strength.

Oracle Autonomous Data Warehouse Performance

Whereas performance is not a Snowflake strength, it is a major strength for Autonomous Data Warehouse. It's purpose-built for performance, elasticity and easy scale. Whereas Snowflake has severe performance issues with concurrency and complex queries, Autonomous Data Warehouse has no such issues, delivering superior concurrency and complex query performance.



Unlike Snowflake which runs on generic hardware, Autonomous Data Warehouse is specifically co-engineered with Oracle Exadata infrastructure that is designed to provide the best possible query performance. Exadata infrastructure running the Autonomous Data Warehouse is unique. No other hardware from any vendor in the cloud or on-premises comes close to matching Exadata's data warehouse performance that scales to hundreds of thousands of GB/s and is optimized for both data warehouses and transactional databases. Scaling is a fundamental Oracle advantage. Data warehouses in the Autonomous Data Warehouse can scale to 25PB (compressed). That is not a hard limit but rather a tested limit. Compute performance and storage scale easily, independently, granularly, autonomously, and application transparently, on-the-fly. Autonomous Data Warehouse's OCPUs or Oracle CPUs (cores) scale automatically up to as much as 3X the configured OCPUs to meet peak demands, and back down when not needed. Customers pay for actual OCPU usage per second. The 3X scalability upwards limitation is to prevent surprise bills. In contrast to Snowflake, there is no need to spin up completely new clusters. No need to guestimate a specific pre-configured fixed hardware size by trial and error. There are no fixed compute sizes to try on or manually upgrade to. Auto-scaling is instantaneous and non-disruptive, adjusting CPU and IO resources based on workload requirements.

Consider that a single data warehouse in the Autonomous Data Warehouse can scale to as many as 4,600 OCPUs – 1,600 database server cores and 3,000 storage server cores that offload many data warehouse tasks from the database servers, 44 TB of DRAM, 1.6 PB of Flash drives and, as previously mentioned, up to 25 PB (Oracle Hybrid Columnar Compression) in data warehouse size.

The synergistic combination of Oracle Data Warehouses on the Oracle Exadata engineered system has proven to be so performant and valuable that currently 86% of the Fortune Global 100 companies run their Oracle databases and data warehouses on Exadata in Oracle Cloud Infrastructure, Exadata Cloud@Customer, and on-prem. Here's why:

Exadata Performance Optimization
Offload to Storage Servers SQL JSON Data Mining Decryption
Smart Scan
Columnar Flash Cache
Storage Indexes
Hybrid Columnar Compression (HCC)
Automated In-Memory
I/O Resource Mgmt
Database Aware Flash Cache
Smart Flash Cache, Log
In-Memory Fault Tolerance
Network Resource Mgmt
ExaFusion Direct-to-Wire Protocol

And even though both Oracle and Snowflake separate storage from compute while autoscaling, Oracle does it at much more granular level and subsequently lower cost. Both shut off computer resources when the data warehouses are idle, so customers only pay for what they use.

Autonomous Data Warehouse automates performance tuning for indexing and partitions. No human intervention is required. That automated tuning is so advanced even the most experienced DBAs cannot match the Autonomous Data Warehouse when it comes to performance optimization.[8]

In-memory joins build on the Oracle deep vectorization single instruction multiple data (SIMD) framework. It leverages hardware acceleration and pipelined execution to optimize operations such as hash join. Hash joins are broken down into smaller operations that can be passed to the vector processor. Then the key value table used is SIMD-optimized and used to match rows on the left and right of the join. This delivers up to a 10X improvement in performance vs Snowflake.

SQL macros for scalars and tables are another way that Autonomous Data Warehouse accelerates performance. SQL scalar macros are a very simple way to encapsulate complex SQL expressions. They act like a SQL pre-processor with reusable code transparent to the SQL optimizer. This makes it easy to create reusable and portable code without expensive context switching. SQL table macros encapsulate the SQL used in the FROM clause.

Load performance is another Autonomous Data Warehouse advantage over Snowflake. Oracle has developed hundreds to thousands of connectors to data sources. These built-in connectors eliminate the need for ETL or ELT tools for most data warehouses. If the data source is one of the rare ones for which Oracle does not have a built-in connector, they have built-in tools, such as Golden Gate, that make it easy to load the source data. Being multi-model, multi-workload, multi-source, and massively parallel enables Autonomous Data Warehouse to ingest and load data orders of magnitude faster than Snowflake.

Based on TPC-DS benchmark tests[9] run by Fivetran – a data pipeline provider that syncs data from apps, databases, and file stores into the customer's data warehouse – on Snowflake and AWS Redshift, Snowflake ranged from approximately 2 to 3X faster than Redshift. A separate test running that same TPC-DS benchmark on Autonomous Data Warehouse revealed performance approximately **4 to 7X better** performance than Snowflake. Snowflake customers that have switched to Autonomous Data Warehouse have found their queries perform at least 2-3 times faster.

Autonomous Data Warehouse leverages the Exadata infrastructure to provide the best data warehouse performance possible in all aspects. No data warehouse cloud service or data warehouses on-prem can match Autonomous Data Warehouse on Exadata elasticity, non-disruptive upgrades, sheer scale and performance. It's not close.

Data Warehouse Cloud Service Security

Security in a data warehouse cloud service or, indeed, any cloud service is paramount today. Between constant cyber-attacks, malware, ransomware, privacy laws, and regulations, security is more important than ever before. A public data warehouse cloud service is generally a shared data center ecosystem. It is essential that no customer be able to access another customer's data without their permission. It is just as essential that no public cloud administrator or employee access a customer's data. And although not directly connected to security, it's nearly as important to avoid the noisy neighbor. In this interconnected world, never has cyber security been so important.

Snowflake Security

Security in Snowflake can be described in one word—basic. It includes access control, single sign on (SSO), authentication, and encryption. Although Snowflake also offers subsetting, data masking, and redaction (policies that enable full or partial field redaction based on access role), they are primitive. Snowflake customers describe them as a "work-in-progress." What that means is that Snowflake's security capabilities tend to be quite limited. Snowflake ***lacks*** capabilities that many IT organizations consider table stakes for data warehouse cloud service security, such as:

[8] Auto-tuning based on best practices, AI-machine learning, and policies was able to provide better tuned more effective and efficient indexes and partitions in 24 hours, than NetSuite had been able to provide over a 15-year period.

[9] TPC-DS is an industry-standard benchmarking meant for data warehouses.

- Fine-grained access control
- Fine-grained auditing
- Row-level security
- Column-level security
- Label security
- Sensitive data discovery
- User risk scoring
- Real application security
- Database firewall

Another concerning Snowflake security deficit is when customers require all or some of their data to remain on-premises. Snowflake does not support data residency on-premises at this time.

Then there's the issue of tenant isolation. Snowflake virtualizes on top of a virtualized shared infrastructure from AWS, Azure, or GCP. The hardware infrastructure is shared among users. The hardware security model is shared. Snowflake does not isolate tenants from one another at the hardware level. It cannot be run on private cloud infrastructure. Snowflake tenant isolation is at the virtual warehouse level. Each and every customer gets their own virtual warehouse. This achieves a pretty good, but not ideal average CPU utilization while not eliminating the noisy neighbor. Tenant isolation is imperfect at best.

Snowflake's security at this time is at best rudimentary and insufficient for mission or business-critical data. It has improved but requires significantly more improvement.

Oracle Autonomous Data Warehouse Security

Oracle Autonomous Data Warehouse is self-securing. It delivers automated strong, rock solid, production-hardened, and proven security. The security is end-to-end unified and integrated across the entire stack of services including applications, database, virtualization, operating system, and Exadata engineered systems. There is a complete tenant stack isolation and tenant application data isolation. Once again, this is unique to Oracle.

The "autonomy" in the Autonomous Data Warehouse means a major decrease in human labor. No human labor translates into no human errors and no overlooked, exposed vulnerabilities. By definition, no human errors improve security.

Frequent patching (on average every 12 days) minimizes vulnerability in attack surfaces. Fewer attack surfaces increases security.

Oracle Autonomous Database's built-in security ***includes***:

- Data Safe
- TDE (Transparent Data Encryption)
- Database Vault
  - Prevents privileged users from accessing business data
- Always encrypted
- Intrusion detection
- Fine-grained access control
- Fine-grained auditing
- Row-level security
- Column-level security
- Label security
- Comprehensive data masking
- Data redaction
- User risk scoring
- Sensitive data discovery
- Real application security
- Blockchain Tables
- Database firewall

Data Safe is the unified control center for the Autonomous Data Warehouse. It identifies and calls out data sensitivity, empowers data risk evaluations, enables wide-ranging sensitive data masking, implements and monitors security controls, assesses user security, monitors user activity, and addresses data security compliance requirements. Data Safe is architected to deliver essential data security capabilities that reduce risks and improves security.

Oracle Database 21c and 19c added support for Autonomous Data Warehouse built-in Blockchain Tables at no additional cost. Oracle Blockchain Tables prevent illicit changes from insiders, hackers, authoritarians, and impersonators. They are incredibly easy to implement and manage for both the expert and non-expert alike. Blockchain has become an increasingly important security capability for making sure there are no unauthorized changes to the data. No other data warehouse cloud service has any equivalent to Blockchain Tables.

When it comes to data warehouse security, Autonomous Data Warehouse sets the bar others are measured against.

Data Warehouse Cloud Service Flexibility

Data warehouse cloud service flexibility refers to the ability to run multi-sources, multi-models, and multi-data types. Those data warehouse cloud services with more flexibility require less effort on the customer side to build useful data warehouses.

Snowflake Flexibility

Snowflakes biggest marketing claim is their ability to share data between any Snowflake customers, thus, minimizing data copies and data movement issues. That’s just one aspect of data warehouse cloud flexibility. Unfortunately, deployment flexibility is not available with Snowflake. It cannot be deployed at this time either as a service on-prem or as a subscribed license on-prem. Snowflake has talked about the possibility of supporting AWS Outposts down the road.

There are several other flexibility shortcomings including support for multiple data types, multiple data sources, and multiple applications that work seamlessly with the data warehouse. In these areas, Snowflake comes up woefully short.

Snowflake supports structured data, semi-structured data, unstructured data, and some graphical and spatial data. But Snowflake currently only supports one coordinate system. It is missing native support for several other unstructured data types including 3D/LiDAR and raster. It is also missing built-in tools for map visualization. Material views are completely manual, as previously discussed, requiring extensive DBA knowledge, skills, and experience. Snowflake also currently lacks self-service visual analytics. Snowflake is severely incomplete when it comes to unstructured data. Those limitations constrain customer use cases including their ability to gain insights using unstructured data.

Snowflake is unique in that it can run in multiple public clouds including AWS, Azure, and GCP. It can even use multiple public clouds for use cases such as DR. Although this can cause quite a large storage egress charge if the customer’s virtual data warehouse and the data are in different public clouds. And Snowflake will pass on those charges to the customer.

Snowflake’s ability to load data from AWS S3, Azure Blob, and Google Object Storage is excellent. However, Snowflake’s ability to load data from transactional or third-party data is generally mixed to poor. There is a dearth of built-in applications requiring most Snowflake customers to rely on third-party tools for analytics, machine learning, and more. This requires them to have multiple vendor and product contracts to manage, integrate, operate, and troubleshoot when issues occur.

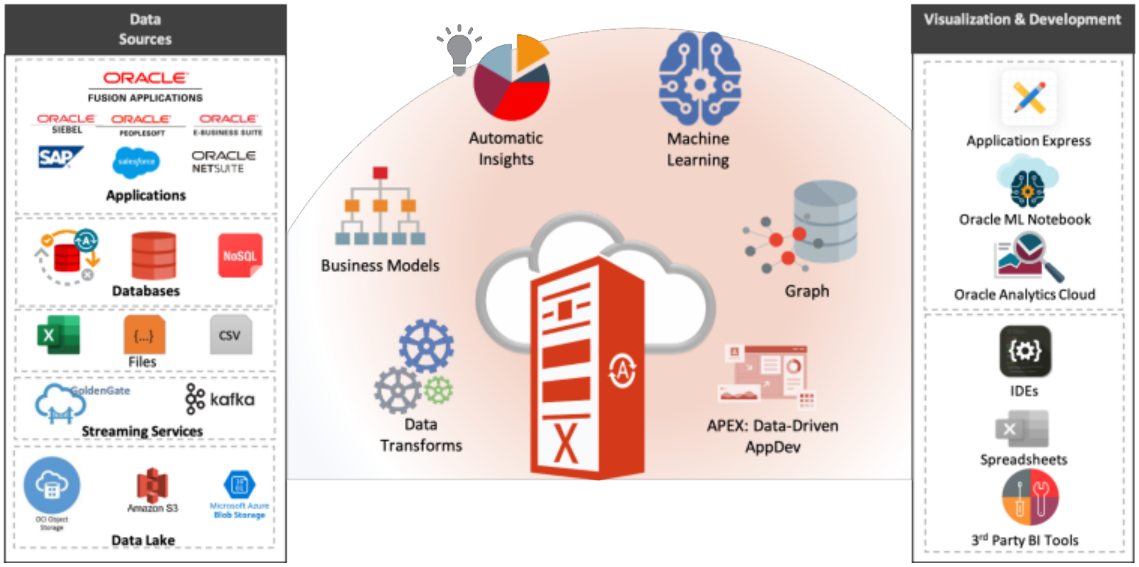
Snowflake’s flexibility is much more limited and rigid than it first appears.

Oracle Autonomous Data Warehouse Flexibility

Oracle Autonomous Data Warehouse is like an Olympic Triathlete comparison. It is a very complete converged database engine. This means it supports a broad range of structured, semi-structured, unstructured, and complete spatial and graph data. Data from any of the other database models are easily ingested into Autonomous Data Warehouse. Thousands of applications work transparently with Autonomous Data Warehouse, including all of Oracle’s applications. APEX, a low-code application development tool is built-in, making it incredibly easy to rapidly build efficient applications that utilize

Autonomous Data Warehouse, such as the current COVID-19 vaccine applications in use by the U.S. government.

Autonomous Data Warehouse has an incredibly rich ecosystem of database types, models, tools, and applications that are second to none. This is why Snowflake ***melts*** when compared to the Autonomous Data Warehouse.



AutoML – automatic AI-machine learning – is also built into Autonomous Data Warehouse. It empowers no-code AI analytics for data analysts and citizen data scientists. Automated algorithm selection identifies the best prediction algorithm for each workload. Automated feature selection identifies the data that best predicts outcome. Automated model tuning identifies the model parameters for best performance. And if the customer does not want to utilize these automated tools, they can choose not to.

Flexibility includes in-memory dual format. Both row AND column formats in the same table. Simultaneously active and transactionally consistent. Data warehouses, analytics, reporting, and time series use in-memory column formats. Whereas OLTP uses a proven row format. This enables Autonomous Data Warehouse to be utilized for multiple database models without ever extracting or transforming the data. That's huge.

When it comes to deployment flexibility, customers have several options. Autonomous Data Warehouse is available on OCI, on-premises with Exadata Cloud@Customer, and in Dedicated Region Cloud@Customer.

In summary, Autonomous Data Warehouse defines flexibility, and it gets continuously better.

Data Warehouse Cloud Service Cost and Cost Controls

The cost of data warehouse cloud services is always a concern for every customer. There are several components to that cost. The primary components include cost of the CPU time consumed based on the number or shape of the compute resources utilized and cost of the stored data. Those costs are generally predictable; however, they are not the only costs.

There are also the costs of managing, operating, tuning, and troubleshooting the data warehouse cloud; optimizing queries; loading data from external sources; writing scripts; building complex SQL statements; third-party tools for data loading, reporting, applications, etc.; and potential cloud storage egress fees. All of these additional costs are considerably less predictable.

Both Snowflake and Oracle take very different paths to cost and cost control.

Snowflake Cost and Cost Controls

Snowflake pricing depends on how the following three services are being utilized by the customer:

- Data Storage.
- Virtual Warehouses (compute).

- Cloud Services – Snowflake tools and services.
  - Third party services are not included in Snowflake rates.

Data storage costs are based on on-demand consumed or pre-paid capacity. Each has separate storage pricing. On-demand storage has a charge of $40 per TB per month. Pre-purchased storage has a charge of $23 per TB per month, consumed as they go. When the customer's pre-purchased storage runs out, they have either to purchase more or convert new storage consumption to on-Demand. All Snowflake storage is public cloud S3 compatible object storage. It is not flash SSD or HDD block storage. Object storage by definition that's slower than block SSDs or HDDs.

Virtual warehouse compute time costs are tied to the different nodal cluster shapes of 1 (X-Small), 2 (Small), 4 (Medium), 8 (Large), 16 (X-Large), 32 (2X-Large), 64 (3X-Large), or 128 (4X-Large) nodes. Snowflake also has five different editions/price points for each cluster shape, each with more features than the last.

Standard	Premier	Enterprise	Business Critical	Virtual Private Snowflake (VPS)
Complete SQL DW	**Standard +**	**Premier +**	**Enterprise +**	**Business Critical +**
Data Sharing	Premier support 24x365	Multi-cluster warehouse	HIPAA support	Customer dedicated virtual servers wherever the encryption key in memory
Bus hr support M-F	Faster response time	Up to 8 days of time travel	PCI compliance	Customer dedicated metadata store
1 day of time travel	SLA w/refund for outage	Annual rekey of encrypted data	Data encryption everywhere	Add'l operational visibility
Always-on Enterprise grade encryption in transit & @ rest		Materialized Views	Tri-Secret Secure using customer managed keys	
Customer dedicated virtual warehouses			AWS PrivateLink support	
			Enhanced security policy	

Each node size increase doubles the cost of the previous size regardless of the edition. This is why whenever a new cluster is spun up, it significantly increases the cost. Each larger Snowflake cluster size is twice the cost of the preceding size. Each edition upgrade also raises the cost per hour (billed per second). Premier is 25% higher than Standard, Enterprise is 20% higher than Premier, and Enterprise for Sensitive Data is 33% higher than Enterprise. Moreover, VPS requires a quote because it can cause sticker shock. Below are Snowflake's compute rates:

Snowflake Compute/Hr in USD $ - Billed/Second

Snowflake Compute	Snowflake Nodes	Standard	Premier	Enterprise	Business Critical
XS	1	$2	$2.50	$3	$4
S	2	$4	$5	$6	$8
M	4	$8	$10	$12	$16
L	8	$16	$20	$24	$32
XL	16	$32	$40	$48	$64
2XL	32	$64	$80	$96	$128
3XL	64	$128	$160	$192	$256
4XL	128	$256	$320	$384	$512

Snowflake's compute rates are not low by any means. Their billing methodology is a bit complicated. Users buy Snowflake credits. Credits are consumed at different rates depending on the Snowflake compute size utilized and Snowflake serverless services utilized. In Snowflake's nomenclature, a node is a credit. Therefore, a 4XL consumes 128 credits per hour versus a XS that consumes 1 credit per hour. The above chart breaks down the compute consumption rates for the US region. Rates vary based on the public cloud region in which the virtual data warehouse is located.

As previously discussed, concurrent queries beyond eight require another cluster to maintain performance. Complex queries reduce the number of concurrent queries, thereby requiring more clusters. Auto-clustering causes materialized views to frequently double in cost. Virtual data warehouse clusters can and do rapidly and surprisingly escalate costs.

Snowflake attempts to limit compute costs by not charging for idle time. Complex queries either take longer—costing more, or require bigger sizes, also costing more. When there are no queries going on in the Snowflake virtual data warehouses, there are no compute charges. Snowflake also offers a 30-day free trial with up to $400 in usage credits.

Customers who wish to move or copy their data between regions or clouds pay data transfer charges. These charges, per Snowflake, are passed through from the respective cloud provider with no Snowflake markup. Snowflake serverless tools are charged in a similar manner to warehouses on a computational basis. The Snowpipe continuous data ingestion tool to load data is one of Snowflake's few tools. Third-party tools have separate rates above and beyond Snowflake charges.

Storage costs can also escalate quickly through several means. Time Travel is a good example. A retention period of 90 days has been found by Snowflake users to increase base table capacities by approximately 200X. This has a lot to do with how data is loaded into a table. Storage capacity consumption is likely to rapidly increase exponentially when there are lots of inserts, updates, data copies created across partitions and clustering—all adding up to more cost. Snowflake suggests keeping these costs down by carefully mapping out the data load strategy and shortening Time Travel retention periods. This is not trivial. It requires a deep understanding of the data being loaded, which takes time and is subject to human error.

Remember Snowflake is supposed to be simple. Spending excessive time mapping a data load strategy is not a likely scenario for customers lacking the expertise or understanding in how it impacts cost. They will not want to spend the time and will be unaware of the upcoming cost impact until they get their bill. In addition, shortening the Time Travel retention period limits the customer's analysis horizon, giving them less useful insights. Neither is a good work-around.

Snowflake storage costs are somewhat mitigated by using their automated columnar compression. That compression reportedly gets a compression ratio range of between 3X to 6X.

Since Snowflake does not own its own and operate its public cloud, it means that there's a middleman who also must make a profit. That puts pressure on Snowflake's cost and pricing. Moreover, Snowflake has no control over the cloud infrastructure cost they utilize. If AWS unexpectedly raises its prices by 20%, then Snowflake either takes a material margin hit or has to pass the price hike down to their customers as well.

What surprises too many Snowflake users are the unexpected indirect costs. For example, when they want to utilize advanced analytics and machine learning they're not available from Snowflake. Customers have to go to a third-party provider, pay their subscription fees, integrate it with Snowflake, and work with multiple vendors. As a result, it take much longer to get valuable actionable insights. That slows productivity, time-to-market, and time-to-revenue.

Third-party ETL, analytics, and reporting tools add significant costs. It's common for ETL tools to run $10,000 per user per year. Analytics and reporting tools typically cost $1,000 per user per year. This adds up quick, particularly in a company with 10s or hundreds of thousands of employees. Migration to the Snowflake data warehouse and integration also add costs. With Snowflake, these costs commonly rise very quickly. Depending on the capabilities and data in the on-premises data warehouse migrating to Snowflake, these costs have been known to grow to hundreds of thousands of dollars and even into the millions. Those costs add up quickly. Snowflake additionally does not have any built-in low code development tools at this time. That's yet another third-party tool that will have to licensed and learned separately.

Snowflake has no built-in advanced analytics. Likewise, Snowflake offers no built-in AI-ML features, let alone automated machine learning. Snowflake customers that require any advanced analytics must subscribe or license from third-party providers such as Alteryx, AWS SageMaker, Big Squid, Dataiku, databricks, DataRobot, Qubole, or others. Snowflake is also missing ML Notebooks. These products and services are expensive, adding significant indirect costs to Snowflake's direct costs. That's not all. The customer must learn a different interface, tools, and support process for each of these third-party software and service offerings.

Another cost is the time lost from outages. Snowflake is generally fast enough for simple queries. The company promises 99.9% uptime or approximately 8 hours and 55 minutes of downtime per year. That's nearly 1,800% more downtime per year than Autonomous Data Warehouse at 99.995% uptime or no more than 30 minutes per year. In addition, Snowflake requires the more costly premier support or higher for the 99.9% uptime guarantee. That uptime guarantee is not available with standard support. Because Snowflake runs on other vendor's public clouds, an outage in one of the underlying public cloud regions will result in some downtime as a switchover to the failover region takes place.

Snowflake positions its service as "budget friendly." For simple small data warehouses, that's likely to be true. For anything else its costs are definitely not budget-friendly. These costs are why many Snowflake users have moved their data warehouse service to Autonomous Data Warehouse. They generally found Snowflake costs 2 – 3X more than Autonomous Data Warehouse.

Oracle Autonomous Data Warehouse Cost and Cost Controls

There are two fees for Autonomous Data Warehouse. The first fee is the OCPU per hour charge of $1.3441—billed per second. This charge is per OCPU. The second fee is for either the Exadata storage cost on shared Exadata infrastructure, or the Exadata infrastructure when using dedicated Exadata Infrastructure. Exadata storage is $118.40 per TB per month on shared Exadata infrastructure. Exadata dedicated infrastructure fees depend on the size of the rack—basic, quarter rack, half rack, or full rack—and the number of racks for very large systems.

Oracle Autonomous Data Warehouse Rates

Exadata Infrastructure	Shared	Dedicated
OCPU per/hr	$1.34	$1.34
Exadata Storage / TB	$118.40	
Exadata 1/4 Rack X8 / hr		$14.52
Exadata 1/2 Rack X8 / hr		$29.03
Exadata Full Rack X8 / hr		$58.06

At first glance, these rates appear to be higher than Snowflake. However, lower rates do not necessarily translate into lower costs. Here's how the cost of Autonomous Data Warehouse is significantly lower than Snowflake.

Oracle does not charge for idle time on the compute side the same as Snowflake. Oracle compute rates are billed per second and tied to Oracle CPUs (OCPU). An OCPU is a core. Customers can configure as low as one core or thousands. They do not have to guess a specific size for their data warehouse. Autonomous Data Warehouse will determine and configure it for them based on their specific business requirements. Their data warehouses can automatically scale down or up. Auto-scale enables the customer's data warehouse to scale 3X automatically based on the performance needs of the moment. It also scales back down when those OCPUs are no longer required. Customers can turn off Auto-scale if they're okay with their performance and do not want to pay for additional scaled up OCPUs. It's analogous to turning off the over-draft on a checking account.

Major and substantial cost savings comes from the comprehensive automation in Autonomous Data Warehouse. There are no levers to pull or knobs to turn. Autonomous Data Warehouse is fully automated. It's self-driving, self-tuning, self-healing, and self-securing. That saves a lot of time upfront and ongoing. It eliminates painstaking troubleshooting and costly human errors. It saves time and money directly.

Much of Autonomous Data Warehouse's cost savings comes from its peerless performance. Queries that complete faster use less OCPU time. Less time equals less OCPU fees charged. This directly correlates to "time is money."

Oracle further controls cost with their free tier. This is an excellent option for developer devops and testdev. That free tier comes with all Autonomous Data Warehouse capabilities. A key cost value of an Autonomous

Data Warehouse subscription is the 60 days of backup retention and point-in-time recoveries: no additional charges even for storage.

Even though Autonomous Data Warehouse Exadata primary storage fees—or dedicated Exadata infrastructure fees—are higher than Snowflake object storage fees, actual storage costs generally come in noticeably less than Snowflake storage costs. Part of that is because of the automatic Hybrid Columnar Compression (HCC) that reduces storage costs by 10 – 15X. HCC reduces data warehouse storage commonly 3X better than Snowflake while customers are not charged at all for storage used for 60 days of backup retention.

Those are direct costs tied to the service. Indirect cost savings versus Snowflake are dramatically lower. Migrating from existing on-premises data warehouses to Autonomous Data Warehouse is a good example. It's much easier than Snowflake, especially if the migration is coming from an Oracle Database. Remember that Oracle Database is a converged database. Accessing data from a different database model such as transactional is simplicity itself. Moreover, for databases not part of the customer's subscription, Autonomous Data Warehouse comes with zero cost, zero downtime migration tools including the data pump migration utility and GoldenGate. There are also low-cost Enterprise data transfer services available such as the OCI Data Transfer Appliance for large data sets. This service enables large data sets to be migrated much faster than over the wide-area-network. Faster large data set migrations from on-prem or other clouds means faster time-to-actionable insights and faster-time-to-market. That equates into unique and faster revenues and/or cost savings. For non-Oracle systems, Oracle provides both the SQL Developer Migration Workbench and GoldenGate Cloud Service for dual running.

These migration tools and services are designed to migrate data warehouses quickly, efficiently, and cost effectively. Unlike Snowflake, Autonomous Data Warehouse does not require strategic planning to minimize storage costs. It's all done automatically for maximum optimization.

One more significant cost savings comes from Autonomous Data Warehouse's integrated and proven advanced analytics and AutoML at zero cost. As previously noted, it comes with that very large library of machine learning algorithms, ML Notebooks With R, Python and SQL functions combined with PL/SQL and Python. All of this enables customers, teams, and data scientists to collaborate to build, evaluate and deploy predictive models and analytical methodologies in Autonomous Data Warehouse. It saves enormous amounts of time and effort in accelerating more desirable outcomes. So not only do customers save money by not paying for third-party AI-ML tools and services, but they also save a lot of time. And time is money.

Autonomous Data Warehouse accelerates time-to-actionable-insights, and, in turn, time-to-market. Those lead to significantly greater unique revenues unobtainable without those gains in time. These revenue gains are frequently much greater than the cost of Autonomous Data Warehouse.

Combining Autonomous Data Warehouse's much lower net costs with its much higher performance demonstrates an overwhelming cost/performance ratio advantage versus Snowflake. Many customers that have switched from Snowflake to Autonomous Data Warehouse state that they are paying ½ or less than what they were paying for Snowflake.

Conclusion

It is crystal clear that when comparing Snowflake to Autonomous Data Warehouse, Snowflake comes up seriously short. Autonomous Data Warehouse has extensive advantages:

- A more complete vision
- Better vision execution
  - Simpler
  - More automation
  - More reliable
  - Significantly greater levels of elasticity
  - 100% non-disruptive online patching
  - Dramatically better performance
  - Better scalability
  - Tighter more complete security
  - Greater flexibility

- Lower costs
- Much better cost/performance

Snowflake sells its services based on the three customer benefits: simplicity, data sharing, and low cost. Snowflake's simplicity comes with architectural tradeoffs such as never being able to do transactions, limited enforcement, and longer query times. Remember that a hammer is a simple tool that makes a terrible wrench. Snowflake's simplicity narrows its use cases and ultimately increases its costs. Moreover, cost tends to be significantly higher than they project because of all the capabilities Snowflake does not and cannot support. Snowflake is a relatively new, immature, and evolving data warehouse cloud service. It has a long way to go compared to Oracle Autonomous Data Warehouse.

Autonomous Data Warehouse is also a relatively new cloud service. It too is evolving. However, Autonomous Data Warehouse is based on production-hardened Oracle Database technologies proven to be the best in the market on a global scale, as well as best practices derived from Oracle's forty years of mission-critical deployments. Even Gartner and Forrester rate the Oracle Autonomous Data Warehouse well above Snowflake.

There is no data warehouse cloud service in the market today, including start-ups, as automated, simple, performant, or as cost effective as Oracle Autonomous Data Warehouse.

For More Information on the Oracle Autonomous Database Services

Go to: Oracle Autonomous Database

Go to: Oracle Autonomous Data Warehouse Database

Paper sponsored by Oracle. **About DSC:** Marc Staimer, as President and CDS of the 22-year-old DSC in Beaverton OR is well known for his in-depth and keen understanding of user problems, especially with storage, networking, applications, cloud services, data protection, and virtualization. Marc has published thousands of technology articles and tips from the user perspective for internationally renowned online trades including many of TechTarget's Searchxxx.com websites and Network Computing and GigaOM. Marc has additionally delivered hundreds of white papers, webinars, and seminars to many well-known industry giants such as: Brocade, Cisco, DELL, EMC, Emulex (Avago), HDS, HPE, LSI (Avago), Mellanox, NEC, NetApp, Oracle, QLogic, SanDisk, and Western Digital. He has additionally provided similar services to smaller, less well-known vendors/startups including: Asigra, Cloudtenna, Clustrix, Condusiv, DH2i, Diablo, FalconStor, Gridstore, ioFABRIC, Nexenta, Neuxpower, NetEx, NoviFlow, Pavilion Data, Permabit, Qumulo, SBDS, StorONE, Tegile, and many more. His speaking engagements are always well attended, often standing room only, because of the pragmatic, immediately useful information provided. Marc can be reached at marcstaimer@me.com, (503)-312-2167, in Beaverton OR, 97007.