ESTUARY

The Real Data Warehouse Benchmark Report

Executive Summary

Organizations, both big and small, often face painful migrations after discovering that their chosen data warehouse cannot scale economically or deliver consistent performance for their specific use cases.

Realizing the full potential of any data warehouse requires an implementation horizon of 8 to 24 months. Consolidating structured and unstructured, live and static data under one roof, optimizing configurations to handle billions of rows, and building institutional knowledge to recoup ROI takes a significant amount of time.

This benchmark report will save you at least a few months and a lot of engineering effort, thereby helping you make an informed decision.

Snowflake, Databricks, Google BigQuery, AWS Redshift, and Microsoft Fabric are comprehensively evaluated in this report.

Note: A few data warehouses that are not mentioned in this list were also evaluated by us, but they could not keep up with our heavy-duty tests. Therefore, we understood they are not heavyweights (despite their claim) and decided not to include them in this report.

Our testing methodology strives to be as impartial as possible; we have used the globally recognized TPCH_SF1000 dataset. However, we created our own tables from scratch with no indexes or special configurations and hydrated them using Estuary Flow – a data integration platform that can handle large-scale streaming and batch workloads efficiently. Where possible, we also opted not to use any kind of caches.

We were alerted by industry professionals that most data warehouses are highly customized to handle TPCH queries; therefore, we created our own benchmark queries and got them reviewed and approved by industry experts.

If you are going to benchmark data warehouses before procuring them, we highly recommend that you simulate your own dataset and queries. The stock datasets and queries will land you in the false positive zone, and will lead to massive wastage of budget, time and effort.

Our findings reveal significant performance and cost variations, with certain warehouses excelling in specific workload patterns while struggling in others.

After reading our report, we hope you make a decision based on long-term TCO vs. initial free tier offering and avoid common pitfalls like selecting platforms incompatible with existing tools and skills, failing to account for concurrent querying demands, and discovering AI-driven performance degradation when data volumes gro

Table of Contents

1 Estuary

1.1 Company Overview

Estuary is a real-time data integration, Change Data Capture (CDC), and ETL/ELT platform designed to simplify and accelerate data movement across diverse systems. With millisecond latency and high throughput, Estuary empowers businesses to unify batch and streaming data pipelines for analytics, operations and AI applications. The platform is purpose-built for reliability, scalability and ease of use, making it ideal for organizations managing complex data workflows.

We offer a flexible 30-day free trial and do not require a credit card when signing up. You could start moving data in just minutes – jump right in and experience our platform without any upfront commitment. Our transparent pricing ensures cost predictability without hidden fees. The platform delivers exceptional performance throughput exceeding 7 GB/s per singular data flow, while providing end-to-end CDC capabilities that capture change data directly from transaction logs. Implementation is streamlined through zero-code pipelines with hundreds of pre-built connectors, intelligent schema evolution, and flexible multi-cloud deployment options with secure private storage.

Estuary eliminates unpredictable pricing with straightforward usage-based costs that make budgeting reliable. Our ecosystem provides unrestricted access to the latest connector versions, ensuring compatibility and preventing vendor lock-in. The intuitive interface reduces setup time from months to days, allowing teams to focus on insights rather than infrastructure management.

2 Foreword

2.1 Foreword by CEO

As Estuary's CEO and co-founder, I've had the unique vantage point of witnessing how strategic data warehouse decisions can propel an organization's analytical capabilities. I've also seen how misguided choices can lead to costly setbacks and demoralized teams. This benchmark report has emerged from numerous customer conversations where organizations found themselves constrained by inflexible data warehouses that failed to scale with their unique requirements or deliver reliable performance without budgetary strain.

The data warehouse ecosystem has grown increasingly intricate, with vendors often promoting ambitious claims that rarely align with real-world operational demands. Our objective with this independent benchmark is to dismantle the marketing hype and deliver unbiased, reproducible performance metrics and cost analyses across leading platforms, empowering you with the knowledge to make confident decisions.

We rigorously tested each platform using identical queries, datasets and methodologies to establish a standardized evaluation framework that exposes its genuine strengths and weaknesses. These insights embody our dedication to transparency in an industry frequently clouded by obscured trade-offs between performance, cost and usability.

Dave Yaffe

2.2 Foreword by Head of Data

Choosing the right data warehouse isn't just a technical task – it's a strategic crossroads for your business. Over years of wrangling massive datasets, standing up complex infrastructures and translating insights into real-world actions, I've learned how profoundly this decision shapes your entire data ecosystem.

What starts as a seemingly simple infrastructure choice quickly ripples through every part of your organization, from engineering velocity and analytical depth to your company's agility in seizing market opportunities. A good data warehouse doesn't merely store data – it accelerates innovation, empowers deeper analysis, and allows your teams to respond rapidly to evolving customer needs.

This benchmark cuts through vendor hype and glossy marketing slides to deliver what data practitioners genuinely need: rigorous, real-world performance data. We've tested warehouses across diverse workloads – linear, parallel, and concurrent queries – and paired those insights with practical cost considerations. The result is an unbiased, comprehensive guide to help you match the right warehouse to your business's unique demands.

As you dive into the results, ask yourself: How will each option integrate into my existing workflows? Will it enhance my team's analytical strengths? Can it scale seamlessly as our ambitions grow? At Estuary, we've repeatedly seen the powerful business transformations that occurs when you strike the perfect balance between cost efficiency and blazing fast performance. That's why we created this benchmark – to equip data leaders with clear, actionable intelligence for smarter decisions.

Dani Palma

3 Fair Comparison: No Hyper-Tuning

In this data warehouse benchmark report, we have adopted a neutral and transparent approach to ensure fairness and comparability across all evaluated platforms. No data warehouse was hyper-tuned, custom-indexed, or optimized in a way that would give any system an unfair advantage. Each platform was tested "as-is," using its out-of-the-box configuration and default settings.

This approach was intentionally chosen to reflect real-world usage during initial evaluations or proof-of-concept stages, where teams often rely on default setups before investing in deep optimization. By avoiding platform-specific tuning, our goal was to provide a level playing field and present an unbiased view of baseline performance, usability and efficiency across systems.

4 Benchmark Report Introduction

4.1 Data Warehouses Featured in This Benchmark

The data warehouses featured in this benchmark are:

- Snowflake
  - Standard Edition: Small, Medium and Large
- Redshift
  - RA3.Large Node-2 and DC2.8XLarge Node-2
- Databricks
  - Classic Edition: Medium (max Node-1), Large max (Node-1) and Xlarge (max Node-1)
- Microsoft Fabric (Azure Synapse Analytics)
  - DW3000c, DW1500C and DW500c
- BigQuery
  - Serverless

During the initial phase of our benchmark process, we added as many data warehouses as possible into the pool based on data warehouse companies' claims that they handle the TPCH SF1000 dataset with ease. However, after loading close to 1TB of data into their systems, we found out their limitations firsthand and decided to eliminate some from our benchmark report.

If you have a data warehouse in mind but don't see it in our report, there is a high chance that we considered it initially. However, due to its scalability limitations, we had to eliminate it from this report.

4.2 Benchmark Objectives

Our primary objectives for this benchmark were to understand

- Response times of queries

- Cost of running queries
- General quality and capabilities of data warehouses

Our secondary objectives were to understand

- Ease of data ingress into the data warehouses
- The ecosystem revolving around the data warehouses
- The feasibility of maintaining a data warehouse in a business setting

5 GitHub Repo Link

Every organization deserves clarity when selecting a data warehouse. We are proud to champion transparency in an industry often clouded by marketing claims and proprietary benchmarks. By making our benchmarking framework completely open source, we are not just sharing tools, we are transforming how the industry evaluates technology.

Our mission is to empower organizations with unbiased and verifiable data. No black boxes, no hidden methodologies, just clear metrics you can trust and reproduce yourself. We believe transparency drives better decisions and ultimately benefits the entire data community.

Access our codebase in our public repository link. With a simple setup process, you can run the same benchmarks we do and see exactly how different warehouses perform with your specific workloads.

6 Dataset Used

For this benchmark report, we have selected the world-renowned TPC-H SF1000 dataset as the foundation for our testing. We deliberately chose a scale factor (SF) of 1000, which equates to approximately 1TB of raw data. Our goal was to push modern data warehouses to their limits, testing not just their ability to run complex SQL queries, but also how they handle large-scale data volumes under real-world stress conditions.

Recognizing the increasing importance of semi-structured and unstructured data in analytics workflows, we extended the dataset to include JSON-formatted data alongside traditional structured tables. This addition allows us to benchmark how well various platforms support querying nested and flexible schemas, without requiring any transformation or schema flattening.

This approach acknowledges the evolving requirements of data teams, who increasingly need to work with raw data in its native form to enhance agility and accelerate time-to-insight. By combining the established rigor of the TPC-H standard with the practical reality of modern, multi-modal data formats,

this benchmark aims to provide a more holistic and future-oriented evaluation of data warehouse performance.

7 Benchmark Queries

Given the widely spread rumor that data warehouses are hyper-tuned to handle standard TPCH queries, we decided to develop our own queries. We looked into common SQL-based requirements from business heads and mimicked them into our benchmark queries.

Our benchmark queries can be found here link.

Our queries are composed of everyday aggregate functions used by almost all data analysts, as well as complex semi-structured joins used in critical use cases to transform data on the fly. We have also deployed computationally heavy text processing and window functions, CTEs, and nested queries.

We strongly believe that our queries handle a broad spectrum of use cases, like e-commerce, as well as domains such as finance, healthcare, telecommunications, retail, logistics, media, travel, manufacturing, energy, education, government, insurance, real estate, gaming and cybersecurity.

If you would like to add your own queries and methodologies to our process, please get in touch with us.

8 Benchmark Testing Methodology

During the benchmark measuring phase, we deployed the Python code hosted on GitHub into an Ubuntu server hosted on UpCloud in the Sweden region. Our server had 2 cores, 8GB of memory and 10GB of storage.

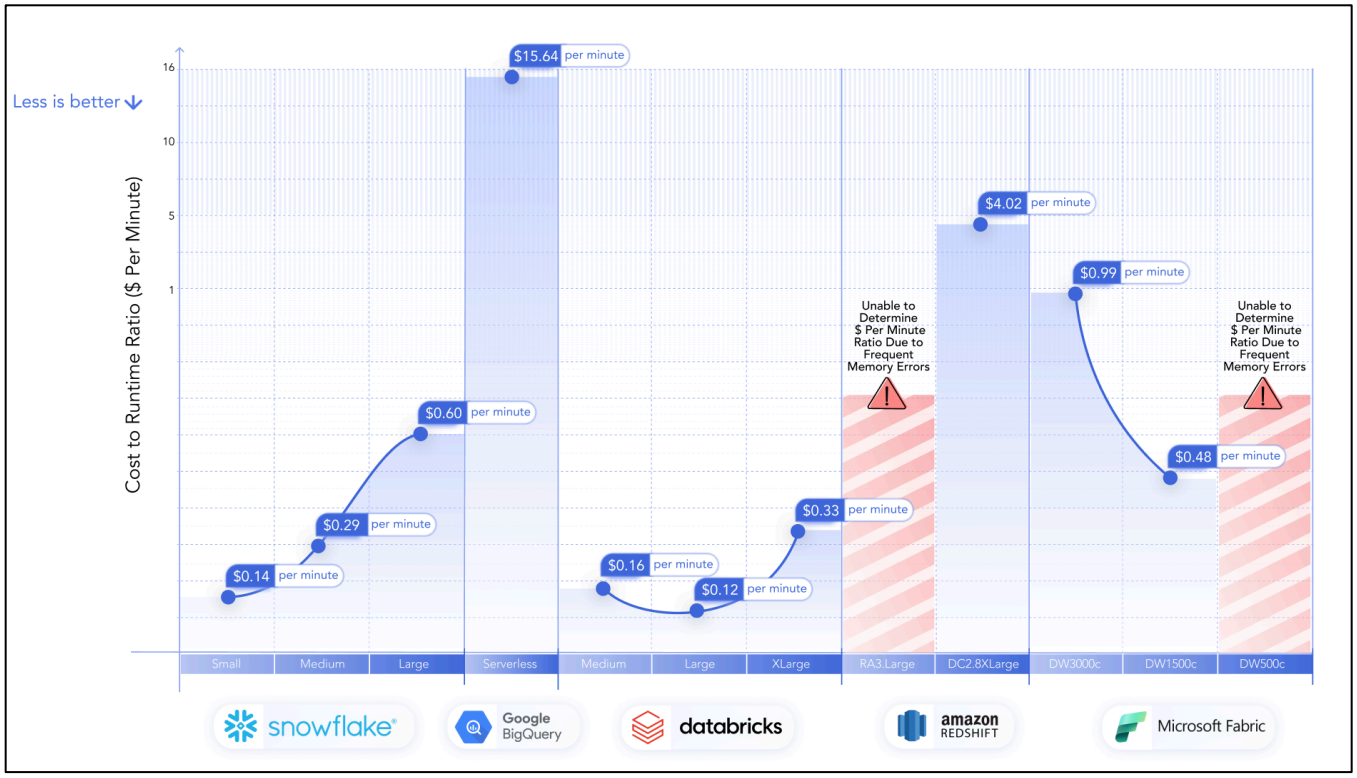
For each data warehouse compute engine, we initiated a dedicated Tmux session to sequentially run the full set of benchmark queries in an automated manner. Query execution timings were captured and logged into a CSV file. After execution, we allowed a 24-hour window for cost data to fully propagate in the billing console before recording the total cost associated with running the queries.

When query response times were accessible via SQL metadata or query history, we retrieved them programmatically. In cases where such access wasn't available programmatically, we measured response times using Python-based instrumentation around the query execution.

9 Performance Benchmarks

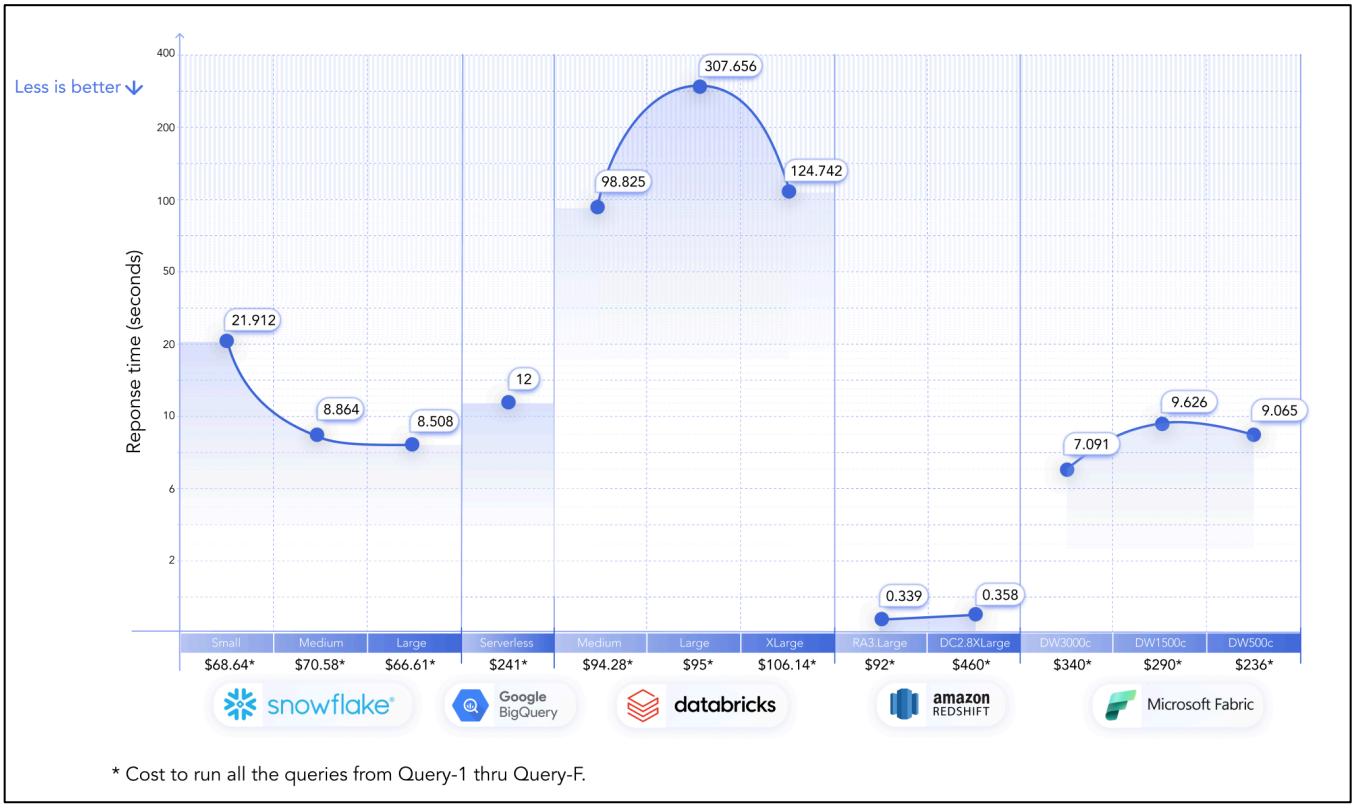
9.1 Cost to Runtime Ratio

We understand that response time alone does not dictate a data warehouse purchase decision. Performance exists within a broader context of total cost of ownership, business value generation and other parameters. Therefore, we wanted to establish the cost-to-performance ratio first, so you have this important context in mind when reviewing the query response times that follow.



9.2 Query-1 Link

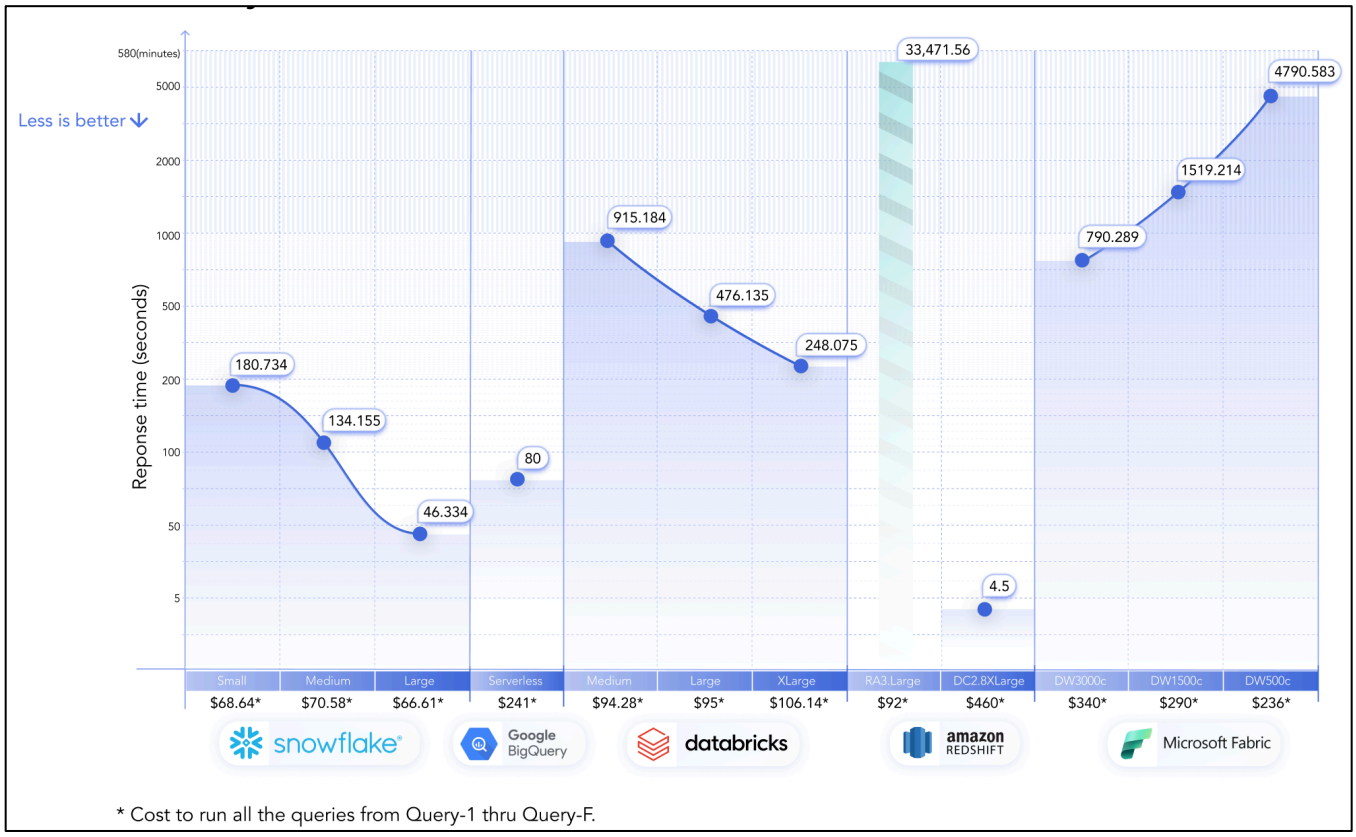
Query Description – A full table scan with a single aggregation using sum (l extended price) on the line item table.



Snowflake	Snowflake exhibits characteristics of the law of increasing returns, with performance improving disproportionately as workload size scales from small to medium.
BigQuery	A few seconds slower than Snowflake Medium and Large engines.
Databricks	Shows a performance degradation from Medium (98.825s) to Large (307.656s), indicating diminishing returns or inefficiencies at scale.
Redshift	Response time is excellent for a basic `SUM` query but comes with a hefty cost.
Microsoft Fabric	Better performance than Snowflake (with additional costs). However, DW1500c underperforms compared to the lighter DW500c, and scaling up resources yields diminishing returns.

9.3 Query-2 Link

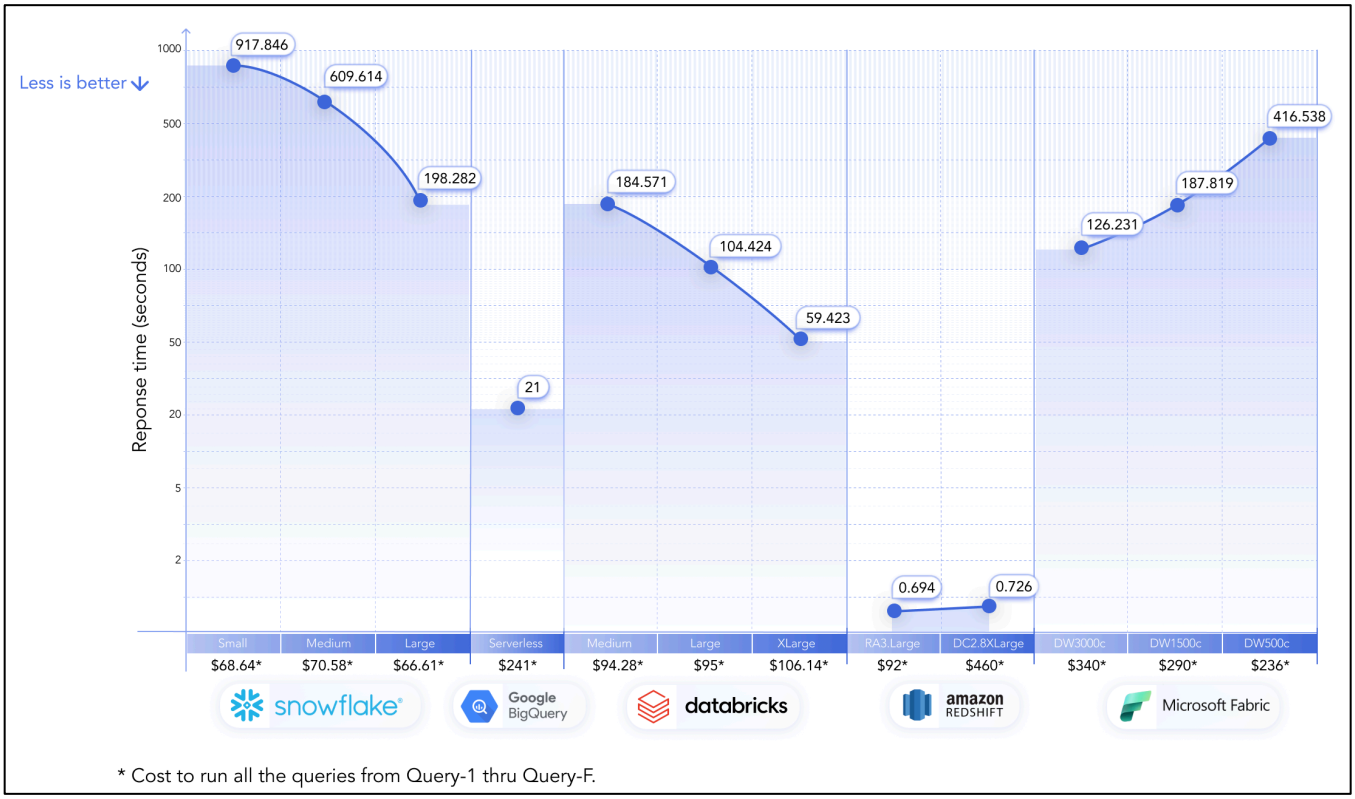
Query Description – A single table scan with multiple aggregations (count, sum, avg, min, max) over semi-structured JSON data in the line item table.



Snowflake	Performance improves significantly with scale. Snowflake Large delivers the best result.
BigQuery	Serverless (80s) is faster than Snowflake Small, but with a very high total cost ($241).
Databricks	Most people in the data warehousing community agree that Databricks' response time is a little off, and this chart shows the same.
Redshift	Redshift's RA3.large configuration with 2 nodes took approximately 574 minutes – over 9 hours – to complete the query, which is clearly unacceptable. Redshift has been known to sporadically take excessive time to execute certain queries, a long-standing issue acknowledged by the community for over a decade. However, DC2.8XLarge gives an unbelievable response time for a SF1000 dataset.
Microsoft Fabric	Inefficient query optimization results in poor performance, even on high-tier engines like DW3000c. Microsoft Fabric shows that higher resource tiers do not always translate to better performance, with diminishing returns on scaling up.

9.4 Query-3 Link

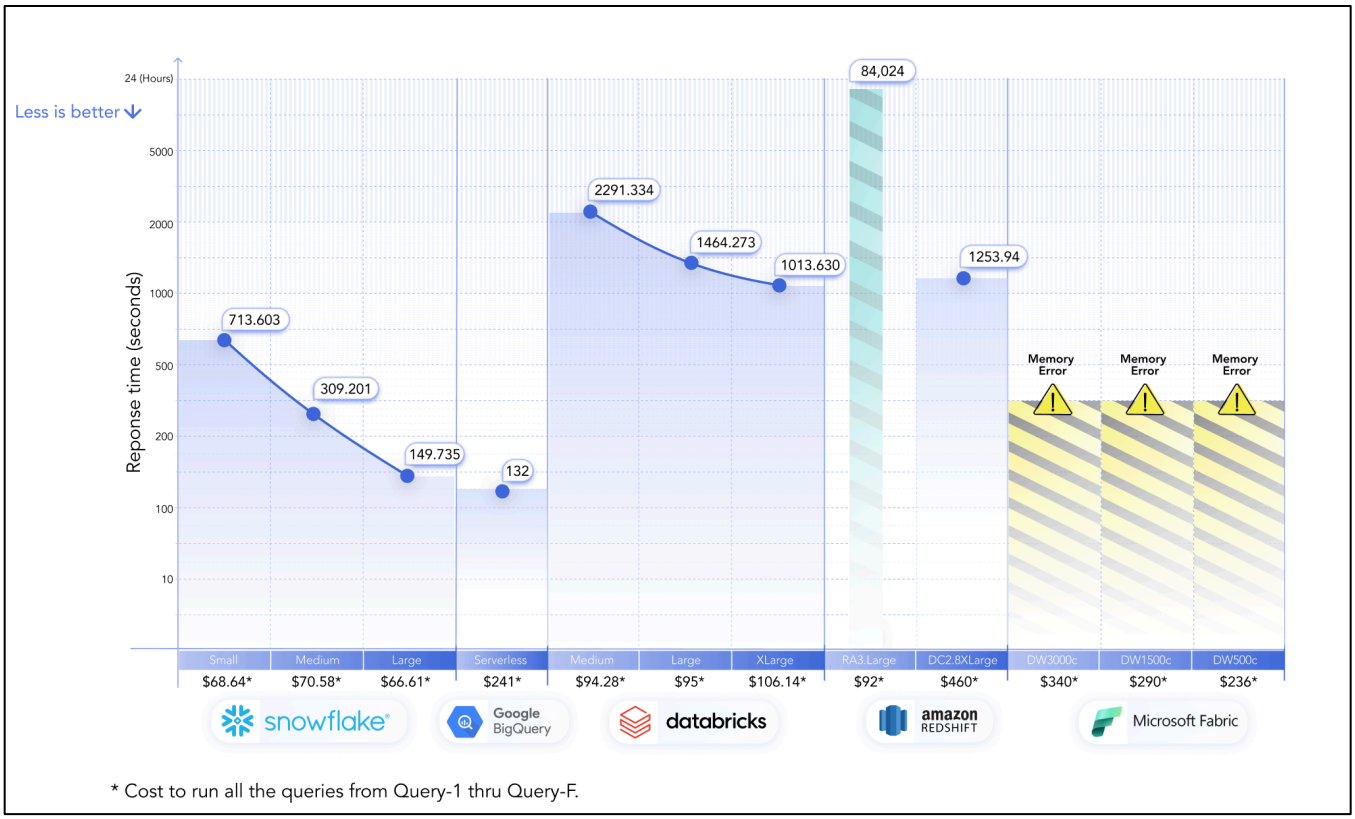
Query Description – A window function query applying lead, lag, and first value over l order key partitions, ordered by l line number, to analyze price and date trends within orders.



Snowflake	Slowest at ~198-917 seconds; maybe it is not optimized to handle functions like lead and lag.
BigQuery	Serverless does its magic significantly outperforming other systems.
Databricks	Outperforms Snowflake on all tested engines, offering faster query response times.
Redshift	Stands out as the only platform achieving sub-second response times for compute intensive queries.
Microsoft Fabric	Microsoft Fabric's DW3000c configuration outperforms all Snowflake configurations for this specific query, and even Microsoft Fabric's DW1500c configuration, at 187.819 seconds, it is slightly better than Snowflake's Large configuration which takes 198.282 seconds.

9.5 Query-4 Link

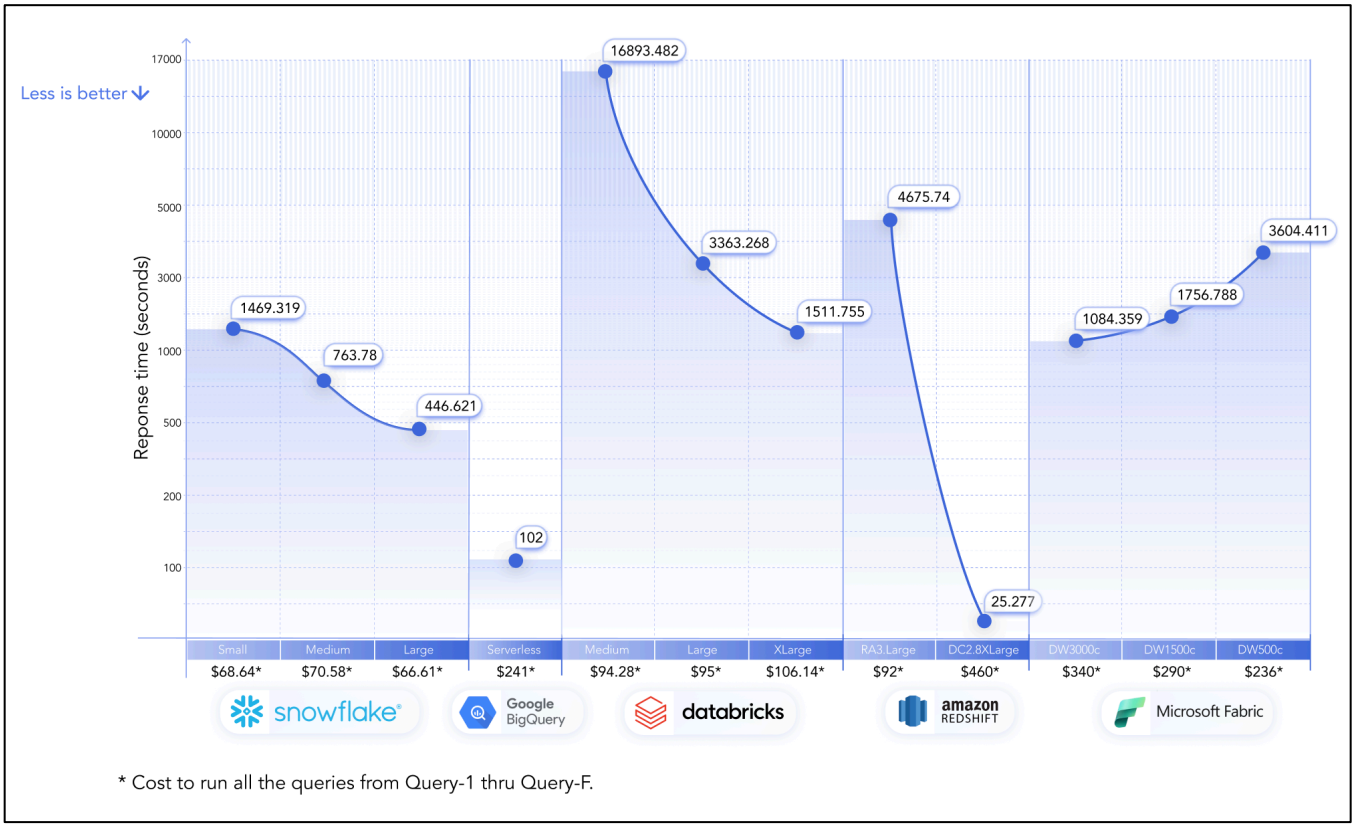
Query Description – A common table expression (CTE) performs aggregations on semi-structured line item and orders data joined by order key, grouped by ship date, ship mode, and order priority; the outer query applies row number window functions partitioned by order priority and ship mode.



Snowflake	Large configurations perform approximately four times better than small configurations and are slightly more cost effective.
BigQuery	Once again demonstrates impressive speed for complex analytical queries in its serverless model, making it a very strong contender for this type of workload.
Databricks	Databricks shows that increasing resources leads to better performance, but the response times are relatively high compared to other systems.
Redshift	Redshift's RA3.large configuration with 2 nodes took approximately one day to complete the query, which is clearly unacceptable. DC2.8XLarge is in the Databricks' performance league.
Microsoft Fabric	All 3 engines encountered memory errors – an unexpected outcome, especially from Fabric which is positioned as an enterprise grade solution.

9.6 Query-5 Link

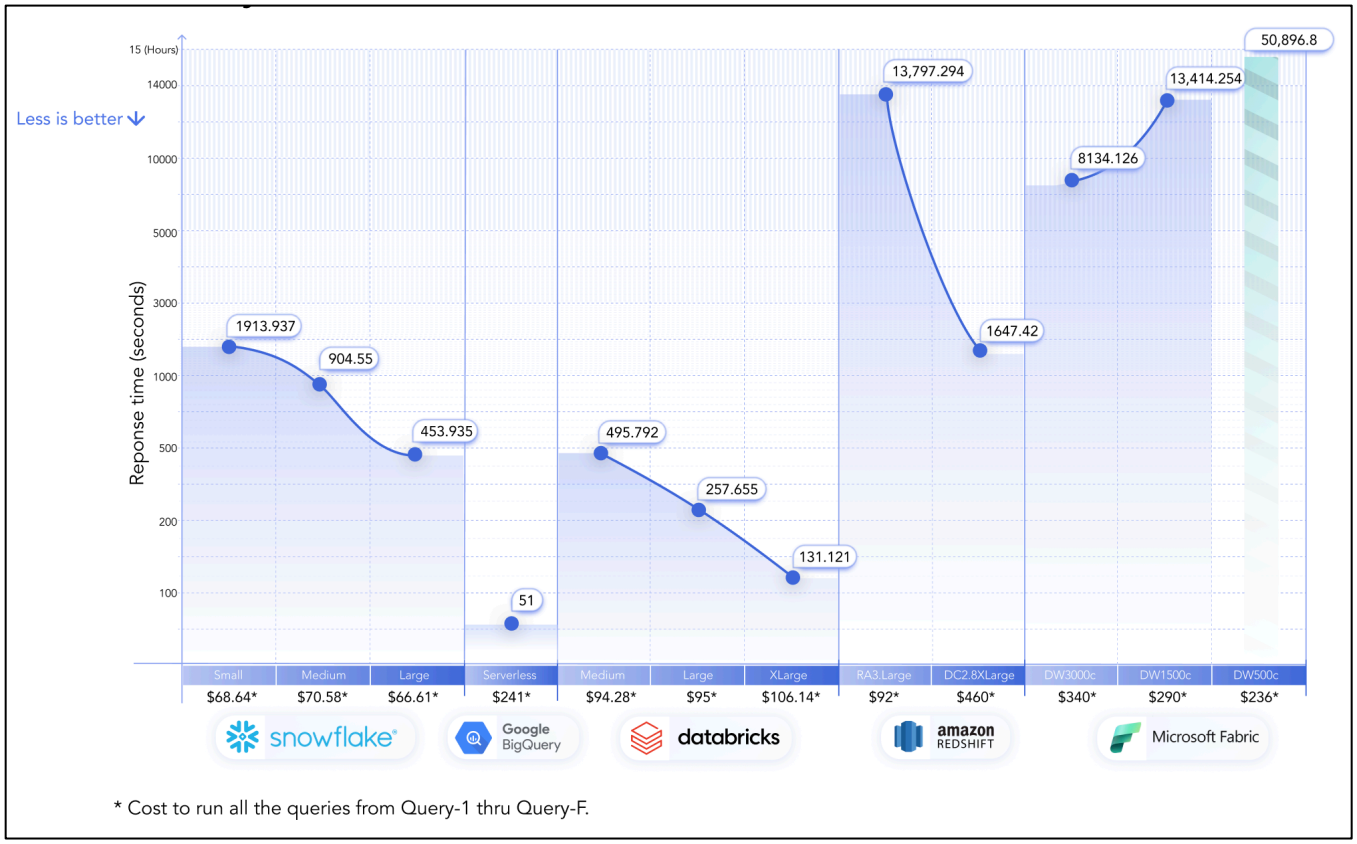
Query Description – A CTE computes monthly customer-level aggregates for total spending and quantity, applies rank window functions within each month, and filters to the top 3 customers by price or quantity per month.



Snowflake	Snowflake's small configuration outperforms Databricks' XLarge setup.
BigQuery	BigQuery has a very fast response time for this query type, significantly outperforming other systems.
Databricks	Databricks shows that increasing resources leads to better performance, but the response times are relatively high compared to other systems, especially for the Medium configuration.
Redshift	The DC2.8XLarge configuration completes the query in just 25 seconds, whereas the RA3 configuration takes 77 minutes for the same task. This significant performance disparity within the Redshift engine is well recognized in the data community.
Microsoft Fabric	For this specific query involving monthly customer-level aggregates and ranking functions, Microsoft Fabric demonstrates better performance across its configurations compared to Databricks.

9.7 Query-6 Link

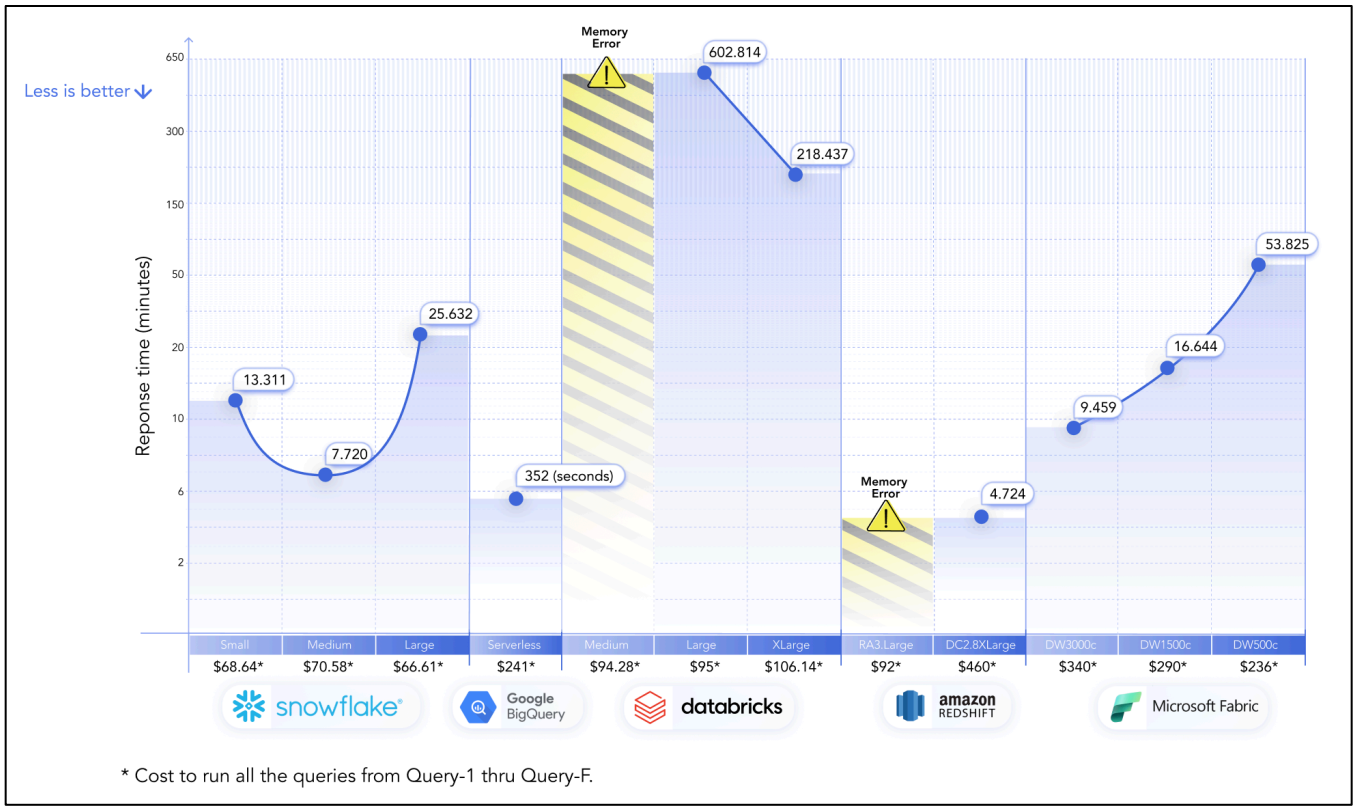
Query Description – A complex query CTE pipeline cleans and tokenizes comments from orders and line item, aggregates word counts by month and ranks words to extract the top 5 most frequent (excluding stop words) per month.



Snowflake	Snowflake underperforms compared to Databricks; even Snowflake's Large instance which takes over 15 minutes, is slower than Databricks' Medium configuration which completes the task in just over 8 minutes.
BigQuery	Fastest at ~51 seconds but comes with a cost.
Databricks	Provides excellent price–performance ratio with strong speeds around 4–8 minutes at competitive pricing.
Redshift	Shows mixed performance with significant variation between configurations, taking 27+ minutes to several hours.
Microsoft Fabric	Taking 14 hours to run a query while competitors take fraction of the time, not acceptable.

9.8 Query-7 Link

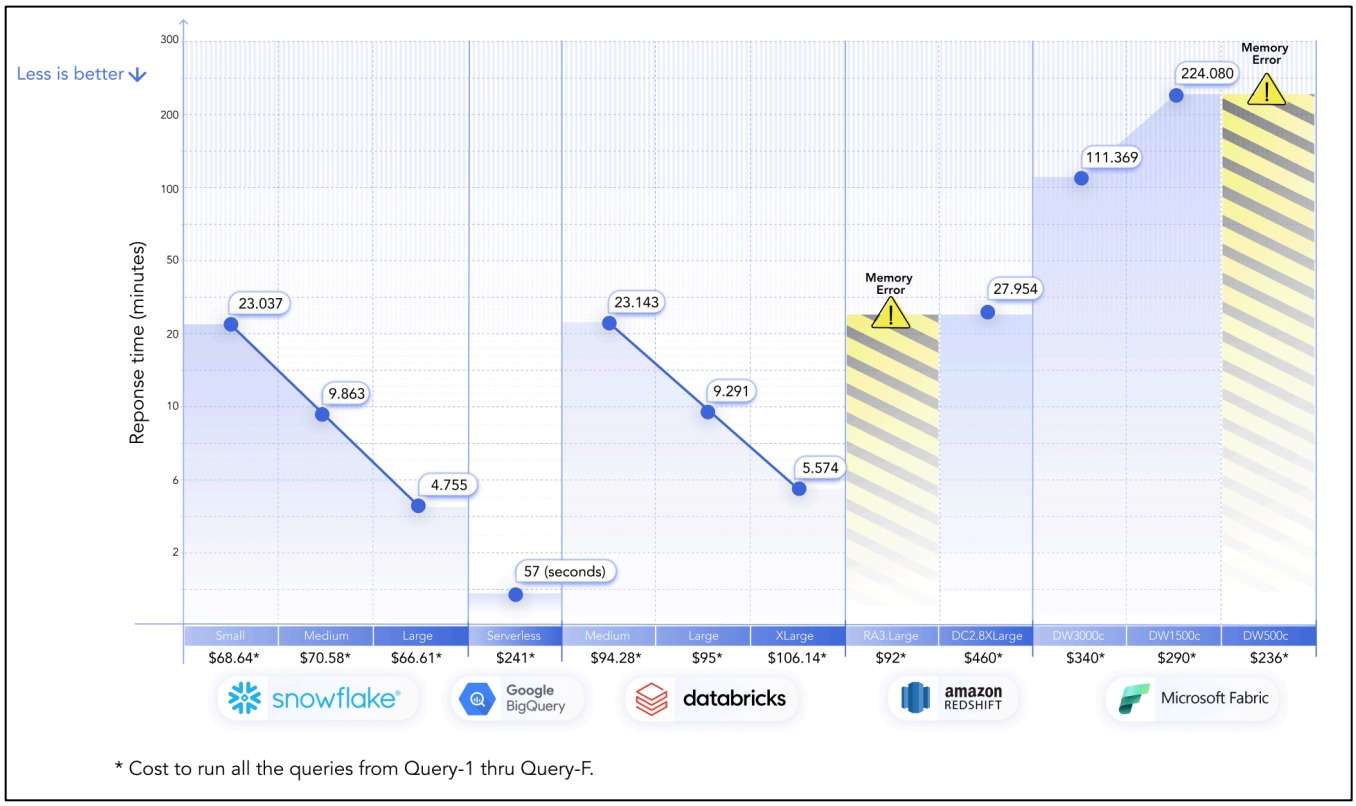
Query Description – A multi-step query on semi-structured JSON tables (orders, line item, and customer) that aggregates monthly customer sales and quantities, ranks customers by spending and quantity, extracts digits from customer names, sums these digits, and filters for top-ranked customers whose digit sum is odd.



Snowflake	Medium compute outperforms large compute. Brute force scaling does not work here.
BigQuery	BigQuery dominates complex JSON operations combined with string processing and mathematical functions.
Databricks	Did not expect memory error from a strong player like Databricks. Struggles severely with this multi-faceted analytical workload.
Redshift	RA3 performs poorly due to lack of computational intensity while DC2.8XLarge wins the show.
Microsoft Fabric	Significant runtime differences observed between DW1500c and DW500c.

9.9 Query-8 Link

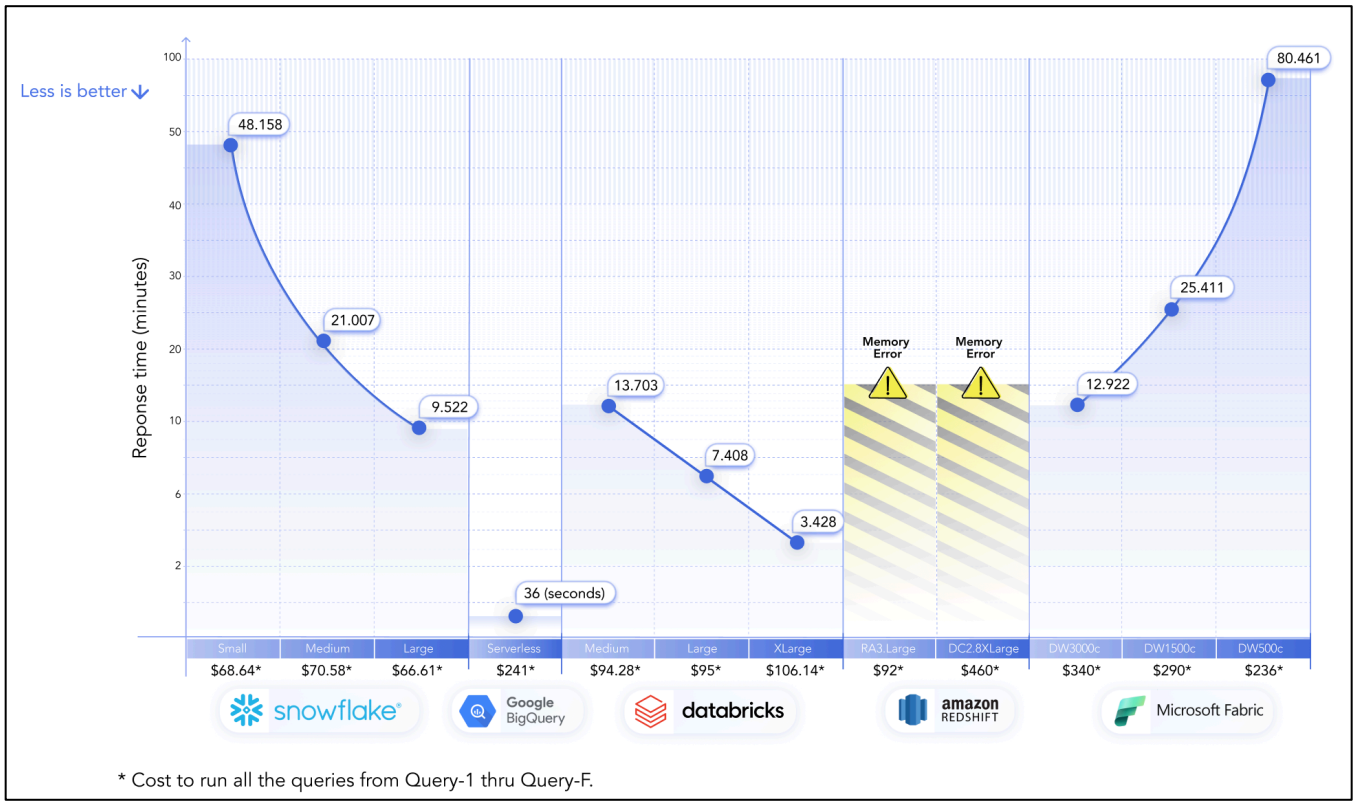
Query Description – A layered query on structured tables (orders, line item, customer) that cleans and classifies comments by length, combines order and line item comments, filters for those containing the keyword “final,” ranks them by comment length per customer, and aggregates the top 5 comments per customer ordered by total comment count.



Snowflake	Shows strong performance across all sizes. The Large configuration completes in 4.755 minutes, which is among the fastest.
BigQuery	Winner, significantly outperforms other systems (without any guardrails on compute and cost).
Databricks	Medium and Large configurations are close to Snowflake in performance, with the Large configuration at 5.574 minutes.
Redshift	Both RA3 and DC2.8XLarge instances exhibited suboptimal performance, with RA3 encountering memory error and DC2.8XLarge showing unexpectedly high runtimes.
Microsoft Fabric	All tested configurations (DW3000c, DW1500c, DW500c) are much slower, ranging from 111.369 to 224.080 minutes, and DW500c encounters a memory error.

9.10Query-9 Link

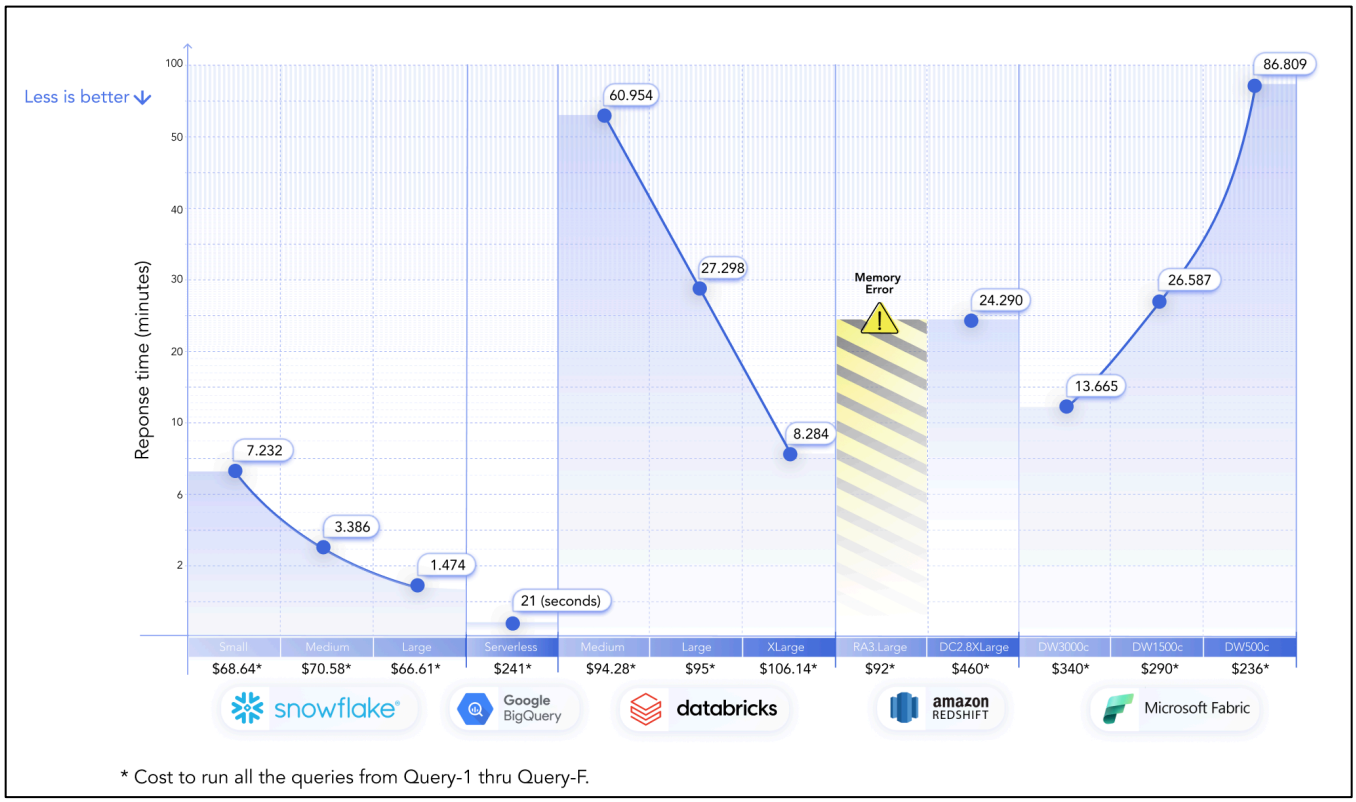
Query Description – An advanced query on structured tables (customer, orders, line item) that calculates per order revenue, and ranks orders chronologically per customer, computes cumulative revenue over time, aggregates monthly order counts and revenue, and generates a summarized monthly revenue report along with lifetime revenue per customer.



Snowflake	Shows a decreasing trend in response time as the warehouse size increases from Small to XLarge, however, is outperformed by Databricks on almost all occasions.
BigQuery	Serverless is the clear leader, completing the query in just 36 seconds (without any guardrails on compute and cost).
Databricks	Databricks shows that increasing resources leads to better performance, with the XLarge tier having the best response time among its configurations.
Redshift	Despite extensive run times, both configurations terminated with memory errors; we were still billed for these incomplete queries.
Microsoft Fabric	Significant runtime delta between DW1500c and DW500c.

9.11 Query-10 Link

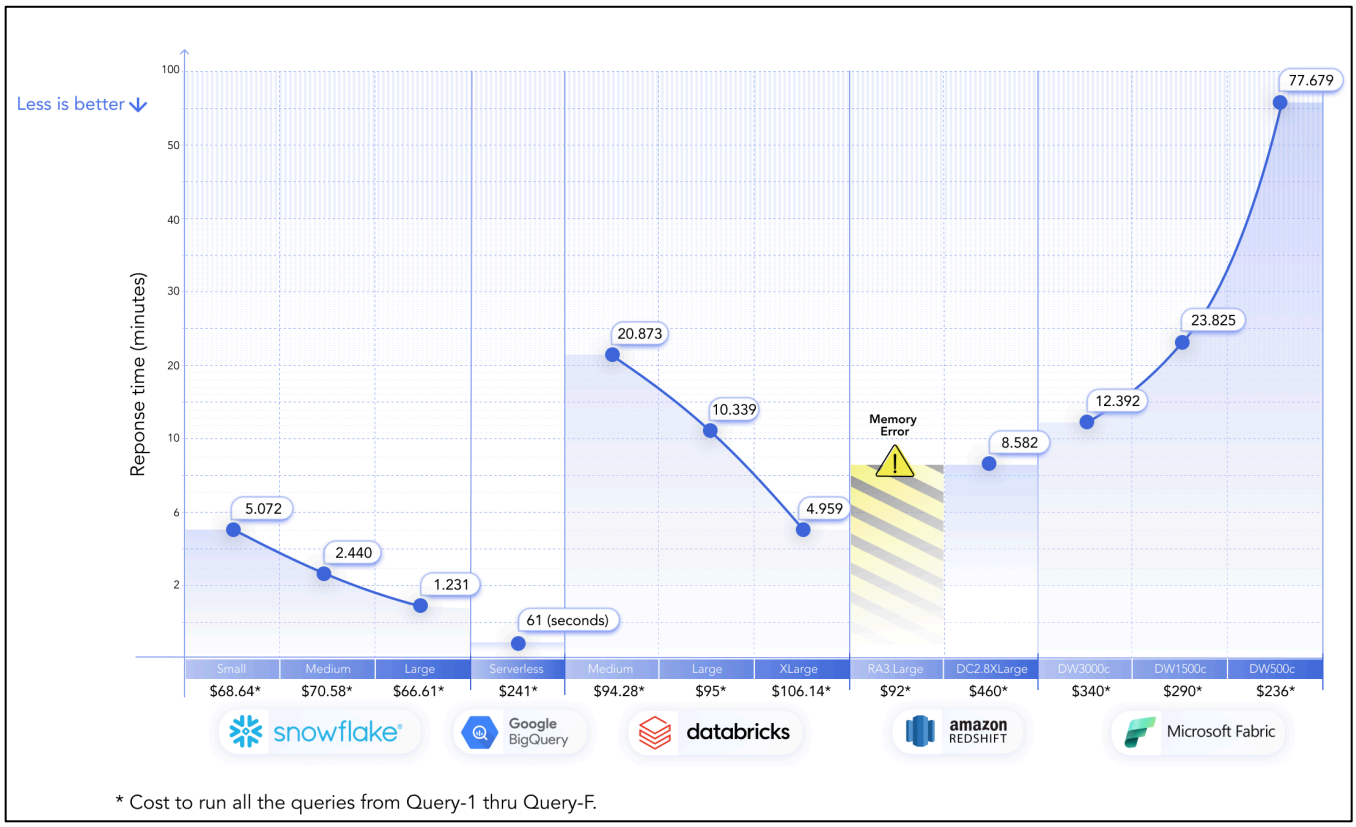
Query Description – Analysis on structured tables (customer, orders, line item) calculating per-customer order counts, revenue, and shipments within 30 days, deriving average revenue per order, computing shipment ratios, and filtering for customers with shipment ratios above 50%, ordered by average revenue per order.



Snowflake	Snowflake demonstrates the best performance, scaling effectively with larger warehouse sizes. Its XLarge warehouse completed the query in just 1.474 minutes.
BigQuery	BigQuery has an excellent response time for this query type, significantly outperforming other systems (without any guardrails on compute and cost).
Databricks	Outperformed by Snowflake, cannot handle query-10's analytical workloads.
Redshift	RA3 performs poorly while DC2.8XLarge does slightly better than Fabric's DW1500c.
Microsoft Fabric	Performs almost in the league of Databricks without bringing cost into the equation.

9.12 Query-11 Link

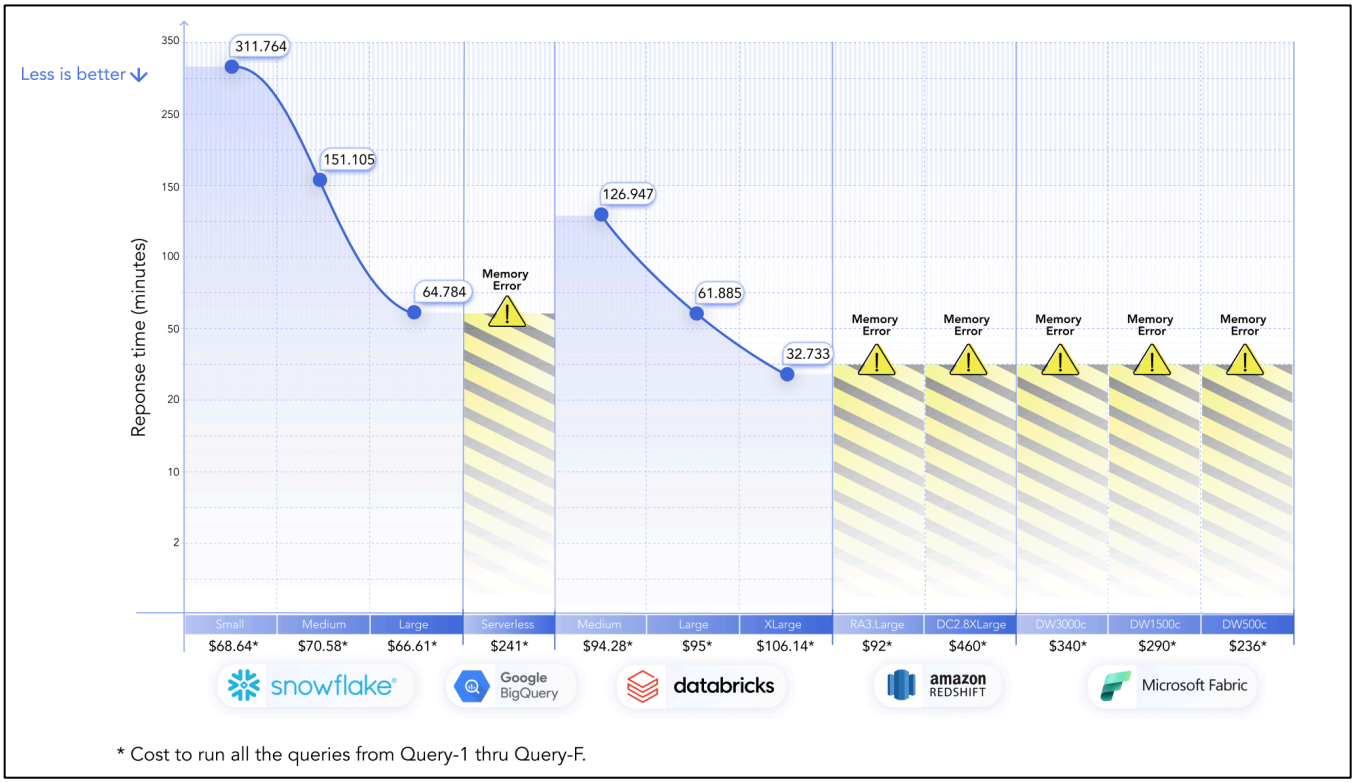
Query Description – Aggregates total revenue and average discount per order from the semi-structured JSON table line item, joins with orders to retrieve order details, filters orders with revenue over 50,000 and low average discount, and sorts by revenue descending.



Snowflake	Snowflake shows a consistent decrease in response time as the configuration size increases.
BigQuery	Undisputed king when it comes to response time (without any guardrails on compute and cost).
Databricks	Databricks showed a response time of 4.959 minutes on its best-performing configuration shown. Other configurations were slower at 10.339 and 20.873 minutes.
Redshift	The RA3 Large configuration proved unsuitable for our dataset size and query patterns, as we consistently encountered memory errors during testing. While DC2.8XLarge does better than Fabric's DW3000c.
Microsoft Fabric	Microsoft Fabric had the slowest performance in this test. Massive datasets affect DW500c's performance.

9.13 Query-F Link

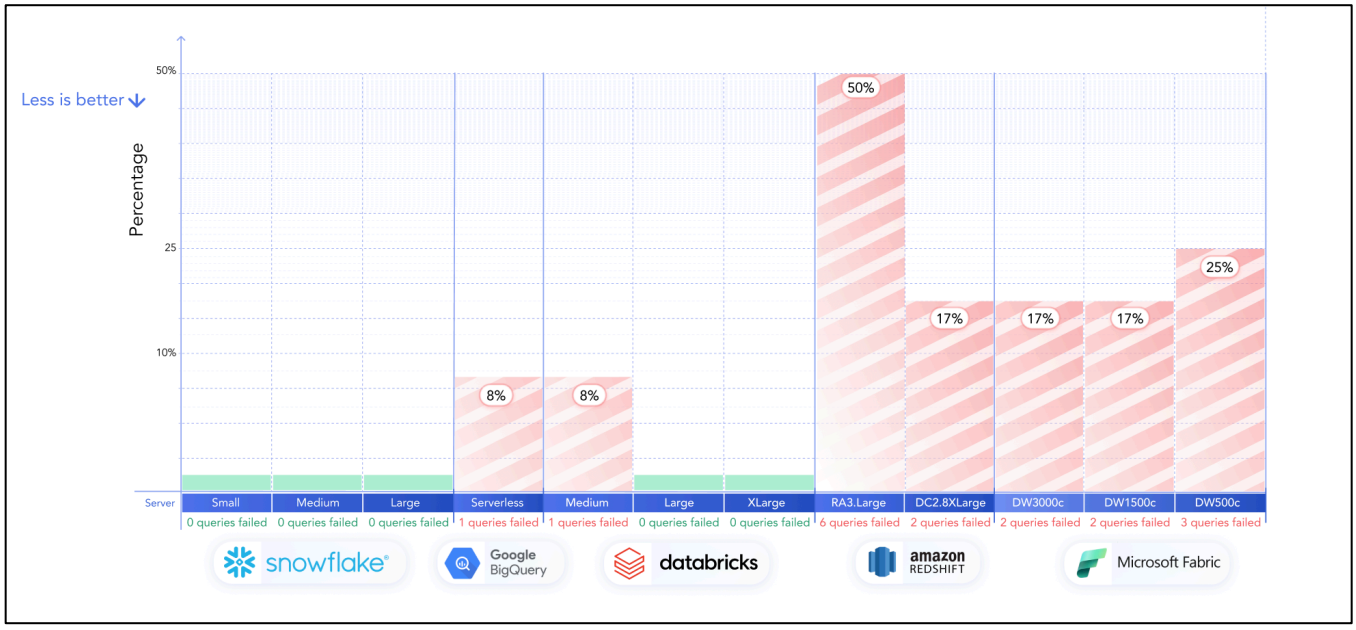
Query Description – The “F” in Query-F stands for Frankenstein. It is designed to push data warehouses to their limits, testing the robustness of their compute engines and the effectiveness of their fail-safe architectures.



Note: Query-F is technically demanding due to its extensive use of Common Table Expressions (CTEs), complex joins, and a variety of aggregations and calculations. It involves multiple window functions, subqueries and text analysis operations, all performed on a large dataset, making it computationally intensive.

Snowflake	Query-F is demanding, and the fact that compute engines didn’t fail is a testament to the robustness of the software behind them.
BigQuery	During our testing, we pushed the default (stock) BigQuery serverless configuration to its operational limits. Memory limits were exceeded in the serverless compute, resulting in a memory error.
Databricks	Databricks' engines demonstrate superior performance regardless of scale or complexity, establishing it as the clear leader in handling complex and demanding SQL queries efficiently.
Redshift	Both tested Redshift configurations encountered memory errors, indicating they struggled with Query-F.
Microsoft Fabric	Given Microsoft's track record with government-grade infrastructure, Fabric's performance issues were unexpected and shocking.

9.14Memory Error Failures vs. Benchmark Queries



Our goal for the benchmark report extended beyond measuring response time and cost to rigorously stress-test each data warehouse's architectural resilience under extreme conditions. We deliberately pushed these platforms to their operational limits by loading the heavyweight TPCH_SF1000 dataset and executing exceptionally complex analytics queries designed to reveal critical differences in compute instance quality, memory management robustness and fail-safe mechanisms.

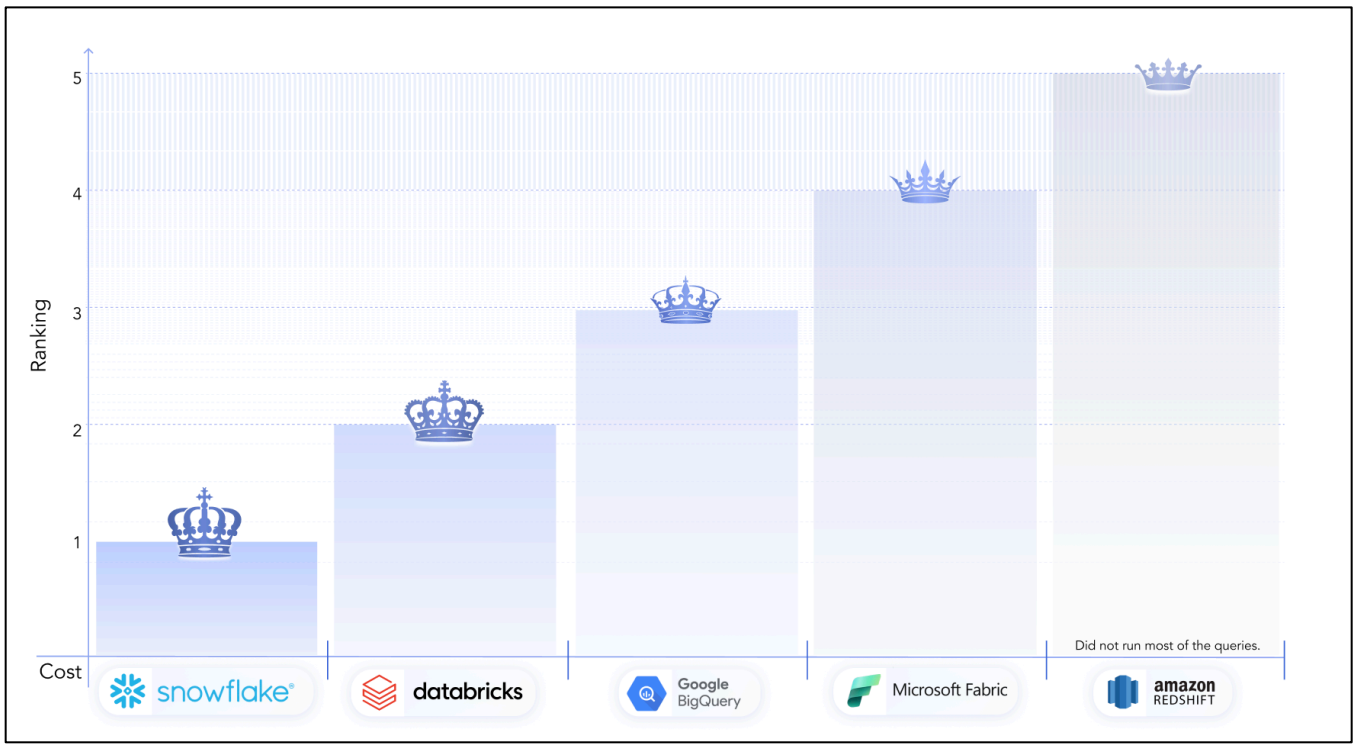
This aggressive testing approach was crucial in revealing significant architectural differences that only surface under extreme stress. The failure patterns triggered by these high-pressure scenarios offered deep insights into each platform's true operational limits and reliability – insights that traditional benchmarks focused solely on speed and cost would have missed.

Snowflake	Zero failures even on the smallest compute engine reinforces operational confidence.
BigQuery	BigQuery's single memory error on Query-F – an isolated edge case – nonetheless reveals a critical architectural limitation, despite Google's strong reputation for scalability. While few real-world workloads reach Query-F's level of complexity, this failure highlights potential processing boundaries.
Databricks	Databricks demonstrates foundational robustness with only one anomaly while running on Query-F on the smallest machine.
Redshift	Half of the queries failing due memory error clearly indicates underlying vulnerable architecture.
Microsoft Fabric	Fabric has yet to reach enterprise-grade maturity, with memory errors occurring consistently across multiple instance sizes.

10 Estuary Ranking

At Estuary, we have developed our own proprietary rankings, drawing heavily from the findings of this Data Warehouse Benchmark Report. These rankings are also shaped by extensive internal discussions and enriched by the firsthand experiences and candid feedback of our customers. This approach ensures that our evaluations are not only grounded in empirical data but also reflect practical, hands-on insights, providing a well-rounded and actionable perspective.

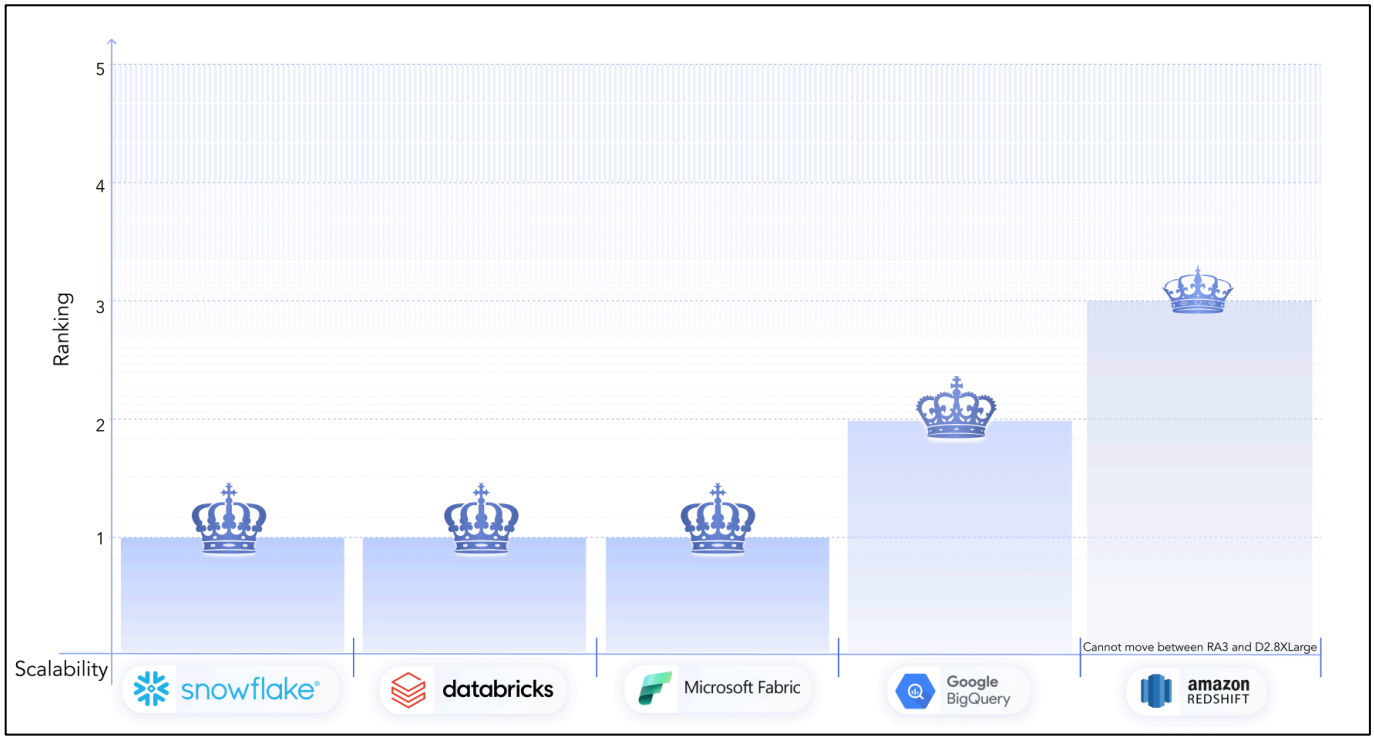
10.1 Cost Ranking



Snowflake	There's a local legend in San Francisco that Snowflake can be expensive. However, in our experience, when compute engines are chosen wisely and auto-shutoff features are deployed, you can effectively manage costs and complete the job within predictable budget. Additionally, Snowflake's engines are known for their reliability, ensuring that there is no money wasted on failed operations.
Databricks	Databricks generally tends to be somewhat more expensive than Snowflake, largely because its query engines may take slightly longer to execute workloads. However, Databricks has built-in effective cost controls to help prevent runaway expenses and keep spending within the customer's budget. Additionally, Databricks is a leader in embedding advanced AI capabilities, optimizing their engines to support sophisticated AI workloads. While this AI focus delivers significant value, it can also contribute to higher overall costs.
BigQuery	BigQuery's deceptive simplicity makes it seem user-friendly at first glance due to its serverless nature, eliminating the need to manage compute engines.

	However, as data grows and queries become more complex, controlling costs becomes challenging because there's no way to limit query expenses. In enterprise environments, financial controllers prefer predictability over unpredictability randomness.
Microsoft Fabric	Enterprise giant Microsoft has a cost meter that seems to tick a little faster when you are not a billion-dollar company. With the right guardrails and reasonably simple workloads, small and mid-tier companies can make it work – assuming they have dedicated resources to manage Fabric. That said, Microsoft Fabric is truly built for enterprises with deep pockets and all their code already living comfortably on Azure.
Redshift	Redshift is simply too expensive to justify using it just for data loading. Much like Microsoft Fabric, it lacks an automatic server shutdown mechanism. If your engineers forget to turn off the clusters at the end of the day, don't be surprised when the next morning's bill is higher than the cost of the actual work. To make matters worse, Redshift's servers aren't particularly robust – meaning you may end up paying for failed or stalled queries too.

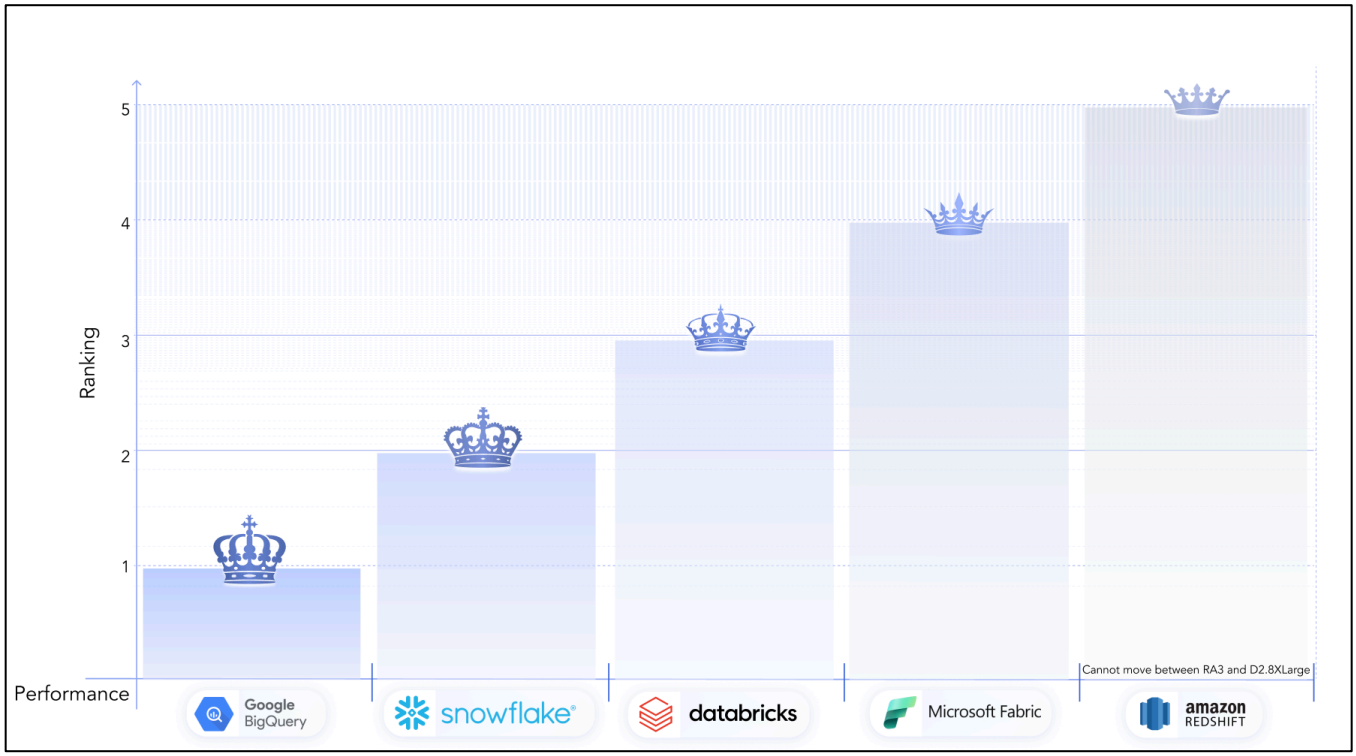
10.2Scalability Ranking



Snowflake	Out of all the platforms we have tested, Snowflake is by far the fastest when it comes to provisioning and cold starting a new server. Even its Standard Edition can comfortably handle the compute, and storage demands of a 40-year-old Fortune 500 company. We have seen clients programmatically spin up

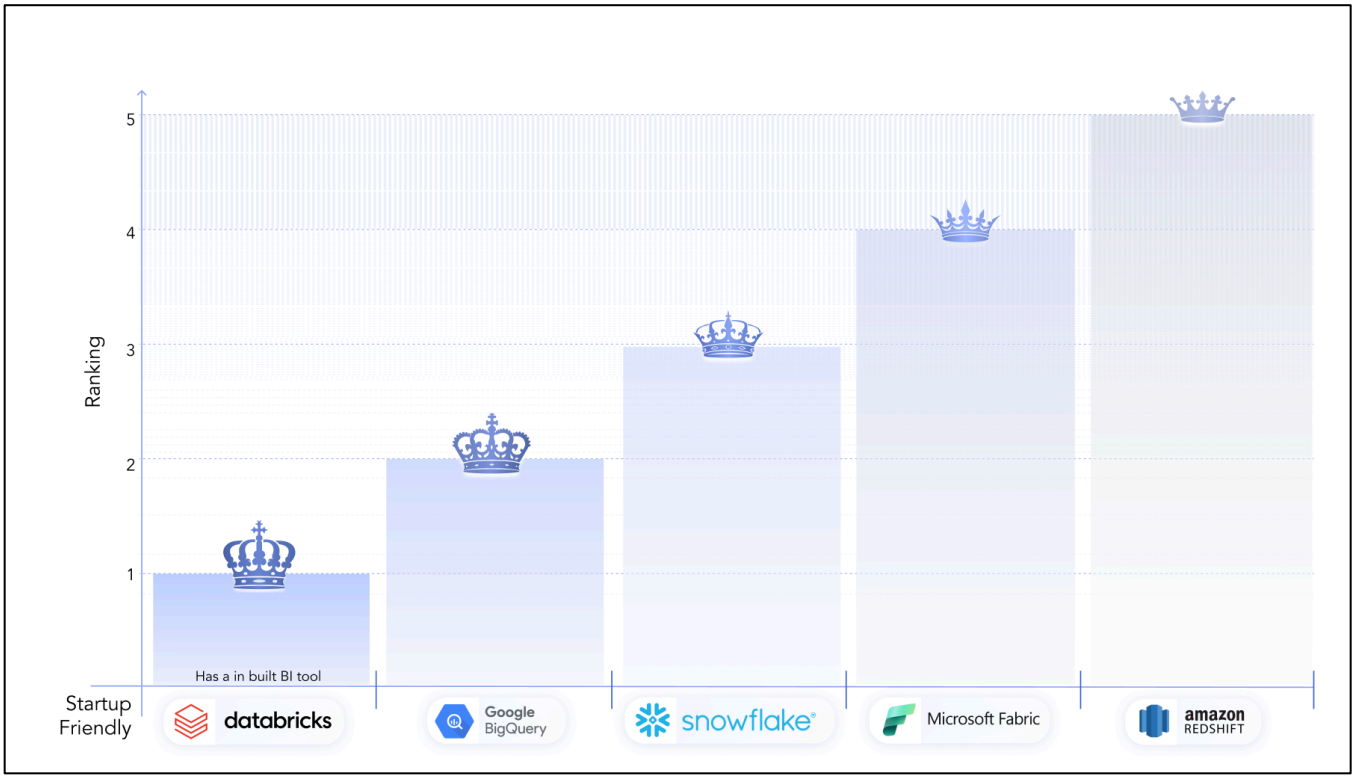
	warehouses and inject terabytes upon terabytes of data –effortlessly and reliably.
Databricks	Databricks takes a noticeable amount of time to provision a new server or cold start an existing compute engine. That said, it's also the platform powering some of the most complex workloads in the world – particularly those involving AI. With a bit of patience, Databricks can scale to support all popular Bitcoin marketplace companies under one roof.
Microsoft Fabric	Enterprise-grade Microsoft Fabric is built to scale and handle highly sophisticated data workloads – think military-grade telemetry. While it can be sluggish when spinning servers up or down, it's otherwise designed to manage real-time data streams like travel logistics, stock market fluctuations, and more with confidence.
BigQuery	BigQuery offers a single serverless compute option – unless you are willing to navigate the bureaucratic hoops required to provision dedicated resources. Its querying engine is undeniably powerful, but it often lacks the blessing of corporate financial controllers, creating a virtual ceiling on how far it can scale within budget-conscious organizations.
Redshift	Even after more than a decade, Redshift still hasn't separated compute from storage. During this very benchmark report, we initially spun up a smaller server – but due to repeated failures in loading our data, we had to start over and migrate everything to a larger instance. In doing so, we paid for unused resources and lost significant time. Many organizations begin with modest infrastructure, expecting to scale compute independently while keeping storage decoupled. Unfortunately, Redshift doesn't offer that flexibility. That's one of the reasons it ranked last in our evaluation, and why we strongly advise thinking twice before opting for a low-storage Redshift setup.

10.3 Performance Ranking



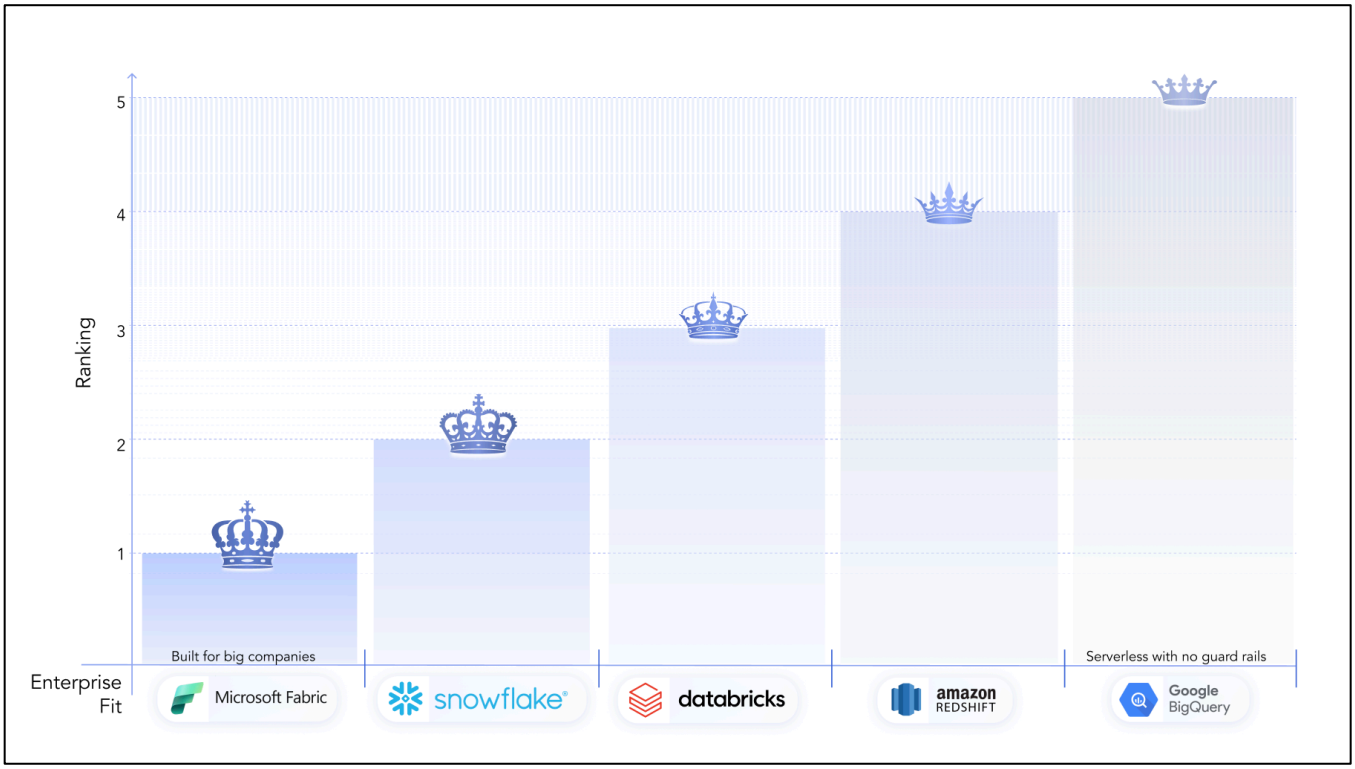
BigQuery	BigQuery is the undisputed king when it comes to raw performance. If your sole purpose of this report is to find out which data warehouse performed better and had a quicker response time you can stop reading the report here. The answer is BigQuery. Note: BigQuery serverless comes with no cost controlling guardrails
Snowflake	Snowflake is a comprehensive and always enterprise-ready data warehouse, yet equally accessible for startups. It delivers consistently strong performance, though it falls just short of the raw speed offered by Google's serverless compute engine.
Databricks	Databricks has openly acknowledged its performance lag in the past, but the team has worked diligently over the years to significantly improve query execution times. While it's not the fastest platform out there, Databricks more than makes up for it with a rich set of features and powerful capabilities.
Microsoft Fabric	We believe that, at its core, Fabric still retains characteristics of SQL Server, which was originally designed for transactional rather than analytical processing. As a result, it struggles to deliver expected performance at scale – particularly when handling large datasets and complex queries.
Redshift	Redshift landed in the last spot due to repeated query failures even after running for hours.

10.4Startup Friendly Ranking



Databricks	Databricks is the only data warehouse that has long included a built-in business intelligence tool and a native Python notebook. Most platforms require you to spend thousands on external BI tools – an added cost often overlooked by data warehouse vendors. This built-in functionality is one of the key reasons we ranked Databricks number one in this category.
BigQuery	Google's recent integration of its Gemini AI tool – combined with existing features like query scheduling, job deployment, and a strong partner ecosystem – makes it easy for startups to hit the ground running. That's why we have placed it in the number 2 spot.
Snowflake	In the data community, developers frequently use APIs and Python to manage tasks adjacent to core data warehousing. Snowflake has recognized this trend and is actively developing native features to reduce friction around these workflows. However, it still lacks an AI chatbot to assist semi-technical users in navigating the platform.
Microsoft Fabric	Fabric includes an internal ETL tool, a robust Python notebook, and Azure Blob Storage to support its data warehouse capabilities. However, from an AI perspective, it has yet to integrate a Copilot-style assistant into the Fabric console. They do not have an out of box data visualization tool.
Redshift	Aside from the Query Editor V2, we haven't seen any significant feature releases that would notably benefit the startup community in quite some time.

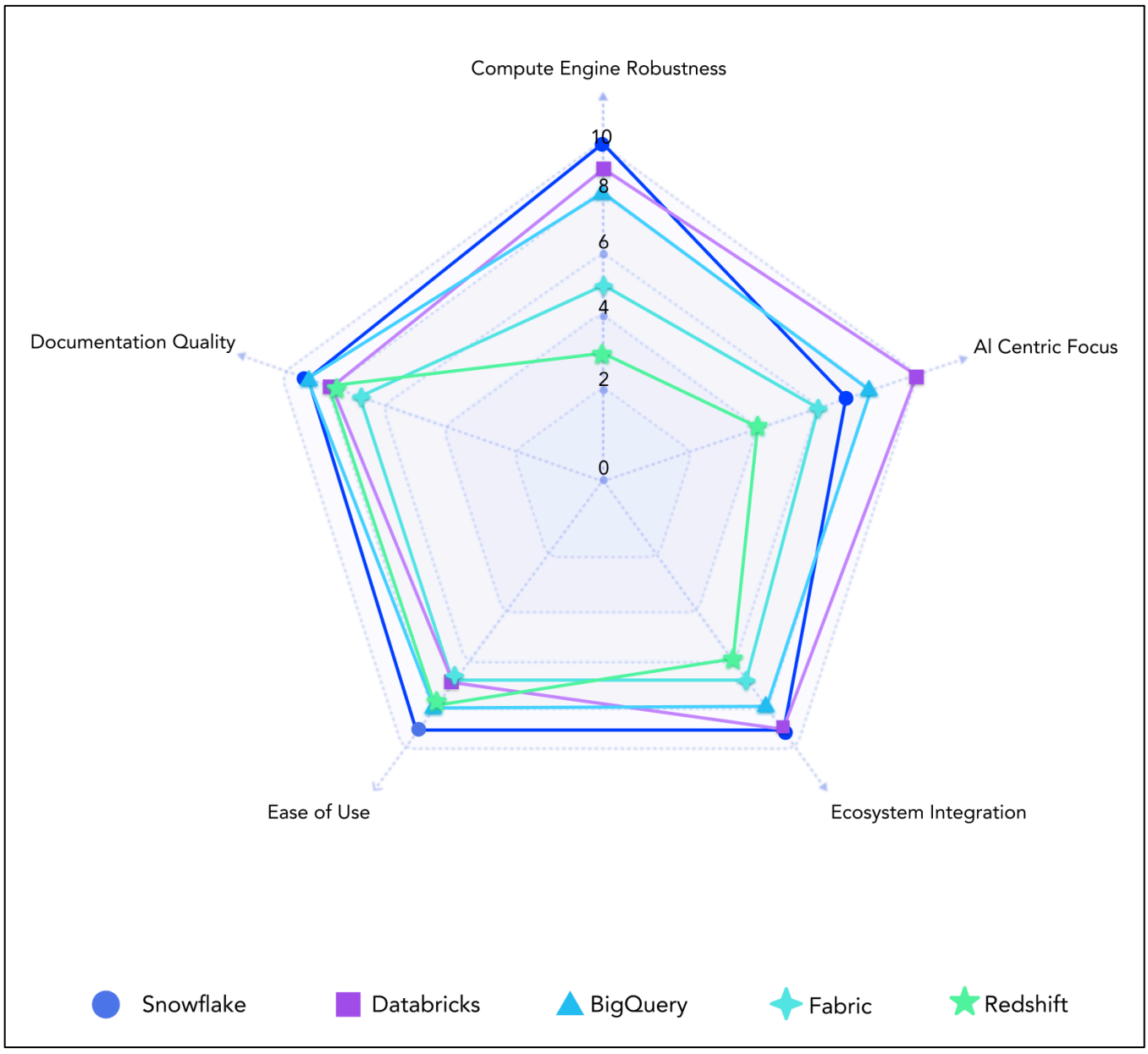
10.5 Enterprise Fit Ranking



Microsoft Fabric	Fabric is primarily designed for large enterprises, especially those with longstanding relationships with Microsoft. Fortune 500 companies often prefer Microsoft products because of the brand's established reputation, proven reliability and seamless integration with their existing technology stacks. For these organizations, choosing Fabric isn't just about features – it's also about trust, vendor support, and the confidence that comes from partnering with a global technology leader they have relied on for decades.
Snowflake	Over the past decade, Snowflake has consistently pushed the boundaries of innovation, repeatedly surpassing industry expectations. Its disruptive approach to cloud data warehousing has not only captured widespread attention but also earned the trust of legendary investors – Warren Buffett, known for his cautious stance on tech stocks, made a notable investment in Snowflake and has expressed satisfaction with the decision. Beyond high-profile endorsements, Snowflake has become the backbone for many large American media conglomerates, powering their data operations daily, and enabling them to unlock valuable insights at scale.
Databricks	Almost every enterprise aiming to leverage AI and related features defaults to Databricks as their data warehouse of choice for running MLOps and building RAG (Retrieval-Augmented Generation) applications. They are not only enterprise-ready but have also consistently delivered on their promises. The recent Databricks conference stands as a strong testament to the platform's readiness and robust capabilities.

Redshift	Amazon was an early player in the cloud space and secured many government contracts that Microsoft missed. Several powerful government organizations host their data on Amazon, where it serves their needs well. However, unlike a decade ago, private companies today are not embracing Redshift with the same enthusiasm.
BigQuery	BigQuery's serverless architecture often faces skepticism from industry veterans with 20+ years of experience who hold decision-making authority in large corporations. Without predictable cost controls and clear accountability, these enterprises are reluctant to adopt a serverless approach that lacks built-in guardrails – unwilling to risk unexpected expenses.

10.6Estuary Radar Ranking



Snowflake excels in Compute Engine Robustness and maintains strong scores across all categories, including Ease of Use and Documentation Quality. Databricks is a close second, with a strong emphasis on AI Centric Focus and high performance in Ecosystem Integration and Compute Robustness. BigQuery shows a well-rounded profile with moderate strengths, especially in Documentation Quality and Ease of Use. Redshift consistently underperforms across most categories, especially in modern AI capabilities and user experience, suggesting it may be falling behind in today's evolving data landscape. Fabric shows strengths in Documentation Quality and Ease of Use but lags in AI Centric Focus and Compute Robustness

11 Warehouse Evaluation

11.1Snowflake

Snowflake is a robust data warehouse. Even its smallest compute engines are designed to prevent memory errors, reflecting the high-quality engineering led by Benoît and the Snowflake team.

Snowflake has maintained its central position in the data ecosystem for over a decade, a remarkable achievement given its dependence on cloud hyperscalers like AWS, Azure, or GCP. This dominance stems from deep technical innovations that address critical enterprise needs, and from its 800+ certified integrations that create a connective tissue across modern data stacks.

Snowflake's multi-cluster shared data architecture uniquely caters to both nimble Y Combinator startups and global Fortune 500 enterprises by aligning with their distinct needs.

For instance, Ryder uses Snowflake's multi-cluster shared data architecture and near infinite scalability to overcome performance challenges while supporting substantially more data, users and workloads, a dual requirement few platforms can satisfy.

The platform's detailed query documentation and execution metadata allow for precise performance tuning, while its pre-built connectors and SQL extensions simplify implementation. Snowflake's query tagging system provides visibility into resource usage, enabling cost attribution at the query level. Teams can audit query patterns across regions and units by adding metadata tags (e.g. project=forecasting_v2) to SQL statements.

Snowflake enables custom integrations via REST API and bidirectionally syncs with CRM tools like Salesforce. Snowflake offers its own SDKs for Python, Java, Go and .NET, enabling seamless integration.

By evolving beyond a traditional data warehouse into a unified platform for analytics, AI and application development, Snowflake has created technical moats that competitors struggle to replicate.

11.2 Databricks

Databricks' query response time is not the fastest among the data warehouses we have benchmarked. This is a well-known issue within the data community, and Databricks is actively working to significantly improve its performance.

However, if you are okay with a slight delay in query response time, Databricks stands out as a pioneer in the data warehouse space, offering a complete package that includes data warehousing, ELT, machine learning, AI capabilities, and a built-in business intelligence tool. Even today, most major players don't provide a native BI tool as part of their warehouse suites, customers typically have to pay extra just to visualize billions of rows.

Unlike traditional warehouses, which confine data within proprietary systems, Databricks stores your data in S3 buckets (or equivalent cloud storage) using open formats. This approach allows you to query your data with any SQL engine or build statistical models using Python, without restrictions. By default, Databricks avoids vendor lock-in, giving you the flexibility to use your preferred tools and technologies.

They were well-prepared to capitalize on the latest advancements in AI. In our view, Databricks is equal parts data warehouse company and AI pioneer. Flo health app with over 420 million users, used Databricks' Mosaic AI as the foundation for Flo's Health Assistant, enabling the chatbot to dynamically adapt responses based on user inputs and historical health data.

Databricks Notebook, supporting Python, Scala, and R is a valuable tool for data professionals performing ad hoc analysis. Its mechanism of allowing Python to be run on top of SQL provides extensive avenues for use. Beyond automation and alerting, some of our clients leverage Databricks Notebook for complete end-to-end workflow orchestration.

Access to challenging datasets like those from the International Monetary Fund or JD Power consumer NPS scores is now democratized through Databricks Marketplace. The datasets come with pre-written code, making it easy to visualize or extrapolate information – often requiring just the click of a button.

11.3 Google BigQuery

Data warehouse engineers and decision-makers may be surprised to learn that BigQuery operates exclusively with serverless compute, offering no straightforward way to limit or control compute usage directly from the console. One has to either strike a year-long contract with them or select a standard edition of BigQuery after going through a bunch of archaic processes. Although serverless compute looks harmless at the surface level, its auto-scaling for large datasets can empty our pockets.

One way to impose limits on BigQuery usage is by setting an upper limit on your monthly cloud budget. However, this approach is far from ideal. It lacks granularity and control – your query might be mid-execution, or a Looker report may be generating, when the budget threshold is hit, abruptly halting operations. For enterprises, this kind of unpredictability is unacceptable.

Not everyone, including analysts and engineers, has permission to configure or calibrate budgets. This unpredictability creates anxiety around cost, which ultimately discourages teams from fully leveraging BigQuery's capabilities.

On the positive side, Gemini AI is integrated into BigQuery, enabling semi-technical users to run queries with ease while helping technical users save valuable time. Additionally, BigQuery's built-in notebooks and workflows offer a full-fledged ETL engine that supports data ingress, egress and transformation, streamlining end-to-end data operations within the platform.

Marketing companies that rely on Google Ads, Google Analytics, and publicly available third-party datasets tend to thrive in BigQuery, largely because there's minimal ETL required to move Google's proprietary data into the warehouse.

For mid-sized companies with under 100GB of data and no reliance on complex nested right joins, BigQuery is an ideal data warehouse solution – cost-effective, scalable and easy to manage. However, if your data volume is expected to grow significantly, it's essential to implement guardrails around BigQuery usage to avoid unpredictable costs.

11.4 AWS Redshift

Estuary customers who host their production data on AWS often choose Redshift as their data warehouse to keep everything within the same cloud ecosystem. This consolidation simplifies data management and avoids the complexity of maintaining disparate systems across multiple platforms.

While we understand the logical rationale behind this conservative approach, we respectfully disagree with our customers' decision.

In today's cloud-native era, there's little justification for a data warehouse to remain running after 20 minutes of inactivity. If Redshift truly aims to empower its customers, it shouldn't continue charging when workloads are idle, especially when its users are asleep.

Redshift has yet to fully decouple storage from compute. During our SF1000 data load, we were forced to upgrade to a more expensive DC2 and RA3 tier solely to accommodate the ingestion process. This kind of architectural constraint was frustrating a decade ago, and it remains unacceptable today.

Among the major cloud data warehouses, Redshift often lags in performance – some basic queries can run for hours, yet its pricing remains relatively high. This mismatch between speed and cost raises concerns about overall value.

We will not recommend Redshift for data-intensive workloads with billions of structured and semi-structured rows involved; however, AWS Cloud holds a dominant 30% market share, and companies like Nasdaq and government agencies like The Contra Costa County District Attorney's Office use Redshift.

11.5 Microsoft Fabric

A significant drawback of Fabric is the lack of an auto-shutdown feature, requiring users to manually power down servers when idle. In high-pressure environments where meeting deliverables is critical, engineers often overlook shutting down unused servers. This manual process adds unnecessary stress on teams and distracts them from focusing on more important tasks.

At its core, we believe Microsoft Fabric runs on Azure Synapse Analytics, which itself retains components of the traditional SQL Server architecture. We highlight this because the table creation syntax and querying style don't fully align with modern analytics needs. For example, Fabric requires using `count_big ()` instead of `count ()` when querying billions of rows, yet the fundamental purpose of an analytical warehouse is to handle billions of rows seamlessly, making this requirement feel counterproductive.

Setting up Fabric is not straightforward and involves numerous prerequisites before you can even initialize the Fabric environment. While Microsoft's focus on enterprise customers makes this understandable, it's still an important consideration to keep in mind.

Once past the steep learning curve, the Fabric ecosystem proves to be truly impressive. Their Data Factory ETL engine offers unique, powerful functionalities unmatched by competitors and can handle nearly all popular use cases with ease.

Although Fabric's query response time may not be the fastest on the market, we would still recommend it to enterprise customers who, due to corporate policies, governance requirements, or brand alignment considerations, feel compelled to select a data warehouse solution from another Fortune 500 company. In such cases, Fabric presents a viable option that balances the need for an established vendor with acceptable performance, ensuring compliance with organizational decorum and procurement standards.

12 Who Are We?

12.1 About Estuary

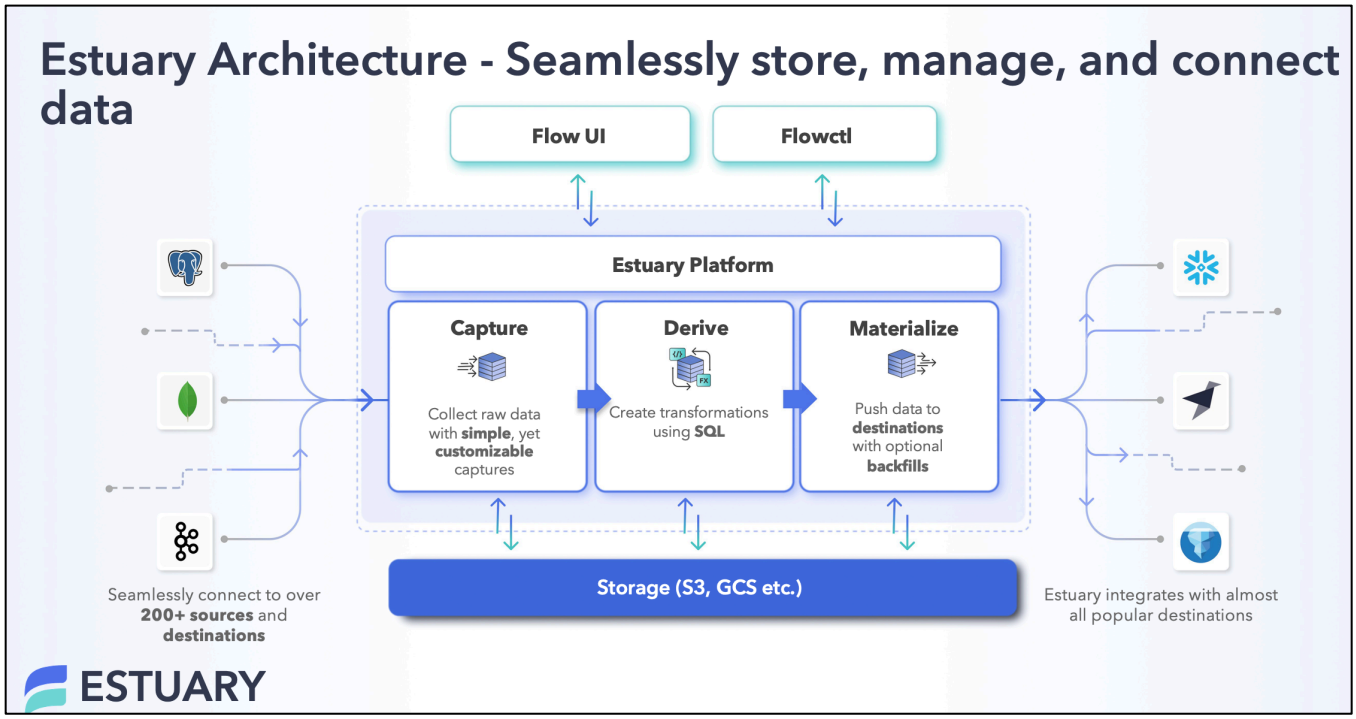
Estuary empowers both enterprise staff engineers and startup data engineers alike to build real-time streaming and ETL pipelines with unprecedented speed and simplicity. Estuary has more than 200+ built-in connectors and can connect almost all popular sources with all popular data warehouses leveraging Change Data Capture (CDC) to power your analytics, operations and AI.

Estuary's native flow, illustrated below, harmonizes information with ease. Batch ETL and real-time streams are handled under one roof.

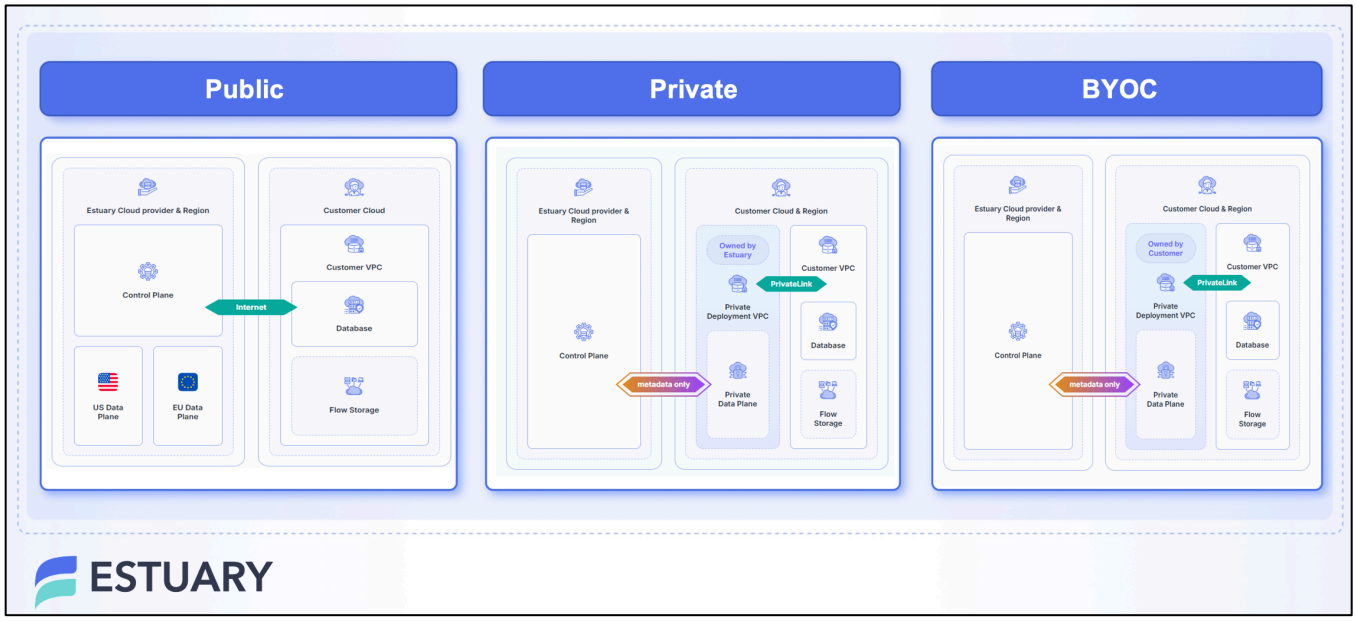
Regardless of source complexity, destination requirements, or in-flight transformation logic, Estuary's built-in engine handles any customer need through three simple steps. Capture -> Derive -> Materialize.

Whether you are working with intricate legacy systems, demanding real-time processing requirements, or sophisticated data transformations, Estuary abstracts away the underlying complexity.

Using the following architecture, Estuary moves 1 petabyte of data per month with less than 100 milliseconds of latency while maintaining 99.9% uptime.



Our global workforce spans Europe, Asia Pacific, and beyond, giving us deep insight into regional data sovereignty and deployment preferences. We understand the importance of EuroStack principles, where European companies prioritize deploying vendor solutions within their own controlled infrastructure environments. No other ETL or real-time streaming player offers deployment options like ours, which are listed below.



12.2Thank You

We would like to thank Christopher Daniel and his team at Analyze.Agency for helping us out.

12.3What Industry Leaders Say About Us

Estuary has built what most data movement teams need – a strong CDC mechanism hooked into a high performance, scalable streaming platform enabling fanout like Kafka, wrapped in an intuitive interface. Like Bruce Lee famously said, “Be like water”, data must flow unimpeded and it’s clear that the founders at Estuary share that vision, I continue to watch them closely.

— Chris Larsen, 25+ Year Data Infrastructure Veteran & Senior Director at Tradeverifyd

Estuary Flow transformed how we operationalize our data for fraud, security, support and beyond. Instead of unreliable, expensive backfills, we have real-time visibility into platform activity. The proactive support and hands-on approach make all the difference.

— Jonni Lundy, COO, Resend

Estuary has been a game-changer for Headset’s data infrastructure. Compared to our previous solutions, it has dramatically improved reliability while reducing our overall costs significantly.

— Scott Vickers, CTO, Headset

We needed something self-serve, fast and reliable, and Estuary delivered exactly that. It’s a huge unlock for our operations, reporting and machine learning.

— Uri Vinetz, Director of Data, Livable

Our migration away from Rockset would’ve been 100x harder without the unique capabilities that Estuary provides. We’re materializing transformations from Snowflake, DynamoDB and MySQL to our warehouse in under a second. Estuary unlocks incremental materialization for any datastore while also providing a Kafka interface for all of our sources and derived collections. We now have an immense amount of flexibility to support any type of workload on our data platform.

— Mays, Principal Engineer, Forward

12.4Sign Up

To conduct the tests for this data warehouse benchmark report, a minimum of 8TB of data was transferred into various systems. Regardless of whether the data was streamed or subjected to batch ETL processes, its ingestion into these systems was facilitated by Estuary at ease.

We provide a generous free tier and do not require a credit card for sign up.

We invite you to sign up for Estuary and discover its powerful data movement features for yourself.