

# GIGAOM

## Data Warehouse in the Cloud Benchmark



By William McKnight, and Jake Dolezal

Sponsored by [Microsoft](#)

# Data Warehouse in the Cloud Benchmark

01/22/2019

## Table of Contents

[1 Cloud Analytics Platform Offerings](#)

[2 Benchmark Setup](#)

[3 Benchmark Results](#)

[4 Price Per Performance](#)

[5 Conclusion](#)

[6 About William McKnight](#)

[7 About Jake Dolezal](#)

[8 About Microsoft](#)

[9 Copyright](#)

# 1 Cloud Analytics Platform Offerings

Big data analytics platforms load, store, and analyze volumes of data at high speed, providing timely insights to businesses. Data-driven organizations leverage this data, for example, for advanced analysis to market new promotions, operational analytics to drive efficiency, or for predictive analytics to evaluate credit risk and detect fraud. Customers are leveraging a mix of relational analytical databases and data warehouses to gain analytic insights.

This report focuses on relational analytical databases in the cloud, because deployments are at an all-time high and poised to expand dramatically. The cloud enables enterprises to differentiate and innovate with these database systems at a much more rapid pace than was ever possible before. The cloud is a disruptive technology, offering elastic scalability vis-à-vis on-premises deployments, enabling faster server deployment and application development, and allowing less costly storage. For these reasons and others, many companies have leveraged the cloud to maintain or gain momentum as a company.

This report outlines the results from an industry standard TPC Benchmark™ H (TPC-H)<sup>1</sup> to compare Amazon Redshift, Azure SQL Data Warehouse, Google Big Query, and Snowflake Data Warehouse—four relational analytical databases based on scale-out cloud data warehouses and columnar-based database architectures. Despite these similarities, there are some distinct differences in the four platforms.

## Amazon Redshift

Amazon Web Services Redshift was the first managed data warehouse cloud service and continues to get a high level of mindshare in this category. It indeed ticks all the table stakes boxes for a cloud analytic database.

In terms of storage, Amazon Redshift has a tight coupling between compute nodes and storage; data is stored on the compute nodes with a backup on AWS S3. While tight coupling keeps the data in close proximity to compute resources, it creates difficulty when scaling. Changing the instance size requires data movement to repartition and redistribute data on the new topology.

Amazon Redshift Spectrum can create an external table to store raw data on S3. You would not want to architect a steady diet of this form of data virtualization, but it is great to have it available for edge queries. Data virtualization has become very interesting for analyzing disparate data sets dispersed across multiple clouds, on-premises, and edge devices. . Redshift Spectrum has some future-proofing that a modern data engineering approach might utilize.

In support of diverse data, Amazon Redshift has a few nice features like the `JSON_EXTRACT_PATH_TEXT` function for noSQL key value databases. However, key value

nesting is limited and does not support large text columns like BLOBs, so it cannot store raw data in a Variable Character field (VARCHAR) larger than 64K.

Amazon Redshift is a fit for organizations needing a data warehouse with little to no administrative overhead and a clear, consistent pricing model. Amazon Web Services supports most of the databases in this report, and then some. Redshift is not the only analytic database on AWS, although sometimes this gets convoluted.

## Azure SQL Data Warehouse

Azure SQL Data Warehouse storage is separate from the compute Data Warehouse Unit (DWU). This enables Azure SQL Data Warehouse to scale columnar storage capacity and compute resources independently. This capability adjusts to various workload demands, offering potential cost savings when demand is low. Azure SQL Data Warehouse can pause and resume compute billing, where only storage is billed during the paused time. Azure SQL Data Warehouse achieves good balance in both configurability and simplicity, in a way that is both easy to administer and flexible in handling almost any usage pattern.

Azure SQL Data Warehouse is fully ANSI-SQL compliant and users familiar with SQL Server will be very comfortable using this environment.

Azure SQL Data Warehouse can export data to a local file the same way an on-premises SQL Server can, e.g., via the SQL Server Import and Export Wizard.

Although concurrency was not tested in the benchmark, Azure SQL Data Warehouse supports [128 concurrent](#) queries. This is many more than BigQuery, which supports a maximum [concurrency of 50](#) per project. Snowflake's maximum concurrency is difficult to calculate because it is a function of the number of queries, the submitted queries' execution plan, the size of the warehouse, and the maximum number of multi-cluster setting. In our experience we saw an X-Large (16 node) Snowflake warehouse run 6 concurrent simple scan queries (SELECT with a single column filter WHERE clause) before starting to queue. Thus, if we set the maximum multi-clusters at 5, we would likely hit a max concurrency of 30 in that scenario. Your results may vary.

## Google BigQuery

Google BigQuery is a managed service with some interesting distinctions. Architecturally, BigQuery normally is not configurable in terms of cluster sizes, node counts, server class, storage, or any of the other typical cloud hardware considerations. Google abstracts the details of the underlying hardware, database, and all configurations. You pay for the amount of data you query and store. You also pay additional fees for using SQL statements. (See the discussion on its SQL usage below.) Budgeting for BigQuery can be tough with potentially high variable costs. However, high usage customers can pre-purchase flat-rate computation "slots" or units in

increments of \$10,000 per month per 500 compute units.

BigQuery is not a data warehouse, *per se*, but a RESTful web service frontend for analyzing data in Google Storage. One might rightfully akin BigQuery more to a technology like MapReduce, rather than a conventional data warehouse. It behaves more like an API with an optional SQL-like window dressing called Data Manipulation Language (DML). Even more interesting, BigQuery incurs additional fees and quotas for the usage of DML above and beyond storage and compute. Users pay for bytes processed by data manipulation statements, including INSERT, UPDATE, DELETE, and MERGE; as well as data definition statements, such as CREATE TABLE AS SELECT.

Additionally, BigQuery has further [limitations](#) when it comes to data manipulation operations (INSERT, UPDATE, and DELETE). These include, but are not limited to:

- If concurrent UPDATE statements are submitted, only one will succeed and the rest will fail.
- Rows written recently via a streaming method cannot be modified using UPDATE, DELETE, or MERGE. (According to Google’s documentation, “recent” roughly means “in the past 30 minutes.”)
- MERGE statements cannot be run concurrently with UPDATE or DELETE statements.

Essentially, given its API-like qualities, BigQuery is natively designed for appending to or completely overwriting existing tables. Loading data cannot be achieved with SQL commands (such as COPY). Users who need to manipulate data after loading must take care when designing jobs to update, upsert, or delete existing records.

## Snowflake Data Warehouse

As a cloud-only, fully managed solution, Snowflake has a clear separation between compute and storage. For Snowflake on AWS, which is what we used for the queries, data is stored in AWS S3 and is cached when queries are executed to bring the data in closer proximity to compute resources. Snowflake essentially offers two configuration “levers” — the size of the warehouse cluster and how many clusters are permitted to spin up to handle concurrency. Snowflake scales by cluster server count in powers of 2 (i.e., 1, 2, 4, 8, 16, and so on.) If enabled, Snowflake will spin up additional clusters to handle multi-user concurrent query workloads. If not enabled, it will place paused queries in a queue until resources free up.

While simple and mostly abstracted for the user, this lack of configurability is a tradeoff and, in our experience, performance is impacted. In our estimation, Snowflake performance hinges greatly on cache. If Snowflake must access Amazon S3 for additional data, query execution slows dramatically. Historically, Snowflake has not disclosed the server class or size of its AWS EC2 instances. Thus, one cannot calculate beforehand what configuration would be most optimal for workloads — it is an exercise in trial and error.

Additionally, Snowflake's multi-cluster option is quite expensive as costs multiply rapidly when multiple clusters are employed. For example, if a customer is paying the \$4 per hour rate for full support and security, an X-Large 16-node cluster normally costs \$64 per hour to run. If users hit Snowflake with a high volume of concurrent query requests (for example, at the end of month), Snowflake will spin up an additional four clusters to handle the workload. This would spike the cost to \$320 per hour. Snowflake would automatically spin the additional clusters down once demand has passed, which helps, but predicting and budgeting cost for performance can be tricky with Snowflake.

Snowflake supports an ANSI-compliant form of SQL. However, we experienced syntax differences when converting existing queries to Snowflake.

Snowflake only supports data export to S3 (on AWS). However, it is possible to use the Snowflake command line interface results to direct data to a file in a Linux shell.

*Table 1. Platform Summary*

	<b>Azure SQL Data Warehouse</b>	<b>Snowflake Data Warehouse</b>	<b>Google BigQuery</b>	<b>Amazon Redshift</b>
<i>Company</i>	Microsoft	Snowflake Computing	Google	Amazon
<i>First Released</i>	1989 (SQL Server) 2005 (Analytics Platform System) 2016 (Azure SQL Data Warehouse)	2014	2010	2014
<i>Current Version</i>	12.0.2000.8	3.0.5	October 11, 2018 release	1.0.1583
<i>SQL</i>	Transact-SQL	Snowflake SQL	DML	PostgreSQL 8
<i>Scale Out Cloud Data Warehouse</i>	√	√		
<i>Columnar</i>	√	√	√	√

The time to reach all worldwide regions for this release of Microsoft Azure SQL DW is January-February 2019.

<sup>1</sup> More can be learned about the TPC-H benchmark at <http://www.tpc.org/tpch/>.

## 2 Benchmark Setup

The benchmark equivalency was established by price, selecting the lowest, no support options across the four platforms. DW30000C costs \$362.40/hour to run. 4X-Large Snowflake @ \$3.00/node costs \$384/hour. In the Price-Performance calculation below, we used the lowest, no support \$2.00/node price for Snowflake. The parameter values for the queries used across all vendors are informed by the TPC Benchmark™ H (TPC-H)<sup>2</sup> spec validation queries. This is not an official TPC benchmark. The queries were executed using the following setup, environment, standards, and configurations.

## Benchmark Data

The data sets used in the benchmark were a workload derived from the well-recognized industry standard TPC Benchmark™ H (TPC-H)<sup>3</sup>.

From tpc.org: “The TPC-H is a decision support benchmark. It consists of a suite of business-oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions.”

To show the data model, the following diagram was taken from page 13 of the TPC-H Revision 2.17.3 [specification document](#).

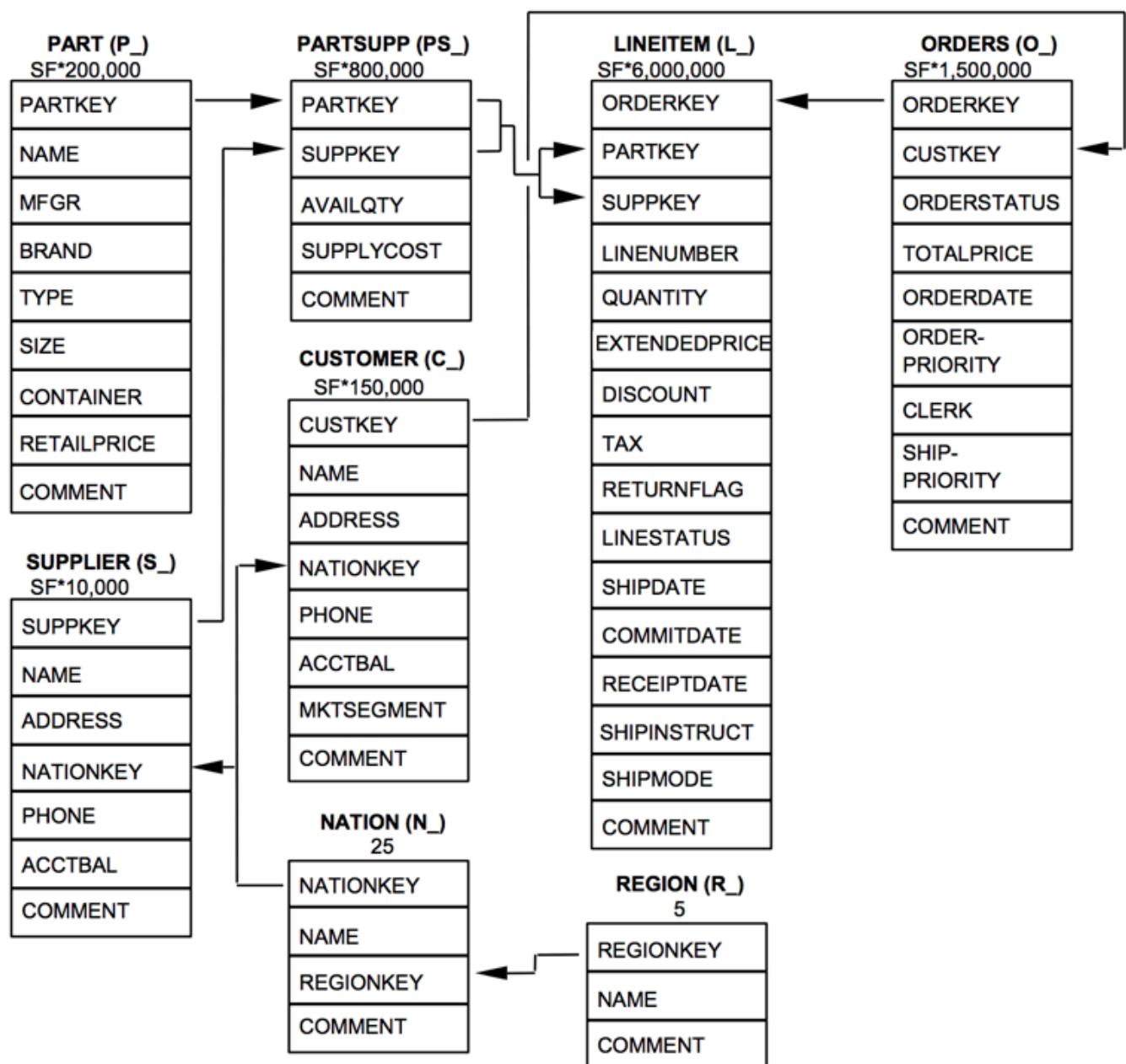


Figure 1. TPC-H Data Model

To give an idea of the data volumes used in our benchmark, the following table gives row counts of the database when loaded with 30TB of TPC-H data:



*Table 2. TPC-H Database Row Count given 30TB*

TPC-H Table	30TB Row Count
Customer	4,500,000,000
Line Item	180,000,000,000
Orders	45,000,000,000
Part	6,000,000,000
Supplier	300,000,000
PartSupp	24,000,000,000

## Cluster Environments

Our benchmark included ten (10) different cluster environments:

- three for Azure SQL Data Warehouse (DW5000C, DW15000C, and DW30000C),
- three for Amazon Redshift (dc2.8xlarge 10N, dc2.8xlarge 30N, dc2.8xlarge 60N with (32 CPUs 244GB RAM each),
- three for Snowflake Data Warehouse (2X-Large, 3X-Large, and 4X-Large),
- and one for Google, for which the underlying architecture is unknown and the environment is scaled automatically

## Queries

We sought to replicate the TPC-H Benchmark queries modified only by syntax differences required by the platforms. The benchmark is a fair representation of enterprise query needs. The TPC-H testing suite has 22 queries, which are described by the table below. The SQL operations parameters used in each query were tpc-h spec validation queries and are noted.

Table 3. TPC-H Query Parameters

Q#	Description	Sum	Sub-query	Join	Min/Max	Avg	Count	Top/Limit
1	Pricing Summary Report	✓				✓	✓	
2	Minimum Cost Supplier		✓		✓			✓
3	Shipping Priority	✓						✓
4	Order Priority Checking		✓				✓	
5	Local Supplier Volume	✓						
6	Forecasting Revenue Change	✓						
7	Volume Shipping	✓	✓					
8	National Market Share	✓	✓					
9	Product Type Profit Measure	✓	✓					
10	Returned Item Reporting	✓						✓
11	Important Stock Identification	✓	✓					
12	Shipping Modes and Order Priority	✓						
13	Customer Distribution		✓	✓			✓	
14	Promotion Effect	✓						
15	Top Supplier		✓		✓			
16	Parts/Supplier Relationship		✓				✓	
17	Small Quantity Order Revenue	✓	✓			✓		
18	Large Volume Customer	✓	✓					✓
19	Discounted Revenue	✓						
20	Potential Part Promotion	✓	✓					
21	Suppliers Who Kept Orders Waiting		✓				✓	✓
22	Global Sales Opportunity	✓	✓			✓	✓	

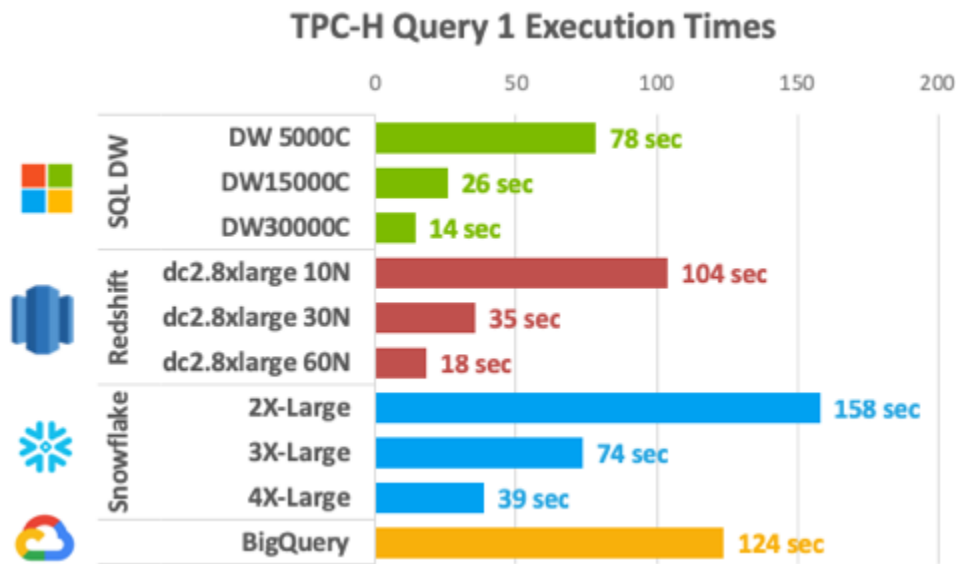
<sup>2</sup> More can be learned about the TPC-H benchmark at <http://www.tpc.org/tpch/>.

<sup>3</sup> More can be learned about the TPC-H benchmark at <http://www.tpc.org/tpch/>.

### 3 Benchmark Results

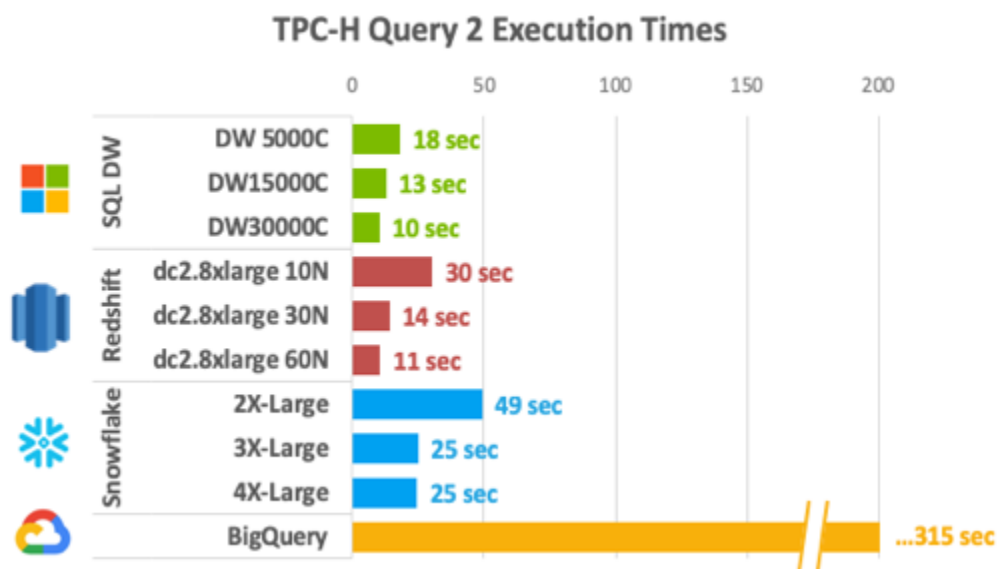
This section analyzes the query results from the fastest runs of the three sets of 22 TPC-H queries described in Table 3.

Table 4. TPC-H Query 1: “Pricing Summary Report” Execution Times



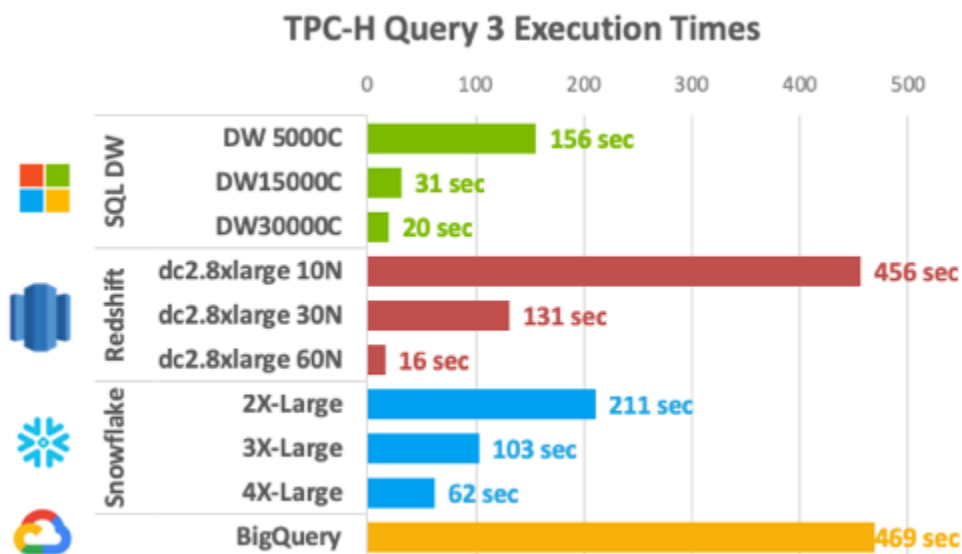
Query 1 is the only query that uses only Sum, Average, and Count. All Azure SQL Data Warehouse configurations were faster than Google and each configuration was faster than the analogous Snowflake and Redshift configurations.

Table 5. TPC-H Query 2: “Minimum Cost Supplier” Execution Times



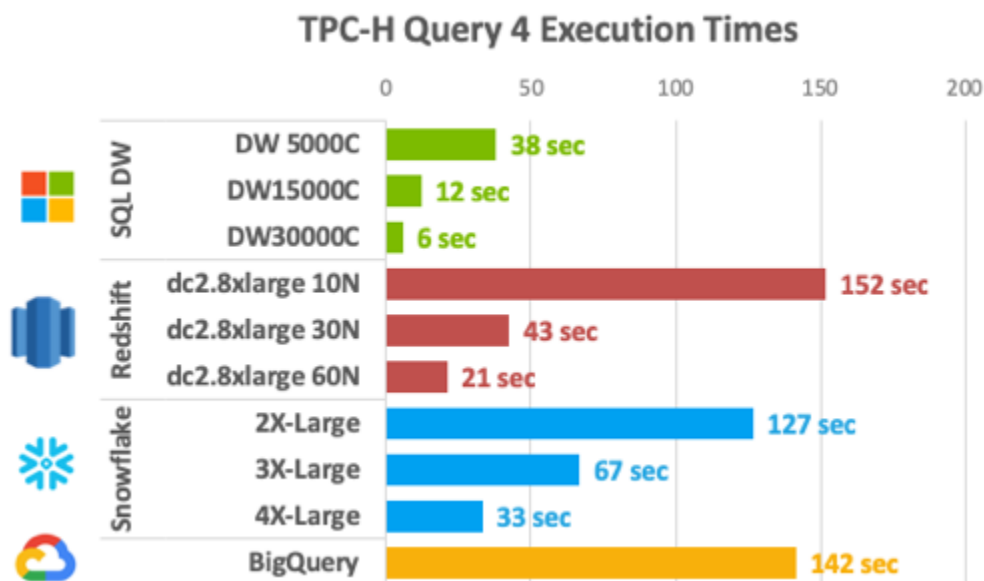
Query 2 was one of two queries that contained a Min/Max. Google had some challenge with the query. Each Azure SQL Data Warehouse configuration was faster than the analogous Snowflake and Redshift configurations.

Table 6. Query 3: “Shipping Priority” Execution Times



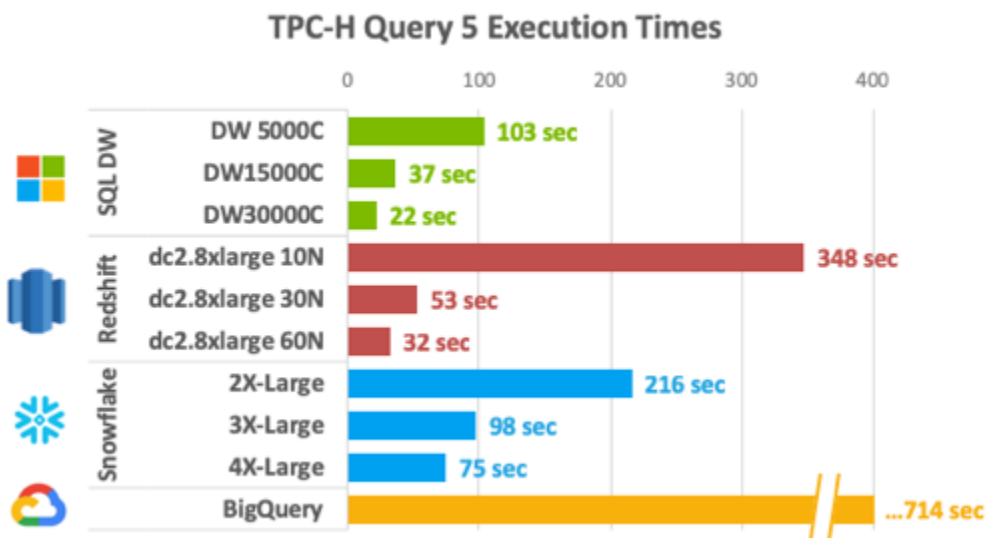
Google and the 10N for Redshift had some challenge with the query and each Azure SQL Data Warehouse configuration was faster, twice by 3X+, than the analogous Snowflake configuration.

Table 7. Query 4: “Order Priority Checking” Execution Times



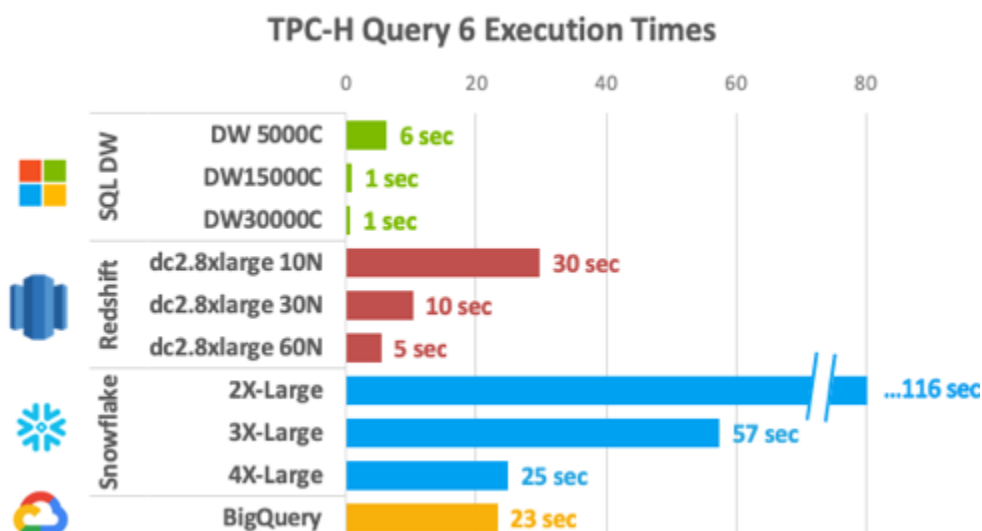
In Query 4, with a subquery and a count, we see the familiar pattern of relative performance except Google performance is closer to the Snowflake 2X Large performance.

Table 8. Query 5: “Local Supplier Volume” Execution Times



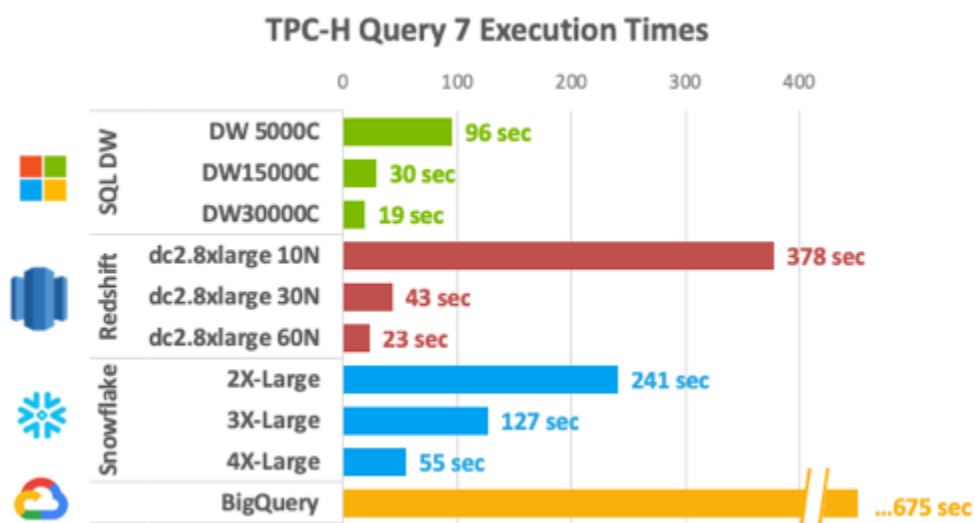
Query 5, which only employs a sum aggregation, favored Azure SQL Data Warehouse as well.

Table 9. Query 6: “Forecasting Revenue Change” Execution Times



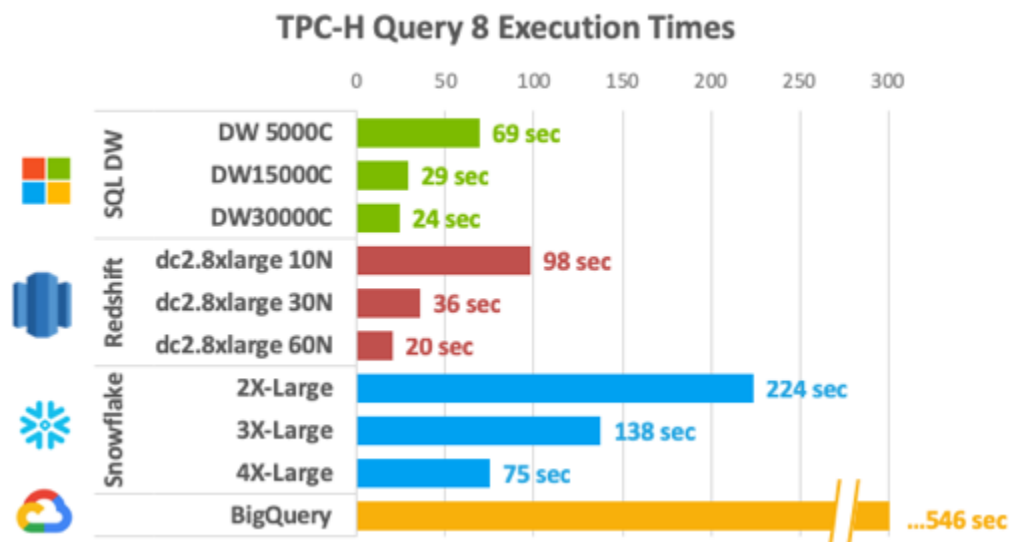
The simple SUM of Query 6 was a very high relative performer for Azure SQL Data Warehouse.

Table 10. Query 7: “Volume Shipping” Execution Times



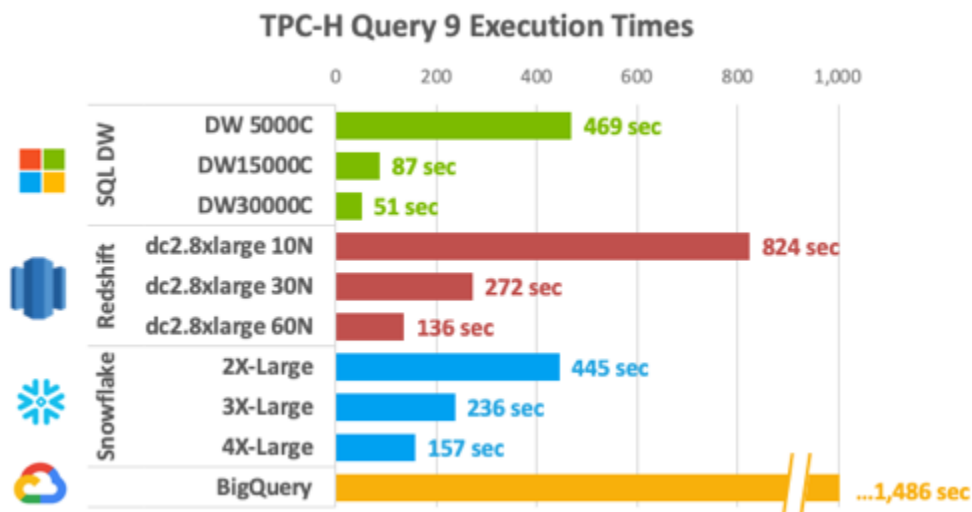
Again, we see the familiar pattern of 2X+ performance between Azure SQL Data Warehouse and the analogous Snowflake configuration, improvement to the Redshift configurations and Google at a different level.

Table 11. Query 8: “National Market Share” Execution Times



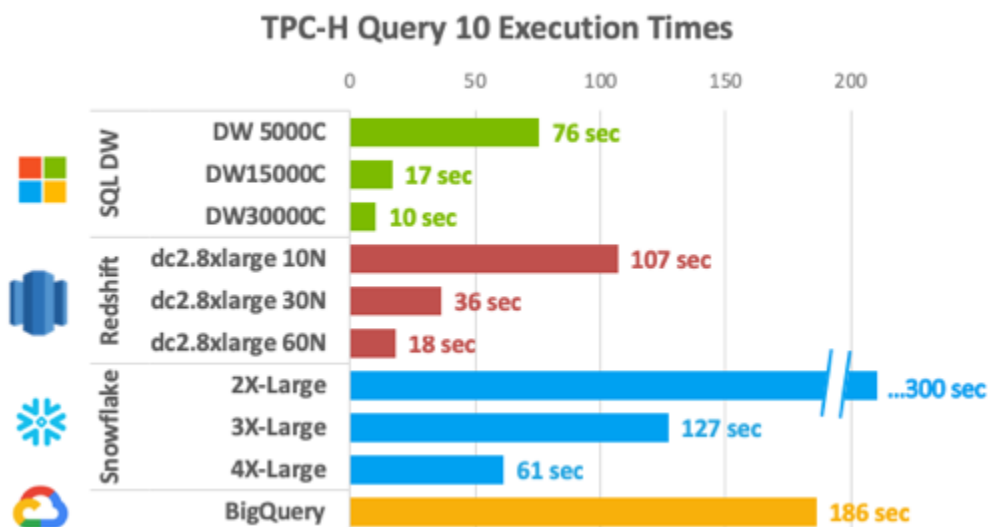
Query 8 performance was similar to Query 7.

Table 13. Query 9: “Product Type Profit Measure” Execution Times



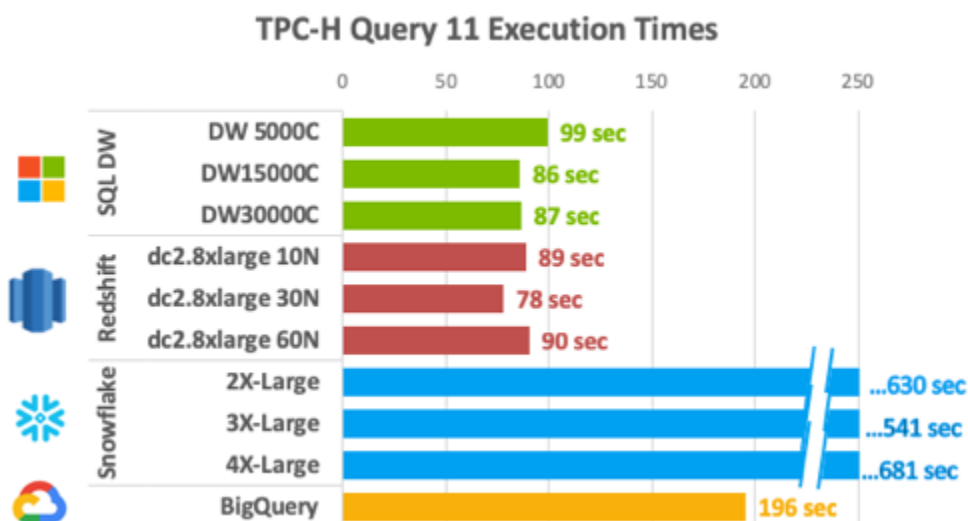
The Snowflake 2X-Large outperformed Azure SQL Data Warehouse 5000C in Query 9.

Table 14. Query 10: “Returned Item Reporting” Execution Times



Query 10 uniquely has a Sum and a Top/Limit. Google, with a runtime of 186 seconds, outperformed the Snowflake 2X-Large. Azure SQL Data Warehouse was the top performer.

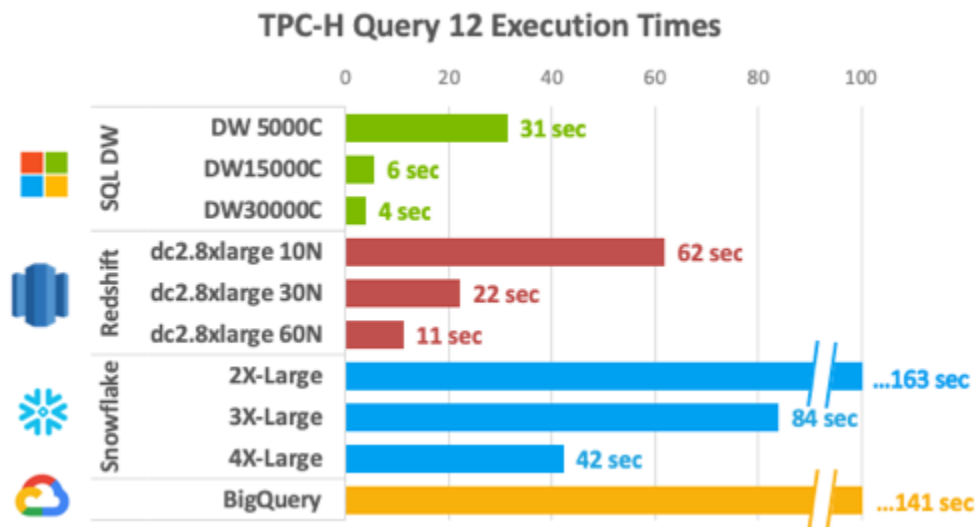
Table 15. Query 11: “Important Stock Identification” Execution Times



Query 11 (another sub-select and a sum operation) saw performance slightly favoring Redshift.

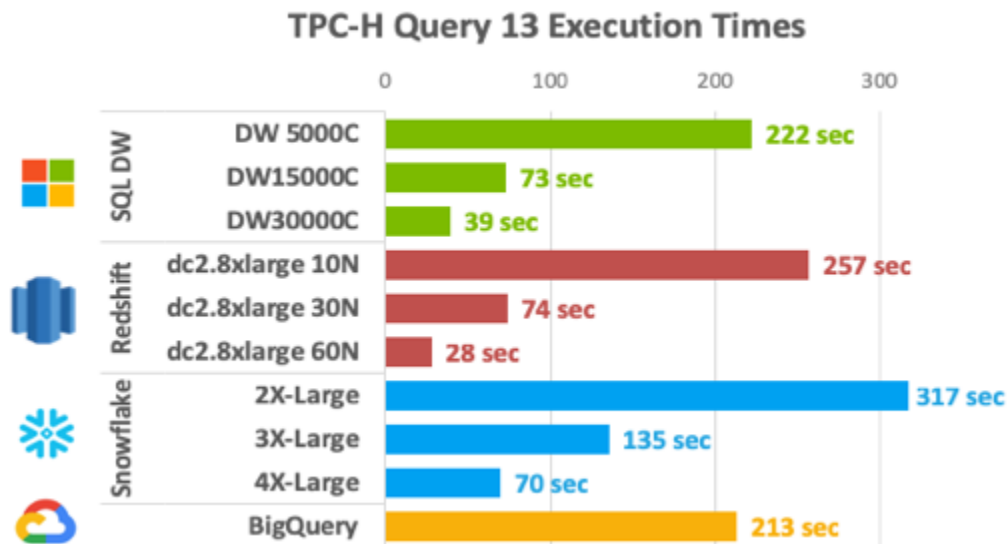


Table 16. Query 12: “Shipping Modes and Order Priority”



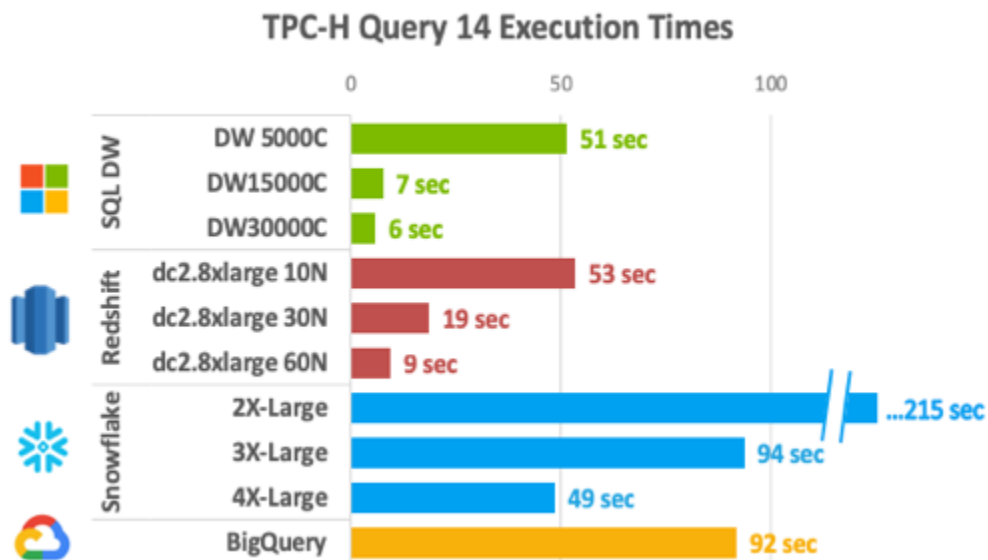
Query 12, another simple SUM, was another plus performance query for Azure SQL Data Warehouse, with Azure SQL Data Warehouse outperforming analogous Snowflake configurations by 5, 14 and 10 times, Redshift by appx. 2, 4 and 3 times and Google by 4 times (5000 C).

Table 17. Query 13: “Customer Distribution” Execution Times



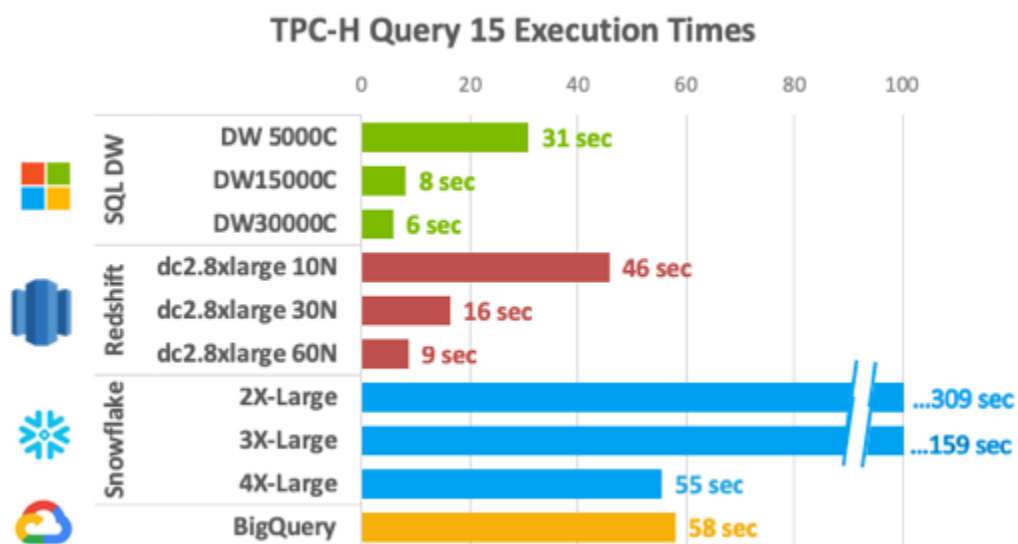
Query 13 is the only TPC-H query with an explicit JOIN. Google outperformed the low-end configurations of Azure SQL Data Warehouse, Redshift and Snowflake.

Table 18. Query 14: “Promotion Effect” Execution Times



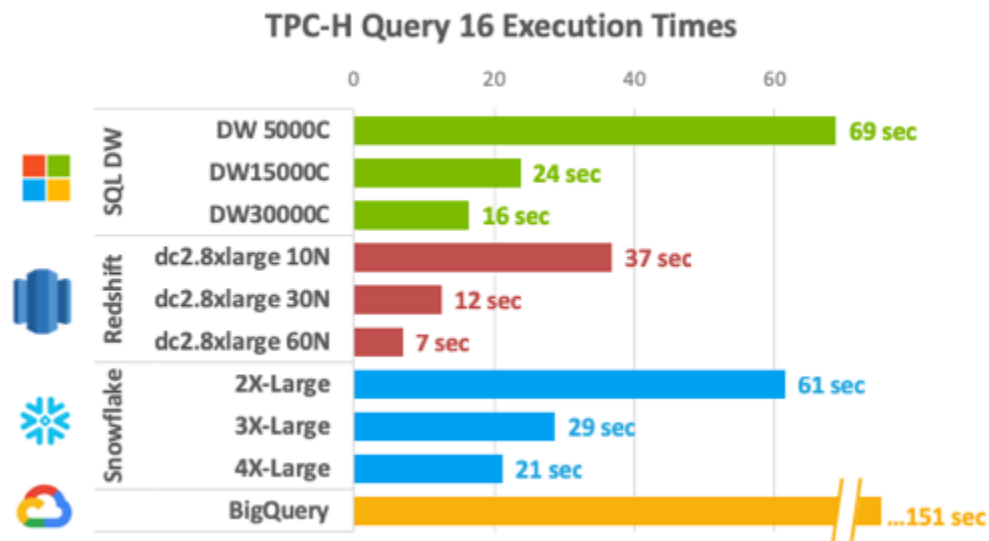
For Query 14, Azure SQL Data Warehouse took only an impressive 7 and 6 seconds on the 15000C and 30000C configurations.

Table 19. Query 15: “Top Supplier” Execution Times



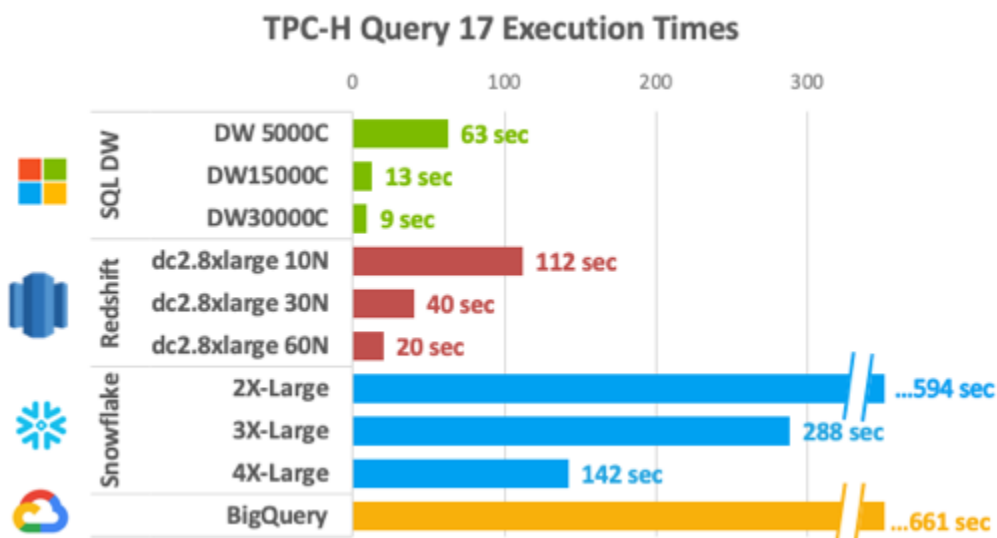
Another query with very fast performance from Azure SQL Data Warehouse, Redshift being competitive and Google being more competitive.

Table 20. Query 16: “Parts/ Supplier Relationship” Execution Times



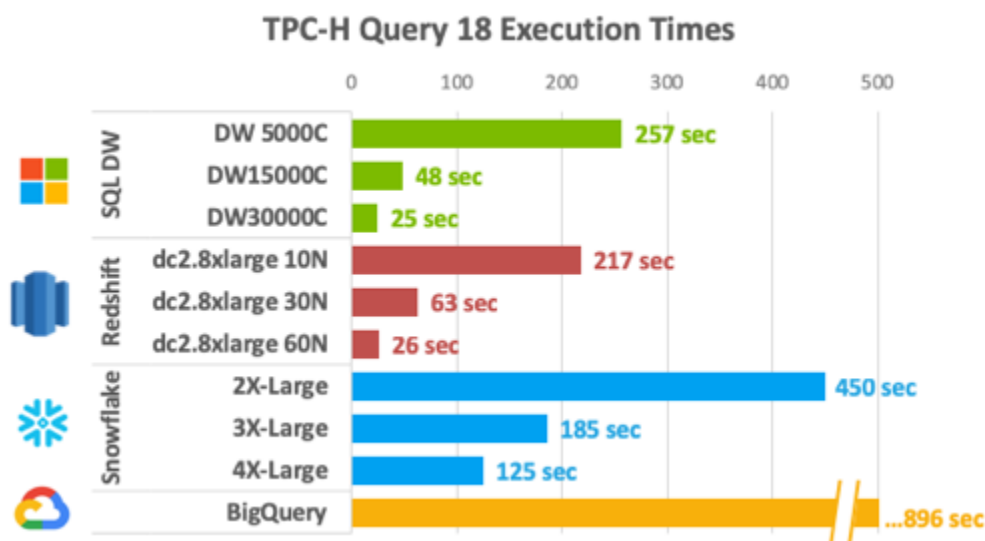
For Query 16, The Snowflake 2X-Large outperformed Azure SQL DW 5000C, as did all of the Redshift configurations.

Table 21. Query 17: “ Small Quantity Order Revenue” Execution Times



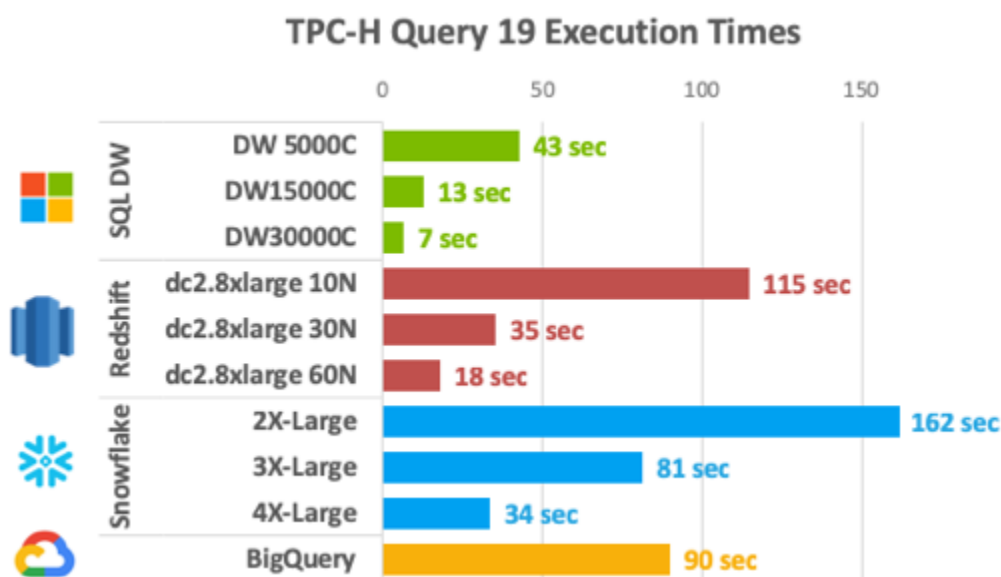
Query 17 yielded performance advantages for Azure SQL Data Warehouse.

Table 22. Query 18: “Large Volume Customer” Execution Times



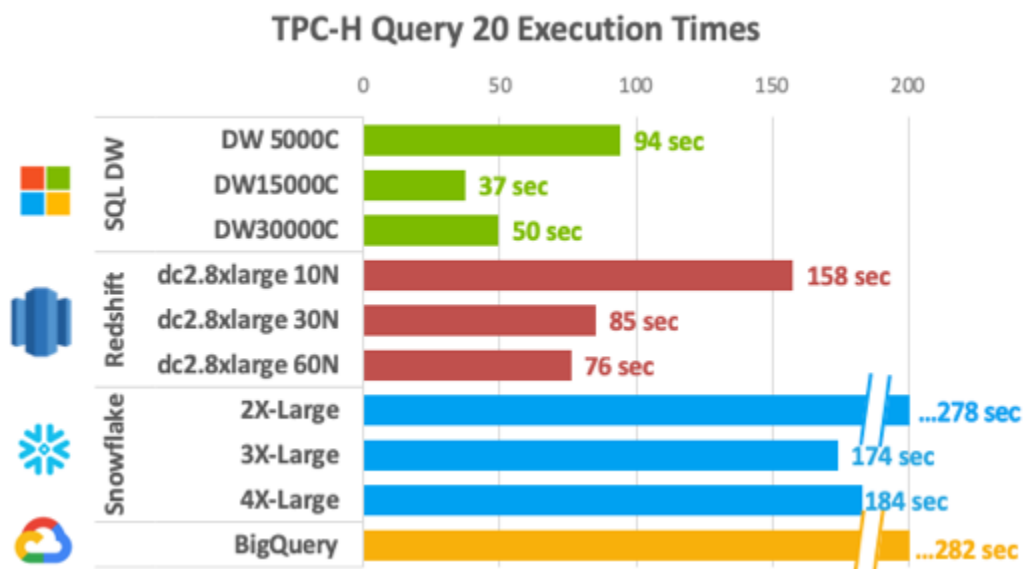
It is worth pointing out that Snowflake 3X-Large and 4X-Large outperform the DW 5000C, but the 15000C and 30000C outperform all like configurations.

Table 23. Query 19: “Discounted Revenue” Execution Times



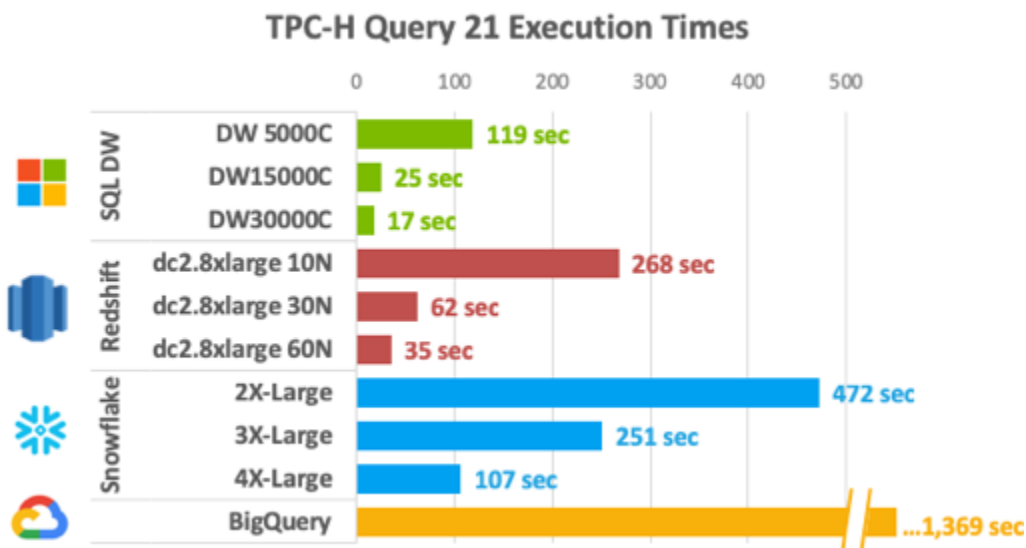
Query 19, a SUM, was another strong performer for Azure SQL Data Warehouse, with even the 5000C outperforming all except the Snowflake and Redshift high specifications.

Table 24. Query 20: “Potential Part Promotion” Execution Times



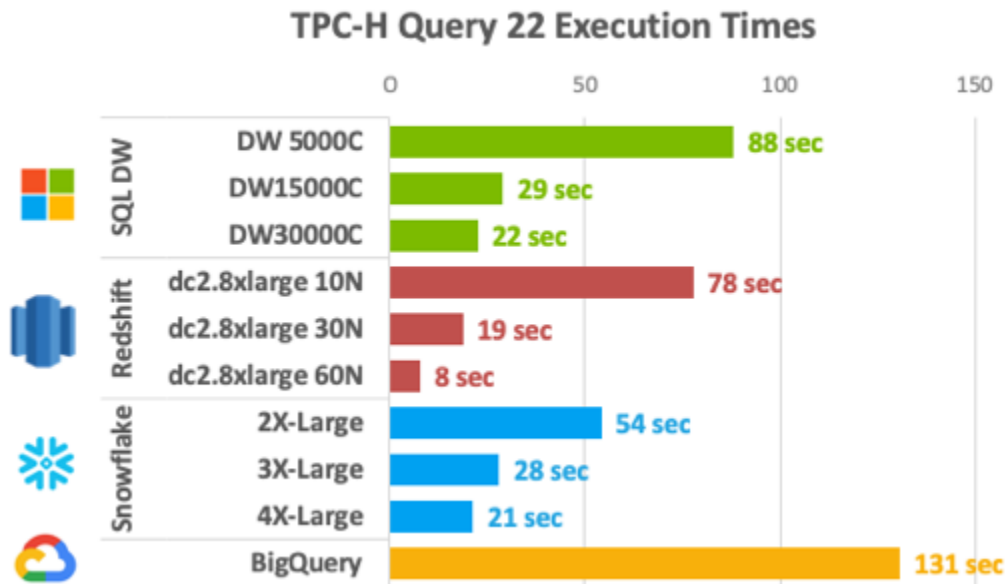
Query 20 showed Azure SQL Data Warehouse to be 3X+ as fast or faster in analogous configurations to Snowflake, faster than analogous configurations to Redshift and, worst case, 3X faster than Google.

Table 25. Query 21: “Suppliers Who Kept Orders Waiting” Execution Times



Query 21 shows the familiar pattern of high performance differences between Azure SQL Data Warehouse and the analogous Snowflake configuration and Google at a different level.

Table 26. Query 22: “Global Sales Opportunity” Execution Times



Snowflake outperformed Azure SQL Data Warehouse in all 3 analogous configurations for Query 22, as did Redshift.

## Benchmark Observed Results

Azure SQL Data Warehouse outperformed the analogous configuration of Snowflake in 62 of the 66 (94%) best times of the TPC-H query runs. Azure SQL Data Warehouse outperformed the analogous configuration of Redshift in 54 of the 66 (82%) best times of the TPC-H query runs. Azure SQL Data Warehouse also outperformed Google BigQuery in all 100% of their respective TPC-H best times.

## Google BigQuery Performance Considerations

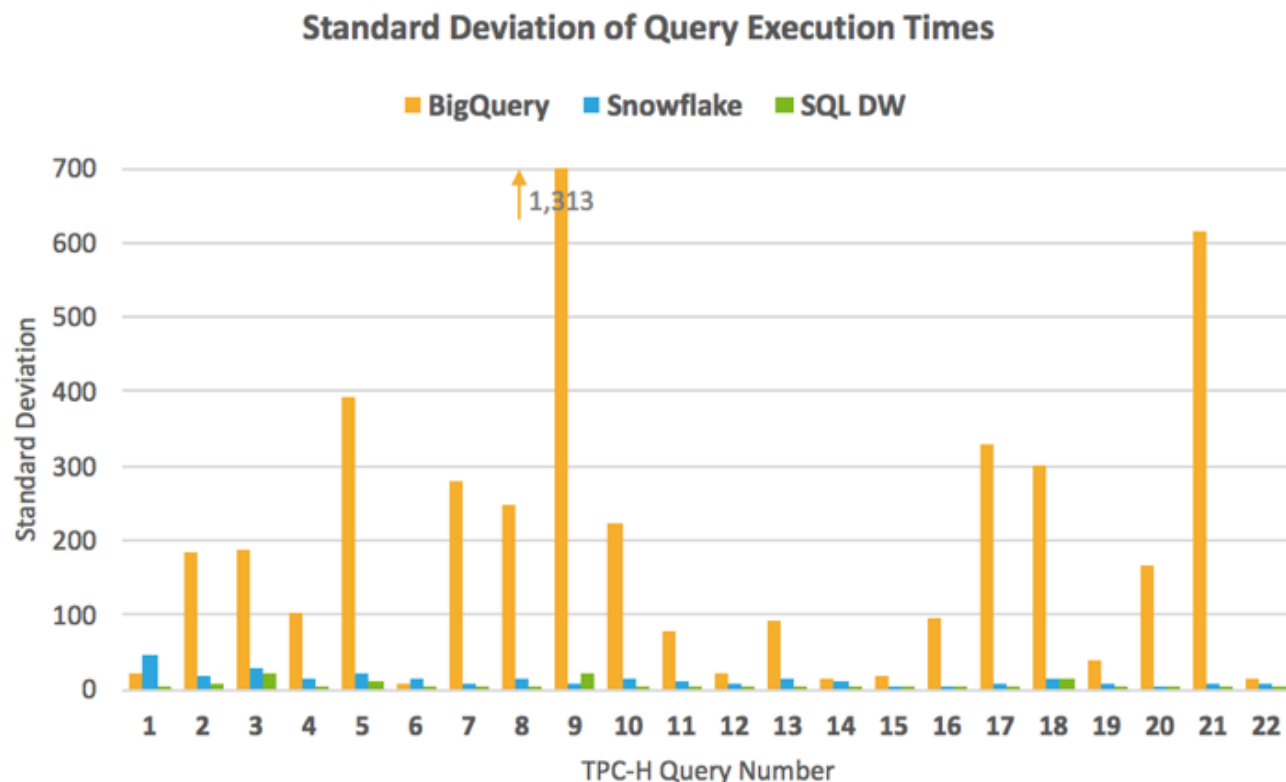
As previously stated, Google BigQuery charges by the byte processed not by the hour, like most cloud analytics vendors. Even though performance does not inherently affect how much you pay, we still made the price for performance calculations for BigQuery in order to understand how they compared to Azure SQL Data Warehouse and Snowflake.

During our benchmark testing, we experienced a high variance in execution times with BigQuery. See the included chart of the standard deviation of execution times in seconds.

As you can see, Azure SQL Data Warehouse and Snowflake’s performance was relatively consistent on the same queries from run to run. However, BigQuery had very wide ranges. (We chose best run instead of the average among the competitors to give BigQuery the benefit of the doubt.) For example, BigQuery’s best time was 42 minutes running TPC-H query #9.

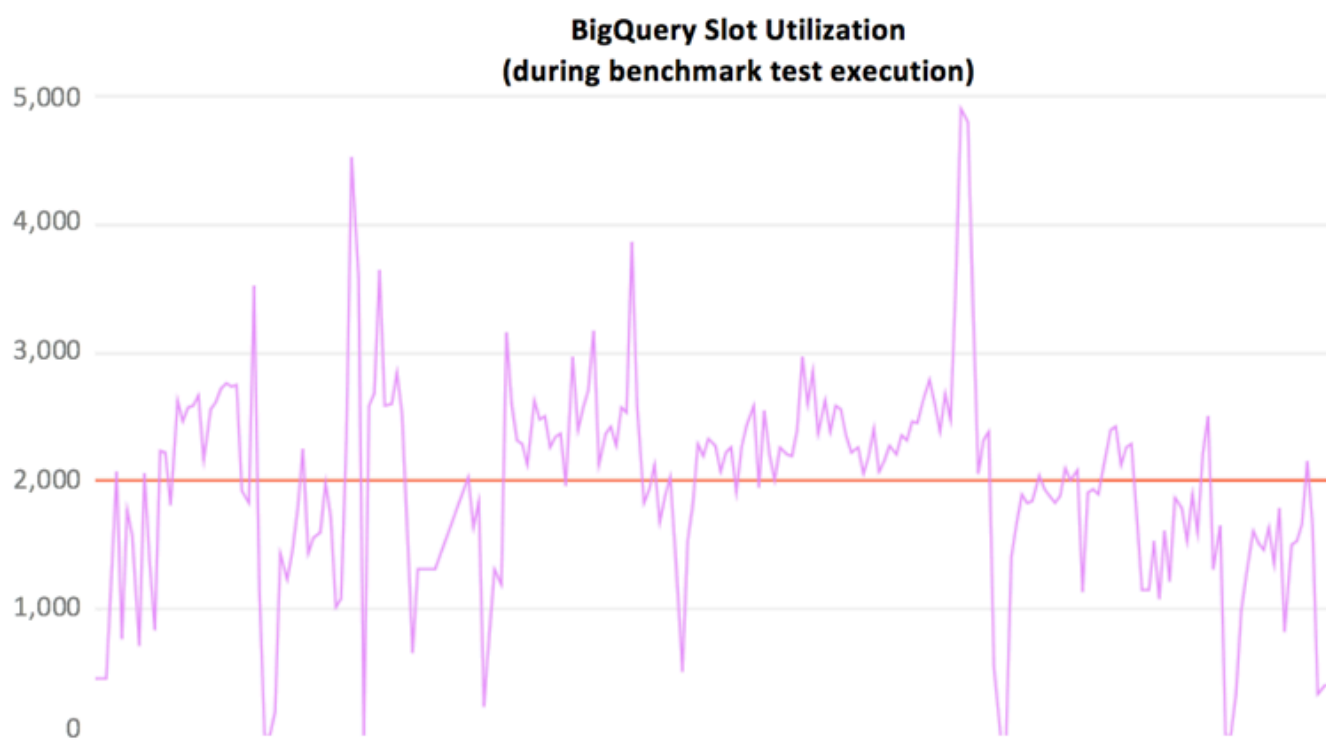
However, it's worst time was 1 hour and 8 minutes. Twenty-six minutes is atypical for a query execution range.

Figure 2. Standard Deviation of Query Execution Times



One possible hypothesis as to why BigQuery performance was so erratic could be the shared, multi-tenancy architecture. Most cloud vendors allow you to choose (or only offer) dedicated hardware for your platform. BigQuery does not offer dedicated hardware. Noisy neighbors can impact performance. To illustrate, see the following chart of the slot (or compute unit) utilization for BigQuery during our benchmark runs. Please note that it would be very difficult to map slots utilized with actual queries. We also are not certain how noisy our same-rack neighbors were during the runs, but we speculate it caused the high variance we experienced.

Figure 3. BigQuery Slot Utilization (during benchmark test execution)



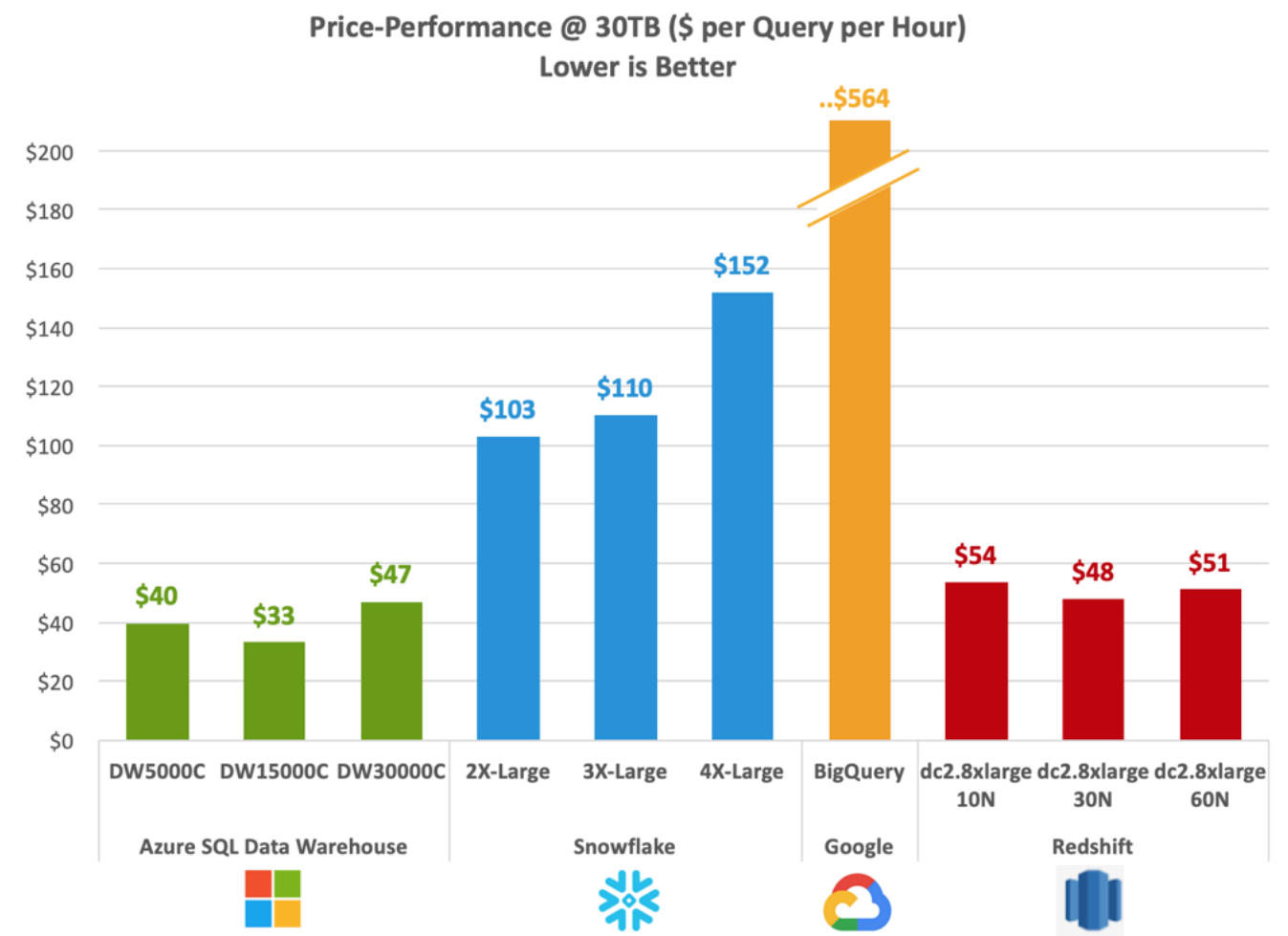


# 4 Price Per Performance

To measure price-performance we used the TPC-H metric of cost of ownership divided by composite query per hour. If you contiguously ran all 22 of these queries to completion of the set, the cost at an hourly basis is indicated in the chart below. Azure SQL Data Warehouse provided consistently best price-performance in our queries.

For Azure SQL Data Warehouse, Redshift and Snowflake, you pay for compute resources as a function of time. With Google BigQuery, you pay for bytes processed. During a single run of the TPC-H suite, we processed roughly 113TB of data at \$5 per TB for BigQuery. For the others, we took the total execution time elapsed for all 22 queries and extrapolated the hourly rate we were paying for each cluster class.

Figure 4. Price-Performance @ 30TB (\$ per Query per Hour)



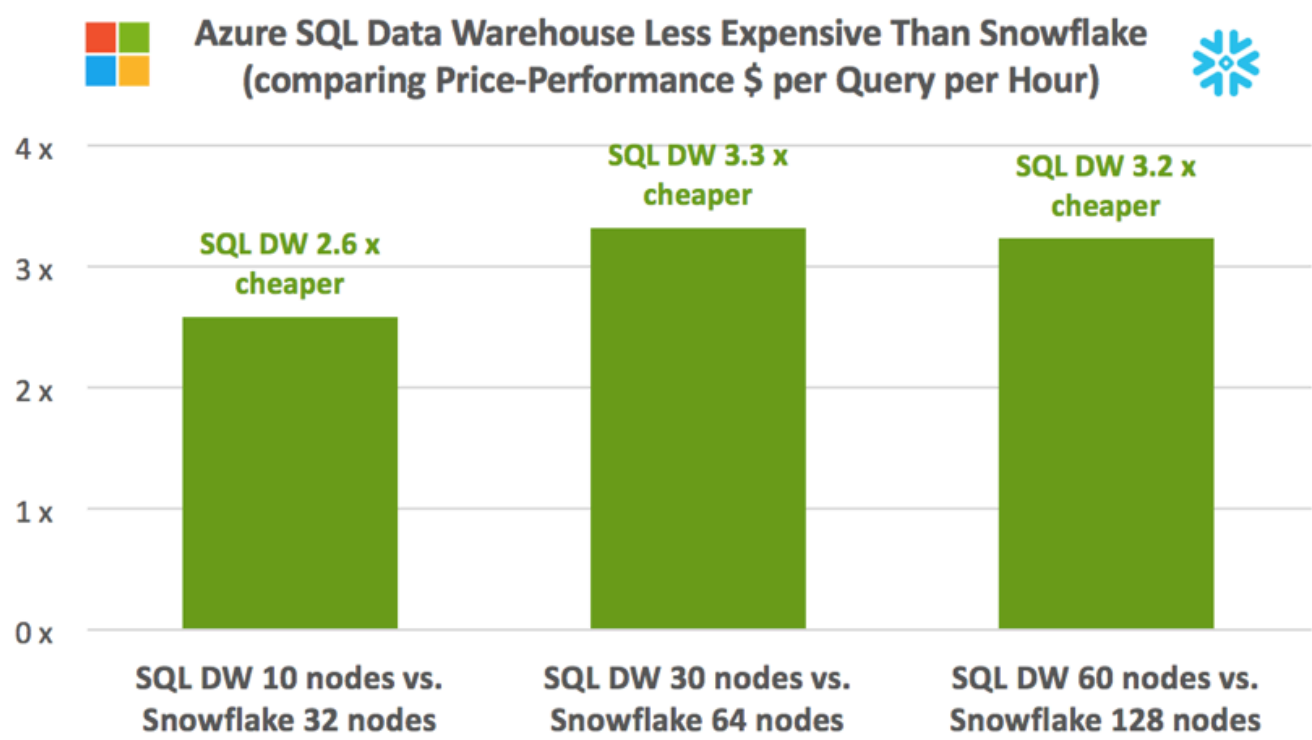
In a direct comparison of Azure SQL Data Warehouse and Snowflake, Figure 5 aligns the following cluster classes:

Figure 5. Comparable cluster nodes for Azure SQL Data Warehouse and Snowflake

Azure SQL Data Warehouse			Snowflake	
DW5000C	10 nodes	→	2X-Large	32 nodes
DW15000C	30 nodes	→	3X-Large	64 nodes
DW30000C	60 nodes	→	4X-Large	128 nodes

Charting the comparison, Azure SQL Data Warehouse ran the TPC-H queries roughly 3 times cheaper than Snowflake in terms of dollars per query per hour across all three clusters. See Figure 6.

Figure 6. Azure SQL Data Warehouse Less Expensive Than Snowflake



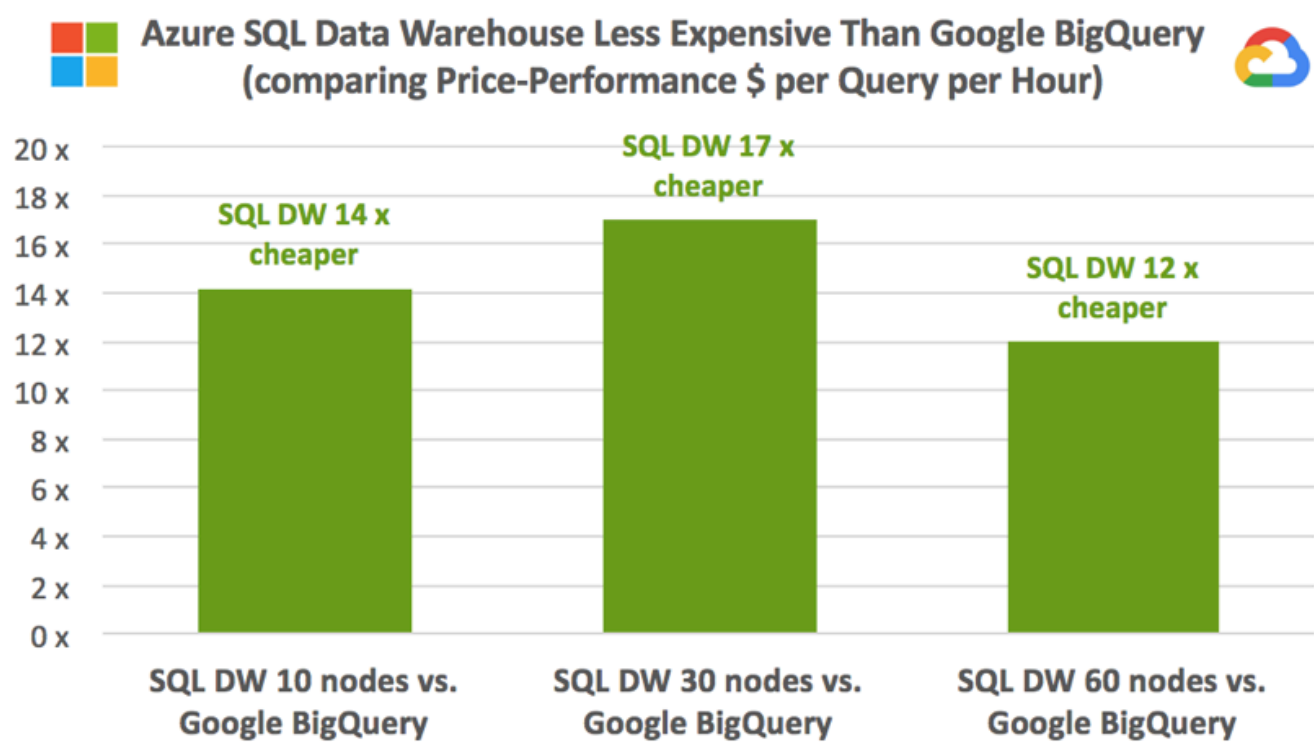
Since Google BigQuery does not have a scale factor, we could only compare the different scaled SQL Data Warehouse classes against BigQuery, whose scale varies automatically with slot (compute unit) allocation chosen by the Google engine.

Figure 7. Comparable cluster nodes for Azure SQL Data Warehouse and Google BigQuery

Azure SQL Data Warehouse			Google BigQuery
DW5000C	10 nodes	→	
DW15000C	30 nodes	→	
DW30000C	60 nodes	→	

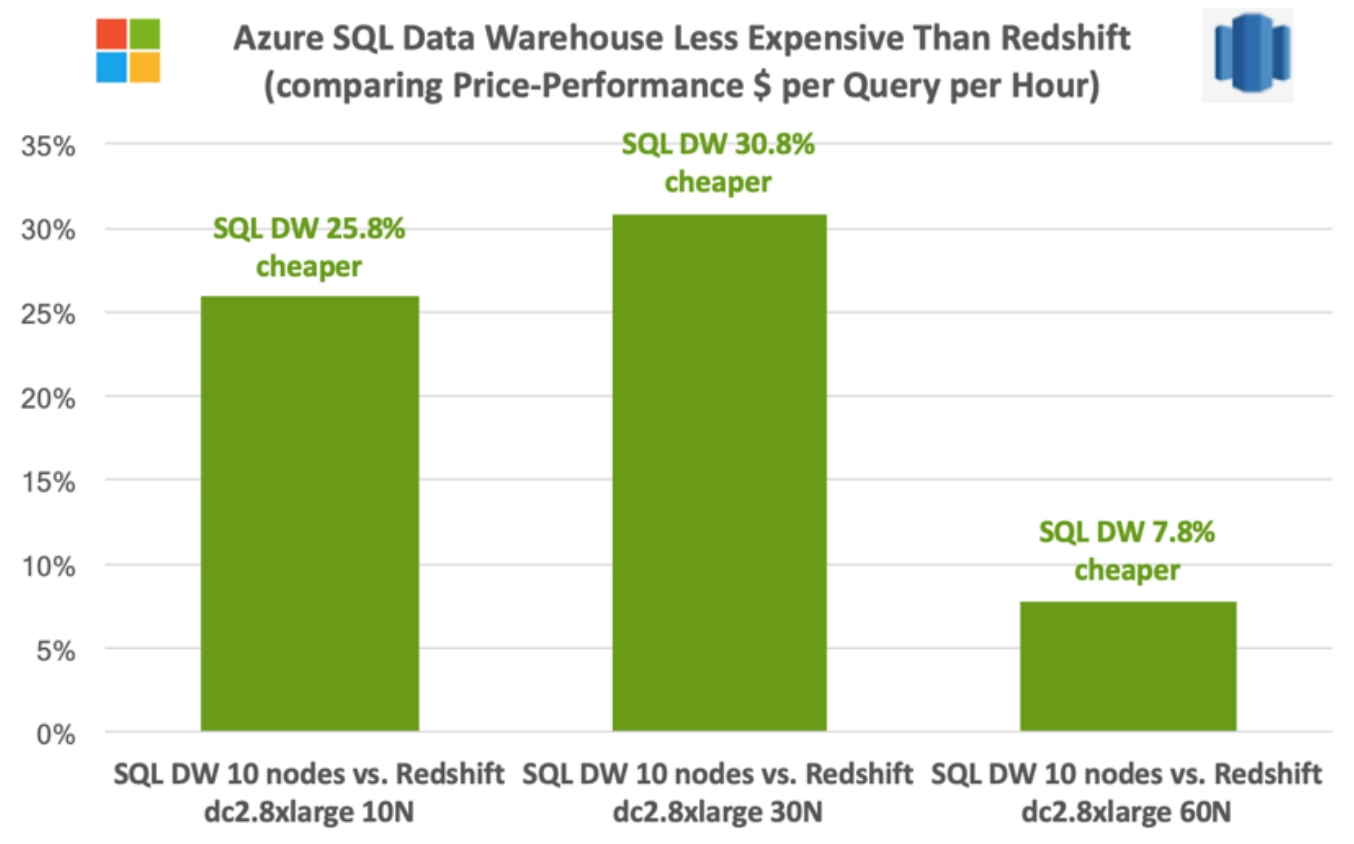
Charting the comparison shows Azure SQL Data Warehouse ran the TPC-H queries between 12 and 17 times more cost-effectively than Google BigQuery in terms of price per query per hour across all three clusters. See Figure 8.

Figure 8. Azure SQL Data Warehouse Less Expensive Than Google BigQuery



Charting the comparison shows Azure SQL Data Warehouse ran the TPC-H queries between 8% and 31% more cost-effectively than Amazon Redshift in terms of price per query per hour across all three clusters. See Figure 9.

Figure 9. Azure SQL Data Warehouse Less Expensive Than Amazon Redshift



## 5 Conclusion

Cloud databases are a way for enterprises to avoid large capital expenditures, provision quickly, and provide performance at scale for advanced analytic queries. Relational databases with analytic capabilities continue to support the advanced analytic workloads of the organization with performance, scale, and concurrency. In a representative set of corporate-complex queries from the well-known TPC-H standard, Azure SQL Data Warehouse consistently outperformed the competition.

Overall, the benchmark results were insightful in revealing query execution performance and some of the differentiators for Azure SQL Data Warehouse, Snowflake Data Warehouse, Amazon Redshift, and Google Big Query. Azure SQL Data Warehouse query response times on the 30TB TPC-H data set were overall seven times faster than Snowflake across all cluster classes, approximately twice the performance of Redshift across all cluster classes and 14 times faster than BigQuery (compared to the SQL DW15000C).

In terms of price per performance, Azure SQL Data Warehouse ran the TPC-H queries roughly 3 times cheaper than Snowflake in terms of cost per query per hour, across three examined cluster classes and 8-31% more cost-effective in terms of cost per query hour across the examined cluster classes for Redshift. Azure SQL Data Warehouse also ran the TPC-H queries between 12 and 17 times cheaper than Google BigQuery in terms of price per query per hour across all three SQL Data Warehouse clusters.

Price and performance are critical points of interest when it comes to selecting an analytics platform, because they ultimately impact total cost of ownership, value, and user satisfaction. Our analysis reveals Azure SQL Data Warehouse to be very powerful and comparative in value.

## 6 About William McKnight



An Ernst & Young Entrepreneur of the Year Finalist and frequent best practices judge, William is a former Fortune 50 technology executive and database engineer. He provides Enterprise clients with action plans, architectures, strategies, and technology tool selection to manage information.

William McKnight is an Analyst for Gigaom Research who takes corporate information and turns it into a bottom-line producing asset. He's worked with companies like Dong Energy, France Telecom, Pfizer, Samba Bank, ScotiaBank, Teva Pharmaceuticals and Verizon — Many of the Global 2000 — and many others. William focuses on delivering business value and solving business problems utilizing proven, streamlined approaches in information management.

He is a frequent international keynote speaker and trainer. William has taught at Santa Clara University, UC-Berkeley and UC-Santa Cruz.

## 7 About Jake Dolezal



As Analytics Practice Lead at McKnight Consulting Group, Jake Dolezal has two decades of experience in the Information Management field with expertise in analytics, data warehousing, master data management, data governance, business intelligence, statistics, data modeling and integration, and visualization. Jake has experience across a broad array of industries, including: healthcare, education, government, manufacturing, engineering, hospitality, and restaurants. He has a doctorate in information management from Syracuse University.

## 8 About Microsoft

Microsoft offers Azure SQL Data Warehouse...

Learn more at <https://azure.microsoft.com/en-us/services/sql-data-warehouse/>.

© [Knowingly, Inc.](#) 2019. "*Data Warehouse in the Cloud Benchmark*" is a trademark of [Knowingly, Inc.](#). For permission to reproduce this report, please contact [sales@gigaom.com](mailto:sales@gigaom.com).