















# Table of Contents

<b>TPC BENCHMARK H OVERVIEW</b> .....	<b>10</b>
<b>0 GENERAL ITEMS</b> .....	<b>11</b>
0.1 BENCHMARK SPONSOR.....	11
0.2 PARAMETER SETTINGS.....	11
0.3 CONFIGURATION DIAGRAM.....	11
<b>1 CLAUSE 1: LOGICAL DATABASE DESIGN RELATED ITEMS</b> .....	<b>13</b>
1.1 DATABASE DEFINITION STATEMENTS.....	13
1.2 PHYSICAL ORGANIZATION.....	13
1.3 HORIZONTAL PARTITIONING.....	13
1.4 REPLICATION.....	13
<b>2 CLAUSE 2: QUERIES AND REFRESH FUNCTIONS</b> .....	<b>14</b>
2.1 QUERY LANGUAGE.....	14
2.2 VERIFYING METHOD FOR RANDOM NUMBER GENERATION.....	14
2.3 GENERATING VALUES FOR SUBSTITUTION PARAMETERS.....	14
2.4 QUERY TEXT AND OUTPUT DATA FROM QUALIFICATION DATABASE.....	14
2.5 QUERY SUBSTITUTION PARAMETERS AND SEEDS USED.....	14
2.6 ISOLATION LEVEL.....	14
2.7 SOURCE CODE OF REFRESH FUNCTIONS.....	14
<b>3 CLAUSE 3: DATABASE SYSTEM PROPERTIES</b> .....	<b>15</b>
3.1 ACID PROPERTIES.....	15
3.2 ATOMICITY REQUIREMENTS.....	15
3.3 CONSISTENCY REQUIREMENTS.....	15
3.4 ISOLATION REQUIREMENTS.....	16
3.5 DURABILITY REQUIREMENTS.....	17
<b>4 CLAUSE 4: SCALING AND DATABASE POPULATION</b> .....	<b>19</b>
4.1 ENDING CARDINALITY OF TABLES.....	19
4.2 DISTRIBUTION OF TABLES AND LOGS ACROSS MEDIA.....	19
4.3 MAPPING OF DATABASE PARTITIONS/REPLICATION.....	19
4.4 IMPLEMENTATION OF RAID.....	19
4.5 DBGEN MODIFICATIONS.....	20
4.6 DATABASE LOAD TIME.....	20
4.7 DATA STORAGE RATIO.....	20
4.8 DATABASE LOAD MECHANISM DETAILS AND ILLUSTRATION.....	20
4.9 QUALIFICATION DATABASE CONFIGURATION.....	20
4.10 MEMORY TO DATABASE SIZE PERCENTAGE.....	21
<b>5 CLAUSE 5: PERFORMANCE METRICS AND EXECUTION RULES RELATED ITEMS</b> .....	<b>22</b>
5.1 SYSTEM ACTIVITY BETWEEN LOAD AND PERFORMANCE TESTS.....	22
5.2 STEPS IN THE POWER TEST.....	22
5.3 TIMING INTERVAL FOR EACH QUERY AND REFRESH FUNCTIONS.....	22
5.4 NUMBER OF STREAMS FOR THE THROUGHPUT TEST.....	22
5.5 START AND END DATE/TIME OF EACH QUERY STREAM.....	22
5.6 TOTAL ELAPSED TIME OF THE MEASUREMENT INTERVAL.....	22
5.7 REFRESH FUNCTION START DATE/TIME AND FINISH DATE/TIME.....	22
5.8 PERFORMANCE METRICS.....	22
5.9 THE PERFORMANCE METRIC AND NUMERICAL QUANTITIES FROM BOTH RUNS.....	22
5.10 SYSTEM ACTIVITY BETWEEN PERFORMANCE TESTS.....	23
5.11 DOCUMENTATION TO SATISFY CLAUSE 5.2.7.....	23
5.12 QUERY OUTPUT VALIDATION.....	23
<b>6 CLAUSE 6: SUT AND DRIVER IMPLEMENTATION RELATED ITEMS</b> .....	<b>24</b>
6.1 DRIVER.....	24
6.2 IMPLEMENTATION SPECIFIC LAYER (ISL).....	24
6.3 PROFILE-DIRECTED OPTIMIZATION.....	24
<b>7 CLAUSE 7: PRICING</b> .....	<b>25</b>
7.1 HARDWARE AND SOFTWARE USED IN THE PRICED SYSTEM.....	25
7.2 TOTAL THREE YEAR PRICE.....	25
7.3 AVAILABILITY DATE.....	25
<b>8 CLAUSE 8: FULL DISCLOSURE</b> .....	<b>26</b>
8.1 SUPPORTING FILES INDEX TABLE.....	26
<b>9 CLAUSE 9: AUDIT RELATED ITEMS</b> .....	<b>27</b>



9.1 AUDITOR'S REPORT.....	27
<b>APPENDIX A: PRICING INFORMATION .....</b>	<b>28</b>

## TPC Benchmark H Overview

The TPC Benchmark™ H (TPC-H) is a Decision Support benchmark. It is a suite of business-oriented adhoc queries and concurrent modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

- Examine large volumes of data
- Execute queries with a high degree of complexity
- Give answers to critical business questions

TPC-H evaluates the performance of various Decision Support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions
- Simulate generated ad-hoc queries
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- Are executed against a database complying to specific population and scaling requirements
- Are implemented with constraints derived from staying closely synchronized with an on-line production database

## 0 General Items

### 0.1 Benchmark Sponsor

*A statement identifying the benchmark sponsor(s) and other participating companies must be provided.*

This TPC-H benchmark is sponsored by Dell Inc. The benchmark implementation was developed and engineered by Exasol AG.

### 0.2 Parameter Settings

*Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:*

- Database Tuning Options
- Optimizer/Query execution options
- Query processing tool/language configuration parameters
- Recovery/commit options
- Consistency/locking options
- Operating system and configuration parameters
- Configuration parameters and options for any other software component incorporated into the pricing structure
- Compiler optimization options

*This requirement can be satisfied by providing a full list of all parameters and options, as long as all those which have been modified from their default values have been clearly identified and these parameters and options are only set once.*

The Supporting Files Archive contains the database and OS parameters used in this benchmark. In addition the Performance Tuning Guidelines for Mellanox Network Adapters ([link](#)) were followed to optimize the network performance.

### 0.3 Configuration Diagram

*Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:*

- Number and type of processors.
- Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.
- Number and type of disk units (and controllers, if applicable).
- Number of channels or bus connections to disk units, including their protocol type.
- Number of LAN (e.g. Ethernet) Connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.
- Type and the run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.).

The System Under Test (SUT), depicted in Figure 1.1, that was used to obtain the results in this benchmark consists of the following components:

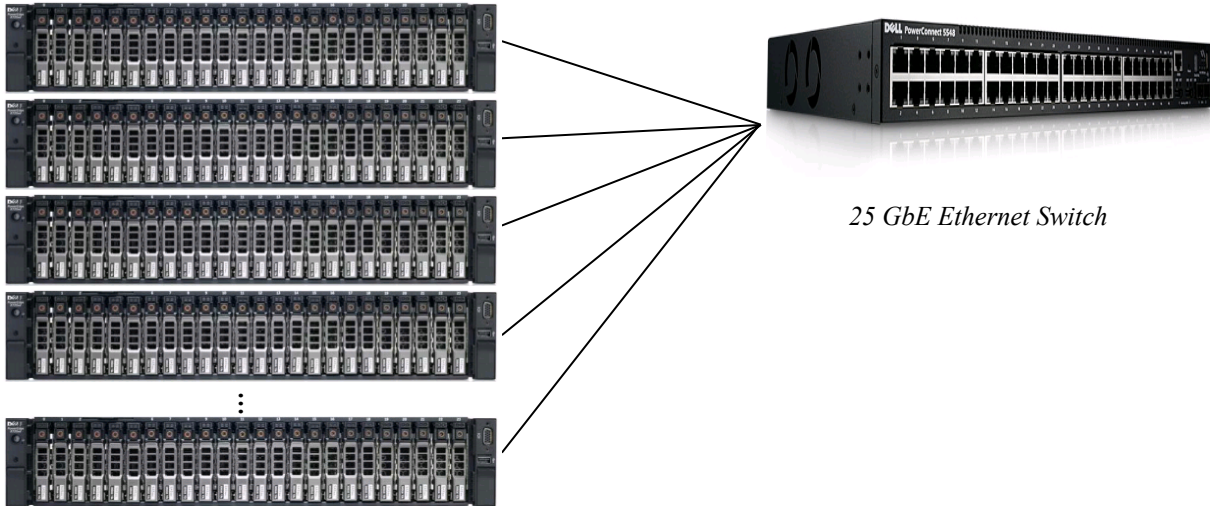
System components:	
20 x	Dell PowerEdge R6415 servers, each with:
	256 GB RAM
	1 x AMD EPYC™ 7551P 2.00GHz/2.55GHz
	2 x 480 GB SSD SATA
	1 x PERC H740P RAID Controller, 8GB NV Cache, Mini card
	1 x Mellanox ConnectX-4 Lx Dual Port 25GbE SFP28
1 x	1600 GB Dell NVMe
1 x	Dell EMC S5048F-ON Switch, 48x 25GbE, 6x 100GbE QSFP28

Network is 1 x 25GbE Ethernet. Each server contains 2 physical disks configured as 1 mirrored pair at the controller level (HW RAID 1). One server contains 1 additional NVMe drive used to store the flat files during the benchmark.

Priced configuration and benchmarked configuration are identical.

**DBMS-Cluster**

*Rack Servers*



*25 GbE Ethernet Switch*

*Figure 1.1: Benchmarked and priced system configuration*

# 1 Clause 1: Logical Database Design Related Items

## 1.1 Database Definition Statements

*Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases.*

The Supporting Files Archive contains the build scripts that define the tables and indices for the TPC-H database.

## 1.2 Physical Organization

*The physical organization of tables and indices, within the test and qualification databases, must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.*

Physical organization requires no user input. All the database data is placed on the same partition.

## 1.3 Horizontal Partitioning

*Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed.*

Horizontal partitioning is used. The data is automatically distributed on the cluster nodes using a hash algorithm. The columns used for the hashing are controlled by DDL statements (see Supporting Files Archive).

## 1.4 Replication

*Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.7.*

No replication was used.

## 2 Clause 2: Queries and Refresh Functions

### 2.1 Query Language

*The query language used to implement the queries must be identified.*

SQL was the query language uniquely used throughout this benchmark.

### 2.2 Verifying Method for Random Number Generation

*The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.*

TPC supplied versions 2.18.0 of DBGEN and QGEN were used in this benchmark.

### 2.3 Generating Values for Substitution Parameters

*The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number, and patch level of QGEN must be disclosed.*

QGEN version 2.18.0 was used to generate the substitution parameters.

### 2.4 Query Text and Output Data from Qualification Database

*The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definition or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.*

The Supporting Files Archive contains the qualification query text and query output. The standard queries were used throughout with the following modifications:

- Quoting of reserved keyword „value“ (Q11)
- LIMIT syntax used to restrict the number of output rows (Q2,Q3,Q10,Q18,Q21)
- Naming of columns of sub-select in Q13
- Used approved variant A of Q15 (Appendix B): ‘with clause’ instead of “create view/drop view”

### 2.5 Query Substitution Parameters and Seeds Used

*The query substitution parameters used for all performance tests must be disclosed in tabular format, along with the seeds used to generate these parameters.*

The Supporting Files Archive contains the seed and query substitution parameters.

### 2.6 Isolation Level

*The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to the levels defined in Clause 3.4, additional descriptive detail must be provided.*

The queries and transactions were run with the isolation level 3.

### 2.7 Source Code of Refresh Functions

*The details of how the refresh functions were implemented must be disclosed (including source code of any non-commercial program used).*

The Supporting Files Archive contains the source code of the refresh functions.

## 3 Clause 3: Database System Properties

### 3.1 ACID Properties

*The ACID (Atomicity, Consistency, Isolation, and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.*

All ACID tests were conducted according to specification. The Supporting Files Archive contains the source code of the ACID test scripts.

### 3.2 Atomicity Requirements

*The system under test must guarantee that transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially completed operations leave any effects on the data.*

#### 3.2.1 Atomicity of the Completed Transactions

*Perform the ACID Transaction for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDERS, LINEITEM, and HISTORY tables.*

The following steps were performed to verify the atomicity of the completed ACID transactions:

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. One ACID Transaction was performed using the order key from step 1.
3. The ACID Transaction was committed.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key.
5. It was verified that the appropriate rows had been changed.

#### 3.2.2 Atomicity of Aborted Transactions

*Perform the ACID Transaction for a randomly selected set of input data, substituting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDERS, LINEITEM, and HISTORY tables.*

The following steps were performed to verify the atomicity of the completed ACID transactions:

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. One ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was rolled back.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key.
5. It was verified that the appropriate rows had not been changed.

### 3.3 Consistency Requirements

*Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another. A consistent state for the TPC-H database is defined to exist when:*

$$O\_TOTALPRICE = SUM(trunc(trunc(L\_EXTENDEDPRICE*(1-L\_DISCOUNT),2)*(1+L\_TAX),2))$$

*For each ORDER and LINEITEM defined by (O\_ORDERKEY = L\_ORDERKEY).*

#### 3.3.1 Consistency Test

*Verify that ORDERS and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDERS and LINEITEM.*

The following queries were executed before and after the durability tests to show that the database was always in a consistent state both initially and after submitting transactions:

```

SELECT *
FROM (
  SELECT o_orderkey, o_totalprice - sum(trunc(trunc(l_extendedprice *
    (1-l_discount),2)*(1+l_tax),2)) part_res
  FROM orders, lineitem
  WHERE o_orderkey=l_orderkey
  GROUP BY o_orderkey, o_totalprice
) WHERE not part_res=0;

```

The following steps were performed to verify the consistency of ACID transactions:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. 100 transactions for each of the 14 execution streams were prepared.
3. The 100 ACID transactions per stream were executed from 14 execution streams.
4. The consistency of the ORDERS and LINEITEM tables was re-verified.

### 3.4 Isolation Requirements

*Operations of concurrent transactions must yield results, which are indistinguishable from the results, which would be obtained by forcing each transaction to be serially executed to completion in some order.*

The steps of the isolation tests were adapted to the Exasol isolation environment.

#### 3.4.1 Isolation Test 1 – Read-Write Conflict with Commit

*Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed*

The following steps were performed to satisfy the test of isolation for a read-only and a read-write committed transaction:

1. Start a query and verify that the row was retrieved.
2. Start an update transaction, read and update the same row. Wait before commit.
3. Start the same query and verify that the row retrieved has not changed.
4. Commit the update transaction
5. Start the same query and verify that the new row is retrieved

#### 3.4.2 Isolation Test 2 – Read-Write Conflict with Rollback

*Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is rolled back.*

The following steps were performed to satisfy the test of isolation for a read-only and a rolled back read-write transaction:

1. Start a query and verify that the row was retrieved.
2. Start an update transaction, read and update the same row. Wait before commit.
3. Start the same query and verify that the row retrieved has not changed.
4. Rollback the update transaction
5. Start the same query and verify that the old row (step 1) is retrieved

#### 3.4.3 Isolation Test 3 – Write-Write Conflict with Commit

*Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.*

The following steps were performed to verify isolation of two update transactions:

1. Start a query and verify that the row was retrieved.
2. Start an update transaction, read and update the same row. Wait before commit.
3. Start another update transaction, read and try to update the same row and verify that the transaction is forced to rollback.
4. commit the update transaction
5. Start the same query and verify that the new row is retrieved

#### 3.4.4 Isolation Test 4 – Write-Write Conflict with Rollback

*Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.*

The following steps were performed to verify isolation of two update transactions after the first one is rolled back:



1. Start a query and verify that the row was retrieved.
2. Start an update transaction, read and update the same row. Wait before commit.
3. Start another update transaction, read and try to update the same row and verify that the transaction is forced to rollback.
4. Rollback the update transaction
5. Start the same query and verify that the old row (step 1) is retrieved.

### 3.4.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables

*Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.*

The following steps were performed to demonstrate the ability of read and write transactions affecting different tables to make progress concurrently:

1. Start a query and verify that the row was retrieved.
2. Start an update transaction, read and update the same row. Wait before commit.
3. Start a second transaction that does the following:  
Select random values of PS\_PARTKEY and PS\_SUPPKEY. Return all columns of the PARTSUPP table for which PS\_PARTKEY and PS\_SUPPKEY are equal to the selected values.
4. Verify that the read transaction completes.
5. Commit the update transaction.
6. Start the same query and verify that the new row is retrieved.

### 3.4.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream

*Demonstrate the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.*

The following query was used to ensure sufficient execution time to perform the test:

```
SELECT l1.l_quantity,
       SUM(l2.l_extendedprice),
       SUM(l3.l_extendedprice),
       SUM(l3.l_quantity)
FROM   lineitem l1, lineitem l2, lineitem l3, lineitem l4, lineitem l5
WHERE  l1.l_shipdate <= DATE '1998-12-01' -0
       AND l1.l_orderkey = l2.l_orderkey
       AND l1.l_linenumber = l2.l_linenumber
       AND l1.l_extendedprice = l3.l_extendedprice
       AND l3.l_quantity < 30
       AND l4.l_quantity = l1.l_quantity
       AND l4.l_orderkey < 150
       AND l5.l_receiptdate = l1.l_receiptdate
       AND l5.l_partkey < 140
GROUP BY l1.l_quantity;
COMMIT;
```

1. A Transaction, T1, which executed the above query against the qualification database, was started using a randomly selected DELTA.
2. An ACID Transaction, T2, was started for a randomly selected O\_KEY, L\_KEY and DELTA.
3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.
4. T1 was still executing.
5. Transaction T1 completed executing the query.

## 3.5 Durability Requirements

*The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3.*

Exasol has serializable isolation level with table level lock concurrency control. The ACID Transaction of stream0 was expanded with 5 seconds delay after the update and before commit after it committed 100 transactions. Since only one update transaction can execute at any one time, the delay should guarantee that the active update transaction is "in-flight" at the time of the failure.

The following steps were performed for the durability test:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. 400 transactions for each of the 14 execution streams were prepared.
3. After that at least 100 ACID transactions were submitted from each of the 14 execution streams.
4. A durability failure was induced (see details for each failure below).
5. After restoration of the system the consistency of the ORDERS and LINEITEM tables was re-verified.
6. The durability success files and the HISTORY table were compared.

All durability tests were performed on an 4-node cluster.

### **3.5.1 Permanent Unrecoverable Failure of Any Durable Medium**

*Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-H database tables or recovery log tables.*

Disk, node and controller failure tests were combined as explained in section 3.5.5.

### **3.5.2 System Crash**

*Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.*

The system crash, memory failure and loss of external power tests were combined as explained in section 3.5.4.

### **3.5.3 Memory Failure**

*Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents). See the previous section.*

The system crash, memory failure and loss of external power tests were combined as explained in section 3.5.4.

### **3.5.4 Loss of External Power**

*Loss of External Power: Guarantee the database and the effects of committed updates are preserved during the loss of all external power to the SUT for an indefinite time period.*

System crash was performed by turning off the power of 4 node cluster during the durability test. When power was restored, the system rebooted automatically and the database was restarted manually.

In addition single node crash was performed like described in section 3.5.5

### **3.5.5 Node or Controller Failure**

*Guarantee the database and committed updates are preserved across failure of the controller or the whole node.*

Cluster of 4 online nodes and 1 offline spare node were used for this test.

This test was executed in following order:

1. One of the online nodes was crashed by removing the power – the system stopped to work.
2. The offline spare node was turned on and manually configured as replacement of the failed node.
3. On the remaining 3 + 1 replacement nodes the database was restarted manually.

## 4 Clause 4: Scaling and Database Population

### 4.1 Ending Cardinality of Tables

The cardinality (e.g., the number of rows) of each table of the test database, as it existed at the completion of the database load (see clause 4.2.5) must be disclosed.

The following table lists the TPC Benchmark H defined tables and the row count for each table as they existed upon completion of the build.

Table	Rows
Lineitem	59,999,994,267
Order	15,000,000,000
Partsupp	8,000,000,000
Part	2,000,000,000
Customer	1,500,000,000
Supplier	100,000,000
Nation	25
Region	5

### 4.2 Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described for the tested and priced systems.

Each server contains 2 physical disks configured as 1 mirrored pairs at the controller level (HW RAID 1). The resulting 1 devices (1 x 480GB) are divided into 4 partitions as shown in the table below. All benchmark- and database-relevant data is stored in partition *centos-root* (Tables, Indexes, Temp). One additional 1.6TB NVMe drive was used on the driver node to store the dbgen RF1 & RF2 flat files.

Partition Name	Type	Partition Size / Devices	Content
sda1	vfat	200 MB of 480 GB	/boot/efi
sda2	xfs	1 GB of 480 GB	/boot
centos-root	xfs	441 GB of 480 GB	OS & DBMS data
centos-swap	swap	4 GB of 480 GB	OS swap
* nvme0n1	ext3	1600 GB of 1600 GB	DBGEN RF1 & RF2 flat files
* present only on the driver node			

### 4.3 Mapping of Database Partitions/Replication

The mapping of database partitions/replications must be explicitly described.

Horizontal partitioning is used. The data is automatically distributed on the cluster nodes using a hash algorithm. The columns used for the hashing are controlled by DDL statements (see Supporting Files Archive).

The data is mirrored across the cluster nodes to achieve redundancy for the purpose of recovery only.

### 4.4 Implementation of RAID

Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID must be disclosed for each device.

Please refer to chapter 4.2.

#### 4.5 DBGEN Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN version 2.18.0 was used, no modifications were made.

#### 4.6 Database Load Time

The database load time for the test database (see Clause 4.3) must be disclosed.

See Numerical Quantities Summary in the Executive Summary.

#### 4.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up. For the reporting of configured disk capacity, gigabyte (GB) is defined to be  $2^{30}$  bytes.

Disk Type	GB per disk*	GiB per disk**	# of disks	Total (GiB)**
Internal	480 GB	447 GiB	2 * 20	17,881
Internal	1,600 GB	1,490 GiB	1	1,490
<b>Total Storage</b>				19,372
<b>Data Storage Ratio</b>				<b>1.94</b>

\* Disk manufacturer definition of 1 GB is  $10^9$  bytes

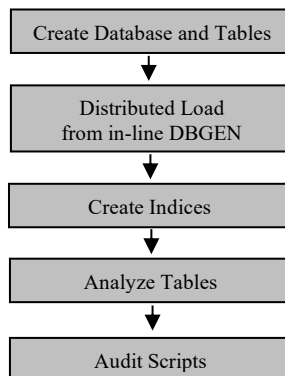
\*\* In this calculation 1 GiB is defined as  $2^{30}$  bytes

#### 4.8 Database Load Mechanism Details and Illustration

The details of the database load must be disclosed, including a block diagram illustrating the overall process. Disclosure of the load procedure includes all steps, scripts, input and configuration files required to completely reproduce the test and qualification databases.

The database was loaded using data generation stored on the flat files all on the tested and priced configuration. DBGEN was used to create the flat files.

The following block diagram describes the process used to load the database.



#### 4.9 Qualification Database Configuration

Any differences between the configuration of the qualification database and the test database must be disclosed.

The qualification database used identical scripts to create and load the data with changes to adjust for the database scale factor.

#### 4.10 Memory to Database Size Percentage

*The memory to database size percentage must be disclosed. It is computed by multiplying by 100 the total memory size priced on the SUT (see clause 6.2.1) and dividing this number by the size chosen for the test database as defined in Clause 4.1.3.1.*

<b>Nodes</b>	<b>RAM per Node</b>	<b>Total Memory</b>	<b>Scale Factor</b>	<b>Memory to Database Size Ratio</b>
20	256 GB	5,120 GB	10,000	51.2%

## **5 Clause 5: Performance Metrics and Execution Rules Related Items**

### **5.1 System Activity between Load and Performance Tests**

*Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed.*

There is no activity on the SUT between the conclusion of the load test and the beginning of the performance test.

### **5.2 Steps in the Power Test**

*The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be disclosed.*

The following steps were used to implement the power test:

1. RF1 refresh function from update stream
2. Stream 0 execution from query stream
3. RF2 refresh function from same update stream

### **5.3 Timing Interval for Each Query and Refresh Functions**

*The timing intervals (see Clause 5.3.7) for each query of the measured set for both refresh functions must be reported for the power test.*

See Numerical Quantities Summary in the Executive Summary.

### **5.4 Number of Streams for the Throughput Test**

*The number of execution streams used for the throughput test must be disclosed.*

One stream was used for the refresh pairs. The number of query streams used is listed in the Numerical Quantities Summary in the Executive Summary.

### **5.5 Start and End Date/Time of Each Query Stream**

*The start time and finish time for each query stream must be reported for the throughput test.*

See Numerical Quantities Summary in the Executive Summary.

### **5.6 Total Elapsed Time of the Measurement Interval**

*The total elapsed time of the measurement interval (see Clause 5.3.6) must be reported for the throughput test.*

See Numerical Quantities Summary in the Executive Summary.

### **5.7 Refresh Function Start Date/Time and Finish Date/Time**

*Start and finish time for each update function in the update stream must be reported for the throughput test.*

See Numerical Quantities Summary in the Executive Summary.

### **5.8 Performance Metrics**

*The computed performance metric, related numerical quantities and price performance metric must be reported.*

See Numerical Quantities Summary in the Executive Summary.

### **5.9 The Performance Metric and Numerical Quantities from Both Runs**

*A description of the method used to determine the reproducibility of the measurement results must be reported. This must include the performance metrics ( $Q_{ppH}$  and  $Q_{thH}$ ) from reproducibility runs.*

	Run 1	Run 2
<b>QphH@10000GB</b>	8,667,578.0	8,774,215.3
<b>QppH@10000GB</b>	8,930,331.9	9,088,968.7
<b>QthH@10000GB</b>	8,412,555.1	8,470,361.8

### 5.10 System Activity between Performance Tests

*Any activity on the SUT that takes place between the conclusion of Run 1 and the beginning of Run 2 must be disclosed.*

There was no system activity between Run 1 and Run 2.

### 5.11 Documentation to satisfy Clause 5.2.7

*All documentation necessary to satisfy Clause 5.2.7 must be made available upon request.*

Exasol documentation is publicly available at [Exasol.com](http://Exasol.com)

### 5.12 Query Output Validation

*The output of the Query Output Validation Test must reported in the supporting files archive.*

The Supporting Files Archive contains the output of the validation test.

## 6 Clause 6: SUT and Driver Implementation Related Items

### 6.1 Driver

*A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.*

All stream executions are performed by a script. QGEN is used to produce query text.

For each power-test run:

1. A shell script is started, executes RF1 and then waits for the query stream to complete.
2. A shell script is started, executes the 22 queries in the required order for stream 0 and then signals to the shell script started in step 1.
3. The shell script started in step 1 is released and executes RF2.

For each throughput-test run:

1. The queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4 from the driver in several streams (the number of streams is listed in the Numerical Quantities).
2. In parallel with the queries, pairs of RF1/RF2 are executed sequentially in one update stream.

The source code of the used scripts are disclosed in the Supporting Files Archive.

### 6.2 Implementation Specific Layer (ISL)

*If an implementation specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.*

The scripts used to implement the ISL are disclosed in the Supporting Files Archive.

### 6.3 Profile-Directed Optimization

*If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed.*

Profile-directed optimization was not used.



## 7 Clause 7: Pricing

### 7.1 Hardware and Software Used in the Priced System

*A detailed list of hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.*

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the executive summary. All prices are currently effective. Third-party price quotations are included in Appendix A.

### 7.2 Total Three Year Price

*The total 3-year price of the entire configuration must be reported including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.*

A detailed pricing sheet of all the hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the executive summary at the beginning of this document.

Exasol raw data subscription pricing. See Appendix A.

### 7.3 Availability Date

*The committed delivery date for general availability of products used in the priced calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided.*

Component	Availability Date
Cluster Hardware	Now (date of publication)
Exasol 6.2	Now (date of publication)

## 8 Clause 8: Full Disclosure

### 8.1 Supporting Files Index Table

An index for all files and/or directories included in the Supporting Files Archive as required by Clauses 8.3.2 through 8.3.8 must be provided in the report.

Clause	Description	Archive Files	Pathname
1	Parameter Settings	benchmark_scripts.zip	RUN/params.log
	OS Tunable Parameters		RUN/rc.local
	DB Creation Scripts		KIT/sql/create_user.sql KIT/sql/create_schema.sql KIT/sql/create_indices.sql KIT/sql/analyse_database.sql
	System Verification		KIT/scripts/tools/hwinfo.sh
	Toolkit Common Scripts		KIT/scripts
2	Minor query modifications	benchmark_scripts.zip	KIT/tpch_archives/tpch_2_18_0.zip.patch
3	ACID Test Scripts	benchmark_scripts.zip	KIT/ACID/
	ACID Test Results		ACID/
4	Database Load Scripts	benchmark_scripts.zip	KIT/scripts/load_init.sh
	Qualification Test Results		VLD/
5	Query Output Results	run1results.zip	RUN//run1/
		run2results.zip	RUN//run2/
6	Source Codes and Scripts of Driver	benchmark_scripts.zip	KIT/scripts/query_streams
7	There are no files to be included for Clause 7.	N/A	N/A
8	Query Parameters & Seeds	run1results.zip	RUN/run1/substitution_parameters.txt
	Executable Query Text		RUN/run1/stream*.sql
	RF function source code		KIT/scripts/tpc_h_run_full.sh

## 9 Clause 9: Audit Related Items

### 9.1 Auditor's Report

*The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.*

This implementation of the TPC Benchmark H was audited by Francois Raab of InfoSizing, a certified TPC-H auditor. Further information regarding the audit process may be obtained from:

**Francois Raab**  
InfoSizing  
20 Kreg Ln.  
Manitou Springs, CO 80829  
Phone: (719) 473-7555  
Email: [francois@sizing.com](mailto:francois@sizing.com)

TPC Benchmark H Full Disclosure Report and other information can be downloaded from the Transaction Processing Performance Council website at [www.tpc.org](http://www.tpc.org)

## Appendix A: Pricing Information

Exasol Inc  
c/o WeWork  
1372 Peachtree Street  
Atlanta, GA, 30309  
United States



**Dell, Inc.**  
One Dell Way  
Round Rock, TX 78682  
United States

Product	Part Number	Unit Price \$	Qty	Ext. Price \$
Exasol 6.2	Exasol 10TB Subscription <sup>1)</sup>	\$15,043	36	\$541,548
<b>Total</b>				<b>\$541,548</b>

<sup>1)</sup> Includes software (DB & OS) support 24x7

This quote is valid until December 15, 2022.

**Exasol Pricing Contact:**

Gunther Schweer  
[sales@Exasol.com](mailto:sales@Exasol.com)