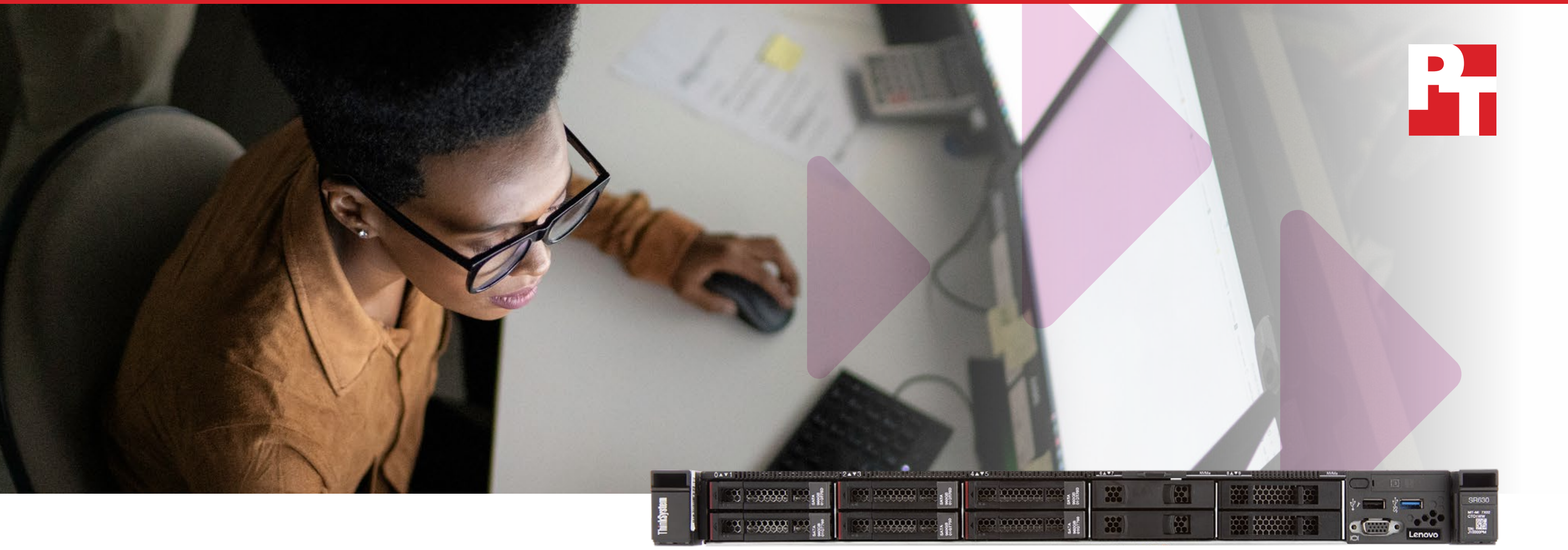
A Principled Technologies report: Hands-on testing. Real-world results.



# Keep your data safe by moving from unsupported SQL Server 2008 to SQL Server 2019 on Lenovo ThinkSystem SR630 servers, powered by Intel Xeon Scalable processors

## We found that the migration process was easy and performance improved on the new platform

Microsoft ended extended support for SQL Server 2008 and 2008 R2 earlier this year. They will provide no more technical assistance and no more security updates. Data is often a company's most valuable resource. If your business has been putting off migrating from one of these versions, your data could be exposed to future exploits or vulnerabilities. To protect your data, you must move to new versions.

Happily, the process can be simple. In the Principled Technologies data center, we migrated a database from a legacy server running these obsolete versions to a Lenovo® ThinkSystem™ SR630 running Windows Server 2019 and SQL Server 2019. Using the Microsoft Data Migration Assistant, the process required less than five minutes of hands-on time. We also measured online analytical processing (OLAP) performance on both platforms and saw it improve dramatically after migration.

**Avoid the worries**
of using unsupported software

**Migrate with little hands-on time**
Less than 5 minutes and only 19 steps

**Improve performance**
34x times greater online analytical processing

## About the Lenovo ThinkSystem SR630

This offering from Lenovo is a dual-socket, 1U rack server powered by two Intel Xeon Gold 6130 processors. It contains 24 DIMM slots that support 768 GB of DDR4 RDIMMs and up to 3 TB of three-dimensional stacked RDIMMs. The SR630 also comes equipped with the latest XClarity Controller chip, which provides "service-processor control, monitoring, and alerting functions."[1] Learn more at https://lnv.gy/34IyBbr.

## About Intel Xeon Scalable processors

The Intel Xeon Scalable processor series includes many upgrades over previous-generation Intel processors. They have as many as 28 cores (which can enable greater performance and scalability), six memory channels, and support up to 1.5 TB of memory per socket.

The series currently includes two generations of four feature configurations: Platinum, Gold, Silver, and Bronze. The Lenovo ThinkSystem SR630 we tested was powered by two first-generation Intel Xeon Gold 6130 processors.

Learn more about Intel Xeon Scalable processors at https://intel.ly/2Z6oBYe.

# Migrating is necessary

For IT admins of small and medium-sized businesses, whose daily to-do lists typically consist of urgent items, it's easy to put off the important job of updating their server applications and operating systems. Microsoft ended extended support for SQL Server 2008 and 2008 R2 on July 9, 2019. For Windows Server 2008 and 2008 R2, extended support ends on January 14, 2020. With the end of both technical assistance and regular security updates, moving your databases becomes both important and urgent.

## Other facts to consider

By continuing to use these outdated versions now that Microsoft no longer provides technical support, **companies risk incurring downtime that interrupts the flow of business** for employees and customers. They can also be forced to contract with third-party support providers.

A 2018 survey found that **nearly 60 percent of breaches were attributable to unpatched software**.[2]

Running **unsupported software makes businesses more susceptible to ransomware attacks**, in which criminals infiltrate a system through a vulnerability and hold the data hostage. According to a 2019 ProPublica report on firms that promise to retrieve such data, "they typically paid the ransom and charged the victims extra."[3]

According to a public statement from the U.S. Securities and Exchange commission, **small and medium businesses are easier targets for cyberattack** than larger organizations because they face the same threat landscape, but have far fewer resources."[4]

A 2017 study by Insureon found that **only 16 percent of small business owners believed they were at risk of a cyberattack**.[5]

An August 2019 Small Business Trends infographic estimates that **downtime can cost SMBs more than $10,000 per hour**.[6]

In a 2017 blog post, WCA Technologies reported that 40 percent of businesses close during unexpected outages, and **of businesses that close during a disaster, only 25 percent reopen**.[7]

In the face of all of this bad news, there is some good news: Migrating your SQL Server 2008 databases to SQL Server 2019 can help protect your business from the dangers of running outdated software—and our testing showed that it can be a simple process.

## What about the public cloud?

As support for SQL Server 2008 winds down, many businesses have elected to migrate databases to Microsoft Azure or other public cloud services. While the public cloud can offer many benefits, it can also introduce concerns, such as:

**Security and compliance**. Using an on-premises database solution lets your business retain control and oversight of the security strategies that protect your business and its customers. When you use a public cloud provider, you must rely on their policies. For companies in the medical and financial spheres, and others subject to regulatory requirements, this control is especially important.

**Connectivity and availability**. Public cloud computing systems are internet based, which means that the possibility of slowdowns and service outages always exists.

**Cost**. While the pay-as-you-go cloud pricing model lets you pay for only the compute resources you actually use, overall costs may not always be lower than with an on-premises solution. It is possible to end up replacing capital expenditures with hefty operational expenses.

## Migrating is easy

Deploying new hardware, installing a new version of SQL Server, and migrating your old database to new software can seem like a large and overwhelming task. However, we found that the Microsoft Data Migration Assistant did most of the work for us.

The Data Migration Assistant is a wizard-based tool that consists of an assessment and a migration. The assessment checks the migration for compatibility issues and new features. The migration backs up the database from the source to a server message block (SMB) file share, and then restores the database to the target from the file share.

The Data Migration Assistant has simple click-through screens that collect details such as server connection information, source and target SQL Server versions, and file share credentials. We found it very easy and straightforward to use. While you will need to plan for downtime during the migration, the process requires very few steps and minimal hands-on administrator time.

### About Microsoft Data Migration Assistant

Microsoft Data Migration Assistant is a free tool that facilitates your migration to an upgraded environment. Some features of Data Migration Assistant include detection of migration-blocking issues, partially or unsupported features, breaking, behavior changes, and more. To learn more about Data Migration Assistant version 5.0, see the Microsoft website: https://bit.ly/2S8KV1P.

**Less than 5 minutes**

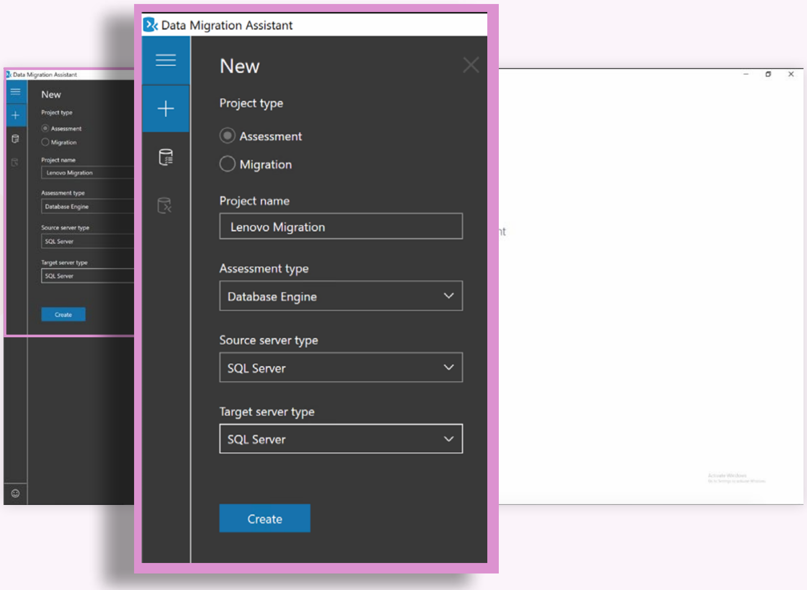
of hands-on time to complete the migration

In our testing, we migrated a 1TB SQL Server database. (See the System configuration information section for more details.) Completing the assessment took only 59 seconds, with the hands-on portion requiring 39 seconds and 12 steps. The hands-on portion of the migration required 7 steps and 1 minute 3 seconds. Once our admin had finished these tasks, the migration ran unattended, leaving them free to work on other activities. For our 1TB database, the unattended migration took 8 hours 28 minutes. Your time will vary depending on your network connection speed, the number of drives and RAID configuration you select, and the size of your database.

On the following page, we present a step-by-step migration guide based on our 1TB database migration.

## Running the pre-migration assessment (12 steps)

In this section, we explain the step-by-step process we followed for the pre-migration assessment in the Data Migration Assistant. Data Migration Assistant requires a dedicated user account for the migration process on the source and target servers if you are not using Active Directory. If you are using Active Directory, ensure that the source and target servers have the correct read and write permissions for the file share. Because we created our database using the HammerDB tool, which is configured to work with several SQL Server versions, we did not run into compatibility issues. If the Data Migration Assistant flags some issues during the assessment stage, use the suggestions it provides to resolve them. You can find examples of resolution suggestions on the Microsoft website. The screenshot below shows the first screen in the tool, where you choose the Assessment option, and the fields required to begin the process.



New

Project type

● Assessment

○ Migration

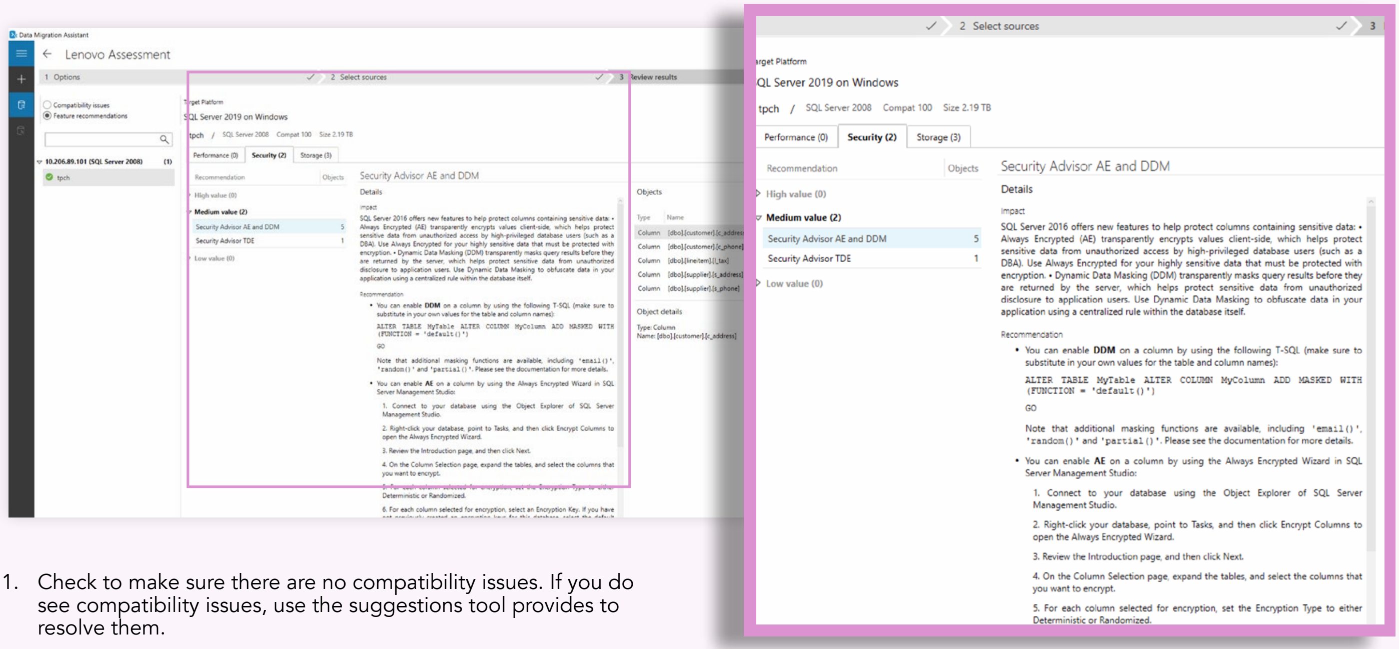
Project name: Lenovo Migration

Assessment type: Database Engine

Source server type: SQL Server

Target server type: SQL Server

1. Follow these steps to create a user account for migration on each server and grant them SQL permissions. This ensures there's a user with the proper SQL permissions on both ends of the migration to handle the database:
   a. Log into the source and target servers, and create a user account for migration on each using the Windows Add and Remove user tool.
   b. On the file share where the backup will be temporarily stored, edit the properties of the shared drive to grant read and write access.
   c. Click Start, and type `Services`
   d. Go to the MSSQLSERVER service, and right-click.
   e. Select Properties.
   f. Select the Log On tab.
   g. Select This Account, and change it to the user you just created.
   h. Enter the password, and click Apply.
   i. Restart the MSSQLSERVER service.
2. Download and install the Microsoft Data Migration Assistant from https://www.microsoft.com/en-us/download/details.aspx?id=53595.
3. Open Microsoft Data Migration Assistant. In the navigation pane on the left, click the plus sign to start a new project, fill out the fields below, and click Create.
   a. Select Assessment, and enter a project name.
   b. From the Assessment Type drop-down menu, select Database Engine.
   c. From the Source Server Type drop-down menu, select SQL Server.
   d. From the Target Server Type drop-down menu, select SQL Server.
4. When the project creation finishes, go to the Options screen, and from the Target Version drop-down menu, select SQL Server 2019 on Windows.
5. Check the boxes beside Compatibility Issues and New Features Recommendation, and click Next.
6. On the Select Sources screen, enter the IP address or fully qualified domain name (FQDN) of the source server.
7. From the drop-down menu, select an authentication type, and, if necessary, enter the username and password credentials below.
8. Check the boxes beside Encrypt connection and Trust server certificate, and click Connect.
9. In the Add Sources sidebar, select the server and database(s) you wish to access, and click Add.
10. Click Start Assessment. When the assessment finishes, the Data Migration Assistant will display any compatibility issues or new features you should know about. The following screenshot shows an example of the output you might see.



Data Migration Assistant

Lenovo Assessment

1 Options ✓ 2 Select sources ✓ 3 Review results

○ Compatibility issues
● Feature recommendations

10.206.89.101 (SQL Server 2008) (1)
tpch

Target Platform
SQL Server 2019 on Windows

tpch / SQL Server 2008 Compat 100 Size 2.19 TB

Performance (0) | **Security (2)** | Storage (3)

| Recommendation | Objects |
|---|---|
| High value (0) | |
| **Medium value (2)** | |
| Security Advisor AE and DDM | 5 |
| Security Advisor TDE | 1 |
| Low value (0) | |

Security Advisor AE and DDM

Details

Impact

SQL Server 2016 offers new features to help protect columns containing sensitive data: • Always Encrypted (AE) transparently encrypts values client-side, which helps protect sensitive data from unauthorized access by high-privileged database users (such as a DBA). Use Always Encrypted for your highly sensitive data that must be protected with encryption. • Dynamic Data Masking (DDM) transparently masks query results before they are returned by the server, which helps protect sensitive data from unauthorized disclosure to application users. Use Dynamic Data Masking to obfuscate data in your application using a centralized rule within the database itself.

Recommendation

- You can enable **DDM** on a column by using the following T-SQL (make sure to substitute in your own values for the table and column names):

  ```
  ALTER TABLE MyTable ALTER COLUMN MyColumn ADD MASKED WITH
  (FUNCTION = 'default()')
  GO
  ```

  Note that additional masking functions are available, including 'email()', 'random()' and 'partial()'. Please see the documentation for more details.

- You can enable **AE** on a column by using the Always Encrypted Wizard in SQL Server Management Studio:

  1. Connect to your database using the Object Explorer of SQL Server Management Studio.
  2. Right-click your database, point to Tasks, and then click Encrypt Columns to open the Always Encrypted Wizard.
  3. Review the Introduction page, and then click Next.
  4. On the Column Selection page, expand the tables, and select the columns that you want to encrypt.
  5. For each column selected for encryption, set the Encryption Type to either Deterministic or Randomized.

Objects

| Type | Name |
|---|---|
| Column | [dbo].[customer].[c_address] |
| Column | [dbo].[customer].[c_phone] |
| Column | [dbo].[lineitem].[l_tax] |
| Column | [dbo].[supplier].[s_address] |
| Column | [dbo].[supplier].[s_phone] |

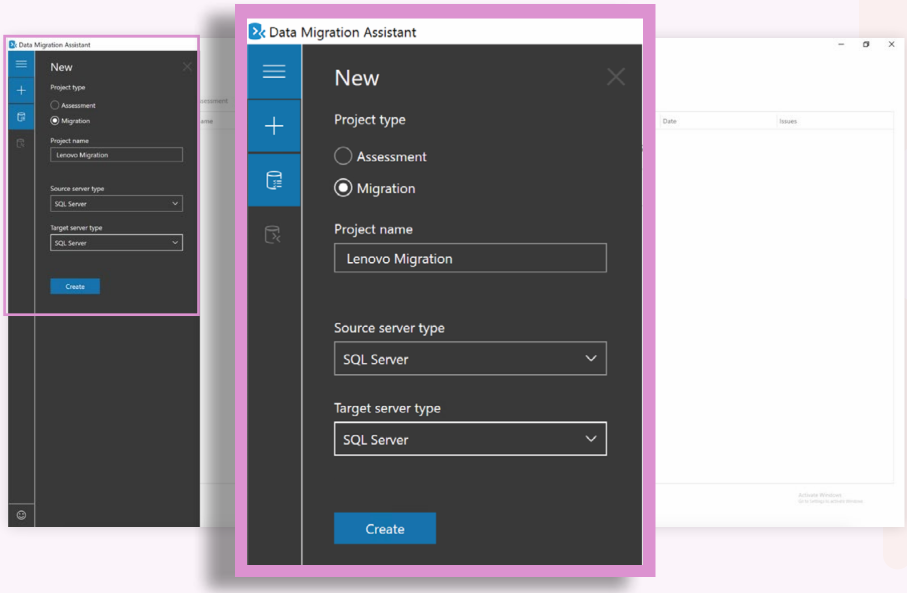
Object details

Type: Column
Name: [dbo].[customer].[c_address]

11. Check to make sure there are no compatibility issues. If you do see compatibility issues, use the suggestions tool provides to resolve them.
12. To save a copy of any feature recommendations post migration, click Export report.

## Running the migration (7 steps)

In this section, we provide the steps we took to migrate our 1TB database. As we mentioned previously, the time this migration takes will depend on your environment. However, the steps should remain the same. The following screenshot shows the Data Migration Assistant screen that starts the migration process.



Data Migration Assistant

New

Project type
○ Assessment
● Migration

Project name
Lenovo Migration

Source server type
SQL Server

Target server type
SQL Server

**Only 19 steps** to complete the assessment and migration

1. Open the Data Migration Assistant. In the navigation pane on the left, click the plus sign to start a new project, fill out the fields below, and click Create.
   a. Select Migration, and enter a project name.
   b. From the Source Server Type drop-down menu, select SQL Server.
   c. From the Target Server Type drop-down menu, select SQL Server.
2. On the Specify source & target screen, enter the IP address or fully qualified domain name (FQDN) of the source server and target servers.
3. From the drop-down menu, select an Authentication type, and, if necessary, enter the username and password credentials for both the source and target servers.
4. Check the boxes beside Encrypt connection and Trust server certificate for both the source and target servers.
5. On the Add Databases screen, enter the IP address or fully qualified domain name (FQDN) of the path to the fileshare.
6. Specify the location to restore the data and log files on the target server, and click Next.
7. On the Select Logins screen, check the box next to the logins you wish to migrate, and click Start Migration. When the Data Migration Assistant marks the migration complete, you can immediately start using your new database or start any modifications that you'd like to make to take advantage of new features.

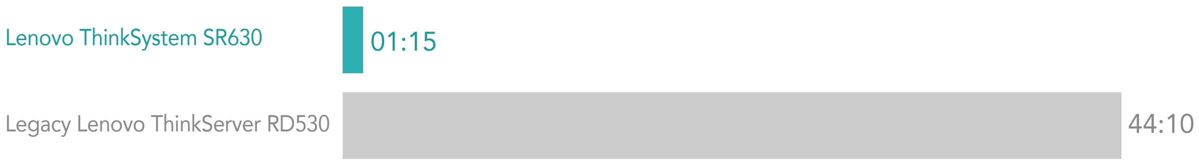
## Migrating improves performance

To compare the database performance of our legacy platform with what the new Lenovo ThinkSystem SR630 achieved, we ran an online transaction processing TPC-H-like workload both before and after we migrated our database. This type of workload measures query analytics in a data warehouse context, and our test query set included 22 complex queries. The chart below illustrates the dramatic improvement the new server running Windows Server 2019 and SQL Server 2019 delivered. Completing the workload took almost two full days on the legacy platform, whereas the new platform whipped through the queries in a little more than an hour. This means the new platform was more than 34 times faster than the legacy platform. Being able to enjoy this enormous performance boost—in addition to reducing the threats that result from trusting your data to unsupported software—is a very nice bonus.

**Save more than 42 hours**
when running a query set

Time for one user to complete 22 queries (hours:minutes)

Lower is better



One of the reasons performance improved so much with SQL Server 2019 is because it supports clustered columnstore indexing, a feature introduced in SQL Server 2014 that can greatly increase performance of warehouse and data analytics databases. To take advantage of it, you will need to drop and recreate some indexes in your database after the migration finishes. This process will require some planning and downtime, but the performance benefits that result from it should help make up for the lost time. In our example, recreating the indexes took approximately 48 hours (a mix of hands-on time and inactive time). Your time will vary depending on the size of your database and other variables in your environment. While this is a long time, remember that it's a one-time process that will drastically reduce the time it takes to query your OLAP database.

### About Windows Server 2019 and SQL Server 2019

The latest version of Windows Server offers a multitude of features and enhancements you can't get with the 2008 R2 version, including support for hyperconverged infrastructure (including unified management and protection in case of drive failures), enhanced security features (such as shielded VMs and disk encryption), and new innovations in application development, such as support for a variety of container platforms. For an exhaustive list and description of the features you'll gain when upgrading, see https://bit.ly/2Z7yBk4.

According to Microsoft, SQL Server 2019 "builds on previous releases to grow SQL Server as a platform that gives you choices of development languages, data types, on-premises or cloud environments, and operating systems."[8] It includes features and enhancements in the following areas: data virtualization and SQLServer 2019 Big Data Clusters, Intelligent Query Processing, In-Memory Database technologies, monitoring, developer experience, security, high availability, platform choice, and more. Learn more at https://bit.ly/35Ik2Wx.

## Conclusion

Many small and medium businesses delay updating their server operating systems and applications. When software reaches end-of-support and the vendor ceases to release security updates and patches, businesses that fail to migrate risk incurring downtime and expense. They may encounter technical problems, and their vital data becomes especially vulnerable to cyber attackers, who often target outdated software.

We migrated a SQL Server 2008 database on a legacy server running Windows Server 2008 R2 to a Lenovo ThinkSystem SR630 running SQL Server 2019 and Windows Server 2019. With the free Microsoft Data Migration Assistant tool, performing the assessment and migration required only 19 steps and a few minutes of hands-on admin time. The new platform dramatically outperformed the older one, completing a complex database query set in only 74 minutes—a small fraction of the almost two days the legacy platform needed.

If you're still entrusting your company's data to SQL Server 2008, it is well past time to find an alternative. Using Microsoft Data Migration Assistant to migrate to the Lenovo ThinkSystem SR630 running SQL Server 2019 and Windows Server 2019 is a simple process that can help keep your data safe while also packing a strong performance boost.

[1] Lenovo XClarity Controller Support on ThinkSystem Servers," accessed December 10, 2019, https://lenovopress.com/lp0880-xcc-support-on-thinksystem-servers.

[2] Unpatched Vulnerabilities the Source of Most Data Breaches, accessed December 10, 2019, https://www.darkreading.com/vulnerabilities---threats/unpatched-vulnerabilities-the-source-of-most-data-breaches/d/d-id/1331465.

[3] Renee Dudley and Jeff Kao, "The Trade Secret: Firms That Promised High-Tech Ransomware Solutions Almost Always Just Pay the Hackers," accessed December 9, 2019, https://features.propublica.org/ransomware/ransomware-attack-data-recovery-firms-paying-hackers/.

[4] "The Need for Greater Focus on the Cybersecurity Challenges Facing Small and Midsize Businesses," accessed December 10, 2019, https://www.sec.gov/news/statement/cybersecurity-challenges-for-small-midsize-businesses.html#_edn6.

[5] "Only 16% of small business owners think they are at risk for a cyberattack," accessed December 10, 2019, https://www.insureon.com/blog/small-business-cybersecurity-poll.

[6] Downtime Can Cost Businesses More Than $10,000 an Hour (INFOGRAPHIC), accessed December 19, 2019, https://smallbiztrends.com/2019/08/cost-of-downtime-small-business.html

[7] 5 Shocking Disaster Recovery Statistics for SMBs, accessed December 19, 2019, https://www.wcatech.com/5-shocking-disaster-recovery-statistics-smbs/

[8] "What's new in SQL Server 2019 (15.x)," accessed December 19, 2019, https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-ver15?view=sql-server-ver15.

We concluded our hands-on testing on December 20, 2019. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on December 18, 2019 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

## Our results

The tables below presents our findings in detail. The first table shows the total time and steps to migrate a 1TB SQL Server database from the Lenovo ThinkServer® RD530 to the Lenovo ThinkSystem™ SR630 using Microsoft Data Migration Assistant. The second table shows the median time for one user to complete a workload of 22 complex data queries.

| | |
|---|---|
| Pre-migration assessment | |
| Number of steps | 12 |
| Hands-on time (hh:mm:ss) | 00:00:39 |
| Total time (hh:mm:ss) | 00:00:59 |
| Migration | |
| Number of steps | 7 |
| Hands-on time (hh:mm:ss) | 00:01:03 |
| Total time (hh:mm:ss) | 08:56:08 |
| Total | |
| Number of steps | 19 |
| Hands-on time (hh:mm:ss) | 00:01:42 |
| Total time (hh:mm:ss) | 08:57:07 |

| | Lenovo ThinkSystem SR630 | Lenovo ThinkServer RD530 |
|---|---|---|
| Median time for one user to complete a workload of 22 complex data queries (hh:mm:ss) | 01:15:36 | 44:09:58 |

# System configuration information

The table below presents detailed information on the systems we tested.

| Server configuration information | Lenovo ThinkSystem SR630 | Lenovo ThinkServer RD530 |
|---|---|---|
| BIOS name and version | UEFI Firmware 2.41 | Lenovo 10.10 |
| Non-default BIOS settings | N/A | N/A |
| Operating system name and version/build number | Microsoft Windows Server 2019 Standard Build 17763.864 | Microsoft Windows Server 2008 R2 Standard SP1 Build 7601 |
| Date of last OS updates/patches applied | 11/21/2019 | 11/21/2019 |
| Power management policy | Maximum Performance | Performance |
| Processor | | |
| Number of processors | 2 | 2 |
| Vendor and model | Intel® Xeon® Gold 6130 | Intel Xeon E5-2640 |
| Core count (per processor) | 16 | 6 |
| Core frequency (GHz) | 2.10 | 2.50 |
| Stepping | 4 | 7 |
| Memory module(s) | | |
| Total memory in system (GB) | 384 | 192 |
| Number of memory modules | 12 | 12 |
| Vendor and model | Hynix® HMA84GR7AFR4N-VK | Samsung® M393B2G70BH0-YH9 |
| Size (GB) | 32 | 16 |
| Type | PC4-2666 | PC3L-10600R |
| Speed (MHz) | 2,666 | 1,333 |
| Speed running in the server (MHz) | 2,666 | 1,333 |
| Storage controller | | |
| Vendor and model | ThinkSystem RAID 930-16i | LSI MegaRAID SAS 9260-8i |
| Cache size | 4GB | 512MB |
| Firmware version | 51.10.0-2837 | 12.15.0-0196 |
| Driver version | 7.705.8.0 | 6.703.3.0 |
| Local storage (OS) | | |
| Number of drives | 2 | 2 |
| Drive vendor and model | Lenovo LITEON CV3-8D128 | Seagate® ST9300653SS |
| Drive size (GB) | 128 | 300 |
| Drive information (speed, interface, type) | M.2, SSD | 6Gbps, SAS, 15K HDD |

| Server configuration information | Lenovo ThinkSystem SR630 | Lenovo ThinkServer RD530 |
|---|---|---|
| Local storage (data) | | |
| Number of drives | 4 | 4 |
| Drive vendor and model | Intel SSDSC2KB960GL | Seagate ST1200MM0017 |
| Drive size (GB) | 960 | 1,200 |
| Drive information (speed, interface, type) | 6Gbps, SATA, SSD | 6Gbps, SAS, 10K HDD |
| Local storage (logs) | | |
| Number of drives | 2 | 2 |
| Drive vendor and model | Intel SSDSC2KB960GL | Seagate ST600MM0006 |
| Drive size (GB) | 960 | 600 |
| Drive information (speed, interface, type) | 6Gbps, SATA, SSD | 6Gbps, SAS, 10K HDD |
| Network adapter | | |
| Vendor and model | Intel X722 | Intel I350 |
| Number and type of ports | 4 x 10Gb | 2 x 1Gb |
| Driver version | 1.8.103.2 | 12.13.27.0 |
| Cooling fans | | |
| Number of cooling fans | 7 | 8 |
| Power supplies | | |
| Vendor and model | ARTESYN® 700-014189-1500 | Delta Electronics® DPS-800RB |
| Number of power supplies | 2 | 2 |
| Wattage of each (W) | 1,100 | 800 |

# How we tested

## Overview

We used a legacy Lenovo ThinkServer RD530 and a new Lenovo ThinkSystem SR630. We configured the legacy system with hard disk drives (HDDs) and the Lenovo ThinkSystem SR630 with Intel® S4500 SATA SSDs. We configured the roles for the drives in each server as follows:

- two drives in a RAID 1 for the operating system (OS)
- four drives in a RAID 5 for the database data files
- two drives in a RAID 1 for the database log files

We installed Windows Server 2008 R2 Standard Edition and SQL Server 2008 on the legacy system. For the Lenovo ThinkSystem SR630, we installed Windows Server 2019 Standard Edition and SQL Server 2019 Standard.

After installing the OS and SQL Server on each system, we created a blank database on the legacy system, and used HammerDB 3.2 to populate it with a 1,000-scale (roughly 1 TB) TPC-H-like build. After populating the database, we backed up the database and used HammerDB to run 22 complex queries. We timed how long the legacy system took to complete the workload. After running the test, we dropped and restored the database, and restarted the server before running the test again. We performed three runs on the legacy system.

Once we finished the queries on the legacy system, we used Data Migration Assistant to migrate the database to the Lenovo ThinkSystem SR630. We made the following assumptions to narrow the scope of the testing:

- The systems would be running on a 1Gb network
- The organization would already have an Active Directory (AD) server and a Server Message Block (SMB) file share set up.
- Both the source system and target system would have read and write permissions to the AD server and the SMB file share.

After migrating our database to the Lenovo ThinkSystem SR630, we made a few changes to the keys and indexes in the database to take full advantage of SQL Server 2019 updated features. This required approximately 48 hours. The section Transact-SQL script has the script that includes all the changes we made. We then ran the same HammerDB benchmark to test the performance of SQL 2019 and clustered columnstore indexes.

## Installing Windows Server 2008 R2 Standard Edition

1. Using the virtual optical drive or another method, attach the Windows Server 2008 Standard installation media to the server.
2. Boot the server to the installation media.
3. At the Language Selection Screen, click Next.
4. Click Install Now.
5. Select Windows Server 2008 R2 Standard (Full Installation), and click Next.
6. Click the I accept the license terms check box, and click Next.
7. Click Custom.
8. Click Drive options (advanced).
9. Select the RAID1 group created for the OS install, and click Next.
10. Click Apply.
11. Click Next.
12. At the User's password must be changed before logging on warning screen, click OK.
13. Type `Password1` as the new password in both fields, and click the arrow to continue.
14. At the Your password has been changed screen, click OK.

## Installing Windows Server 2019 Standard Edition

1. Using the virtual optical drive or another method, attach the Windows Server 2019 Standard installation media to the server.
2. Boot the server to the installation media.
3. At the language selection screen, click Next.
4. Click Install Now.
5. Enter the product key, and click Next.
6. Select Windows Server 2019 Standard (Desktop Experience), and click Next.
7. Check the I accept the license terms checkbox, and click Next.
8. Click Custom: Install Windows only (advanced).
9. Select the RAID1 group created for the OS install, and click Next.
10. Once the installation completes, enter a password for the Administrator user, and click Finish.
11. We completed the Windows Server 2019 configuration by disabling the firewall, real-time protection, IE security, and running all Windows Updates. We also set the time zone, and enabled Remote Desktop access.

## Installing SQL Server 2008 Standard Edition on the ThinkServer RD530

1. Insert the installation DVD for SQL Server 2008.
2. If AutoPlay does not begin the installation, navigate to the SQL Server 2008 DVD, and double-click.
3. If prompted with a .NET installation prompt, click Yes to enable the .NET Framework Core role.
4. At the SQL Server Installation Center screen, click Installation.
5. Click New installation or Add features to an existing installation.
6. At the Setup Support Rules screen, click OK.
7. At the Product Key screen, specify the key or choose free Enterprise edition evaluation, and click Next. Note: We used and assumed Standard edition for our testing.
8. At the License Terms screen, accept the license terms, and click Next.
9. At the Setup Support Files screen, click Install.
10. At the Setup Support Rules screen, click Next.
11. At the Setup Role screen, choose SQL Server Feature Installation, and click Next.
12. At the SQL Server 2008 Feature Selection screen, select the following features: Database Engine Services, Client Tools Connectivity, Client Tools Backwards Compatibility, Management Tools – Basic, Management Tools – Complete, and click Next.
13. At the Installation Rules screen, click Next.
14. At the Instance Configuration screen, leave the defaults, and click Next.
15. At the Disk Space Requirements screen, click Next.
16. At the Server Configuration screen, choose the service account, enter a password if necessary, and click Next.
17. At the Database Engine Configuration screen, select Mixed Mode, enter a password for the system administrator (sa) account, click Add Current User, and click Next.
18. At the Error Reporting screen, click Next.
19. At the Installation Configuration Rules screen, click Next.
20. At the Installation screen, click Install.
21. At the Complete screen, click Close.
22. After installation completes, click Start→All Programs→Microsoft SQL Server 2008→Configuration Tools→SQL Server Configuration Manager.
23. Expand the Network Configuration node, and select the relevant instance.
24. In the right pane, right-click TCP/IP, and choose Enable.
25. In the left pane, click SQL Server Services.
26. In the right pane, right-click the SQL Server service, and select Restart.

## Installing SQL Server 2019 Standard Edition on the ThinkSystem SR630

1. Attach the installation media ISO for SQL Server 2019 to the VM.
2. Click Run SETUP.EXE. If Autoplay does not begin the installation, navigate to the SQL Server 2019 DVD, and double-click it.
3. In the left pane, click Installation.
4. Click New SQL Server stand-alone installation or add features to an existing installation.
5. At the Product Key screen, specify the key or choose free Enterprise edition evaluation, and click Next. Note: We used and assumed Standard edition for our testing and analysis.
6. To accept the license terms, click the checkbox, and click Next.
7. Click Use Microsoft Update to check for updates, and click Next.
8. To install the setup support files, click Install.
9. If no failures appear, click Next.
10. At the Setup Role screen, choose SQL Server Feature Installation, and click Next.
11. At the Feature Selection screen, select Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, and Client Tools Backwards Compatibility.
12. Click Next.
13. At the Instance Configuration screen, leave the instance default selection, and click Next.
14. At the Database Engine Configuration screen, select the authentication method you prefer. For our testing purposes, we selected Mixed Mode.
15. Enter and confirm a password for the system administrator account.
16. Click Add Current user. This may take several seconds.
17. Click the Data Directories tab.
18. Change the database directory, the database log directory, and the backup directory storage locations to the data, log, and backup volumes.
19. Click the TempDB tab.
20. Change the number of files to 8.

21. Change the initial file size to 1,024 MB.
22. Change the Data Directory to the data volume.
23. Change the intial size of the log to 1,024 MB.
24. Change the Log Directory to the log volume.
25. Click Next.
26. At the Error and usage reporting screen, click Next.
27. At the Installation Configuration Rules screen, check that there are no failures or relevant warnings, and click Next.
28. At the Ready to Install screen, click Install.
29. When the install finishes, go to the Installation tab in the Installation Center, and click Install SQL Server Management Tools.
30. In the browser that pops up, click Download SQL Server Management Studio 18.X.
31. To open the installer, click the download.
32. Click Run.
33. Click Install.
34. Once the installation completes, close the installation window.

## Enabling Lock Pages in Memory

1. Click Start, type `gpedit.msc`, and press Enter.
2. In the Local Group Policy Editor, expand Computer Configuration→Windows Settings→Security Settings→Local Policies.
3. Select User Rights Assignment, and in the right pane, scroll down to Lock pages in memory.
4. Right-click Lock pages in memory, and click Properties.
5. Click Add User or Group, and add your SQL user account (NT SERVICE\MSSQLSERVER).
6. Click OK.

## Creating a database on SQL Server 2008

1. Open SQL Server Management Studio.
2. Select SQL Server Authentication, type `sa` and `Password1` as your login and password respectively, and click Connect.
3. Right-click Databases→New Database.
4. Name the database. We named ours `tpch`.
5. Click OK.

## Installing HammerDB 3.2 and Microsoft ODBC Driver 17 for SQL Server

1. To download the most recent HammerDB release, navigate to https://hammerdb.com/download.html/.
2. Run the HammerDB installation file, select your language, and click OK.
3. To install HammerDB when prompted, click Yes.
4. At the Welcome window, click Next.
5. Accept the default install location, and click Next.
6. Accept the default setup type, and to start the installation, click Next.
7. To complete the installation, click Finish.
8. To download Microsoft ODBC Driver 17 for SQL Server, navigate to https://docs.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-ver15.
9. Run the downloaded installer.
10. In the Welcome window, click Next.
11. Accept the terms of the license agreement, and click Next.
12. Accept the default installation options, and click Next.
13. To begin the installation, click Install.
14. To close the installer when the installation completes, click Finish.

## Building the database

1. Open HammerDB.
2. Select Options→Benchmark.
3. In the Benchmark Options window, select MSSQL Server and TPC-H, and click OK.
4. In the left pane, expand TPC-H, then Schema Build, and double-click Options.
5. Use the following options for the TPC-H Build Options:
   - SQL Server: (local)
   - SQL Server ODBC Driver: ODBC Driver 17 for SQL Server
   - Authentication: SQL Server Authentication
   - SQL Server User ID: sa (default)
   - SQL Server User Password: Your password
   - SQL Server Database: tpch
   - MAXDOP: 4
   - Scale Factor: 1000
   - Virtual Users to Build Schema: 24
6. Click OK.
7. To start the database population, in the left pane under Schema Build, double-click Build.

## Running the test

1. Open HammerDB.
2. Select Options→Benchmark.
3. In the Benchmark Options window, select MSSQL Server and TPC-H, and click OK.
4. In the left pane, expand TPC-H and Driver Script, and double-click Options.
5. Use the following options for the TPC-C Driver Script:
   - SQL Server: (local)
   - SQL Server ODBC Driver: ODBC Driver 17 for SQL Server
   - Authentication: SQL Server Authentication
   - SQL Server User ID: sa (default)
   - SQL Server User Password: Your password
   - SQL Server Database: tpch
   - MAXDOP: 0
   - Total Query Sets per User: 1
6. Click OK.
7. To load the driver script, in the left pane, double-click Load.
8. In the left pane, expand Virtual User, and double-click Options.
9. Set the Virtual Users field to 1, and check all boxes.
10. Click OK.
11. To begin the test, under Virtual User in the left pane, double-click Run.

## Running the pre-migration assessment

1. Follow these steps to create a user account for migration on each server and to grant them SQL permissions. This ensures there's a user with the proper SQL permissions on both ends of the migration to handle the database:
   a. Log into the source and target servers, and using the Windows Add and Remove user tool, create a user account for migration on each.
   b. On the file share where the backup will be temporarily stored, edit the properties of the shared drive to grant read and write access.
   c. Click Start, and type `Services`.
   d. Go to the MSSQLSERVER service, and right-click.
   e. Select Properties.
   f. Select the Log On tab.
   g. Select This Account, and change it to the user you just created.
   h. Enter the password, and click Apply.
   i. Restart the MSSQLSERVER service.

2. Download & install the Data Migration Assistant from https://www.microsoft.com/en-us/download/details.aspx?id=53595.
3. Open Data Migration Assistant. In the navigation pane on the left, click the plus sign to start a new project, fill out the fields below, and click Create.
    a. Select Assessment, and enter a project name.
    b. From the Assessment Type drop-down menu, select Database Engine.
    c. From the Source Server Type drop-down menu, select SQL Server.
    d. From the Target Server Type drop-down menu, select SQL Server.
4. When the project creation finishes, go to the Options screen, and from the Target Version drop-down men, select SQL Server 2019 on Windows.
5. Check the boxes beside Compatibility Issues and New Features Recommendation, and click Next.
6. On the Select Sources screen, enter the IP address or fully qualified domain name (FQDN) of the source server.
7. From the drop-down menu, select an authentication type, and, if necessary, enter the username and password credentials below.
8. Check the boxes beside Encrypt connection and Trust server certificate, and click Connect.
9. In the Add Sources sidebar, select the server and database(s) you wish to access, and click Add.
10. Click Start Assessment. When the assessment finishes, the Data Migration Assistant will display any compatibility issues or new features you should know about.
11. Check to make sure there are no compatibility issues. If you do see compatibility issues, use the suggestions the Data Migration Assistant provides to resolve them.
12. To save a copy of any feature recommendations post migration, click Export report.

## Initiating the migration

1. Open the Data Migration Assistant, and in the navigation pane on the left, click the plus sign to start a new project, fill out the fields below, and click Create.
    a. Select Migration, and enter a project name.
    b. From the Source Server Type drop-down menu, select SQL Server.
    c. From the Target Server Type drop-down menu, select SQL Server.
2. On the Specify source & target screen, enter the IP address or fully qualified domain name (FQDN) of the source server and target servers.
3. From the drop-down menu, select an Authentication type, and, if necessary, enter the username and password credentials for both the source and target servers.
4. Check the boxes beside Encrypt connection and Trust server certificate for both the source and target servers.
5. On the Add Databases screen, enter the IP address or fully qualified domain name (FQDN) of the path to the fileshare.
6. To restore the data and log files on the target server, specify the location, and click Next.
7. On the Select Logins screen, check the box next to the logins you wish to migrate, and click Start Migration.

# T-SQL script for index recreation

```
use [tpch];
GO
use [master];
GO
USE [master]
GO
ALTER DATABASE [tpch] SET COMPATIBILITY_LEVEL = 140
GO
ALTER DATABASE [tpch] SET RECOVERY SIMPLE WITH NO_WAIT
GO

/* Drop customer nation foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[customer] DROP CONSTRAINT [customer_nation_fk]
GO

/* Drop LineItem order foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem] DROP CONSTRAINT [lineitem_order_fk]
```

```
GO

/* Drop LineItem partkey foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem] DROP CONSTRAINT [lineitem_partkey_fk]
GO


/* Drop LineItem partsupp foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem] DROP CONSTRAINT [lineitem_partsupp_fk]
GO

/* Drop LineItem suppkey foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem] DROP CONSTRAINT [lineitem_suppkey_fk]
GO

/* Drop Nation region foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[nation] DROP CONSTRAINT [nation_region_fk]
GO

/* Drop Orders customer foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[orders] DROP CONSTRAINT [order_customer_fk]
GO

/* Drop PartSupp part foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[partsupp] DROP CONSTRAINT [partsupp_part_fk]
GO

/* Drop ParttSupp supplier foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[partsupp] DROP CONSTRAINT [partsupp_supplier_fk]
GO

/* Drop Supplier nation foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[supplier] DROP CONSTRAINT [supplier_nation_fk]
GO

/* Drop Customer primary key and index */
USE [tpch]
GO

/****** Object:  Index [customer_pk]    Script Date: 11/21/2019 2:07:15 PM ******/
ALTER TABLE [dbo].[customer] DROP CONSTRAINT [customer_pk] WITH ( ONLINE = OFF )
```

```
GO



/* Drop Nation primary key and index */
USE [tpch]
GO

/****** Object:  Index [nation_pk]    Script Date: 11/21/2019 2:09:03 PM ******/
ALTER TABLE [dbo].[nation] DROP CONSTRAINT [nation_pk] WITH ( ONLINE = OFF )
GO

/* Drop Orders primary key and index */
USE [tpch]
GO

/****** Object:  Index [orders_pk]    Script Date: 11/21/2019 2:10:06 PM ******/
ALTER TABLE [dbo].[orders] DROP CONSTRAINT [orders_pk]
GO

/* Drop Part primary key and index */
USE [tpch]
GO

/****** Object:  Index [part_pk]    Script Date: 11/21/2019 2:12:46 PM ******/
ALTER TABLE [dbo].[part] DROP CONSTRAINT [part_pk] WITH ( ONLINE = OFF )
GO

/* Drop PartSupp primary key and index */
USE [tpch]
GO

/****** Object:  Index [partsupp_pk]    Script Date: 11/21/2019 2:13:33 PM ******/
ALTER TABLE [dbo].[partsupp] DROP CONSTRAINT [partsupp_pk] WITH ( ONLINE = OFF )
GO

/* Drop Region primary key and index */
USE [tpch]
GO

/****** Object:  Index [region_pk]    Script Date: 11/21/2019 2:14:10 PM ******/
ALTER TABLE [dbo].[region] DROP CONSTRAINT [region_pk] WITH ( ONLINE = OFF )
GO

/* Drop Supplier primary key and index */
USE [tpch]
GO

/****** Object:  Index [supplier_pk]    Script Date: 11/21/2019 2:14:32 PM ******/
ALTER TABLE [dbo].[supplier] DROP CONSTRAINT [supplier_pk] WITH ( ONLINE = OFF )
GO

/* Create customer unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [customer_pk]    Script Date: 11/21/2019 2:17:08 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [customer_pk] ON [dbo].[customer]
(
    [c_custkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

/* Create Customer clustered columnstore index */
```

```
USE [tpch]
GO

/****** Object:  Index [cust_cs]    Script Date: 11/21/2019 2:18:38 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [cust_cs] ON [dbo].[customer] WITH (DROP_EXISTING = OFF, 
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Drop LineItem shipdate index and recreate as a non-clustered index */
USE [tpch]
GO

/****** Object:  Index [l_shipdate_ind]    Script Date: 11/21/2019 2:23:11 PM ******/
DROP INDEX [l_shipdate_ind] ON [dbo].[lineitem] WITH ( ONLINE = OFF )
GO

USE [tpch]
GO

/****** Object:  Index [l_shipdate_ind]    Script Date: 11/21/2019 2:24:53 PM ******/
CREATE NONCLUSTERED INDEX [l_shipdate_ind] ON [dbo].[lineitem]
(
    [l_shipdate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, 
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 95) ON [PRIMARY]
GO

/* Create LineItem clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [lineit_cs]    Script Date: 11/21/2019 2:26:21 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [lineit_cs] ON [dbo].[lineitem] WITH (DROP_EXISTING = OFF, 
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Create Nation unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [nation_pk]    Script Date: 11/21/2019 2:31:55 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [nation_pk] ON [dbo].[nation]
(
    [n_nationkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, 
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

/* Create Nation clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [nation_cs]    Script Date: 11/21/2019 2:33:08 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [nation_cs] ON [dbo].[nation] WITH (DROP_EXISTING = OFF, 
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Drop Orders orderdate index and recreate as a non-clustered index */
USE [tpch]
GO

/****** Object:  Index [o_orderdate_ind]    Script Date: 11/21/2019 2:38:59 PM ******/
DROP INDEX [o_orderdate_ind] ON [dbo].[orders] WITH ( ONLINE = OFF )
GO
```

```
USE [tpch]
GO

/****** Object:  Index [o_orderdate_ind]    Script Date: 11/21/2019 2:40:09 PM ******/
CREATE NONCLUSTERED INDEX [o_orderdate_ind] ON [dbo].[orders]
(
    [o_orderdate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 95) ON [PRIMARY]
GO

/* Create Orders unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [orders_pk]    Script Date: 11/21/2019 2:35:05 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [orders_pk] ON [dbo].[orders]
(
    [o_orderkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 95) ON
[PRIMARY]
GO

/* Create Orders clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [ord_cs]    Script Date: 11/21/2019 2:36:12 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [ord_cs] ON [dbo].[orders] WITH (DROP_EXISTING = OFF, COMPRESSION_
DELAY = 0) ON [PRIMARY]
GO

/* Create Part unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [part_pk]    Script Date: 11/21/2019 2:48:55 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [part_pk] ON [dbo].[part]
(
    [p_partkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

/* Create Part clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [part_cs]    Script Date: 11/21/2019 2:49:23 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [part_cs] ON [dbo].[part] WITH (DROP_EXISTING = OFF, COMPRESSION_
DELAY = 0) ON [PRIMARY]
GO

/* Create PartSupp unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [partsupp_pk]    Script Date: 11/21/2019 2:54:37 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [partsupp_pk] ON [dbo].[partsupp]
(
    [ps_partkey] ASC,
    [ps_suppkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
GO

/* Create PartSupp clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [psupp_cs]    Script Date: 11/21/2019 2:56:21 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [psupp_cs] ON [dbo].[partsupp] WITH (DROP_EXISTING = OFF,
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Create Region unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [region_pk]    Script Date: 11/21/2019 2:58:54 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [region_pk] ON [dbo].[region]
(
    [r_regionkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

/* Create Region clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [region_cs]    Script Date: 11/21/2019 3:00:09 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [region_cs] ON [dbo].[region] WITH (DROP_EXISTING = OFF,
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Create Supplier unique non-clustered index */
USE [tpch]
GO

/****** Object:  Index [supplier_pk]    Script Date: 11/21/2019 3:01:16 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [supplier_pk] ON [dbo].[supplier]
(
    [s_suppkey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

/* Create Supplier clustered columnstore index */
USE [tpch]
GO

/****** Object:  Index [suppl_cs]    Script Date: 11/21/2019 3:02:14 PM ******/
CREATE CLUSTERED COLUMNSTORE INDEX [suppl_cs] ON [dbo].[supplier] WITH (DROP_EXISTING = OFF,
COMPRESSION_DELAY = 0) ON [PRIMARY]
GO

/* Create Customer nation foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[customer]  WITH NOCHECK ADD  CONSTRAINT [customer_nation_fk] FOREIGN KEY([c_
nationkey])
REFERENCES [dbo].[nation] ([n_nationkey])
GO

ALTER TABLE [dbo].[customer] CHECK CONSTRAINT [customer_nation_fk]
GO
```

```
/* Create LineItem order foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem]  WITH NOCHECK ADD  CONSTRAINT [lineitem_order_fk] FOREIGN KEY([l_
orderkey])
REFERENCES [dbo].[orders] ([o_orderkey])
GO

ALTER TABLE [dbo].[lineitem] CHECK CONSTRAINT [lineitem_order_fk]
GO

/* Create LineItem partkey foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem]  WITH NOCHECK ADD  CONSTRAINT [lineitem_partkey_fk] FOREIGN KEY([l_
partkey])
REFERENCES [dbo].[part] ([p_partkey])
GO

ALTER TABLE [dbo].[lineitem] CHECK CONSTRAINT [lineitem_partkey_fk]
GO

/* Create LineItem partsupp foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem]  WITH NOCHECK ADD  CONSTRAINT [lineitem_partsupp_fk] FOREIGN KEY([l_
partkey], [l_suppkey])
REFERENCES [dbo].[partsupp] ([ps_partkey], [ps_suppkey])
GO

ALTER TABLE [dbo].[lineitem] CHECK CONSTRAINT [lineitem_partsupp_fk]
GO

/* Create LineItem suppkey foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[lineitem]  WITH NOCHECK ADD  CONSTRAINT [lineitem_suppkey_fk] FOREIGN KEY([l_
suppkey])
REFERENCES [dbo].[supplier] ([s_suppkey])
GO

ALTER TABLE [dbo].[lineitem] CHECK CONSTRAINT [lineitem_suppkey_fk]
GO

/* Create Nation region foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[nation]  WITH NOCHECK ADD  CONSTRAINT [nation_region_fk] FOREIGN KEY([n_
regionkey])
REFERENCES [dbo].[region] ([r_regionkey])
GO

ALTER TABLE [dbo].[nation] CHECK CONSTRAINT [nation_region_fk]
GO

/* Create Orders customer foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[orders]  WITH NOCHECK ADD  CONSTRAINT [order_customer_fk] FOREIGN KEY([o_custkey])
```

```
REFERENCES [dbo].[customer] ([c_custkey])
GO

ALTER TABLE [dbo].[orders] CHECK CONSTRAINT [order_customer_fk]
GO

/* Create PartSupp part foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[partsupp]  WITH NOCHECK ADD  CONSTRAINT [partsupp_part_fk] FOREIGN KEY([ps_
partkey])
REFERENCES [dbo].[part] ([p_partkey])
GO

ALTER TABLE [dbo].[partsupp] CHECK CONSTRAINT [partsupp_part_fk]
GO

/* Create PartSupp supplier foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[partsupp]  WITH NOCHECK ADD  CONSTRAINT [partsupp_supplier_fk] FOREIGN KEY([ps_
suppkey])
REFERENCES [dbo].[supplier] ([s_suppkey])
GO

ALTER TABLE [dbo].[partsupp] CHECK CONSTRAINT [partsupp_supplier_fk]
GO

/* Create Supplier nation foreign key */
USE [tpch]
GO

ALTER TABLE [dbo].[supplier]  WITH NOCHECK ADD  CONSTRAINT [supplier_nation_fk] FOREIGN KEY([s_
nationkey])
REFERENCES [dbo].[nation] ([n_nationkey])
GO

ALTER TABLE [dbo].[supplier] CHECK CONSTRAINT [supplier_nation_fk]
GO

USE [tpch]
GO
DBCC SHRINKFILE (N'tpch' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch02' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch03' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch04' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch05' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch06' , 131072)
```

```
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch07' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch08' , 131072)
GO
USE [tpch]
GO
DBCC SHRINKFILE (N'tpch_log' , 20480)
GO

USE [master]
GO
ALTER DATABASE [tpch] SET RECOVERY FULL WITH NO_WAIT
GO
```

This project was commissioned by Lenovo.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:
Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.