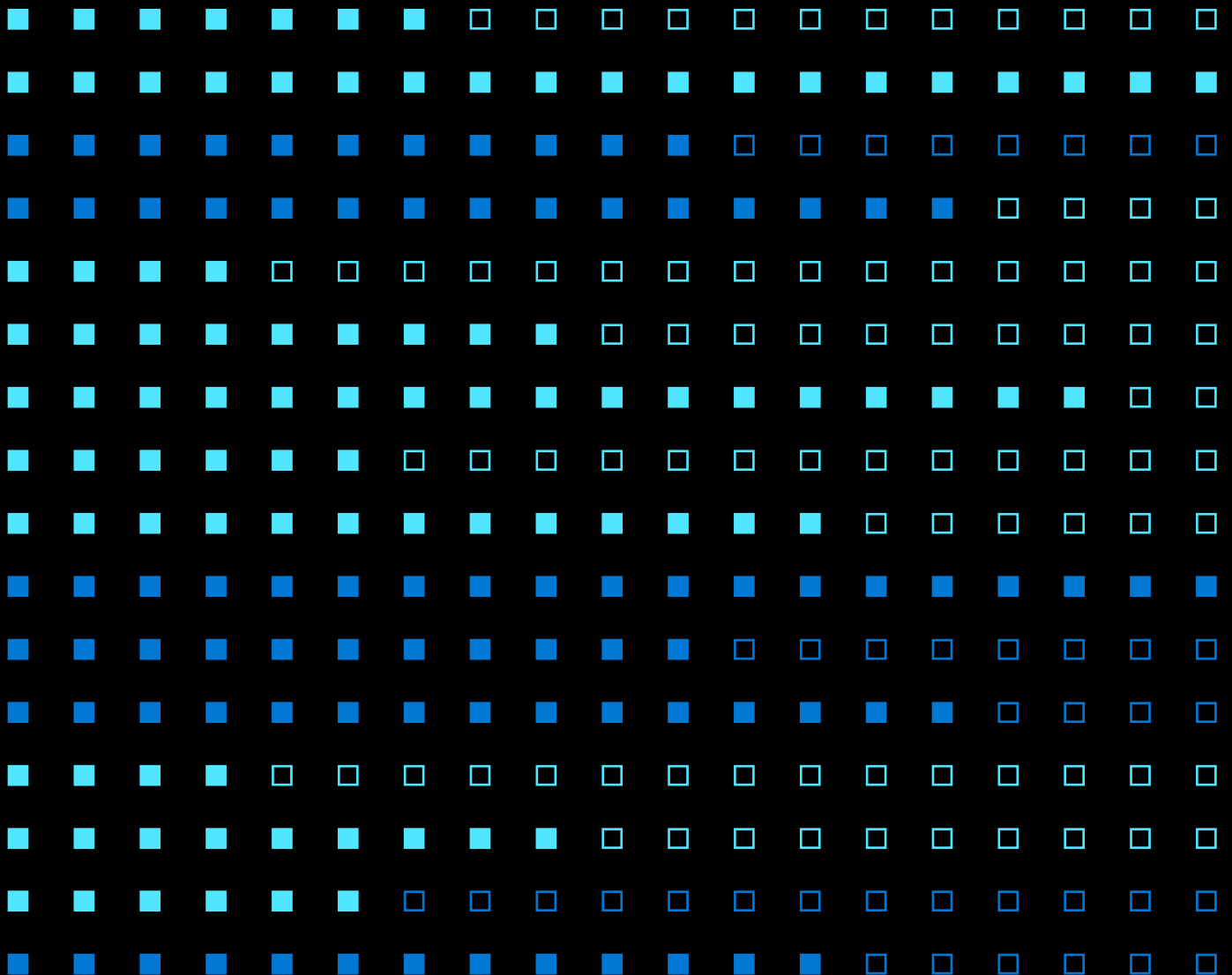


# Microsoft SQL Server 2019

## Technical white paper

**Published:** September 2018

**Applies to:** Microsoft SQL Server 2019 CTP 2.0 for Windows, Linux, and Docker containers



## Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. This content was developed prior to the product or service' release and as such, we cannot guarantee that all details included herein will be exactly as what is found in the shipping product. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. The information represents the product or service at the time this document was shared and should be used for planning purposes only.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Information subject to change at any time without prior notice.

Microsoft, Active Directory, Azure, Bing, Excel, Power BI, SharePoint, Silverlight, SQL Server, Visual Studio, Windows, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

© 2018 Microsoft Corporation. All rights reserved.

# Contents

Summary .....	4
Industry landscape and trends .....	4
Data virtualization .....	4
Platform flexibility in the data estate .....	5
SQL Server 2019: power and flexibility.....	6
Enhanced PolyBase —query over any type of data .....	7
SQL Server Big Data Clusters —scalable compute and storage .....	9
Database engine enhancements .....	11
Performance and scale.....	11
High availability .....	11
Security and compliance .....	12
UTF-8 support .....	14
SQL Server on Linux .....	14
Containers .....	15
Machine learning.....	15
SQL Graph.....	15
Intelligent database and query processing .....	16
Troubleshooting and diagnostics .....	17
Business Intelligence .....	18
Reporting Services.....	18
Power BI Report Server .....	18
Analysis Services .....	19
Enterprise Information Management .....	21
SQL Server Integration Services .....	21
Master Data Services .....	21
SQL Server 2019 tooling .....	22
Conclusion .....	23
Calls to action.....	23

## Summary

Microsoft SQL Server 2019 powers your organization by providing a data hub that you can use to access structured and unstructured data sources from across your entire data estate through a consistent interface. The relational database engine scales to petabytes of data, and **enhancements to PolyBase allow you to process diverse big data and relational data sources using Transact-SQL from SQL Server.**

Building on SQL Server on Linux in Docker containers, Apache Spark and the Hadoop ecosystem, and the rapidly-forming industry consensus on **Kubernetes as a container orchestrator**, with SQL Server 2019 Big Data Clusters you can deploy scalable clusters of SQL Server containers to read, write, and process big data from Transact-SQL, allowing you to easily combine your high-value relational data with high-volume big data with a single query.

The SQL Server 2019 database engine supports an even wider choice of platform and programming language—including support for third-party language runtimes—and bringing SQL Server on Linux closer to feature parity with SQL Server on Windows.

**SQL Server remains the only commercial database with AI built in**, and now supports even more machine learning scenarios. **SQL Server Machine Learning Services gives you the ability to do end to end machine learning in the database without moving data.** You can train the models using open source R or Python, and Microsoft's scalable algorithms. Once trained, making machine learning scripts and models operational is as simple as embedding them in Transact-SQL scripts. Any application connecting to SQL Server can take advantage of the predictions and intelligence from these models by simply calling a stored procedure.

SQL Server 2019 builds on previous versions of SQL Server, which are industry leaders in performance and security; **SQL Server has been a leader in TPC-E and TPC-H benchmarks for the last five years, and the least vulnerable database during the last eight years.** It offers better performance than ever before, and new features to help manage data security and compliance.

*Please note: this document describes the features available in the first public preview of SQL Server 2019; CTP 2.0. More features will be added in later releases.*

## Industry landscape and trends

### Data virtualization

Recognizing that **different storage technologies are more appropriate for different types of data**, a modern enterprise is likely to have data stored in a mixture of relational and non-relational data stores—often from several different vendors. A challenge for developers, data scientists, and business analysts is that to extract business value from this data, they typically need to combine data from disparate sources; they typically do this by bringing all the relevant data from the source systems together on a single platform.

**In traditional business intelligence systems, copies of data are created and loaded into a reporting platform with extract-transform-load (ETL) processes; reporting and analysis is carried out on the copies.** Whilst enabling enterprises to extract business value from their data, **ETL processes have several common issues:**

- **Expensive to develop, maintain, and support**—if they are to be repeatable and robust, ETL processes require effort to create, effort to keep them up to date, and effort to keep them running.
- **Slow**—ETL processes introduce an **inherent delay**. An IDC study<sup>1</sup> found that more than 80% of data sets delivered by ETL processes is between 2 and 7 days old by the time it reaches an analytical system. 75% of businesses reported that delays in data processing had inhibited business opportunities.
- **Must be secured**—Each copy of a data set must be secured against unauthorized access, especially if the data set contains personally identifying information (PII).
- **Require storage**—Each copy of a data set requires disk space to store—these costs grow if a data set is very large or is copied many times.

An alternative to ETL is data virtualization. Data virtualization integrates data from disparate sources, locations and formats, without replicating or moving the data, to create a single "virtual" data layer that delivers unified data services to support multiple applications and users. The virtual data layer—sometimes referred to as a data hub or data lake—allows users to query data from many sources through a consistent interface. Users' access to sensitive data sets can be controlled from a single location, and the delays inherent to ETL need not apply; data sets can be up to date.

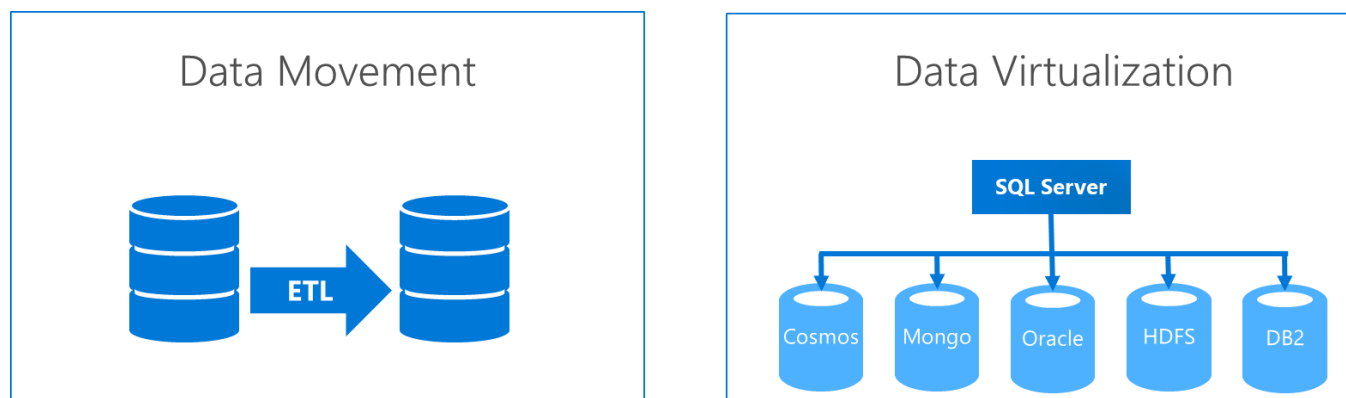


Figure 1: Data movement and data virtualization

## Platform flexibility in the data estate

Enterprises want the flexibility to run best-in-class database software on any platform, as shown by the success of SQL Server on Linux and SQL Server in Docker containers. SQL Server 2017 on Linux is Microsoft's most successful SQL Server product ever, with over seven million downloads since its release in October 2017. With the continued rise of container orchestration systems like Kubernetes, database systems must be supported on the widest range of operating systems and virtualization platforms.

<sup>1</sup> 3rd Platform Information Management Requirements Survey, IDC, October, 2016, n=502

## SQL Server 2019: power and flexibility

SQL Server 2019 builds on the industry-leading<sup>2</sup> capabilities of SQL Server 2017, holding benchmarks in such areas as:

- **Performance**—SQL Server owns the top TPC-E<sup>3</sup> performance benchmarks for transaction processing, the top TPC-H<sup>4</sup> performance benchmarks for data warehousing—at 1,000 GB, 10,000 GB, and 30,000 GB—and the top performance benchmarks with leading business applications.
- **Security**—According to the National Institute of Standards and Technology (NIST) public security board, SQL Server continues to have the lowest number of reported security vulnerabilities across the major database vendors (NIST, 2010-2017).

SQL Server 2019 continues the evolution of SQL Server, bringing new capabilities to the modern data ecosystem to better support and enhance data management and data-driven applications.

Enhancements in SQL Server 2019 fall into five main themes:

- **Reason over data anywhere**—including better integration with big data systems, and new connectors for data virtualization.
- **Choice of language and platform**—including more container scenarios, more supported platforms, and greater extensibility.
- **Industry leading performance and security**—including better performance, extensions to intelligent query processing, and additional features to support GDPR compliance.
- **The only commercial database with AI built in**—built-in machine learning is now supported in more scenarios, including machine learning in SQL Server on Linux, and support for machine learning in failover cluster instances.
- **Enhancing SQL Server on Linux**—bringing SQL Server on Linux closer to feature-parity with SQL Server on Windows, including support for transactional replication and distributed transactions.

---

<sup>2</sup> Gartner has rated Microsoft as a leader with the most complete vision and highest ability to execute of any operational database management system for three consecutive years (2015, 2016, and 2017). *Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.*

<sup>3</sup> TPC-E Top Ten Performance Results, TPC.org ([link](#))

<sup>4</sup> TPC-H - Top Ten Performance Results - Non-Clustered, TPC.org ([link](#))

## Enhanced PolyBase —query over any type of data

First added to the SQL Server database engine in SQL Server 2016, PolyBase allowed customers to query big data stored in HDFS-compatible Hadoop distributions and file systems such as HortonWorks, Cloudera, and Azure Blob Storage from Transact-SQL by defining an external table to represent HDFS data in SQL Server. Users can write Transact-SQL queries that reference the external table as if it were a normal SQL Server table; when the query is executed, data from the external table is retrieved and displayed to the user.

SQL Server 2019 extends capabilities of PolyBase with new connectors; you can now create external tables that link to a variety of data stores, including SQL Server, Oracle, Teradata, MongoDB, or any data source with an ODBC driver.

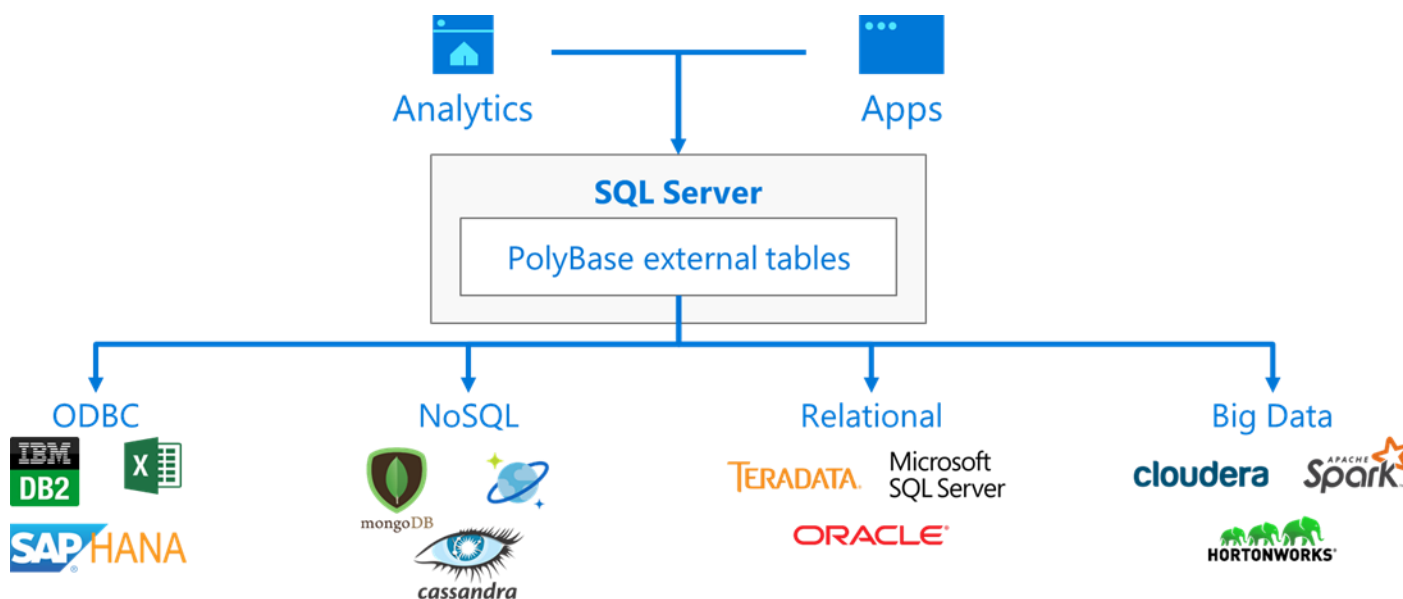


Figure 2: Data sources available with the enhanced PolyBase

Once you have created external tables in SQL Server, you can use Active Directory to control access to data sources, granting access to external tables to Active Directory users and groups.

PolyBase already optimizes performance by using push-down computation—operations including projections, predicates, aggregates, limit, and homogeneous joins are all pushed to the source system, and the results of these operations are returned to SQL Server—improving performance by reducing network traffic. In SQL Server 2019 Big Data Clusters the SQL Server engine has gained the ability to read HDFS files natively, and by using SQL Server instances on the HDFS data nodes to filter and aggregate data locally.

You can further increase the performance and capacity of PolyBase with scale-out of SQL Server instances; many SQL Server instances can be added to a PolyBase group, under the control of a group head node. You issue PolyBase queries to the head node, which distributes the workload across the PolyBase group's compute instances; this enables parallel ingestion and processing of external data.

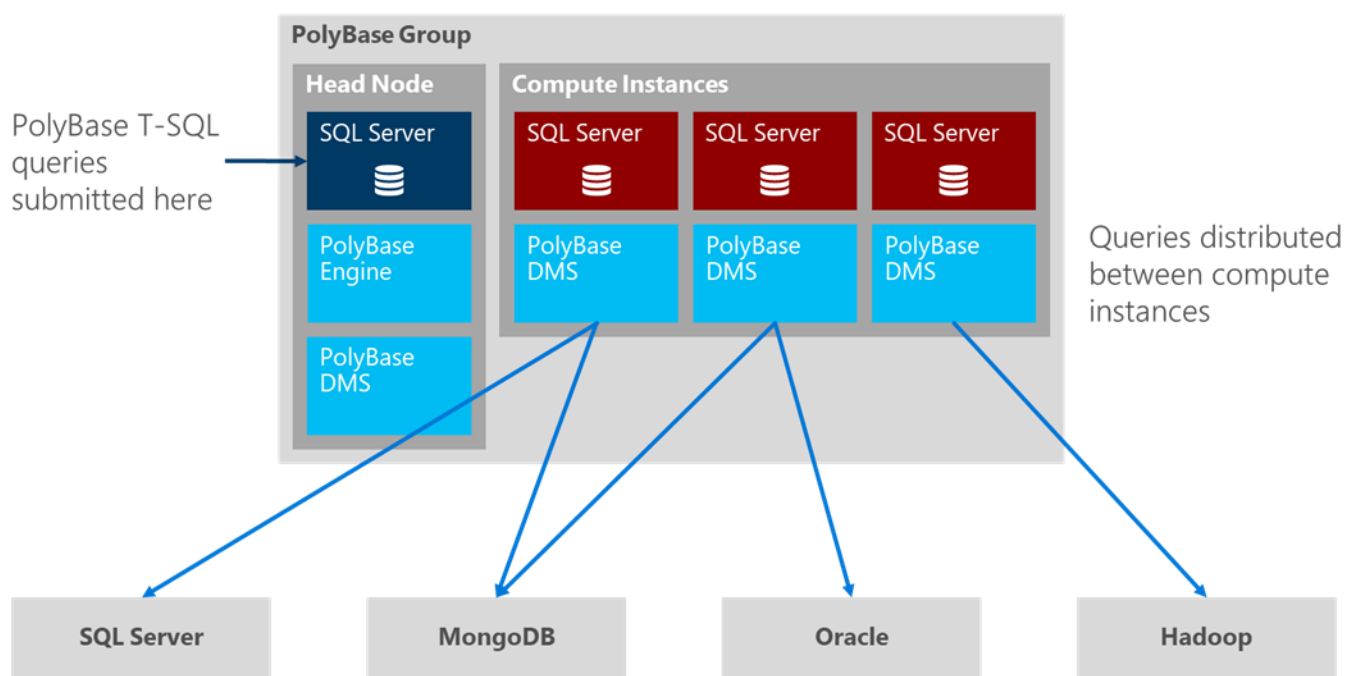


Figure 3: PolyBase scale-out

As you add more data sources and data volumes increase, you can scale out the PolyBase group by adding more compute instances to maintain consistent performance. As your data virtualization workloads change you can add and remove compute instances on-the-fly within seconds.

With enhanced PolyBase, you can bring together and secure many disparate data sources for reporting and analysis inside SQL Server, without the need to develop and run ETL processes.



## SQL Server Big Data Clusters —scalable compute and storage

SQL Server 2019 Big Data Clusters take the enhancements to PolyBase to improve the data virtualization experience between SQL Server and other database engines, and add faster, more secure, and bi-directional integration with big data Hadoop and Apache Spark systems. SQL Server Big Data Clusters bring SQL Server together with industry-standard big data tools in a package supported by Microsoft to offer deep integration between SQL Server and big data in a form that is easy to deploy and manage. It offers three major pieces of functionality:

- Data virtualization—Combine data from many sources without moving or replicating it. Scale out compute and caching to boost performance.
- Managed SQL Server, Spark and data lake—Store high volume data in a data lake and access it easily using either SQL or Spark. Management services, admin portal, and integrated security make it all easy to manage.
- Complete AI platform—Easily feed integrated data from many sources to your model training. Ingest and prep data, and then train, store, and operationalize your models all in one system.

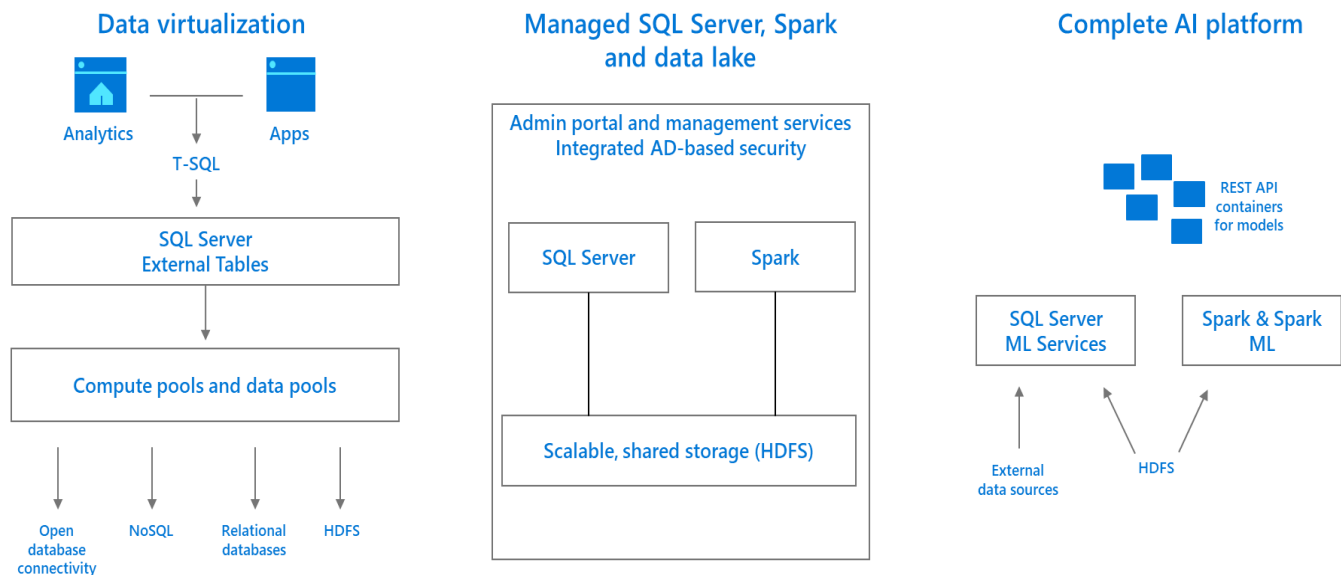


Figure 4: SQL Server Big Data Clusters functionality

The highest value enterprise data has long since been stored in a relational database like SQL Server, but some interesting new types of data are being collected and primarily stored in HDFS—for example, data from Internet of Things (IoT) devices. **The true value of that data is locked up in big data systems and can realistically only be analyzed by big data engineers and data scientists.** To get the value out of big data, data scientists typically export high-value data out of the enterprise database and import it into Hadoop, so that they can join it with the new data streams. Without the context that the dimensional high-value data stored in the enterprise database provides, high-volume big data doesn't have much value; big data streams need to be joined to data like customers, accounts, parts, products, marketing campaigns, and so on. SQL Server Big Data Clusters make it easy for big data sets to be joined to the dimensional and fact data typically stored in the enterprise database and make it easy for people and apps that use SQL Server to query big data. The value of the big data greatly increases when it is not just in the hands of the data scientists and big data engineers but is also included in

reports, dashboards, and applications. At the same time, the data scientists can continue to use Hadoop ecosystem tools and have easy, real time access to the high-value data in SQL Server.

With SQL Server Big Data Clusters , existing customers can do more with their enterprise data lakes. SQL developers and analysts can:

- Build applications consuming enterprise data lakes.
- Reason over all data with Transact-SQL queries.
- Use the existing ecosystem of SQL Server tools and applications to access and analyze enterprise data.
- Reduce the need for data movement through data virtualization and data marts.

Big data engineers and data scientists can:

- Continue to use Spark for big data scenarios.
- Build intelligent enterprise applications using:
  - Spark to train models over data lakes.
  - Operationalize models in SQL Server.
- Stream data directly into Enterprise data marts for real-time analytics.
- Explore data visually using interactive analysis and BI tools.

SQL Server Big Data Clusters take the form of a compute cluster of containers running SQL Server and big data services. Tasks can be parallelized across different SQL Server instances, whether they are running in the compute, storage or data pools, or Spark executors running inside containers in the cluster.

Cluster activities are marshalled by a SQL Server master instance; tasks are parallelized across Docker containers running on a scalable group of nodes that make up a compute tier. Big Data Clusters are built on SQL Server on Linux in Docker containers in concert with open-source container management and big data tools, including Kubernetes, HDFS, and Spark.

## Database engine enhancements

Enhancements in the SQL Server 2019 database engine are not limited to PolyBase and Big Data Clusters; many components of the database engine have new features and capabilities with this release.

### Performance and scale

The SQL Server 2019 database engine includes performance and scale improvements in diverse areas of functionality. Support for Persistent Memory (PMEM) devices is improved in this release. Any SQL Server file that is placed on a PMEM device operates in *enlightened* mode. SQL Server directly accesses the device, bypassing the storage stack of the operating system. PMEM is also known as storage class memory (SCM).

Also included are several features to improve the performance of columnstore indexes, such as automated columnstore index maintenance, better columnstore metadata memory management, a low-memory load path for columnstore tables, and improved performance for bulk loading to columnstore indexes. The server startup process has been made faster for databases that use in-memory columnstore tables for hybrid transactional and analytical processing (HTAP).

SQL Server 2019 also includes a database-scoped setting for online and resumable DDL operations, as well as support for resumable online index creation.

### High availability

SQL Server 2019 adds support for even more high availability scenarios and platforms, including:

**Enable high availability configurations for SQL Server running in containers**—SQL Server 2019 enables customers to configure highly-available systems with Always On Availability Groups using Kubernetes as an orchestration layer.

**Up to five synchronous replica pairs** – SQL Server 2019 increases the limit for synchronous replica pairs from three (in SQL Server 2017) to five. Users can now configure up to five synchronous replicas (1 Primary and up to 4 secondary replicas) with automatic failover between these replicas.

**Better scale-out with automatic redirection of connections based on read/write intent**—Configuring an Always On Availability can be challenging for a number of reasons, including:

- In SQL Server 2017, an administrator must configure the Availability Group listener (and the corresponding cluster resource) to direct SQL Server traffic to the primary replica to ensure that clients are transparently reconnected to the active primary node upon failover; however, there are cluster technologies that support SQL Server Availability Groups that do not offer a listener-like capability.
- In a multi-subnet configuration such as Azure or multi-subnet floating IP address in an availability group using Pacemaker, configurations become complex, prone to errors and difficult to troubleshoot due to multiple components involved.
- When the availability group is configured for read scale-out or DR and cluster type is NONE, there is no straightforward mechanism to ensure transparent reconnection upon manual failover.

To address these challenges, SQL Server 2019 adds a new feature for Availability Groups: secondary to primary replica connection redirection. With this feature, client applications can connect to any of the replicas of the Availability Group and the connection will be redirected to the primary replica, according to the Availability Group configuration and the connection intent (read only or read/write) specified in the connection string.

**High availability with remote storage on Kubernetes**—an option for high availability is to use a container orchestrator such as Kubernetes, and deploy SQL Server in a configuration that is like a shared disk failover cluster instance but uses the container orchestrator capabilities instead of a failover cluster. This deployment model does not require specific enhancements from SQL Server perspective, and is not new capability in SQL Server 2019—a tutorial is available on how to implement this configuration using SQL Server 2017<sup>5</sup>—but SQL Server 2019 enables enhanced instance health check monitoring using the same operator pattern as Availability Group health checks.

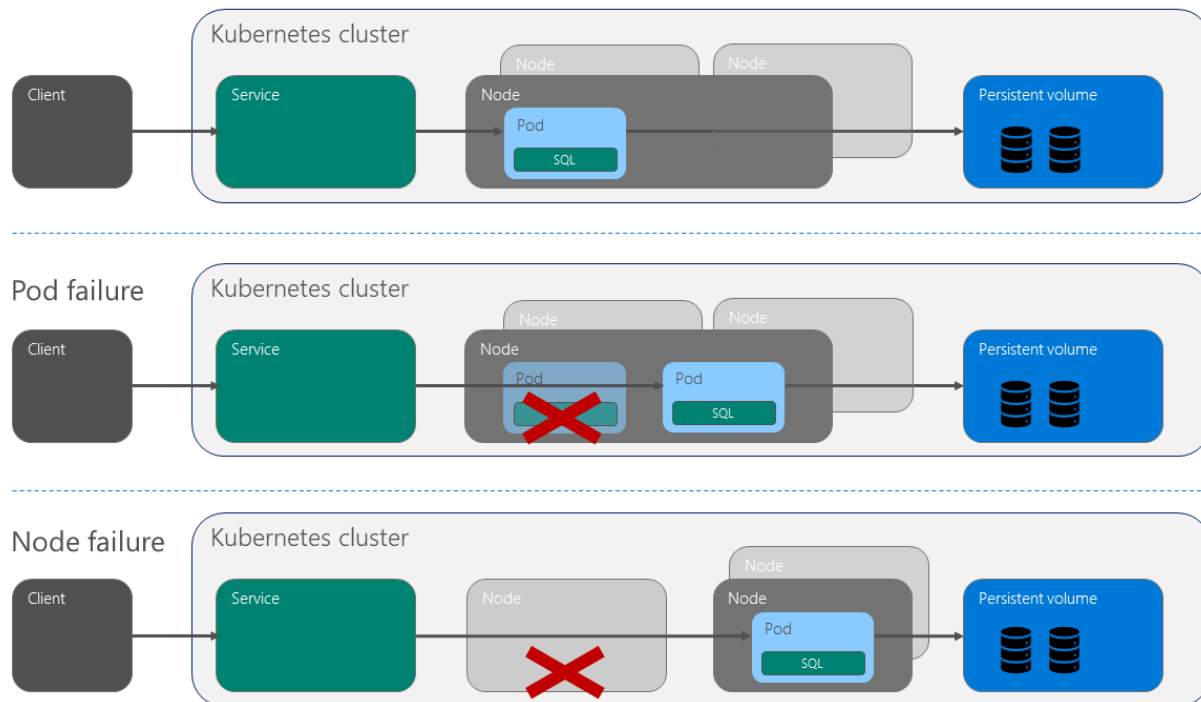


Figure 5: Failover with containers and Kubernetes

## Security and compliance

SQL Server 2019 includes several enhancements that make it easier for database administrators and information protection officers to secure sensitive data and comply with data protection policies and legislation.

**SQL data discovery and classification**—added in SQL Server Management Studio (SSMS) 17.5, SQL Data Discovery and Classification allows you to classify columns in your database that contain sensitive information. You can classify columns by the type of information they contain—names, addresses, social security numbers, and so on—and by the level of sensitivity of the data in the column—including levels such as *public*, *general*, *confidential*, and *restricted*. You can easily generate reports from the classification you have applied to enable

<sup>5</sup> <https://docs.microsoft.com/en-us/sql/linux/tutorial-sql-server-containers-kubernetes?view=sql-server-linux-2017>

you to meet statutory and regulatory requirements, such as EU GDPR. SSMS also includes the SQL Data Discovery and Classification wizard, which will try to identify columns in your database that could contain sensitive information. SQL Data Discovery and Classification uses the underlying mechanism of SQL Server Extended Properties, and so is backwards-compatible with SQL Server 2008 and later.

**Always Encrypted with Secure Enclaves**—Introduced in SQL Server 2016, Always Encrypted is an encryption technology that protects the confidentiality of sensitive data from malware and high-privileged, but unauthorized users of SQL Server, including DBAs, machine admins and cloud admins. Sensitive data is never visible in plaintext to those users. If a DBA, who does not have access to the keys, queries the encrypted database columns, the query will either fail or return encrypted data, depending on the database connection properties. If a user or a process with machine admin privileges scans the memory of the SQL Server process, they will not be able to see the data in plaintext.

In SQL Server versions prior to SQL Server 2019, Always Encrypted achieves the above security benefits by encrypting data on the client side and never allowing the data or the corresponding cryptographic keys to appear in plaintext inside the SQL Server process. As a result, the functionality on encrypted database columns inside the database is severely restricted: the only operation the SQL Server Engine can perform on encrypted data is equality comparison (and it is only available with deterministic encryption). All other operations, including cryptographic operations, or rich computations, such as pattern matching, are not supported inside the database and must be carried out in the application layer.

The secure enclave technology, introduced in SQL Server 2019, addresses the limitations of Always Encrypted by allowing computations on plaintext data within a secure enclave inside the SQL Server process. A secure enclave is a protected region of memory within the SQL Server process, and it acts as a trusted execution environment for processing sensitive data inside the SQL Server Engine. A secure enclave extends client applications' trust boundary to the server side. While it is contained by the SQL Server environment, the secure enclave is not accessible to SQL Server, the operating system, or to the database or system administrators.

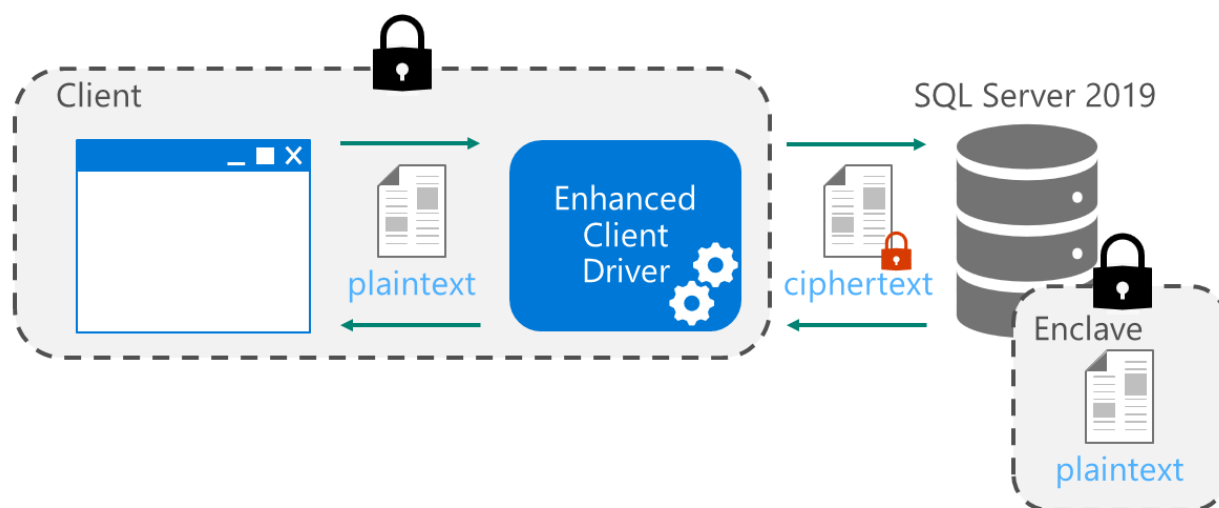


Figure 6: Always Encrypted with Secure Enclaves

In SQL Server 2019, Always Encrypted with secure enclaves uses Virtualization-Based security (VBS) secure memory enclaves (also known as Virtual Secure Mode, or VSM, enclaves) in Windows Server 2019 and Windows 10 (build 17704 or later).

**Vulnerability assessment**—you can use [vulnerability assessment to track compliance of your SQL Server instances and Azure SQL Database instances with recognized security best practices](#). For on-premises SQL Server instances, you can access vulnerability assessments through SSMS (the feature was added in SSMS 17.4). Vulnerability assessment is also available for Azure SQL Database instances through the Azure Portal. The scanning process compares the configuration of your database with a list of security best-practices maintained by Microsoft; at the end of the scan, the tool generates a report of your security state, and details of any security issues found, the severity of each issue, and remediation steps—no changes are made to your database. You can choose whether to implement the generated recommendations on your database. Vulnerability assessment gives you a simple way to proactively monitor and improve your database security posture, and to better comply with data protection regulations such as EU GDPR.

**Certificate Management functionality in SQL Server Configuration Manager**—SSL/TLS certificates are widely used to secure access to SQL Server. With earlier versions of SQL Server, organizations with large SQL Server estates have needed to expend considerable effort to maintain their SQL Server certificate infrastructure—often through developing scripts and running manual commands. With SQL Server 2019, certificate management is integrated into the SQL Server Configuration Manager UI, simplifying common tasks such as:

- Viewing and validating certificates installed in a SQL Server instance.
- Report on certificates close to expiration.
- Deploy certificates across machines participating in Always On Availability Groups (from the node holding the primary replica).
- Deploy certificates across machines participating in a failover cluster instance (from the active node).

The user executing these tasks requires administration privileges in all the affected nodes.

## UTF-8 support

SQL Server 2019 includes full support for the widely used UTF-8 character encoding as an import or export encoding, or as database-level or column-level collation for text data.

UTF-8 is allowed in the CHAR and VARCHAR datatypes, and is enabled when creating or changing an object's collation, to a collation with the "UTF8" suffix, such as LATIN1\_GENERAL\_100\_CI\_AS\_SC to [LATIN1\\_GENERAL\\_100\\_CI\\_AS\\_SC\\_UTF8](#). UTF-8 is only available to windows collations that support supplementary characters, as introduced in SQL Server 2012. Note that NCHAR and NVARCHAR allow UTF-16 encoding only, and remain unchanged.

Significant storage savings can also be achieved, depending on the character set in use. For example, changing an existing column data type from NCHAR(10) using UTF-16 to CHAR(10) using an UTF-8 enabled collation, translates into nearly 50% reduction in storage requirements. This is because NCHAR(10) requires 22 bytes for storage, whereas CHAR(10) requires 12 bytes for the same Unicode string.

## SQL Server on Linux

In SQL Server 2019, SQL Server on Linux is closer to feature parity with SQL Server on Windows. Features added to SQL Server on Linux include:

**Replication**—in SQL Server 2019, SQL Server on Linux instances can participate in transactional, merge, and snapshot replication topologies in the publisher, distributor, or subscriber roles.

**Active Directory integration**—integration between SQL Server on Linux and Active Directory is much improved. SQL Server on Linux instances can be configured use Active Directory for authentication of users, for replication, and for distributed queries. SQL Server on Linux instances can now participate in Availability Groups that are authenticated by Active Directory, in addition to the support for certificate-based authentication included in SQL Server 2017 on Linux. As an alternative to Active Directory, SQL Server on Linux instances can now use OpenLDAP as a directory provider, making it easier to manage domain-joined Linux servers.

**Distributed transactions**—SQL Server on Linux instances can now initiate and participate in distributed transactions. This is achieved with a Linux version of the **Microsoft Distributed Transaction Coordinator (MSDTC)** that runs within the SQL Server process to help SQL Server participate in distributed transactions. With access to MSDTC, SQL Server on Linux can participate in distributed transactions with other third-party transaction coordinators, or if you have specific processing needs there is support for you to develop your own.

## Containers

**New master container registry**—the primary publishing location for SQL Server container images is a new container registry—the **Microsoft Container Registry at [mcr.microsoft.com](https://mcr.microsoft.com)**. Microsoft Container Registry is the official container registry for the distribution of Microsoft product containers. In addition, certified RHEL-based images are now published.

## Machine learning

There are many advantages to bringing the machine learning compute to the data instead of moving data out to compute. These advantages include the elimination of data movement, ease of deployment, improved security and better scale and performance. These advantages also make SQL Server a powerful end to end machine learning platform. Enhancements to Machine Learning in SQL Server 2019 CTP 2.0 include:

**Machine Learning on Linux**—SQL Server 2019 Machine Learning Services (In-Database) is now supported on Linux.

**Input data partitioning**—Without changing your R or Python scripts, you can process data at table partition level. This allows you to train a model for each table partition and parallelize model training per partition.

**Failover cluster support**—You can install SQL Server 2019 Machine Learning Services (In-Database) on a Windows failover cluster to meet your requirements for redundancy and uptime in the event your primary server fails over.

**Java language extension**—In addition to R and Python runtimes, SQL Server 2019 adds a Java language extension. This will allow you to call a pre-compiled Java program and securely execute Java code on SQL Server. This reduces the need to move data and improves application performance by bringing your workloads closer to your data. You specify the Java runtime you want to use, by installing the JDK distribution and Java version of your choice.

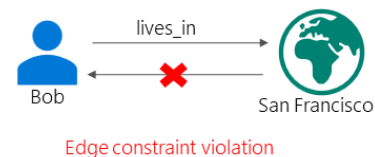
## SQL Graph

Introduced in SQL Server 2017, SQL Graph provides graph database capabilities to SQL Server, allowing easier modelling of data sets with many—often complex—relationships as a set of nodes (or vertexes) and edges (or

relationships). To make SQL Graph more powerful and easier to work with, SQL Server 2019 CTP 2.0 adds the following features:

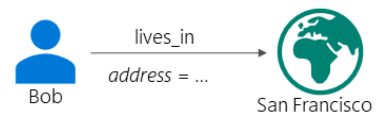
**Edge constraints**—In the first release of SQL Graph, an edge could connect any node to any other node in the database. Edge constraints help users put some restrictions on this behavior. Using the new **CONNECTION** constraint, you can now limit the type of nodes a given edge type can connect. This helps users enforce special semantics on the edge table and also maintain data integrity in their graph database.

In this example, an edge constraint allows a *lives\_in* edge to connect a Person node to city node. The reverse relationship is not permitted because it violates the constraint.



**MATCH support Merge DML**—The MERGE statement performs insert, update, or delete operations on a target table based on the results of a join with a source table. For example, you can synchronize two tables by inserting, updating, or deleting rows in a target table based on differences between the target table and the source table. Using MATCH predicates in a MERGE statement is now supported in SQL Server. That is, it is now possible to merge your current graph data (node or edge tables) with new data using the MATCH predicates to specify graph relationships in a single statement, instead of separate INSERT/UPDATE/DELETE statements.

In this example, Bob lives in San Francisco. A MERGE statement can be used to insert a new *lives\_in* edge between Bob and San Francisco, if it does not already exist in the graph database. If the edge already exists, the merge statement will update the address property of the edge with Bob's new street address.



## Intelligent database and query processing

**Batch mode on row store**—Batch mode for columnstore was introduced in SQL Server 2017; it allows query operators to process data more efficiently by working on a batch of rows at a time—instead of one row at a time—but only for tables with columnstore indexes. SQL Server 2019 supports batch mode for row store tables; the same batch mode performance enhancements are available for tables without a columnstore index. This feature provides high speed relational data warehouse performance for scenarios where creating a columnstore index adds too much overhead to a transactional workload. Several other scalability improvements are tied to batch-mode processing—and now many more workloads can benefit without any changes required to the database code.

**Row mode memory grant feedback**—Addressing repeating workloads, memory grant feedback recalculates the actual memory required for a query and then updates the grant value for the cached plan. When an identical query statement is executed, the query uses the revised memory grant size, reducing excessive memory grants that impact concurrency and fixing underestimated memory grants that cause expensive spills to disk. In SQL Server 2017, memory grant feedback was only available in batch mode—for tables with a columnstore index; in SQL Server 2019, memory grant feedback is available in both batch mode and row mode.

**Table variable deferred compilation**—to improve the performance of workloads that use table variables, the SQL Server 2019 database engine delays the first compilation of statements that reference one or more table



variables, so that the actual cardinality of the table variables can be used to generate the query plan. This results in improved plan quality and performance.

**Approximate COUNT DISTINCT**—in some data-warehousing and big data scenarios, the cost of aggregating a distinct count of the values in a column on a very large data set can be prohibitive, especially when the precision of the answer is not critical. SQL Server 2019 provides an **APPROX\_COUNT\_DISTINCT** aggregate operator for these scenarios, giving the benefit of high performance and a low memory footprint.

## Troubleshooting and diagnostics

**Improve truncation message for ETL DW scenarios**—the error message **ID 8152 String or binary data would be truncated** is familiar to many SQL Server developers and administrators who develop or maintain data movement workloads; the error is raised during data transfers between a source and a destination with different schemas when the source data is too large to fit into the destination data type. This error message can be time-consuming to troubleshoot because of its lack of specificity. SQL Server 2019 introduces a new, more specific error message for this scenario: **ID 2628 String or binary data would be truncated in table '%.\*s', column '%.\*s'. Truncated value: '%.\*s'**. The new error message provides more context for the problem, simplifying the troubleshooting process. So that it cannot break existing applications parsing message ID 8152, this new message ID 2628 is an opt-in replacement, which can be enabled with trace flag 460.

**Lightweight query profiling infrastructure enabled by default**—the lightweight query profiling infrastructure offers a query execution statistics collection mechanism with a maximum overhead of 2% CPU, compared with an overhead of up to 75% CPU for the legacy collection mechanism. The lightweight query profiling infrastructure was introduced in SQL Server 2016 SP1, but it was not enabled by default; database administrators could enable it with a trace flag. In SQL Server 2019, the lightweight query profiling infrastructure is enabled by default.

**Additional support to troubleshoot columnstore indexes**—columnstore statistics are now included in **DBCC CLONEDATABASE** to enable performance troubleshooting without the need to manually capture the statistics information. The ability to see the large space savings possible when using the highly efficient compressed columnstore indexes has been added to **sp\_estimate\_data\_compression\_savings**. In SQL Server 2019 both **COLUMNSTORE** and **COLUMNSTORE\_ARCHIVE** have been added to allow you to estimate the space savings if either of these indexes are used on a table.

**Enhanced database level health detection for Availability Group databases**—database level health detection for Availability Group databases has been improved to detect additional error conditions related to the availability of a database and initiate a failover (or restart) of the Availability Group resource if required.

**Improved information to investigate page-related waits**—a new dynamic management function **sys.dm\_db\_page\_info** returns information about the header from a page and when used in conjunction with **sys.dm\_exec\_requests** and **sys.sysprocesses** it is possible to gain greater insight into why a page is causing performance issues.

# Business Intelligence

## Reporting Services

SQL Server Reporting Services (SSRS) provides a modern, on-premises solution for creating, deploying, and managing reports within your organization. Since SQL Server 2016, Reporting Services brings modern design to enterprise reporting. You can create mobile reports optimized for smartphones and tablets to bring the power of data to more users in the organization, in addition to creating modern paginated reports. The native [Power BI apps](#) for mobile devices for Windows 10, iOS, Android, and HTML5 allow easy access to mobile reports and dashboards both in online and offline mode.

This server-based reporting platform includes a complete set of tools to create, manage, and deliver reports and APIs that enable developers to integrate or extend data and report processing in custom applications. These tools work within the Microsoft Visual Studio environment and are fully integrated with SQL Server tools and components. Report Builder enables report authors to design paginated reports and publish them to a report server. SQL Server Mobile Report Publisher enables report authors to design mobile reports and publish them to a report server. SQL Server Data Tools (SSDT) integrates into Visual Studio 2017 and enables developers to design paginated reports within the Visual Studio integrated development environment and take advantage of projects, source control, build, and deployment.

SSRS provides a responsive web portal built on HTML5 that enables users to browse, search, view, and manage reports (both paginated and mobile) using a modern browser to access all your reports in one place. A scheduling and delivery agent refreshes data sets and executes reports on a schedule, and delivers paginated reports to users via email and other means. The report server database, built on the SQL Server Database Engine, stores and manages the catalog of content, including data sources, data sets, paginated reports, mobile reports, and KPIs. The database can be on the report server or on a different server running SQL Server.

Reporting Services supports both traditional paginated reports, in addition to mobile reports and dashboards.

For more information about the capabilities of Reporting Services, see [Reporting Services](#). Note that Reporting Services is currently a Windows-only feature.

## Power BI Report Server

Power BI is a collection of services and features that enable your organization to share, visualize, and analyze data in a collaborative self-service fashion. Power BI can connect to a mix of on-premises and online data sources, which can be automatically refreshed on a fixed schedule. You can also use Power BI to showcase your existing on-premises reports and data models.

With Power BI Desktop, users can self-serve to design reports, dashboards, and data models that connect to a broad range of data sources. For more information about the capabilities of Power BI Desktop, see the [Power BI Desktop product page](#).

With the Power BI service, users can store and access Power BI reports from the Microsoft public cloud, using a browser or mobile devices. The Power BI Premium service allows users to share and collaborate on Power BI reports. For more information about the capabilities of Power BI, see the [Power BI product page](#).

Power BI Report Server builds on SQL Server 2017 Reporting Services to allow reports designed in Power BI Desktop to be deployed on an on-premises server, rather than from the Power BI cloud service. Power BI Report Server allows organizations with data protection policies that would otherwise bar them from using the public cloud Power BI service to reap the benefits of Power BI reports. A subscription to the Power BI Premium service incorporates a license to Power BI Report Server, allowing organizations to start building on-premises Power BI reports that can later be migrated into the cloud.

For more information about the capabilities of Power BI Report Server, see the [Power BI Report Server product page](#). Note that Power BI Report Server is currently a Windows-only feature.

## Analysis Services

SQL Server Analysis Services (SSAS) boasts modern data connectivity and transformation capabilities, with support for Power BI data sources. SSAS also supports some advanced BI modeling capabilities, such as data mashup transformations, drill-throughs, and ragged hierarchies. SSAS can be configured to work with in-memory tabular models, or traditional multidimensional OLAP cubes.

SQL Server Analysis Services provides several approaches for creating a business intelligence semantic model: Tabular, Multidimensional (OLAP cubes), and Power Pivot for SharePoint.

Having more than one approach enables a modeling experience tailored to different business and user requirements. Multidimensional is a mature technology built on open standards—embraced by numerous vendors of BI software—but it can be hard to master. Tabular offers a relational modeling approach that many developers find more intuitive. Power Pivot is even simpler, offering visual data modeling in Excel, with server support provided via SharePoint. All models are deployed as databases that run on an Analysis Services instance, accessed by client tools using a single set of data providers, and visualized in interactive and static reports via Excel, Reporting Services, Power BI, and BI tools from other vendors.

Tabular and multidimensional solutions are built using SQL Server Data Tools (SSDT) and are intended for corporate BI projects that run on a standalone Analysis Services instance on-premises, and for tabular models, an Azure Analysis Services server in the cloud. Both solutions yield high performance analytical databases that integrate easily with BI clients.

Tabular and multidimensional models use imported data from external sources. The amount and type of data you need to import can be a primary consideration when deciding which model type best fits your data. Both tabular and multidimensional solutions use data compression that reduces the size of the Analysis Services database relative to the data warehouse from which you are importing data. Because actual compression will vary based on the characteristics of the underlying data, there is no way to know precisely how much disk and memory will be required by a solution after data is processed and used in queries.

Tabular databases run either in-memory or in DirectQuery mode that offloads query execution to an external database. For tabular in-memory analytics, the database is stored entirely in memory, which means you must have sufficient memory to not only load all the data, but also to create additional data structures to support queries. DirectQuery, revamped in SQL Server 2016, has fewer restrictions than before, and better performance. Taking advantage of the backend relational database for storage and query execution makes building a large-scale Tabular model more feasible than was previously possible. For multidimensional offloading, data storage and query execution are available via ROLAP. On a query server, rowsets can be cached, and stale ones paged out.

For more information about the capabilities of Analysis Services, see [Analysis Services](#). Note that Analysis Services is currently a Windows-only feature.

*Note: SQL Server 2019 CTP 2.0 does not introduce new features for Business Intelligence.*

# Enterprise Information Management

## SQL Server Integration Services

A feature since SQL Server 2005, SQL Server Integration Services (SSIS) is a platform for building enterprise-level data integration and data transformations solutions. You use Integration Services to solve complex business problems by copying or downloading files, sending e-mail messages in response to events, updating data warehouses, cleaning and mining data, and managing SQL Server objects and data. The packages can work alone or in concert with other packages to address complex business needs. Integration Services can extract and transform data from a wide variety of sources such as XML data files, flat files, and relational data sources, and then load the data into one or more destinations.

Integration Services includes a rich set of built-in tasks and transformations; tools for constructing packages; and the Integration Services service for running and managing packages. You can use the graphical Integration Services tools to create solutions without writing a single line of code; or you can program the extensive Integration Services object model to create packages programmatically and code custom tasks and other package objects.

Improvements in SQL Server 2019 to scaling out SSIS packages can be installed side-by-side with the SQL Server 2017 version. If you prefer you can also choose to upgrade scale out management, masters, and workers in place.

For more information about the capabilities and features of SSIS, see [SQL Server Integration Services](#). Note that the SSIS engine runs on Windows and Linux.

## Master Data Services

A feature since SQL Server 2008, Master Data Services enables you to manage a master set of your organization's data. You can organize the data into models, create rules for updating the data, and control who updates the data. With an Excel add-in and a web application, you can share the master data set with other people in your organization. The Master Data Manager (MDM) web application had a dependency on Silverlight. All the former Silverlight components have been replaced with HTML controls in SQL Server 2019.

In Master Data Services, the model is the highest-level container in the structure of your master data. You create a model to manage groups of similar data—for example, to manage online product data. A model contains one or more entities, and entities contain members that are the data records. For example, your online product model might contain entities such as product, color, and style. The color entity might contain members for the colors red, silver, and black.

For more information about the capabilities of Data Quality Services, see [Master Data Services Overview](#). Note that Master Data Services is currently a Windows-only feature.

## SQL Server 2019 tooling

SQL Server 2019 is supported by enhancements and additions to accompanying free tools. Reaching generally availability at Microsoft Ignite 2018, Azure Data Studio<sup>7</sup> is a new, open source, cross-platform (Windows, Linux, macOS) desktop application that for administering and querying SQL Server instances running on any platform, Azure SQL Data Warehouse, and Azure SQL Database instances. With Azure Data Studio, you can run queries, carry out administration tasks, and create customized dashboards to monitor the state of your SQL Server instances. Azure Data Studio is based on the same platform as Visual Studio Code, and includes Git integration and extensibility out of the box; several Microsoft, 3<sup>rd</sup> party, and community extensions are already available. Database Administrators and data scientists can interact with SQL Server Big Data Clusters using the Scale-out Data Management extension and Jupyter Notebooks. Find out more about Azure Data Studio on <http://aka.ms/azuredatstudio>

SQL Server Management Studio is the free Windows-only desktop application for administering and querying SQL Server instances running on any platform, Azure SQL Data Warehouse, and Azure SQL Database instances. Since 2016, SQL Server Management Studio has its own release cycle independent of SQL Server, allowing more frequent updates. Find out more about SQL Server Management Studio on [Microsoft Docs](#).

Visual Studio Code is a free, lightweight, extensible cross-platform IDE. With the mssql extension, you can use Visual Studio Code to execute Transact-SQL scripts on SQL Server instances running on any platform, Azure SQL Data Warehouse, and Azure SQL Database instances. Find out more about Visual Studio Code at <https://code.visualstudio.com/>. Find out more about working with the mssql Visual Studio Code extension on [Microsoft Docs](#).

Several command-line/shell tools are available for interacting with SQL Server, including:

- **sqlcmd**—used to execute Transact-SQL commands from the command-line, either dynamically or from a file.
- **bcp**—used to bulk-import and bulk-export data to SQL Server instances.
- **sqlpackage**—used to script databases from and deploy databases to dacpac packages, to simplify database deployment and migration.

Cross-platform versions of sqlcmd and bcp are already available. A cross-platform release of sqlpackage is currently in preview.

**mssql-cli** is a new, cross-platform, command-line/shell tool that aims to provide a more user-friendly shell environment for interacting with SQL Server; mssql-cli improves upon sqlcmd by adding modern features such as **syntax highlighting, auto-completion, and adaptive result-set formatting**. mssql-cli is built on the open-source dbcli project and is dependent on the Python language and the Python package management tool pip. Find out more about mssql-cli at the project page [on GitHub](#).

**SQL Server Data Tools** is a workload customization for Visual Studio that you use to develop, test, and deploy SQL Server database engine, Integration Services, Analysis Services, and Reporting Services tabular objects. SQL Server Data Tools is compatible with all versions of Visual Studio 2017, including Community Edition. Find out more about SQL Server data tools on [Microsoft Docs](#).

---

<sup>7</sup> Azure Data Studio was released in preview under the name SQL Operations Studio.

## Conclusion

SQL Server continues to evolve in SQL Server 2019; new features and new technology make SQL Server an even more compelling tool for the enterprise. Developers, analysts, data scientists, and database administrators can be even more productive than ever before with new features across the database engine. SQL Server 2019 Big Data Clusters make a completely new use-case for SQL Server, making it easy to bring high-value relational data and high-volume big data together on a single scalable platform. With a greater range of supported platforms and usage scenarios, including SQL Server containers on Kubernetes and Service Fabric, organizations have even more flexibility when designing their data ecosystems.

## Calls to action

For more information about SQL Server 2019, see: [aka.ms/ss19](https://aka.ms/ss19)

For more information about Azure Data Studio, see: [aka.ms/azuredatstudio](https://aka.ms/azuredatstudio)

For SQL documentation, see: <https://docs.microsoft.com/en-us/sql/sql-hub-menu>

