



Microsoft SQL Server 2017

Technical white paper



Published: September 2017

Applies to: Microsoft SQL Server 2017 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. 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.

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.

© 2017 Microsoft Corporation. All rights reserved.

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.

Contents

Summary	5
Industry landscape and trends.....	5
SQL Server 2017: the next step forward	7
The power of SQL Server now on Linux and containers.....	7
Native Linux experience with support for existing SQL Server tools	8
Platform abstraction with containers.....	8
Enabling DevOps practices with containers.....	9
SQL Server 2017 in operational usage for relational and graph data	9
In-Memory Online Transaction Processing	9
In-Memory OLTP enhancements in SQL Server 2017	11
Real-time operational analytics with hybrid transactional-analytical processing	11
Adaptive Query Processing feature family	12
Automatic plan correction	12
Resumable Online Index Rebuild	13
SQL Graph.....	13
Resource Governor.....	15
Mission-critical high availability on any platform	16
Instance-level high availability with Always On Failover Cluster Instances	16
Database-level high availability with Always On Availability Groups	17
Flexibility for HA architectures.....	17
Always On Availability Groups for Linux.....	17
Load balancing of readable secondary replicas	18
SQL Server 2017 in Data Warehousing and Big Data.....	19
Petabyte-scale data warehousing	19
Query over any type of data.....	20
Clustered columnstore indexes for data warehousing	21
Table partitioning	21
SQL Server 2017 in Business Intelligence.....	23
Reporting Services.....	23

Power BI Report Server.....	24
Analysis Services.....	25
SQL Server 2017 in Advanced Analytics	28
SQL Server 2017 in Data Integration and Information Management.....	29
SQL Server Integration Services (SSIS)	29
Master Data Services	29
Data Quality Services.....	30
SQL Server 2017 Security.....	31
Always Encrypted	31
Transparent Data Encryption.....	31
Row-Level Security	32
Dynamic data masking	32
Auditing	32
SQL Server 2017 Cloud.....	33
Backup to Azure	33
SQL Server 2017 in Azure VMs.....	34
Deploy SQL Server by operational need, not feature set.....	35
Cross-box scale limits.....	35
Conclusion	35
Calls to action.....	36

Summary

The IT landscape is constantly changing and increasingly diversifying, particularly operating systems, including Linux. A massive data growth to 50 ZB is projected by 2020. Customers are investing in data, and this creates new opportunities if they can modernize their applications with built-in analytics to drive digital transformation. Customers have widely available unstructured data, and they want to deliver powerful insights through big data. Today's organizations must adapt quickly to change by using new technologies that fuel competitive advantage, or risk getting left behind. That's why it's imperative to make the most of big data, the cloud, and intelligence capabilities, all of which help companies accelerate their speed of business through smarter decision-making and faster execution.

The complete modern data estate can utilize both structured data—such as OLTP, mobile, ERP, and LOB data—and unstructured data—such as graph data, social media, and IoT data. Customers need to be able to handle both operational and data warehouse data, and process big data. They need insights both to look back retrospectively (business intelligence) and to make predictions about the future using machine learning algorithms (advanced analytics with predictive and prescriptive analysis). Their data estate needs to secure data, access, and applications. They also need the flexibility of using the platform, development language, and location of their choice, in both the private and public cloud.

Microsoft SQL Server 2017 powers your entire data estate by supporting structured and unstructured data sources. It builds on previous versions of SQL Server, which have been industry leading for four years in a row and a leader in TPC-E. It scales to petabytes of data and allows customers to process big data through PolyBase using T-SQL over any data. SQL Server has also been the least vulnerable database during the last seven years. SQL Server 2017 brings data insights with business intelligence capabilities that provide analytics at a fraction of the cost on any device—along with advanced analytics with support for R and Python.

Application developers can build their applications using any language, including Node.JS, .NET, Java, and PHP, and deploy their solutions on platforms such as Windows, Linux, and Docker containers—all in a private cloud (on-premises), Microsoft Azure, third-party clouds, or a hybrid environment.

Industry landscape and trends

In addition to negotiating a constantly changing landscape, IT organizations must contend with multiple data types, different development languages, and a mix of on-premises/cloud/hybrid environments. Maintaining increasingly complex environments is a daunting task that requires people and processes to keep the pace and somehow simultaneously reduce operational costs.

Microsoft is committed to enabling customers to choose the best platform for their data and applications. This includes providing interoperability with open-source solutions. For example, Linux distributions like Red Hat Enterprise Linux, Ubuntu, and SUSE are becoming more popular for relational and nonrelational workloads.

Customers are also using open-source technologies like Hadoop, Spark, R, and Python to solve data management and analytical challenges. With Microsoft, they can choose the best platform for their solutions. Interoperability with open-source solutions is provided, and solutions like HDInsight (a fully managed cloud Apache Hadoop offering in Azure) and Microsoft Machine Learning Server can be deployed on both Windows and Linux.

The Microsoft commitment extends into application development, including .NET core as an open-source project and the newly announced Visual Studio Code extension for SQL Server. Similarly, SQL Server drivers, connectivity tools, and APIs are available for nearly any environment, enabling any application to integrate with SQL Server, regardless of programming language or environment.

SQL Server 2017: the next step forward

SQL Server 2017 builds on the industry-leading¹ capabilities of SQL Server 2016, holding benchmarks in such areas as:

- **Performance**—SQL Server owns the top TPC-E performance benchmarks for transaction processing, the top TPC-H performance benchmarks for data warehousing, and the top performance benchmarks with leading business applications². In April 2017, Hewlett Packard Enterprise published a new TPC-H 1TB world record with SQL Server 2017 running on Linux³.
- **Security**—According to the National Institute of Standards and Technology (NIST) public security board, SQL Server has the lowest number of reported security vulnerabilities across the major database vendors (NIST, 2016).
- **Total cost of ownership**—SQL Server has a significantly lower total cost of ownership (TCO) than similar enterprise data solutions. In some cases, the TCO for SQL Server 2016 was found to be as low as 1/12th the cost of comparable products/features.

SQL Server 2017 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. The following scenarios represent potential uses for SQL Server 2017, in addition to the latest features.

The power of SQL Server now on Linux and containers

SQL Server 2017 brings the industry-leading Microsoft relational database engine to the enterprise Linux ecosystem. This includes SQL Server Agent, Active Directory authentication, best-in-class high availability/disaster recovery, and unparalleled data security features. It's important to note that SQL Server 2017 on Linux is not a port or rewrite. This is the world-class Microsoft RDBMS now available on more operating systems—like Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu—and more cloud and container platforms like Docker.

SQL Server 2017 offers the best performance and security features, and they're now available on all supported platforms, including Windows, Linux, and containers. High-performance features bring record-breaking speed to data-driven applications. These features include columnstore (which provides column-based data storage and processing to achieve up to 10 times the query performance and data compression of row-based storage) and in-

¹ Gartner has rated Microsoft as a leader with the most complete vision and highest ability to execute of any operational database management system for two consecutive years. *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.*

² Lenovo has announced the TPC-H 10 TB benchmark world record (Lenovo Server Performance Lab, 2016) using SQL Server 2016. In May 2016, Lenovo also published a new TPC-H 30 TB world record (Lenovo Server Performance Lab, 2016).

³ www.tpc.org, "TPC-H Result Highlights HPE Proliant DL380 Gen9", March 2017 ([link](#))

memory OLTP (which brings transaction processing to memory-optimized tables at more than 2.5 times the speed of disk-based tables). Security features like Auditing, Transparent Data Encryption, Row-Level Security, Dynamic Data Masking and Always Encrypted provide server-side security measures that vastly simplify the process for keeping data safe from unauthorized access, without the need to modify existing client applications. Auditing enables teams to monitor access and track potentially suspicious activity. Transparent Data Encryption protects data at rest at a file level, while Always Encrypted secures data in motion and at rest. With these capabilities available on all SQL Server editions, organizations can choose their deployment environment by operational need, as opposed to desired features.

SQL Server 2017 on Linux is not a rewrite or a port; SQL Server on Windows and on Linux share a common code base that accesses low-level operating system functions through a platform abstraction layer. While not all the scenarios and features covered in this white paper are supported on Linux yet, SQL Server 2017 on Linux is ready to support transactional and data warehousing workloads, in addition to participating in Availability Groups. The majority of Database Engine workloads can be moved from Windows to Linux without modification. Microsoft offers tools such as Data Migration Assistant to assist with moving existing workloads to SQL Server 2017. For more information about SQL Server features not currently supported on Linux, see the [SQL Server on Linux release notes](#).

Native Linux experience with support for existing SQL Server tools

Microsoft has focused on providing a Linux-native user experience for SQL Server, starting with the installation process. SQL Server 2017 uses the standard package-based installation method for Linux using yum for Fedora-based distributions and apt-get for Debian-based distributions. Administrators can update SQL Server 2017 instances on Linux by using their existing package update/upgrade processes.

The SQL Server service runs natively using systemd, and performance can be monitored through the file system as expected. Linux file paths are supported in T-SQL statements and scripts to do things like defining/changing the location of data files or database backup files. High availability clustering can be managed with popular Linux high availability solutions like Pacemaker and Corosync.

Full-Text Search is now available for Linux. This feature enables you to run full-text queries against character-based data in SQL Server tables. Full-text queries perform linguistic searches against text data in full-text indexes by operating on words and phrases based on the rules of a language, such as English or Japanese. Full-text queries can include simple words and phrases or multiple forms of a word or phrase. A full-text query returns any documents that contain at least one match (also known as a *hit*). A match occurs when a target document contains all the terms specified in the full-text query and meets any other search conditions, like the distance between the matching terms. For more information on this feature, see [SQL Server Full-Text Search on Linux](#).

Platform abstraction with containers

Containers are software-defined spaces with some similarities to virtual machines. Containers don't use hardware virtualization; instead, the host operating system is abstracted from the space, allowing a container to hold only the dependencies required for the application and the application itself. A Linux-based container can be deployed to any Linux machine (physical or virtual) running Docker and can be expected to run without changes to the host operating system. Windows Server 2016 includes native support for Windows containers. Docker EE for Windows Server brings the same capabilities to Windows solutions.

SQL Server 2017 supports installation on Linux operating systems like Red Hat Enterprise Linux and, by extension, Linux containers. Similarly, Docker CE for Mac enables developers to run Linux containers with SQL Server 2017 on macOS. SQL Server also supports installation on Windows containers. One of the key benefits of using containers in the development process is the ability to work in various environments. With containers, development teams can work in dev/test environments that are functionally identical to production environments. In addition, container orchestration solutions can manage the deployment of containers automatically, aiding the automation of testing and deployment.

Enabling DevOps practices with containers

DevOps is about bringing great applications to customers through people, processes, and tools. Taking a lean approach to product development (for example, splitting work into small batches and implementing customer feedback) predicts higher IT performance and less deployment pain.⁴ Compared to the challenges of a traditional development approach, or even an Agile approach, DevOps is gaining ground as a best practice for delivering high-performance solutions to market. Traditional hardware and VM installations of SQL Server have had a difficult time fitting into this framework. With SQL Server now available on containers, several DevOps practices are accessible to data-driven application development—including the ability to build a container image that can be used in any environment.

SQL Server 2017 on containers aligns well with DevOps principles and enables several key practices that play a pivotal role in bringing mission-critical, intelligent apps to production quickly, with high cost efficiency, high quality, and high customer satisfaction. From environment parity with dev/test to high-velocity deployments through CI/CD (continuous integration and continuous delivery), SQL Server on Linux and containers make a capable addition to any organization's DevOps toolbox.

SQL Server 2017 in operational usage for relational and graph data

In-Memory Online Transaction Processing

A feature since SQL Server 2014, In-memory technology for SQL Server dramatically improves the throughput and latency of SQL Server OLTP capabilities. It is designed to meet the requirements of the most demanding transaction processing applications, and Microsoft has worked closely with several companies to prove these gains. The feature set of In-Memory OLTP includes the following:

- **Memory-optimized tables:** There are two types of memory-optimized tables. Durable tables are fully logged and persist over server restarts. Nondurable tables do not persist over server restarts and are most commonly used in place of global temp tables in the user database or in scenarios where persistence is not needed, such as staging tables in an Extract Transform Load (ETL) process.
- **Memory-optimized table variables:** These variables are created using memory-optimized table types. Variables are stored in-memory, leading to more efficient data access because they use the same memory-optimized algorithms and data structures as memory-optimized tables—particularly when using natively compiled stored procedures.

⁴ Puppet and Dora, 2016 State of DevOps Report ([link](#)).

- **Natively compiled stored procedures:** SQL Server can natively compile stored procedures that access memory-optimized tables. Native compilation enables faster data access and more efficient query execution than interpreted (traditional) Transact-SQL. Natively compiled stored procedures are parsed and compiled when they are loaded to native DLLs (dynamic-link libraries). This contrasts with other stored procedures that are compiled on first run. They have an execution plan created and reused, and they use an interpreter for execution.
- **Natively compiled scalar user-defined functions (UDFs):** These replace traditional scalar UDFs that do not perform data access, and this replacement reduces UDF runtime. Natively compiled scalar UDFs cannot access disk-based tables. If data access is required, consider migrating the table to memory-optimized (if no data access occurs, migration is not required).

In-Memory OLTP is designed on the following architectural principles:

- **Optimize for main-memory data access.** Storage-optimized engines (such as the current OLTP engine in SQL Server) will retain hot data in a main-memory buffer pool based on frequency of access. The data access and modification capabilities, however, are designed so that data can be paged in or out to disk at any point. With In-Memory OLTP, you place tables used in the extreme transaction-processing portion of an application into memory-optimized main-memory structures. The remaining application tables, such as reference data details or historical data, are left in traditional storage-optimized structures. This approach enables you to optimize hotspots for memory use, without having to manage multiple data engines. Main-memory structures for In-Memory OLTP eliminate the overhead and indirection of the storage-optimized view while still providing the full atomicity, consistency, isolation, and durability (ACID) properties you expect from a database system.
- **Include tools for migration.** To identify the appropriate tables and memory structures for utilizing In-Memory OLTP, SQL Server Management Studio includes tools designed to assist users in transitioning to In-Memory OLTP. These include transaction performance analysis to identify objects that would benefit from migration, migration advisors to assist in migrating disk-based tables to memory-optimized tables, and migration of traditional stored procedures and functions to natively compiled objects.
- **Accelerate business-logic processing.** In-Memory OLTP, queries, and procedural logic in procedures that are stored in Transact-SQL (T-SQL) are compiled directly into machine code through aggressive optimizations that are applied at compilation time. Consequently, the stored procedure can be executed at the speed of native code.
- **Provide frictionless scale-up.** In-Memory OLTP implements a highly scalable concurrency control mechanism and uses a series of lock-free data structures to eliminate traditional locks and latches while guaranteeing the correct transactional semantics that ensure data consistency.
- **Integrate into SQL Server.** One of the most impressive things about In-Memory OLTP is that it achieves breakthrough improvements in transactional processing capabilities without requiring a separate data management product or new programming model. This enables an integrated developer and database administrator (DBA) experience with the same T-SQL, client stack, tooling, backup and restore, and Always On capabilities. By offering in-memory functionality within SQL Server, your total cost of ownership ends up being lower than it would be if you were to purchase, manage, and maintain a separate system for handling in-memory processing.

In-Memory OLTP enhancements in SQL Server 2017

Performance, supportability, and scaling improvements

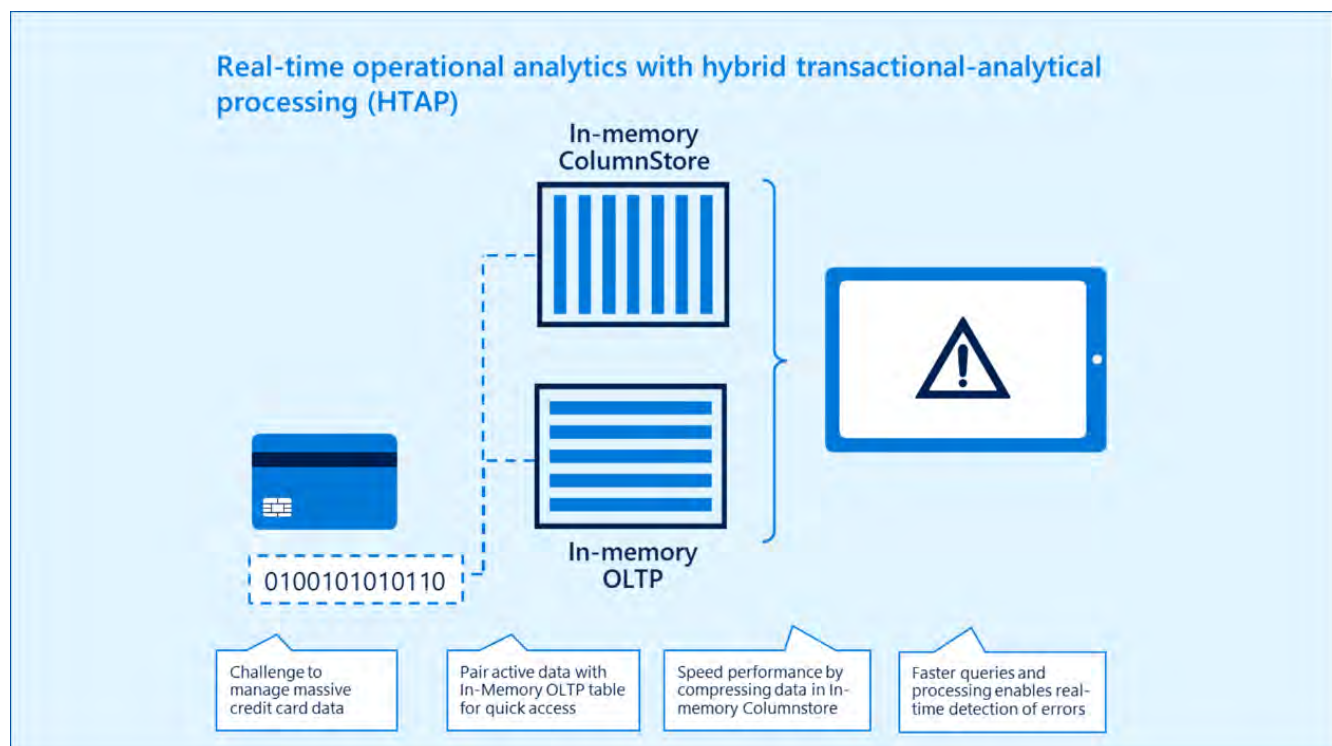
SQL Server 2017 adds features to improve the performance and supportability of In-Memory OLTP workloads. In addition, many limitations on tables and stored procedures have been removed to make it easier to migrate your applications and take advantage of the benefits of In-Memory OLTP. Scalability improvements include:

- The limitation of eight indexes on memory-optimized tables has been eliminated.
- Transaction log redo of memory-optimized tables is now done in parallel. This bolsters faster recovery times and significantly increases the sustained throughput of Always On availability group configuration.
- Performance of btree (nonclustered) index rebuild for MEMORY_OPTIMIZED tables during database recovery has been significantly optimized. This improvement substantially reduces the database recovery time when nonclustered indexes are used.
- `sp_spaceused` is now supported for memory-optimized tables.
- `sp_rename` is now supported for memory-optimized tables and natively compiled T-SQL modules.
- `ALTER TABLE` against memory-optimized tables is now substantially faster in most cases.
- Memory-optimized filegroup files can now be stored on Azure Storage. Backup/Restore of memory-optimized files on Azure Storage is supported.

Transact-SQL improvements

Query surface area in native modules has been improved to include full support for JSON functions. Additional native support for query constructs such as `CROSS APPLY`, `CASE`, and `TOP (N) WITH TIES` is now available. Memory-optimized tables now support computed columns.

Real-time operational analytics with hybrid transactional-analytical processing



A feature since SQL Server 2016, real-time operational analytics removes data latency from analytics by enabling an analytical workload to run on a table's columnstore index, in parallel with the transactional workload running on the table (also called "rowstore"). Traditionally, an analytical workload would run on a separate data warehouse store and was fed into a cube for analytical processing. Real-time operational analytics is possible because columnstore indexes mirror rowstore data, are updated nearly instantly, and operate as a separate workload on SQL Server—enabling real-time analysis without the need to wait for slow, costly ETL warehousing operations. Because the indexed data mirrors rowstore data, real-time operational analytics can run alongside an OLTP workload without impacting its performance.

These real-time analytics capabilities, coupled with the memory-optimized table capabilities of SQL Server, enable unprecedented speed with in-memory OLTP—creating a powerful hybrid transactional / analytical processing (HTAP) platform. This means customers can process high-performance transactional and analytical workloads all on a single platform.

Adaptive Query Processing feature family

New in SQL Server 2017, Adaptive Query Processing introduces new capabilities that allow the SQL Server query processor to modify plan choices based on runtime characteristics.

During query processing and optimization, the cardinality estimation (CE) process is responsible for estimating the number of rows processed at each step in an execution plan. Inaccurate estimates can cause slow query response time, excessive resource utilization (CPU, Memory, IO), and reduced throughput and concurrency. To improve CE techniques, SQL Server 2017 introduces a new feature family: adaptive query processing (AQP). AQP improves the handling of the more intractable CE issues. Features included in the AQP feature family are:

Interleaved Execution: Materializes problematic estimates of multi-statement table valued functions (MSTVF) that propagate to downstream operations, correcting the inaccurate estimates and enabling the Query Optimizer to revise plan choices based on accurate estimates. The first version of Interleaved Execution addresses cardinality estimates for MSTVFs.

Batch mode adaptive join: Enables the choice of a hash join or nested loop join method against a columnstore table to be deferred until after the first join input has been scanned. Adaptive join evaluates the input and executes the most efficient of the two join algorithms at execution time.

Batch mode memory grant feedback: Tracks the actual memory required for a query, and when an identical query statement is called, enables a more accurate memory grant size. This avoids excessive memory grants, which can reduce concurrency, in addition to underestimated memory grants, which can cause expensive spills to disk.

Automatic plan correction

New features in SQL Server 2017 detect plan choice regressions and give recommendations on how to fix the problem. These automatic plan correction features help to maintain the performance of data queries, even when application changes occur.

Automatic tuning: This database feature provides insight into potential query performance problems, recommends solutions, and automatically fixes identified problems.

Forcing last good plan: To prevent unexpected performance issues, users must periodically monitor the system and look for queries that regressed. If a plan has regressed, it's beneficial to find a previous good plan and force it, instead of using the current one. With this feature, you can monitor the performance of an executed query using the forced plan and verify that the plan works as expected.

Automatic regression detection: The Database Engine detects potential plan choice regressions and shows recommended actions to be applied in the `sys.dm_db_tuning_recommendations` view. This view shows information about the problem, the importance of the issue, and details such as the identified query, the ID of the regressed plan, the ID of the plan used as the baseline for comparison, and the Transact-SQL statement that can be executed to fix the problem.

Automatic plan tuning: The Database Engine can automatically switch to the last known good plan whenever a regression is detected.

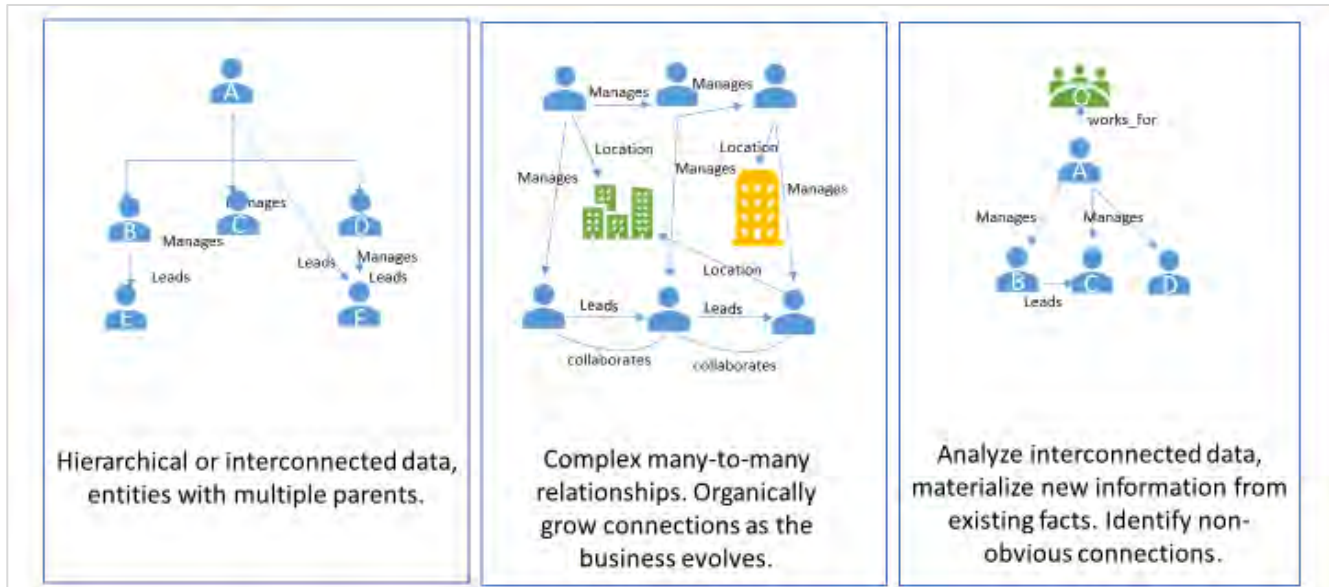
Resumable Online Index Rebuild

Planning, maintaining, and managing a large index online can be challenging. The bigger the index, the more difficult index maintenance becomes. Reorganizing and rebuilding indexes can be especially cumbersome. Resumable Online Index Rebuild in SQL Server 2017 enables continuation of an online index rebuild operation after a failure (such as a failover to a replica or insufficient disk space). Resumable Online Index Rebuild also enables pausing and later resuming an online index rebuild operation. For example, you might need to temporarily free up system resources to execute a high-priority task, or complete the index rebuild at another time if the available maintenance window is too short for a large table. Finally, Resumable Online Index Rebuild does not require significant log space, which allows you to perform log truncation while the resumable rebuild operation is running.

Non-resumable online index maintenance operations have been a feature of SQL Server Enterprise Edition since SQL Server 2005.

SQL Graph

Customers need to do more than just manage large volumes of data. They also need to analyze their existing data more effectively to understand its relationships and patterns. Querying data from a relational schema by using traditional SQL queries can be a complex task. SQL Server 2017 introduces SQL Graph to make modeling and analyzing relationships easier by allowing users to handle the relationships in a more flexible and agile way.



A graph database is a set of nodes (or vertexes) and edges (or relationships). A graph database is useful for representing data that includes many—often complex—relationships. SQL Graph in SQL Server 2017 brings graph processing capabilities to SQL Server, enabling users to link different pieces of connected data to help them gather powerful insights and increase operational agility. This is well suited for applications in which relationships are important, such as fraud detection, risk management, social networks, recommendation engines, predictive analysis, and IoT suites.

Create Graph Objects

- Create Nodes and Edges
- Properties associated with Nodes and Edges

```
CREATE TABLE Person (ID INTEGER PRIMARY KEY, name
VARCHAR(100)) AS NODE;
```

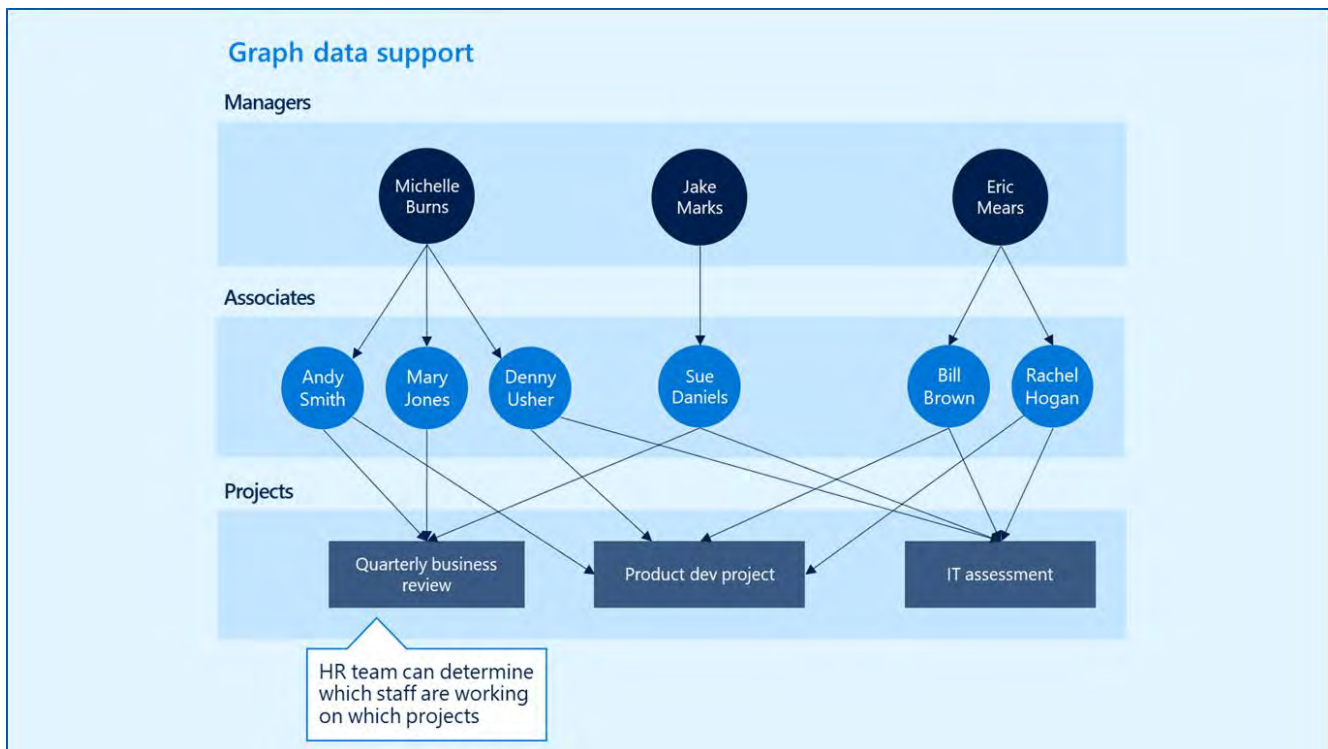
```
CREATE TABLE Restaurant (ID INTEGER PRIMARY KEY,
name VARCHAR(100)) AS NODE;
```

```
CREATE TABLE likes AS EDGE;
```

```
CREATE TABLE friends (StartDate date) AS EDGE;
```



CRUD operations with SQL Graph effectively create nodes to represent various entities and create edges to represent relationships between any two nodes. Both nodes and edges can have properties associated with them. In addition, SQL Graph can perform multi-hop navigation in a graph using pattern matching (no joins). SQL language extensions for graph support enable join-free, pattern-matching queries for multi-hop navigation.



For more information about the graph database capabilities, see [Graph processing with SQL Server and Azure SQL Database](#).

Resource Governor

A feature since SQL Server 2008, Resource Governor provides powerful and flexible controls to dictate and monitor how a SQL Server instance behaves in terms of CPU, memory and response time. Resource Governor can be used to assign a portion of total CPU, memory, and IO resources to a workload either at an aggregate level (database users and groups) or at an individual request level.

You can use Resource Governor not only to limit maximum usage in certain scenarios, but also to try to guarantee minimums, enabling you to fine tune and balance your workloads, and give the right resources to the right users at the right times. You can also use Resource Governor to observe resource usage, allowing you to record the results and to tweak your settings periodically to maximize effectiveness.

There are three fundamental components to the Resource Governor: resource pools, workload groups, and the classifier function. A resource pool can be thought of as a “slice” of all the CPU, memory, and IO resources available to the SQL Server instance, and a workload group as a set of similar requests (that you define). The classifier function enables incoming requests to be associated with a specific workload group (and, in turn, a single resource pool). These requests are then constrained by the Resource Governor (although in some configurations, constraints are applied only when there is significant contention for server resources).

Scenarios where Resource Governor can provide value include:

- **Consolidation:** You can use the metrics gathered by Resource Governor to assess server resource consumption by workload when identifying candidates for consolidation onto a shared hardware platform.

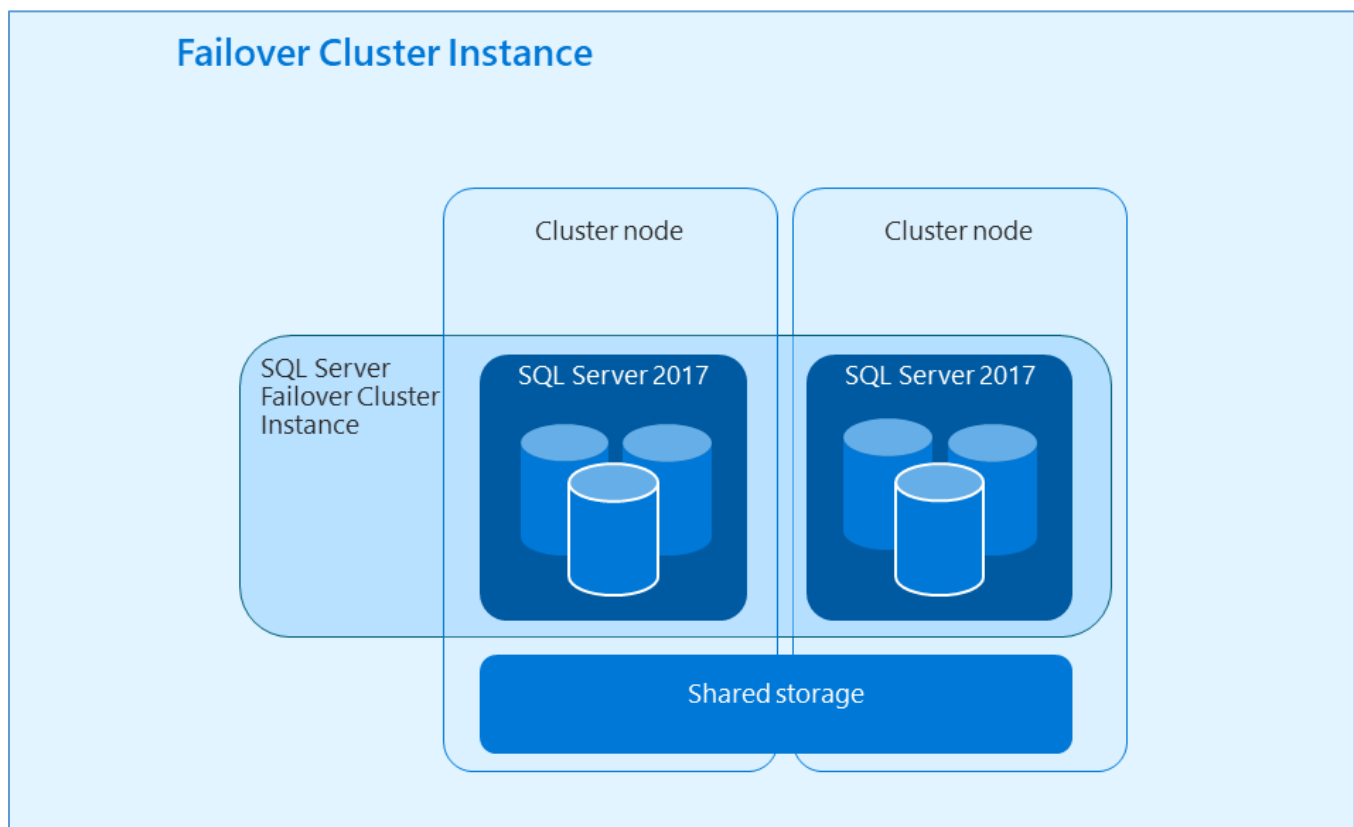
- **Predictable performance:** You can use Resource Governor limits to prevent a runaway query (typically an ad-hoc query) from consuming all server resources and harming the performance of other workloads on the same server.
- **SLAs:** You can use Resource Governor minimum configuration to guarantee that a percentage of server resources will always be available to a certain group or workload, regardless of the behavior of any other workloads active on the server.
- **Chargebacks:** Resource Governor can allow you to track resource usage among many different workloads, groups, or applications over time. This means you not only bill business units appropriately, but you also gauge which of those groups are your biggest resource consumers.

Mission-critical high availability on any platform

SQL Server high availability solutions provide mission-critical uptime, fast failover, improved manageability, and better use of hardware resources.

Instance-level high availability with Always On Failover Cluster Instances

An Always On Failover Cluster Instance (FCI) provides instance-level redundancy, allowing a SQL Server instance to remain available during planned and unplanned operating system outages caused by hardware failure, software failure, or system maintenance. FCIs are supported on Windows and Linux.



An FCI comprises two or more cluster nodes with access to cluster shared storage (either SAN or direct-attached storage). Only one node is active at a time—secondary node(s) are available but passive, ready to assume the role of active node during failover. An FCI requires a cluster manager to marshal cluster resources. On Windows, FCIs

use Windows Server Failover Clustering (WSFC) as the cluster manager; on Linux, the supported cluster manager is Pacemaker. For more information about the capabilities of this feature, see [Always On Failover Cluster Instances \(SQL Server\)](#).

Database-level high availability with Always On Availability Groups

An availability group supports a replicated environment for a discrete set of user databases, known as availability databases. You can create an availability group for high availability (HA) or for read-scale. An HA availability group is a group of databases that fail over together. A read-scale availability group is a group of databases that are copied to other instances of SQL Server for read-only workload. An availability group supports one set of primary databases and one to eight sets of corresponding secondary databases. Always On Availability Groups offers the same level of high availability and disaster recovery as Oracle Real Application Clusters but does so on fewer servers, and is included in the core SQL Server license cost.

SQL Server 2017 introduces the following enhanced features focused on ensuring high availability while running mission-critical workloads.

Flexibility for HA architectures

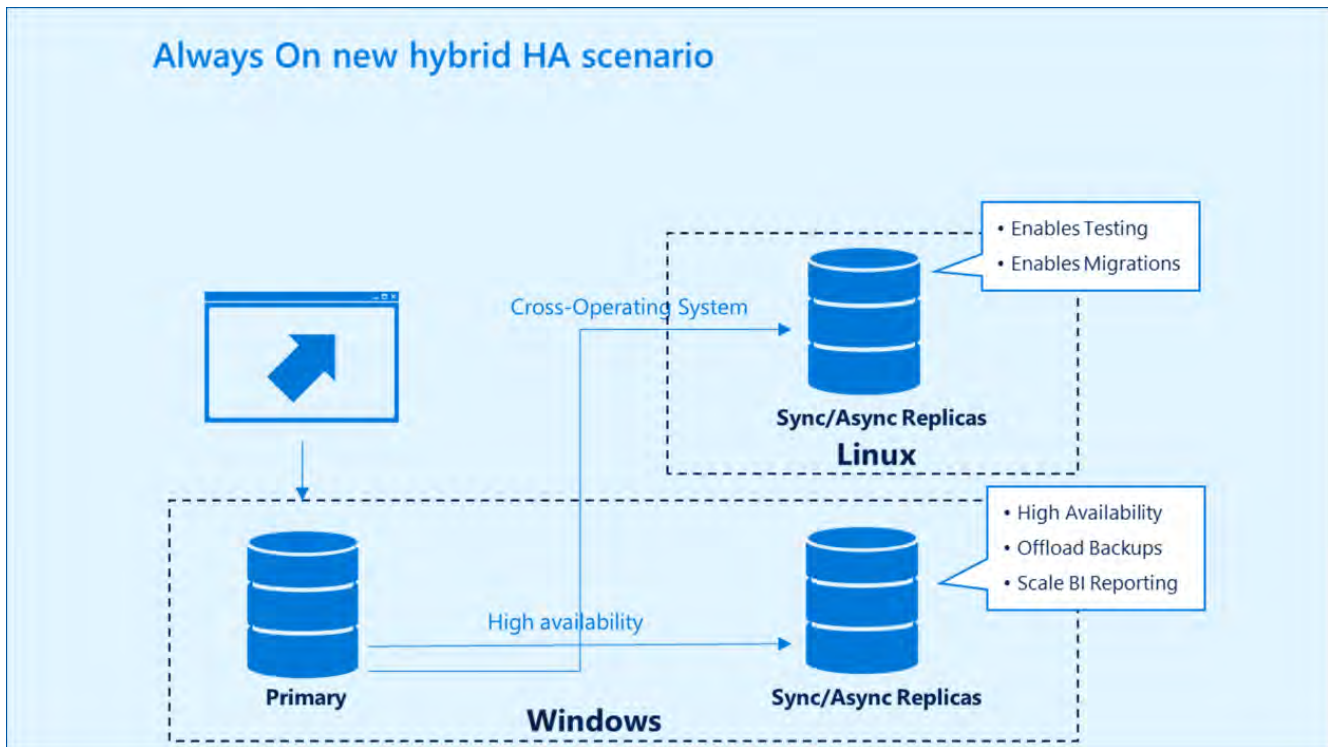
SQL Server 2017 supports two different architectures for availability groups: Always On and Read-Scale.

Always On Availability Groups: These provide high availability, disaster recovery, and read-scale balancing. These availability groups require a cluster manager—WSFC for Windows clusters and Pacemaker for Linux clusters.

Read-Scale Availability Groups: These provide read-only workload replicas but not high availability. With this architecture, there is no need for a cluster manager. The benefit of this is the ability to have secondary replicas in mixed-OS environments. Read-scale availability groups are a new feature in SQL Server 2017.

Always On Availability Groups for Linux

Always On Availability Groups have been added to the Linux edition, enabling customers to test the database software's hardiness while running critical workloads. This feature is now available on all Linux OS distributions supported by SQL Server 2017—Red Hat Enterprise Linux, Ubuntu, and SUSE Linux Enterprise Server. All the capabilities that make availability groups a flexible, integrated, and efficient HADR solution are available on Linux, including multi-database failover, multiple synchronous and asynchronous secondaries, manual or automatic failover, active secondaries for read and backup workloads, and more. For more information about the capabilities of this feature, see [Always On Availability Groups for SQL Server on Linux](#).



Member instances of a single Always On Availability Group are permitted to run on Windows, Linux, or a mixture of both operating systems. This enables organizations that plan to migrate their SQL Servers to Linux to easily test workloads and applications before switch-over.

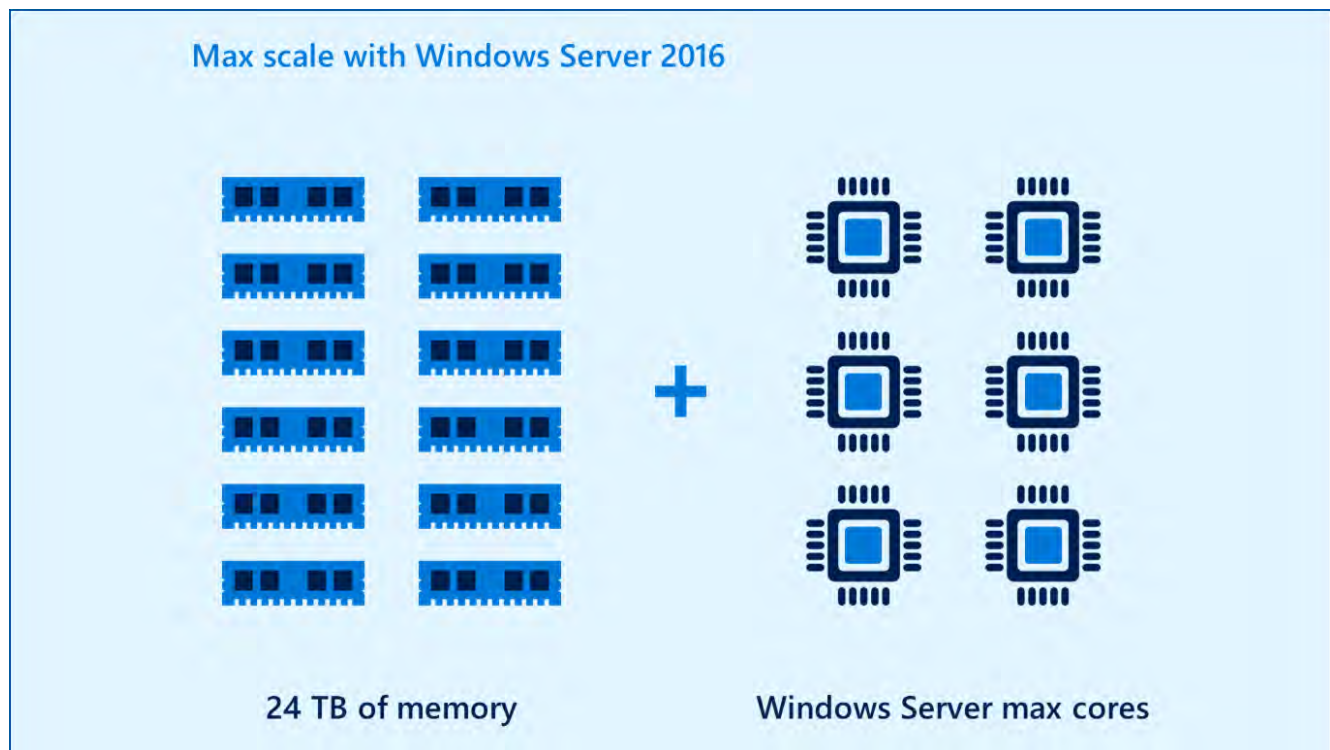
Load balancing of readable secondary replicas

Secondary replicas support read-only access to all the secondary databases. Typically, these replicas are in sync with the primary replica, including full-text indexes and durable memory-optimized tables. This means that secondary replicas can provide services somewhat like a data mart, giving read-only access to production data with little latency. Routing read-only requests can also be load balanced on the availability group listener, giving organizations control over how to route read-only workloads to their secondary replicas. For more information, see [Configure Read-Only Routing for an Availability Group](#).

SQL Server 2017 in Data Warehousing and Big Data

Petabyte-scale data warehousing

As SQL Server has evolved, Microsoft has maintained a focus on achieving petabyte-scale data warehousing. With SQL Server 2017, this scale is also available on Linux. A recent new world record in 1 TB TPC-H benchmark in non-clustered data warehouse performance was achieved with SQL Server 2017 on Red Hat Enterprise Linux and HPE ProLiant hardware, demonstrating SQL Server performance equally well on Windows and Linux operating systems.^{5 6}

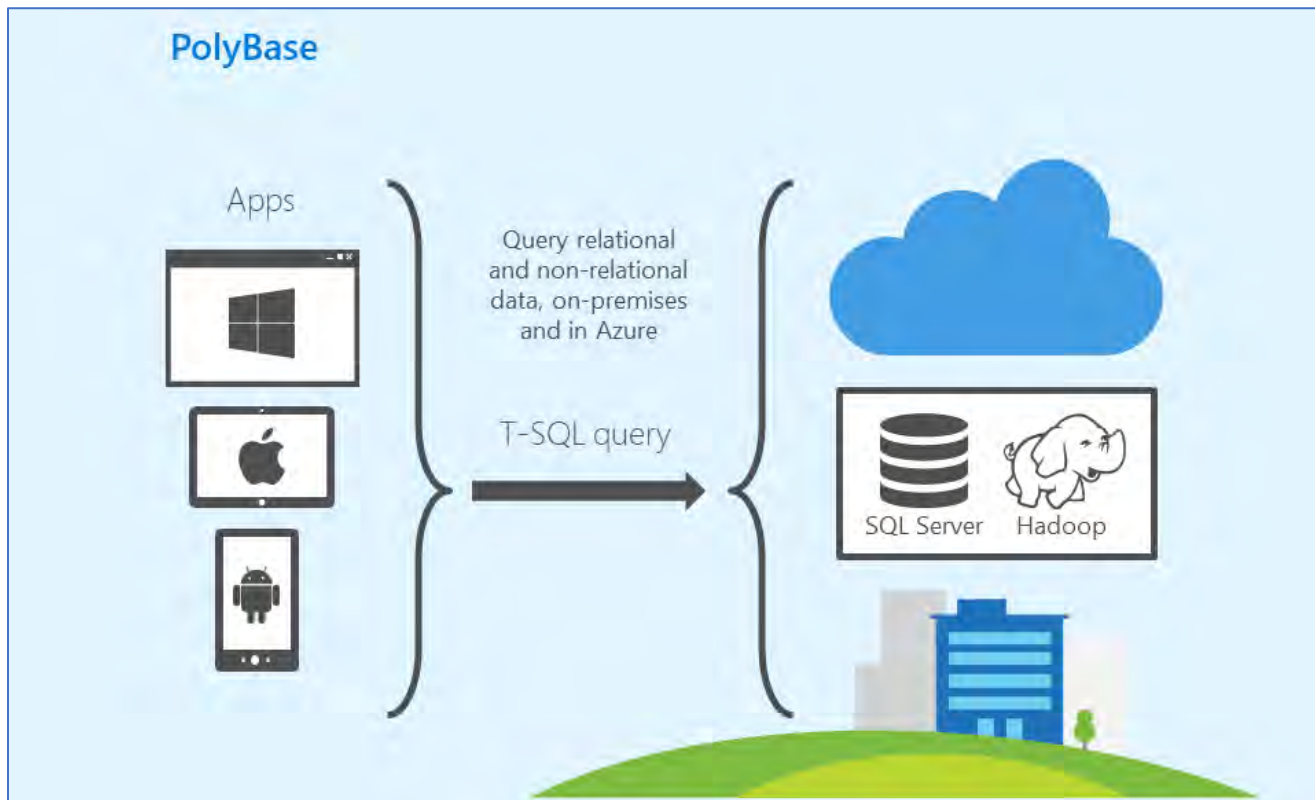


⁵ Delivering AI with data: the next generation of the Microsoft data platform ([link](#))

⁶ Microsoft SQL Server, Linux and HPE Performance Benchmark Love Story ([link](#))

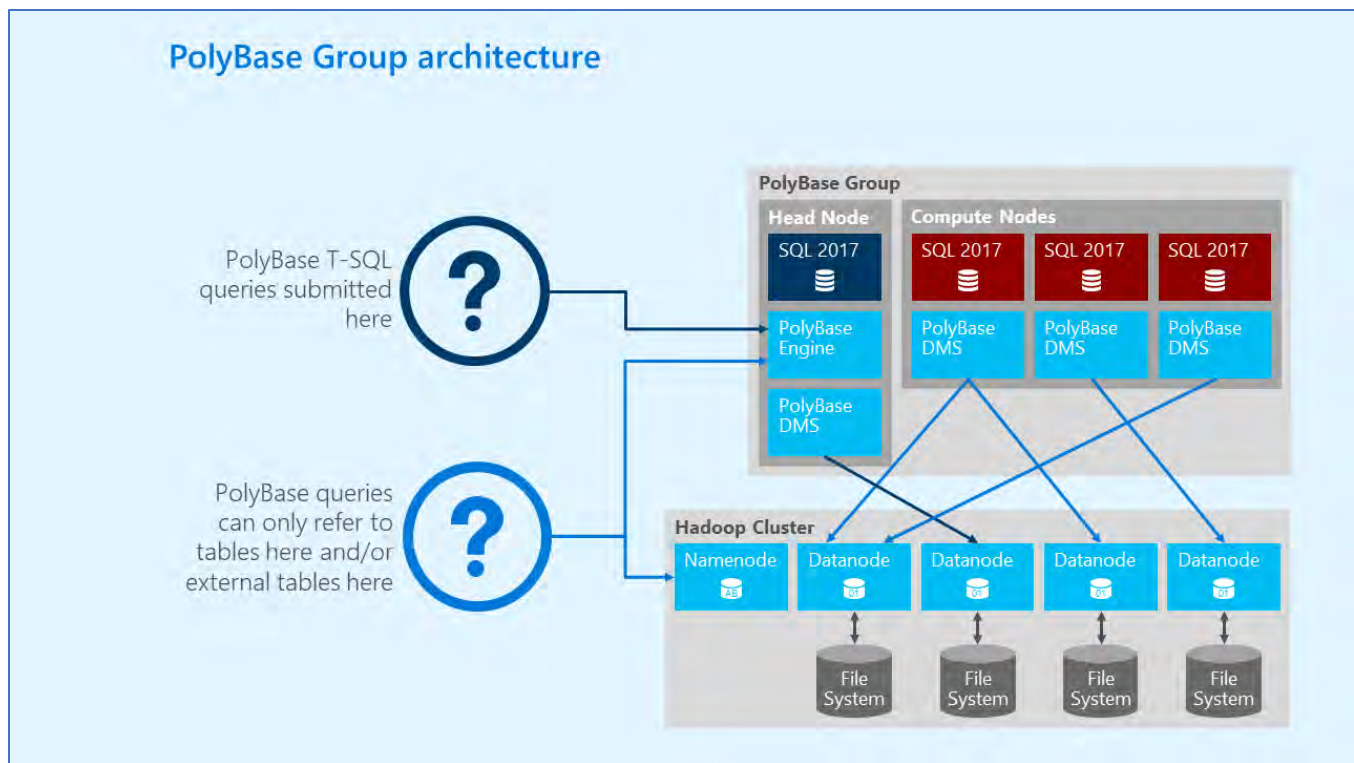
Query over any type of data

SQL Server PolyBase enables organizations to marry structured data with semi-structured and unstructured data in platforms like Azure Blob storage or Hadoop. It supports querying data stored in both stores using T-SQL, removing the need to incorporate additional query languages. PolyBase can also import and export data from or to Hadoop, Azure Blob storage, or Azure Data Lake Store without the need for separate ETL or import processes. In addition, PolyBase query optimization can push computation to a Hadoop cluster to create MapReduce jobs for more efficient distributed query performance.



While PolyBase allows you to move data in a hybrid scenario, it's also common to leave data where it resides and query it from source. This ties into the concept of a data lake. Think of a data lake as providing full access to raw big data without moving it. This can be viewed as an alternate approach to processing big data to make its analysis easier, instead of moving and synchronizing it into a data warehouse. There are several benefits to not moving data. It generally means that, beyond setting up the connectivity in the data lake, no additional development is required. Also, organizational limits to moving or modifying the data might become irrelevant with this approach. Finally, data processing and synchronization can be complex operations, and you might not know in advance how to process the data to deliver the best insights. SQL Server 2017 and PolyBase can be an important component in setting up a data lake, combining it with your relational data, and performing analysis and BI on it.

To preserve performance at scale, the PolyBase architecture supports scale-out of SQL Server nodes; 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 nodes.



Please note that PolyBase is currently a Windows-only feature.

Clustered columnstore indexes for data warehousing

First introduced in SQL Server 2014, SQL Server supports updatable clustered columnstore indexes, which replace the traditional rowstore tables. The clustered columnstore index allows users to modify data and load data concurrently for data warehouse and Decision Support System (DSS) workloads. Improved query performance of up to 100x speed-up is provided with reduced I/O and optimized query execution using techniques such as applying predicates in compressed format, pushing down predicates to storage layer when possible, using new processor architectures, and a new BATCH execution mode.

A typical scenario would be to convert a fact table from a rowstore heap or clustered index to a clustered columnstore index. A clustered columnstore index is fast, updateable, and allows fast ad-hoc queries without the need for additional indexes.

Table partitioning

A feature since SQL Server 2005, SQL Server supports table and index partitioning. The data of partitioned tables and indexes is divided into units that can be spread across more than one filegroup in a database. The data is partitioned horizontally, so that groups of rows are mapped into individual partitions. All partitions of a single index or table must reside in the same database. The table or index is treated as a single logical entity when queries or updates are performed on the data.

Partitioning large tables or indexes can have the following manageability and performance benefits:

- You can transfer or access subsets of data quickly and efficiently, while maintaining the integrity of a data collection. For example, an operation such as loading data from an OLTP to an OLAP system takes only seconds, instead of the minutes and hours the operation takes when the data is not partitioned.
- You can perform maintenance operations on one or more partitions more quickly. The operations are more efficient because they target only these data subsets, instead of the whole table. For example, you can choose to compress data in one or more partitions or rebuild one or more partitions of an index.
- You might improve query performance, based on the types of queries you frequently run and on your hardware configuration. For example, the query optimizer can process equi-join queries between two or more partitioned tables faster when the partitioning columns in the tables are the same, because the partitions themselves can be joined.
- When SQL Server performs data sorting for I/O operations, it sorts the data first by partition. SQL Server accesses one drive at a time, and this might reduce performance. To improve data sorting performance, stripe the data files of your partitions across more than one disk by setting up a RAID. In this way, although SQL Server still sorts data by partition, it can access all the drives of each partition at the same time. In addition, you can improve performance by enabling lock escalation at the partition level instead of a whole table. This can reduce lock contention on the table.

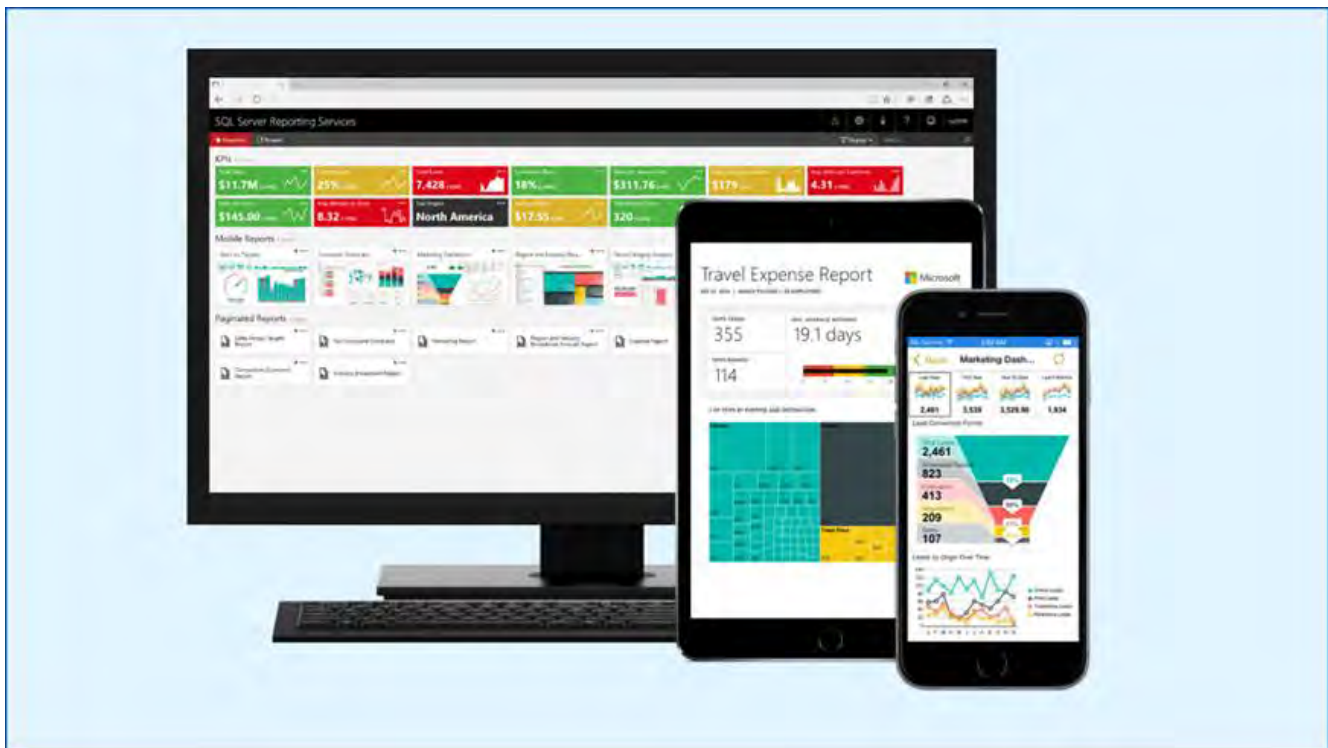
SQL Server 2017 in Business Intelligence

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.

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. 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 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 tabular reports, in addition to mobile reports and dashboards.

SQL Server 2017 adds several new features for Reporting Services, including:

- **Comments:** Comments are now available for reports, to add perspective and collaborate with others. You can also include attachments with comments.
- **Broader DAX support:** You are now able to use Report Builder and SQL Server Data Tools to create native DAX queries against supported SQL Server Analysis Services tabular data models by dragging the desired fields to the query designers.

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

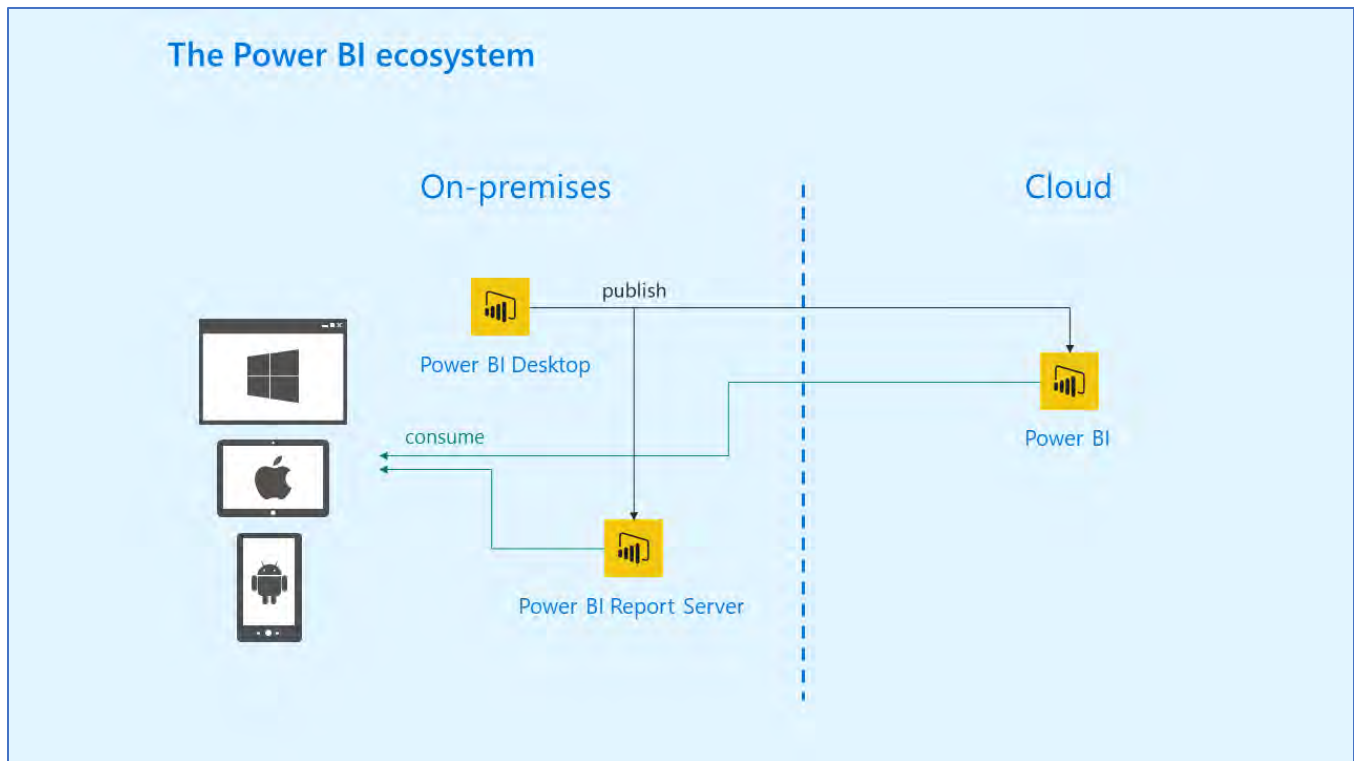
Power BI is a collection of services and features that enable your organization to share, visualize, and analyze data in a collaborative self-service fashion. Power BI can connect to a mix of on-premises and online data sources, which can be automatically refreshed on a fixed schedule. You can also use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

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 hub](#).

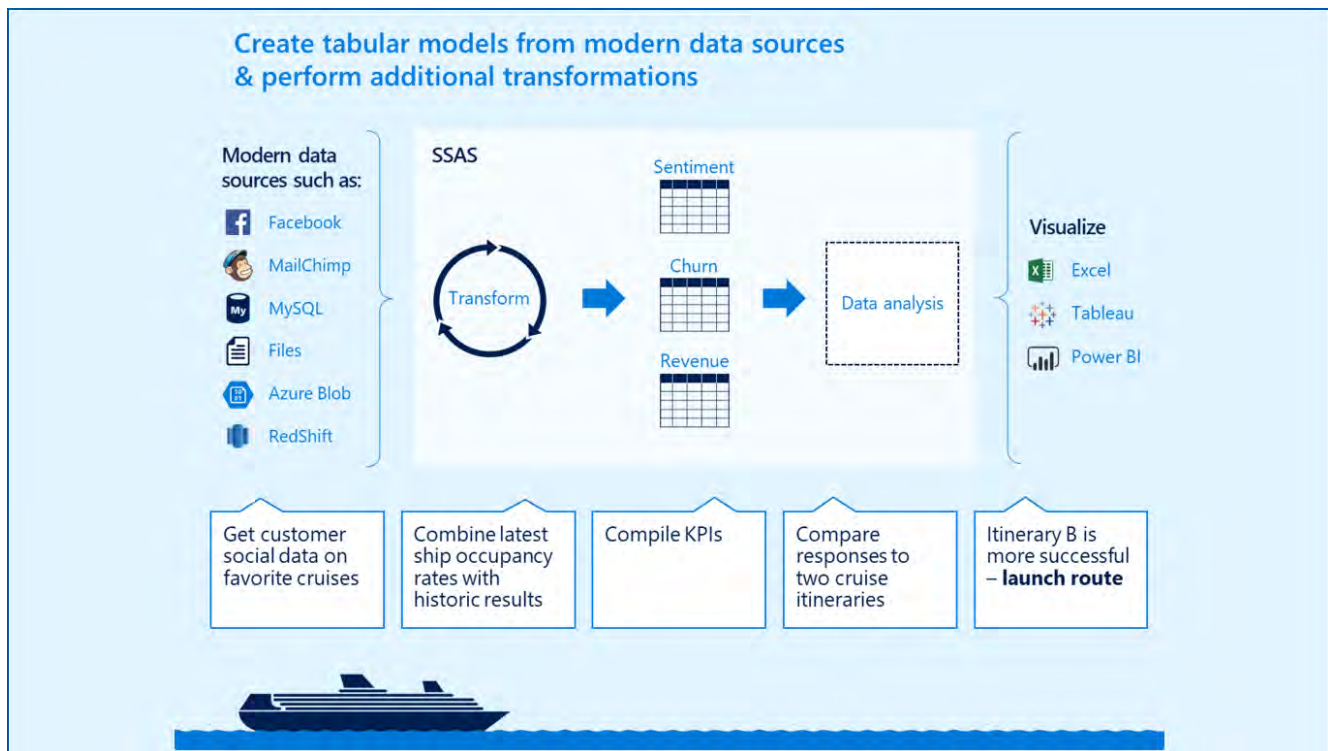
Power BI Report Server builds on SQL Server 2017 Reporting Services to allow reports designed in Power BI Desktop to be deployed on an on-premises server, rather than from the Power BI cloud service. Power BI Report Server allows organizations with data protection policies that would otherwise bar them 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 hub](#). Note that Power BI Report Server is currently a Windows-only feature.



Analysis Services

SQL Server 2017 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 is available via ROLAP. On a query server, rowsets can be cached, and stale ones paged out.

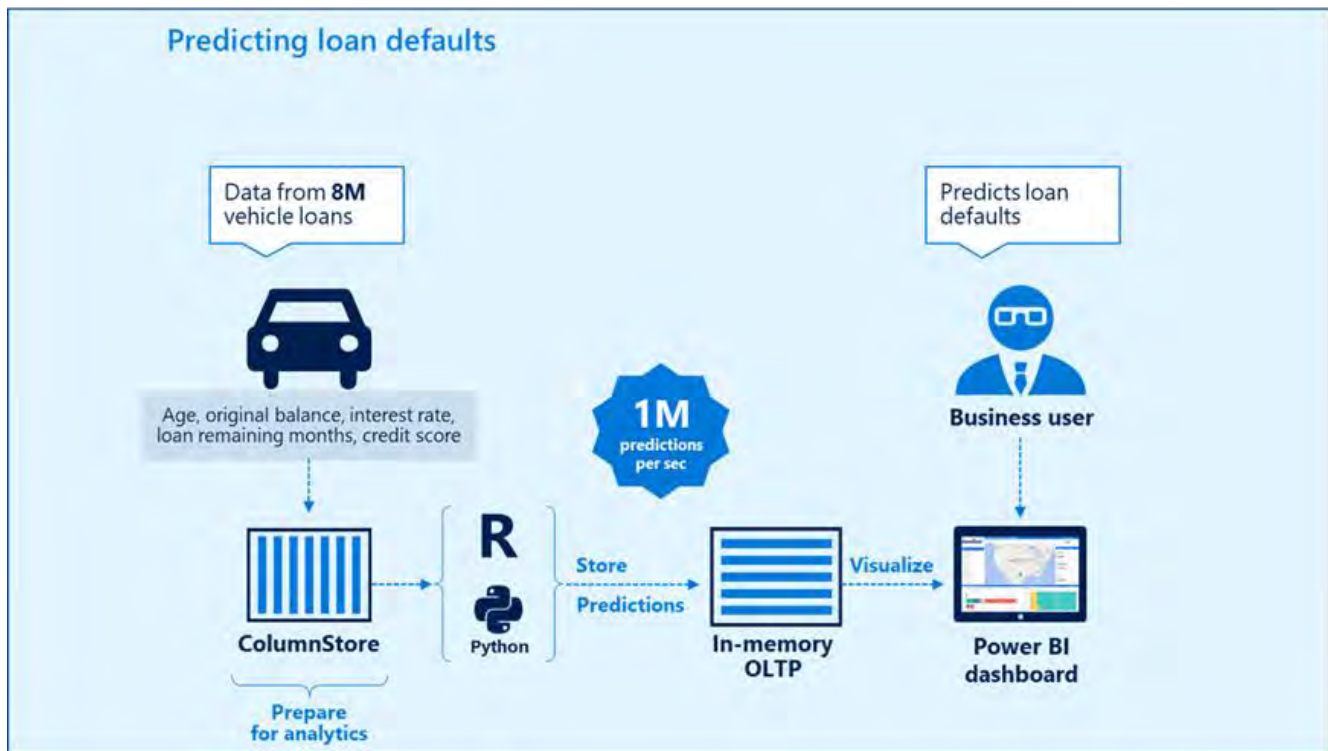
SQL Server 2017 adds several new features for Analysis Services, including:

- **Object level security for tabular models**
- **Get Data enhancements:** New data sources, modern experience for tabular models.
- **Enhanced ragged hierarchy support:** A new **Hide Members** to hide blank members in ragged hierarchies.

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

SQL Server 2017 in Advanced Analytics

With the addition of Python in SQL Server 2017, SQL R Services is being renamed **SQL Server Machine Learning Services**. With SQL Server Machine Learning Services, customers can access both scalable R and Python for their machine learning and data science needs within SQL Server. As before, a standalone version for Windows can optionally be installed. With SQL Server 2017, this standalone version for Windows is being renamed **Microsoft Machine Learning Server** and includes support for both scalable R and Python.



Industry-leading innovation in advanced analytics helps organizations discover insights faster. To start, support for the R and Python languages enables users to work with their preferred tools and push intelligence to where data lives. In addition, users can take advantage of parallelization at scale and advanced machine learning algorithms with GPUs. For example, a user can prepare data in a columnstore, use R or Python to create predictions, store the data in in-memory OLTP tables, and visualize it by using Power BI or Reporting Services. In short, this is the first commercial database that has built-in advanced analytics and machine learning.

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

SQL Server 2017 in Data Integration and Information Management

SQL Server Integration Services (SSIS)

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.

SQL Server 2017 adds several new features for Integration Services, including:

- **Linux support:** You can now run SSIS packages on computers running Linux.
- **Scale-out support:** SSIS Scale Out consists of an SSIS Scale Out Master and one or more SSIS Scale Out Workers. The Scale Out Master is responsible for Scale Out management and receives package execution requests from users. Scale Out Workers pull execution tasks from the Scale Out Master and do the package execution work. Integration Services Scale Out can be configured on one machine, where a Scale Out Master and a Scale Out Worker are set up side-by-side on the machine. Scale Out can also run on multiple machines, where each Scale Out Worker is on a different machine.

For more information about the capabilities of Integration Services, see [SQL Server Integration Services](#).

Master Data Services

A feature since SQL Server 2008 R2, 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.

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.

SQL Server 2017 includes performance enhancements for Master Data Services.

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.

Data Quality Services

A feature since SQL Server 2012, Data Quality Services (DQS) is a knowledge-driven data quality product. DQS enables you to build a knowledge base and use it to perform a variety of critical data quality tasks, including correction, enrichment, standardization, and de-duplication of your data. DQS enables you to perform data cleansing by using cloud-based reference data services provided by reference data providers. DQS also provides you with profiling that is integrated into its data-quality tasks, enabling you to analyze the integrity of your data.

DQS consists of Data Quality Server and Data Quality Client, both of which are installed as part of SQL Server 2017. Data Quality Server is a SQL Server instance feature that consists of three SQL Server catalogs with data-quality functionality and storage. Data Quality Client is a SQL Server shared feature that business users, information workers, and IT professionals can use to perform computer-assisted data quality analyses and manage their data quality interactively. You can also perform data quality processes by using the DQS Cleansing component in Integration Services and the Master Data Services (MDS) data quality functionality, both of which are based on DQS.

The data-quality solution provided by DQS enables a data steward or IT professional to maintain the quality of their data and ensure that the data is suited for its business usage. DQS is a knowledge-driven solution that provides both computer-assisted and interactive ways to manage the integrity and quality of your data sources. DQS enables you to discover, build, and manage knowledge about your data. You can then use that knowledge to perform data cleansing, matching, and profiling. You can also use the cloud-based services of reference data providers in a DQS data-quality project.

DQS provides the following features to resolve data quality issues.

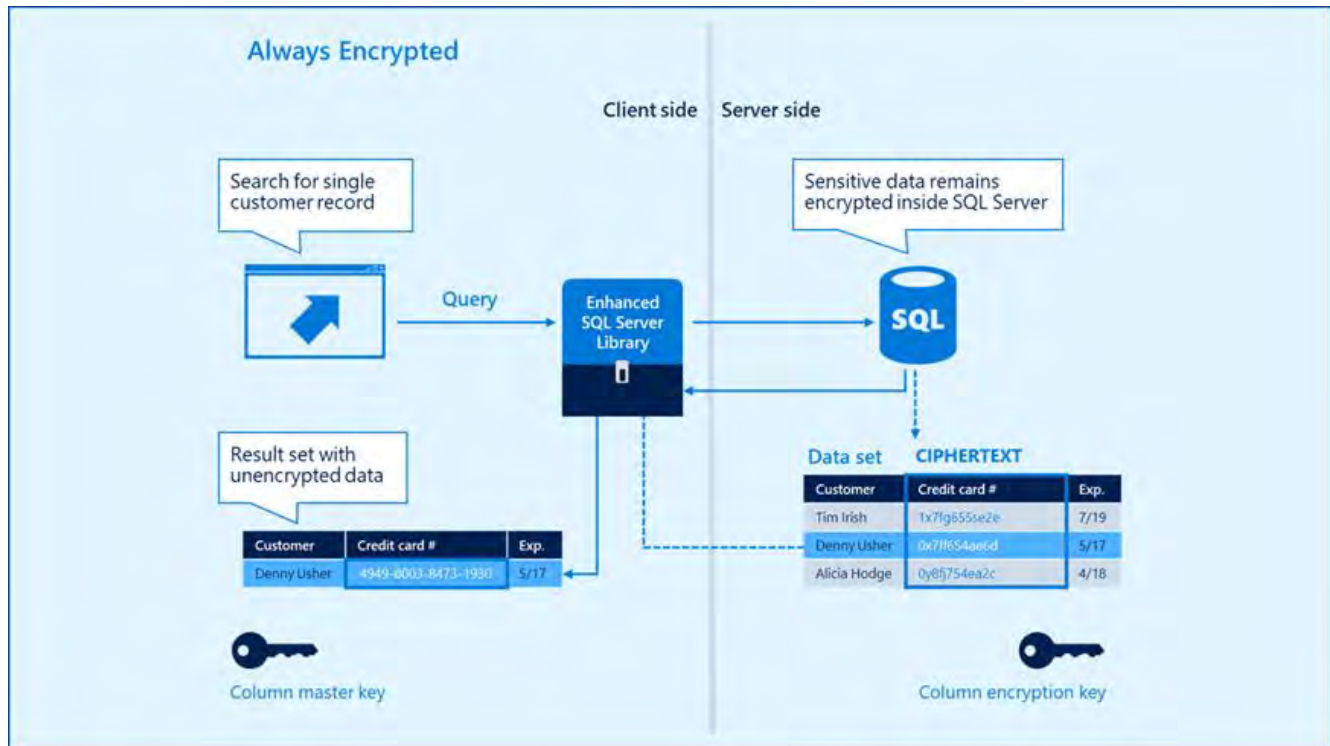
- **Data Cleansing:** The modification, removal, or enrichment of data that is incorrect or incomplete, using both computer-assisted and interactive processes.
- **Matching:** The identification of semantic duplicates in a rules-based process that enables you to determine what constitutes a match and perform de-duplication.
- **Reference Data Services:** Verification of the quality of your data using the services of a reference data provider. You can use reference data services from Microsoft Azure Marketplace to cleanse, validate, match, and enrich data.
- **Profiling:** The analysis of a data source to provide insight into the quality of the data at every stage in the knowledge discovery, domain management, matching, and data cleansing processes. Profiling is a powerful tool in a DQS data quality solution. You can create a data quality solution in which profiling is just as important as knowledge management, matching, or data cleansing.
- **Monitoring:** The tracking and determination of the state of data quality activities. Monitoring enables you to verify that your data quality solution is doing what it was designed to do.
- **Knowledge Base:** Data Quality Services is a knowledge-driven solution that analyzes data based on knowledge that you build with DQS. This enables you to create data quality processes that continually enhance the knowledge about your data—this helps to continually improve the quality of your data.

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

SQL Server 2017 Security

Every edition of SQL Server provides a robust set of features designed to keep organizational data separate, secure, and safe.

Always Encrypted



Always Encrypted is a feature designed to protect sensitive data, such as credit card numbers or national identification numbers (social security numbers), stored in Azure SQL Database or SQL Server databases. Always Encrypted allows customers to encrypt sensitive data inside their applications and never reveal the encryption keys to the Database Engine (SQL Database or SQL Server). As a result, Always Encrypted provides separation between those who own the data (and can view it) and those who manage the data (but should have no access). By ensuring that on-premises database administrators, cloud database operators, and other highly privileged but unauthorized users can't access the encrypted data, Always Encrypted enables customers to confidently store sensitive data outside of their direct control. This allows organizations to encrypt data at rest and in use for storage in Azure, to enable delegation of on-premises database administration to third parties, and to reduce security clearance requirements for their own DBA staff. For more information, see [Always Encrypted](#).

Transparent Data Encryption

Transparent Data Encryption (TDE) encrypts SQL Server, Azure SQL Database, and Azure SQL Data Warehouse data files. This is known as encrypting data at rest. Organizations can take several precautions to help secure the database, such as designing a secure system, encrypting confidential assets, and building a firewall around the database servers. However, in a scenario where the physical media (such as drives or backup tapes) is stolen, a malicious party can restore or attach the database and browse the data. One solution is to encrypt the sensitive data in the database and protect the keys that are used to encrypt the data with a certificate. This prevents anyone without the keys from using the data, but this protection must be configured in advance.

TDE performs real-time I/O encryption and decryption of the data and log files. The encryption uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. For more information, see [Transparent Data Encryption \(TDE\)](#).

Row-Level Security

Row-Level Security (RLS) enables customers to control access to rows in a database table based on the characteristics of the user executing a query (for example, group membership or execution context).

Row-Level Security simplifies the design and coding of security in an application. Row-Level Security enables organizations to implement restrictions on data row access. For example, an organization can ensure that employees can access only those data rows that are pertinent to their department, or restrict a customer's data access to only the data relevant to their company.

The access-restriction logic is in the database tier rather than separate from the data in another application tier. The database system applies the access restrictions every time data access is attempted from any tier. This makes the security system more reliable and robust by reducing its surface area. For more information, see [Row-Level Security](#).

Dynamic data masking

Dynamic data masking (DDM) limits sensitive data exposure by masking it to nonprivileged users. It can be used to greatly simplify the design and coding of security in an application.

Dynamic data masking helps prevent unauthorized access to sensitive data by enabling organizations to designate how much of the sensitive data to reveal, and it has minimal impact on the application layer. Dynamic data masking can be configured on the database to hide sensitive data in the result sets of queries over designated database fields without changing the data in the database. Dynamic data masking is easy to use with existing applications because masking rules are applied in the query results. In many applications, sensitive data can be masked without changes to existing queries. For more information, see [Dynamic Data Masking](#).

Auditing

SQL Server audit allows customers to track and log events that take place on an instance of the Database Engine, or on an individual database. Server audits can contain server audit specifications for server level events, and database audits can contain specifications for database level events. Audited events can be written to the event logs or to audit files.

There are several levels of auditing for SQL Server, depending on government or standards requirements for your installation. SQL Server Audit provides the tools and processes you must have to enable, store, and view audits on various server and database objects. For more information, see [SQL Server Audit](#).

SQL Server 2017 Cloud

The concept of hybrid cloud recognizes that organizations typically have a portfolio of different applications deployed across their business and a breadth of environments that have unique requirements. Some applications require detailed and complex hardware configurations that defy deployment into the type of commoditized, “one-size-fits-all” environment offered by cloud computing. Equally, massive-scale public clouds are a compelling resource for businesses with applications that experience massive peaks and troughs in demand, because it can be economically infeasible to allocate sufficient levels of on-premises hardware to deal with peak demand. The Microsoft goal for hybrid cloud is to offer organizations breadth of choice in how and where they run their applications, while at the same ensuring they can use a common set of server products, tools, and expertise across a portfolio of solutions.

Backup to Azure

Your backup strategy can be greatly enhanced with a hybrid cloud approach. SQL Server has several options for backing up to Azure, including managed backup, backup to Azure block blobs, and Azure Storage snapshot backup. Managed backup provides the capability to easily manage and automate SQL Server backups to Azure Blob storage. You can manage the whole instance or individual databases with an easy interface that can be accessed directly in the SQL Server Management Studio Object Explorer in the management node. It provides off-site, geo-redundant backups with the ability to control the retention period, and supports point-in-time restore for the retention period specified.

Managed backup can also be configured at the database level or at the SQL Server instance level. When configuring at the instance level, any new databases are also backed up automatically. Settings at the database level can be used to override instance-level defaults. You can also encrypt the backups for additional security, and set up an automated and custom schedule to control when the backups are taken. Backup to Azure block blobs allows you to manage backups to Azure Blob storage with fine-grained control over the process.

SQL Server 2014 introduced data files in Microsoft Azure, which enables native support for SQL Server database files stored as Azure blobs. This allows you to create a database in SQL Server running on-premises or in a virtual machine on Azure with a dedicated storage location for your data in Azure Blob storage. This also provides an alternative storage location for your database backup files by allowing you to restore them from or to Azure Storage. Backup with file snapshot builds on this capability, and provides the fastest and cheapest method for creating backups and running restores. It uses Azure snapshots to provide nearly instantaneous backups and quicker restores for database files that are stored using Azure Blob storage. This capability enables you to simplify your backup and restore policies.

Unlike Azure backup with block blobs, data is not actually moved. Instead, when SQL Server database files are directly stored in Azure Storage, a snapshot of those files is created. You only need to run a full backup once to establish the backup chain. Snapshot backups minimize the use of SQL Server resources to create the backup. This is especially useful for moderate to very large databases, where the impact of backups can be significant.

Because each file snapshot backup set contains a snapshot of each database file, a restore process requires, at most, two adjacent backup sets. This is true regardless of whether the backup set is from a full database backup or log backup. This is very different than the restore process when using traditional streaming backup files to perform the restore process. With traditional streaming backup, the restore process requires the use of an entire chain of backup sets: the full backup, a differential backup, and one or more transaction log backups. The recovery

portion of the restore process remains the same, regardless of whether the restore is using a file snapshot backup or streaming backup set.

SQL Server 2017 in Azure VMs

The Azure platform offers a quick and simple way to run SQL Server in a public cloud. A wide choice of SQL Server versions and editions is available; the Azure gallery contains prebuilt VM images for all currently supported SQL Server versions:

- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017

In all editions:

- Enterprise
- Standard
- Web
- Developer
- Express

SQL Server 2017 images are available running on Windows Server 2016 or Linux; earlier SQL Server versions are available running on Windows. Gallery images are regularly patched with security and functionality releases. The gallery images allow you to spin up a new SQL Server VM in less than 10 minutes.

SQL Server on Azure VMs can be licensed through the per-minute cost of running the gallery image—meaning you pay only for what you use—or customers with Software Assurance can transfer existing licenses into Azure with the bring-your-own-license (BYOL) gallery images.

Azure VMs offer performance options to suit any workload, from single-core machines with 750 MB RAM to 128 cores with 2048 GB RAM. Azure VMs support elastic scaling, so you can add or remove computing power in response to demand. Azure Premium Storage offers easy access to high-performance storage, with an average disk latency less than 4ms, plus a local SSD read-cache with latency of less than 1ms. Your data is protected against hardware failure with three local copies and three remote copies of every disk.

Azure is secure and currently holds more security certifications than any other cloud provider. Tight physical security restricts access to Azure data centers; encryption protects your data from unauthorized access. Network security restricts access to servers and services to IP addresses inside your Azure network. Access to Azure from external networks can be secured with TLS over the public internet, or over a virtual private network connection to your data centers. To learn more about the Azure security, visit the [Azure Trust Center](#).

In addition to enabling you to easily move existing SQL Server workloads to the cloud, SQL Server on Azure VMs can participate in other ways in your SQL Server estate—for example, in an Always On Availability Group, by performing the role of a third data center for the purposes of disaster recovery or geographical distribution of data.

Deploy SQL Server by operational need, not feature set

Starting with SQL Server 2016 SP1 (released in November 2016), Microsoft has made key improvements that provide a more consistent programmability surface area for developers and organizations across SQL Server editions. This enables customers and partners to build advanced applications that scale across editions and the cloud as they grow. Developers and application partners can build to a single programming surface area when creating or upgrading intelligent applications and use the edition that scales to each application's needs. SQL Server 2017 now brings this capability to other platforms. This includes features like In-Memory OLTP, In-Memory Columnstore, PolyBase, compression, and partitioning; security features like Always Encrypted, Row-Level Security, and Dynamic Data Masking; and basic reporting, analytics and machine learning (with R and Python) that are available everywhere.

Cross-box scale limits

Feature	Enterprise	Standard	Express
Maximum compute capacity used by a single instance—SQL Server Database Engine	Operating system maximum	Limited to the lesser of 4 sockets or 24 cores	Limited to the lesser of 1 socket or 4 cores
Maximum memory utilized per instance of SQL Server Database Engine	Operating system maximum	128 GB	1.4 GB
Maximum relational database size	524 PB	524 PB	10 GB

This table is a summary—for a complete list of features by edition, see [Editions and supported features of SQL Server 2017](#).

Considerations for performance begin with core count and memory allocation. They also include the maximum size for databases and, potentially, I/O capabilities and data partitioning—capabilities dictated by the expected load for the hosted data. High availability considerations focus on the number of secondary replicas required and whether any secondary requires backups, read-only access, and so on. If requirements change, transitioning an application to a different SQL Server edition is a matter of migration, not a rewrite of involved components.

Conclusion

SQL Server 2017 is another evolutionary step for SQL Server. It continues to build on the industry-leading performance and security capabilities of SQL Server through new technology and innovation. With these enterprise-class features now available on any SQL Server edition, organizations can also choose the best edition based on their operational needs. New features like Resumable Online Index Rebuild, Adaptive Query Processing, and automatic plan correction enable organizations to further optimize their data processing capabilities. SQL Graph brings the ability to map and query relationships in a graph structure rather than using traditional relational

model. Finally, with SQL Server now available on Linux operating systems, in addition to Linux and Windows containers, organizations have a full range of options for building or extending their data ecosystems.

Calls to action

For more information about SQL Server, see: <https://docs.microsoft.com/en-us/sql/sql-hub-menu>

To download the SQL Server 2017 Evaluation, see: <http://www.microsoft.com/en-us/sql-server/sql-server-2017#resources>