**What is the Azure SQL Database service?**

SQL Database is a general-purpose relational database service in Microsoft Azure that supports structures such as relational data, JSON, spatial, and XML. It delivers [dynamically scalable performance](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-service-tiers) and provides options such as [columnstore indexes](https://docs.microsoft.com/sql/relational-databases/indexes/columnstore-indexes-overview) for extreme analytic analysis and reporting, and [in-memory OLTP](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-in-memory) for extreme transactional processing. Microsoft handles all patching and updating of the SQL code base seamlessly and abstracts away all management of the underlying infrastructure.

SQL Database shares its code base with the [Microsoft SQL Server database engine](https://docs.microsoft.com/sql/sql-server/sql-server-technical-documentation). With Microsoft's cloud-first strategy, the newest capabilities of SQL Server are released first to SQL Database, and then to SQL Server itself. This approach provides you with the newest SQL Server capabilities with no overhead for patching or upgrading - and with these new features tested across millions of databases. For information about new capabilities as they are announced, see:

* [**Azure Roadmap for SQL Database**](https://azure.microsoft.com/roadmap/?category=databases): A place to find out what’s new and what’s coming next.
* [**Azure SQL Database blog**](https://azure.microsoft.com/blog/topics/database): A place where SQL Server product team members blog about SQL Database news and features.

SQL Database delivers predictable performance at multiple service levels that provides dynamic scalability with no downtime, built-in intelligent optimization, global scalability and availability, and advanced security options — all with near-zero administration. These capabilities allow you to focus on rapid app development and accelerating your time to market, rather than allocating precious time and resources to managing virtual machines and infrastructure. The SQL Database service is currently in 38 data centers around the world, with more data centers coming online regularly, which enables you to run your database in a data center near you.

Note

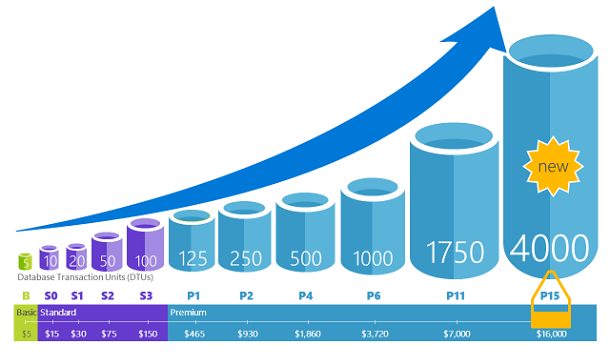
See [Azure Trust Center](https://azure.microsoft.com/support/trust-center/security/) for information about Azure's platform security.

**Scalable performance and pools**

With SQL Database, each database is isolated from each other and portable, each with its own [service tier](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-service-tiers) with a guaranteed performance level. SQL Database provides different performance levels for different needs, and enables databases to be pooled to maximize the use of resources and save money.

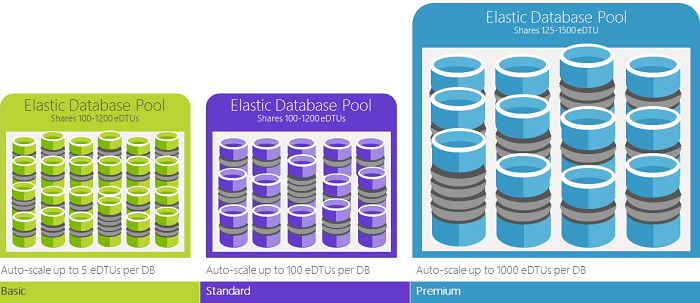
**Adjust performance and scale without downtime**

SQL Database offers four service tiers to support lightweight to heavyweight database workloads: Basic, Standard, Premium, and Premium RS. You can build your first app on a small, single database at a low cost per month and then change its service tier manually or programmatically at any time to meet the needs of your solution. You can adjust performance without downtime to your app or to your customers. Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements and enables you to only pay for the resources that you need when you need them.



**Elastic pools to maximize resource utilization**

For many businesses and applications, being able to create single databases and dial performance up or down on demand is enough, especially if usage patterns are relatively predictable. But if you have unpredictable usage patterns, it can make it hard to manage costs and your business model. [Elastic pools](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-elastic-pool) are designed to solve this problem. The concept is simple. You allocate performance resources to a pool rather than an individual database, and pay for the collective performance resources of the pool rather than for single database performance.



With elastic pools, you don’t need to focus on dialing database performance up and down as demand for resources fluctuates. The pooled databases consume the performance resources of the elastic pool as needed. Pooled databases consume but don’t exceed the limits of the pool, so your cost remains predictable even if individual database usage doesn’t. What’s more, you can [add and remove databases to the pool](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-elastic-pool-manage-portal), scaling your app from a handful of databases to thousands, all within a budget that you control. You can also control the minimum and maximum resources available to databases in the pool to ensure that no database in the pool uses all the pool resources and that every pooled database has a guaranteed minimum amount of resources. To learn more about design patterns for SaaS applications using elastic pools, see [Design Patterns for Multi-tenant SaaS Applications with SQL Database](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-design-patterns-multi-tenancy-saas-applications).1

**Blend single databases with pooled databases**

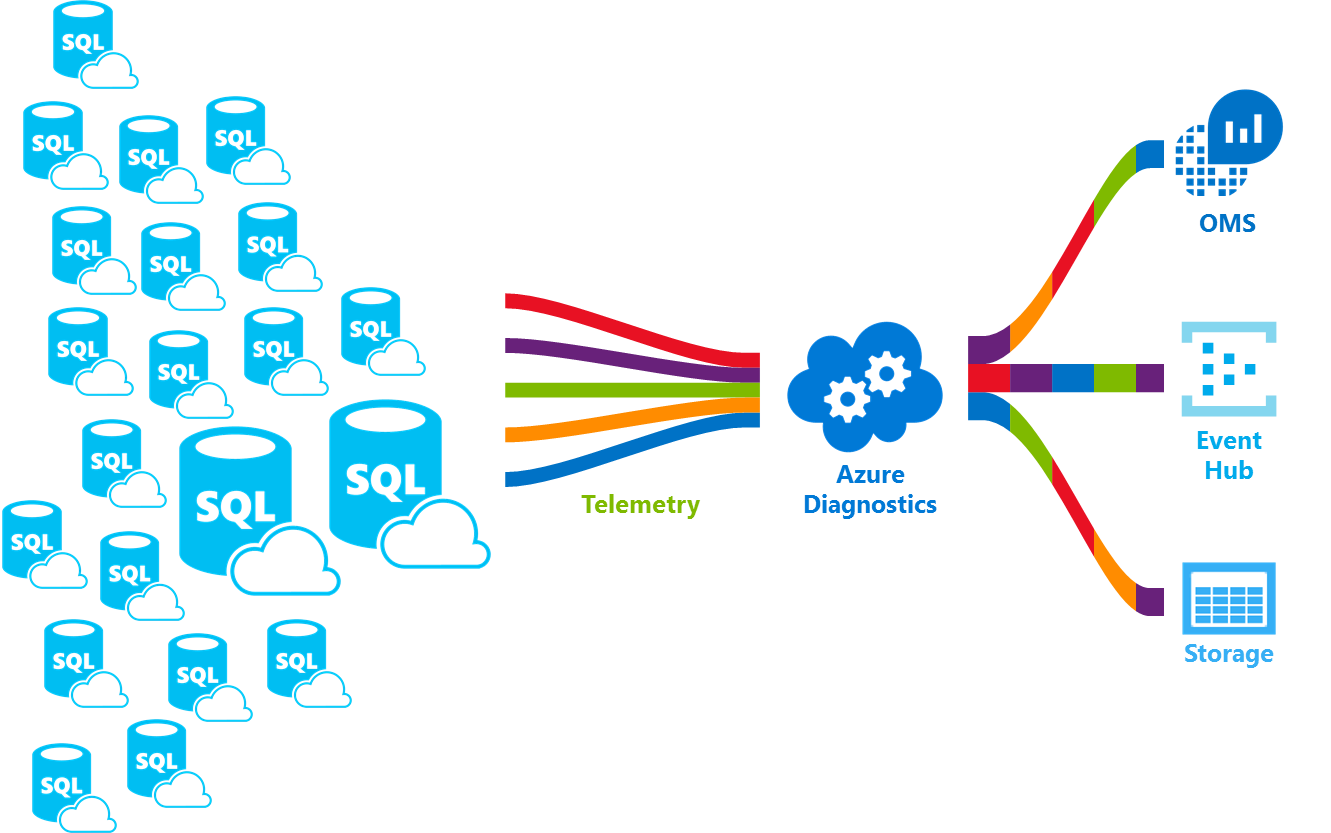
Either way you go — single databases or elastic pools — you are not locked in. You can blend single databases with elastic pools, and change the service tiers of single databases and elastic pools quickly and easily to adapt to your situation. With the power and reach of Azure, you can mix-and-match other Azure services with SQL Database to meet your unique app design needs, drive cost and resource efficiencies, and unlock new business opportunities.

**Extensive monitoring and alerting capabilities**

But how can you compare the relative performance of single databases and elastic pools? How do you know the right click-stop when you dial up and down? You use the [built-in performance monitoring](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-performance) and [alerting](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-insights-alerts-portal) tools, combined with the performance ratings based on [Database Transaction Units (DTUs) for single databases and elastic DTUs (eDTUs) for elastic pools](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-what-is-a-dtu). Using these tools, you can quickly assess the impact of scaling up or down based on your current or project performance needs. See [SQL Database options and performance: Understand what's available in each service tier](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-service-tiers) for details.

Additionally, SQL Database can [emit metrics and diagnostic logs](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-metrics-diag-logging) for easier monitoring. You can configure SQL Database to store resource usage, workers and sessions, and connectivity into one of these Azure resources:

* **Azure Storage**: For archiving vast amounts of telemetry for a small price
* **Azure Event Hub**: For integrating SQL Database telemetry with your custom monitoring solution or hot pipelines
* **Azure Log Analytics**: For built-in monitoring solution with reporting, alerting, and mitigating capabilities



**Availability capabilities**

Azure's industry leading 99.99% availability service level agreement [(SLA)](http://azure.microsoft.com/support/legal/sla/), powered by a global network of Microsoft-managed datacenters, helps keep your app running 24/7. In addition, SQL Database provides built-in [business continuity and global scalability](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-business-continuity) features, including:

* [**Automatic backups**](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-automated-backups): SQL Database automatically performs full, differential, and transaction log backups.
* [**Point-in-time restores**](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-recovery-using-backups): SQL Database supports recovery to any point in time within the automatic backup retention period.
* [**Active geo-replication**](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-geo-replication-overview): SQL Database allows you to configure up to four readable secondary databases in either the same or globally distributed Azure data centers. For example, if you have a SaaS application with a catalog database that has a high volume of concurrent read-only transactions, use active geo-replication to enable global read scale and remove bottlenecks on the primary that are due to read workloads.
* [**Failover groups**](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-geo-replication-overview): SQL Database allows you to enable high availability and load balancing at global scale, including transparent geo-replication and failover of large sets of databases and elastic pools. Failover groups and active geo-replication enables creation of globally distributed SaaS applications with minimal administration overhead leaving all the complex monitoring, routing, and failover orchestration to SQL Database.

**Built-in intelligence**

With SQL Database, you get built-in intelligence that helps you dramatically reduce the costs of running and managing databases and maximizes both performance and security of your application. Running millions of customer workloads around-the-clock, SQL Database collects and processes a massive amount of telemetry data, while also fully respecting customer privacy behind the scenes. Various algorithms are continuously evaluating the telemetry data so that the service can learn and adapt with your application. Based on this analysis, the service comes up with performance improving recommendations tailored to your specific workload.

**Automatic performance monitoring and tuning**

SQL Database provides detailed insight into the queries that you need to monitor. SQL Database's learns about your database patterns and enables you to adapt your database schema to your workload. SQL Database provides [performance tuning recommendations](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-advisor), where you can review tuning actions and apply them.

However, constantly monitoring database is a hard and tedious task, especially when dealing with many databases. [Intelligent Insights](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-intelligent-insights) does this job for you by automatically monitoring SQL Database performance at scale and it informs you of performance degradation issues, it identifies the root cause of the issue and provides performance improvement recommendations when possible.

Managing a huge number of databases might be impossible to do efficiently even with all available tools and reports that SQL Database and Azure portal provide. Instead of monitoring and tuning your database manually, you might consider delegating some of the monitoring and tuning actions to SQL Database using [automatic tuning](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-automatic-tuning). SQL Database automatically apply recommendations, tests, and verifies each of its tuning actions to ensure the performance keeps improving. This way, SQL Database automatically adapts to your workload in controlled and safe way. Automatic tuning means that the performance of your database is carefully monitored and compared before and after every tuning action, and if the performance doesn’t improve, the tuning action is reverted.

Today, many of our partners running [SaaS multi-tenant apps](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-design-patterns-multi-tenancy-saas-applications) on top of SQL Database are relying on automatic performance tuning to make sure their applications always have stable and predictable performance. For them, this feature tremendously reduces the risk of having a performance incident in the middle of the night. In addition, since part of their customer base also uses SQL Server, they are using the same indexing recommendations provided by SQL Database to help their SQL Server customers.

There are two automatic tuning aspects that are [available in SQL Database](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-automatic-tuning):

* **Automatic index management**: Identifies indexes that should be added in your database, and indexes that should be removed.
* **Automatic plan correction**: Identifies problematic plans and fixes SQL plan performance problems (coming soon, already available in SQL Server 2017).

**Adaptive query processing**

We are also adding the [adaptive query processing](https://docs.microsoft.com/en-us/sql/relational-databases/performance/adaptive-query-processing) family of features to SQL Database, including interleaved execution for multi-statement table-valued functions, batch mode memory grant feedback, and batch mode adaptive joins. Each of these adaptive query processing features applies similar “learn and adapt” techniques, helping further address performance issues related to historically intractable query optimization problems.

**Intelligent threat detection**

[SQL Threat Detection](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-threat-detection) leverages [SQL Database auditing](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-auditing) to continuously monitor Azure SQL databases for potentially harmful attempts to access sensitive data. SQL threat detection provides a new layer of security, which enables customers to detect and respond to potential threats as they occur by providing security alerts on anomalous activities. Users receive alerts upon suspicious database activities, potential vulnerabilities, and SQL injection attacks, and anomalous database access patterns. SQL threat detection alerts provide details of suspicious activity and recommend action on how to investigate and mitigate the threat. Users can explore the suspicious events to determine if the event results from an attempt to access, breach, or exploit data in the database. Threat detection makes it simple to address potential threats to the database without the need to be a security expert or manage advanced security monitoring systems.

**Advanced security and compliance**

SQL Database provides a range of [built-in security and compliance features](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-security-overview) to help your application meet various security and compliance requirements.

**Auditing for compliance and security**

[SQL Database Auditing](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-auditing) tracks database events and writes them to an audit log in your Azure storage account. Auditing can help you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.

**Data encryption at rest**

SQL Database [transparent data encryption](https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption-azure-sql) helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. Starting in May 2017, all newly created Azure SQL databases are automatically protected with transparent data encryption (TDE). TDE is SQL’s proven encryption-at-rest technology that is required by many compliance standards to protect against theft of storage media. Customers can manage the TDE encryption keys and other secrets in a secure and compliant manner using Azure Key Vault.

**Data encryption in motion**

SQL Database is the only database system to offer protection of sensitive data in flight, at rest and during query processing with [Always Encrypted](https://docs.microsoft.com/sql/relational-databases/security/encryption/always-encrypted-database-engine). Always Encrypted is an industry-first that offers unparalleled data security against breaches involving the theft of critical data. For example, with Always Encrypted, customers’ credit card numbers are stored encrypted in the database always, even during query processing, allowing decryption at the point of use by authorized staff or applications that need to process that data.

**Dynamic data masking**

[SQL Database dynamic data masking](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-dynamic-data-masking-get-started) limits sensitive data exposure by masking it to non-privileged users. Dynamic data masking helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal impact on the application layer. It’s a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed.

**Row-level security**

[Row-level security](https://docs.microsoft.com/sql/relational-databases/security/row-level-security) enables customers to control access to rows in a database table based on the characteristics of the user executing a query (such as by group membership or execution context). Row-level security (RLS) simplifies the design and coding of security in your application. RLS enables you to implement restrictions on data row access. For example ensuring that workers can access only those data rows that are pertinent to their department, or restricting a customer's data access to only the data relevant to their company.

**Azure Active Directory integration and multi-factor authentication**

SQL Database enables you to centrally manage identities of database user and other Microsoft services with [Azure Active Directory integration](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-aad-authentication). This capability simplified permission management and enhances security. Azure Active Directory supports [multi-factor authentication](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-ssms-mfa-authentication)(MFA) to increase data and application security while supporting a single sign-on process.

**Compliance certification**

SQL Database participates in regular audits and has been certified against several compliance standards. For more information, see the [Microsoft Azure Trust Center](https://azure.microsoft.com/support/trust-center/), where you can find the most current list of [SQL Database compliance certifications](https://azure.microsoft.com/support/trust-center/services/).

**Easy-to-use tools**

SQL Database makes building and maintaining applications easier and more productive. SQL Database allows you to focus on what you do best: building great apps. You can manage and develop in SQL Database using tools and skills you already have.

* [**The Azure portal**](https://portal.azure.com/): A web-based application for managing all Azure services
* [**SQL Server Management Studio**](https://docs.microsoft.com/sql/ssms/download-sql-server-management-studio-ssms): A free, downloadable client application for managing any SQL infrastructure, from SQL Server to SQL Database
* [**SQL Server Data Tools in Visual Studio**](https://docs.microsoft.com/sql/ssdt/download-sql-server-data-tools-ssdt): A free, downloadable client application for developing SQL Server relational databases, Azure SQL databases, Integration Services packages, Analysis Services data models, and Reporting Services reports.
* [**Visual Studio Code**](https://code.visualstudio.com/docs): a free, downloadable, open source, code editor for Windows, macOS, and Linux that supports extensions, including the [mssql extension](https://aka.ms/mssql-marketplace) for querying Microsoft SQL Server, Azure SQL Database, and SQL Data Warehouse.

SQL Database supports building applications with Python, Java, Node.js, PHP, Ruby, and .NET on the MacOS, Linux, and Windows. SQL Database supports the same [connection libraries](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-libraries) as SQL Server.

**Engage with the SQL Server engineering team**

* [DBA Stack Exchange](https://dba.stackexchange.com/questions/tagged/sql-server): Ask database administration questions
* [Stack Overflow](http://stackoverflow.com/questions/tagged/sql-server): Ask development questions
* [MSDN Forums](https://social.msdn.microsoft.com/Forums/en-US/home?category=sqlserver): Ask technical questions
* [Microsoft Connect](https://connect.microsoft.com/SQLServer/Feedback): Report bugs and request feature
* [Reddit](https://www.reddit.com/r/SQLServer/): Discuss SQL Server