# Frequently asked questions about DocumentDB

## Database questions about Microsoft Azure DocumentDB fundamentals

### What is Microsoft Azure DocumentDB?

Microsoft Azure DocumentDB is a highly-scalable NoSQL document database-as-a-service that offers rich querying over schema-free data, helps deliver configurable and reliable performance, and enables rapid development, all through a managed platform backed by the power and reach of Microsoft Azure. DocumentDB is the right solution for web, mobile, gaming and IoT applications when predictable throughput, low latency, and a schema-free data model are key requirements. DocumentDB delivers schema flexibility and rich indexing via a native JSON data model, and includes multi-document transactional support with integrated JavaScript.

For more database questions, answers, and instructions on deploying and using this service, see the [DocumentDB documentation page](https://azure.microsoft.com/documentation/services/documentdb/).

### What kind of database is DocumentDB?

DocumentDB is a NoSQL document oriented database that stores data in JSON format. DocumentDB supports nested, self-contained-data structures that can be queried through a rich DocumentDB [SQL query grammar](/documentation/articles/documentdb-sql-query). DocumentDB provides high performance transactional processing of server side JavaScript through [stored procedures, triggers, and user defined functions](/documentation/articles/documentdb-programming). The database also supports developer tunable consistency levels with associated [performance levels](/documentation/articles/documentdb-performance-levels).

### Do DocumentDB databases have tables like a relational database (RDBMS)?

No, DocumentDB stores data in collections of JSON documents. For information on DocumentDB resources, see [DocumentDB resource model and concepts](/documentation/articles/documentdb-resources).

### Do DocumentDB databases support schema-free data?

Yes, DocumentDB allows applications to store arbitrary JSON documents without schema definition or hints. Data is immediately available for query through the DocumentDB SQL query interface.

### Does DocumentDB support ACID transactions?

Yes, DocumentDB supports cross-document transactions expressed as JavaScript stored procedures and triggers. Transactions are scoped to a single partition within each collection and executed with ACID semantics as all or nothing isolated from other concurrently executing code and user requests. If exceptions are thrown through the server side execution of JavaScript application code, the entire transaction is rolled back.

### What are the typical use cases for DocumentDB?

DocumentDB is a good choice for new web, mobile, gaming and IoT applications where automatic scale, predictable performance, fast order of millisecond response times, and the ability to query over schema-free data is important. DocumentDB lends itself to rapid development and supporting the continuous iteration of application data models. Applications that manage user generated content and data are [common use cases for DocumentDB](/documentation/articles/documentdb-use-cases).

### How does DocumentDB offer predictable performance?

A Request Unit (RU) is the measure of throughput in DocumentDB. 1 RU corresponds to the throughput of the GET of a 1KB document. Every operation in DocumentDB, including reads, writes, SQL queries, and stored procedure executions has a deterministic Request Unit value based on the throughput required to complete the operation. Instead of thinking about CPU, IO and memory and how they each impact your application throughput, you can think in terms of a single Request Unit measure.

Each DocumentDB collection can be reserved with provisioned throughput in terms of Request Units of throughput per second. For applications of any scale, you can benchmark individual requests to measure their request unit values, and provision collections to handle the sum total of request units across all requests. You can also scale up or scale down your collection’s throughput as the needs of your application evolve. For more information about request units and for help determining your collection needs, please read [Manage Performance and Capacity](/documentation/articles/documentdb-manage).

### Is DocumentDB HIPAA compliant?

Yes, DocumentDB is HIPAA compliant. HIPAA establishes requirements for the use, disclosure, and safeguarding of individually identifiable health information. For more information, see the [Microsoft Trust Center](https://www.microsoft.com/en-us/TrustCenter/Compliance/HIPAA).

### What are the storage limits of DocumentDB?

There is no theoretical limit to the total amount of data that a collection can store in DocumentDB. If you would like to store over 250 GB of data within a single collection, please [contact support](/documentation/articles/documentdb-increase-limits) to to have your account quota increased.

### What are the throughput limits of DocumentDB?

There is no theoretical limit to the total amount of throughput that a collection can support in DocumentDB, if your workload can be distributed roughly evenly among a sufficiently large number of partition keys. If you wish to exceed 250,000 request units/second per collection or account, please [contact support](/documentation/articles/documentdb-increase-limits) to to have your account quota increased.

### How much does Microsoft Azure DocumentDB cost?

Please refer to the [DocumentDB pricing details](https://azure.microsoft.com/pricing/details/documentdb/) page for details. DocumentDB usage charges are determined by the number of collections in use, the number of hours the collections were online, and the consumed storage and provisioned throughput for each collection.

### Is there a free account available?

If you are new to Azure, you can sign up for an [Azure free account](https://azure.microsoft.com/pricing/free-trial/), which gives you 30 days and $200 to try all the Azure services. Or, if you have an Visual Studio subscription, you are eligible for [$150 in free Azure credits per month](https://azure.microsoft.com/pricing/member-offers/msdn-benefits-details/) to use on any Azure service.

### How can I get additional help with DocumentDB?

If you need any help, please reach out to us on [Stack Overflow](http://stackoverflow.com/questions/tagged/azure-documentdb), the [Azure DocumentDB MSDN Developer Forums](https://social.msdn.microsoft.com/forums/azure/home?forum=AzureDocumentDB), or schedule a [1:1 chat with the DocumentDB engineering team](http://www.askdocdb.com/). To stay up to date on the latest DocumentDB news and features, follow us on [Twitter](https://twitter.com/DocumentDB).

## Set up Microsoft Azure DocumentDB

### How do I sign-up for Microsoft Azure DocumentDB?

Microsoft Azure DocumentDB is available in the [Azure Portal](https://portal.azure.com). First you must sign up for a Microsoft Azure subscription. Once you sign up for a Microsoft Azure subscription, you can add a DocumentDB account to your Azure subscription. For instructions on adding a DocumentDB account, see [Create a DocumentDB database account](/documentation/articles/documentdb-create-account).

### What is a master key?

A master key is a security token to access all resources for an account. Individuals with the key have read and write access to the all resources in the database account. Use caution when distributing master keys. The primary master key and secondary master key are available in the **Keys** blade of the [Azure Portal](https://portal.azure.com). For more information about keys, see [View, copy, and regenerate access keys](/documentation/articles/documentdb-manage-account#keys).

### How do I create a database?

You can create databases using the Azure Portal as described in [Create a DocumentDB database](/documentation/articles/documentdb-create-database), one of the [DocumentDB SDKs](/documentation/articles/documentdb-dotnet), or through the [REST APIs](https://msdn.microsoft.com/library/azure/dn781481.aspx).

### What is a collection?

A collection is a container of JSON documents and the associated JavaScript application logic. A collection is a billable entity, where the [cost](/documentation/articles/documentdb-performance-levels) is determined by the performance level associated with the collection. Collections can span one or more partitions/servers and can scale to handle practically unlimited volumes of storage or throughput.

Collections are also the billing entities for DocumentDB. Each collection is billed hourly based on the provisioned throughput and the storage space used. For more information, see [DocumentDB pricing](https://azure.microsoft.com/pricing/details/documentdb/).

### How do I set up users and permissions?

You can create users and permissions using one of the [DocumentDB SDKs](/documentation/articles/documentdb-dotnet) or through the [REST APIs](https://msdn.microsoft.com/library/azure/dn781481.aspx).

## Database questions about developing against Microsoft Azure DocumentDB

### How to do I start developing against DocumentDB?

[SDKs](/documentation/articles/documentdb-dotnet) are available for .NET, Python, Node.js, JavaScript, and Java. Developers can also leverage the [RESTful HTTP APIs](https://msdn.microsoft.com/library/azure/dn781481.aspx) to interact with DocumentDB resources from a variety of platforms and languages.

Samples for the DocumentDB [.NET](/documentation/articles/documentdb-dotnet-samples), [Java](https://github.com/Azure/azure-documentdb-java), [Node.js](/documentation/articles/documentdb-nodejs-samples), and [Python](/documentation/articles/documentdb-python-samples) SDKs are available on GitHub.

### Does DocumentDB support SQL?

The DocumentDB SQL query language is an enhanced subset of the query functionality supported by SQL. The DocumentDB SQL query language provides rich hierarchical and relational operators and extensibility via JavaScript based user defined functions (UDFs). JSON grammar allows for modeling JSON documents as trees with labels as the tree nodes, which is used by both the DocumentDB automatic indexing techniques as well as the SQL query dialect of DocumentDB. For details on how to use the SQL grammar, please see the [Query DocumentDB](/documentation/articles/documentdb-sql-query) article.

### What are the data types supported by DocumentDB?

The primitive data types supported in DocumentDB are the same as JSON. JSON has a simple type system that consists of Strings, Numbers (IEEE754 double precision), and Booleans - true, false, and Nulls. More complex data types like DateTime, Guid, Int64, and Geometry can be represented both in JSON and DocumentDB through the creation of nested objects using the { } operator and arrays using the [ ] operator.

### How does DocumentDB provide concurrency?

DocumentDB supports optimistic concurrency control (OCC) through HTTP entity tags or ETags. Every DocumentDB resource has an ETag, and DocumentDB clients include their latest read version in write requests. If the ETag is current, the change is committed. If the value has been changed externally, the server rejects the write with a “HTTP 412 Precondition failure” response code. Clients must read the latest version of the resource and retry the request.

### How do I perform transactions in DocumentDB?

DocumentDB supports language-integrated transactions via JavaScript stored procedures and triggers. All database operations inside scripts are executed under snapshot isolation scoped to the collection if it is a single-partition collection, or documents with the same partition key value within a collection, if the collection is partitioned. A snapshot of the document versions (ETags) is taken at the start of the transaction and committed only if the script succeeds. If the JavaScript throws an error, the transaction is rolled back. See [DocumentDB server-side programming](/documentation/articles/documentdb-programming) for more details.

### How can I bulk insert documents into DocumentDB?

There are three ways to bulk insert documents into DocumentDB:

* The data migration tool, as described in [Import data to DocumentDB](/documentation/articles/documentdb-import-data).
* Document Explorer in the Azure Portal, as described in [Bulk add documents with Document Explorer](/documentation/articles/documentdb-view-json-document-explorer#BulkAdd).
* Stored procedures, as described in [DocumentDB server-side programming](/documentation/articles/documentdb-programming).

### Does DocumentDB support resource link caching?

Yes, because DocumentDB is a RESTful service, resource links are immutable and can be cached. DocumentDB clients can specify an “If-None-Match” header for reads against any resource like document or collection and update their local copies only when the server version has change.