MongoDB Features

Learn more about MongoDB Atlas

MongoDB is a scalable, flexible NoSQL document database platform designed to overcome the relational databases approach and the limitations of other NoSQL solutions. MongoDB products are well known for their horizontal scaling and load balancing capabilities, which has given application developers an unprecedented level of flexibility and scalability.

MongoDB Atlas is the leading global cloud database service for modern applications. Using Atlas, developers can deploy fully managed cloud databases across AWS, Azure, or Google Cloud. Best-in-class data security and privacy standards practices means that developers can rest easy knowing that they have instant access to the availability, scalability, and compliance they require for enterprise-level application development.

As of 2020, MongoDB has been downloaded over 30 million times with over 730,000 MongoDB University registrations. There are drivers for 10+ languages, with dozens more added by the community. Best of all, MongoDB is completely free to use.

MongoDB provides developers with a number of useful out-of-the-box capabilities, whether you need to run privately on site or in the public cloud.

Let's take a look at MongoDB's top five technical features:

1. Ad-hoc queries for optimized, real-time analytics

When designing the schema of a database, it is impossible to know in advance all the queries that will be performed by end users. An ad hoc query is a short-lived command whose value depends on a variable. Each time an ad hoc query is executed, the result may be different, depending on the variables in question.

Optimizing the way in which ad-hoc queries are handled can make a significant difference at scale, when thousands to millions of variables may need to be considered. This is why MongoDB, a document-oriented, flexible schema database, stands apart as the cloud database platform of choice for enterprise applications that require real-time analytics. With ad-hoc query support that allows developers to update ad-hoc queries in real time, the improvement in performance can be game-changing.

MongoDB supports field queries, range queries, and regular expression searches. Queries can return specific fields and also account for user-defined functions. This is made possible because MongoDB indexes BSON documents and uses the MongoDB Query Language (MQL).

2. Indexing appropriately for better query executions

In our experience, the number one issue that many technical support teams fail to address with their users is indexing. Done right, indexes are intended to improve search speed and performance. A failure to properly define appropriate indices can and usually will lead to a myriad of accessibility issues, such as problems with query execution and load balancing.

Without the right indices, a database is forced to scan documents one by one to identify the ones that match the query statement. But if an appropriate index exists for each query, user requests can be optimally executed by the server. MongoDB offers a broad range of indices and features with language-specific sort orders that support complex access patterns to datasets.

Notably, MongoDB indices can be created on demand to accommodate real-time, ever-changing query patterns and application requirements. They can also be declared on any field within any of your documents, including those nested within arrays.

Learn more about MongoDB indexing

Best Practices for MongoDB Indexing

3. Replication for better data availability and stability

When your data only resides in a single database, it is exposed to multiple potential points of failure, such as a server crash, service interruptions, or even good old hardware failure. Any of these events would make accessing your data nearly impossible.

Replication allows you to sidestep these vulnerabilities by deploying multiple servers for disaster recovery and backup. Horizontal scaling across multiple servers that house the same data (or shards of that same data) means greatly increased data availability and stability. Naturally, replication also helps with load balancing. When multiple users access the same data, the load can be distributed evenly across servers.

In MongoDB, replica sets are employed for this purpose. A primary server or node accepts all write operations and applies those same operations across secondary servers, replicating the data. If the primary server should ever experience a critical failure, any one of the secondary servers can be elected to become the new primary node. And if the former primary node comes back online, it does so as a secondary server for the new primary node.

Learn more about MongoDB replication

4. Sharding

When dealing with particularly large datasets, sharding–the process of splitting larger datasets across multiple distributed collections, or “shards”–helps the database distribute and better execute what might otherwise be problematic and cumbersome queries. Without sharding, scaling a growing web application with millions of daily users is nearly impossible.

Like replication via replication sets, sharding in MongoDB allows for much greater horizontal scalability. Horizontal scaling means that each shard in every cluster houses a portion of the dataset in question, essentially functioning as a separate database. The collection of distributed server shards forms a single, comprehensive database much better suited to handling the needs of a popular, growing application with zero downtime.

All operations in a sharding environment are handled through a lightweight process called mongos. Mongos can direct queries to the correct shard based on the shard key. Naturally, proper sharding also contributes significantly to better load balancing.

Learn more about MongoDB sharding

Best Practices for MongoDB Sharding

5. Load balancing

At the end of the day, optimal load balancing remains one of the holy grails of large-scale database management for growing enterprise applications. Properly distributing millions of client requests to hundreds or thousands of servers can lead to a noticeable (and much appreciated) difference in performance.

Fortunately, via horizontal scaling features like replication and sharding, MongoDB supports large-scale load balancing. The platform can handle multiple concurrent read and write requests for the same data with best-in-class concurrency control and locking protocols that ensure data consistency. There’s no need to add an external load balancer–MongoDB ensures that each and every user has a consistent view and quality experience with the data they need to access