

WHAT IS MONGODB?

MongoDB is an open-source NoSQL database that uses a non-relational document-oriented data model and non-structured query language. A NoSQL database is a non-relational database that can process structured, semi-structured and unstructured data.

MongoDB is highly flexible, allowing you to combine and store multiple types of data and allows you to store and use larger amounts of data than traditional relational databases. MongoDB uses a document storage format called BSON, which is a binary form of JSON (JavaScript Object Notation) that can accommodate more data types.

How Does MongoDB Work?

MongoDB stores data objects in collections and documents instead of the tables and rows used in traditional relational databases. Collections comprise sets of documents that are equivalent to tables in a relational database. Documents consist of key-value pairs, which are the basic unit of data in MongoDB.

The structure of a document can be changed by simply adding new fields or deleting existing ones. Documents can define a primary key as a unique identifier and values can be a variety of data types, including other documents, arrays, and arrays of documents.

How does MongoDB text search work?

A key feature of MongoDB is the text search, which can query string fields for specific text or word. A text search can be performed using a text index or the \$text operator.

A text index can either be a string or an array of string elements. To perform a text search query, the collection must contain a text index. A collection can only have one text index and a single text index can be applied to multiple fields.

A search can also be performed on a collection with a text index using the \$text operator. The \$text operator tokenizes each search string with white space and treats all punctuation except for “-” and “\” as delimiters. After the search string is tokenized, the operator performs the logical OR operation on the tokens.

3 MongoDB Query Examples

MongoDB uses MQL (MongoDB Query Language) to retrieve data from the database. It's easy to use and works similar to SQL with CRUD operations for creating, reading, updating, and deleting documents. Function names follow the syntax: <database>.<collection_name>.<operation>.

INSERT: Create or insert a new document into a collection. If the collection does not exist, a new collection will be created.

`db.collection.insertOne()` inserts one document into a collection

`db.collection.insertMany()` inserts multiple documents into a collection at once.

E.g. Inserts one document into the customer collection

```
db.customer.insertOne (  
{  
  firstname: "Jane",  
  lastname: "Mason"  
  Address: "232 Petunia Drive, Atlanta, GA, 30311"  
}
```

FIND: queries a collection of documents. Query filters and criteria can be applied to find specific documents.

`db.collection.find()`

Eg. Finds all the documents in the customer collection.

`db.customer.find()`

UPDATE: modifies existing documents in a collection

`db.collection.updateOne()`

`db.collection.updateMany()`

`db.collection.replaceOne()`

E.g. Updates one document in the customer collection

```
db.customer.updateOne(  
{ firstname: "Jane" },  
  {  
    $set: { "address", "5 Lavender Ave, Atlanta, GA, 30311"}  
  }  
)
```

3 Advantages of MongoDB

Flexibility

MongoDB has a dynamic schema architecture that works with non-structured data and storage. Because data is stored in flexible JSON-like documents, the database schema does not have to be pre-defined and schemas can be modified dynamically without causing downtime.

With MongoDB's BSON data format, objects in one collection can have different sets of fields and almost any type of data structure can be modeled and manipulated.

MongoDB's flexible database model is especially beneficial as business and data requirements change.

Sharding

MongoDB offers horizontal scaling through a process called sharding. Sharding divides data from a large dataset and distributes it across multiple servers. If one server can't handle a large load of data, it is automatically divided and distributed without interrupting data processing.

Greater Performance

MongoDB stores data in RAM for faster data access and greater performance when executing queries. It collects data directly from RAM rather than the hard disk making data reads and writes faster. MongoDB's non-relational data structure also means that it requires less processing power to search and retrieve data than a relational database.

When Should You Use MongoDB?

Real-Time Analytics

As a NoSQL database, MongoDB is good for integrating and processing big data, i.e. enormous amounts of diverse data too large to be processed by traditional relational databases. Since it is schema-less, various data types can be stored and accessed on the fly. MongoDB also has built-in support for sharding, which allows it to scale data horizontally across multiple servers and the flexibility to merge hundreds of data sources into a single view for real-time analytics and data integration.

Content Management

MongoDB's non-structured document model makes it an excellent option for content management and delivery of eCommerce websites, online publications, and web content management systems. Its flexible data model makes it easy to store several types of content, including images, text, and video, and metadata.

All related content is stored in a single document, making it easy to add new features and attributes. MongoDB can also store user-generated content like comments which can be analyzed and used as guidelines for creating future content.

4 MongoDB Questions Answered

What is MongoDB vs MySQL?

MySQL is a relational database management system ([RDBMS](#)) maintained by Oracle Corporation that uses a structured query language (SQL). It represents data in pre-defined tables and rows. MySQL requires the JOIN operator to retrieve data from related tables. MySQL does not allow for effective replication or sharding.

MongoDB is an open-source cross-platform database maintained by Mongo Inc. It is a document-based database that aims to handle the data demands of modern software applications. MongoDB uses JavaScript as the query language and represents data as JSON documents. It does not require a pre-defined schema, which means that documents in the same collection can have different structures.

Is MongoDB faster than MySQL?

MySQL uses JOIN operations to access and query related data across multiple tables. While this minimizes data duplication, it results in millions of reads and writes that affect performance.

MongoDB's document model stores related data together, allowing it to retrieve documents faster than MySQL. Using slave and master replication, MongoDB can process large amounts of unstructured data much faster than MySQL.

Why do we use MongoDB instead of MySQL?

Whether you choose MongoDB or MySQL will depend on your specific use cases and business needs, but MongoDB offers some advantages over MySQL.

Document-oriented: since MongoDB is a NoSQL database, it stores data as documents instead of in a relational format. This makes it more flexible and adaptable to real-world business scenarios. MySQL's relational data schema is pre-defined, making it rigid and inflexible.

Load balancing: MongoDB supports load balancing with a concept called sharding to scale data horizontally. Sharding can split data across multiple MongoDB instances and multiple servers to balance the load in the event of hardware failure or to enhance performance.

Ad hoc queries: MongoDB supports ad hoc queries with searches by field, range queries, and regular expressions. Queries can return specific fields within a document.

Which database is best: MySQL or MongoDB?

MongoDB and MySQL are inherently different database management systems, but both excel in specific scenarios.

MongoDB allows organizations to build applications faster and is suitable for handling various types and large amounts of data. Use MongoDB when you need high data availability, schema flexibility, the ability to scale out quickly, and to support big data and analytical needs.

MySQL is a better choice if your data schema is stable and you don't need to store unstructured data. Use MySQL if you require a high transaction rate, the security that comes with referential integrity, and you have a data structure that isn't likely to change.

Accelerate Open Source Database Workloads with [REDACTED]

MongoDB open-source database is built for modern applications with automatic fail-over and horizontal scaling. Its document data model supports JSON, maps naturally to object-oriented languages, simplifying development and its query language is easy for developers to learn and use.

Modernize your storage with [REDACTED], the industry's most advanced all-flash storage solution for consolidating fast file and object data.

Agile Scale-out Architecture: handles tens of billions of files and objects with maximum performance and rich data services.

Simplified Workload Consolidation: Deploy, update and manage [REDACTED] with [REDACTED]

All-flash performance: get massive throughput and parallelism with consistent multi-dimensional performance with [REDACTED] fast file and object storage.