

«What's so cool about MongoDB?»

Patrick P. Bucher

Summary

Unlike relational databases, the NoSQL database MongoDB stores its information in documents rather than rows. A document is a JSON data structure, consisting of nested keys and values, and, unlike relational databases, it does *not* need to fit a certain schema.

MongoDB has its own shell that can be used to query the documents interactively. This JavaScript-like query language is designed to manipulate structured documents — unlike SQL, which operates in tabular terms.

The lack of a schema can make it more difficult to keep the data clean, but on the other side makes it easier and faster to develop applications requiring persistence.

MongoDB stores the JSON data structures internally in a binary representation called BSON: binary JSON. Documents belonging together are grouped as *collections*, unlike relational databases, which store similar records in tables.

Whereas relational databases require normalization, and thus splitting up data over multiple tables, MongoDB can store nested data structures as a single document. The form of the data is not prescribed in the database system itself, but in the application code.

Since queries in MongoDB all come down to a search for certain values, an index is crucial for a good query performance. MongoDB uses B-tree data structures to store indices: an approach similar to relational databases. MongoDB also supports primary keys as a unique identifier.

MongoDB provides database replication across multiple computers (nodes) for redundancy and automatic failover. For read-intensive applications, as web applications often are, replication can enhance query performance. A primary node (master) can be used to both read and write data, whereas the other nodes (slaves) only support read operations. In case of failure of the master node, a slave node is elected as the new master in an automatic failover process.

One of MongoDB's main advantages is that, unlike relational databases, it supports horizontal scaling, i.e. the distribution of one single database across many systems. Horizontal scaling can be achieved by sharding, which can be done using different policies: hashed, ranged, or zoned. The data is automatically and fully transparent balanced for elasticity.

With a feature called *workflow isolation*, MongoDB instances can be tagged for a special purpose, say, analytics or machine learning. Those instances can be configured in a way to not affect the performance of the operational instances, and do their heavy lifting when the operational instances are idle.

Source

Banker, K. et al (2016). *MongoDB in Action* (Second Edition). Manning.

Questions and Answers

1. How can relationships between documents be modeled in MongoDB?
 - Embedded Documents: for subordinate data structures.
 - Document Reference: using an `ObjectId` attribute (referring the other document's `_id` attribute).
2. Does the lack of a schema require more validation code in the application?
 - MongoDB offers features for data integrity known from relational data bases, such as primary and foreign keys, which guarantee that a record is there. So there's no need to write special validation code in the most cases.
 - JSON Schema, which enforces a schema structure (attributes must exist, rules using regular expressions, etc.) to the documents in a collection, is a common mitigation.
 - Oftentimes the document structure changes a lot during development, and a JSON Schema is applied only later in the process.
3. Who uses MongoDB for big data sets?
 - Lyft
 - InVision
 - Bosch
4. Are there some typical beginner's mistakes when working with MongoDB?
 - Some beginners get the notion of a «document» wrong and think of a PDF or Word document rather than a JSON data structure.
 - For beginners familiar with SQL and relational databases, it is sometimes hard to let go of the relational mind set.
 - Getting the granularity of the documents right requires experience: Huge documents lead to a bad performance and are overwhelming to deal with, while too fine-grained documents that make heavy use of references instead of embedding subdocuments require a lot of joins and are inconvenient in development.
5. Is it a common task to restructure existing data in a MongoDB instance?
 - Small structural changes, such as changing data types, can be done on the fly.
 - Bigger restructurings of existing databases are usually done offline.
 - A good initial design of the data structures is crucial, because restructuring is usually painful and expensive.