

# The SQL vs NoSQL Difference: MySQL vs MongoDB



Xplenty

[Follow](#)

Sep 28, 2017 · 5 min read



When it comes to choosing a database, one of the biggest decisions is picking a relational (SQL) or non-relational (NoSQL) data structure. While both are viable options, there are certain key differences between the two that users must keep in mind when making a decision.

Here, we break down the most important distinctions and discuss two of the key players in the relational vs non-relational debate: MySQL and MongoDB.

*Your Quora Question: Is MongoDB dead?*

## The Big Picture Differences

### The Language

Think of a town - we'll call it Town A - where everyone speaks the same language. All of the businesses are built around it, every form of communication uses it - in short, it's the only way that the residents understand and interact with the world around them. Changing that language in one place would be confusing and disruptive for everyone.

Now, think of another town, Town B, where every home can speak a different language. Everyone interacts with the world differently, and there's no "universal" understanding or set organization. If one home is different, it doesn't affect anyone else at all.

This helps illustrate one of the fundamental differences between SQL relational and NoSQL non-relational databases, and this distinction has big implications. Let's explain:

**SQL databases** use structured query language (SQL) for defining and manipulating data. On one hand, this is extremely powerful: SQL is one of the most versatile and widely-used options available, making it a safe choice and especially great for complex queries. On the other hand, it can be restrictive. SQL requires that you use predefined schemas to determine the structure of your data before you work with it. In addition, all of your data must follow the same structure. This can require significant up-front preparation, and, as with Town A, it can mean that a change in the structure would be both difficult and disruptive to your whole system.

A **NoSQL database**, on the other hand, has dynamic schema for unstructured data, and data is stored in many ways: it can be column-



oriented, document-oriented, graph-based or organized as a KeyValue store. This flexibility means that:

- You can create documents without having to first define their structure
- Each document can have its own unique structure
- The syntax can vary from database to database, and
- You can add fields as you go.

### **The Scalability**

In most situations, SQL databases are vertically scalable, which means that you can increase the load on a single server by increasing things like CPU, RAM or SSD. NoSQL databases, on the other hand, are horizontally scalable. This means that you handle more traffic by sharding, or adding more servers in your NoSQL database. It's like adding more floors to the same building versus adding more buildings to the neighborhood. The latter can ultimately become larger and more powerful, making NoSQL databases the preferred choice for large or ever-changing data sets.

### **The Structure**

SQL databases are table-based, while NoSQL databases are either document-based, key-value pairs, graph databases or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions - such as an accounting system - or for legacy systems that were built for a relational structure.

Some examples of SQL databases include MySQL, Oracle, PostgreSQL, and Microsoft SQL Server. NoSQL database examples include MongoDB, BigTable, Redis, RavenDB Cassandra, HBase, Neo4j and CouchDB.

## **SQL vs NoSQL: MySQL vs MongoDB**

Now that we've established the key structural differences between SQL and NoSQL databases, let's delve into the key functional differences between the two, looking specifically at MySQL and MongoDB as examples.

### MySQL: The SQL Relational Database

The following are some MySQL benefits and strengths:

- **Maturity:** MySQL is an extremely established database, meaning that there's a huge community, extensive testing and quite a bit of stability.
- **Compatibility:** MySQL is available for all major platforms, including Linux, Windows, Mac, BSD and Solaris. It also has connectors to languages like Node.js, Ruby, C#, C++ , Java, Perl, Python and PHP, meaning that it's not limited to SQL query language.
- **Cost-effective:** The database is open source and free.
- **Replicable:** The MySQL database can be replicated across multiple nodes, meaning that the workload can be reduced and the scalability and availability of the application can be increased.
- **Sharding:** While sharding cannot be done on most SQL databases, it can be done on MySQL servers. This is both cost-effective and good for business.

### MongoDB: The NoSQL Non-Relational Database

The following are some of MongoDB benefits and strengths:

- **Dynamic schema:** As mentioned, this gives you flexibility to change your data schema without modifying any of your existing data.
- **Scalability:** MongoDB is horizontally scalable, which helps reduce the workload and scale your business with ease.
- **Manageability:** The database doesn't require a database administrator. Since it is fairly user-friendly in this way, it can be used by both developers and administrators.
- **Speed:** It's high-performing for simple queries.
- **Flexibility:** You can add new columns or fields on MongoDB without affecting existing rows or application performance.

### So Which Database Is Right For Your Business?

MySQL is a strong choice for any business that will benefit from its pre-defined structure and set schemas. For example, applications that require multi-row transactions - like accounting systems or systems that monitor inventory - or that run on legacy systems will thrive with the MySQL structure.

MongoDB, on the other hand, is a good choice for businesses that have rapid growth or databases with no clear schema definitions. More specifically, if you cannot define a schema for your database, if you find yourself denormalizing data schemas, or if your schema continues to change - as is often the case with mobile apps, real-time analytics, content management systems, etc.- MongoDB can be a strong choice for you.

If you’re still uncertain about which type of database is best for your business, you can schedule a consultation with one of our experts today.

. . .

Xplenty's platform allows organizations to integrate, process, and prepare data for analytics on the cloud. With Xplenty, every company can have immediate connectivity to a variety of data stores and a rich set of out-of-the-box data transformation components.

. . .

Sign up for our newsletter

Join over 15,000 subscribers and get the best content on data integration and data analytics.

Email

Sign up

☐ I agree to leave Medium.com and submit this information, which will be collected and used according to [Upscribe's privacy policy](#).

