MYSQL VS MONGODB: COMPARING DATABASES



MySQL is one of the most popular database systems, and the one that almost everyone learns first. So, I decided to compare MongoDB with MySQL. In this article, I'll cover

- MySQL key features
- MongoDB key features
- Global use
- Data storage
- Common queries
- When to use Mongo
- Disadvantages of each

(This article is part of our MongoDB Guide. Use the right-hand menu to navigate.)

What is MySQL?

MySQL is a cross-platform database system written in C and C++ and maintained by Oracle Corporation. It's a relational database system based on SQL (Structured Query Language). MySQL was first released in 1995 and has come a long way, attracting a large user base. By now, it's a very stable product. The latest version, 8.0, was released in April 2020.

Key features of MySQL

In MySQL, you store data in tables, composed of rows and columns. That means there is a predefined schema for databases. Here are a few more key features of MySQL:

- Community-driven open-source database system with a GitHub repository
- Supports all major operating systems such as Windows, MacOS, and Linux
- Easy to use, fast, and secure
- Follows client/server architecture
- Supports multi-versioning concurrency control
- Supports multi-threading, allowing databases to be scalable
- Very flexible, as it supports a lot of embedded applications

Key features of MongoDB

MongoDB is a cross-platform database system written in C++ and maintained by Mongo Inc. MongoDB was released in 2009, targeting the most modern data handling demands of software applications. The product gets better with each release, allowing developers to make the most of it—and attracting more users every day.

Unlike MySQL, MongoDB a document-oriented NoSQL database system that consists of collections and documents. MongoDB is a schema-less database system, which means that documents in the same collection can have different structures. Here are a few more key features of MongoDB.

- Open-source database system with a GitHub repository
- Compatible with Windows, Mac OS, and Linux
- Supports ad hoc queries, including regular expression searches
- Uses replica sets to create multiple copies of the data
- The auto-election feature lets you set up a secondary database that automatically takes over if your primary database fails
- Sharding allows horizontal scaling, which is a better solution for large databases

Companies that use MySQL or MongoDB

These are some well-known companies and organizations use MySQL or MongoDB, illustrating the popularity of both. (I have a hunch many companies use both.)

MySQL	MongoDB
Airbnb	Nokia
Sony	Adobe
NASAU.S. Navy	Google
Netflix	eBay
YouTube	EA Sports
CERN	Cisco
Twitter	Verizon
Pearson	PayPal
	Pearson

Data storage in MongoDB and MySQL

MySQL stores data in tabular structures called tables. Instead of tables, MongoDB has collections.

Here is a row in a MySQL table, which is called a record:

name	university	age	address
John	UCSC	23	766/A, ABC Street, are DEF City

A row in a MySQL table

In MongoDB, each record is stored as a JSON-like document:

```
{
   name: 'John',
   university : 'UCSC',
   age : 23,
   address : {
                  no : '766/A',
                  street : 'ABC Street',
                  city : 'DEF City'
             }
}
```

A document in a MongoDB collection

Different documents in a single collection can have different structures, while every record in a MySQL table should have the same structure.

Common queries in MongoDB and MySQL

Now let's have a look at how queries are used in both databases. In each one, let's see how to do some basic operations:

- Selecting data
- Inserting data
- Updating data

You might notice that MongoDB queries are much more familiar to developers as they are written in JavaScript.

Select records in MySQL

```
SELECT * FROM employee
```

Select documents in MongoDB

```
db.employee.find()
```

Insert records in MySQL

```
INSERT INTO employee (emp_id, emp_name, role) VALUES ('emp001', 'Mike',
'Admin')
```

Insert documents in MongoDB

```
db.employee.insert({
    emp_id: 'emp001',
```

```
    name: 'mike',
    role: 'admin'
})
```

Update records in MySQL

```
UPDATE employee SET role = 'manager' WHERE name = 'Mike'
```

Update documents in MongoDB

```
db.employee.update(
{"name" : "mike"},
{$set: { "role" : "manager"}}
);
```

When to use MongoDB over MySQL

We already know that there are two types of database systems:

- Relational databases
- NoSQL databases

The differences between those database types are rooted in: how they are designed, how they store data, and the data types they support. To maximize your benefit, always select the most suitable database according to the goals and requirements of your project.

Now let's discuss scenarios in which it is more convenient to use MongoDB over MySQL.

1. If you need high availability of data and fast recovery of data as some of your main requirements, MongoDB is more suitable for your system.
2. If the database grows hefty in the future, MongoDB can facilitate horizontal scaling through the built-in sharding feature.
3. If the database schema is not fixed, you can reduce the schema migration costs by using MongoDB.
4. If most of the services in the system are cloud-based, it's better to use MongoDB because it is also a cloud-based service.

There's one more reason you might choose MongoDB over MySQL. A general distinction between these two databases is that MongoDB is much more developer-friendly than MySQL. Therefore, if you are not able or willing to hire a database administrator (DBA), you might want to use MongoDB for your application.

Disadvantages of each

Here are some common issues with MySQL:

- MySQL is not efficient when it comes to large databases.
- MySQL has performance issues when scaling databases.
- Because MySQL databases have a schema, they are vulnerable to SQL injection attacks.
- MySQL databases have a predefined structure. Changing the structure later can be costly.

- Working with rows and columns is more complicated than working with JSON documents.

Here are some common disadvantages of MongoDB:

- MongoDB doesn't support Joins like relational databases. That means you'll need to update the schema of the database frequently based on how you access data.
- As MongoDB stores keys for each value pair, the data redundancy results in high memory usage.
- The document size is limited to 16MB.
- MongoDB doesn't follow ACID property strictly, so handling complex transactions can be complicated.
- It's challenging to apply business logic at the database level because MongoDB doesn't contain a provision for stored procedure or functions.

Summary: MySQL vs MongoDB

Database performance can differ depending on:

- Database load
- Query patterns
- Data structure

For example, if you develop a system that processes a huge amount of transactions, like an accounting system, you should use MySQL. However, if you are going to build a real-time analytics system, MongoDB would be better. Therefore, I always recommend deciding which database management system to use based on the type of your software application.

Additional resources

For related reading, explore these resources:

- An Introduction to Database Reliability
- What Is a Database Reliability Engineer (DBRE)?
- PostgreSQL vs MongoDB: Comparing Databases
- MongoDB vs Cassandra: NoSQL Databases Compared
- DBMS: An Intro to Database Management Systems