Executive Report  
SQL vs NOSQL  
And what’s the difference?

short line

Big Money Inc.

TEL:867-5309

EMAIL:[bigmoney@hotmail.com](mailto:bigmoney@hotmail.com)

MEMBERS: David Thomsen // Micah Kezar

# Table of Contents

[**Abstract**](#_lzr2si3oshbv) **2**

[**Introduction**](#_77frxfedweqi) **2**

[**Overview of MongoDB**](#_1t1094rd0ngq) **2**

[**MongoDB Architecture**](#_akytff7touv5) **3**

[**Querying MongoDB**](#_5hfdv172vxpr) **3**

[**Example Images**](#_753orj25mxq4) **3**

[**Comparison to MySQL**](#_5tzgo0tfmp90) **3**

[Advantages of using MongoDB](#_rlmkspxv63pr) 3

[Disadvantages of using MongoDB](#_uy3np2v7np9n) 3

[**Conclusion**](#_f362znpkjw2r) **3**

[**Citations**](#_zekm2ghrzpy) **4**

# Abstract

NoSQL databases embrace either a flexible schema or a schemaless approach, contrary to its SQL counterpart. This allows for the storage of unstructured or partially structured data in various formats. This includes documents, key-value pairs, graphs, and wide-column stores. Having this level of versatility proves extremely useful in modern applications. This paper provides an overview of NoSQL databases, with a spotlight on MongoDB. Introduced in 2009, MongoDB gave users a way to flee from the tabular structure of relational databases, and instead came up with a document-based solution to give the option to have more flexible schemas.

This paper goes into detail about the document-oriented storage MongoDB uses: JavaScript Object Notation (JSON). Furthermore, it explores querying mechanisms with the MongoDB Query Language (MQL), as well as the various supported languages that can be used to access the MongoDB API.

The goal of this paper is to educate readers on the ever-growing potential of NoSQL databases such as MongoDB. We aim to deliver a clear and concise breakdown of basic query functions to demonstrate the ease-of-use NoSQL databases offer to developers. By comparing and contrasting the techniques used between relational and NoSQL databases, developers may have a more open understanding when deciding which path to choose to manage their datasets, and will have the ability to choose what is right for them.

short dash

# Introduction

What is a NoSQL Database? NoSQL is a term used to describe databases that hold and query data in different formats other than the traditional relational database. Contrary to traditional SQL databases, these “non-SQL” databases either have a flexible schema, or no schema at all. As a result, a user can store unstructured or partially structured data. This includes data formats such as documents, key-value pairs, graphs, and wide-column stores. The flexibility of allowed data types in NoSQL databases allow them to be well-suited for applications with evolving data.

Another big difference between NoSQL and SQL databases is their approach to query languages. While SQL databases primarily use the ANSI-SQL language standard for their query languages, NoSQL databases often use their own querying language tailored to the specific data models being used. For example, MongoDB uses its own Query Language called MongoDB Query Language (or MQL). This Query Language differs from the traditional ANSI-SQL standard.

Lastly, MongoDB allows for multiple programming languages to interact with the API. MongoDB *officially* supports 13 languages, such as the entire C suite, Java, Python, Node.js, PHP, Ruby, Rust, etc. The MongoDB community has also added support to various languages such as Elixr, Mongoose, Prisma, and R.

short dash

# Overview of MongoDB

MongoDB was introduced in 2009 as a way to address a few challenges surrounding the current pool of NoSQL databases. Such databases would either offer more flexibility in the data model, but would also remove a lot of important features associated with relational databases. MongoDB keeps the best parts of both relational and NoSQL databases while also providing a solution to meet the needs of applications used today.

MongoDB replaces the table format found in relational databases with a new structure. Instead of storing data in columns and rows, MongoDB uses document databases that can store data as JavaScript Object Notation (JSON). These documents can store any type of data, and fields can be added without affecting the other documents in a collection. As a result, The MongoDB equivalent of table creation outside of a relational database can be made.

MongoDB Developers have documented the three primary advantages to using document data. These advantages are Intuitiveness, Having a Flexible Schema, and Universality, as JSON documents are used everywhere. An example of how the JSON format structure being faster and easier for developers can be found below. This JSON shows a customer object with all of the parameters embedded as subdocuments and arrays. As a result, you are greatly condensing the amount of work needed. The below code would have been the relational database equivalent of seven separate parent-child tables linked by foreign keys.

{

“\_id”:

ObjectId(“5ad88534e3632e1a35a58d00”),

“name”: {

“first”: “John”,

“last”: “Doe” },

“address”: [

{ “location”: “work”,

“address”: {

“street”: “16 Hatfields”,

“city”: “London”,

“postal\_code”: “SE1 8DJ”},

“geo”: { “type”: “Point”, “coord”: [

51.5065752,-0.109081]}},

],

“phone”: [

{ “location”: “work”,

“number”: “+44-1234567890”},

],

“dob”: ISODate(“1977-04-01T05:00:00Z”),

“retirement\_fund”:

NumberDecimal(“1292815.75”)

}



short dash

# Querying MongoDB

Python is one of the most common as well as having a human-readable syntax. We have included a breakdown of code from [This Link](https://www.w3schools.com/python/python_mongodb_query.asp) in order to explain a very basic query function within Python, but support for many other languages is available [HERE](https://www.mongodb.com/resources/languages). With Python there is much more that can be done with all of this but this is the simplest way to break down a query to its main components.

import pymongo

#Imports the Python MongoDB package

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

#Create the Client Variable to connect to the MongoDB instance

mydb = myclient["mydatabase"]

#Grabs the database that is specified. Similar to "using Employees"

mycol = mydb["customers"]

#Grabs the Column that is specified.

myquery = { "address": "Park Lane 38" }

#The actual Query that the DB will receive

mydoc = mycol.find(myquery)

#Create a variable to store information from query

for x in mydoc:

print(x)

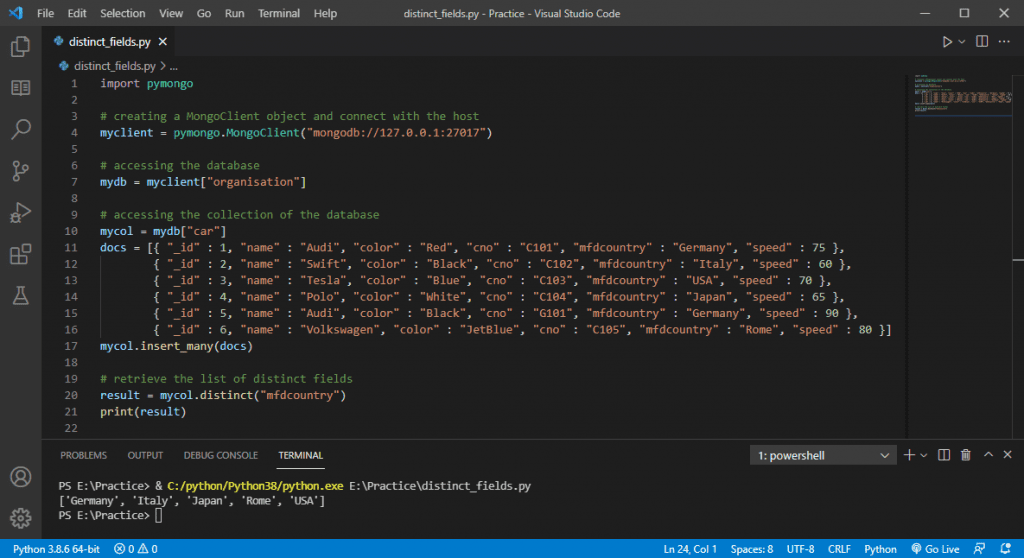
#Output data in the terminal for the user.



short dash

# 

# Example Image



This is an example from [Here](https://databasefaqs.com/distinct-query-in-mongodb/) showing the previous code running with a query as well. In this example, the database created was “Car” and provides much information about the car’s manufacturing information.

The example query uses a function mycol.distinct along with the field “mfdcountry” and the function returns each unique country value that was stored in the table.

short dash

# Comparison to MySQL

## Overview

Both MySQL and MongoDB are database management systems that are meant to create databases and store relational data. The way that both of these frameworks work, are very different however. MySQL, as mentioned above, uses the ANSI-SQL language standard for all operations within the system. Databases are created using the SQL language and the data is stored in tables that must be joined in order to create queries.

MongoDB takes a much different approach to how it manages and interacts with the data and databases. One key distinction lies in its data storage model. While MySQL stores data in tables with rows and columns, MongoDB stores data within JSON documents. Additionally, MongoDB officially supports multiple language options to assist with DB interaction, while a supportive community assists with the languages not supported by MongoDB.

## Advantages of using MongoDB

1. **Varying Schemas:** MongoDB allows the storing of documents of varying schemas, including unstructured data sets.
2. **High Performance:** The JSON format utilized by MongoDB allows for more efficient data retrieval, and faster read and write times.
3. **Better Query Language:** MongoDB’s JSON-based structure means you can leave the rigid SQL syntax of tables, columns, and rows behind. Instead, MongoDB allows documents within a collection to have varying structures.

## Disadvantages of using MongoDB

1. **Joining Database Data:** MongoDB does not support Database Joining the same way that MySQL does. Rather, it must be manually coded in the language of choice that is used to interface with MongoDB
2. **Sizing Limitations:** MongoDB limits document size to 16MB and nesting is only allowed to be 100 layers (which is a lot but still a limit nonetheless)
3. **Memory Usage:** Due to reliance on good indexing, there are often issues with high memory usage in comparison to an average MySQL database. This can cause many issues but requires additional attention to maintain.

short dash

# Conclusion

In conclusion, NoSQL databases provide a crucial solution to many problems and lack of features found within the current MySQL relational database. MongoDB has been deemed as one of the most popular and influential non-tabular databases with their document-oriented storage approach and very simple querying mechanisms. Having the ability to utilize the API with multiple programming languages proves useful to many developers across many industries. This paper opens the eyes of SQL developers to explore the world of NoSQL, and the many benefits and solutions it offers to many everyday problems accompanied with relational databases.

short dash

# 

# Citations

“MongoDB Architecture Guide.” MongoDB, 2024, [www.mongodb.com/collateral/mongodb-architecture-guide](http://www.mongodb.com/collateral/mongodb-architecture-guide).

“Comparing the Differences - MongoDB vs MySQL.” *MongoDB*, 2024, [www.mongodb.com/compare/mongodb-mysql](http://www.mongodb.com/compare/mongodb-mysql). Accessed 1 May 2024.

Juneja, Ishita. “Is MongoDB Query Language Similar to SQL?” *Medium*, 25 Aug. 2023, <https://medium.com/@ishitajuneja01/is-mongodb-query-language-similar-to-sql-486e0fa75dcb>. Accessed 1 May 2024.

“Start Developing with MongoDB — MongoDB Drivers.” Www.mongodb.com, 2024, [www.mongodb.com/docs/drivers/](http://www.mongodb.com/docs/drivers/).

“MongoDB Supported Languages.” *MongoDB*, 2024, [www.mongodb.com/resources/languages](http://www.mongodb.com/resources/languages). Accessed 1 May 2024.

“Understanding the Key Mongodb Pros and Cons | ThinkAutomation.” *ThinkAutomation*, [www.thinkautomation.com/our-two-cents/understanding-the-key-mongodb-pros-and-cons](http://www.thinkautomation.com/our-two-cents/understanding-the-key-mongodb-pros-and-cons).

short dash