

MONGO DB

Introduction to NoSQL Databases:

A database Management System provide the mechanism to store and retrieve the data. There are different kinds of database Management Systems:

1. RDBMS (Relational Database Management Systems)
2. OLAP (Online Analytical Processing)
3. NoSQL (Not only SQL)

In this guide, We will discuss NoSQL. NoSQL databases were created to overcome the limitations of relational databases.

What is a NoSQL database?

- NoSQL databases are different than relational databases like MySQL. In relational database you need to create the table, define schema, set the data types of fields etc before you can actually insert the data. In NoSQL you don't have to worry about that, you can insert, update data on the fly.
- One of the advantage of NoSQL database is that they are really easy to scale and they are much faster in most types of operations that we perform on database. There are certain situations where you would prefer relational database over NoSQL, however when you are dealing with huge amount of data then NoSQL database is your best choice.

Limitations of Relational databases:

1. In relational database we need to define structure and schema of data first and then only we can process the data.
2. Relational database systems provides consistency and integrity of data by enforcing ACID properties (Atomicity, Consistency, Isolation and Durability). There are some scenarios where this is useful like banking system. However in most of the other cases these properties are significant performance overhead and can make your database response very slow.
3. Most of the applications store their data in JSON format and RDBMS don't provide you a better way of performing operations such as create, insert, update, delete etc on this data. On the other hand NoSQL store their data in JSON format, which is compatible with most of the today's world application.

What are the advantages of NoSQL?

There are several advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability.

High scalability:

NoSQL database such as MongoDB uses sharding for horizontal scaling. Sharding is partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved. Vertical scaling means adding more resources to the existing machine while horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement, on the other hand horizontal scaling is easy to implement. Horizontal scaling database examples: MongoDB, Cassandra etc. Because of this feature NoSQL can handle huge amount of data, as the data grows NoSQL scale itself to handle that data in efficient manner.

High Availability:

Auto replication feature in MongoDB makes it highly available because in case of any failure data replicates itself to the previous consistent state.

Types of NoSQL database

Here are the types of NoSQL databases and the name of the databases system that falls in that category. MongoDB falls in the category of NoSQL document based database.

Key Value Store: Memcached, Redis, Coherence

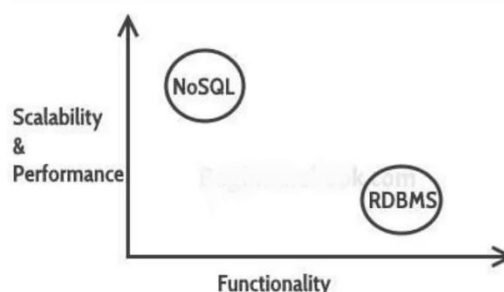
Tabular: Hbase, Big Table, Accumulo

Document based: MongoDB, CouchDB, Cloudant

RDBMS Vs NOSQL

RDBMS: It is a structured data that provides more functionality but gives less performance.

NoSQL: Structured or semi structured data, less functionality and high performance.



So when I say less functionality in NoSQL what's missing:

1. You can't have constraints in NoSQL.

2. Joins are not supported in NoSQL.

These supports actually hinders the scalability of a database, so while using NoSQL database like MongoDB, you can implements these functionalities at the application level.

When to go for NoSQL

When you would want to choose NoSQL over relational database:

1. When you want to store and retrieve huge amount of data.
2. The relationship between the data you store is not that important.
3. The data is not structured and changing over time.
4. Constraints and Joins support is not required at database level.
5. The data is growing continuously and you need to scale the database regular to handle the data.

What is MongoDB?

MongoDB is a opensource, crossplatform, document-oriented database program.

Classified as a NoSQL. Database program, MongoDB uses JSON-Like documents.

MongoDB is a learning NoSQL database.

Differences between Mongo DB & SQL:

SQL Database	NoSQL Database (MongoDB)
<ul style="list-style-type: none">• Is Relational database.• It supports SQL query language.• It is generally Table-based structure.• It follows Row-based structure.• It follows Column-based structure.• It supports foreign keys.• It supports triggers.• Schema is predefined.• Not good to use for hierarchical data storage.• Due to Vertically scalability-user can increase RAM.• Highlights on ACID properties (Atomicity, Consistency, Isolation and Durability).• Tools used are Microsoft SQL Server, PostgreSQL, MySQL, Oracle, etc	<ul style="list-style-type: none">• Is Non-relational database.• It supports JSON query language.• It is mostly Collection-based and key-value pair structure.• It follows Document-based structure.• It follows Field-based structure.• It does not support foreign keys.• It does not support triggers.• Schema is dynamic.• Good to use for hierarchical data storage.• Due to Horizontally scalability - user can add mote servers.• Highlights on CAP theorem• Tools used are MongoDB, Cassandra, CouchDB, Bigtable, FlockDB, ArangoDB, etc

Performance comparison between MongoDB and SQL

- **SQL databases** performance tuning consists of making queries of a relational database run as fast as possible. Indexing in data structure improves the speed of data retrieval operations on a database table by providing rapid random lookups and efficient access of ordered records. It's high-performing for complex queries.
- **MongoDB Performance** As you develop and operate applications with MongoDB, you may need to analyse the performance of the application and its database. When you come across tarnished performance, it is mostly due to database access strategies or hardware availability or the number of open database connections. It's high-performing for simple query.

MongoDB: which type of DB?

- **Mongo DB-Document Oriented NoSQL Database Approach**

In mongo DB, Querying on this model is easy, since the schema is de-normalized. No joins are required. So we plan to use the Mongo DB based solution.


Introduction to MongoDB

MongoDB is an open source, document oriented database that stores data in form of documents (key and value pairs). As discussed in our last tutorial (NoSQL, introduction) that document based databases are one of types of NoSQL databases.

What is a document?

If you came from a relational database background then you can think of them as rows in RDBMS. The mapping between relational database and MongoDB is covered in the next tutorial so if you want to know the equivalent of rows, tables, columns in MongoDB, you should definitely check it: Mapping Relational database to MongoDB.

```
{  
  name: "varshini", ←Field: Value  
  age: 20←Field: Value  
  hobbies: ["drawing", "watching tv"]←Field: Value  
}
```



This is a JSON like structure. Where data is stored in form of key and value pairs.

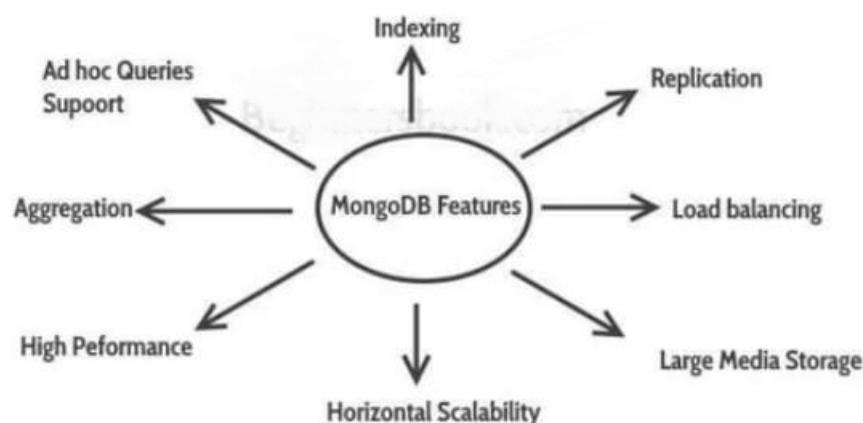
History of MongoDB

MongoDB was created by Eliot and Dwight (founders of DoubleClick) in 2007, when they faced scalability issues while working with relational database. The organization that developed MongoDB was originally known as 10gen.

In Feb 2009, they changed their business model and released MongoDB as an open source Project. The organization changed its name in 2013 and now known as MongoDB Inc.

Features of MongoDB

1. MongoDB provides high performance. Most of the operations in the MongoDB are faster compared to relational databases.
2. MongoDB provides auto replication feature that allows you to quickly recover data in case of a failure.
3. Horizontal scaling is possible in MongoDB because of sharding. Sharding is partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved.



Horizontal scaling vs vertical scaling:

Vertical scaling means adding more resources to the existing machine while horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement, on the other hand horizontal scaling is easy to implement. Horizontal scaling database examples: MongoDB, Cassandra etc.

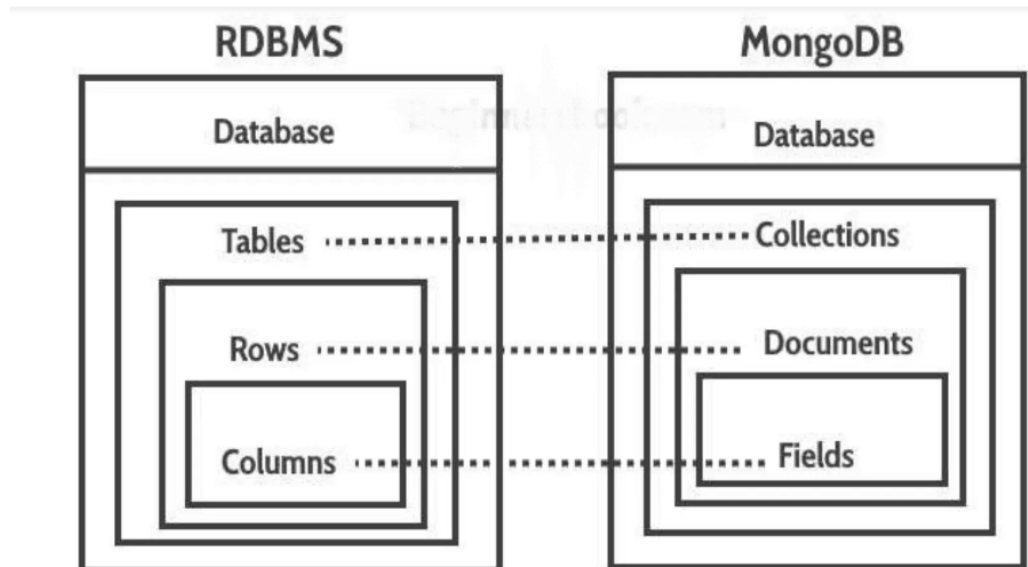
4. Load balancing: Horizontal scaling allows MongoDB to balance the load.

5. High Availability: Auto Replication improves the availability of MongoDB database.

6. Indexing: Index is a single field within the document. Indexes are used to quickly locate data without having to search every document in a MongoDB database. This improves the performance of operations performed on the MongoDB database.

Mapping Relational Databases to MongoDB

If you are coming from a relational database background then it might be difficult for you to relate the RDBMS terms with MongoDB. In this guide, we will see the mapping between relational database and MongoDB.



Collections in MongoDB is equivalent to the tables in RDBMS.

Documents in MongoDB is equivalent to the rows in RDBMS.

Fields in MongoDB is equivalent to the columns in RDBMS.

Fields (key and value pairs) are stored in document, documents are stored in collection and collections are stored in database.

This is how a document looks in MongoDB: As you can see this is similar to the row in RDBMS. The only difference is that they are in JSON format.

```
{  
  name: "varshini", ←Field: Value  
  age: 20←Field: Value  
  hobbies: ["drawing", "watching tv"]←Field: Value  
}
```

Table vs Collection

Here we will see how a table in relational database looks in MongoDB. As you see columns are represented as key-value pairs(JSON Format), rows are represented as documents.

MongoDB automatically inserts a unique_id(12-byte field) field in every document, this serves as primary key for each document.

	A	B	C	D	E	F	G	H
1	name	age	courses	gpa	home_city	blood_group	is_hotel_resident	
2	Student 94	19	['English', 'Mathematics']	3.44	City 2	O+	TRUE	
3	Student 15	20	['Physics', 'Mathematics']	2.27	City 4	O-	TRUE	
4	Student 31	20	['Physics', 'Mathematics']	2.32		B+	TRUE	
5	Student 34	25	['Mathematics', 'Physics']	3.31	City 8	O-	TRUE	
6	Student 93	25	['English', 'Mathematics']	3.63	City 3	A-	TRUE	
7	Student 30	24	['History', 'Mathematics']	3.4	City 6	O+	TRUE	
8	Student 26	21	['Mathematics', 'Physics']	3.98		A+	FALSE	
9	Student 56	18	['Mathematics', 'Physics']	2.25		AB+	FALSE	

1.Values stored in CSV

2.MONGO DB

ADD DATA
EXPORT DATA
UPDATE
DELETE

```

_id: ObjectId('665899c18fb275953df12837')
name: "Alice Smith"
age: 20
courses: Array (3)
gpa: 3.4
home_city: "New York City"
blood_group: "A+"
is_hotel_resident: true

```

```

_id: ObjectId('665899c18fb275953df12838')
name: "Bob Johnson"
age: 22
courses: Array (3)
gpa: 3.8
home_city: "Los Angeles"
blood_group: "O-"
is_hotel_resident: false

```

Another cool thing about MongoDB is that it supports dynamic schema which means one document of a collection can have 4 fields while the other document has only 3 fields. This is not possible in relational database.