MongoDB Class Notes

About Author

Narasimha Reddy K is a Full Stack Developer, working as "Tech Lead" and loves programming. He possesses over a decade of experience working on Open Source technologies, open source softwares, cloud, databases etc.

He uses technology to solve business problems and make someone's life "easy".
He contributes back to the community, i.e., to help other developers, share knowledge via different medium articles, speaking etc.

https://www.linkedin.com/in/narasimhareddyk/

Narasimha Reddy K
Full Stack Developer & Trainer

Introduction

With the rise in data all around the world, there has been an observable and increasing interest surrounding the wave of non- relational databases, also known as “NoSQL”.

Businesses and Organizations are seeking new methods to manage the flood of data and are drawn towards alternate database management tools and systems that are different from the traditional relational database systems.

Here comes MongoDB into the picture.

What is MongoDB?

MongoDB is a free and open source cross Platform Document-Oriented Database.

MongoDB is classified as NoSQL Database Program

MongoDB uses JSON-Like documents with schemas.

What cross platform? Run it on any machine(win,mac,linux)

What is a Document Oriented Database?

A Document-oriented Database provides API or query/update language

that exposes the ability to query or update based on the internal structure in the document.

What is a Document?

MongoDB documents are composed of field-value pairs and have the following structure.

```
{  key1:value1,
        key2:value2
}
```

What is MongoDB?

MongoDB is an open-source document-based database management tool that stores data in JSON-like formats.

It is a highly scalable, flexible and distributed NoSQL database.

- ☐ Highly scalable
- ☐ Flexible and
- ☐ Distributed NoSQL Database

The database: In simple words it can be called as the physical container for data.

Each of the databases has its own set of files on the file system with multiple databases existing on a single MongoDB server.

The Collection: A group of database documents can be called as a collection.

· The RDBMS equivalent of the collection is a table.

The entire collection exists within a single database.

The Document: A set of key-value pairs can be designated as a document.

Documents are associated with dynamic schemas. The benefit of having dynamic schemas is that documents in a single collection do not have to have the same structure or fields.

SQL Table Form

S No	**Name**	**Salary**
101	**Rahul**	**45000**
102	**Sonia**	
103		

Note: SQL Table from – Memory waste high.

JSON Form

```
[
{ id:101, name:”Rahul”, salary:45000},
{ id:102, name”Sonia”},
{id:103}
]
```

Retrieving the data from MongoDB is fast, when compared to SQL Database.

- ☐ **MongoDB is a schema-less database.**
- ☐ **When we have data, we directly run from the application.**
- ☐ **Mongo DB Application Driven Database.**

Application Driven Database?

I don’t want to ask DB Developer/Admin to create tables, delete tables etc.

When uses SQL?

- ☐ SQL is the easiest language used to communicate with the RDBMS.
- ☐ Analyzing behavioral related and customized sessions
- ☐ Building custom dashboards
- ☐ It allows you to store and gets data from the database quickly
- ☐ Preferred when you want to use joins and execute complex queries

When to use NoSQL?

- ☐ When ACID support is not needed
- ☐ When Traditional RDBMS model is not enough
- ☐ Data which need a flexible schema
- ☐ Constraints and validations logic not required to be implemented in
- ☐ database
- ☐ Logging data from distributed sources
- ☐ It should be used to store temporary data like shopping carts, wish list and session data

Summary

Structured Query language (SQL) pronounced as "S-Q-L" or as "See-Quel" is the standard language

NoSQL is a non-relational DMS, that does not require a fixed schema, avoids joins, and is easy to scale

SQL databases are primarily called RDBMS or Relational Databases

NoSQL databases are primarily called Non-relational or distributed database

SQL databases are table based databases

NoSQL databases can be document based, key-value pairs, graph databases

SQL should be used to communicate with the RDBMS

NOSQL should be used when Traditional RDBMS model is not enough

Mongo DB - saves data in the form of JSON - [{}, {}, {}]

Retrieving data from MongoDB is very fast.

People will go MongoDB, because of

- ☐ **Fast**
- ☐ **usage**
- ☐ **data structure**
- ☐ **less wastage of memory**
- ☐ **schema less database**
- ☐ **application Driven Database**

MongoDB is a schema-less database.

It’s more like an Application Driven Database.

More role of a developer inside the database.

Developers play a more role as compared to DBA.

When it comes to backup, replication - we require DBA. Doing CRUD - no DBA required.

Note: Everything will handle application. That's why we call it an application driven database.

As a Developer, we will use DBA to create Tables, anything.

As a Developer, we will handle everything for Database.

What MongoDB is not?

- Let's understand what mongoDB is not:
- MongoDB is NOT a RDBMS system
- MongoDB does not have any concept of joins.
- MongoDB is not tough or complicated.

Some of the languages supported by MongoDB

1. PHP
2. NodeJS
3. Python
4. Value
5. C
6. C++

MongoDB Install

1. Community Server
2. Visual Studio Extension
3. MongoDB Atlas

Let's Install MongoDB on our machine.

Visit official Website: https://www.mongodb.com/try/download/community

Download the latest stable version from community server

The community server will also install the following apps

Community Server

compass - GUI Tool for MongoDB

Steps:

1] Install from here (https://www.mongodb.com/try/download/community)

2] While installation you can check MongoDB Compass (GUI to access MongoDB just like PHPMyAdmin)

3] Config Path in Environment Variables unser System Variables

Click on Path and Edit: Add this line C:\Program Files\MongoDB\Server\6.0\bin at the last.

4] To access the db we can use MongoDB Compass or Shell.

5] To Install shell (https://www.mongodb.com/try/download/shell)

after download shell extract from zip folder add in C: drive to start the server go to bin folder of download folder shell folder and start mongosh.exe

Note : before start please create a folder like the following.

c:\data\db

mongodb server will start with port No: 27017

mongodb://localhost:27017/

How to set the mongo DB Path in Windows?

C:\Program Files\MongoDB\Server\5.0\bin

Using Environment variable

Database Checks

1) Database is installed in your system or not

Solution: C:\Program Files\MongoDB\Server\5.0\bin

2) 'mongo' is not recognized as an internal or external command

Solution: Setup the environment variables

3) exception: connect failed

Solution: Windows + R (Run Command) -> type "services.msc" -> Start "MongoDB" service

If you want to use it through the VS Code?

Install VS Code extension ie MongoDB for VS Code - Extension

Installation Via cloud MongoDB ie MongoDB - Atlas

- Cloud Hosted and Fully Managed MongoDB
- Pay as you go Model
- Very cost-effective
- Fully secured and reliable

What is Database?

Database is collection of Data or data container

In MongoDB context:

Database can also be described as a physical container of collection of Database can have any number of collections.

Each database gets it;s own set of files on the file system. A MongoDB server can hosts multiple databases inside

What is MongoDB collection?

Collection is a group of MongoDB Documents

You can relate a collection as a SQL Table.

Note: Not literally but hypothetically.

Unlike tables, collections do not have any schema definition.

Unlike RDBMS Database-the collection DO Not have any concept of Joins.

However we can achieve joins functionality using Aggregations in MongoDB.

What is a Document?

- Document in MongoDB is a set of "key-value" pairs
- Every document in MongoDB has a unique value via key "_id"
- Documents have flexible and dynamic schemas.
- Document schema is user-defined and is not fixed or static.

Document can hold any data as along as they are valid data types in MongoDB

Documents within a collection can have different schemas or fields.

Documents within a collection are related data belonging to a particular subject.

Collections

- **set of documents**
- **can have any number of documents**
- **document can have any dynamic schema**
    - **they can be same or differ**

Operations (CRUD)

C - CREATE	R - READ	U - UPDATE	D - DELETE

How many ways we can do CRUD with MongoDB?

1) CMD (DBA)

2) Standalone (Unit-Testing)

3) Website (Developer)

Two installs of MongoDB

1) Local MongoDB (installed in your system) (mongodb://127.0.0.1:27017)

2) Cloud MongoDB (configure in Google / Online)

UPDATE

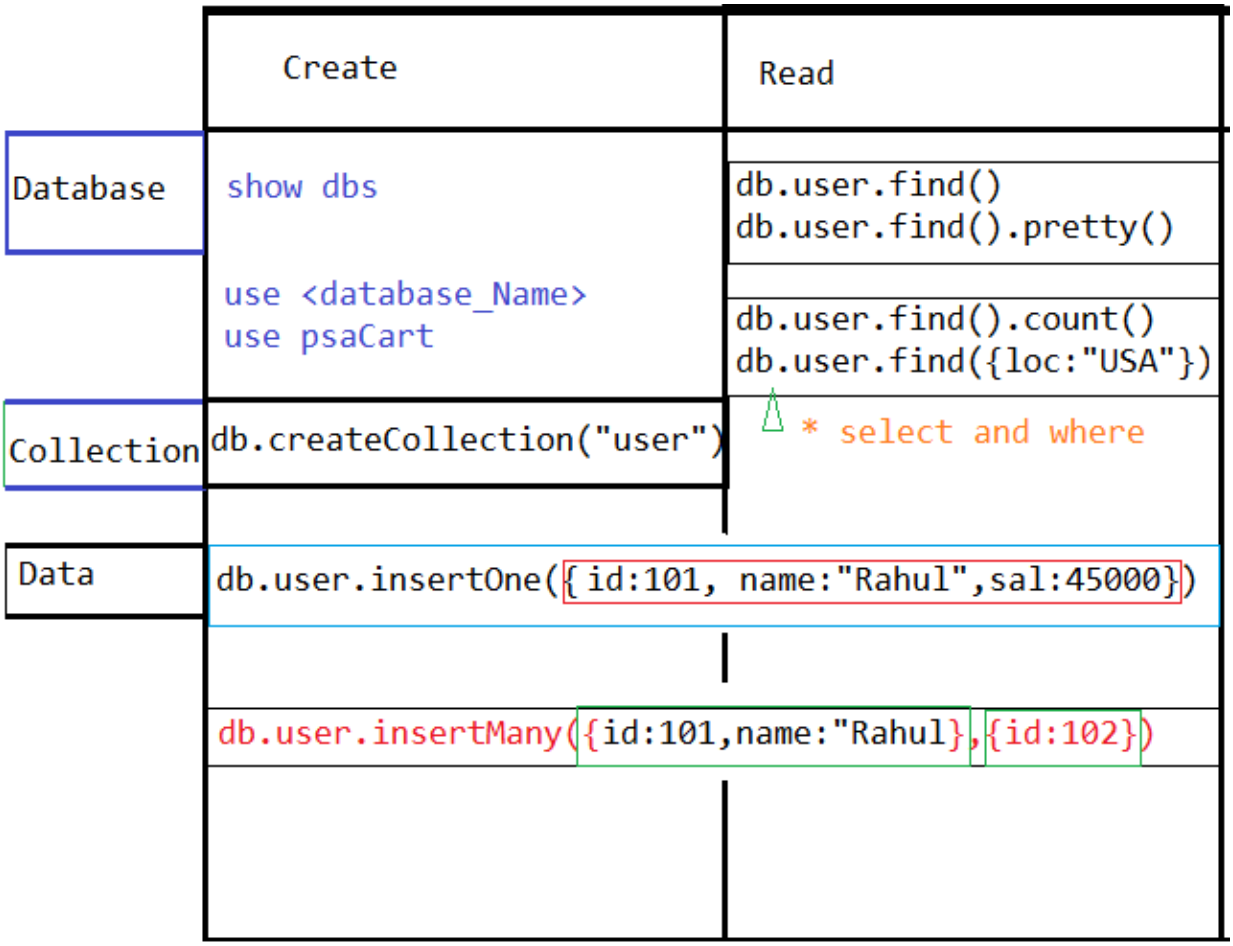
```
db.employees.updateOne({name : 'Raj'} , {
        $set : {
                age : 28,
                designation : 'Sr. Software Engineer'
        }
});

db.employees.updateOne({name: 'Wilson'} , {
        $set: {
                age: 48,
                designation: 'Program Director'
        }
});
```

DELETE

```
db.employees.deleteOne({name : 'Rajan'});

db.employees.deleteOne({_id : ObjectId("5f4f00caf4646a0e2d2cf0d")});
```


Create
Read
Database
show dbs
use <database_Name>
use psaCart
db.user.find()
db.user.find().pretty()
db.user.find().count()
db.user.find({loc:"USA"})
Collection
db.createCollection("user")
* select and where
Data
db.user.insertOne({id:101, name:"Rahul",sal:45000})
db.user.insertMany({id:101,name:"Rahul},{id:102})

Update	Delete

```
db.user.updateOne({name:"Sonia"},
                  {$set:
                        {sal:56000}
                  }
                  )
```

```
db.user.deleteOne({name:"Sonia"})
```

```
db.user.remove({}) #to remove all
```

```
db.user.drop()      # Drop collection
```

```
db.dropDatabase()  # Drop Database
```

	Create	Read	Update	Delete
Database	show dbs use <database_Name> use psaCart	db.user.find() db.user.find().pretty() db.user.find().count() db.user.find({loc:"USA"})	db.user.updateOne({name:"Sonia"}, {$set: {sal:56000} })	
Collection	db.createCollection("user")	* select and where	db.user.deleteOne({name:"Sonia"})	
Data	db.user.insertOne({id:101, name:"Rahul",sal:45000})		db.user.remove({}) #to remove all	
	db.user.insertMany({id:101,name:"Rahul},{id:102})		db.user.drop() # Drop collection	
			db.dropDatabase() # Drop Database	

NOTE:

MongoDB Compass (GUI)

2) Cloud MongoDB (configure in Google / Online)

Step 1) Login / Signup to "MongoDB Atlas" -> https://www.mongodb.com/cloud/atlas

Step 2) Create a Project

Step 3) Create a Cluster

Step 4) Create Database Access / User

Step 5) Setup the Network Access

MySQL	MongoDB
Database	Database
Table	Collection
Index	Index
Row	BSON Document
Column	BSON Field
Join	Embedded Documents and Linking
Primary Key	Primary Key
Group By	Aggregation

MySQL	MongoDB
INSERT	
INSERT INTO account (`A/c number`, `first name`, `last name`) VALUES ('16002149754', 'Sam', 'Taylor');	db.account.insert({ A/c number: "16002149754", first name: "Sam", last name: "Taylor" });
UPDATE	
UPDATE account SET contact number = 9426227364 WHERE A/c number = '16002149754'	db.account.update({ A/c number: '16002149754' }, { $set: {contact number: 9426227364 } });
DELETE	
DELETE FROM account WHERE e-mail address = 'st2345@gmail.com';	db.account.remove({ "E-mail address": "st2345@gmail.com" });

MongoDB vs MySQL: Quick Comparison	
mongoDB	MySQL
NoSQL document-oriented database	Relational database management system
Supports dynamic schema designs	Store data in tables
Sharding allows horizontal scaling	Vertical or horizontal scaling (no sharding)
Used by: Google cisco Adobe IBM.	Used by twitter Google facebook.