All MongoDb command you will ever need (MongoDb Cheatsheet)...

- **Installation & About MongoDB**

Mongo DB is Database Management System It is Different from Mysql Database or RDBMS.

It Store Data in Form of Document Oriented.

Open Source , Cross Platform, Document-oriented Database Written in C++.

Features:

- **Develop Faster**
- **Deploy Easier**
- **Scale Bigger**

MongoDB Development Started in 2007.

Latest Version is 2.4.9 & Stable Version.

Document Oriented Database.

MongoDB Stores Data as documents, so it is Known as DOD

Data Stores as BSON

If You Are From Relational DBMS Background.

Database = Database.

Tables = Collections.

Rows = Document.

When You Install Mongo DB Then Its Port is 27017. On Localhost.

Mongo is Command Line Shell to Connect With Mongod.

First We Have to Start Mongo Database From Terminal>mongod.

Then You Have to Connect With Mongo Command on Terminal to Connect With Mongod. Here You Can Use Any Language To Connect With MongoDB Like Python, any Languages.

Now Here We Discuss About Installation Of MongoDB.

Download MongoDB From Internet & Install It Then You Have to Create Folder On C Drive With Name data & inside it You Have to Make Other Folder With Name db. & Now You Go C:\Program Files\MongoDB\Server\4.2\bin

And Add This to Your Environment Variables In Your System Then In This Path You Open Your Terminal and Start You MongoDB By Command mongod. Now to Have to Connect You MongoDB Then Command is mongo.

1. Database Commands

View all databases

```
show dbs
```

Create a new or switch databases

```
use dbName
```

View current Database

```
db
```

Delete Database

```
db.dropDatabase()
```

2. Collection Commands

Show Collections

```
show collections
```

Create a collection named 'comments'

```
db.createCollection('comments')
```

Create a collection named 'comments'

```
db.cmments.drop()
```

3. Row Commands

Show all Rows in a Collection

```
db.comments.find()
```

Show all Rows in a Collection (Prettified)

```
db.comments.find().pretty()
```

Find the first row matching the object

```
db.comments.findOne({name: 'Harry'})
```

Insert One Row

```
db.comments.insert({
    'name': 'Harry',
    'lang': 'JavaScript',
    'member_since': 5
 })
```

Insert many Rows

```
db.comments.insertMany([{
    'name': 'Harry',
    'lang': 'JavaScript',
    'member_since': 5
    },
    {'name': 'Rohan',
    'lang': 'Python',
    'member_since': 3
    },
    {'name': 'Lovish',
    'lang': 'Java',
    'member_since': 4
}])
```

Search in a MongoDb Database

```
db.comments.find({lang:'Python'})
```

Limit the number of rows in output

```
db.comments.find().limit(2)
```

Count the number of rows in the output

```
db.comments.find().count()
```

Update a row

```
db.comments.update({name: 'Shubham'},
{'name': 'Harry',
    'lang': 'JavaScript',
    'member_since': 51
}, {upsert: true})
```

Mongodb Increment Operator

```
db.comments.update({name: 'Rohan'},
{$inc:{
    member_since: 2
}})
```

Mongodb Rename Operator

```
db.comments.update({name: 'Rohan'},
{$rename:{
    member_since: 'member'
}})
```

Delete Row

```
db.comments.remove({name: 'Harry'})
```

Less than/Greater than/ Less than or Eq/Greater than or Eq

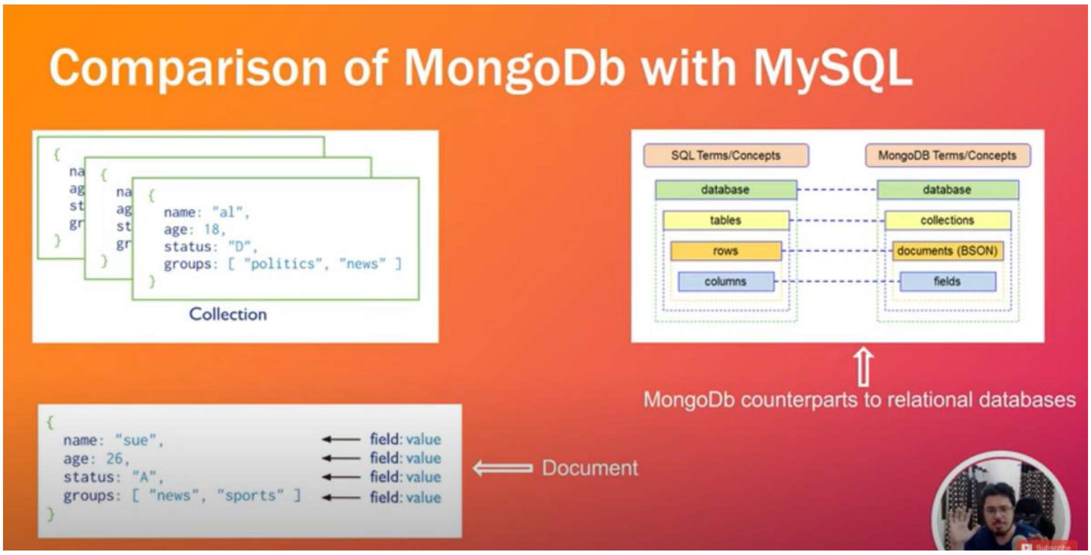
```
db.comments.find({member_since: {$lt: 90}})
db.comments.find({member_since: {$lte: 90}})
db.comments.find({member_since: {$gt: 90}})
db.comments.find({member_since: {$gte: 90}})
```

- **Some Terminology Use In MongoDB Like**

Collection ----------> Called Tables in MySQL.

Documents------------------> Called Rows in MySQL.(Store In From Of BSON Or JSON.

Fields --------------------------> Called Columns in MySQL.



Comparison of MongoDb with MySQL

```
{
   name: "al",
   age: 18,
   status: "D",
   groups: [ "politics", "news" ]
}
```

Collection

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
tables	collections
rows	documents (BSON)
columns	fields

MongoDb counterparts to relational databases

```
{
   name: "sue",          ←— field: value
   age: 26,              ←— field: value
   status: "A",          ←— field: value
   groups: [ "news", "sports" ]   ←— field: value
}
```

⟸ Document

- **Here MongoDB Is Best When We Compare With Other RDMS Like MySQL, SQL Bla Bla..**
- **Here We Have Also a GUI interface of Creating Database & Manage Everything With the Help of MongoD Compass...**
- **We Create Simple LocalHost Server And Use This..**
- **Open Terminal and Run This (Mongo) After The Installation..**

Mongo or MongoD (Demon)

Mongo is Cmd Line Utility For Fire Command And Mongo Demon is The Main Server or Place Where all Commands Executed.. That's All About MongoDb..

- **MongoDB Compass & Mongoose.**

All About MongoDB Compass About GUI Database.

To Use MongoDB GUI First We Run Mongod In Terminal.

And You Know All Things According To GUI Need.

Now We Know About Mongoose For Connect MongoDB With Nodejs.

Then Simply Install Mongoose Using npm install Mongoose.

And Check Out MongoDB Folder. For Know More About It.

```
// getting-started.js
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/Satya', { useNew
UrlParser: true,
useUnifiedTopology: true });

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error
:'));
db.once('open', function () {
    // we're connected!
    console.log("We Are Connected!")
});
```

This is For Connection MongoDB Connection with Help of Nodejs.

- **Using Mongoose In Nodejs**

You Can CheckOut MongoDB Folder.

Check Out Video And Codes.

```
 // getting-started.js.
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/Satya', { useNewUrlParser: true,
useUnifiedTopology: true });

//Here We Confirm About Our Connection.
const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));

//Here If Our Connection is Successful Then Print in The Console is We Are
Connected.
db.once('open', function () {
    // we're connected!
    console.log("We Are Connected!")
});
```

```
//Here Make Schema is a Restriction in Which we Confirm
About Our Database
Accept Data In String & You Want Then Here You Have to
Define.
var kittySchema = new mongoose.Schema({
    name: String
});

//Here if We Wants To Change It Then We Use This Functio
ns
kittySchema.methods.speak = function () {
    var greeting ="My Name is " + this.name
    console.log(greeting);
}

//Here You Make Model of Your Schema If You Make Model
 of Your Schema Then You Don't Have Permission To Chan
ge it.
//Here Satya is Collection Of Your Database MongoDB auto
matically Create it By Using This Name And Put "s" in The
Collections.
var Satya = mongoose.model('Satya', kittySchema);

//Now We Make a Document of Our Collection Satyas
var Satya01 = new Satya({name: 'Satya01'});
var Misha = new Satya({name: 'Misha'});


//Here You are Save Your Data In Database.It Take Two Ar
gu In Which Err or
Speak Argu.
Satya01.save(function (err, satu) {
    if (err) return console.error(err);
    satu.speak();
});
```

MongoDB Atlas --------->

For Cloud Database in Which We Connect Our App or Website with MongoDB Atlas And Perform Operation In Our Database. Simple Login to MongoDB Atles Visit This One.

Project Make a App & Connect With MongoDB and Perform Operations…