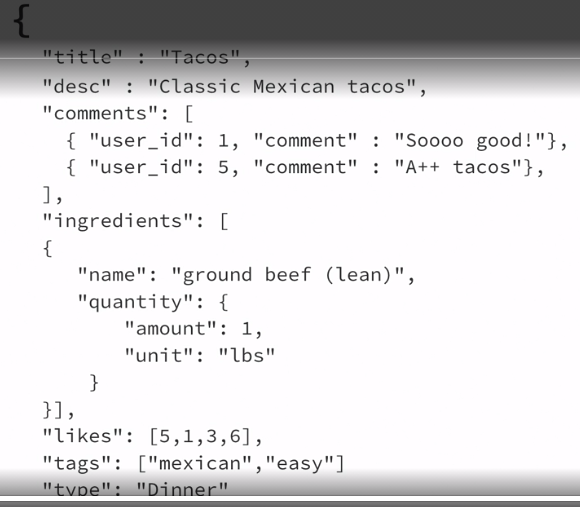
**MongoDB Basic knowledge**

**Why MongoDB?**

* Modern and flexible Json document model
* JavaScript inspired query language



**Install MongoDB on Windows**

2 Editions Enterprise and Community

MongoDB Atlas – Cloud version

UI – MongoDB Compass

Can Install as a service

Check If installed correctly: Open terminal and write

mongo –version

If can’t run mongo on terminal add it to Environment Variables.

Open Environment Variables, system variables and find path.

Find the path to the mongodb folder in your program files and add it.

C:\Program Files\MongoDB\Server\4.4\bin

**How to start manually**

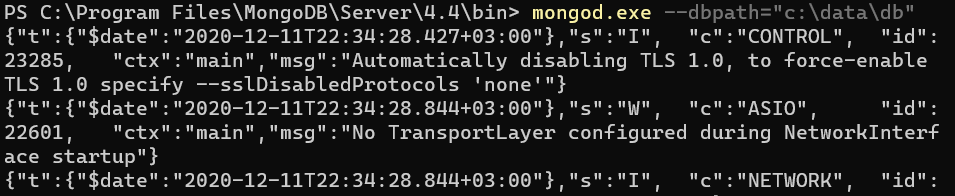
1. Create database directory

cd C:\

md "\data\db"

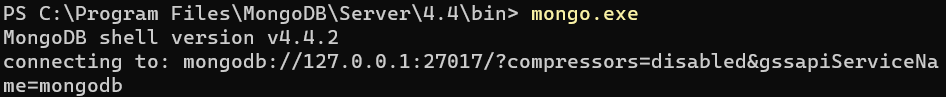
1. Start your MongoDB database

C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe --dbpath="c:\data\db"



1. Get into DB

ProgramFiles\MongoDB\Server\4.4\bin mongo



**Localhost Binding by Default**

By default, MongoDB launches with bindIp set to 127.0.0.1, which binds to the localhost network interface. This means that the mongod.exe can only accept connections from clients that are running on the same machine. Remote clients will not be able to connect to the mongod.exe, and the mongod.exe will not be able to initialize a replica set unless this value is set to a valid network interface. Before binding to a non-localhost (e.g. publicly accessible) IP address, ensure you have secured your cluster from unauthorized access.

**MongoDB Terminal Commands**

show dbs;

exit;

mongoimport : To make sure you have tools like mongoimport and mongoexport you may need to download the MongoDB Database Tools.

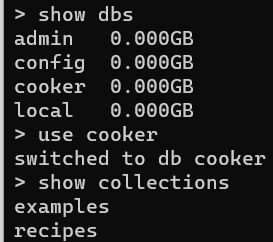
**MongoDB Database Tools**

Starting with MongoDB 4.4, the MongoDB Database Tools are now released separately from the MongoDB Server and use their own versioning, with an initial version of 100.0.0. Previously, these tools were released alongside the MongoDB Server and used matching versioning.



use cooker;

show collectios;



**Document ?**

Field-Value Pairs stored in a JSON like format = BSON



Doc 🡪 db.tacos.insertOne(doc) add to collection

db.tacos.find().pretty(); pretty format

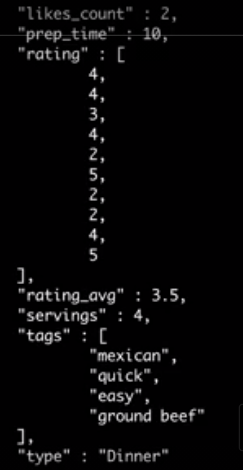
**Find() Command**

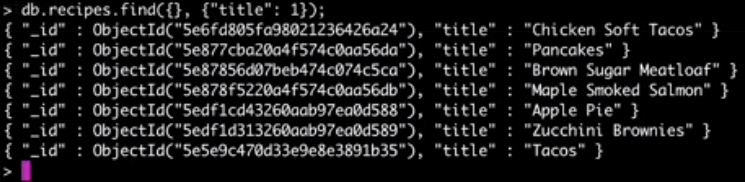
Much like a select statement in sql server.

db.recipes.find( { “title”: “Tacos”} ) search for a specific title

db.recipes.find( { “title”: “Tacos”}, { “title”:1} ) 2. Param means - just return the title column (1 = true)

db.recipes.find( { “title”: “Tacos”}, { “title”:0} ) Exclude title field from the list





We use regex instead of like in SQL because it is javascript-based.

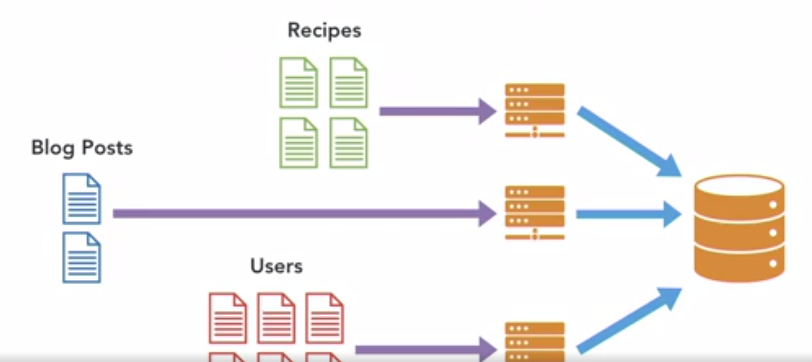
db.recipes.find( { “title”: { $regex:/tacos/i}}, { “title”:1} )

**What can we store in a document ?**

String, numbers, arrays of string/objects

**Collections**

Like tables in a RDB.



show collections;

db.cool\_NewStuff.insertOne({}); Create a collection with empty doc

db.cool.newStuff.insertOne({}); Create a sub-collection

**Query MongoDB**

Sort, limit, skip be careful about performance

db.recipes.find().sort({“title”:1}).limit(2).skip(5)

When we do a query, the database returns a cursor to us in the background.

db.recipes.find( ).count()

db.recipes.find( ).limit(5)

db.recipes.find( ).sort({“title”:1}) 1 = ASC, -1 = DESC

**Comparison Operators**

$gt > $lt < $lte <=

db.recipes.find( {“cook\_time”} : {$lte:30}, {“title”:1} ) lte example

OR example

db.recipes.find( { $or: [ {“cook\_time” : {$lte:30}, “prep\_time”: {$lte:10}} ] }, {“title”:1} )

Search in array, both tags need to be

db.recipes.find( { tags: { $all : [“easy”,”desm”]}}, {“title”:1 , “tags”:1})

Search in array with or

db.recipes.find( { tags: { $in : [“easy”,”desm”]}}, {“title”:1 , “tags”:1})

Search in object

db.recipes.find( { “ingredients.name” : “egg” } )

**Update Docs**

$set command



$unset command



$inc increment

db.products.update(

{ sku: "abc123" },

{ $inc: { quantity: -2, "metrics.orders": 1 } }

)

$inc operator to decrease the quantity field by 2 (i.e. increase by -2) and increase the "metrics.orders" field by 1:

**Array Operators**

$push add $pull remove

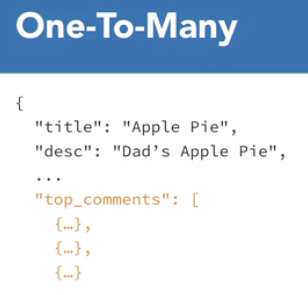


**DELETE**

deleteOne ile \_id geçerek istdiğimiz doc’u silebiliriz. Yada deleteMany ile çoklu silebiliriz.

**Data Modeling**

“Data that is accessed together should be stored together” – MongoDB

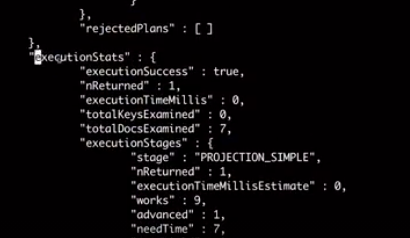


If we want to keep the recipe and comments, it would be more appropriate to do it as follows;

We can keep the other comments in the comment doc by keeping only the top 3 comment in the recipe doc.

**Indexes**

db.recipes.find( { “cook\_time”: 10} ).explain(“executionStats”);

****

db.recipes.**getIndexes()**

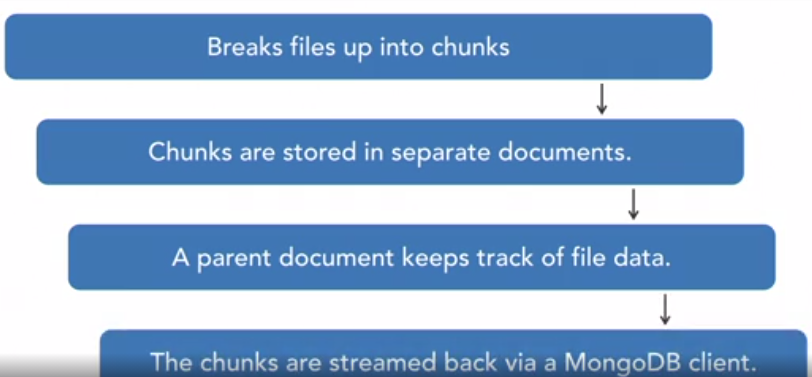
**Create Index**

db.recipes.createIndex( {“cook\_time”: -1} ): -1 means desc order

db.recipes.dropIndex(“cook\_time\_-1”)

Note: default name (cook\_time\_-1) of the index can be changed.

**GridFS – Store Files**

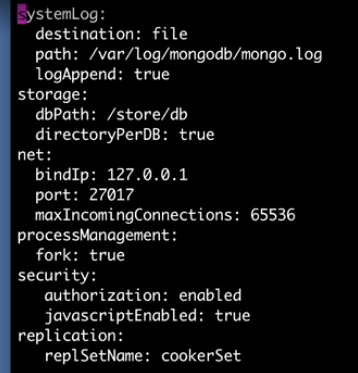


Put files to GridFS.



**MongoDB Config File**

YAML



**Replication**

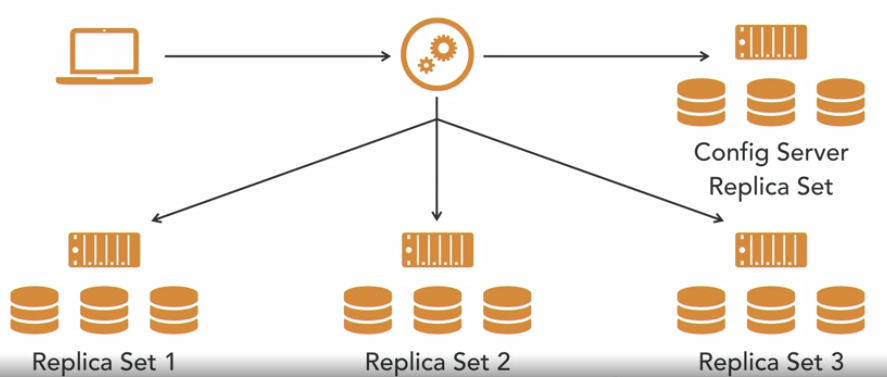
By creating Replica Sets, we can create a primary and many slaves. If something happens to the primary, one of the slaves takes the primary role.

**Sharding**

Is the concept of breaking up or partitioning your data across multiple servers.

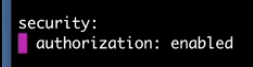
Auto-shard: Easy in MongoDB

Basic sharding setup



**Security**

By default, authorization is disabled in the config. Define a user and enable it.



**Backups**

We can do it by copying DB files. Before, we need to stop writing to db.

db.fsynclock() – lock the writes, start writing to memory, need to be quick for copying files

db.fsynUnclock()

1. If you have a Replica Set, you can take one of them and copy the files without experiencing downtime.
2. Mongodumb – mongorestore

mongodump is a utility bundled with the MongoDB database. It is used in creating binary exports of the database contents. It can also be downloaded as a separate tool from the MongoDB Download Center.

For a complete backup and recovery strategy, mongodump is used alongside the mongorestore tool whereby the latter is used in the restoration process. It is also possible to have partial backups based on a collection, query, changing a replica set storage engine to a standalone or syncing from production to development environment.