MongoDB

* It is a NoSQL database
* It stores records as document which is a Javascript Object which will have key and value
* It is built for modern applications, because the data structure would not be always same
* It doesn’t follow any rules of SQL, like table must have only related data and relationship of the tables is created in another table foreign key constraint and so on
* Performance wise it is better and it gives lot of inbuilt functions to perform operations on the database
* It can handle the data in any structure and also data structure can be dynamic
* MongoDB stores the related data in as an embedded document, it means it doesn’t keep the related data in a separate place, so that the read operations would be faster

Advantages of using documents;

1. Documents are like objects, they are easily compatible with many programming languages like Javascript, Java, C#, because it stores in a Javascript format or JSON format
2. Embedded documents and arrays can reduce expensive read and join operations
3. Dynamic Schema support

MongoDB Terminologies

1. Collections: It is going to store documents, it will have a name using which you can store the documents, it is like a table in RDBMS
2. Document: It is a record that will have data with properties & values, it can be a simple document or arrays of document or lot of embedded documents, it is like a row in a table
3. Database: It is an instance that will have multiple collections.

Note: MongoDB is case sensitive, if you create a collection with emp and another collection with EMP then both are different they are not one and the same.

How to install MongoDB

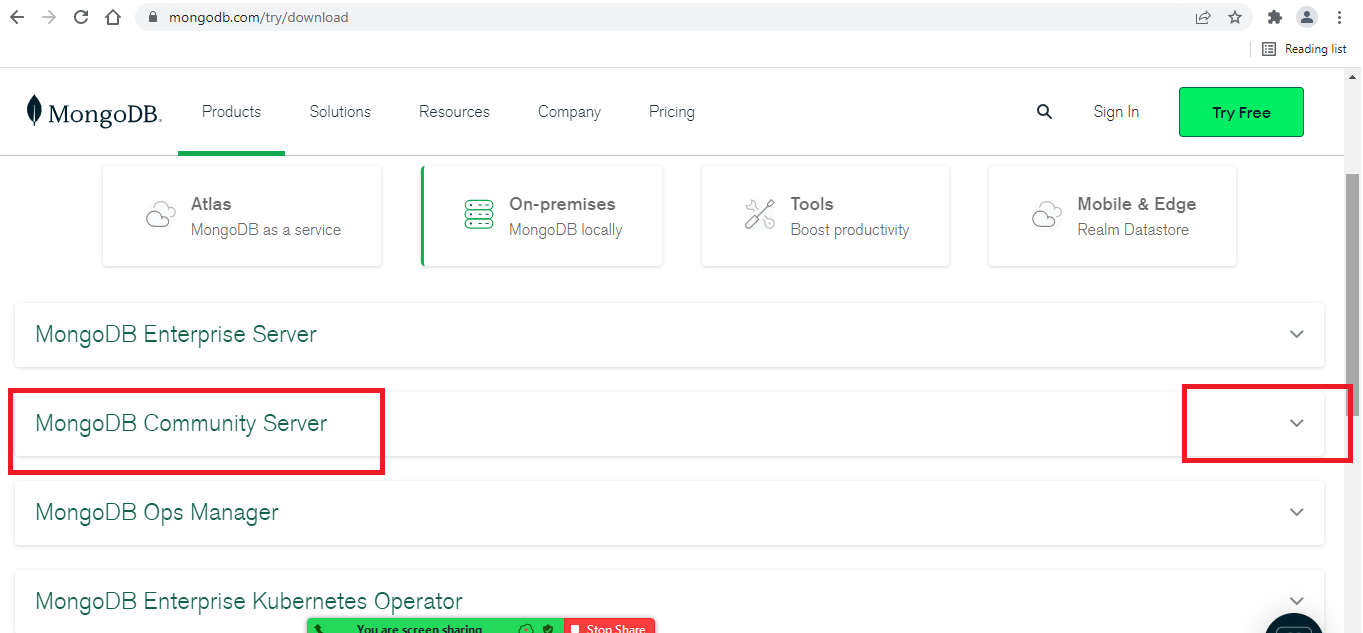
You can install Mongodb in 2 ways.

1. Installer file
2. Zip file having the installation folder (easier)

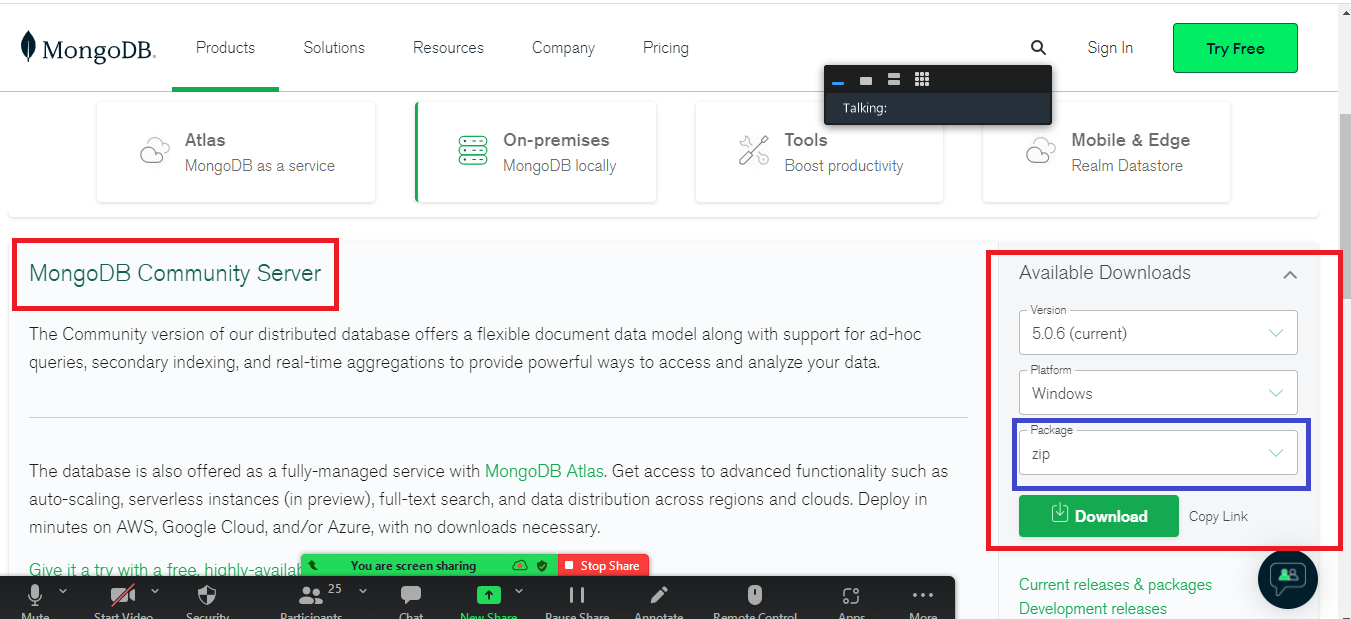
2nd way you can prefer as you will get the installation directory once you unzip the zip file

You can download the installation zip file from the below website

<https://www.mongodb.com/try/download/community>



You have to select MongoDB Community Server from where you can select the mongodb zip file that will have the installation folder zipped.



Once the zip file is downloaded, extract the zip and you will get an installation folder which is ready to use.

Once you get the mongodb installed, you need to start the mongodb database and you can use mongo-shell to interact with mongodb database, you will get to commands from /bin folder of mongodb installation folder to start the database & interact with the database, the two commands are:-

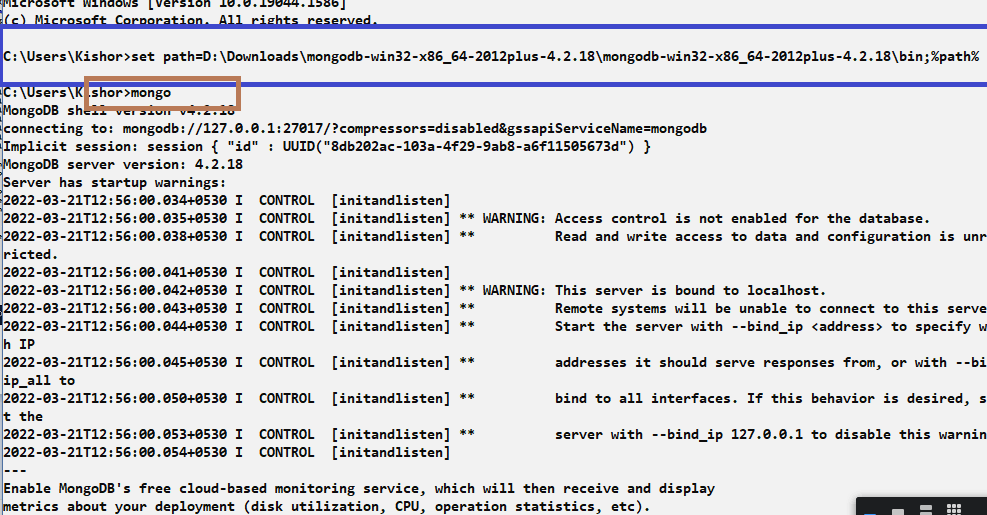
1. mongod
2. mongo

Both the above commands are present in the mongodb-instllation-folder/bin,

mongod: It is used to start the mongodb database, it usually takes 27017 port number and it starts the database by looking at a folder data/db in either C: drive or D: drive depending on where you enter mongod command.

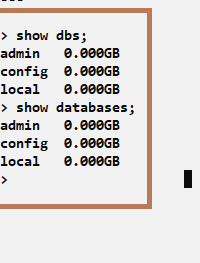
mongo: It is used to interact with mongodb database, even this command also your terminal must recognize.

Note: You need to have minimum 2 command prompts open, one for the mongod database and another is for mongo shell(terminal)

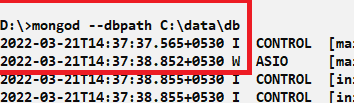


Mongo Shell: It is a terminal to interact with mongodb, which allows you to perform all the CRUD operations, similar to MySQL client

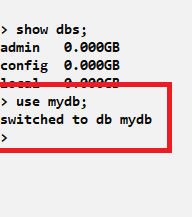
MongoDB has some inbuilt databases



dbpath: It is an option you can use while entering the mongod command to specify the path of the database, you need to mention the location of the database the mongodb should use to store and retrieve the data



Switching to the database



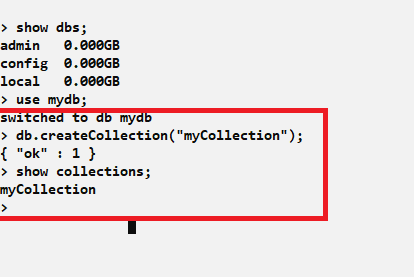
This command switches to the mydb which is a database instance, you can create collections inside the mydb.

Collection: It is like a table to store the documents

Syntax to create collection

db.createCollection(“name”);

db is going to use the current database you are using and createCollection is a function that creates the collection, it takes an argument that is the name of the collection.



show collections: It is a command to list all the collections present in the database

You can perform various operations in the mongodb like store, retrieve, update, delete, to perform these operations we have mongodb inbuilt functions like insert(), update(), remove(), find() and so on

Storing records

You have a command called insert() and insertOne(), to store the document, insert() was a old function that only returns the status, but insertOne() is a new function that returns the status as well as the id of the record

What is id:

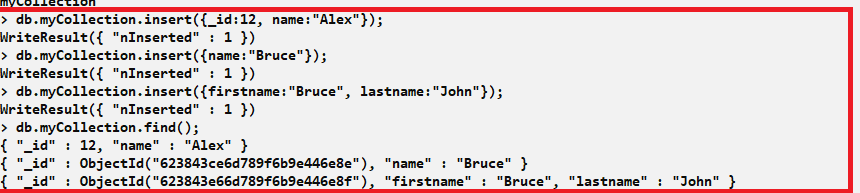
id is a primary key in the document, which must exist, else mongodb itself creates an id, the id will be prefixed with \_, so it will be \_id, \_id must always be unique

insert() vs insertOne()

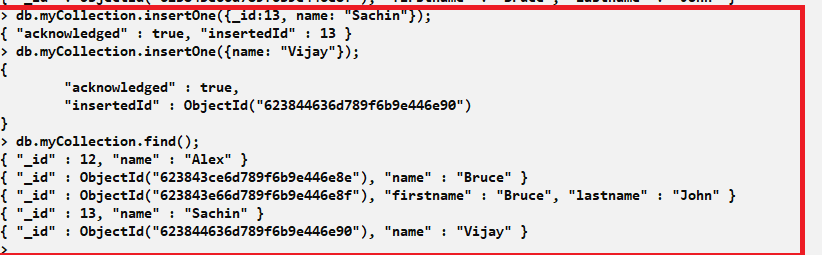
insert() is a function that stores the record and returns the status how many records inserted, it is deprecated

insertOne() is also a function that stores the record and returns the \_id and the status, which would be very useful to the caller so that he can use that \_id to perform some operations

Using insert()



Using insertOne()



find(): It allows you to view all the documents in the collection

Bulk insert: This allows you to insert multiple documents at a time, it must be wrapped in the [], which means its an array of documents, i.e., array of objects.

insertMany() is the function we must use for bulk operations

Syntax:

db.collection\_name.insertMany([{..}, {..}, {..}, …]



Activities:

1. Try above examples
2. Create 2 collections with the same name & different case ex: employee & EMPLOYEE and insert documents in both the collections and observe the result
3. Create a collection user with \_id, name, password
4. Store some documents in the user collection
5. Find the user based on the \_id i.e., view the document on a particular \_id
6. Delete the user based on the \_id
7. Delete all the users on one go.
8. Store some more documents in the user collection
9. Update the password of a particular user using \_id

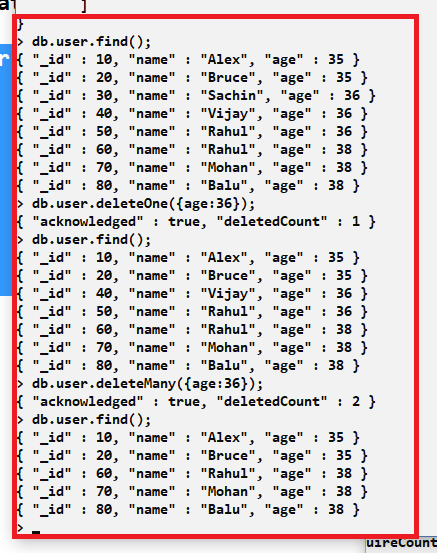
Delete Documents:

You can delete documents using deleteOne() & deleteMany() function, deleteOne() deletes only one record though there are more than one document matching to the condition, deleteMany() deletes multiple records matching to the condition.

Syntax:

db.collectionName.deleteOne({filter});

db.collectionName.deleteMany({filter});



Update document

To update the document you can use two different functions

1. updateOne()
2. updateMany()

Syntax:

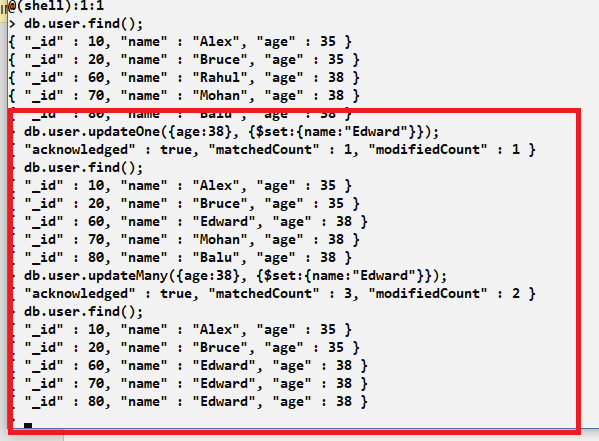
updateOne(filter, operation)

updateMany(filter, operation)

Ex:

db.collectionName.updateOne({property:value}, {$set:{property:value}});

db.collectionName.updateMany({property:value}, {$set:{property:value}});



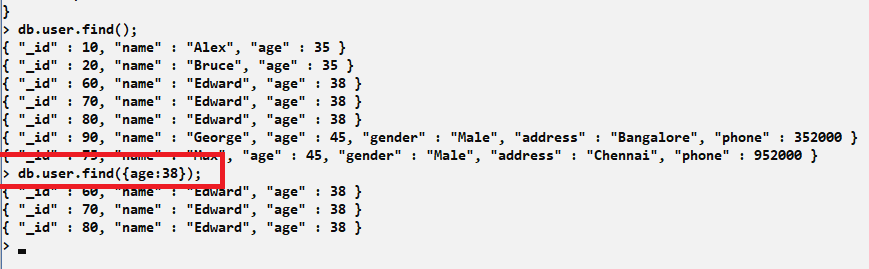
Query documents

You can use find() with conditions to query the documents



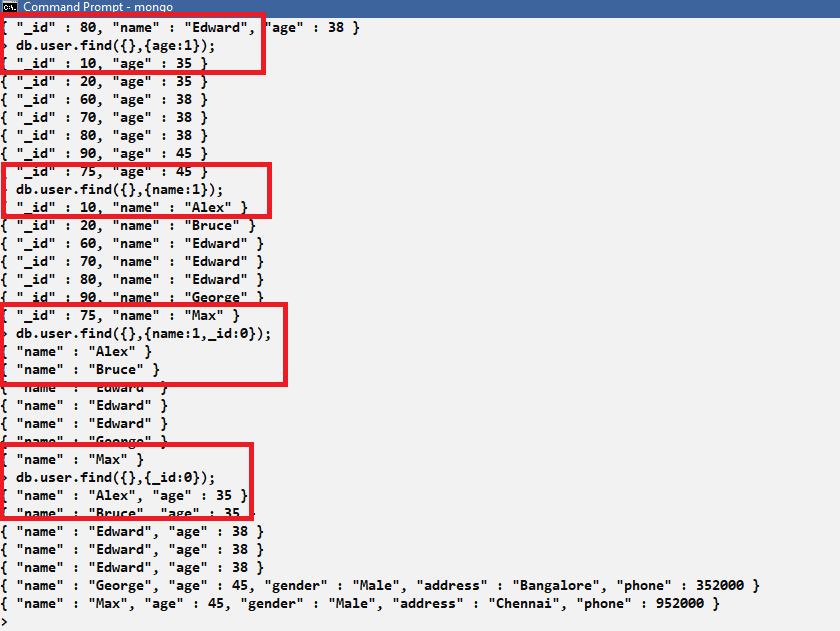
pretty(): It is a function that can provide indentations to the document while querying so that it can have a good readability.

find(filter):

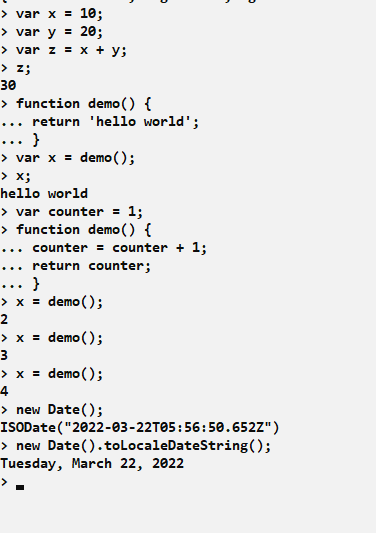


You can hide some properties while querying so that it doesn’t show in the result

You can use 0 to hide and 1 to show

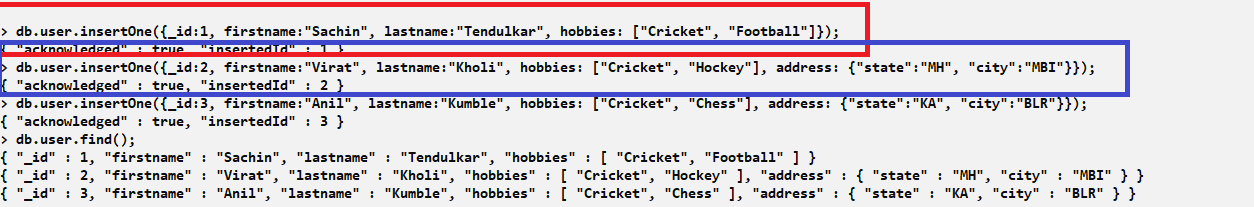


Mongodb supports javascript objects as well as other javascript features like inbuilt objects, functions and so on.

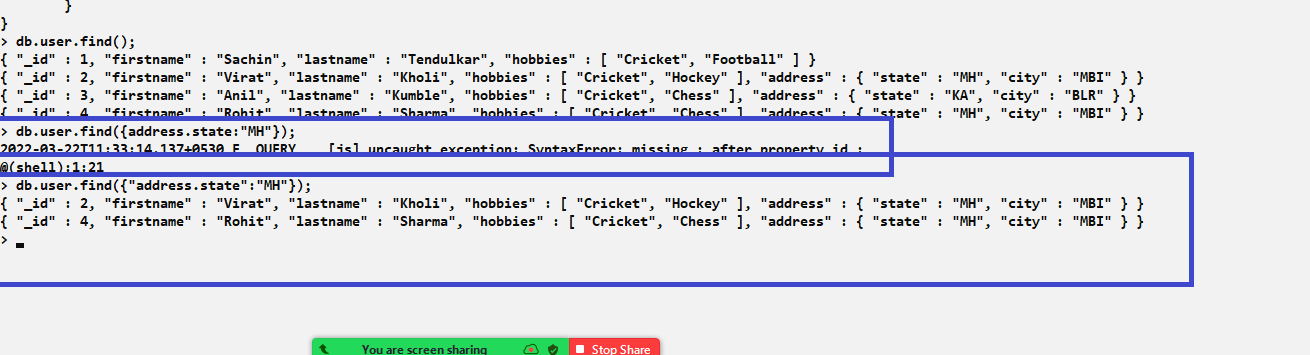


You can also call these functions while working with the mongodb documents

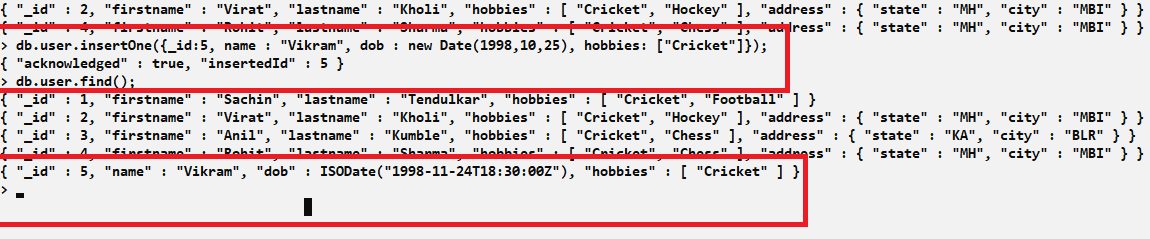
Inserting arrays and Nested documents:



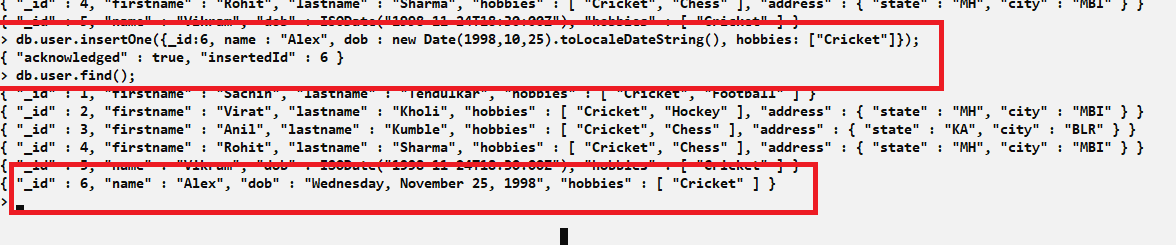
Querying nested documents:



Storing inbuilt javascript objects



In the above example we have store the date object, since month starts from 0, the month 10 would be treated as November.

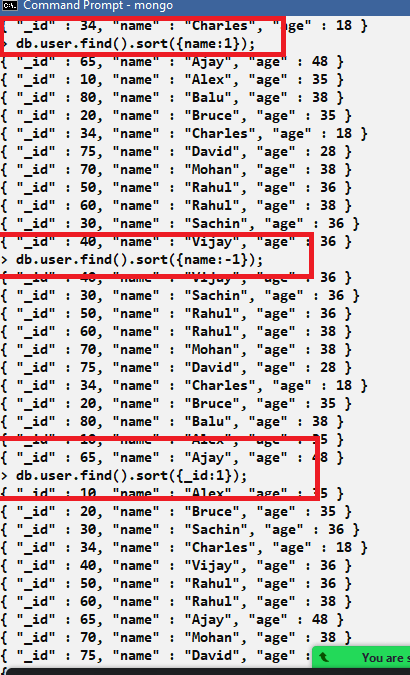


Sorting and limiting

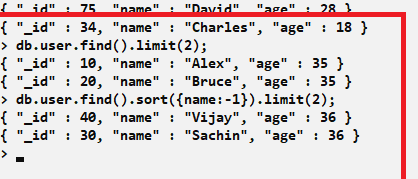
You can call sort() function and limit() function while querying the documents to sort the document and also limit the results to certain numbers.

ex: db.collection.find().sort({key:value});

ex: db.collection.find().limit(number);



Here sort({}) accepts the key with 1 and -1, where 1 is used for ascending and -1 is used for descending



Backup and Restoration

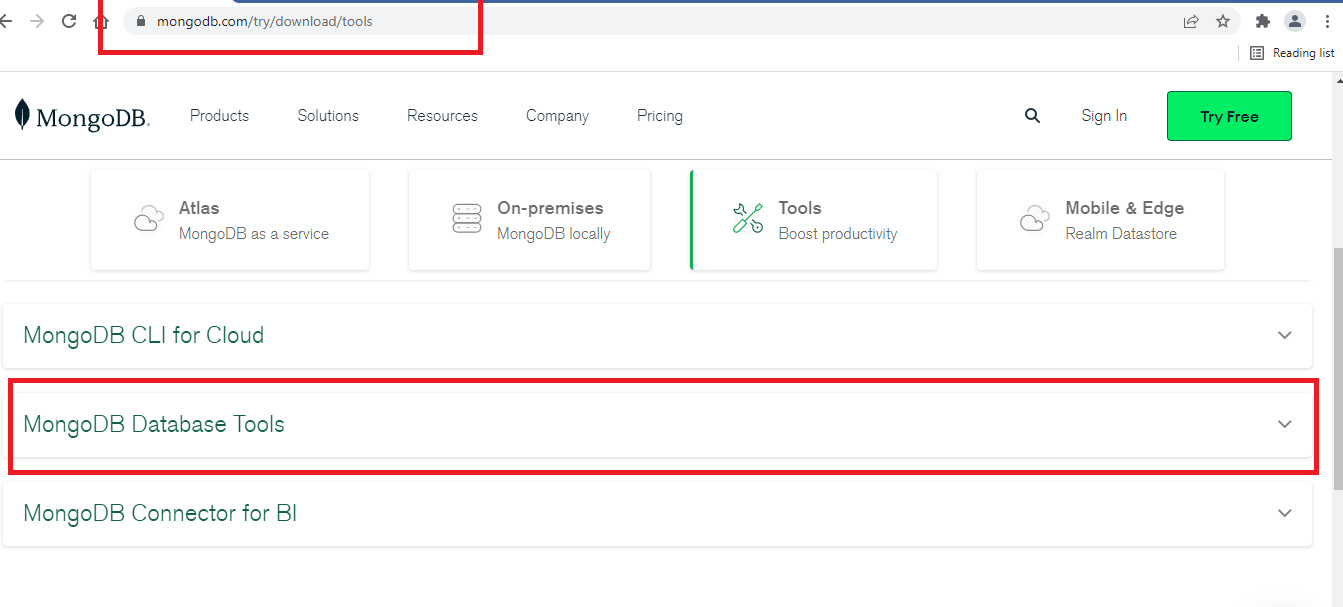
We have backup and restoring technique to get back the data when something wrong happens, so that data wouldn’t be lost, mongodb has many strategies to take backup:

1. Backup with Mongodb cloud or Ops manager: it is an automated service, which frequently takes the backup of the database and can be restored if the database data is lost, it allows to get the data from a specific restore point as well, this is used in the production environment.
2. Backup with MongoDump: It is a database tool that you need to install to manually take the backup and restore the data, it works fine with the local machine also, database tools you need to install in your machine first.

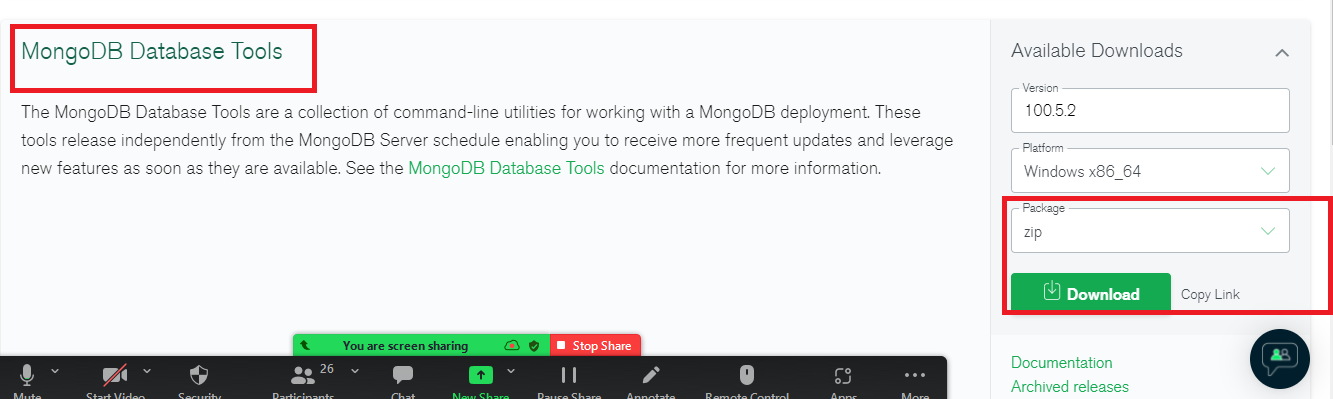
Since mongodb is running in our local machine, we can use mongo dump to manually take the backup, mongodb cloud or ops manager will have some scripts that will be executed when the schedule time is reached, mongodump works the same way but it is manual backup strategy.

Installing mongodb database tools

<https://www.mongodb.com/try/download/database-tools>

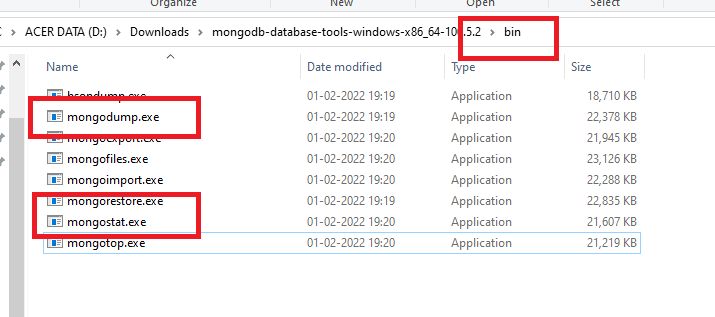


You can download the zip and extract



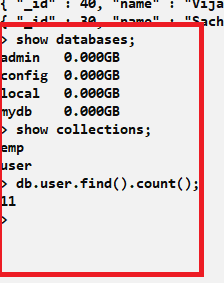
The unzipped folder has two commands

* mongodump: This takes the backup
* mongorestore: This restores the database till what is backed up



Note: Your command prompt will not recognize these commands, hence you need to set path so that it can recognize the commands of this bin folder.

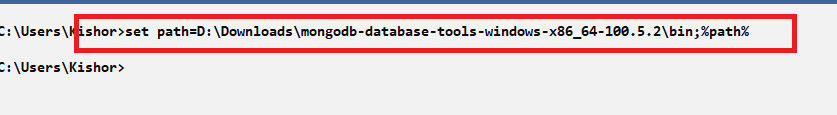
Currently the database has mydb database & two collections: emp & user, we can take the backup from mongodump and if the data/db folder is deleted we should able to get that back through mongorestore



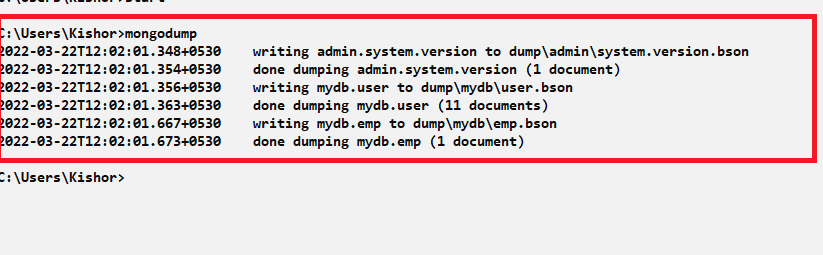
Purposely will delete the data/db folder, ensure the mongod is stopped as well

Steps:

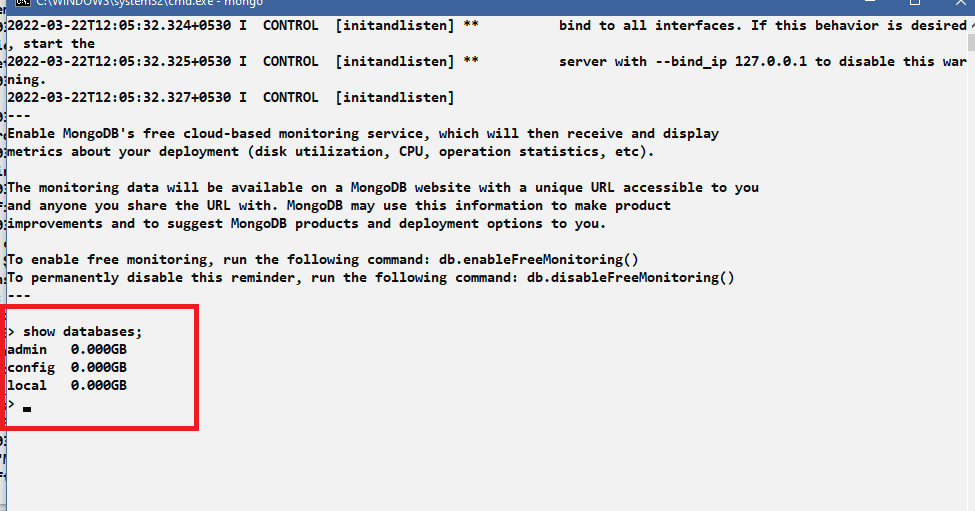
1. Set path to database tools/bin folder



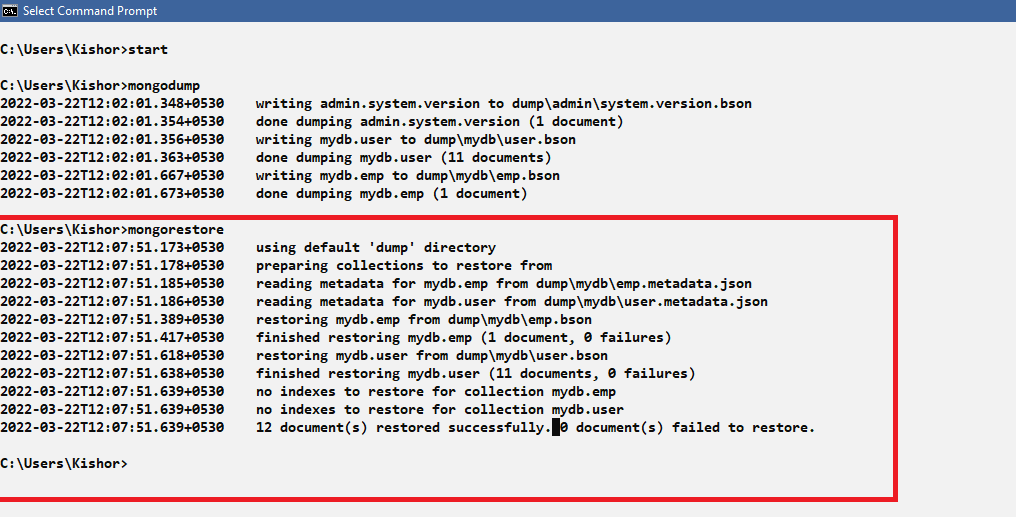
1. Use mongodump to take the backup, ensure the database is running



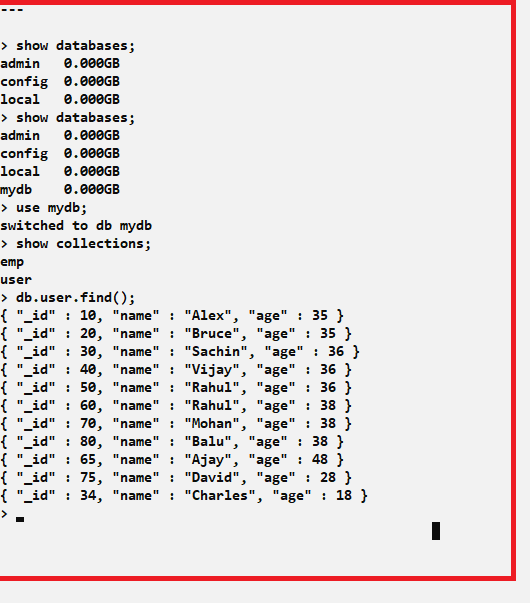
1. You can delete all the files & folders of data/db or data/db itself, but while restoring you must have data/db folder, ensure the database is stopped while deleting
2. You can recreate data/db and run the database where you don’t see any documents and collections like emp & user



1. Since the data/db is newly created we don’t see mydb and collections like emp & user, but we have taken the backup so we can restore
2. Enter mongorestore command from another command prompt



1. Now you must able to see mydb and documents in the user and emp



Query operators:

These are used to apply various types of conditions

Comparison: $eq, $lt, $gt, $lte, $gte, $ne, $in

Logical: $or, $and, $not, $nor

Evaluation: $regex, $where

Example:

db.user.find({$or:[{“name”:”Alex”}, {“salary”:45000}]});

db.user.find({$and:[{“age”:$gte:{18}}, {“salary”:$gte:{30000}}]});

$eq: Returns the documents that matches to the value

$gt: Returns the documents whose value is greater than specified value

$gte: Returns the documents whose value is greater than or equal of the specified value

$lte: less than equal

$ne: not equal

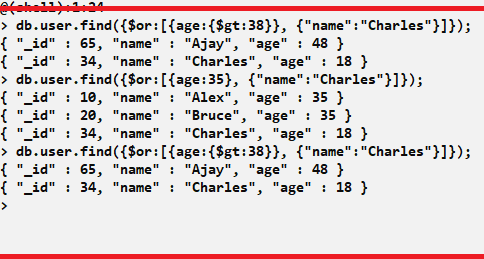
$nor: joins query with logical NOR returns all the documents that fail to match the clauses

$or: joins query with logical OR returns all the documents that match the conditions of either clause

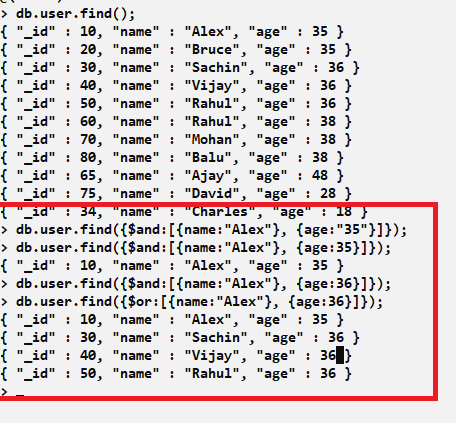
$and: joins query with logical AND returns all documents that match the conditions of both clause

$exists: Checks the existence of a value

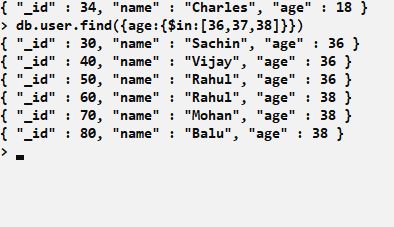
$in: It is for matching either of the value in the array



Using $or, $and



Using $in



Activity:

1. Try all the above exercises
2. Try out all the query operations like $lt, $lte, $in, $gte, $not, $nin, $where, $exists and keep the commands in a text file and push to the GIT
3. Try some features using javascript functions
   1. Autogenerate values to \_id for a user like 1, 2, 3, 4 and so, so that you don’t manually pass the \_id, the autogenerating function should generate the next sequence of number even if you terminate the database and restart it, i.e., don’t just create a temporary variable and increment it
   2. Generate values to \_id for a student like a university number, ex: if the university is ABC, then the numbers must be ABC1, ABC2, ABC3 and so on, this must also be auto-generate through the javascript function, it must generate the next sequence even if you terminating the database and restart it

Note: Keep all the queries and functions in the txt file and push it to the GIT