db.collection.deleteMany({})

Ans:
db.collection.drop() --->Both collection and documents will be deleted.
db.collection.deleteMany({}) --->Only documents will be deleted but not collection.

Q2. What is the difference between the following commands?

> db.collection.deleteMany({})
> db.collection.remove({})
> db.collection.remove({},false)

ANSWER :All are equal and deleting all documents

MongoDB Utilities (Database Tools):

----
1. mongoimport
2. mongoexport
3. mongodump
4. mongorestore
etc

All these are separate applications.
These can be used for data management in mongodb.
These are commond line utilities and we have to execute from command prompt only but not from mongodb shell.

Bydefault these tools are not available and we have to install

separately.

How to install these tools:

---

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

Download

we will get zip file: mongodb-database-tools-windows-x86_64-100.3.0

Extract this zip file and copy the utility tools to our mongodb bin folder.

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

1. mongoimport:

We can use this tool to import data into mongodb database from the files like json file, csv file etc

eg-1: To import data from json file:

Syntax:

mongoimport -d databaseName -c collectionName --file fileName --jsonArray

eg:

mongoimport -d durgadb -c students --file students.json --jsonArray

eg-2: To import data from csv file:

Syntax:

mongoimport -d databaseName -c collectionName --type csv --headerline --drop emp.csv

2. mongoexport:

We can use this tool to export specific data from the given collection to the files.

The data will be stored in the file in json format.

Syntax:

mongoexport -d databaseName -c collectionName -o fileName

-d ===>databaseName

-c ===>collectionName

-o ===>Name of the file where exported data sholud be written.

eg: To export data from employees collection of durgadb database to emp.txt file

mongoexport -d durgadb -c employees -o emp.txt

```
C:\Users\lenovo\Desktop>mongoexport -d durgadb -c employees -o
emp.txt
2021-02-14T09:18:07.170+0530    connected to: mongodb://localhost/
2021-02-14T09:18:07.178+0530    exported 8 records
```

emp.txt:

```
-------
{"_id":{"$oid":"5fe220cd573f5ff261265825"},"eno":100.0,"ename":"Sunn
y","esal":1000.0,"eaddr":"Mumbai"}
{"_id":{"$oid":"5fe221b6573f5ff261265826"},"eno":200.0,"ename":"Bunn
y","esal":2000.0,"eaddr":"Hyd"}
{"_id":{"$oid":"5fe221b6573f5ff261265827"},"eno":300.0,"ename":"Chin
ny","esal":3000.0,"eaddr":"Chennai"}
{"_id":{"$oid":"5fe221b6573f5ff261265828"},"eno":400.0,"ename":"Vinn
y","esal":4000.0,"eaddr":"Delhi"}
{"_id":{"$oid":"5fe222ac573f5ff261265829"},"eno":777.0,"ename":"Sunn
y","esal":1000.0,"eaddr":"Mumbai"}
{"_id":{"$oid":"5fe222e9573f5ff26126582a"},"eno":888.0,"ename":"Bunn
y","esal":2000.0,"eaddr":"Hyd"}
```

{"_id":{"$oid":"5fe222e9573f5ff26126582b"},"eno":999.0,"ename":"Chinny","esal":3000.0,"eaddr":"Chennai"}
{"_id":{"$oid":"5fe2241b573f5ff26126582c"},"eno":77777.0,"ename":"Sachin","esal":99999.0,"eaddr":"Mumbai"}

MongoDB backup and restore by using mongodump and mongorestore tools:

By using mongodump and mongorestore commands we can take backup of database and we can restore the backup data.

mongodump ===>Create a dump from mongodb database.
mongorestore ===>To restore data from dump.

Case-1: To create dump for all databases:

We have to use mongodump command without any arguments.

```
C:\Users\lenovo\Desktop>mongodump
2021-02-14T09:32:01.762+0530    writing admin.system.version to
dump\admin\system.version.bson
2021-02-14T09:32:01.764+0530    done dumping admin.system.version
(1 document)
2021-02-14T09:32:01.765+0530    writing durgadb.employees3 to
dump\durgadb\employees3.bson
2021-02-14T09:32:01.773+0530    done dumping durgadb.employees3
(1 document)
2021-02-14T09:32:01.773+0530    writing durgadb.employees2 to
dump\durgadb\employees2.bson
2021-02-14T09:32:01.775+0530    done dumping durgadb.employees2
(0 documents)
2021-02-14T09:32:01.776+0530    writing studentdb.sssdb1 to
dump\studentdb\sssdb1.bson
```

2021-02-14T09:32:01.777+0530 done dumping studentdb.sssdb1 (0 documents)
2021-02-14T09:32:02.067+0530 writing durgadb.employees to dump\durgadb\employees.bson
2021-02-14T09:32:02.067+0530 writing studentdb.sssdb2 to dump\studentdb\sssdb2.bson
2021-02-14T09:32:02.067+0530 writing studentdb.students to dump\studentdb\students.bson
2021-02-14T09:32:02.070+0530 done dumping studentdb.students (1 document)
2021-02-14T09:32:02.075+0530 done dumping studentdb.sssdb2 (1 document)
2021-02-14T09:32:02.075+0530 done dumping durgadb.employees (8 documents)

Note: dump folder got created which contains data from all databases.
For every collection 2 files will be created.
bson file==>contains original data in bson format
json file==>contains metadata related to that collection.

```
C:\Users\lenovo\Desktop>tree /f dump
Folder PATH listing
Volume serial number is 00000220 526A:E31B
C:\USERS\LENOVO\DESKTOP\DUMP
├───admin
│       system.version.bson
│       system.version.metadata.json
│
├───durgadb
│       employees.bson
│       employees.metadata.json
│       employees2.bson
│       employees2.metadata.json
```

```
|       employees3.bson
|       employees3.metadata.json
|
└———studentdb
        sssdb1.bson
        sssdb1.metadata.json
        sssdb2.bson
        sssdb2.metadata.json
        students.bson
        students.metadata.json
```

Note:

The original data stored in bson form where as metadata stored in json form.

But we can convert bson data to json format by using bsondump tool.

bsondump ===>To convert data from bson format to json format

eg-1: Convert bson data from employees.bson to json form and display to the console.

```
C:\Users\lenovo\Desktop\dump\durgadb>bsondump employees.bson
{"_id":{"$oid":"5fe220cd573f5ff261265825"},"eno":{"$numberDouble":"100.0"},"ename":"Sunny","esal":{"$numberDouble":"1000.0"},"eaddr":"Mumbai"}
{"_id":{"$oid":"5fe221b6573f5ff261265826"},"eno":{"$numberDouble":"200.0"},"ename":"Bunny","esal":{"$numberDouble":"2000.0"},"eaddr":"Hyd"}
{"_id":{"$oid":"5fe221b6573f5ff261265827"},"eno":{"$numberDouble":"300.0"},"ename":"Chinny","esal":{"$numberDouble":"3000.0"},"eaddr":"Chennai"}
{"_id":{"$oid":"5fe221b6573f5ff261265828"},"eno":{"$numberDouble":"400.0"},"ename":"Vinny","esal":{"$numberDouble":"4000.0"},"eaddr":"Delhi"}
```

{"_id":{"$oid":"5fe222ac573f5ff261265829"},"eno":{"$numberDouble":"777.0"},"ename":"Sunny","esal":{"$numberDouble":"1000.0"},"eaddr":"Mumbai"}
{"_id":{"$oid":"5fe222e9573f5ff26126582a"},"eno":{"$numberDouble":"888.0"},"ename":"Bunny","esal":{"$numberDouble":"2000.0"},"eaddr":"Hyd"}
{"_id":{"$oid":"5fe222e9573f5ff26126582b"},"eno":{"$numberDouble":"999.0"},"ename":"Chinny","esal":{"$numberDouble":"3000.0"},"eaddr":"Chennai"}
{"_id":{"$oid":"5fe2241b573f5ff26126582c"},"eno":{"$numberDouble":"77777.0"},"ename":"Sachin","esal":{"$numberDouble":"99999.0"},"eaddr":"Mumbai"}
2021-02-14T09:41:01.199+0530 8 objects found

eg-2: Convert bson data from employees.bson to json form and write to emp.json file

Bydefault bsondump tool outputs data to the standard output(console). We can write converted data to the files also. For this we have to use --outFile argument.

```
bsondump --outFile=emp.json  employees.bson

C:\Users\lenovo\Desktop\dump\durgadb>bsondump --outFile=emp.json employees.bson
2021-02-14T09:44:29.662+0530    8 objects found
```

emp.json:

{"_id":{"$oid":"5fe220cd573f5ff261265825"},"eno":{"$numberDouble":"100.0"},"ename":"Sunny","esal":{"$numberDouble":"1000.0"},"eaddr":"Mumbai"}

{"_id":{"$oid":"5fe221b6573f5ff261265826"},"eno":{"$numberDouble":"200.0"},"ename":"Bunny","esal":{"$numberDouble":"2000.0"},"eaddr":"Hyd"}
{"_id":{"$oid":"5fe221b6573f5ff261265827"},"eno":{"$numberDouble":"300.0"},"ename":"Chinny","esal":{"$numberDouble":"3000.0"},"eaddr":"Chennai"}
{"_id":{"$oid":"5fe221b6573f5ff261265828"},"eno":{"$numberDouble":"400.0"},"ename":"Vinny","esal":{"$numberDouble":"4000.0"},"eaddr":"Delhi"}
{"_id":{"$oid":"5fe222ac573f5ff261265829"},"eno":{"$numberDouble":"777.0"},"ename":"Sunny","esal":{"$numberDouble":"1000.0"},"eaddr":"Mumbai"}
{"_id":{"$oid":"5fe222e9573f5ff26126582a"},"eno":{"$numberDouble":"888.0"},"ename":"Bunny","esal":{"$numberDouble":"2000.0"},"eaddr":"Hyd"}
{"_id":{"$oid":"5fe222e9573f5ff26126582b"},"eno":{"$numberDouble":"999.0"},"ename":"Chinny","esal":{"$numberDouble":"3000.0"},"eaddr":"Chennai"}
{"_id":{"$oid":"5fe2241b573f5ff26126582c"},"eno":{"$numberDouble":"77777.0"},"ename":"Sachin","esal":{"$numberDouble":"99999.0"},"eaddr":"Mumbai"}

mongoimport
mongoexport
mongodump
- **bson format**
- **bsondump**

Case-2: To restore all databases from dump by using mongorestore tool:

We have to use mongorestore command without any arguments.

```
C:\Users\lenovo\Desktop>mongorestore
2021-02-15T09:22:32.927+0530    using default 'dump' directory
2021-02-15T09:22:32.928+0530    preparing collections to restore from
2021-02-15T09:22:32.930+0530    reading metadata for
durgadb.employees from dump\durgadb\employees.metadata.json
2021-02-15T09:22:32.930+0530    reading metadata for
studentdb.students from dump\studentdb\students.metadata.json
2021-02-15T09:22:32.930+0530    reading metadata for
durgadb.employees3 from dump\durgadb\employees3.metadata.json
2021-02-15T09:22:33.232+0530    restoring studentdb.students from
dump\studentdb\students.bson
2021-02-15T09:22:33.232+0530    reading metadata for
studentdb.sssdb2 from dump\studentdb\sssdb2.metadata.json
2021-02-15T09:22:33.243+0530    no indexes to restore
2021-02-15T09:22:33.243+0530    finished restoring studentdb.students
(1 document, 0 failures)
2021-02-15T09:22:33.243+0530    reading metadata for
studentdb.sssdb1 from dump\studentdb\sssdb1.metadata.json
2021-02-15T09:22:33.756+0530    restoring durgadb.employees3 from
dump\durgadb\employees3.bson
2021-02-15T09:22:33.767+0530    no indexes to restore
2021-02-15T09:22:33.767+0530    finished restoring
durgadb.employees3 (1 document, 0 failures)
2021-02-15T09:22:33.767+0530    reading metadata for
durgadb.employees2 from dump\durgadb\employees2.metadata.json
2021-02-15T09:22:33.823+0530    restoring durgadb.employees from
dump\durgadb\employees.bson
2021-02-15T09:22:33.840+0530    no indexes to restore
2021-02-15T09:22:33.840+0530    finished restoring
durgadb.employees (8 documents, 0 failures)
```

2021-02-15T09:22:33.934+0530 restoring studentdb.sssdb2 from dump\studentdb\sssdb2.bson
2021-02-15T09:22:33.952+0530 no indexes to restore
2021-02-15T09:22:33.952+0530 finished restoring studentdb.sssdb2 (1 document, 0 failures)
2021-02-15T09:22:34.012+0530 restoring studentdb.sssdb1 from dump\studentdb\sssdb1.bson
2021-02-15T09:22:34.026+0530 no indexes to restore
2021-02-15T09:22:34.026+0530 finished restoring studentdb.sssdb1 (0 documents, 0 failures)
2021-02-15T09:22:34.176+0530 restoring durgadb.employees2 from dump\durgadb\employees2.bson
2021-02-15T09:22:34.192+0530 no indexes to restore
2021-02-15T09:22:34.192+0530 finished restoring durgadb.employees2 (0 documents, 0 failures)
2021-02-15T09:22:34.192+0530 11 document(s) restored successfully. 0 document(s) failed to restore.

Q. sir is it possible to auto dump everyday if we didnt dump manually?
cronjobs/schedulers ==>python script

Case-3: How to backup and restore a single database?

We have to specify required database as argument to mongodump command.
mongodump --db durgadb

C:\Users\lenovo\Desktop>mongodump --db durgadb
2021-02-15T09:30:28.759+0530 writing durgadb.employees3 to dump\durgadb\employees3.bson
2021-02-15T09:30:28.761+0530 done dumping durgadb.employees3 (1 document)

2021-02-15T09:30:29.058+0530 writing durgadb.employees2 to dump\durgadb\employees2.bson
2021-02-15T09:30:29.060+0530 writing durgadb.employees to dump\durgadb\employees.bson
2021-02-15T09:30:29.061+0530 done dumping durgadb.employees2 (0 documents)
2021-02-15T09:30:29.062+0530 done dumping durgadb.employees (8 documents)

C:\Users\lenovo\Desktop>tree /f dump
Folder PATH listing
Volume serial number is 000001A3 526A:E31B
C:\USERS\LENOVO\DESKTOP\DUMP
└───durgadb
        employees.bson
        employees.metadata.json
        employees2.bson
        employees2.metadata.json
        employees3.bson
        employees3.metadata.json

To restore a single database:

---

mongorestore --db durgadb dump\durgadb

C:\Users\lenovo\Desktop>mongorestore --db durgadb dump\durgadb
2021-02-15T09:34:35.985+0530 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2021-02-15T09:34:35.986+0530 building a list of collections to restore from dump\durgadb dir

2021-02-15T09:34:35.988+0530 reading metadata for durgadb.employees from dump\durgadb\employees.metadata.json
2021-02-15T09:34:35.989+0530 reading metadata for durgadb.employees3 from dump\durgadb\employees3.metadata.json
2021-02-15T09:34:35.989+0530 reading metadata for durgadb.employees2 from dump\durgadb\employees2.metadata.json
2021-02-15T09:34:36.172+0530 restoring durgadb.employees3 from dump\durgadb\employees3.bson
2021-02-15T09:34:36.184+0530 no indexes to restore
2021-02-15T09:34:36.184+0530 finished restoring durgadb.employees3 (1 document, 0 failures)
2021-02-15T09:34:36.556+0530 restoring durgadb.employees from dump\durgadb\employees.bson
2021-02-15T09:34:36.568+0530 no indexes to restore
2021-02-15T09:34:36.568+0530 finished restoring durgadb.employees (8 documents, 0 failures)
2021-02-15T09:34:36.661+0530 restoring durgadb.employees2 from dump\durgadb\employees2.bson
2021-02-15T09:34:36.680+0530 no indexes to restore
2021-02-15T09:34:36.680+0530 finished restoring durgadb.employees2 (0 documents, 0 failures)
2021-02-15T09:34:36.680+0530 9 document(s) restored successfully. 0 document(s) failed to restore.

Case-4: How to backup and restore a single collection:

---

mongodump --db durgadb --collection employees

C:\Users\lenovo\Desktop>mongodump --db durgadb --collection employees

2021-02-15T09:37:26.592+0530 writing durgadb.employees to dump\durgadb\employees.bson
2021-02-15T09:37:26.594+0530 done dumping durgadb.employees (8 documents)

C:\Users\lenovo\Desktop>tree /f dump
Folder PATH listing
Volume serial number is 000001B4 526A:E31B
C:\USERS\LENOVO\DESKTOP\DUMP
└───durgadb
employees.bson
employees.metadata.json

Restore a particular collection:

---

mongorestore --db durgadb --collection employees dump\durgadb\employees.bson

C:\Users\lenovo\Desktop>mongorestore --db durgadb --collection employees dump\durgadb\employees.bson
2021-02-15T09:40:50.660+0530 checking for collection data in dump\durgadb\employees.bson
2021-02-15T09:40:50.663+0530 reading metadata for durgadb.employees from dump\durgadb\employees.metadata.json
2021-02-15T09:40:50.813+0530 restoring durgadb.employees from dump\durgadb\employees.bson
2021-02-15T09:40:50.856+0530 no indexes to restore
2021-02-15T09:40:50.856+0530 finished restoring durgadb.employees (8 documents, 0 failures)
2021-02-15T09:40:50.857+0530 8 document(s) restored successfully. 0 document(s) failed to restore.

Note: We can take dump and restore all databases or a particular database or a particular collection.

Summary:

----

1. mongoimport --->To import data from the files

2. mongoexport --->To export mongodb data to the files

3. mongodump --->To create dump for database

4. bsondump --->To convert bson data into json data

5. mongorestore --->To restore data from the dump.

Q. Sir, one last question: I want to know how many databases contain "books" collection. How can I perform a database level search from mongo shell?

python script:

show dbs

use every db

show collections

if books collection is there then print database name

GUI Tools for MongoDB Operations:

Upto this we performed the database operations by using mongo shell. The advantage of using shell is we have to do everything so that learning opportunity is more.

But in real time usage of shell is not recommended because of the following reasons:

1. Auto completion is not available. We have to type complete command.

2. While writing complex queries, more error prone.

3. No help tips
4. Readability is not up to the mark.
5. No coloring
6. Operations will become very complex
7. Not that much convenient to use
etc

To overcome these problems we have to use GUI based tools like

1. **Robo 3T**
2. **Studio 3T**
3. **Compass**
4. **NoSQL Manager**

etc

1. Robo 3T:

The Robo 3T (formerly Robomongo) tool has been acquired by 3T Software Labs, the creators of the MongoDB client Studio 3T (formerly MongoChef).
It is freeware and lightweight GUI tool for MongoDB operations.

website: robomongo.org
Two tools: Robo 3T and Studio 3T
Robo 3T is freeware and Studio 3T is licensed.

Download: studio-3t-robo-3t-windows-double-pack.zip

Create a new database
Create a new collection
CRUD Opearations

2. Studio 3T:

Formerly MongoChef.

The most popular DB Tool for MongoDB operations.

https://robomongo.org/

studio-3t-x64.exe

3. NoSQL Manager for MongoDB:

NoSQL Manager for MongoDB Desktop tool for Mongo database management, administration and development

https://www.mongodbmanager.com/

NoSQL Manager for MongoDB Freeware

Free for non-commercial use, limited functional

mongodbmanagerfree_inst.exe

Features

Full support for MongoDB and MongoDB Enterprise versions from 2.2 to 4.4

Certified on MongoDB Enterprise. Supports Kerberos, LDAP and MONGODB-X509 authentications

Fully functional MongoDB UI Shell with code autocompletion, syntax highlighting and hints

Fully compatible with Amazon DocumentDB and MongoDB on Azure Cosmos DB

Support of replica sets, standalone hosts and sharded clusters connections
Easy-to-use document viewer and editor with Tree, Table and JSON view modes
Simple view and management of all MongoDB objects: databases, collections, views, indices, users, roles and functions
SSH tunneling for MongoDB connections
Performance monitoring tools
Import tables from MySQL and SQL Server databases
Export documents to CSV, XML, JSON and XLSX file formats
Import documents from JSON and CSV files
LINQ Query Tool

4. MongoDB Compass:
----

https://www.mongodb.com/products/compass

The GUI for MongoDB.
The easiest way to explore and manipulate your MongoDB data.
Visually explore your data.
Run ad hoc queries in seconds.
Interact with your data with full CRUD functionality.
View and optimize your query performance.
Available on Linux, Mac, or Windows.
Compass empowers you to make smarter decisions about indexing, document validation, and more.

Connection String:
mongodb://localhost:27017

MongoDB Indexing: