**Problem Statement-1**

In an e-commerce company, customers browse the website to place orders for various brands and items. In the process, they must register themselves, sign in with a username and password, search for specific brands and items, and place an order. While placing the order they can add/remove products from the shopping cart, discard the shopping cart, or proceed with the payment using various payment options. The data stored from all the transactions and activities happening on the website is complex in nature. It could be in structured or unstructured format. It might be in text, document, or JSON format. Being the Database Administrator of your organization, you are responsible to use a database which has flexible schema, reduces the operational procedure, and is highly scalable and reliable.

Approach

Installation and Configuration of MongoDB:

1.Download the MongoDB setup

2.Install MongoDB

3.Configure the Mongo Shell

4.Scripting through Mongo Shell

**Install MongoDB**

## What You’ll Need

MongoDB supports a variety of 64-bit platforms. Refer to the [Supported Platforms](https://docs.mongodb.com/manual/installation/#mongodb-supported-platforms) table to verify that MongoDB is supported on the platform to which you wish to install it.

## Procedure

### Install MongoDB

* **Windows**
* **macOS**
* **Linux**

Download the binaries from the [MongoDB Download Center](https://www.mongodb.com/download-center#production).

1. Open Windows Explorer/File Explorer.
2. Change the directory path to where you downloaded the MongoDB .msi file. By default, this is %HOMEPATH%\Downloads.
3. Double-click the .msi file.
4. The Windows Installer guides you through the installation process.

If you choose the **Custom** installation option, you may specify an installation directory.

MongoDB does not have any other system dependencies. You can install and run MongoDB from any folder you choose.

**NOTE**

This tutorial assumes that you installed MongoDB in C:\Program Files\MongoDB\Server\4.2\.

### Run MongoDB

* **Windows**
* **macOS**
* **Linux**

**WARNING**

Do not make [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe) visible on public networks without running in “Secure Mode” with the auth setting. MongoDB is designed to be run in trusted environments, and the database does not enable “Secure Mode” by default.

**1 Set up the MongoDB environment**

MongoDB requires a [data directory](https://docs.mongodb.com/manual/reference/glossary/#term-dbpath) to store all data. MongoDB’s default data directory path is the absolute path \data\db on the drive from which you start MongoDB. Create this folder by running the following command in a Command Prompt:

md \data\db

You can specify an alternate path for data files using the [--dbpath](https://docs.mongodb.com/manual/reference/program/mongod/#cmdoption-mongod-dbpath) option to [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe), for example:

"C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe" --dbpath d:\test\mongodb\data

If your path includes spaces, enclose the entire path in double quotes, for example:

"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe" --dbpath "d:\test\mongo db data"

You may also specify the dbpath in a [configuration file](http://docs.mongodb.com/manual/reference/configuration-options).

**2 Start MongoDB**

To start MongoDB, run [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe). For example, from the **Command Prompt**:

"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe"

This starts the main MongoDB database process. The waiting for connections message in the console output indicates that the [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe) process is running successfully.

Depending on the security level of your system, Windows may pop up a **Security Alert** dialog box about blocking “some features” of C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe from communicating on networks. All users should select Private Networks, such as my home or work network and click Allow access. For additional information on security and MongoDB, please see the [Security Documentation](http://docs.mongodb.com/manual/security).

**3 Verify that MongoDB has started successfully**

Verify that MongoDB has started successfully by checking the process output for the following line:

[initandlisten] waiting for connections on port 27017

The output should be visible in the terminal or shell window.

You may see non-critical warnings in the process output. As long as you see the log line shown above, you can safely ignore these warnings during your initial evaluation of MongoDB.

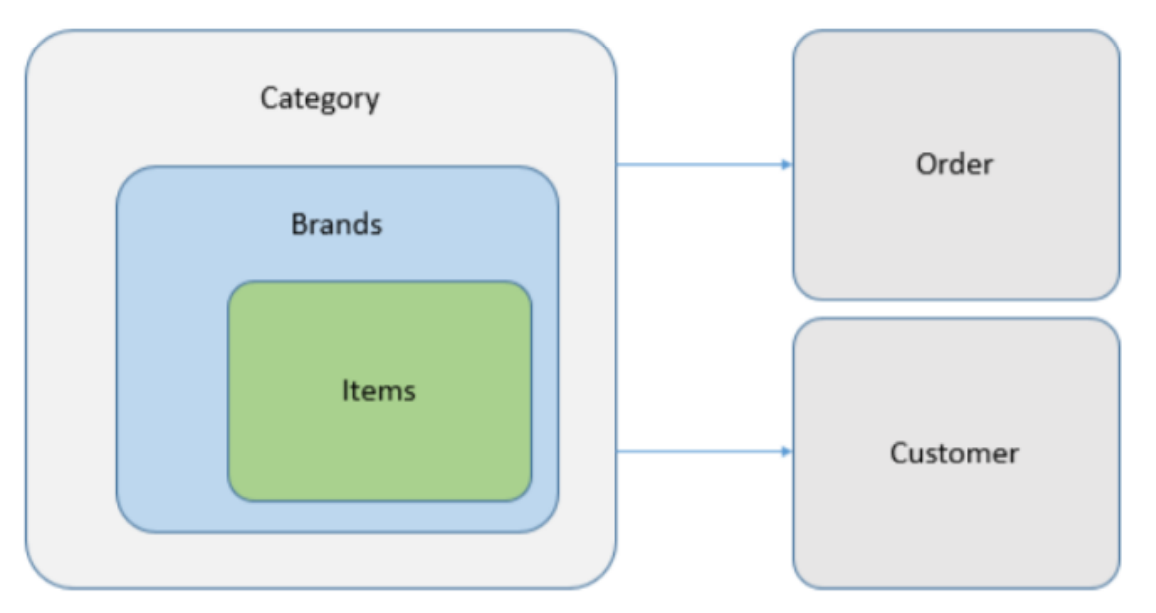
**4 Connect to MongoDB**

To connect to MongoDB through the [~bin.mongo.exe](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell, open another **Command Prompt**.

"C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe"

**Problem Statement 2-**

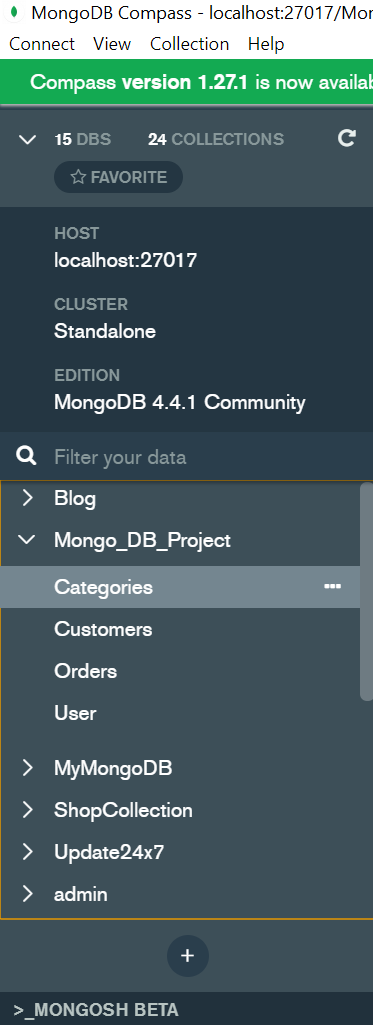
As per given diagram and problem statement No SQL, document-based schema best for it, which comes under MongoDB because it is dynamic schema which updated frequently.



**Problem Statement 3-**

After you create a logical schema, next step is to create a database named as “Mongo\_DB\_Project” which would be handled by a user “Adam”. You must maintain orders, customers and categories of data with brands and items in segregated form. You are also responsible for modifying the documents and removing duplicate or redundant documents.

1.Create Database “Mongo\_DB\_Project” using “use DBName”

****2.Create User with roles:

db.createUser({

user: "Adam",



pwd: "adam",

roles: ["readWrite","dbAdmin"]

});



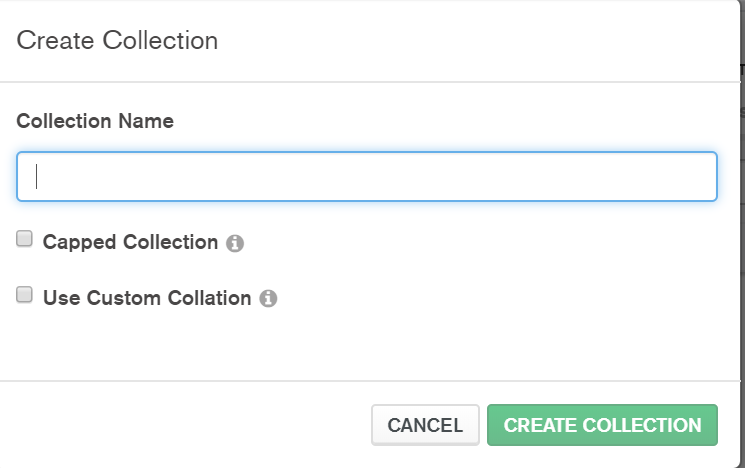
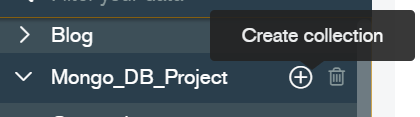
3.Create 3 Collections as below:

a.Orders: Order details and item details

b.Categories: Categories with brands and items details

c.Customers: Customer details with Items

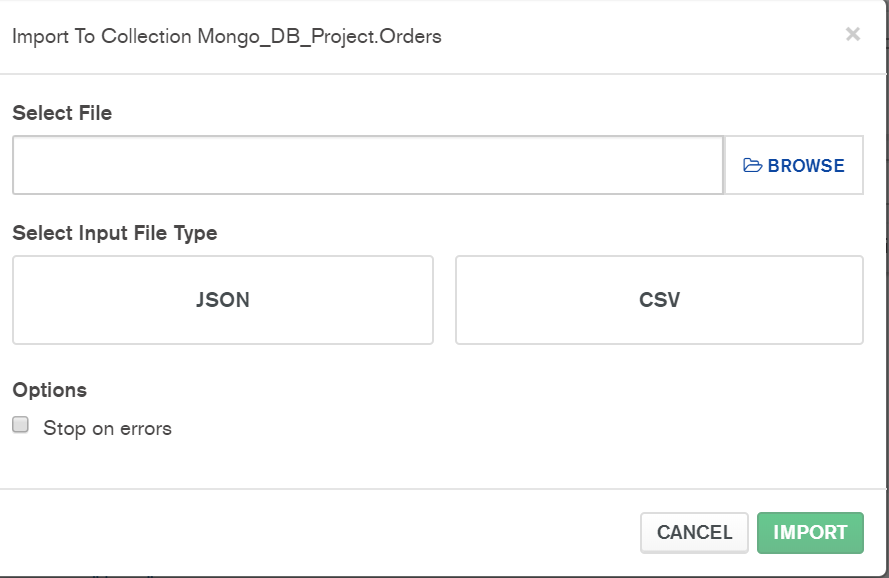
Using Mongo Compass, Its very easy to cerate collection , just click on **create collection** button, then Collection window will be popup, just write collection name and click on create



4.Insert documents in the collections:

1.We can insert document using mongo Compass

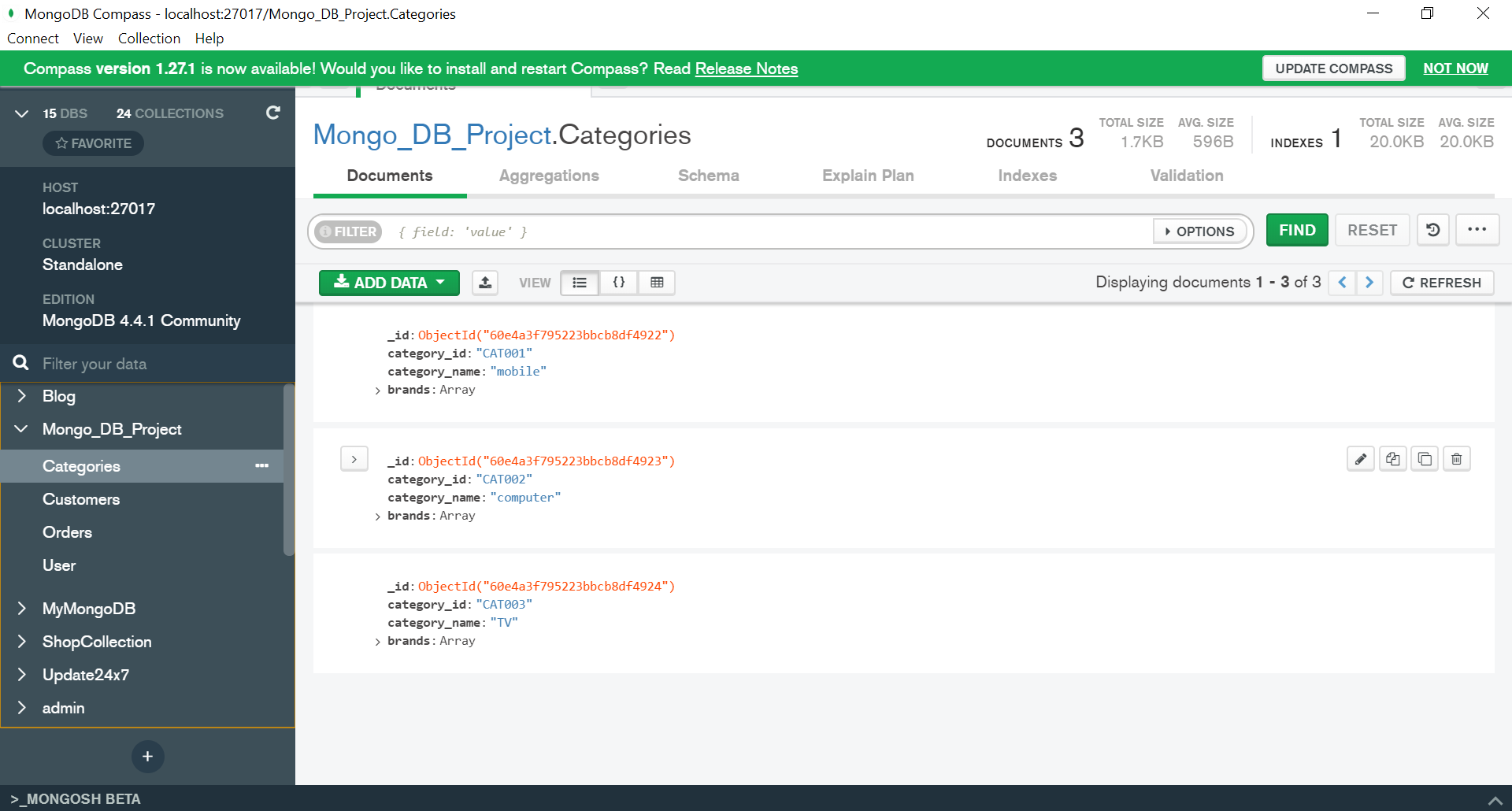
* 1. Open the collection, then click on **ADD DATA🡪** click on **Import File**



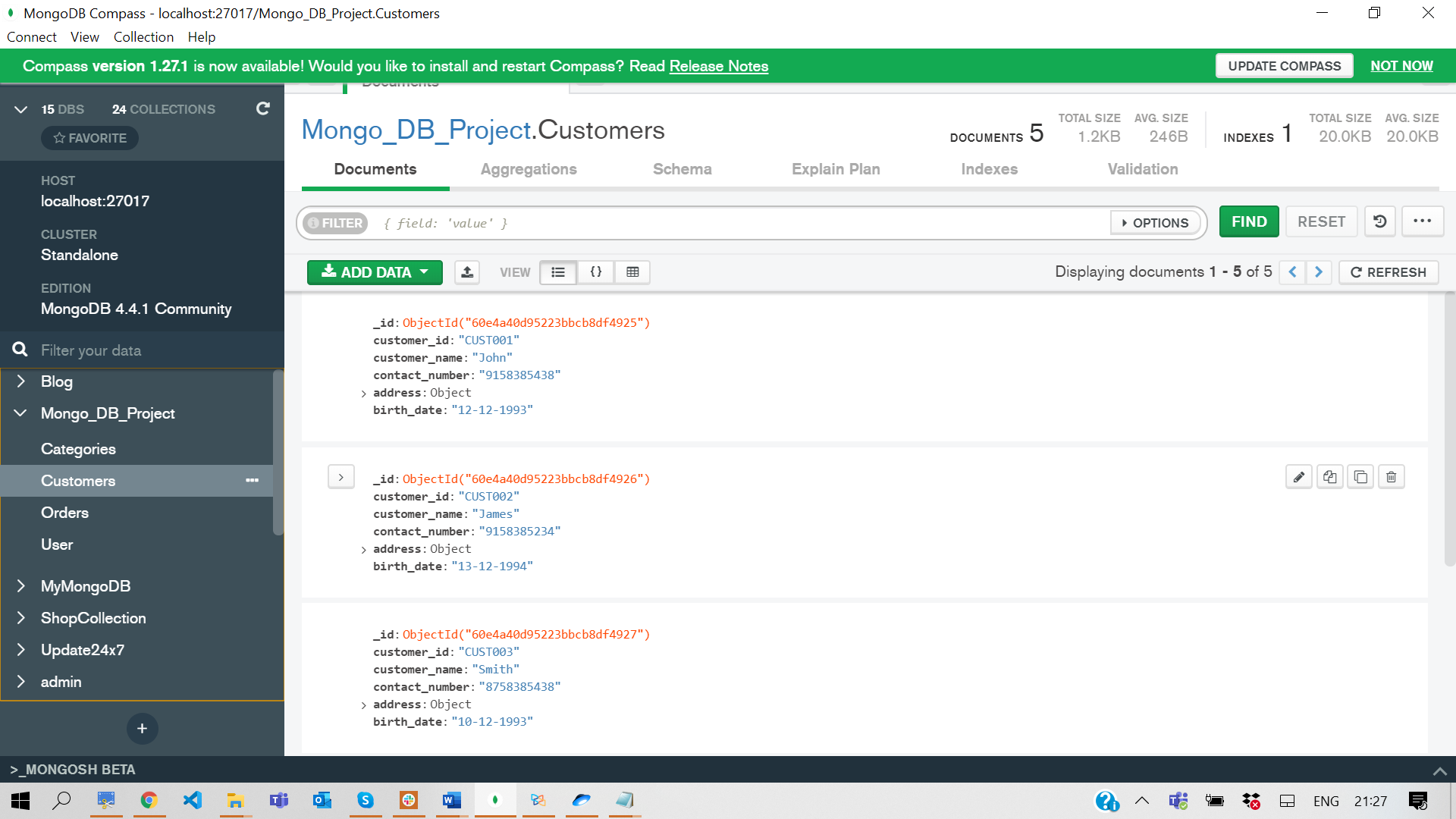
Browse the file which has Json/CSV format type data.

Or we can also insert data manually

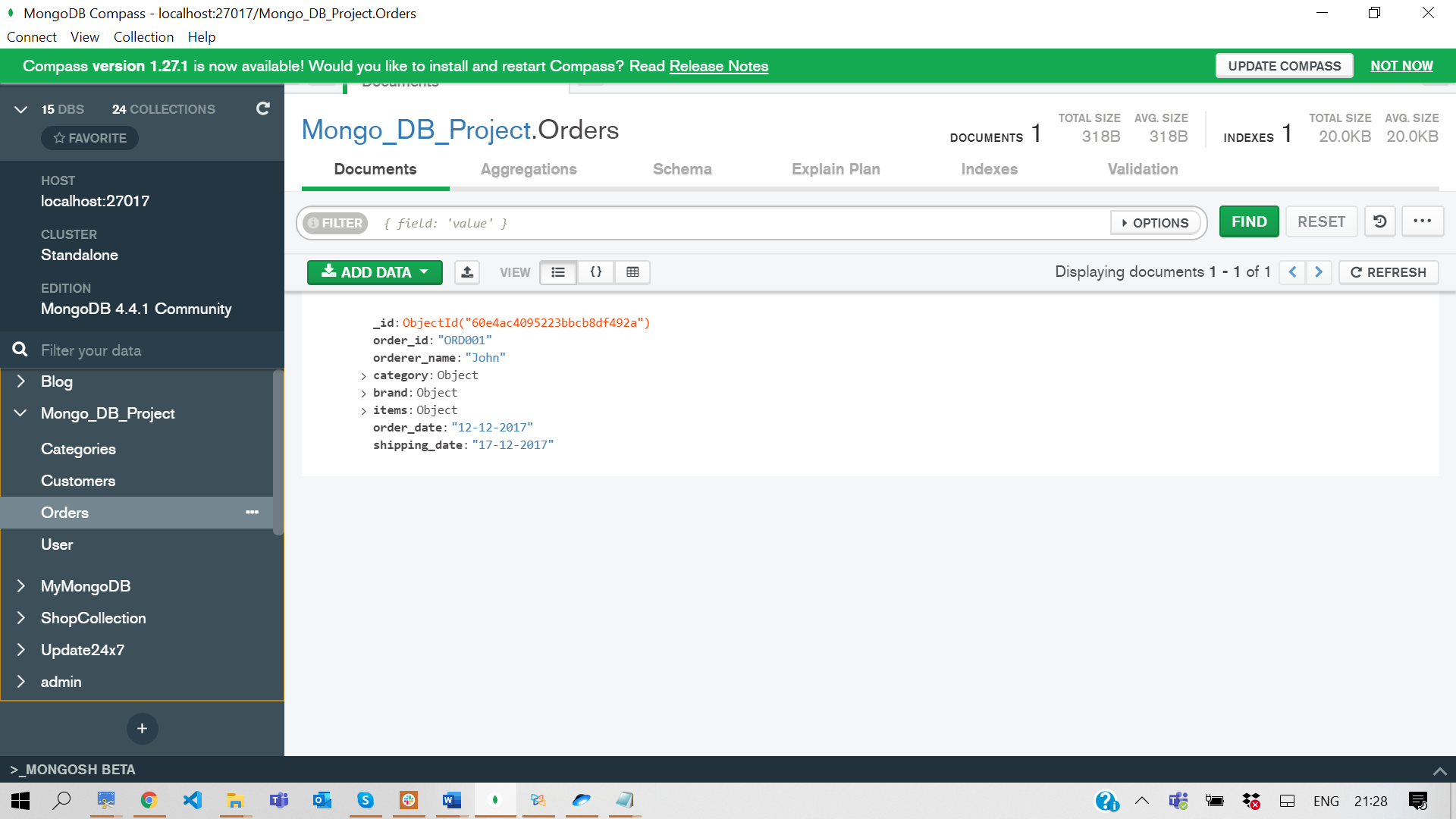
Categories Collection:



Customers Collection:



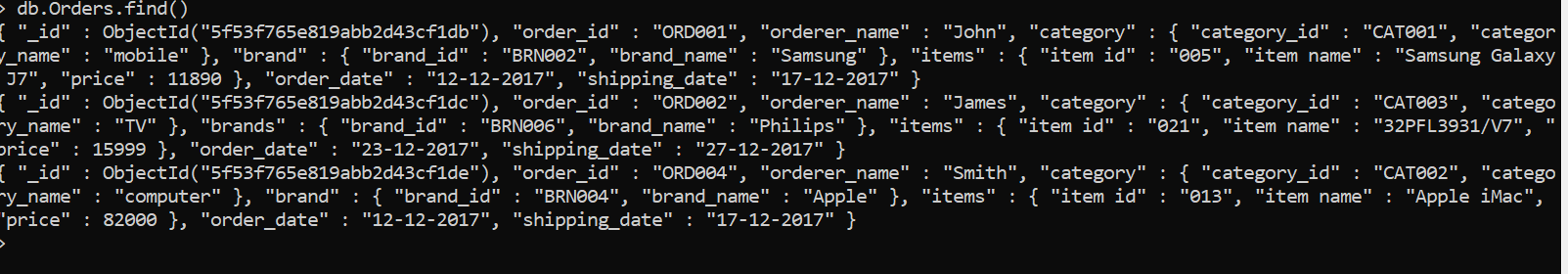
Orders Collection:

command 

5.Query data from MongoDB using find

**Syntax: db.collectionname.find()**

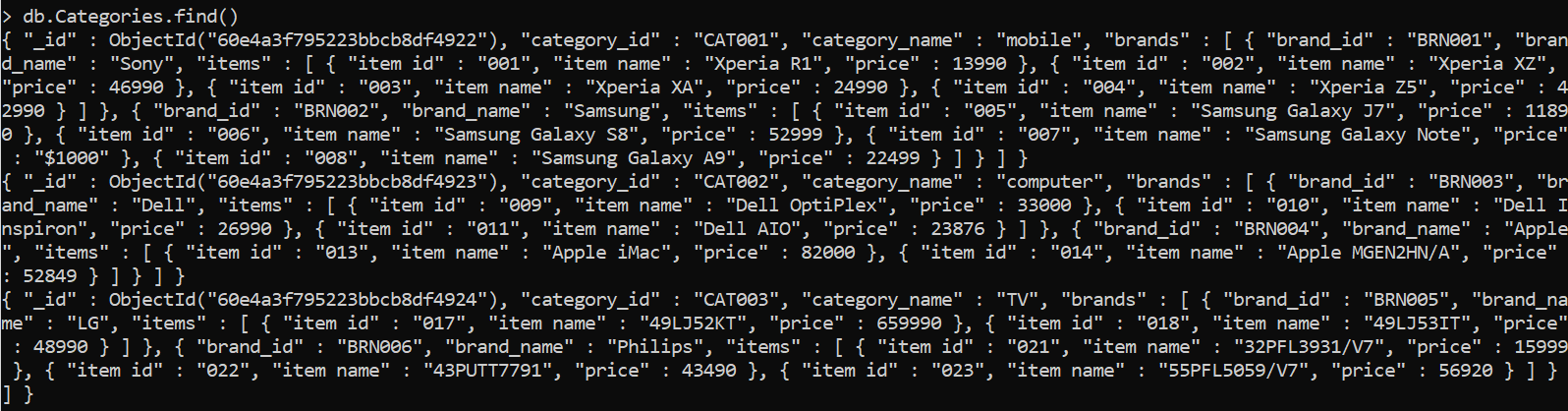
Ex:1) db.Orders.find()



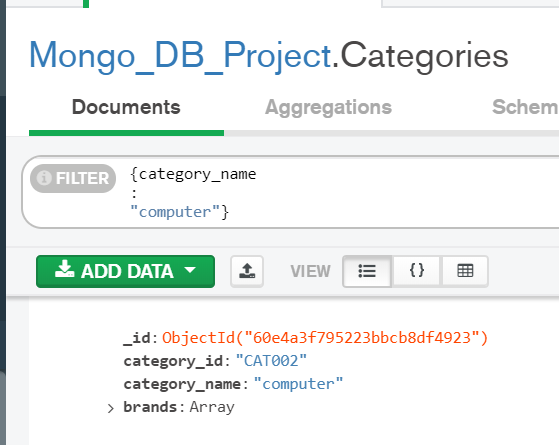
2) db.Customers.find()



3) db.Categories.find()



OR



6.The price of Dell OptiPlex computer has increased from 32200 to 33000. Update the document in Category collection accordingly.



7.Observe the documents to find a duplicate document with different order\_id.

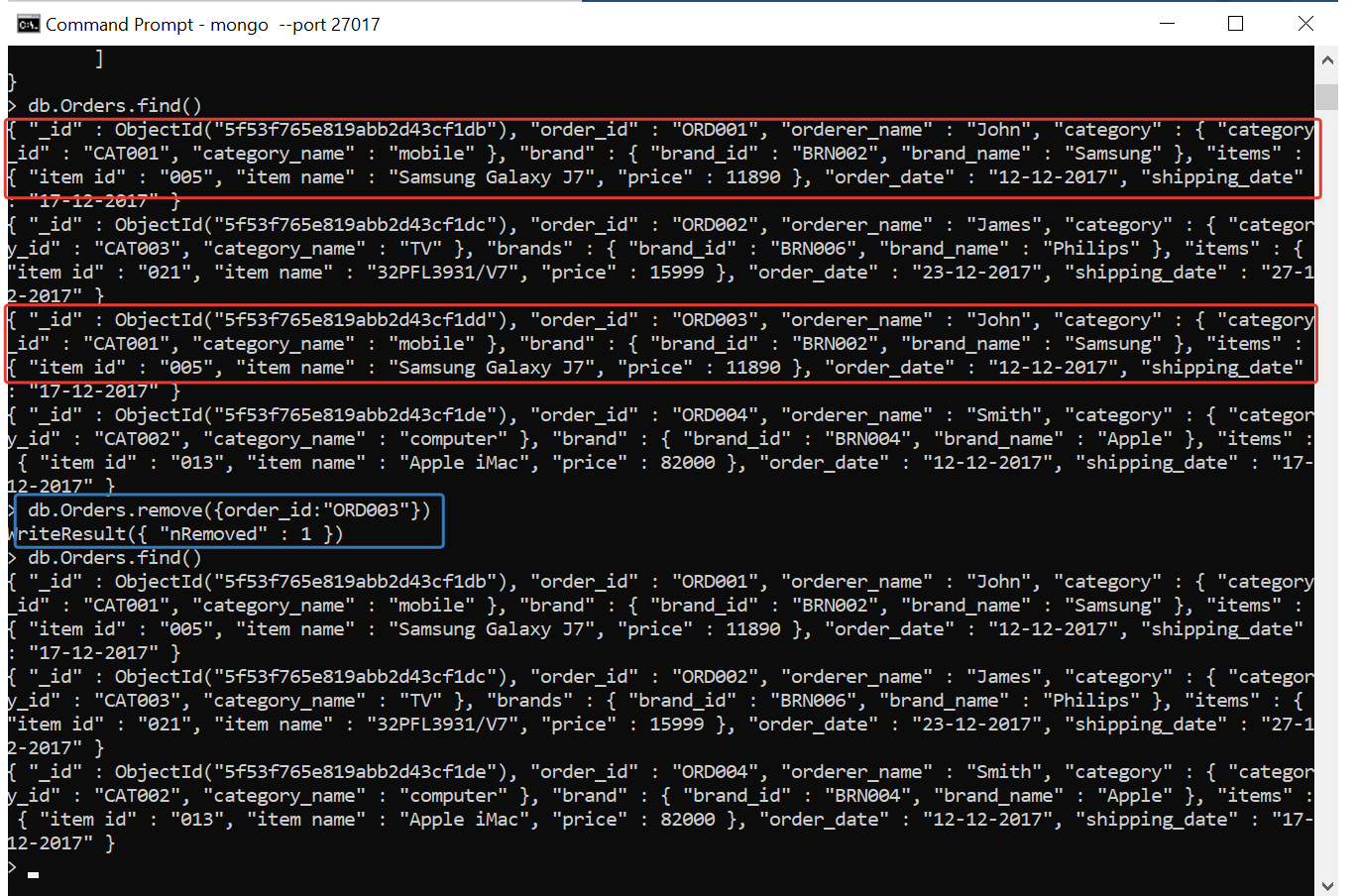
8.Delete the duplicate document(s)

As we know orderid:001 and orderid:003 are ids are only different but rest of the documents are same so no need to maintain duplicate documents again so we can eliminate orderid:003

Syntax: db.collectionname.remove({condition},n)

Eg: db.Orders.remove({order\_id:"ORD003"})

MongoDB Enterprise > db.Orders.remove({order\_id:"ORD003"}) WriteResult({ "nRemoved" : 1 })

O/P: Problem Statement

**Problem Statement 4**

Your organization keeps consolidated data in a single file for all categories with distinct brands and items for each brand with item details. So, you must take a backup of the data.

1. Backup a database with mongoexport

2. Restore the data using mongoimport

1. Backup a database with mongoexport:

Syntax: Mongoexport.exe -d databasename -c collectionname -o “location where you want to export”

**C:\Program Files\MongoDB\Server\4.4\bin>mongoexport.exe -d Mongo\_DB\_Project -c Orders -o "C:\MongoDB\_Certification\_Project\Orders.json"**

**2021-07-06T15:52:16.329+0530 connected to: localhost**

**2021-07-06T15:52:16.340+0530 exported 4 records**

**C:\Program Files\MongoDB\Server\4.4\bin>mongoexport.exe -d Mongo\_DB\_Project -c Categories -o "C:\MongoDB\_Certification\_Project\Categories.json"**

**2021-07-06T15:53:28.572+0530 connected to: localhost**

**2021-07-06T15:53:28.586+0530 exported 3 records**

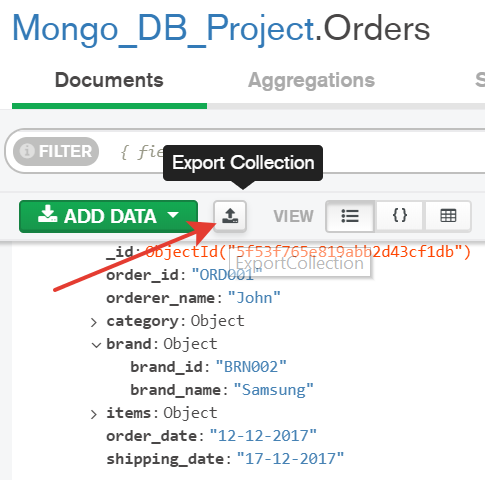
**C:\Program Files\MongoDB\Server\4.4\bin>mongoexport.exe -d Mongo\_DB\_Project -c Customers -o "C:\MongoDB\_Certification\_Project\Customers.json"**

**2021-07-06T15:54:11.682+0530 connected to: localhost**

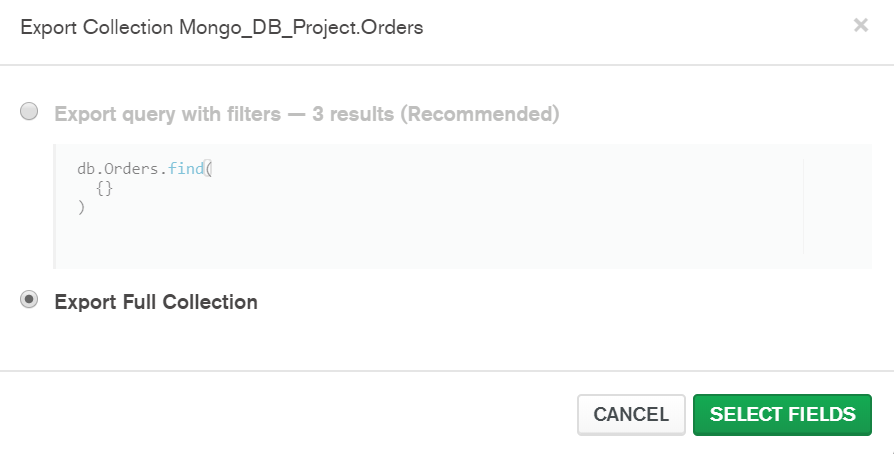
**2021-07-06T15:54:11.693+0530 exported 5 records**

OR

1. Click on Export Collection Button

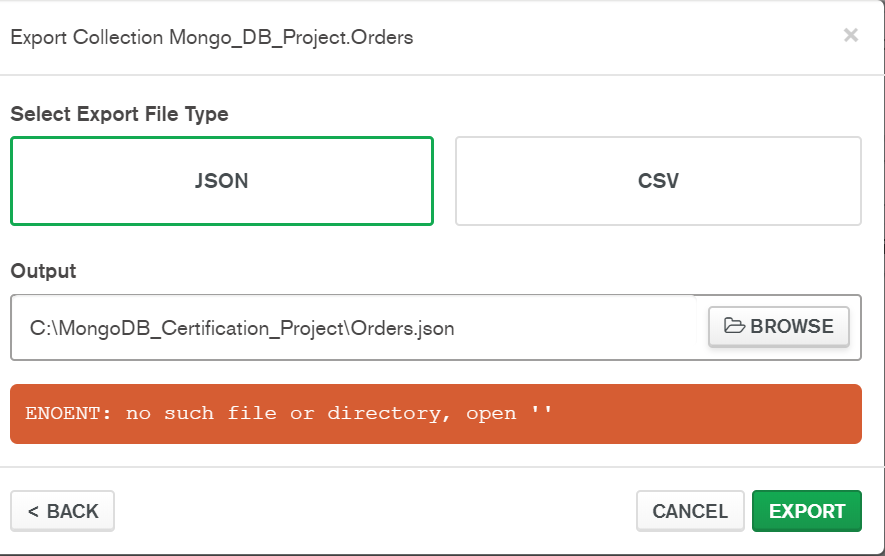


1. **Choose Export Full Collection, click on SELECT FILEDS**

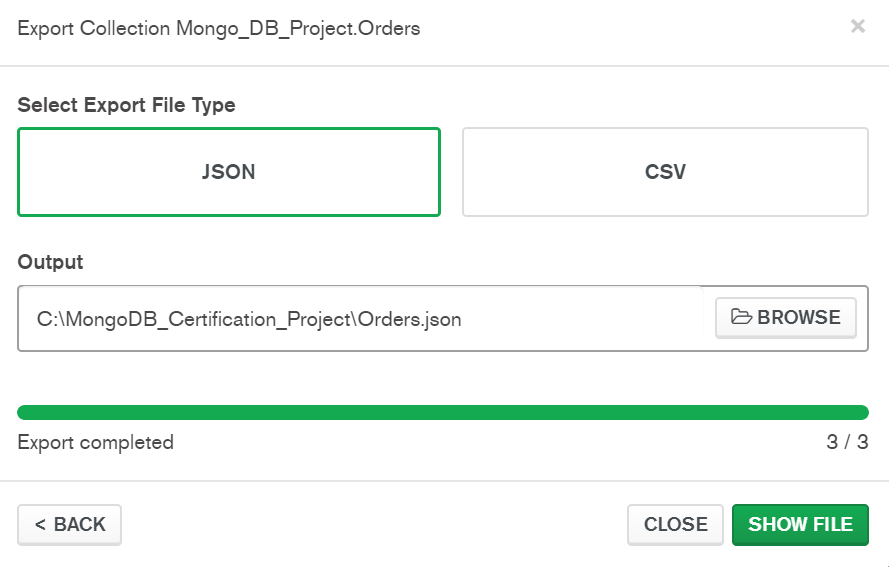


1. Choose File Type and File Path where you want to store export file

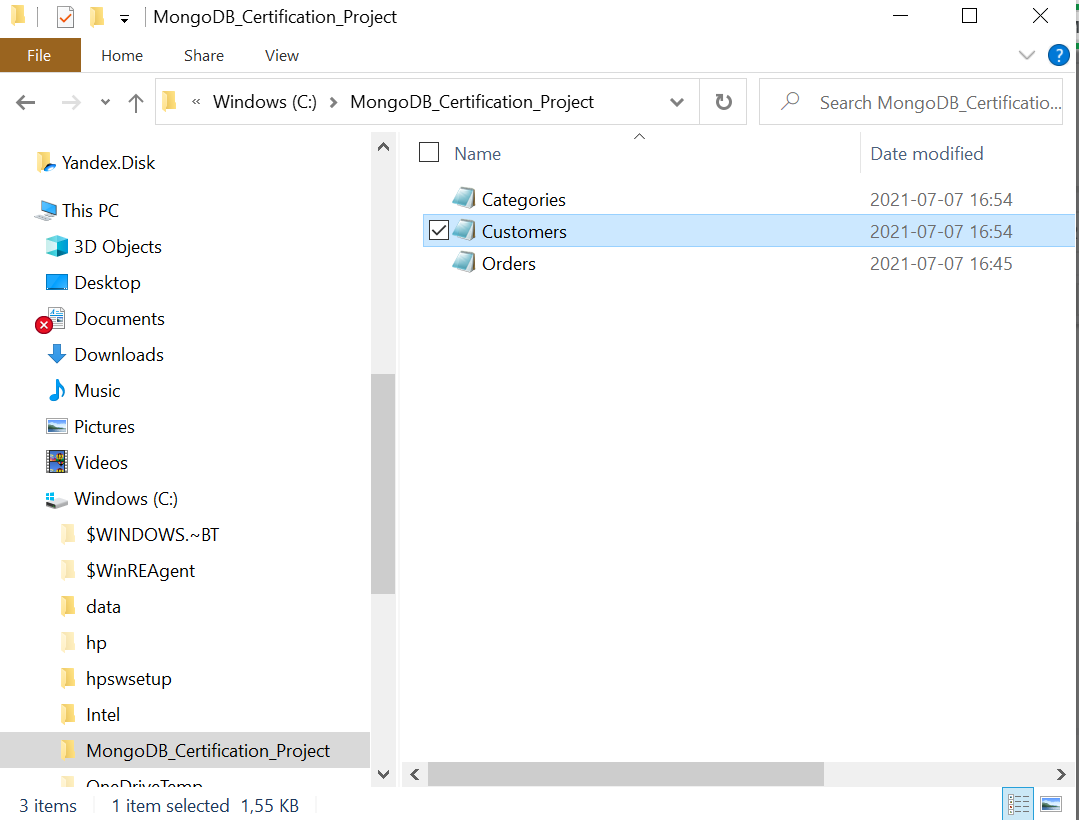
Ex-



1. Click On Export button



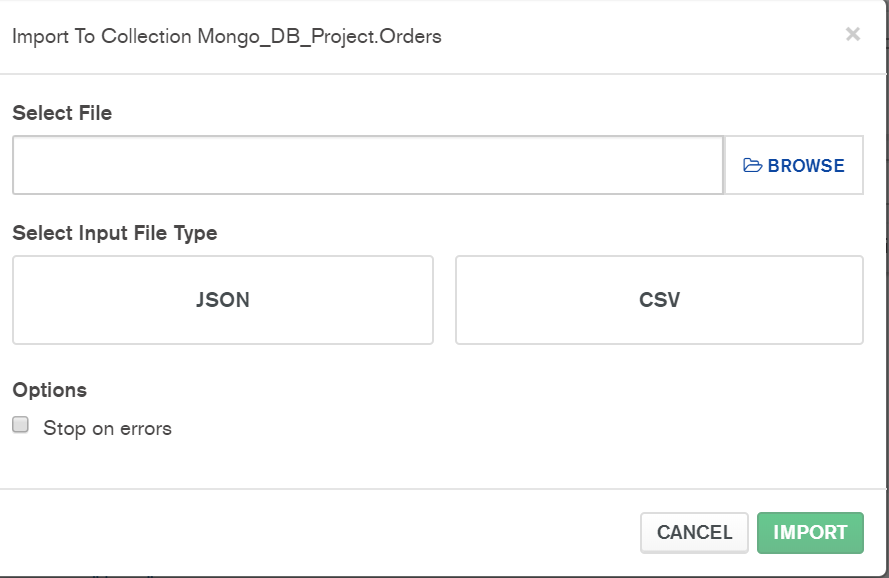
1. Similar, we export Categories, Customers collection…



2. Restore the data using mongoimport

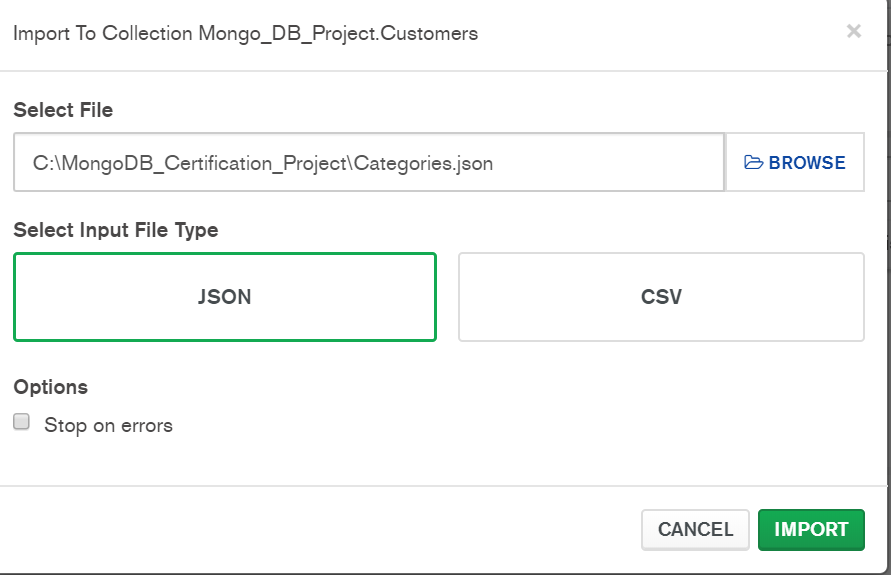
1.We can import document using mongo Compass

* 1. Open the collection, then click on **ADD DATA🡪** click on **Import File**



Browse the file which has Json/CSV format type data.

Or we can also insert data manually



Restore the data using mongoimport

Syntax: mongoimport.exe -d databasename -c collectionname –file “location of file”

Note: As already we have inserted in this database again we no need to import but giving example to show how we can use

Eg**: C:\Program Files\MongoDB\Server\4.4\bin>mongoimport.exe -d Mongo\_DB\_Edureka\_Project -c Orders --file "D:\MongoDB\_Certification\_Project\Orders.json"**

**2021-07-06T16:02:45.777+0530 connected to: localhost**

**2021-07-06T16:02:45.797+0530 imported 4 documents**

**C:\Program Files\MongoDB\Server\4.4\bin>mongoimport.exe -d Mongo\_DB\_Edureka\_Project -c Customers --file "D:\MongoDB\_Certification\_Project\Customers.json"**

**2021-07-06T16:03:24.590+0530 connected to: localhost**

**2021-07-06T16:03:24.657+0530 imported 5 documents**

**C:\Program Files\MongoDB\Server\4.4\bin>mongoimport.exe -d Mongo\_DB\_Edureka\_Project -c Categories --file "D:\MongoDB\_Certification\_Project\Categories.json"**

**2021-07-06T16:03:47.309+0530 connected to: localhost**

**2021-07-06T16:03:47.318+0530 imported 3 documents**

**Problem Statement 5**

With the increasing data, the query performance of MongoDB has declined. You are responsible for optimizing the query performance of the database. Which steps would you take?

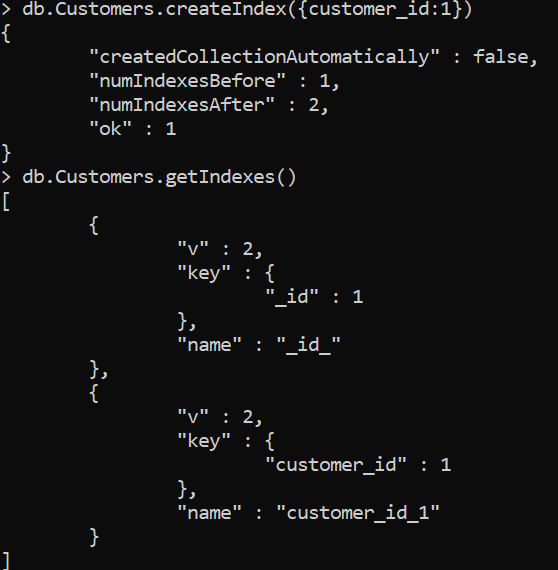
Approach Perform indexing Creation of index:

**Syntax to create index: db.collectionname.createIndex({value on what we want to create index})**

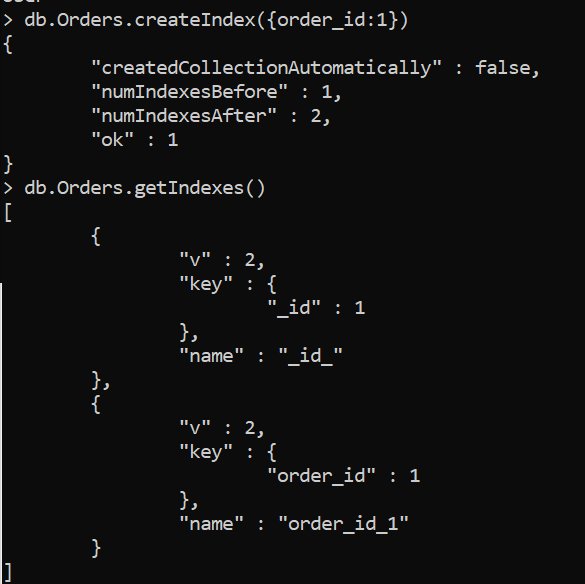
**Ex: db.Categories.createIndex({category\_id:1}**

****

**Ex: db.Customers.createIndex({customer\_id:1}**

****

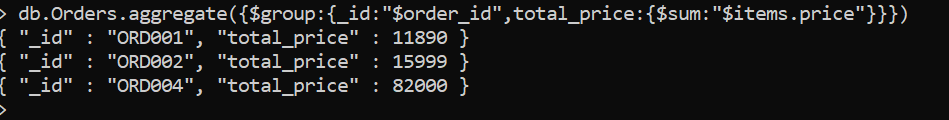
**Ex: db. Orders.createIndex({order\_id:1}**

****

**Problem Statement 6**

Your manager has asked you to provide total price obtained from order placed for various items.

Approach Perform aggregation functions

Solution: Here we have to consider Orders collection in order to perform aggregation functions

**Problem Statement 8:**

You have a lot of unwanted, empty and invalid collections which are consume the memory space of the server machine. Some users have also complained about some warnings and errors encountered while performing various tasks. What would you do?

Solution

1. Run the mongod.exe and mongo.exe files.

2. Open Nosqlclient application.

3. Connect to the Mongo\_DB\_Project database.

4. Expand the Management tab in the Navigation Pane and select Admin Queries option.

5. To get information of all the valid and invalid collection, select validate Collection from the

Admin Query dropdown and enter Order in the Collection name field.

6. Click Execute and observe the result below the Execute button.

7. To get information of all the valid and invalid collection, select list Databases from the

Admin Query dropdown and click Execute. You can observe the result below the Execute

button.

8. To get log of the activities which encountered some warnings, select command from the

Admin Query dropdown and execute the following command to observe result:

db.runCommand({ getLog: "startupWarnings"})

**Problem Statement 9:**

The developer of your organization have created a new application. You are asked to

perform a test for the same application, where you will insert few documents in MongoDB

database using a Rest Client.

Solution

1. Create a collection organization\_data in Mongo\_DB\_Project database.

2. Download Tomcat binary distribution: https://tomcat.apache.org/download-80.cgi

3. Download latest binary distribution for example: 32-bit/64-bit Windows Service Installer

4. Locate the downloaded file and double click to start installation.

5. In the installation wizard, specify and note the Username and Password in the

Configuration step.

6. Go to C: > Program Files> Apache Software Foundation>Tomcat 8.5 then bin folder.

Launch Tomcat8.exe

7. Extract the zip file provided to you with this project in your local machine.

8. Now, in the browser, enter http://localhost:8080.

9. Select Manager Apps option available at the right side of your screen.

10. Insert username/password.

11. Scroll down to WAR file to deploy section.

12. Choose .war file to deploy and hit Deploy button.

13. Now, click on MongoRest in Application section. To access MongoDB Database using a

Rest application perform the following steps:

14. Run the mongod.exe and mongo.exe file.

15. Open POSTMAN application from Google Chrome application. 16. Insert the following

details to delete a document from the MongoDB database.

**Problem Statement 10:**

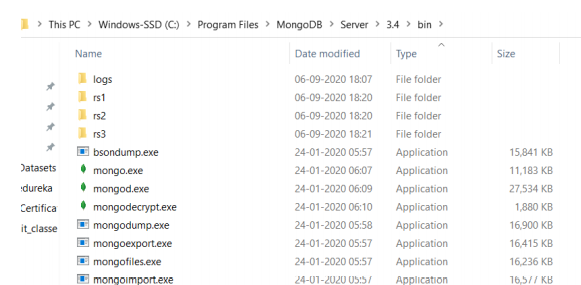
As a database Administrator, you must be well prepared to face situations like system failure, disaster management recovery. You should have replica sets created of your server database.

Approach 1. Create primary replica and secondary replica

Solution:

Create folders of rs1 rs2 rs3 and logs where the binaries are present

Ex: **C:\Program Files\MongoDB\Server\3.4\bin>mkdir rs1 rs2 rs3 logs**



Open 3 command prompts to create replica servers

Execute these commands in respective individual command prompts

**start /b mongod --replSet rs --dbpath rs1 --logpath logs\rs1.log --logappend**

**start /b mongod --replSet rs --dbpath rs2 --port 27018 --logpath logs\rs2.log --logappend**

**start /b mongod --replSet rs --dbpath rs3 --port 27019 --logpath logs\rs3.log –logappend**

Once executes servers starts running and by default no server is primary or secondary we

have to make it as one primary and rest of them as secondaries

Eg:Once connected to client we can give rs.status() to check by default it is not any of primary

or secondary give rs.inititate() to make it as primary, attached screenshot for reference



After few seconds it turns into primary and we can check status now using rs.status()



Ex: From Primary we have to create the rest of the sets as secondaries for that we have to give

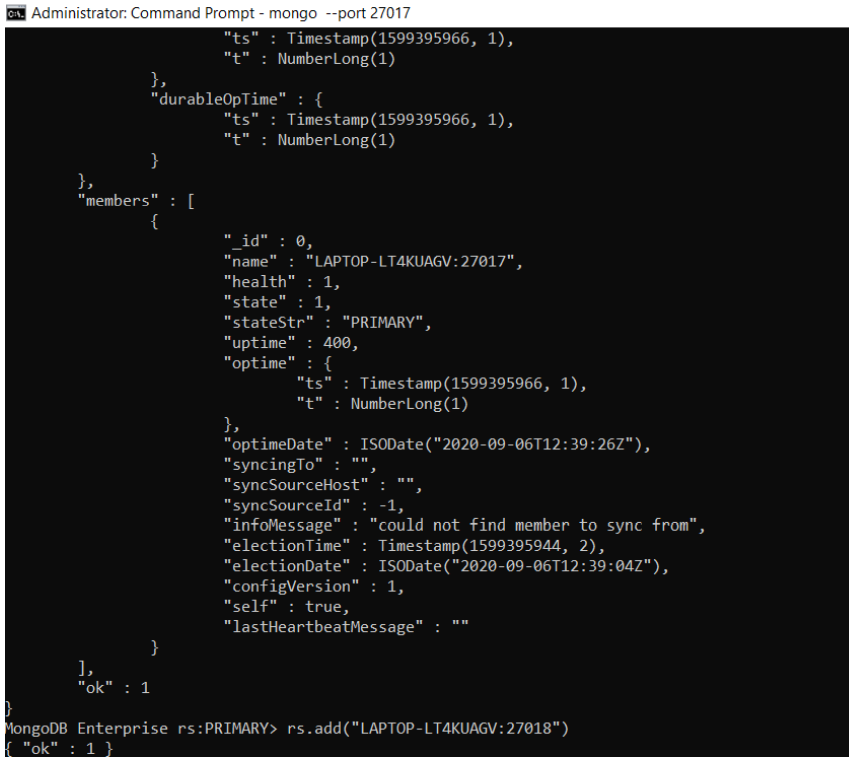
**rs.add(“name of it”)**

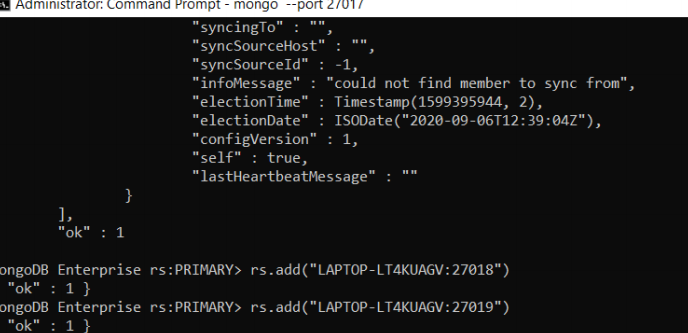
**Ex: MongoDB Enterprise rs:PRIMARY> rs.add("LAPTOP-LT4KUAGV:27018")**

**{ "ok" : 1 }**

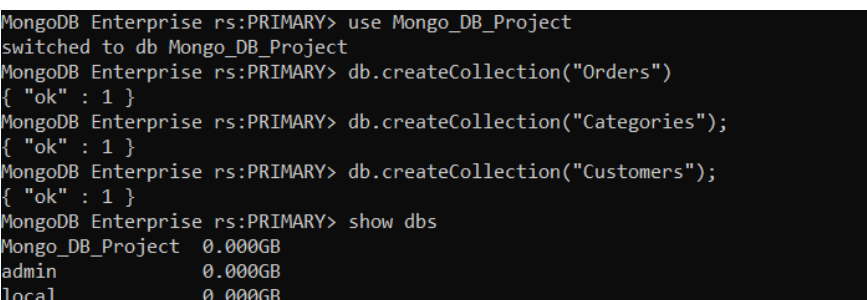
**MongoDB Enterprise rs:PRIMARY> rs.add("LAPTOP-LT4KUAGV:27019")**

**{ "ok" : 1 }**

****

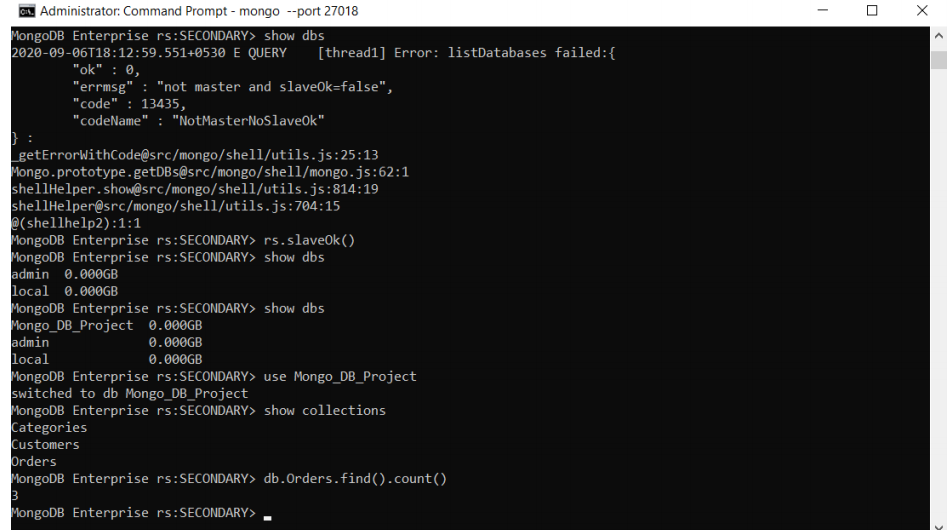


As primary supports both read and write operations we can create the databases, collections and inserting documents etc

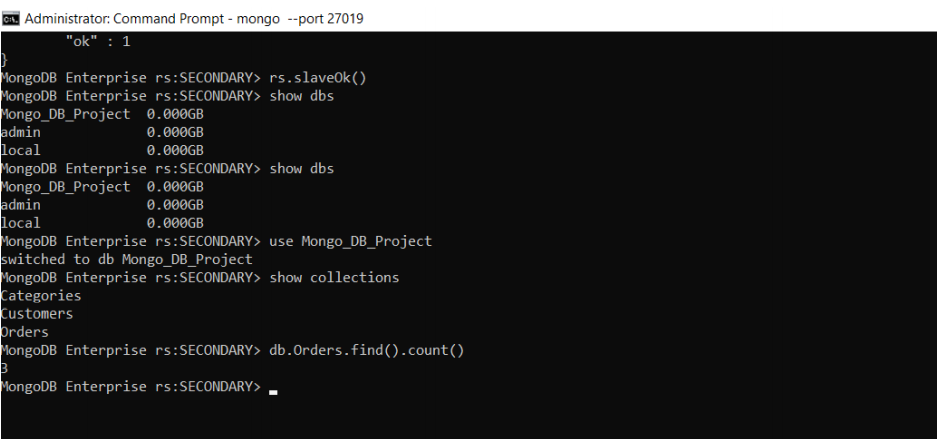


By default secondaries cannot perform read operations to make it enable we have to give

**rs.slaveOk()** in secondaries

****

**rs.slaveOk()**

****

Note: The data which is present in primary replica is the same can be available in

secondary replicas