MongoDB Activity

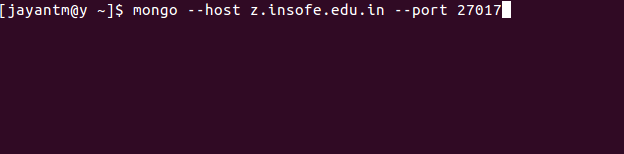
# Learning Outcomes:

1. Interacting with the mongo shell.
2. Creating a database in mongodb
3. Collections in mongodb.
4. Loading data json data into a collection (mongoimport)
5. Performing CRUD Operatins using mongo shell.
6. Querying MongoDB with conditions
7. Aggrgation with MongoDB.

Mongodb is installed on three nodes of the cluster with the following host name and port numbes.

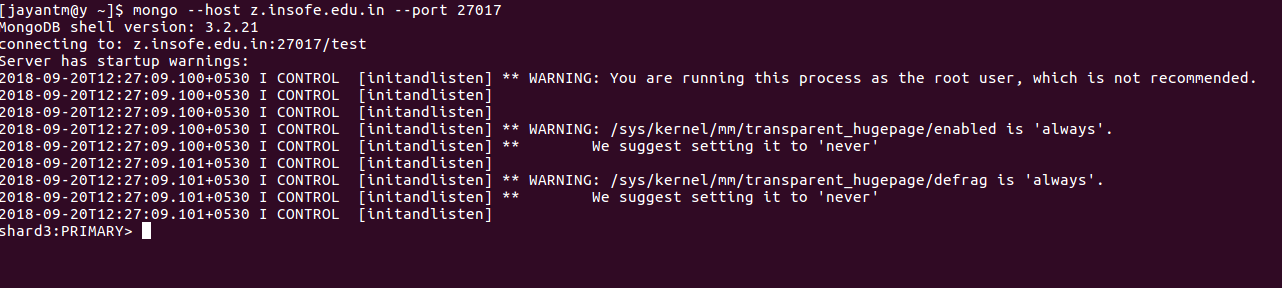
HostName : v.insofe.edu.in, y.insofe.edu.in, z.insofe.edu.in , port : 27017

Interacting with the mongo shell

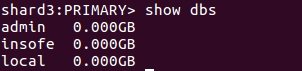


* **To login to the mongo shell**, execute the following command:

The mongo shell prompt will be as shown below.



Display Databases in the mongo shell:



Creating a database in mongodb

The below command will create the database. *use <database name>*



To display the database being used, print the below command.



Collections in mongodb

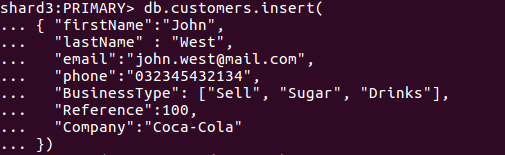
Collections are basic structure in which data is stored. They are similar to tables in relational databases.

Mongodb creates a collection while inserting the data into collection.

Creating a collection in mongodb

In the below step, customers collection is created while inserting the data into the collection.

db.collectionName.insert({})



Loading data into collection in mongodb

Lets say we need to import the data into the file from json or csv file, then one can use mongoimport to import the data into mongodb.

*The command could be executed from the terminal:*

*Command to be executed from the linux terminal and not from mongo prompt.*

*mongoimport --host z.insofe.edu.in --port 27017 --db usermanaged --jsonArray --collection transactions --drop --file /home/jayantm/datasets/transactions\_mong.json*

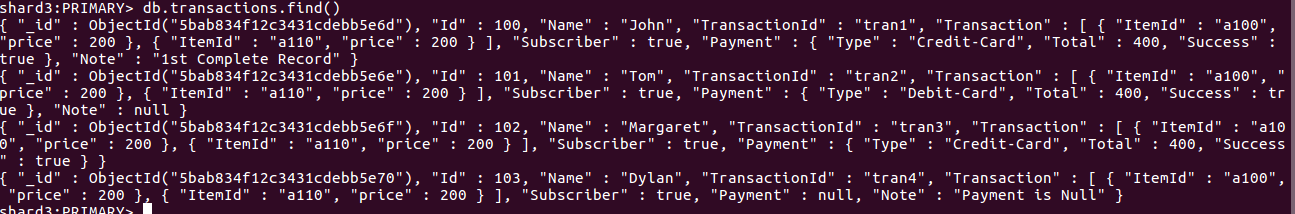
Printing the data into collection in mongodb

find()

find prints the output to the console. Most of the output is printed as a single line.

Syntax : db.<collectionName>.find()

Example : *db.transactions.find()*



find().pretty()

The output prints in more structured format using the pretty() option.

Syntax : db.<collectionName>.find().pretty()

Example : *db.transactions.find().pretty()*

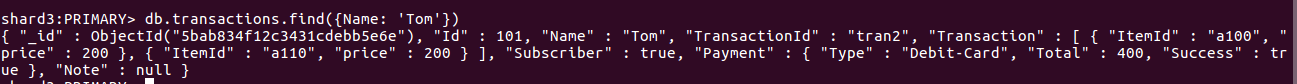


finding value in collection using key value pair in mongodb

Finding the data based on name in the collection.

Syntax : db.<collectionName>.find({key:value})

Example : *db.transactions.find({Name: 'Tom'})*



Inserting data in mongodb :

Data can be inserted into mongodb using the following syntax.

Syntax : db.collections.insert({})

*db.transactions.insert({*

*"Id": 110,*

*"Name": "Inserted Record",*

*"TransactionId": "tranNew1",*

*"Transaction": [*

*{*

*"ItemId":"c324",*

*"price": 456*

*},*

*{*

*"ItemId":"d456",*

*"price": 543*

*}*

*],*

*"Subscriber": false,*

*"Payment": {*

*"Type": "Debit-Card",*

*"Total": 999,*

*"Success": true*

*},*

*"Note":'Hello World'*

*})*

Selecting data in mongodb :

Finding the data in the transactions collection using key value pair.

Syntax : db.<collectionName>.find({key, Value Pair})

*db.transactions.find({“Id”:110})*

Updating data in mongodb:

Updating the data in collection using the update function on the collection.

Syntax : db.<collectionName>.update({keyValue pairs and values to be set})

*db.transactions.update({“Id”:110},{$set:{Name:'Updated Record',Note:'Updated!'}})*

Deleting data in mongodb:

Syntax : db.<collectionName>.remove({key value pair})

*db.transactions.remove({“Id”:110})*

Querying data with Conditions:

* find record in transactions where name is Tom :

*db.transactions.find({Name:”Tom”})*

* find a transaction where amount is 400:

*db.transactions.find({"Payment.Total": 400 })  
db.transactions.find({"Payment.Total": {$eq: 400}})*

* find a transactions collection where price is greater than 400

*db.transactions.find({"Transaction.price": {$gt: 400} })*

*db.transactions.find({"Transaction.price": {$gte: 400} })*