BIG DATA LAB

Name:S.L.A.Laisha
USN:1NT19IS147
SEC:C1

Date: 05-04-2022

To check if Mongodb is installed or not and version:

```
hdoop@admin1-HP-280-G4-MT-Business-PC:~$ mongo
MongoDB shell version v5.0.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8d344186-416a-480a-a10b-31997c22046a") }
MongoDB server version: 5.0.5
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-04-05T08:51:24.915+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.
mongodb.org/core/prodnotes-filesystem
        2022-04-05T08:51:28.655+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unre
stricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

To start mongodb:

>Mongo

To check the databases present

>show dbs

```
> show dbs
<laisha>  0.000GB
admin     0.000GB
config    0.000GB
lab       0.000GB
local     0.000GB
nandy     0.000GB
test      0.000GB
vansh     0.000GB
```

Now creating the database,if its already present then it will use that database otherwise it will create new database with that name

use<laisha>

Using db command we can find the current database we are using.

To display the database name in show dbs atleast one value should be inserted..

If no values are present then that database will be deleted.

Creating a collection in the database:

db.createCollection(“names”)

To display the commands we use

Show collections

```
> use <laisha>
switched to db <laisha>
> db
<laisha>
> show collections
Details
```

Inserting one value to the collection created

```
> use <laisha>
switched to db <laisha>
> db
<laisha>
> show collections
Details
> db.Details.insert({"Name":"laisha","age":20,"gender":"F"})
WriteResult({ "nInserted" : 1 })
```

Inserting multiple values to the collection created

>db.Details.find() —--used to print the values inserted

```
> db.Details.insertMany([{"Name":"Rani","age":22,"gender":"F"},{"Name":"Mohan","age":20,"gender":"M"},{"Name":"Sita","age":21,"gender":"F"}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("624bc119d463bbbd0d2e4208"),
                ObjectId("624bc119d463bbbd0d2e4209"),
                ObjectId("624bc119d463bbbd0d2e420a")
        ]
}
> db.Details.find()
{ "_id" : ObjectId("624bbc4699ca2d0fb8ca99f7"), "Name" : "Laisha" }
{ "_id" : ObjectId("624bc02dd463bbbd0d2e4207"), "Name" : "laisha", "age" : 20, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4208"), "Name" : "Rani", "age" : 22, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4209"), "Name" : "Mohan", "age" : 20, "gender" : "M" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e420a"), "Name" : "Sita", "age" : 21, "gender" : "F" }
```

Updating a single value

>db.Details.updateone({Name:”Sita”},{$set:{“age”:23}})

```
> db.Details.find()
{ "_id" : ObjectId("624bbc4699ca2d0fb8ca99f7"), "Name" : "Laisha" }
{ "_id" : ObjectId("624bc02dd463bbbd0d2e4207"), "Name" : "laisha", "age" : 20, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4208"), "Name" : "Rani", "age" : 22, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4209"), "Name" : "Mohan", "age" : 20, "gender" : "M" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e420a"), "Name" : "Sita", "age" : 21, "gender" : "F" }
> db.Details.updateOne({Name:"Sita"},{$set:{"age":23}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.Details.find()
{ "_id" : ObjectId("624bbc4699ca2d0fb8ca99f7"), "Name" : "Laisha" }
{ "_id" : ObjectId("624bc02dd463bbbd0d2e4207"), "Name" : "laisha", "age" : 20, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4208"), "Name" : "Rani", "age" : 22, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4209"), "Name" : "Mohan", "age" : 20, "gender" : "M" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e420a"), "Name" : "Sita", "age" : 23, "gender" : "F" }
```

Sita’s age is updated from 21 to 23

Updating multiple values

>db.Details.update({Name:”Mohan”},{$set:{“age”:22,”Name”:”Mohith”}})

```
> db.Details.find()
{ "_id" : ObjectId("624bbc4699ca2d0fb8ca99f7"), "Name" : "Laisha" }
{ "_id" : ObjectId("624bc02dd463bbbd0d2e4207"), "Name" : "laisha", "age" : 20, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4208"), "Name" : "Rani", "age" : 22, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4209"), "Name" : "Mohan", "age" : 20, "gender" : "M" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e420a"), "Name" : "Sita", "age" : 23, "gender" : "F" }
> db.Details.update({Name:"Mohan"},{$set:{"age":22,"Name":"
uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:57
> db.Details.update({Name:"Mohan"},{$set:{"age":22,"Name":"Mohith"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Details.find()
{ "_id" : ObjectId("624bbc4699ca2d0fb8ca99f7"), "Name" : "Laisha" }
{ "_id" : ObjectId("624bc02dd463bbbd0d2e4207"), "Name" : "laisha", "age" : 20, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4208"), "Name" : "Rani", "age" : 22, "gender" : "F" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e4209"), "Name" : "Mohith", "age" : 22, "gender" : "M" }
{ "_id" : ObjectId("624bc119d463bbbd0d2e420a"), "Name" : "Sita", "age" : 23, "gender" : "F" }
>
```