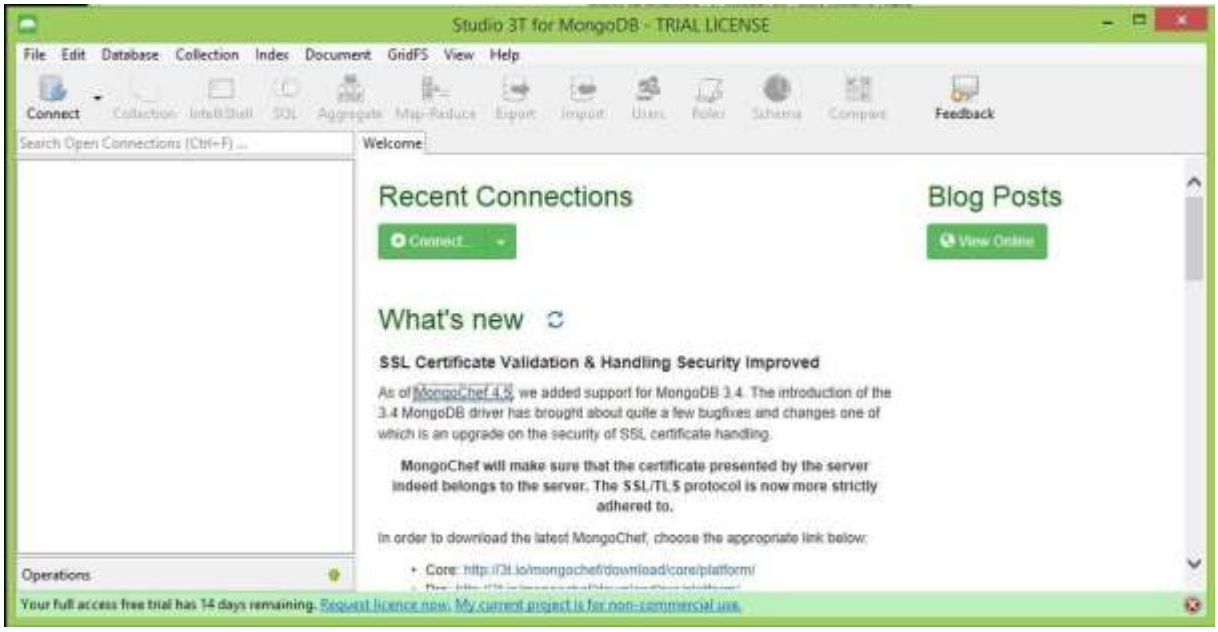
1. Installing Mongo shell.

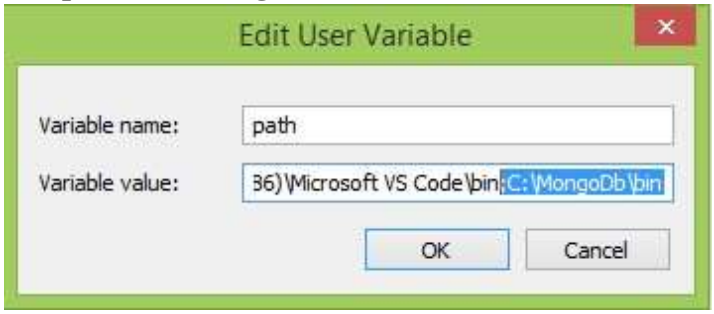
Download MongoChef by using link: https://studio3t.com/download-thank-you/?OS=win64

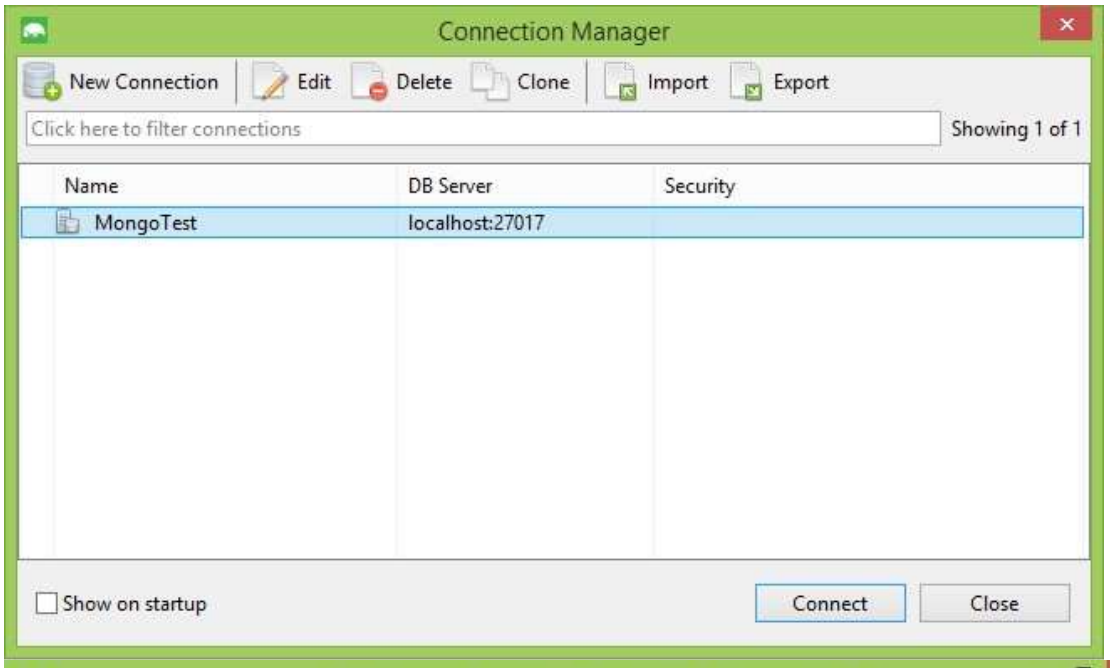
After completion of installation of MongoChef, when you launch it, it's first interface looks like as below.

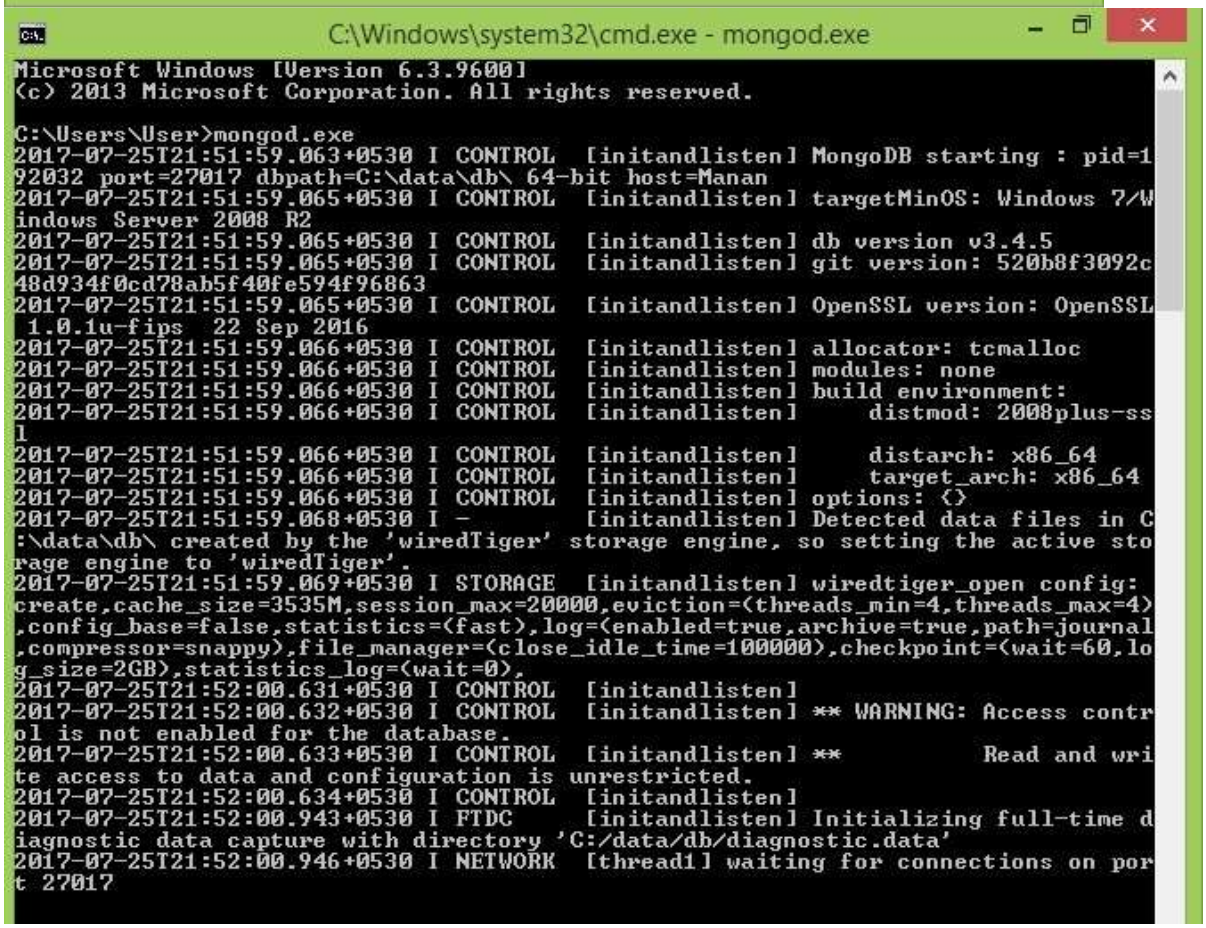


Open terminal and execute mongod.exe command, to make background daemon wait to establish connection.

Set path of MongoDB in environment variable as:



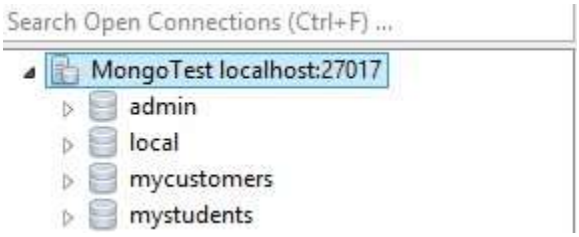

Connection Manager
New Connection
Edit
Delete
Clone
Import
Export
Click here to filter connections
Showing 1 of 1
Name
DB Server
Security
MongoTest
localhost:27017
Show on startup
Connect
Close


C:\Windows\system32\cmd.exe - mongod.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\User>mongod.exe
2017-07-25T21:51:59.063+0530 I CONTROL [initandlisten] MongoDB starting : pid=1
92032 port=27017 dbpath=C:\data\db\ 64-bit host=Manan
2017-07-25T21:51:59.065+0530 I CONTROL [initandlisten] targetMinOS: Windows 7/W
indows Server 2008 R2
2017-07-25T21:51:59.065+0530 I CONTROL [initandlisten] db version v3.4.5
2017-07-25T21:51:59.065+0530 I CONTROL [initandlisten] git version: 520b8f3092c
48d934f0cd78ab5f40fe594f96863
2017-07-25T21:51:59.065+0530 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.1u-fips 22 Sep 2016
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] allocator: tcmalloc
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] modules: none
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] build environment:
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] distmod: 2008plus-ss
l
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] distarch: x86_64
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] target_arch: x86_64
2017-07-25T21:51:59.066+0530 I CONTROL [initandlisten] options: {}
2017-07-25T21:51:59.068+0530 I - [initandlisten] Detected data files in C
:\data\db\ created by the 'wiredTiger' storage engine, so setting the active sto
rage engine to 'wiredTiger'.
2017-07-25T21:51:59.069+0530 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=3535M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal
,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,lo
g_size=2GB),statistics_log=(wait=0),
2017-07-25T21:52:00.631+0530 I CONTROL [initandlisten]
2017-07-25T21:52:00.632+0530 I CONTROL [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2017-07-25T21:52:00.633+0530 I CONTROL [initandlisten] ** Read and wri
te access to data and configuration is unrestricted.
2017-07-25T21:52:00.634+0530 I CONTROL [initandlisten]
2017-07-25T21:52:00.943+0530 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory 'C:/data/db/diagnostic.data'
2017-07-25T21:52:00.946+0530 I NETWORK [thread1] waiting for connections on por
t 27017

In MongoChef click on Connect, then click on New Connection. Give name of connection and click on save.

Select your connection and click on connect.

It will display all the collections under connection.



Click on IntelliShell. In that shell we can execute our executable statement or script.

COMMANDS

1. **CREATE USER**

```
db.createUser({  user:
        "Patel",
        pwd: "123",
       roles:[{role: "userAdminAnyDatabase" , db:"admin"}]
})
```

```
> db.createUser({  user: "Patel",
...  pwd: "123",
...  roles:[{role: "userAdminAnyDatabase" , db:"admin"}]
... })
Successfully added user: {
        "user" : "Patel",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
>
```

2. **CREATE DATABASE**
   Use emp

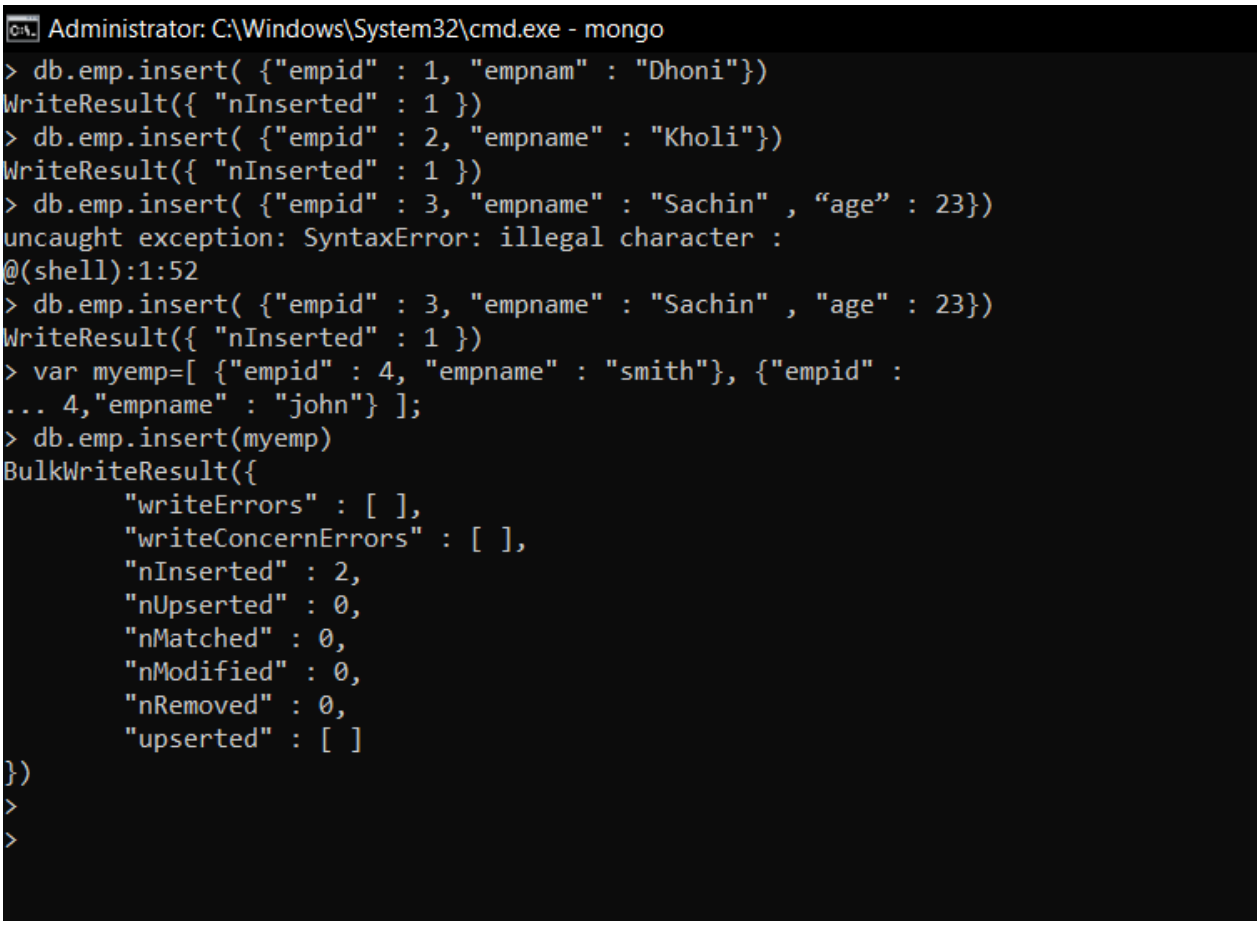
```
> use emp
switched to db emp
>
```

3. **CREATE COLLECTION**
   db.createCollection("emp")

```
> db.createCollection("emp")
{ "ok" : 1 }
>
```

4. **INSERT DATA INTO DATABASE**
   - **db.emp.insert({"empid" : 1, "empnam" : "Dhoni"})**
   - **db.emp.insert({"empid" : 2, "empname" : "Kholi"})**

- **db.emp.insert({"empid" : 3, "empname" : "Sachin" , "age" : 23})**
- **var myemp=[{"empid" : 4, "empname" : "smith"}, {"empid" : 4,"empname" : "john"}];**
  db.emp.insert(myemp)



5. UPDATE AND DELETE DATA

- **db.emp.update({"empid" :1},{$set:{"empname" : "Patel"}})**
  db.emp.update({"empid" :1},{$set:{age:50})

```
> db.emp.update({"empid" :1},{$set:{"empname" : "Patel"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.emp.update({"empid" :1},{$set:{age:50})
...
...
>
> db.emp.update({"empid" :1},{$set:{age:50}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

- **remove particular field:**
- **db.emp.update({"empid" :1},{$unset:{age:40})**
- **update value of age by 5:**
  db.emp.update({"empid" :1},{$inc:{age:5})
- **removing whole document db.emp.remove({empid:3})**
- **If no document matching with particular condition exists, then if we set upsert “true”, then initially it will also create it.**

```
> db.emp.update({"empid" :1},{$unset:{age:40}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.emp.update({"empid" :1},{$inc:{age:5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.emp.remove({empid:3})
WriteResult({ "nRemoved" : 1 })
>
```

db.emp.update({empid:3},{empid:5,empname:"Patel"},{upsert:true });

```
> db.emp.update({empid:3},{empid:5,empname:"Patel"},{upsert:true
... });
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("622418dd6d73fb91e6b5849e")
})
>
```

- **Renaming any particular field**
  db.emp.update({empid:4},{$rename:{"empname":"nameemp"}});

```
> db.emp.update({empid:4},{$rename:{"empname":"nameemp"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

6. **RETRIVAL OF DATA**
   - **db.emp.find().forEach(printjson)**

```
> db.emp.find().forEach(printjson)
{
        "_id" : ObjectId("62241696c4aba460b7903c08"),
        "empid" : 1,
        "empnam" : "Dhoni",
        "empname" : "Patel",
        "age" : 5
}
{
        "_id" : ObjectId("622416a4c4aba460b7903c09"),
        "empid" : 2,
        "empname" : "Kholi"
}
{
        "_id" : ObjectId("62241701c4aba460b7903c0b"),
        "empid" : 4,
        "nameemp" : "smith"
}
{
        "_id" : ObjectId("62241701c4aba460b7903c0c"),
        "empid" : 4,
        "empname" : "john"
}
{
        "_id" : ObjectId("622418dd6d73fb91e6b5849e"),
        "empid" : 5,
        "empname" : "Patel"
}
>
```

- **db.emp.find().pretty()**

```
> db.emp.find().pretty()
{
        "_id" : ObjectId("62241696c4aba460b7903c08"),
        "empid" : 1,
        "empnam" : "Dhoni",
        "empname" : "Patel",
        "age" : 5
}
{
        "_id" : ObjectId("622416a4c4aba460b7903c09"),
        "empid" : 2,
        "empname" : "Kholi"
}
{
        "_id" : ObjectId("62241701c4aba460b7903c0b"),
        "empid" : 4,
        "nameemp" : "smith"
}
{
        "_id" : ObjectId("62241701c4aba460b7903c0c"),
        "empid" : 4,
        "empname" : "john"
}
{
        "_id" : ObjectId("622418dd6d73fb91e6b5849e"),
        "empid" : 5,
        "empname" : "Patel"
}
> _
```

- **db.emp.find({ })**
- **db.emp.find({empname : "smith"}).forEach(printjson)**

```
> db.emp.find({nameemp : "smith"} ).forEach(printjson)
{
        "_id" : ObjectId("62241701c4aba460b7903c0b"),
        "empid" : 4,
        "nameemp" : "smith"
}
>
```

- **db.emp.find({empid : {$gt:2}})**

```
> db.emp.find({empid : {$gt:2}})
{ "_id" : ObjectId("62241701c4aba460b7903c0b"), "empid" : 4, "nameemp" : "smith" }
{ "_id" : ObjectId("62241701c4aba460b7903c0c"), "empid" : 4, "empname" : "john" }
{ "_id" : ObjectId("622418dd6d73fb91e6b5849e"), "empid" : 5, "empname" : "Patel" }
```

var myemp=db.emp.find({ empid : {$gt:2}})
while(myemp.hasNext()){ print(tojson(myemp.next()));
}

```
> var myemp=db.emp.find( { empid : {$gt:2}}) ;while(myemp.hasNext()){ print(tojson(myemp.next())); }
{
        "_id" : ObjectId("62241701c4aba460b7903c0b"),
        "empid" : 4,
        "nameemp" : "smith"
}
{
        "_id" : ObjectId("62241701c4aba460b7903c0c"),
        "empid" : 4,
        "empname" : "john"
}
{
        "_id" : ObjectId("622418dd6d73fb91e6b5849e"),
        "empid" : 5,
        "empname" : "Patel"
}
>
```

7. DISPLAY DATABASE show dbs

```
> show dbs
admin   0.000GB
config  0.000GB
emp     0.000GB
local   0.000GB
vandan  0.000GB
>
```