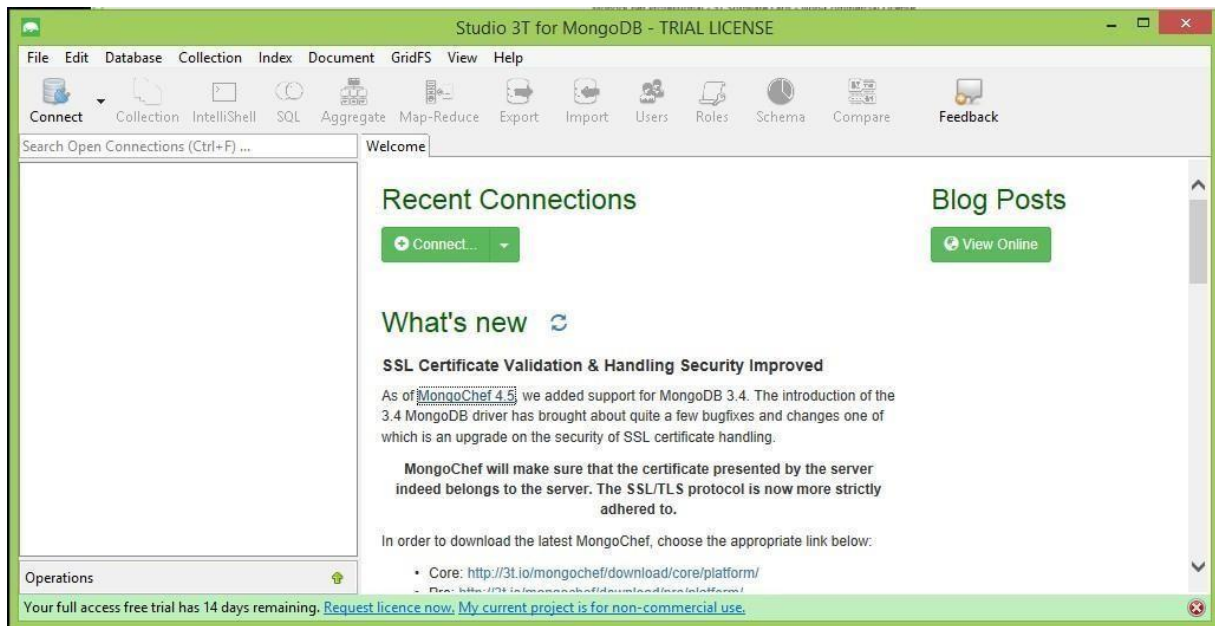


## Practical-2 NoSQL

### 1. Installing Mongo shell.

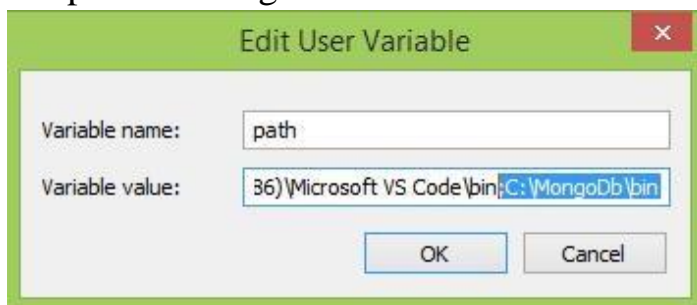
Download MongoChef by using link: <https://studio3t.com/download-thankyou/?OS=win64>

After completion of installation of MongoChef, when you launch it, it's first interfacelookslikeas below.

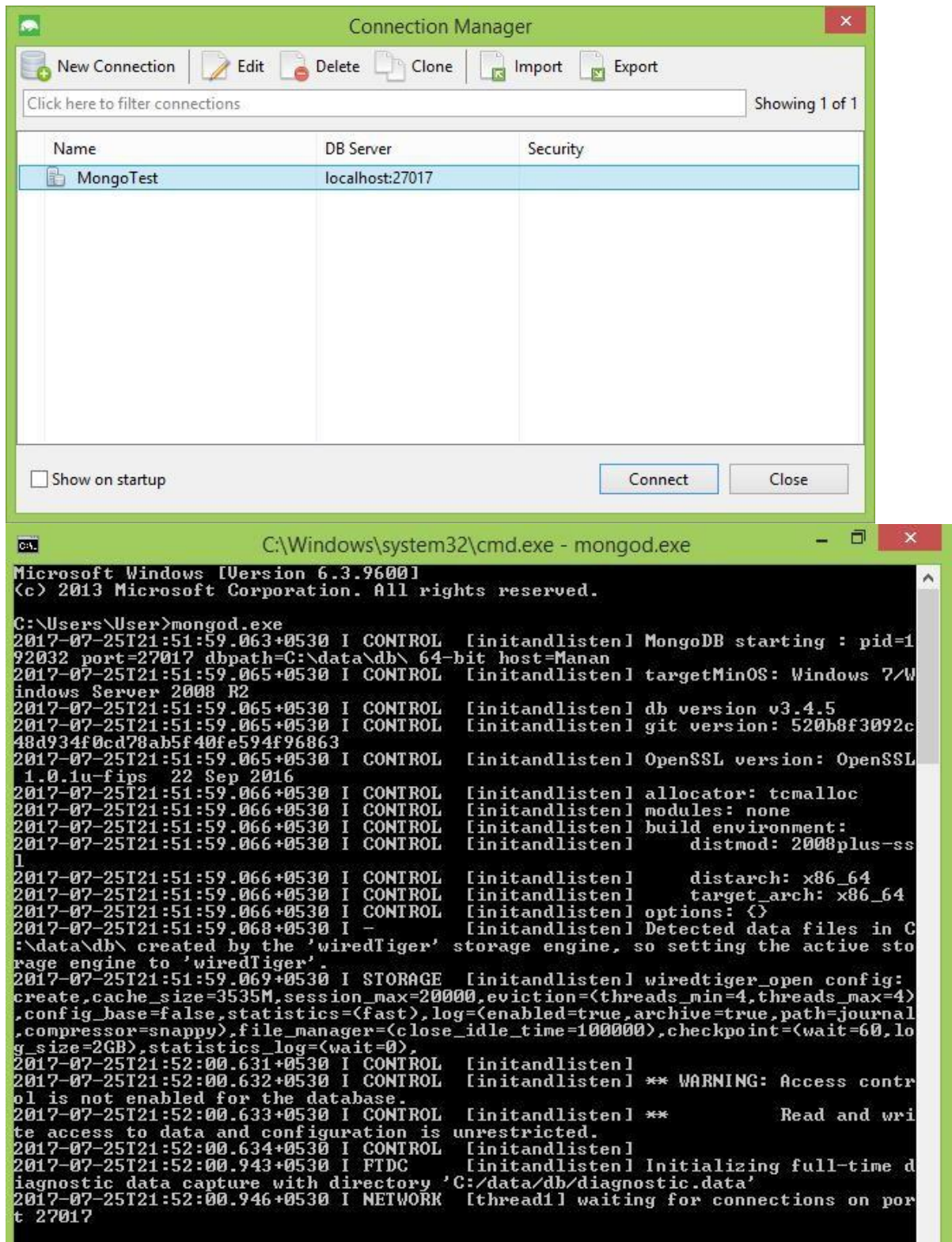


Open terminal and execute mongod.exe command, to make background daemon wait to establish connection.

Set path of MongoDB in environment variable as:



## Practical-2 NoSQL



The image shows two windows from a Windows operating system. The top window is the 'Connection Manager' application, which displays a table with one connection named 'MongoTest' connected to 'localhost:27017'. The bottom window is a command prompt running 'mongod.exe', showing the MongoDB server startup logs. The logs indicate that the server is starting on port 27017, using the default data path, and initializing various components like the storage engine, WiredTiger, and network threads.

**Connection Manager**

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
```

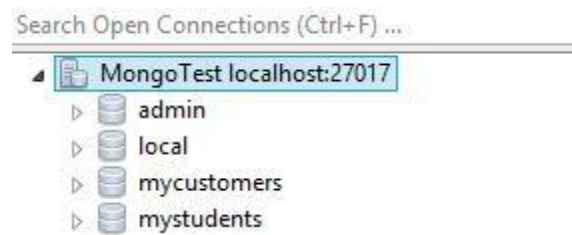
In MongoChef click on Connect, then click on New Connection. Give name of connection and click on save.

Select your connection and click on connect.

## Practical-2 NoSQL

---

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:
  "prachi", pwd: "pds",
  })
```

**DESCRIPTION :-** This is used to create user and put your username,pwd in it.

#### 2. CREATE DATABASE

Use emp

#### DESCRIPTION :-

This used to create database named 'emp'.

#### 3. CREATE COLLECTION

```
db.createCollection("emp")
```

#### DESCRIPTION :-

In this we can create collection named 'emp'.

#### 4. INSERT DATA INTO DATABASE

```
○ db.emp.insert( {"empid" : 1, "empname" : "martin"}) ○
db.emp.insert( {"empid" : 1, "empname" : "martin"}) ○
db.emp.insert( {"empid" : 2, "empname" : "neha", "age" : 23}) ○ var
myemp=[ {"empid" : 3, "empname" : "smith"}, {"empid" :
  4, "empname" : "john"}]; db.emp.insert(myemp)
```

**DESCRIPTION :-** By this we can input command in it and by var variable we can input using variable.

## 5. UPDATE AND DELETE DATA

- `db.emp.update({'empid' :1},{ $set:{'empname' : 'pds'}})` □  
`db.emp.update({'empid' :1},{ $set:{age:40}})` □ 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:3},{empid:5,empname:"heena"},{upsert:true});`
- Renaming any particular field  
`db.emp.update({empid:4},{ $rename:{'empname':'nameemp'}});`

**DESCRIPTION :-** by update command we can update anything in that data and by remove command we can remove anything from that data.

## 6. RETRIVAL OF DATA

- `db.emp.find().forEach(printjson)`
- `db.emp.find().pretty()`
- `db.emp.find( { } )`
- `db.emp.find({empname : "smith"} ).forEach(printjson)`
- `db.emp.find({empid : {$gt:2}})`
- `var myemp=db.emp.find( { empid : {$gt:2}})`  
`while(myemp.hasNext()){ print(tojson(myemp.next()));`  
`}`

**DESCRIPTION:-** By use of find command we can find anything from data.

## 7. DISPLAY DATABASE show dbs

## Practical-2 NoSQL

**DESCRIPTION :-** We use this if we want to display all database which was in our server

### EXTRA :-

create a user of your name with every database's admin.

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

create a database named uvpce

```
> use uvpce
switched to db uvpce
```

create collection named "your name"

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

insert 7 records into that collection

```
{ "_id" : ObjectId("620a20da4dd9e59a778ba9a4"), "empid" : 7, "empname" : "khatri", "empno" : 20012011120, "age" : 26, "city" : "kolkata" }
{ "_id" : ObjectId("620a214c4dd9e59a778ba9a5"), "empid" : 6, "empname" : "deepo", "empno" : 20012011122, "age" : 25, "city" : "chennai" }
{ "_id" : ObjectId("620a21664dd9e59a778ba9a6"), "empid" : 5, "empname" : "teja", "empno" : 20012011127, "age" : 24, "city" : "delhi" }
{ "_id" : ObjectId("620a21824dd9e59a778ba9a7"), "empid" : 4, "empname" : "shivam", "empno" : 20012011125, "age" : 23, "city" : "goa" }
{ "_id" : ObjectId("620a21be4dd9e59a778ba9a8"), "empid" : 3, "empname" : "saumil", "empno" : 20012011124, "age" : 22, "city" : "ahmedabad" }
{ "_id" : ObjectId("620a21dd4dd9e59a778ba9a9"), "empid" : 2, "empname" : "dhyey", "empno" : 20012011089, "age" : 21, "city" : "mumbai" }
{ "_id" : ObjectId("620a21fa4dd9e59a778ba9aa"), "empid" : 1, "empname" : "tirth", "empno" : 20012011129, "age" : 20, "city" : "surat" }
{ "_id" : ObjectId("6219ebcd43659c4521e9eb6"), "empid" : 8, "empname" : "pujan", "empno" : 20012011130, "age" : 27, "city" : "baroda" }
```

compulsory fields in every document: enno, age, city

1. display all the records having age 20.

```
> db.emp.find({"age" : 20})
{ "_id" : ObjectId("620a21fa4dd9e59a778ba9aa"), "empid" : 1, "empname" : "tirth", "empno" : 20012011129, "age" : 20, "city" : "surat" }
```

2. display city and age of enno 4.

```
> db.emp.find({"empno":20012011124},{ "city":1,"age":1})
{ "_id" : ObjectId("620a21be4dd9e59a778ba9a8"), "age" : 22, "city" : "ahmedabad" }
```



## Practical-2 NoSQL

### 3. update the age to 25 whoes enno is greater than 3.

```
> db.emp.updateMany({"empno":{"gt:20012011123}},{$set:{age:25}})
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 5 }
> db.emp.find({})
{ "_id" : ObjectId("620a20da4dd9e59a778ba9a4"), "empid" : 7, "empname" : "khatri", "empno" : 20012011120, "age" : 26, "city" : "kolkata" }
{ "_id" : ObjectId("620a214c4dd9e59a778ba9a5"), "empid" : 6, "empname" : "deepo", "empno" : 20012011122, "age" : 25, "city" : "chennai" }
{ "_id" : ObjectId("620a21664dd9e59a778ba9a6"), "empid" : 5, "empname" : "teja", "empno" : 20012011127, "age" : 25, "city" : "delhi" }
{ "_id" : ObjectId("620a21824dd9e59a778ba9a7"), "empid" : 4, "empname" : "shivam", "empno" : 20012011125, "age" : 25, "city" : "goa" }
{ "_id" : ObjectId("620a21be4dd9e59a778ba9a8"), "empid" : 3, "empname" : "saumil", "empno" : 20012011124, "age" : 25, "city" : "ahmedabad" }
{ "_id" : ObjectId("620a21dd4dd9e59a778ba9a9"), "empid" : 2, "empname" : "dhyey", "empno" : 20012011089, "age" : 21, "city" : "mumbai" }
{ "_id" : ObjectId("620a21fa4dd9e59a778ba9aa"), "empid" : 1, "empname" : "tirth", "empno" : 20012011129, "age" : 25, "city" : "surat" }
{ "_id" : ObjectId("6219ebcd4f3659c4521e9eb6"), "empid" : 8, "empname" : "pujan", "empno" : 20012011130, "age" : 25, "city" : "baroda" }
```

### 4. add a new field to enno 5.

```
> db.emp.find({"empno:20012011125"})
{ "_id" : ObjectId("620a21824dd9e59a778ba9a7"), "empid" : 4, "empname" : "shivam", "empno" : 20012011125, "age" : 25, "city" : "goa", "branch" : "C.E" }
```

### 5. delete the document for which city is baroda.

```
> db.emp.remove({"city":"baroda"})
WriteResult({ "nRemoved" : 1 })
> db.emp.find({})
{ "_id" : ObjectId("620a20da4dd9e59a778ba9a4"), "empid" : 7, "empname" : "khatri", "empno" : 20012011120, "age" : 26, "city" : "kolkata" }
{ "_id" : ObjectId("620a214c4dd9e59a778ba9a5"), "empid" : 6, "empname" : "deepo", "empno" : 20012011122, "age" : 25, "city" : "chennai" }
{ "_id" : ObjectId("620a21664dd9e59a778ba9a6"), "empid" : 5, "empname" : "teja", "empno" : 20012011127, "age" : 25, "city" : "delhi" }
{ "_id" : ObjectId("620a21824dd9e59a778ba9a7"), "empid" : 4, "empname" : "shivam", "empno" : 20012011125, "age" : 25, "city" : "goa", "branch" : "C.E" }
{ "_id" : ObjectId("620a21be4dd9e59a778ba9a8"), "empid" : 3, "empname" : "saumil", "empno" : 20012011124, "age" : 25, "city" : "ahmedabad" }
{ "_id" : ObjectId("620a21dd4dd9e59a778ba9a9"), "empid" : 2, "empname" : "dhyey", "empno" : 20012011089, "age" : 21, "city" : "mumbai" }
{ "_id" : ObjectId("620a21fa4dd9e59a778ba9aa"), "empid" : 1, "empname" : "tirth", "empno" : 20012011129, "age" : 25, "city" : "surat" }
```

### 6. rename the field enno to rollno for every document.

```
> db.emp.updateMany({},{$rename:{"empno":"rollno"}})
{ "acknowledged" : true, "matchedCount" : 7, "modifiedCount" : 7 }
> db.emp.find({})
{ "_id" : ObjectId("620a20da4dd9e59a778ba9a4"), "empid" : 7, "empname" : "khatri", "age" : 26, "city" : "kolkata", "rollno" : 20012011120 }
{ "_id" : ObjectId("620a214c4dd9e59a778ba9a5"), "empid" : 6, "empname" : "deepo", "age" : 25, "city" : "chennai", "rollno" : 20012011122 }
{ "_id" : ObjectId("620a21664dd9e59a778ba9a6"), "empid" : 5, "empname" : "teja", "age" : 25, "city" : "delhi", "rollno" : 20012011127 }
{ "_id" : ObjectId("620a21824dd9e59a778ba9a7"), "empid" : 4, "empname" : "shivam", "age" : 25, "city" : "goa", "branch" : "C.E", "rollno" : 20012011125 }
{ "_id" : ObjectId("620a21be4dd9e59a778ba9a8"), "empid" : 3, "empname" : "saumil", "age" : 25, "city" : "ahmedabad", "rollno" : 20012011124 }
{ "_id" : ObjectId("620a21dd4dd9e59a778ba9a9"), "empid" : 2, "empname" : "dhyey", "age" : 21, "city" : "mumbai", "rollno" : 20012011089 }
{ "_id" : ObjectId("620a21fa4dd9e59a778ba9aa"), "empid" : 1, "empname" : "tirth", "age" : 25, "city" : "surat", "rollno" : 20012011129 }
```