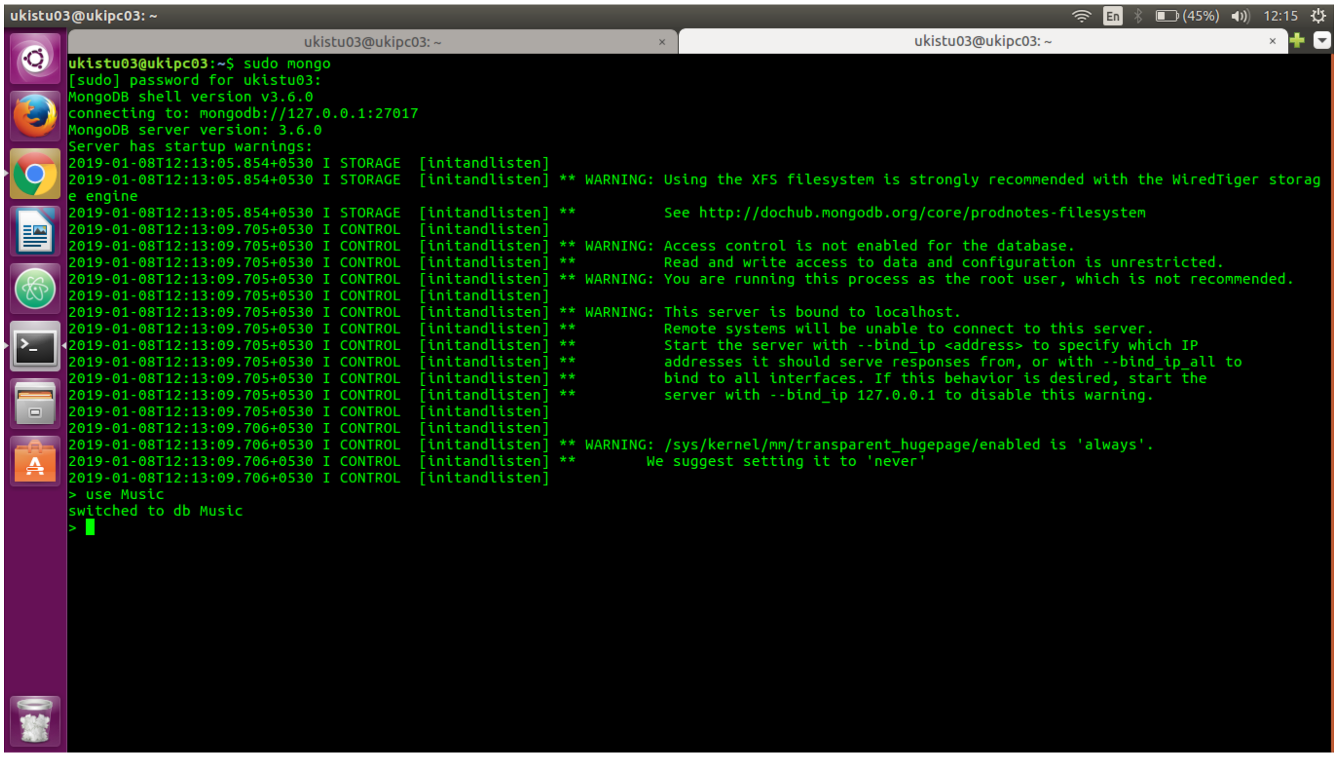
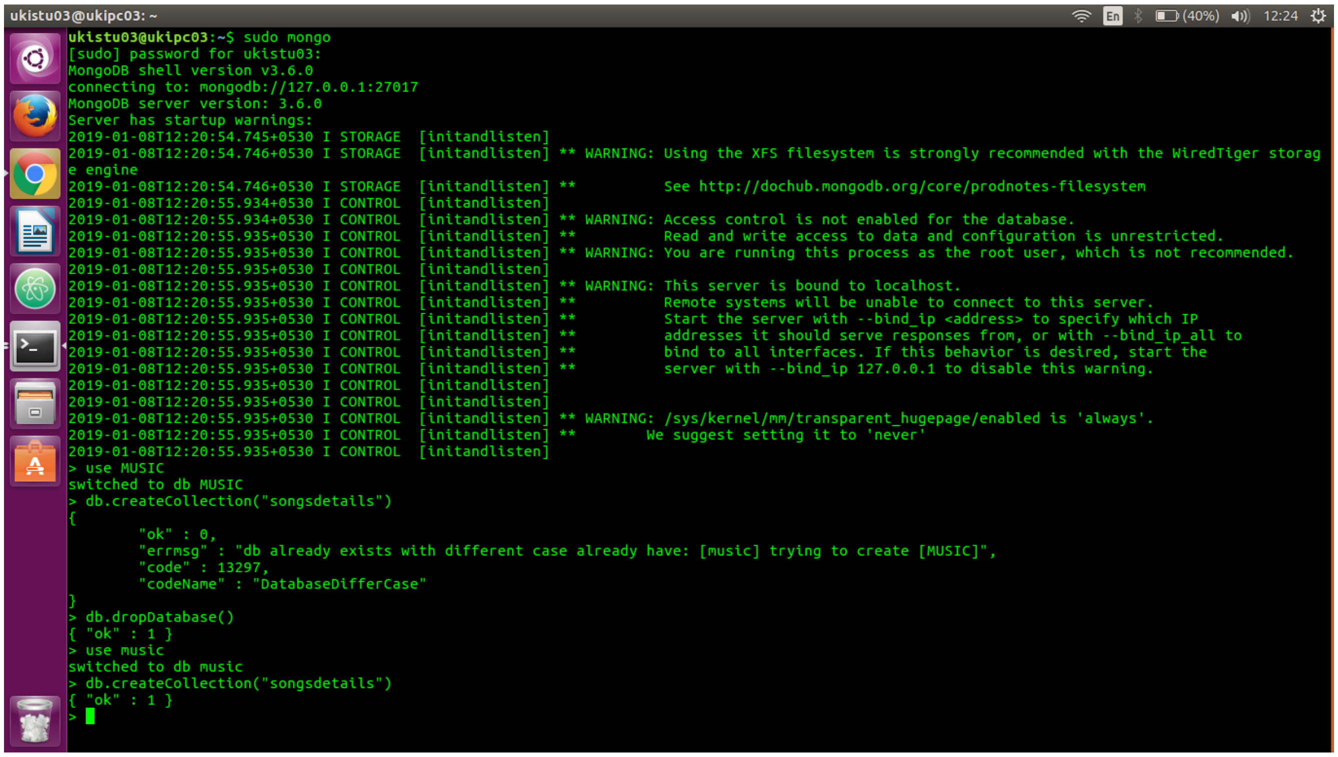
1.Create a Database called music.
   Use music



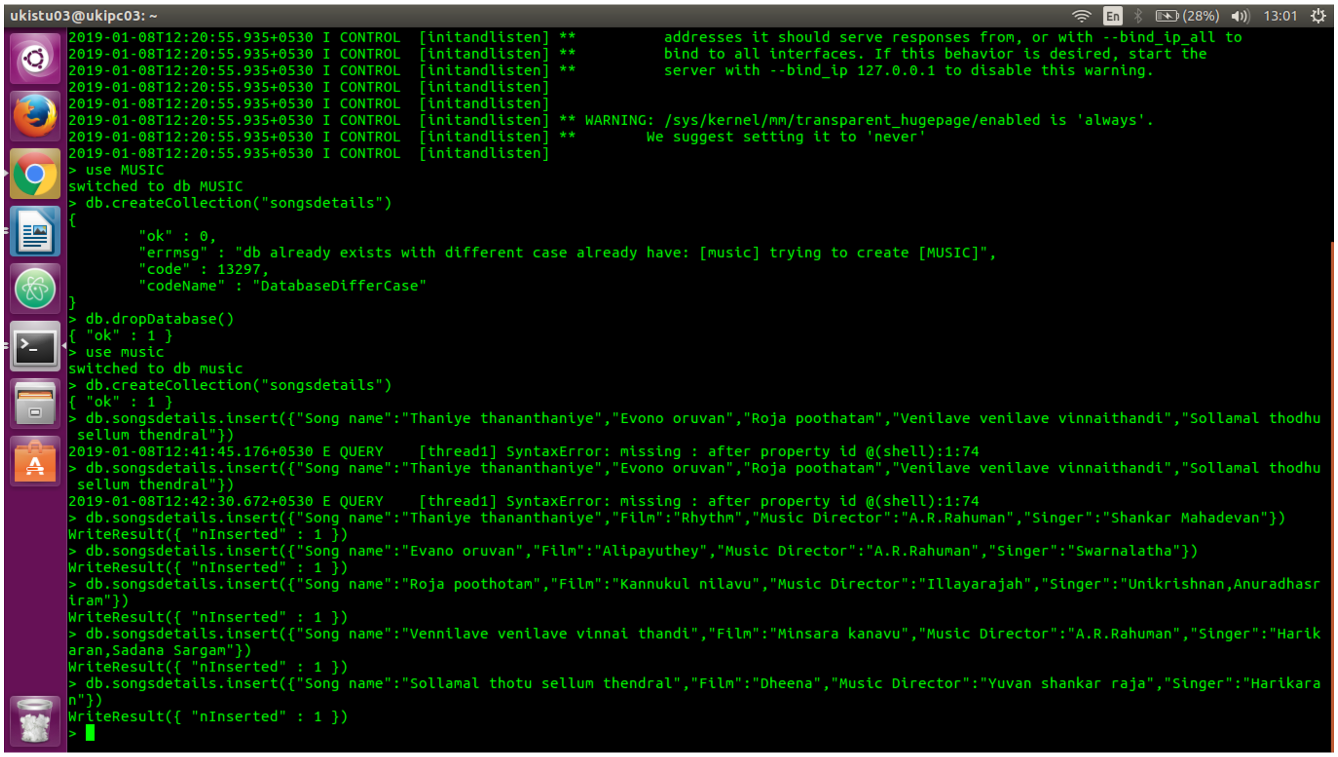
```
ukistu03@ukipc03:~$ sudo mongo
[sudo] password for ukistu03:
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2019-01-08T12:13:05.854+0530 I STORAGE  [initandlisten]
2019-01-08T12:13:05.854+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-08T12:13:05.854+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten]
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten]
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T12:13:09.705+0530 I CONTROL  [initandlisten]
2019-01-08T12:13:09.706+0530 I CONTROL  [initandlisten]
2019-01-08T12:13:09.706+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:13:09.706+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-08T12:13:09.706+0530 I CONTROL  [initandlisten]
> use Music
switched to db Music
> 
```

2.Create a collection called songdetails.
  db.createDatacollection(“songsdetails”)



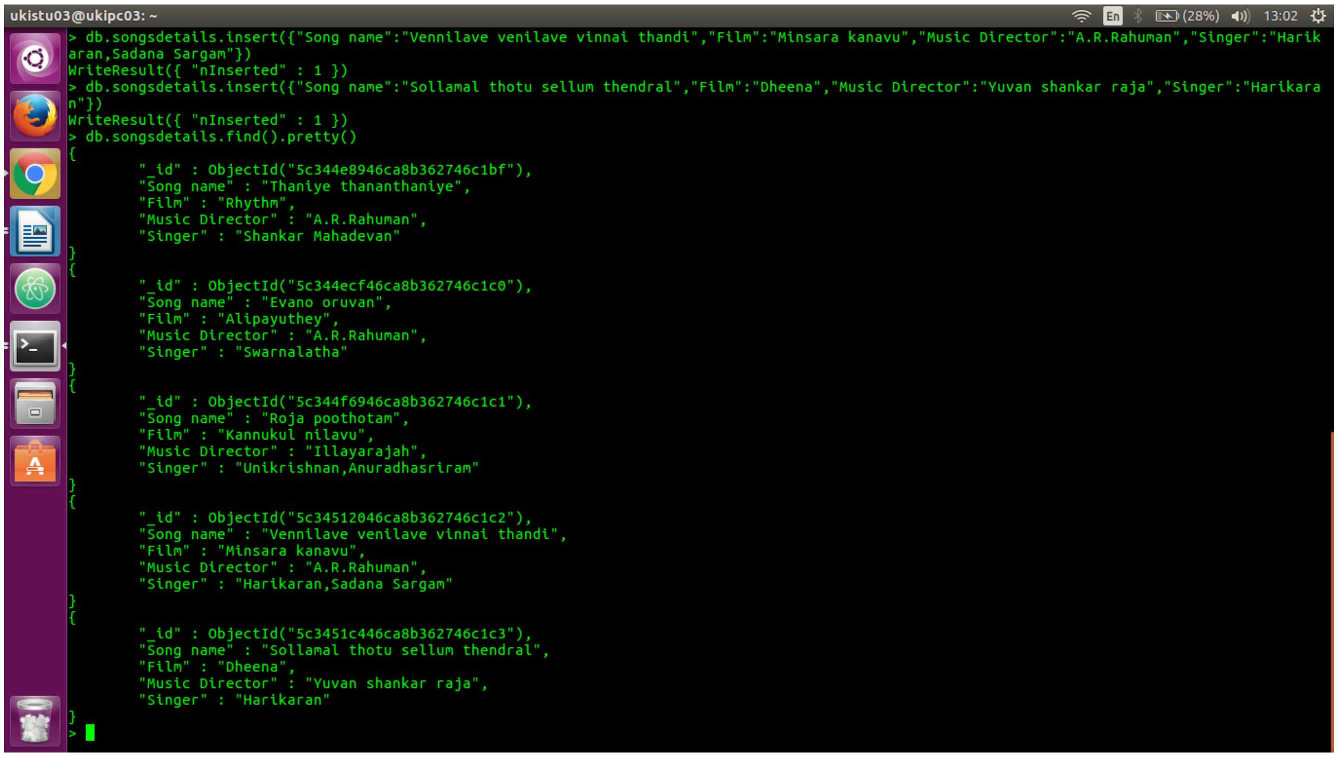
```
ukistu03@ukipc03:~$ sudo mongo
[sudo] password for ukistu03:
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2019-01-08T12:20:54.745+0530 I STORAGE  [initandlisten]
2019-01-08T12:20:54.746+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-08T12:20:54.746+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T12:20:55.934+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.934+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
> use MUSIC
switched to db MUSIC
> db.createCollection("songsdetails")
{
        "ok" : 0,
        "errmsg" : "db already exists with different case already have: [music] trying to create [MUSIC]",
        "code" : 13297,
        "codeName" : "DatabaseDifferCase"
}
> db.dropDatabase()
{ "ok" : 1 }
> use music
switched to db music
> db.createCollection("songsdetails")
{ "ok" : 1 }
> 
```

3.Create the above 5 song documents.



```
ukistu03@ukipc03: ~
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-08T12:20:55.935+0530 I CONTROL  [initandlisten]
> use MUSIC
switched to db MUSIC
> db.createCollection("songsdetails")
{
        "ok" : 0,
        "errmsg" : "db already exists with different case already have: [music] trying to create [MUSIC]",
        "code" : 13297,
        "codeName" : "DatabaseDifferCase"
}
> db.dropDatabase()
{ "ok" : 1 }
> use music
switched to db music
> db.createCollection("songsdetails")
{ "ok" : 1 }
> db.songsdetails.insert({"Song name":"Thaniye thananthaniye","Evono oruvan","Roja poothatam","Venilave venilave vinnaithandi","Sollamal thodhu
 sellum thendral"})
2019-01-08T12:41:45.176+0530 E QUERY    [thread1] SyntaxError: missing : after property id @(shell):1:74
> db.songsdetails.insert({"Song name":"Thaniye thananthaniye","Evono oruvan","Roja poothatam","Venilave venilave vinnaithandi","Sollamal thodhu
 sellum thendral"})
2019-01-08T12:42:30.672+0530 E QUERY    [thread1] SyntaxError: missing : after property id @(shell):1:74
> db.songsdetails.insert({"Song name":"Thaniye thananthaniye","Film":"Rhythm","Music Director":"A.R.Rahuman","Singer":"Shankar Mahadevan"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"Song name":"Evano oruvan","Film":"Alipayuthey","Music Director":"A.R.Rahuman","Singer":"Swarnalatha"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"Song name":"Roja poothotam","Film":"Kannukul nilavu","Music Director":"Illayarajah","Singer":"Unikrishnan,Anuradhasr
iram"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"Song name":"Vennilave venilave vinnai thandi","Film":"Minsara kanavu","Music Director":"A.R.Rahuman","Singer":"Harik
aran,Sadana Sargam"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"Song name":"Sollamal thotu sellum thendral","Film":"Dheena","Music Director":"Yuvan shankar raja","Singer":"Harikara
n"})
WriteResult({ "nInserted" : 1 })
> 
```

4.List all documents created.
db.songsdetails.find().pretty()



```
ukistu03@ukipc03: ~
> db.songsdetails.insert({"Song name":"Vennilave venilave vinnai thandi","Film":"Minsara kanavu","Music Director":"A.R.Rahuman","Singer":"Harik
aran,Sadana Sargam"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"Song name":"Sollamal thotu sellum thendral","Film":"Dheena","Music Director":"Yuvan shankar raja","Singer":"Harikara
n"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find().pretty()
{
        "_id" : ObjectId("5c344e8946ca8b362746c1bf"),
        "Song name" : "Thaniye thananthaniye",
        "Film" : "Rhythm",
        "Music Director" : "A.R.Rahuman",
        "Singer" : "Shankar Mahadevan"
}
{
        "_id" : ObjectId("5c344ecf46ca8b362746c1c0"),
        "Song name" : "Evano oruvan",
        "Film" : "Alipayuthey",
        "Music Director" : "A.R.Rahuman",
        "Singer" : "Swarnalatha"
}
{
        "_id" : ObjectId("5c344f6946ca8b362746c1c1"),
        "Song name" : "Roja poothotam",
        "Film" : "Kannukul nilavu",
        "Music Director" : "Illayarajah",
        "Singer" : "Unikrishnan,Anuradhasriram"
}
{
        "_id" : ObjectId("5c34512046ca8b362746c1c2"),
        "Song name" : "Vennilave venilave vinnai thandi",
        "Film" : "Minsara kanavu",
        "Music Director" : "A.R.Rahuman",
        "Singer" : "Harikaran,Sadana Sargam"
}
{
        "_id" : ObjectId("5c3451c446ca8b362746c1c3"),
        "Song name" : "Sollamal thotu sellum thendral",
        "Film" : "Dheena",
        "Music Director" : "Yuvan shankar raja",
        "Singer" : "Harikaran"
}
> 
```