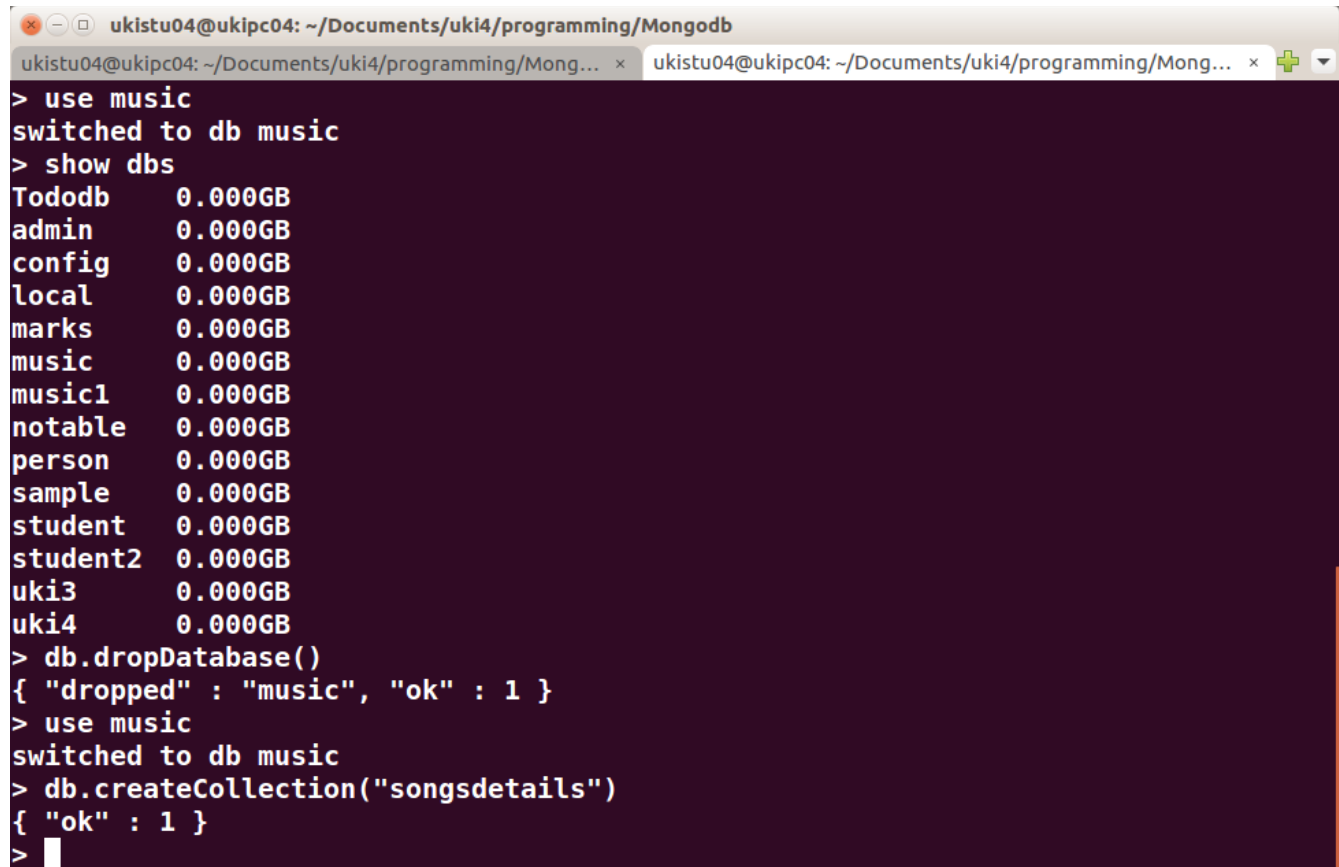


1.Create a Database called music.

use music

A terminal window with a dark purple background. The window title is 'ukistu04@ukipc04: ~/Documents/uki4/programming/Mongodb'. The terminal shows the following commands and output:

```
> use music
switched to db music
> show dbs
Tododb      0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
marks       0.000GB
music       0.000GB
music1      0.000GB
notable     0.000GB
person      0.000GB
sample      0.000GB
student     0.000GB
student2    0.000GB
uki3        0.000GB
uki4        0.000GB
> db.dropDatabase()
{ "dropped" : "music", "ok" : 1 }
> use music
switched to db music
> db.createCollection("songsdetails")
{ "ok" : 1 }
>
```

2.Create a collection called songdetails.

db.createCollection("songsdetails")

```
ukistu04@ukipc04: ~/Documents/uki4/programming/Mongodb
ukistu04@ukipc04: ~/Documents/uki4/programming/Mong... x ukistu04@ukipc04: ~/Documents/uki4/programming/Mong... x +
local      0.000GB
marks      0.000GB
music      0.000GB
music1     0.000GB
notable    0.000GB
person     0.000GB
sample     0.000GB
student    0.000GB
student2   0.000GB
uki3       0.000GB
uki4       0.000GB
> db.dropDatabase()
{ "dropped" : "music", "ok" : 1 }
> use music
switched to db music
> db.createCollection("songsdetails")
{ "ok" : 1 }
> db.songsdetails.insert({"songname":"Thaniye
2019-01-08T12:30:52.560+0530 E QUERY [thread1] SyntaxError: unterminated stri
ng literal @(shell):1:35
> db.songsdetails.insert({"song name":"Thaniye Thananthaniye", "Film":"Rhythm","
Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"})
WriteResult({ "nInserted" : 1 })
>
```

3. Create the above 5 song documents.

```
db.songsdetails.insert({"song name":"Thaniye Thananthaniye", "Film":"Rhythm","Music
Director":"A.R.Rahman","Singer":"Shankar mahadevan"})
db.songsdetails.insert({"song name":"Evano oruvan", "Film":"Alai payuthey","Music
Director":"A.R.Rahman","Singer":"Swarnalatha"})
db.songsdetails.insert({"song name":"Roja poonthotham", "Film":"Kannukul nilavu","Music
Director":"Ilaiyaraaja","Singer":"Unnikrishnan,Anuradha sriram"})
db.songsdetails.insert({"song name":"Vennilavae vennilavae vinnai thandi",
"Film":"Minsarakanavu","Music Director":"A.R.Rahman","Singer":"Hariharan,SahanaSargam"})
db.songsdetails.insert({"song name":"Sollamal thottu chellum thendral", "Film":"Dheena","Music
Director":"Yuvan shankar raja","Singer":"Hariharan"})
```

```

ukistu04@ukipc04: ~/Documents/uki4/programming/Mongodb
sample 0.000GB
student 0.000GB
student2 0.000GB
uki3 0.000GB
uki4 0.000GB
> db.dropDatabase()
{ "dropped" : "music", "ok" : 1 }
> use music
switched to db music
> db.createCollection("songsdetails")
{ "ok" : 1 }
> db.songsdetails.insert({"songname":"Thaniye
2019-01-08T12:30:52.560+0530 E QUERY [thread1] SyntaxError: unterminated string literal @(shell):1:35
> db.songsdetails.insert({"song name":"Thaniye Thananthaniye", "Film":"Rhythm","Music Director":"A.R.Rahman",
,"Singer":"Shankar mahadevan"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Evano oruvan", "Film":"Alai payuthey","Music Director":"A.R.Rahman",
"Singer":"Swarnalatha"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Roja poonthotham", "Film":"Kannukul nilavu","Music Director":"Ilaiya
raaja","Singer":"Unnikrishnan,Anuradha sriram"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Vennilavae vennilavae vinnai thandi", "Film":"Minsarakanavu","Music
Director":"A.R.Rahman","Singer":"Hariharan,SahanaSargam"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Sollamal thottu chellum thendral", "Film":"Dheena","Music Director":
"Yuvan shankar raja","Singer":"Hariharan"})
WriteResult({ "nInserted" : 1 })
> █

```

4.List all documents created.

db.songsdetails.find()

```

ukistu04@ukipc04: ~/Documents/uki4/programming/Mongodb
"Singer":"Swarnalatha"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Roja poonthotham", "Film":"Kannukul nilavu","Music Director":"Ilaiya
raaja","Singer":"Unnikrishnan,Anuradha sriram"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Vennilavae vennilavae vinnai thandi", "Film":"Minsarakanavu","Music
Director":"A.R.Rahman","Singer":"Hariharan,SahanaSargam"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"song name":"Sollamal thottu chellum thendral", "Film":"Dheena","Music Director":
"Yuvan shankar raja","Singer":"Hariharan"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find.pretty()
2019-01-08T12:54:14.154+0530 E QUERY [thread1] TypeError: db.songsdetails.find.pretty is not a function
:
@(shell):1:1
> show collections
songsdetails
> db.songsdetails.find()
{ "_id" : ObjectId("5c344c7a70b9978d43c75a7b"), "song name" : "Thaniye Thananthaniye", "Film" : "Rhythm", "
Music Director" : "A.R.Rahman", "Singer" : "Shankar mahadevan" }
{ "_id" : ObjectId("5c344d0070b9978d43c75a7c"), "song name" : "Evano oruvan", "Film" : "Alai payuthey", "Mu
sic Director" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5c344df970b9978d43c75a7d"), "song name" : "Roja poonthotham", "Film" : "Kannukul nilavu",
"Music Director" : "Ilaiyaraaja", "Singer" : "Unnikrishnan,Anuradha sriram" }
{ "_id" : ObjectId("5c344ec270b9978d43c75a7e"), "song name" : "Vennilavae vennilavae vinnai thandi", "Film"
: "Minsarakanavu", "Music Director" : "A.R.Rahman", "Singer" : "Hariharan,SahanaSargam" }
{ "_id" : ObjectId("5c344f3f70b9978d43c75a7f"), "song name" : "Sollamal thottu chellum thendral", "Film" :
"Dheena", "Music Director" : "Yuvan shankar raja", "Singer" : "Hariharan" }
> █

```

5.List A.R.Rahman's songs.

```
db.songsdetails.find({"Music Director":"A.R.Rahman"}).pretty()
```

```
ukistu04@ukipc04: ~
"Singer" : "Hariharan,SahanaSargam" }
> db.songsdetails.find({"Music Director":"A.R.Rahman"}).pretty()
{
  "_id" : ObjectId("5c344c7a70b9978d43c75a7b"),
  "song name" : "Thaniye Thananthaniye",
  "Film" : "Rhythm",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Shankar mahadevan"
}
{
  "_id" : ObjectId("5c344d0070b9978d43c75a7c"),
  "song name" : "Evano oruvan",
  "Film" : "Alai payuthey",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Swarnalatha"
}
{
  "_id" : ObjectId("5c344ec270b9978d43c75a7e"),
  "song name" : "Vennilavae vennilavae vinnai thandi",
  "Film" : "Minsarakanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan,SahanaSargam"
}
>
```

6.List A.R.Rahman's songs sung by Unnikrishnan.

```
db.songsdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikrishnan"},"song name":1})
```

```

ukistu04@ukipc04: ~
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find({}, {"Singer":1})
{ "_id" : ObjectId("5c344c7a70b9978d43c75a7b"), "Singer" : "Shankar mahadevan" }
{ "_id" : ObjectId("5c344df970b9978d43c75a7d"), "Singer" : "Unnikrishnan,Anuradha sriram" }
{ "_id" : ObjectId("5c344ec270b9978d43c75a7e"), "Singer" : "Hariharan,SahanaSargam" }
{ "_id" : ObjectId("5c344f3f70b9978d43c75a7f"), "Singer" : "Hariharan" }
{ "_id" : ObjectId("5c383e63b41a1be67e18c231") }
> db.songsdetails.find({"Singer": "Hariharan", "Film": "Minsarakanavu"})
> db.songsdetails.insert({"song name":"Roja poonthotham", "Film":"Kannukul nilavu", "Music Director":"Ilaiyaraaja", "Singer":["Unnikrishnan","Anuradha sriram"]})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find({"Music Director":"Ilaiyaraaja", "Singer":"Unnikrishnan"}, {"songname":1})
{ "_id" : ObjectId("5c38437bb41a1be67e18c232") }
> db.songsdetails.find({"Music Director":"Ilaiyaraaja", "Singer":"Unnikrishnan"}, {"songname":1})
{ "_id" : ObjectId("5c38437bb41a1be67e18c232") }
> db.songsdetails.find({"Music Director":"Ilaiyaraaja", "Singer":"Unnikrishnan"}, {"song name":1})
{ "_id" : ObjectId("5c38437bb41a1be67e18c232"), "song name" : "Roja poonthotham" }
>

```

7.Delete the song which you don't like.

```
db.songsdetails.remove({"song name":"Evano oruvan"})
```

```

ukistu04@ukipc04: ~
    "Music Director" : "A.R.Rahman",
    "Singer" : "Shankar mahadevan"
}
{
    "_id" : ObjectId("5c344d0070b9978d43c75a7c"),
    "song name" : "Evano oruvan",
    "Film" : "Alai payuthey",
    "Music Director" : "A.R.Rahman",
    "Singer" : "Swarnalatha"
}
{
    "_id" : ObjectId("5c344ec270b9978d43c75a7e"),
    "song name" : "Vennilavae vennilavae vinnai thandi",
    "Film" : "Minsarakanavu",
    "Music Director" : "A.R.Rahman",
    "Singer" : "Hariharan,SahanaSargam"
}
> db.songsdetails.find({"Music Director":"A.R.Rahman","Singer":
pretty()
> db.songsdetails.find({"Music Director":"Ilaiyaraaja","Singer":
.pretty()
> db.songsdetails.remove({"song name":"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> █

```

8.Add new song which is your favourite.

```
db.songsdetails.insert({"song name":"neeya pesiyathu"})
```

```

ukistu04@ukipc04: ~
}
{
  "_id" : ObjectId("5c344d0070b9978d43c75a7c"),
  "song name" : "Evano oruvan",
  "Film" : "Alai payuthey",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Swarnalatha"
}
{
  "_id" : ObjectId("5c344ec270b9978d43c75a7e"),
  "song name" : "Vennilavae vennilavae vinnai thandi",
  "Film" : "Minsarakanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan,SahanaSargam"
}
> db.songsdetails.find({"Music Director":"A.R.Rahman","Singer":
pretty()
> db.songsdetails.find({"Music Director":"Ilaiyaraaja","Singer":
.pretty()
> db.songsdetails.remove({"song name":"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> db.songsdetails.insert({"song name":"neeya pesiyathu"})
WriteResult({ "nInserted" : 1 })
>

```

9.List Songs sung by Hariharan from Minsara kanavu film.

```
db.songsdetails.find({"Singer":"Hariharan","film":"Minsara kanavu"},{"songs name":1})
```



```

ukistu04@ukipc04: ~
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] ** WARNING: This server
is bound to localhost.
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          Remote syste
ms will be unable to connect to this server.
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          Start the se
rver with --bind_ip <address> to specify which IP
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          addresses it
should serve responses from, or with --bind_ip_all to
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          bind to all
interfaces. If this behavior is desired, start the
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          server with
--bind_ip 127.0.0.1 to disable this warning.
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten]
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten]
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/
mm/transparent_hugepage/enabled is 'always'.
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten] **          We suggest set
ting it to 'never'
2019-01-14T08:55:56.057+0530 I CONTROL [initandlisten]
> db.songsdetails.find({"Singer":"Hariharan","film":"Minsara kanavu"},{"songs na
me":1})
2019-01-14T08:56:15.986+0530 E QUERY [thread1] SyntaxError: illegal character
@(shell):1:22
>

```

10. List out the singers' names in your document.  
 db.songsdetails.find({}, {"Singer":1})



ukistu04@ukipc04: ~

```
{
  "_id" : ObjectId("5c344ec270b9978d43c75a7e"),
  "song name" : "Vennilavae vennilavae vinnai thandi",
  "Film" : "Minsarakanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan,SahanaSargam"
}
> db.songsdetails.find({"Music Director":"A.R.Rahman","Singer":"Unnikrishnan"}).
pretty()
> db.songsdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikrishnan"})
.pretty()
> db.songsdetails.remove({"song name":"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> db.songsdetails.insert({"song name":"neeya pesiyathu"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find({},{"Singer":1})
{ "_id" : ObjectId("5c344c7a70b9978d43c75a7b"), "Singer" : "Shankar mahadevan" }
{ "_id" : ObjectId("5c344df970b9978d43c75a7d"), "Singer" : "Unnikrishnan,Anuradh
a sriram" }
{ "_id" : ObjectId("5c344ec270b9978d43c75a7e"), "Singer" : "Hariharan,SahanaSarg
am" }
{ "_id" : ObjectId("5c344f3f70b9978d43c75a7f"), "Singer" : "Hariharan" }
{ "_id" : ObjectId("5c383e63b41a1be67e18c231") }
>
```