

1. Create a Database called music.
Use music

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
ukistu03@ukipc03:~$ sudo mongo  
[sudo] password for ukistu03:  
MongoDB shell version v3.6.0  
connecting to: mongod://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] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage 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] ** 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] ** 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 addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning.  
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'  
> use Music  
switched to db Music  
>
```

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

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03:~$ sudo mongo  
[sudo] password for ukistu03:  
MongoDB shell version v3.6.0  
connecting to: mongod://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] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage 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] ** WARNING: Access control is not enabled for the database.  
2019-01-08T12:20:55.934+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] ** 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 addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning.  
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'  
> use MUSIC  
switched to db MUSIC  
> db.createCollection("songdetails")  
{  
  "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("songdetails")  
{ "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 vinnathandi", "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 vinnathandi", "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 poothatam", "Film": "Kannukul nilavu", "Music Director": "Illayarajah", "Singer": "Unikrishnan, Anuradhasriram"})  
WriteResult({"nInserted" : 1 })  
> db.songsdetails.insert({"Song name": "Vennilave venilave vinnai thandi", "Film": "Minsara kanavu", "Music Director": "A.R.Rahuman", "Singer": "Harikaran, Sadana Sargam"})  
WriteResult({"nInserted" : 1 })  
> db.songsdetails.insert({"Song name": "Sollamal thodu sellum thendral", "Film": "Dheena", "Music Director": "Yuvan shankar raja", "Singer": "Harikaran"})  
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": "Harikaran, Sadana Sargam"})  
WriteResult({"nInserted" : 1 })  
> db.songsdetails.insert({"Song name": "Sollamal thodu sellum thendral", "Film": "Dheena", "Music Director": "Yuvan shankar raja", "Singer": "Harikaran"})  
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 poothatam",  
  "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 thodu sellum thendral",  
  "Film" : "Dheena",  
  "Music Director" : "Yuvan shankar raja",  
  "Singer" : "Harikaran"  
}  
>
```

5. List the A.R. Rahman songs

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
{"_id": ObjectId("5c344e8946ca8b362746c1bf"),  
  "Song name": "Thaniye thananthaniye",  
  "Film": "Rhythm",  
  "Music Director": "A.R. Rahman",  
  "Singer": "Shankar Mahadevan"}  
{  
  "id": ObjectId("5c344ecf46ca8b362746c1c0"),  
  "Song name": "Evano oruvan",  
  "Film": "Alipayuthey",  
  "Music Director": "A.R. Rahman",  
  "Singer": "Swarnalatha"}  
{  
  "id": ObjectId("5c344f6946ca8b362746c1c1"),  
  "Song name": "Raja poothotam",  
  "Film": "Kannukul nilavu",  
  "Music Director": "Ilaiyaraajah",  
  "Singer": "Unikrishnan, Anuradhasriram"}  
{  
  "id": ObjectId("5c34512046ca8b362746c1c2"),  
  "Song name": "Vennilave venilave vinnai thandi",  
  "Film": "Minsara kanavu",  
  "Music Director": "A.R. Rahman",  
  "Singer": "Harikaran, Sadana Sargam"}  
{  
  "id": ObjectId("5c3451c446ca8b362746c1c3"),  
  "Song name": "Sollamal thodu sellum thendral",  
  "Film": "Dheena",  
  "Music Director": "Yuvan shankar raja",  
  "Singer": "Harikaran"}  
> db.songsdetails.find({"Music Director": "A.R. Rahman"})  
{ "_id": ObjectId("5c344e8946ca8b362746c1bf"), "Song name": "Thaniye thananthaniye", "Film": "Rhythm", "Music Director": "A.R. Rahman", "Singer": "Shankar Mahadevan" }  
{ "_id": ObjectId("5c344ecf46ca8b362746c1c0"), "Song name": "Evano oruvan", "Film": "Alipayuthey", "Music Director": "A.R. Rahman", "Singer": "Swarnalatha" }  
{ "_id": ObjectId("5c34512046ca8b362746c1c2"), "Song name": "Vennilave venilave vinnai thandi", "Film": "Minsara kanavu", "Music Director": "A.R. Rahman", "Singer": "Harikaran, Sadana Sargam" }  
>
```

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

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
{  
  "id": ObjectId("5c34512046ca8b362746c1c2"),  
  "Song name": "Vennilave venilave vinnai thandi",  
  "Film": "Minsara kanavu",  
  "Music Director": "A.R. Rahman",  
  "Singer": "Harikaran, Sadana Sargam"}  
{  
  "id": ObjectId("5c3451c446ca8b362746c1c3"),  
  "Song name": "Sollamal thodu sellum thendral",  
  "Film": "Dheena",  
  "Music Director": "Yuvan shankar raja",  
  "Singer": "Harikaran"}  
{  
  "id": ObjectId("5c383f0c2c9d6aa4112fe0a7"),  
  "Song name": "Kadavul thantha alagiya valgai"}  
{  
  "id": ObjectId("5c38413d2c9d6aa4112fe0a8"),  
  "Song name": "Raja poothotam"}  
{  
  "id": ObjectId("5c3841ea2c9d6aa4112fe0a9"),  
  "Song name": "Raja poothotam",  
  "Film": "Kannukul nilavu",  
  "Music Director": "Ilaiyaraajah",  
  "Singer": "Unikrishnan, Anuradhasriram"}  
{  
  "id": ObjectId("5c3842202c9d6aa4112fe0aa"),  
  "Song name": "Raja poothotam",  
  "Film": "Kannukul nilavu",  
  "Music Director": "Ilaiyaraajah",  
  "Singer": "Unikrishnan, Anuradhasriram"}  
> db.songsdetails.insert({"Song name": "Raja poothotam", "Film": "Kannukul nilavu", "Music Director": "Ilaiyaraajah", "Singer": ["Unikrishnan", "Anuradhasriram"]})  
WriteResult({ "nInserted": 1 })  
> db.songsdetails.find({"Music Director": "Ilaiyaraajah", "Singer": "Unikrishnan"}, {"Song name": 1})  
{ "_id": ObjectId("5c3843182c9d6aa4112fe0ab"), "Song name": "Raja poothotam" }  
>
```

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

```
ukistu03@ukipc03: ~
switched to db music
> show dbs
admin      0.000GB
exercise1  0.000GB
local      0.000GB
markes     0.000GB
movies     0.000GB
music      0.000GB
mydb       0.000GB
person     0.000GB
student    0.000GB
test       0.000GB
uk13       0.000GB
uk14       0.000GB
> db.songsdetails.find()
{ "_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" : "Illyarajah", "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" }
> db.songsdetails.find({"Music Director":"Illyarajah","Singer":"Unikrishnan"})
> db.songsdetails.find({"Music Director":"Illyarajah","Singer":"Unikrishnan"},{"Music Director":1,"Singer":1})
> db.songsdetails.find({"Singer"},{"Singer":1})
2019-01-11T12:14:20.271+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30
> db.songsdetails.find({},{"Singer":1})
{ "_id" : ObjectId("5c344e8946ca8b362746c1bf"), "Singer" : "Shankar Mahadevan" }
{ "_id" : ObjectId("5c344ecf46ca8b362746c1c0"), "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5c344f6946ca8b362746c1c1"), "Singer" : "Unikrishnan,Anuradhasriram" }
{ "_id" : ObjectId("5c34512046ca8b362746c1c2"), "Singer" : "Harikaran,Sadana Sargam" }
{ "_id" : ObjectId("5c3451c446ca8b362746c1c3"), "Singer" : "Harikaran" }
> db.songsdetails.remove("Song name" : "Roja poothotam",1)
2019-01-11T12:27:27.207+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:35
> db.songsdetails.find({"Singer"},{"Singer":1})
2019-01-11T12:27:57.023+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30
> db.songsdetails.remove({"Song name" : "Roja poothotam"})
WriteResult({ "nRemoved" : 1 })
>
```

8) Add new song which is your favourite.

```
ukistu03@ukipc03: ~
admin      0.000GB
exercise1  0.000GB
local      0.000GB
markes     0.000GB
movies     0.000GB
music      0.000GB
mydb       0.000GB
person     0.000GB
student    0.000GB
test       0.000GB
uk13       0.000GB
uk14       0.000GB
> db.songsdetails.find()
{ "_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" : "Illyarajah", "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" }
> db.songsdetails.find({"Music Director":"Illyarajah","Singer":"Unikrishnan"})
> db.songsdetails.find({"Music Director":"Illyarajah","Singer":"Unikrishnan"},{"Music Director":1,"Singer":1})
> db.songsdetails.find({"Singer"},{"Singer":1})
2019-01-11T12:14:20.271+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30
> db.songsdetails.find({},{"Singer":1})
{ "_id" : ObjectId("5c344e8946ca8b362746c1bf"), "Singer" : "Shankar Mahadevan" }
{ "_id" : ObjectId("5c344ecf46ca8b362746c1c0"), "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5c344f6946ca8b362746c1c1"), "Singer" : "Unikrishnan,Anuradhasriram" }
{ "_id" : ObjectId("5c34512046ca8b362746c1c2"), "Singer" : "Harikaran,Sadana Sargam" }
{ "_id" : ObjectId("5c3451c446ca8b362746c1c3"), "Singer" : "Harikaran" }
> db.songsdetails.remove("Song name" : "Roja poothotam",1)
2019-01-11T12:27:27.207+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:35
> db.songsdetails.find({"Singer"},{"Singer":1})
2019-01-11T12:27:57.023+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30
> db.songsdetails.remove({"Song name" : "Roja poothotam"})
WriteResult({ "nRemoved" : 1 })
> db.songsdetails.insert({"Song name" : "Kadavul thantha alagiya valgal"})
WriteResult({ "nInserted" : 1 })
>
```

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

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
"Mus  
"Sng  
"Unikrishnan",  
"Anuradhasriram"  
}  
{  
  "id" : ObjectId("5c3847862c9d6aa4112fe0ac"),  
  "Song name" : "Vennilave venilave vinnai thandi",  
  "Film" : "Minsara kanavu",  
  "Music Director" : "A.R.Rahuman",  
  "Singer" : [  
    "Harikaran,Sadana Sargam"  
  ]  
}  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"}{"Song name":1})  
2019-01-11T13:17:26.571+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:67  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"}{"Song name":1})  
2019-01-11T13:17:39.606+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:67  
> db.songsdetails.remove("Film":"Minsara kanavu")  
2019-01-11T13:20:22.642+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:29  
> db.songsdetails.remove({"Film":"Minsara kanavu"})  
WriteResult({ "nRemoved" : 1 })  
> db.songsdetails.insert({"Song name":"Vennilave venilave vinnai thandi","Film":"Minsara kanavu","Music Director":"A.R.Rahuman","Singer":["Harikaran","Sadana Sargam"]})  
...  
> db.songsdetails.insert({"Song name":"Vennilave venilave vinnai thandi","Film":"Minsara kanavu","Music Director":"A.R.Rahuman","Singer":["Harikaran,Sadana Sargam"]})  
WriteResult({ "nInserted" : 1 })  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"}{"Song name":1})  
2019-01-11T13:23:35.136+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:67  
> db.songsdetails.insert({"Song name":"Vennilave venilave vinnai thandi","Film":"Minsara kanavu","Music Director":"A.R.Rahuman","Singer":["Harikaran","Sadana Sargam"]})  
WriteResult({ "nInserted" : 1 })  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"}{"Song name":1})  
2019-01-11T13:27:41.373+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:67  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"}{"Song name":1})  
2019-01-11T13:59:32.952+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:67  
> db.songsdetails.find({"Film":"Minsara kanavu","Singer":"Harikaran"},{"Song name":1})  
{ "id" : ObjectId("5c384c6ecd303699cbac8fed"), "Song name" : "Vennilave venilave vinnai thandi" }  
>
```

10)List out the singers' names in your document.

```
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
ukistu03@ukipc03: ~  
2019-01-11T11:33:49.532+0530 I CONTROL [initandlisten]  
2019-01-11T11:33:49.532+0530 I CONTROL [initandlisten]  
2019-01-11T11:33:49.532+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2019-01-11T11:33:49.532+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2019-01-11T11:33:49.532+0530 I CONTROL [initandlisten]  
> use music  
switched to db music  
> show dbs  
admin 0.000GB  
exercise1 0.000GB  
local 0.000GB  
markes 0.000GB  
movies 0.000GB  
music 0.000GB  
mydb 0.000GB  
person 0.000GB  
student 0.000GB  
test 0.000GB  
uki3 0.000GB  
uki4 0.000GB  
> db.songsdetails.find()  
{ "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" }  
> db.songsdetails.find({"Music Director":"Illayarajah","Singer":"Unikrishnan"})  
> db.songsdetails.find({"Music Director":"Illayarajah","Singer":"Unikrishnan"},{"Music Director":1,"Singer":1})  
> db.songsdetails.find({"Singer"},{"Singer":1})  
2019-01-11T12:14:20.271+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30  
> db.songsdetails.find({},{"Singer":1})  
{ "id" : ObjectId("5c344e8946ca8b362746c1bf"), "Singer" : "Shankar Mahadevan" }  
{ "id" : ObjectId("5c344ecf46ca8b362746c1c0"), "Singer" : "Swarnalatha" }  
{ "id" : ObjectId("5c344f6946ca8b362746c1c1"), "Singer" : "Unikrishnan,Anuradhasriram" }  
{ "id" : ObjectId("5c34512046ca8b362746c1c2"), "Singer" : "Harikaran,Sadana Sargam" }  
{ "id" : ObjectId("5c3451c446ca8b362746c1c3"), "Singer" : "Harikaran" }  
>
```