

Create a Database called music

```
ukistu15@ukipc15: ~  
ukistu15@ukipc15:~$ sudo mongo  
[sudo] password for ukistu15:  
MongoDB shell version v3.6.5  
connecting to: mongod://127.0.0.1:27017  
MongoDB server version: 3.6.5  
Server has startup warnings:  
2018-07-04T20:12:25.729+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2018-07-04T20:12:25.729+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.  
2018-07-04T20:12:28.961+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.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] **  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] **  
> use music  
switched to db music  
>
```

Create a collection called songdetails.

```
ukistu15@ukipc15: ~  
ukistu15@ukipc15:~$ sudo mongo  
MongoDB shell version v3.6.5  
connecting to: mongod://127.0.0.1:27017  
MongoDB server version: 3.6.5  
Server has startup warnings:  
2018-07-04T20:12:25.729+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2018-07-04T20:12:25.729+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.  
2018-07-04T20:12:28.960+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.  
2018-07-04T20:12:28.961+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.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] **  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2018-07-04T20:12:28.961+0530 I CONTROL [initandlisten] **  
> show dbs  
Music 0.000GB  
admin 0.000GB  
config 0.000GB  
local 0.000GB  
uki3 0.000GB  
> use Music  
switched to db Music  
> db.createCollection("Songdetails")  
{  
  "ok" : 0,  
  "errmsg" : "a collection 'Music.Songdetails' already exists",  
  "code" : 48,  
  "codeName" : "NamespaceExists"  
}  
> db.createCollection("Songdetails")  
{  
  "ok" : 0,  
  "errmsg" : "a collection 'Music.Songdetails' already exists",  
  "code" : 48,  
  "codeName" : "NamespaceExists"  
}  
> show dbs  
Music 0.000GB  
admin 0.000GB  
config 0.000GB  
local 0.000GB  
uki3 0.000GB  
> use Music  
switched to db Music  
> db.createCollection("Songdetails")  
{  
  "ok" : 0,  
  "errmsg" : "a collection 'Music.Songdetails' already exists",  
  "code" : 48,  
  "codeName" : "NamespaceExists"  
}  
> db.createCollection("Song details")  
{  
  "ok" : 1  
}
```

Create the above 5 song documents.

```
ukistu15@ukipc15: ~  
Server has startup warnings:  
2018-07-05T10:01:13.139+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2018-07-05T10:01:13.139+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.  
2018-07-05T10:01:15.931+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.  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten]  
  
> show dbs  
Music 0.000GB  
admin 0.000GB  
config 0.000GB  
local 0.000GB  
uk13 0.000GB  
  
> db.Songdetails.insert([{"Song Name":"Thaiye Thananthanite","Filin":"Rhytham","Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"}, {"Song Name":"Evano Otuvan","Filin":"Alai Payuthe","Music Director":"A.R.Rahman","Singer":"Swarnalatha"}, {"Song Name":"Raja Poonthottam","Filin":"K annukkul Nilavu","Music Director":"iliyaraaja","Singer":"Unnikrishnan,Anuradha Sargam"}, {"Song Name":"Vennilave Vennilave vinnithaandi","Filin":"Minsara Kanavu","Music Director":"A.R.Rahman","Singer":"Hariharan"}])  
BulkWriteResult({  
  "writeErrors" : [ ],  
  "writeConcernErrors" : [ ],  
  "nInserted" : 4,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,  
  "nRemoved" : 0,  
  "upserted" : [ ]  
})  
>
```

List all documents created

```
ukistu15@ukipc15: ~  
Server has startup warnings:  
2018-07-05T10:01:13.139+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2018-07-05T10:01:13.139+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.  
2018-07-05T10:01:15.931+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.  
2018-07-05T10:01:15.931+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.  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2018-07-05T10:01:15.932+0530 I CONTROL [initandlisten]  
  
> show dbs  
Music 0.000GB  
admin 0.000GB  
config 0.000GB  
local 0.000GB  
uk13 0.000GB  
  
> db.Songdetails.insert([{"Song Name":"Thaiye Thananthanite","Filin":"Rhytham","Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"}, {"Song Name":"Evano Otuvan","Filin":"Alai Payuthe","Music Director":"A.R.Rahman","Singer":"Swarnalatha"}, {"Song Name":"Raja Poonthottam","Filin":"K annukkul Nilavu","Music Director":"iliyaraaja","Singer":"Unnikrishnan,Anuradha Sargam"}, {"Song Name":"Vennilave Vennilave vinnithaandi","Filin":"Minsara Kanavu","Music Director":"A.R.Rahman","Singer":"Hariharan"}])  
BulkWriteResult({  
  "writeErrors" : [ ],  
  "writeConcernErrors" : [ ],  
  "nInserted" : 4,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,  
  "nRemoved" : 0,  
  "upserted" : [ ]  
})  
>
```



```
ukistu15@ukipc15: ~  
annukkul Nilavu","Music Director":"iliyaraaja","Singer":"Unnikrishnan,Anuradha Sargam"),{"Song Name":"Vennilave Vennilave vinnithaandi","Filim"  
:"Minsara Kanavu","Music Director":"A.R.Rahman","Singer":"Hariharan"]}]  
BulkWriteResult({  
  "writeErrors" : [ ],  
  "writeConcernErrors" : [ ],  
  "nInserted" : 4,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,  
  "nRemoved" : 0,  
  "upserted" : [ ]  
})  
> db.Songdetails.find().pretty()  
{  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c575"),  
  "Song Name" : "Thaiye Thananthanite",  
  "Filim" : "Rhytham",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Shankar mahadevan"  
}  
{  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c576"),  
  "Song Name" : "Evano Otuvan",  
  "Filim" : "Alai Payuthe",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Swarnalatha"  
}  
{  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c577"),  
  "Song Name" : "Roja Poonthottam",  
  "Filim" : "Kannukkul Nilavu",  
  "Music Director" : "iliyaraaja",  
  "Singer" : "Unnikrishnan,Anuradha Sargam"  
}  
{  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c578"),  
  "Song Name" : "Vennilave Vennilave vinnithaandi",  
  "Filim" : "Minsara Kanavu",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Hariharan"  
}  
>
```

List A.R.Rahman's songs

```
b.Songdetails.find({"Music Director":"A.R.Rahman"}).pretty()  
  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c575"),  
  "Song Name" : "Thaiye Thananthanite",  
  "Filim" : "Rhytham",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Shankar mahadevan"  
  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c576"),  
  "Song Name" : "Evano Otuvan",  
  "Filim" : "Alai Payuthe",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Swarnalatha"  
  
  "_id" : ObjectId("5b3dacdab82fc6ebb588c578"),  
  "Song Name" : "Vennilave Vennilave vinnithaandi",  
  "Filim" : "Minsara Kanavu",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Hariharan"
```

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

```
> db.Songdetails.find({"Music Director":"A.R.Rahman","Singer":"Unnikrishnan" })
```

Delete the song which you don't like

```
db.Songdetails.remove({"Song Name":"Vennilave Vennilave vinnaitaandi"})
writeResult({ "nRemoved" : 0 })
db.Songdetails.remove({"Song Name":"Vennilave Vennilave vinnithaandi"})
writeResult({ "nRemoved" : 1 })
```

Add new song which is your favourite

```
db.Songdetails.update({"Song Name":"Evano Otuvan"},{$set{"Kadhal kankadduthe"}})
018-07-05T11:57:42.523+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell
db.Songdetails.update({"Song Name":"Evano Otuvan"},{$set{"Kadhal kankadduthe"}})
018-07-05T11:58:09.856+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell
db.Songdetails.update({"Song Name":"Evano Otuvan"},{$set{"Song Name":"Kadhal kankadduthe"}})
018-07-05T11:59:10.959+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell
db.Songdetails.update({"Filim":"Alai Payuthe"},{$set{"Song Name":"Kadhal kankadduthe"}})
018-07-05T12:00:28.595+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell
db.Songdetails.update({"Song Name":"Evano Otuvan"},{$set{"Song Name":"Kadhal kankadduthe"}})
writeResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
db.Songdetails.find()
  "_id" : ObjectId("5b3dacdab82fc6ebb588c575"), "Song Name" : "Thaiye Thananthanite", "Filim" : "
ger" : "Shankar mahadevan" }
  "_id" : ObjectId("5b3dacdab82fc6ebb588c576"), "Song Name" : "Kadhal kankadduthe", "Filim" : "Al
Singer" : "Swarnalatha" }
  "_id" : ObjectId("5b3dacdab82fc6ebb588c577"), "Song Name" : "Roja Poonthottam", "Filim" : "Kann
"Singer" : "Unnikrishnan,Anuradha Sargam" }
db.Songdetails.find().pretty
unction () {
  this._prettyShell = true;
  return this;
```

List out the singers' names in your document

```
db.Songdetails.find({},{"Singer":1,_id:0}).sort({"Singer":-1})
"Singer" : "Unnikrishnan,Anuradha Sargam" }
"Singer" : "Swarnalatha" }
"Singer" : "Shankar mahadevan" }
k
```