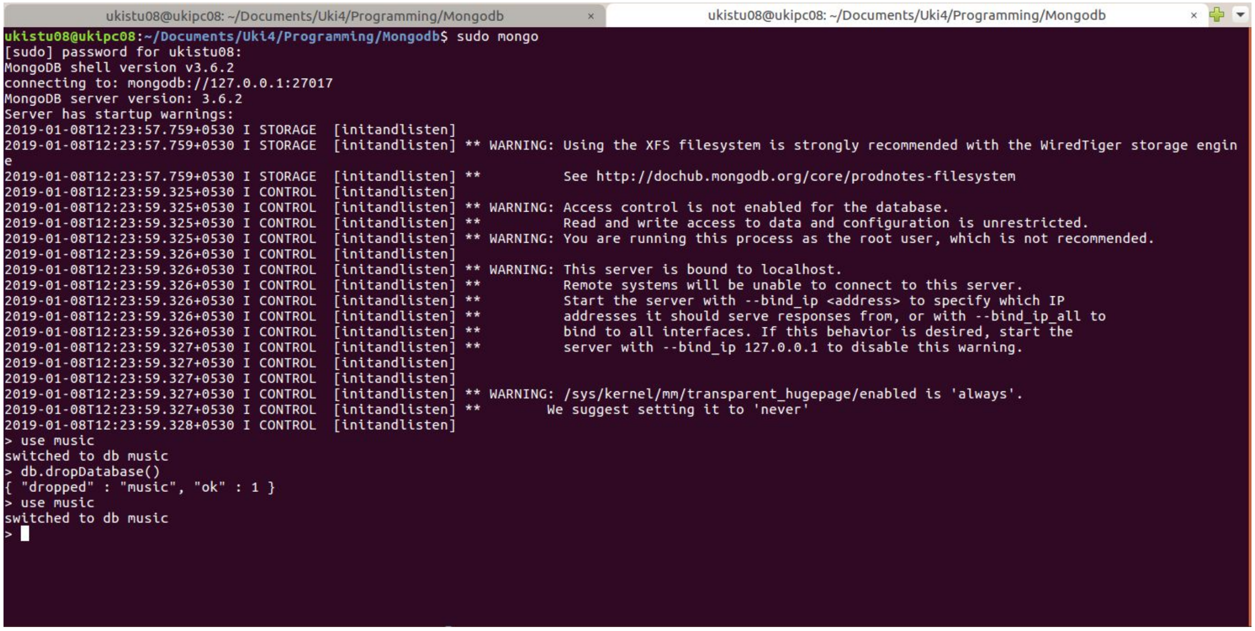
MongoDB Exercise 1

1) Create a Database called music.

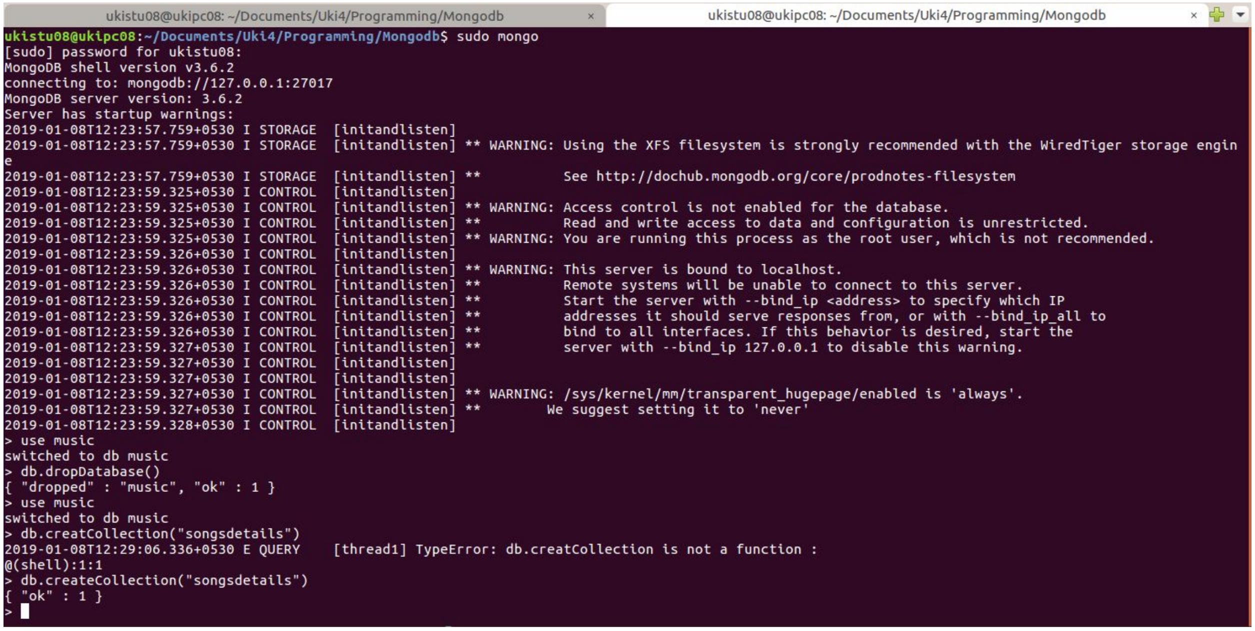
Use music


ukistu08@ukipc08: ~/Documents/Uki4/Programming/Mongodb
ukistu08@ukipc08: ~/Documents/Uki4/Programming/Mongodb

```
ukistu08@ukipc08:~/Documents/Uki4/Programming/Mongodb$ sudo mongo
[sudo] password for ukistu08:
MongoDB shell version v3.6.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.2
Server has startup warnings:
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten]
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engin
e
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-08T12:23:59.328+0530 I CONTROL  [initandlisten]
> use music
switched to db music
> db.dropDatabase()
{ "dropped" : "music", "ok" : 1 }
> use music
switched to db music
> 
```

2) Create a collection called songdetails.

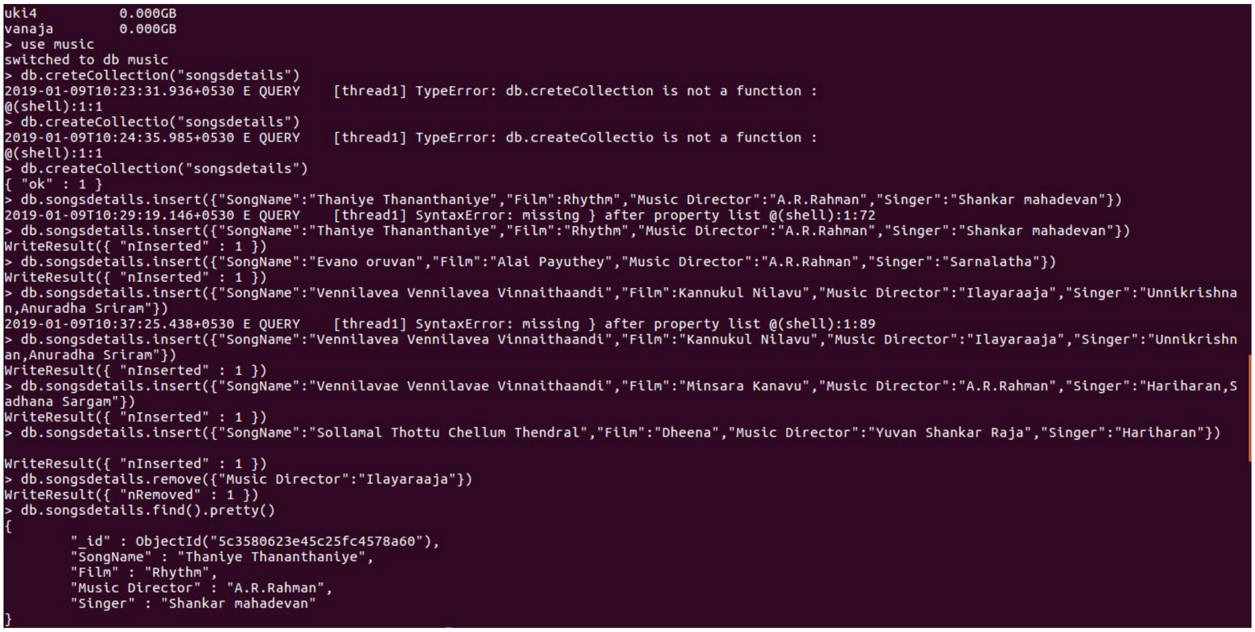
db.createDatabase(“songsdetails”)


ukistu08@ukipc08: ~/Documents/Uki4/Programming/Mongodb
ukistu08@ukipc08: ~/Documents/Uki4/Programming/Mongodb

```
ukistu08@ukipc08:~/Documents/Uki4/Programming/Mongodb$ sudo mongo
[sudo] password for ukistu08:
MongoDB shell version v3.6.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.2
Server has startup warnings:
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten]
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engin
e
2019-01-08T12:23:57.759+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-08T12:23:59.325+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-08T12:23:59.326+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten]
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:23:59.327+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-08T12:23:59.328+0530 I CONTROL  [initandlisten]
> use music
switched to db music
> db.dropDatabase()
{ "dropped" : "music", "ok" : 1 }
> use music
switched to db music
> db.creatCollection("songsdetails")
2019-01-08T12:29:06.336+0530 E QUERY    [thread1] TypeError: db.creatCollection is not a function :
@(shell):1:1
> db.createCollection("songsdetails")
{ "ok" : 1 }
> 
```

3) Create the above 5 song documents.

```
> db.songsdetails.insert({"SongName":"Thaniye Thananthaniye","Film":"Rhythm","Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"})
> db.songsdetails.insert({"SongName":"Evano oruvan","Film":"Alai Payuthey","Music Director":"A.R.Rahman","Singer":"Sarnalatha"})
> db.songsdetails.insert({"SongName":"Vennilavae Vennilavae Vinnaithaandi","Film":"Minsara Kanavu","Music Director":"A.R.Rahman","Singer":"Hariharan,Sadhana Sargam"})
> db.songsdetails.insert({"SongName":"Sollamal Thottu Chellum Thendral","Film":"Dheena","Music Director":"Yuvan Shankar Raja","Singer":"Hariharan"})
> db.songsdetails.insert({"SongName":"Roja Poonthottam","Film":"Kannukul Nilavu","Music Director":"Ilayaraaja","Singer":"Unnikrishnan,Anuradha Sriram"})
```

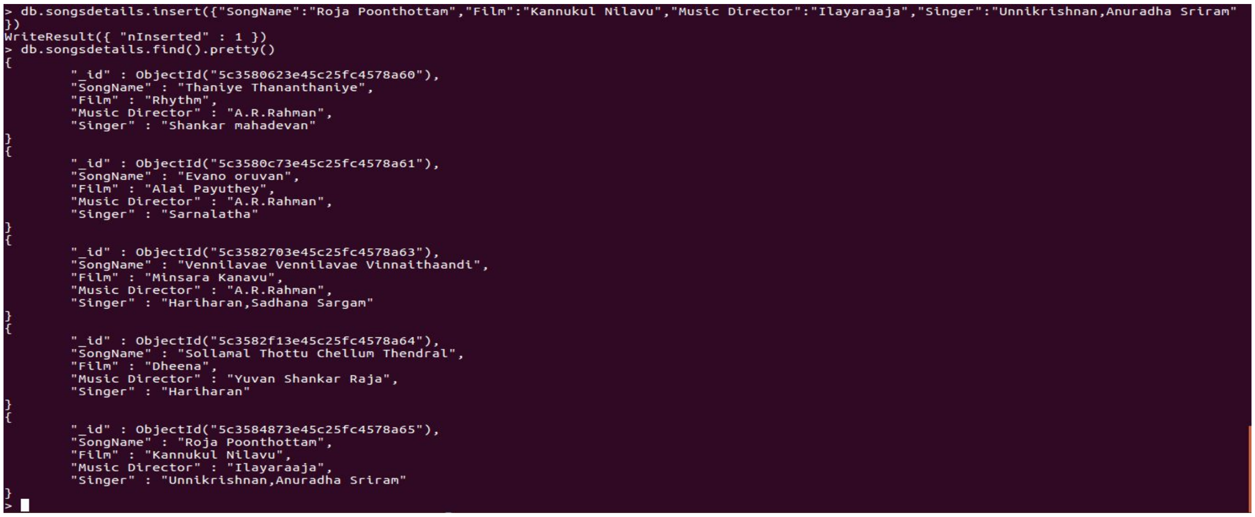


```
uki4          0.000GB
vanaja        0.000GB
> use music
switched to db music
> db.creteCollection("songsdetails")
2019-01-09T10:23:31.936+0530 E QUERY    [thread1] TypeError: db.creteCollection is not a function :
@(shell):1:1
> db.createCollectio("songsdetails")
2019-01-09T10:24:35.985+0530 E QUERY    [thread1] TypeError: db.createCollectio is not a function :
@(shell):1:1
> db.createCollection("songsdetails")
{ "ok" : 1 }
> db.songsdetails.insert({"SongName":"Thaniye Thananthaniye","Film":Rhythm","Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"})
2019-01-09T10:29:19.146+0530 E QUERY    [thread1] SyntaxError: missing } after property list @(shell):1:72
> db.songsdetails.insert({"SongName":"Thaniye Thananthaniye","Film":"Rhythm","Music Director":"A.R.Rahman","Singer":"Shankar mahadevan"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"SongName":"Evano oruvan","Film":"Alai Payuthey","Music Director":"A.R.Rahman","Singer":"Sarnalatha"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"SongName":"Vennilavea Vennilavea Vinnaithaandi","Film":Kannukul Nilavu","Music Director":"Ilayaraaja","Singer":"Unnikrishna
n,Anuradha Sriram"})
2019-01-09T10:37:25.438+0530 E QUERY    [thread1] SyntaxError: missing } after property list @(shell):1:89
> db.songsdetails.insert({"SongName":"Vennilavea Vennilavea Vinnaithaandi","Film":"Kannukul Nilavu","Music Director":"Ilayaraaja","Singer":"Unnikrishn
an,Anuradha Sriram"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"SongName":"Vennilavae Vennilavae Vinnaithaandi","Film":"Minsara Kanavu","Music Director":"A.R.Rahman","Singer":"Hariharan,S
adhana Sargam"})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.insert({"SongName":"Sollamal Thottu Chellum Thendral","Film":"Dheena","Music Director":"Yuvan Shankar Raja","Singer":"Hariharan"})

WriteResult({ "nInserted" : 1 })
> db.songsdetails.remove({"Music Director":"Ilayaraaja"})
WriteResult({ "nRemoved" : 1 })
> db.songsdetails.find().pretty()
{
        "_id" : ObjectId("5c3580623e45c25fc4578a60"),
        "SongName" : "Thaniye Thananthaniye",
        "Film" : "Rhythm",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Shankar mahadevan"
}
```

4) List all documents created.

> db.songsdetails.find().pretty()



```
> db.songsdetails.insert({"SongName":"Roja Poonthottam","Film":"Kannukul Nilavu","Music Director":"Ilayaraaja","Singer":"Unnikrishnan,Anuradha Sriram"
})
WriteResult({ "nInserted" : 1 })
> db.songsdetails.find().pretty()
{
        "_id" : ObjectId("5c3580623e45c25fc4578a60"),
        "SongName" : "Thaniye Thananthaniye",
        "Film" : "Rhythm",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Shankar mahadevan"
}
{
        "_id" : ObjectId("5c3580c73e45c25fc4578a61"),
        "SongName" : "Evano oruvan",
        "Film" : "Alai Payuthey",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Sarnalatha"
}
{
        "_id" : ObjectId("5c3582703e45c25fc4578a63"),
        "SongName" : "Vennilavae Vennilavae Vinnaithaandi",
        "Film" : "Minsara Kanavu",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Hariharan,Sadhana Sargam"
}
{
        "_id" : ObjectId("5c3582f13e45c25fc4578a64"),
        "SongName" : "Sollamal Thottu Chellum Thendral",
        "Film" : "Dheena",
        "Music Director" : "Yuvan Shankar Raja",
        "Singer" : "Hariharan"
}
{
        "_id" : ObjectId("5c3584873e45c25fc4578a65"),
        "SongName" : "Roja Poonthottam",
        "Film" : "Kannukul Nilavu",
        "Music Director" : "Ilayaraaja",
        "Singer" : "Unnikrishnan,Anuradha Sriram"
}
> 
```

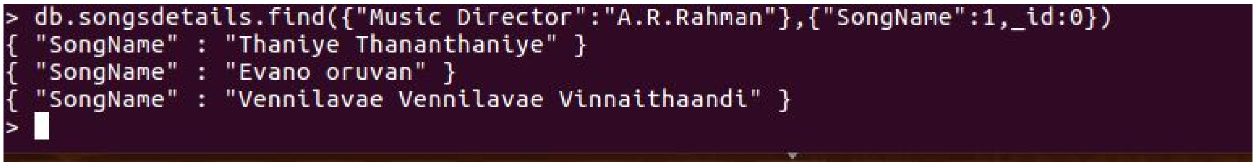
5) List A.R.Rahman’s songs.

db.songsdetails.find({"Music Director":"A.R.Rahman"},{"SongName":1,_id:0})

{ "SongName" : "Thaniye Thananthaniye" }

{ "SongName" : "Evano oruvan" }

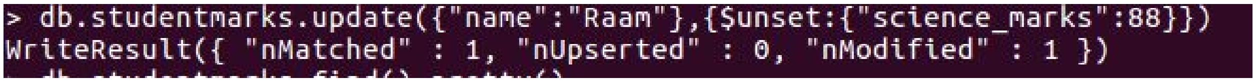
{ "SongName" : "Vennilavae Vennilavae Vinnaithaandi" }



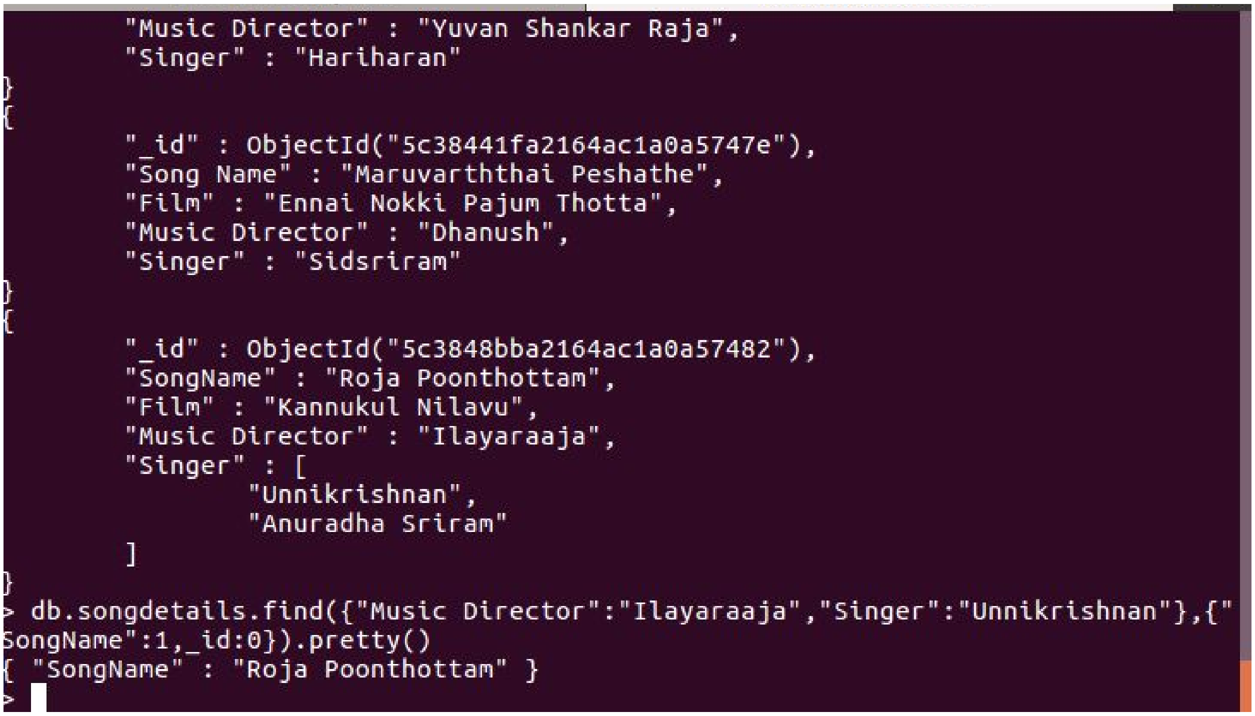
```
> db.songsdetails.find({"Music Director":"A.R.Rahman"},{"SongName":1,_id:0})
{ "SongName" : "Thaniye Thananthaniye" }
{ "SongName" : "Evano oruvan" }
{ "SongName" : "Vennilavae Vennilavae Vinnaithaandi" }
> 
```

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

db.songsdetails.insert({"SongName":"Roja Poonthottam","Film":"Kannukul Nilavu","Music Director":"Ilayaraaja","Singer":["Unnikrishnan","Anuradha Sriram"]})



```
> db.studentmarks.update({"name":"Raam"},{$unset:{"science_marks":88}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.studentmarks.find().pretty()
```

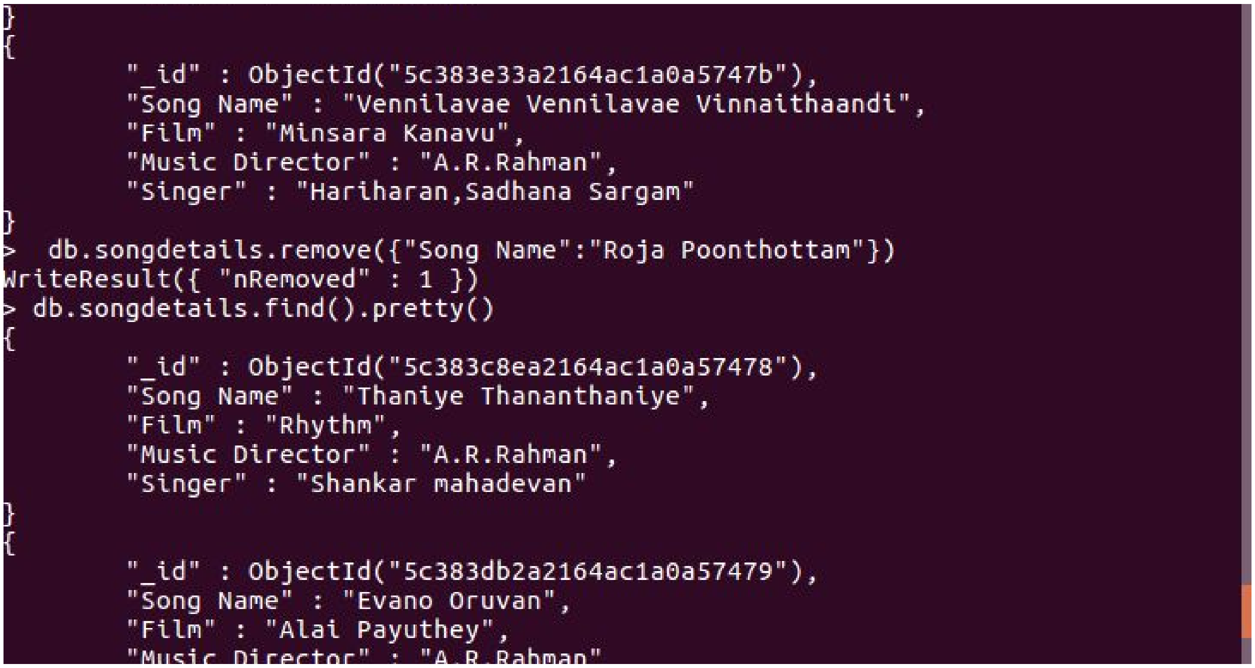


```
        "Music Director" : "Yuvan Shankar Raja",
        "Singer" : "Hariharan"
}
{
        "_id" : ObjectId("5c38441fa2164ac1a0a5747e"),
        "Song Name" : "Maruvarththai Peshathe",
        "Film" : "Ennai Nokki Pajum Thotta",
        "Music Director" : "Dhanush",
        "Singer" : "Sidsriram"
}
{
        "_id" : ObjectId("5c3848bba2164ac1a0a57482"),
        "SongName" : "Roja Poonthottam",
        "Film" : "Kannukul Nilavu",
        "Music Director" : "Ilayaraaja",
        "Singer" : [
                "Unnikrishnan",
                "Anuradha Sriram"
        ]
}
> db.songdetails.find({"Music Director":"Ilayaraaja","Singer":"Unnikrishnan"},{"
SongName":1,_id:0}).pretty()
{ "SongName" : "Roja Poonthottam" }
>
```

7.Delete the song which you don’t like

> db.songdetails.remove({"Song Name":"Roja Poonthottam"}).pretty()

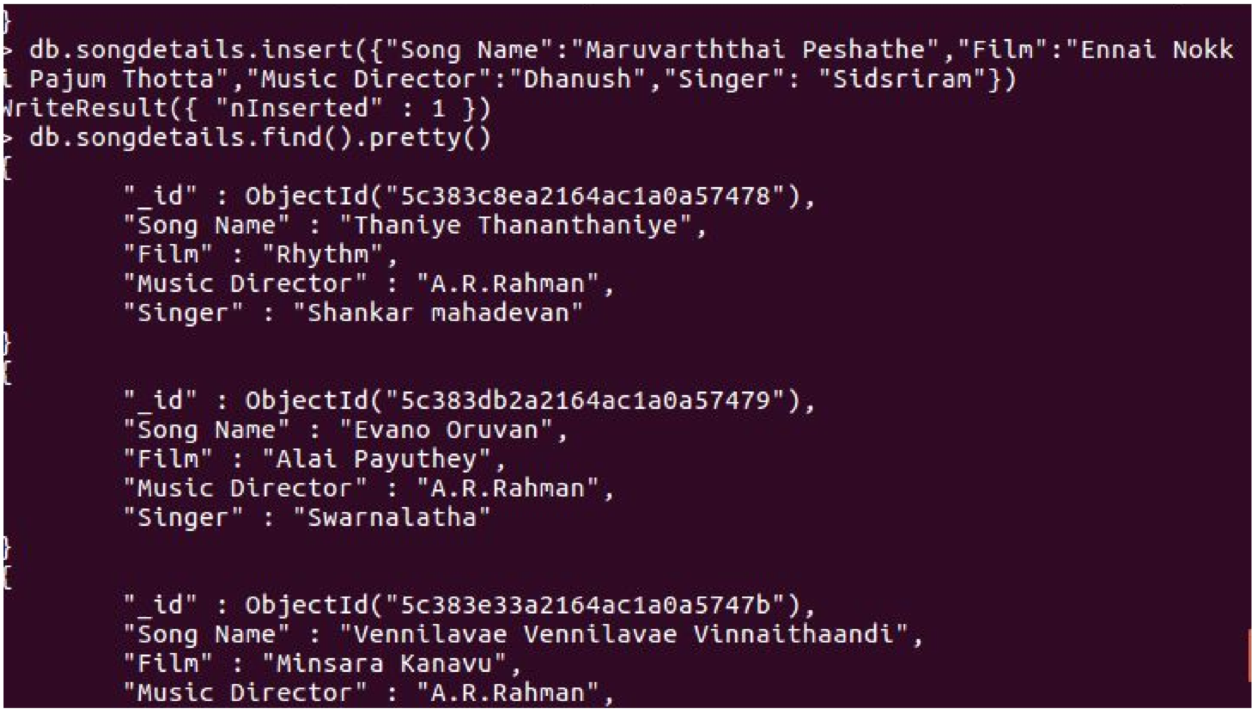
Screenshot:



```
}
{
        "_id" : ObjectId("5c383e33a2164ac1a0a5747b"),
        "Song Name" : "Vennilavae Vennilavae Vinnaithaandi",
        "Film" : "Minsara Kanavu",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Hariharan,Sadhana Sargam"
}
>  db.songdetails.remove({"Song Name":"Roja Poonthottam"})
WriteResult({ "nRemoved" : 1 })
> db.songdetails.find().pretty()
{
        "_id" : ObjectId("5c383c8ea2164ac1a0a57478"),
        "Song Name" : "Thaniye Thananthaniye",
        "Film" : "Rhythm",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Shankar mahadevan"
}
{
        "_id" : ObjectId("5c383db2a2164ac1a0a57479"),
        "Song Name" : "Evano Oruvan",
        "Film" : "Alai Payuthey",
        "Music Director" : "A.R.Rahman"
```

8.Add new song which is your favourite

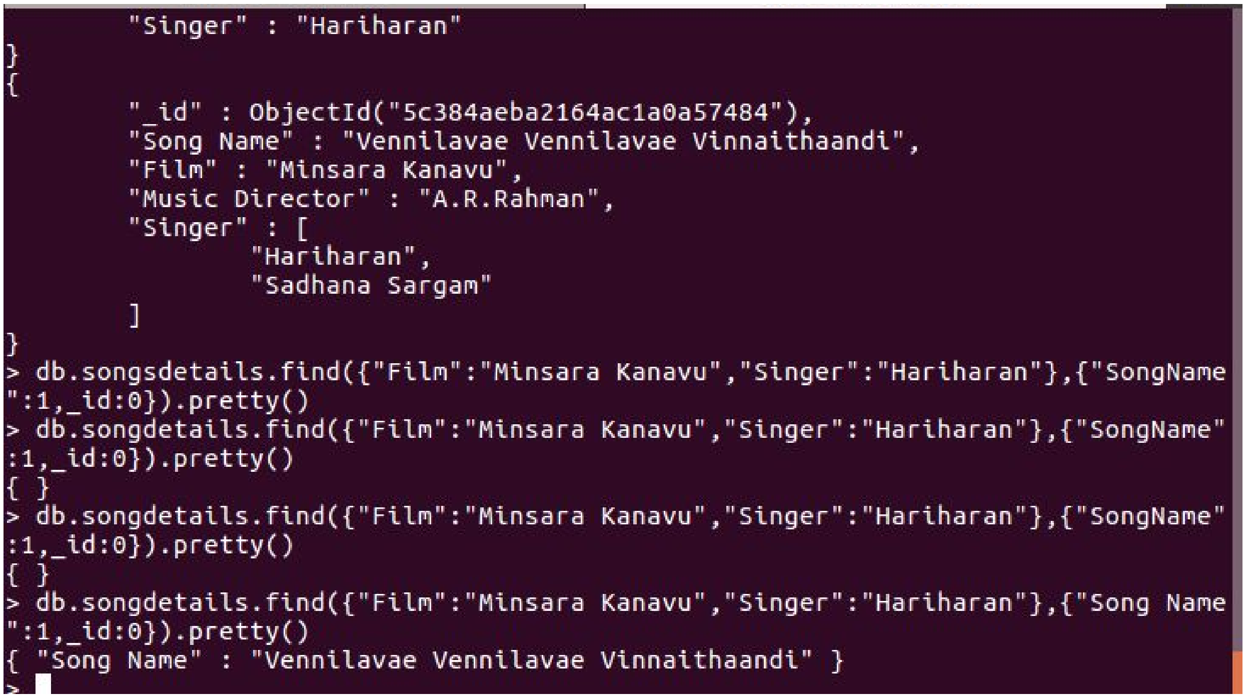
> db.songdetails.insert({"Song Name":"Maruvarththai Peshathe","Film":"Ennai Nokki Pajum Thotta","Music Director":"Dhanush","Singer": "Sidsriram"})



```
}
> db.songdetails.insert({"Song Name":"Maruvarththai Peshathe","Film":"Ennai Nokk
i Pajum Thotta","Music Director":"Dhanush","Singer": "Sidsriram"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find().pretty()
{
        "_id" : ObjectId("5c383c8ea2164ac1a0a57478"),
        "Song Name" : "Thaniye Thananthaniye",
        "Film" : "Rhythm",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Shankar mahadevan"
}
{
        "_id" : ObjectId("5c383db2a2164ac1a0a57479"),
        "Song Name" : "Evano Oruvan",
        "Film" : "Alai Payuthey",
        "Music Director" : "A.R.Rahman",
        "Singer" : "Swarnalatha"
}
{
        "_id" : ObjectId("5c383e33a2164ac1a0a5747b"),
        "Song Name" : "Vennilavae Vennilavae Vinnaithaandi",
        "Film" : "Minsara Kanavu",
        "Music Director" : "A.R.Rahman",
```

9.List Songs sung by Hariharan from Minsara kanavu film

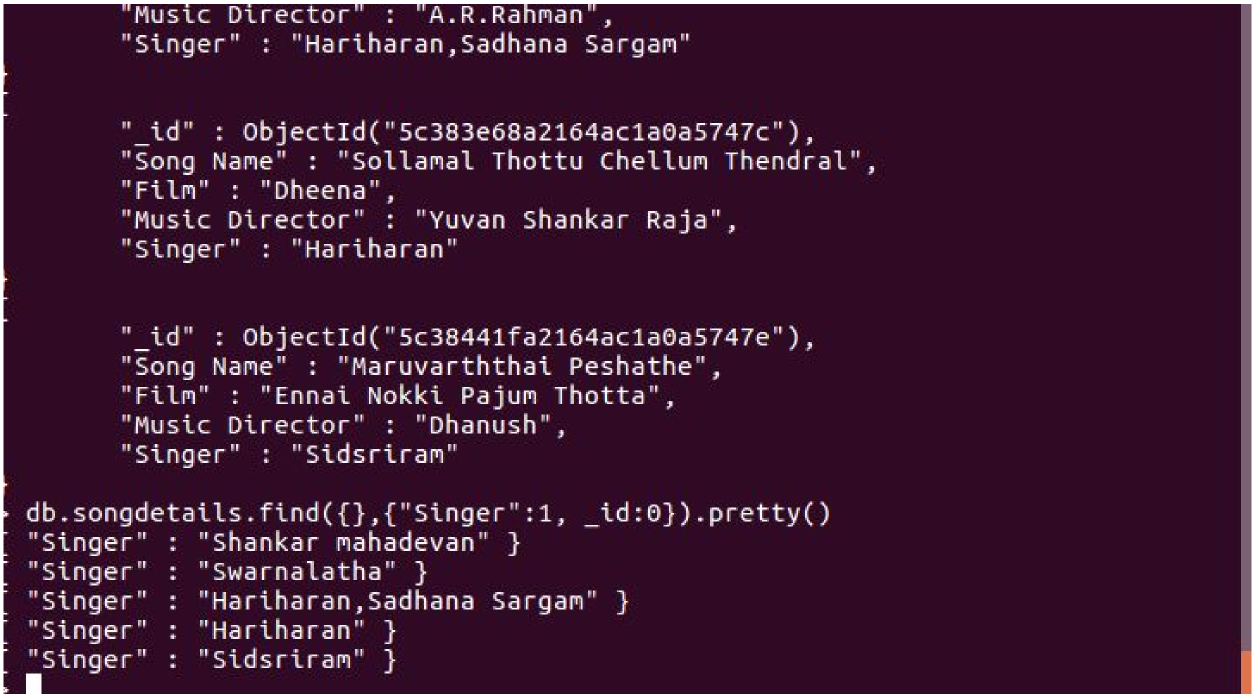
>db.songdetails.find({"Film":"Minsara Kanavu","Singer":"Hariharan"},{"Song Name":1,_id:0}).pretty()



```
        "Singer" : "Hariharan"
}
{
        "_id" : ObjectId("5c384aeba2164ac1a0a57484"),
        "Song Name" : "Vennilavae Vennilavae Vinnaithaandi",
        "Film" : "Minsara Kanavu",
        "Music Director" : "A.R.Rahman",
        "Singer" : [
                "Hariharan",
                "Sadhana Sargam"
        ]
}
> db.songsdetails.find({"Film":"Minsara Kanavu","Singer":"Hariharan"},{"SongName
":1,_id:0}).pretty()
> db.songdetails.find({"Film":"Minsara Kanavu","Singer":"Hariharan"},{"SongName"
:1,_id:0}).pretty()
{ }
> db.songdetails.find({"Film":"Minsara Kanavu","Singer":"Hariharan"},{"SongName"
:1,_id:0}).pretty()
{ }
> db.songdetails.find({"Film":"Minsara Kanavu","Singer":"Hariharan"},{"Song Name
":1,_id:0}).pretty()
{ "Song Name" : "Vennilavae Vennilavae Vinnaithaandi" }
>
```

10.List out the singers’ names in your document

> db.songdetails.find({},{"Singer":1, _id:0}).pretty()



```
        "Music Director" : "A.R.Rahman",
        "Singer" : "Hariharan,Sadhana Sargam"
}
{
        "_id" : ObjectId("5c383e68a2164ac1a0a5747c"),
        "Song Name" : "Sollamal Thottu Chellum Thendral",
        "Film" : "Dheena",
        "Music Director" : "Yuvan Shankar Raja",
        "Singer" : "Hariharan"
}
{
        "_id" : ObjectId("5c38441fa2164ac1a0a5747e"),
        "Song Name" : "Maruvarththai Peshathe",
        "Film" : "Ennai Nokki Pajum Thotta",
        "Music Director" : "Dhanush",
        "Singer" : "Sidsriram"
}
> db.songdetails.find({},{"Singer":1, _id:0}).pretty()
{ "Singer" : "Shankar mahadevan" }
{ "Singer" : "Swarnalatha" }
{ "Singer" : "Hariharan,Sadhana Sargam" }
{ "Singer" : "Hariharan" }
{ "Singer" : "Sidsriram" }
>
```