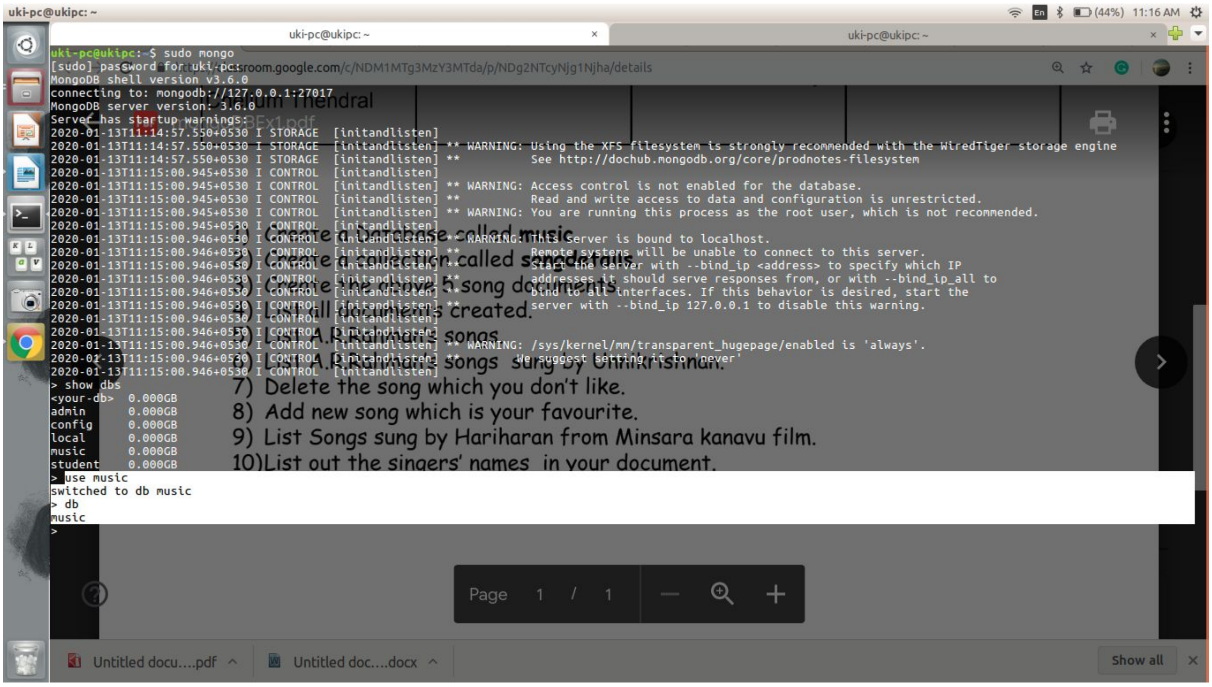
Mongodb Ex1

Q1

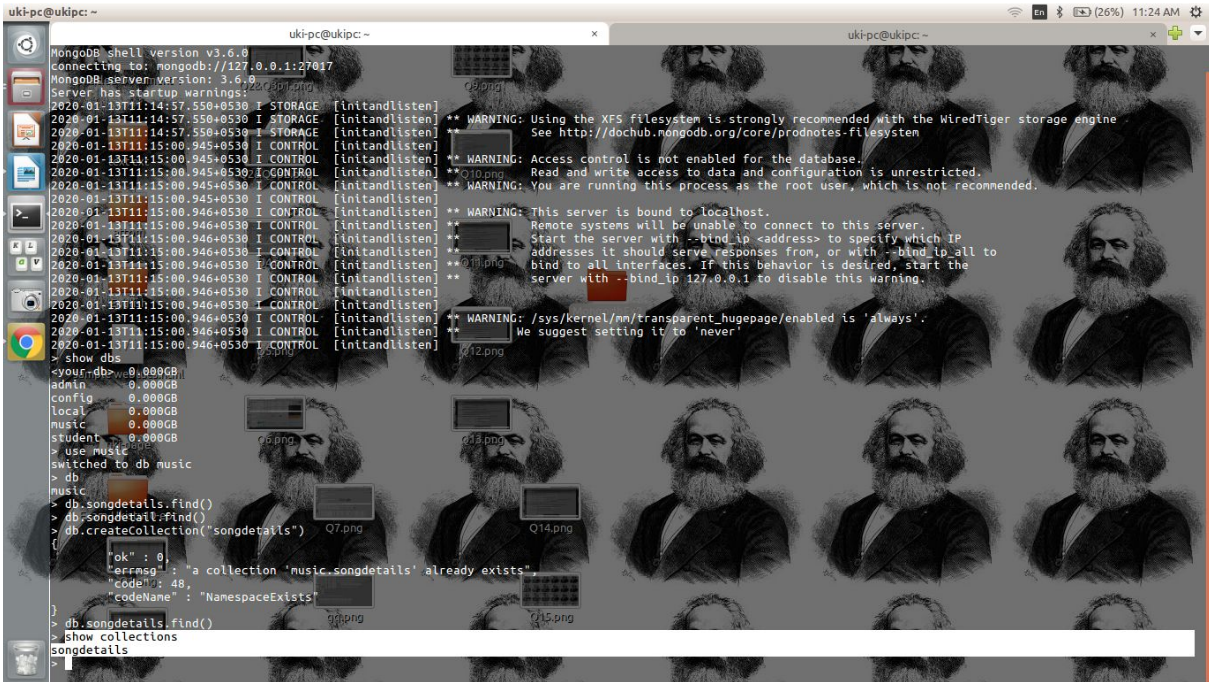


```
uki-pc@ukipc:~$ sudo mongo
[sudo] password for uki-pc:
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten]
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
> show dbs
<your-db>  0.000GB
admin      0.000GB
config     0.000GB
local      0.000GB
music      0.000GB
student    0.000GB
> use music
switched to db music
> db
music
>
```

Chelum Thendral

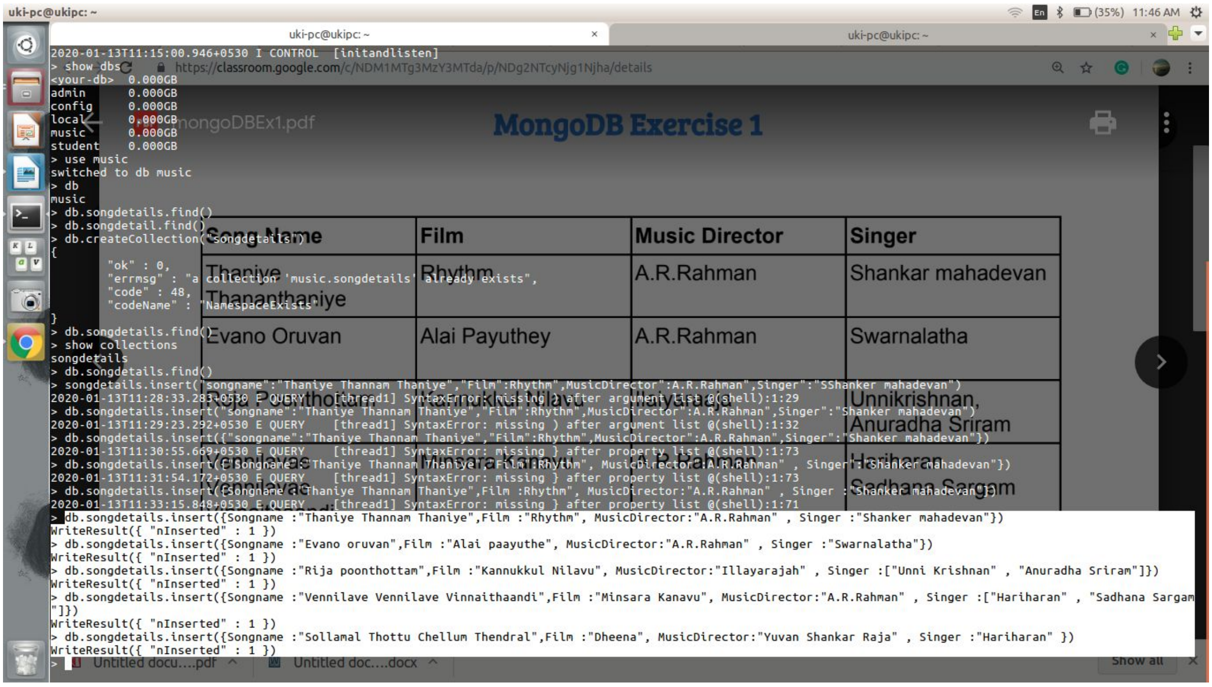
1) Create a Database called **music**
2) Create a collection called **songdetails**
3) Create the above 5 song documents.
4) List all documents created.
5) List A.R.Rahman's songs
6) List A.R.Rahman's songs sung by Unnikrishnan.
7) Delete the song which you don't like.
8) Add new song which is your favourite.
9) List Songs sung by Hariharan from Minsara kanavu film.
10)List out the singers' names in your document.

Q2



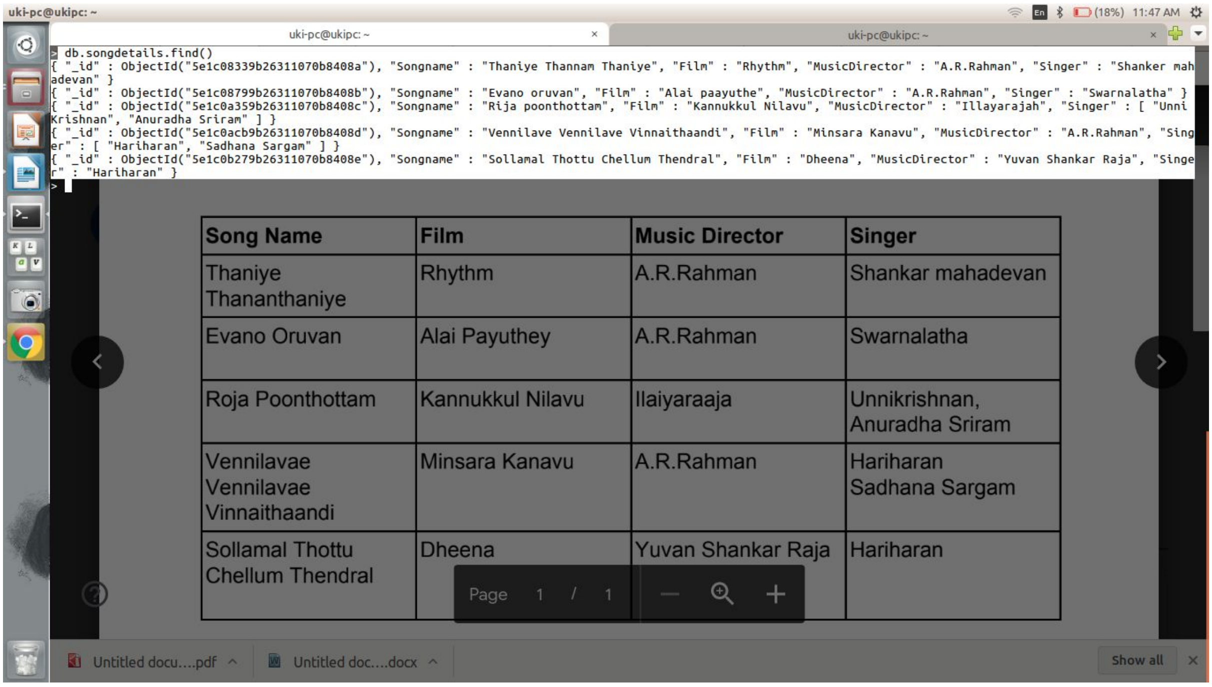
```
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten]
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-01-13T11:14:57.550+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2020-01-13T11:15:00.945+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
> show dbs
<your-db>  0.000GB
admin      0.000GB
config     0.000GB
local      0.000GB
music      0.000GB
student    0.000GB
> use music
switched to db music
> db
music
> db.songdetails.find()
> db.songdetail.find()
> db.createCollection("songdetails")
{
        "ok" : 0,
        "errmsg" : "a collection 'music.songdetails' already exists",
        "code" : 48,
        "codeName" : "NamespaceExists"
}
> db.songdetails.find()
> show collections
songdetails
>
```

Q3



```
2020-01-13T11:15:00.946+0530 I CONTROL  [initandlisten]
> show dbs
<your-db>  0.000GB
admin      0.000GB
config     0.000GB
local      0.000GB
music      0.000GB
student    0.000GB
> use music
switched to db music
> db
music
> db.songdetails.find()
> db.songdetail.find()
> db.createCollection("songdetails")
{
        "ok" : 0,
        "errmsg" : "a collection 'music.songdetails' already exists",
        "code" : 48,
        "codeName" : "NamespaceExists"
}
> db.songdetails.find()
> show collections
songdetails
> db.songdetails.find()
> songdetails.insert("songname":"Thaniye Thannam Thaniye","Film":Rhythm",MusicDirector":A.R.Rahman",Singer":"SShanker mahadevan")
2020-01-13T11:28:33.283+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:29
> db.songdetails.insert("songname":"Thaniye Thannam Thaniye","Film":Rhythm",MusicDirector":A.R.Rahman",Singer":"Shanker mahadevan")
2020-01-13T11:29:23.292+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:32
> db.songdetails.insert({"songname":"Thaniye Thannam Thaniye","Film":Rhythm",MusicDirector":A.R.Rahman",Singer":"Shanker mahadevan"})
2020-01-13T11:30:55.669+0530 E QUERY    [thread1] SyntaxError: missing } after property list @(shell):1:73
> db.songdetails.insert({"Songname :"Thaniye Thannam Thaniye","Film :Rhythm", MusicDirector:"A.R.Rahman" , Singer":"Shanker mahadevan"})
2020-01-13T11:31:54.172+0530 E QUERY    [thread1] SyntaxError: missing } after property list @(shell):1:73
> db.songdetails.insert({Songname :"Thaniye Thannam Thaniye",Film :Rhythm", MusicDirector:"A.R.Rahman" , Singer :"Shanker mahadevan"})
2020-01-13T11:33:15.848+0530 E QUERY    [thread1] SyntaxError: missing } after property list @(shell):1:71
> db.songdetails.insert({Songname :"Thaniye Thannam Thaniye",Film :"Rhythm", MusicDirector:"A.R.Rahman" , Singer :"Shanker mahadevan"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({Songname :"Evano oruvan",Film :"Alai paayuthe", MusicDirector:"A.R.Rahman" , Singer :"Swarnalatha"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({Songname :"Rija poonthottam",Film :"Kannukkul Nilavu", MusicDirector:"Illayarajah" , Singer :["Unni Krishnan" , "Anuradha Sriram"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({Songname :"Vennilave Vennilave Vinnaithaandi",Film :"Minsara Kanavu", MusicDirector:"A.R.Rahman" , Singer :["Hariharan" , "Sadhana Sargam
"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({Songname :"Sollamal Thottu Chellum Thendral",Film :"Dheena", MusicDirector:"Yuvan Shankar Raja" , Singer :"Hariharan" })
WriteResult({ "nInserted" : 1 })
>
```

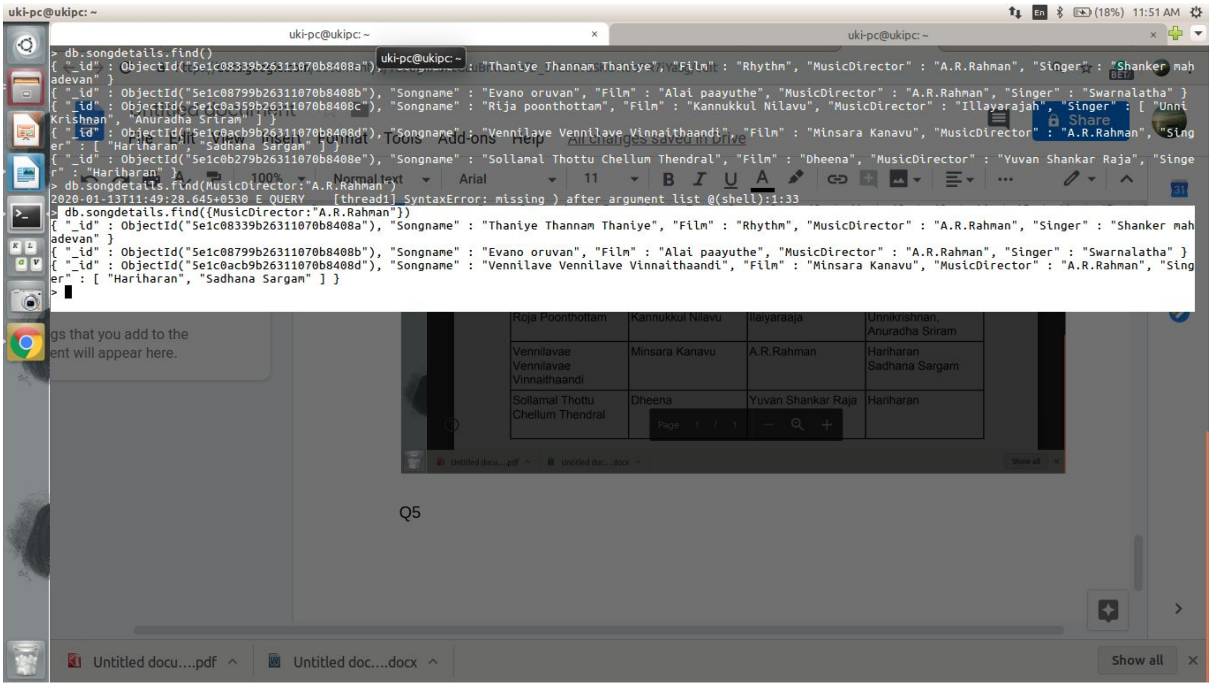
Q4



```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
>
```

Song Name	Film	Music Director	Singer
Thaniye Thananthaniye	Rhythm	A.R.Rahman	Shankar mahadevan
Evano Oruvan	Alai Payuthey	A.R.Rahman	Swarnalatha
Roja Poonthottam	Kannukkul Nilavu	Ilaiyaraaja	Unnikrishnan, Anuradha Sriram
Vennilavae Vennilavae Vinnaithaandi	Minsara Kanavu	A.R.Rahman	Hariharan Sadhana Sargam
Sollamal Thottu Chellum Thendral	Dheena	Yuvan Shankar Raja	Hariharan

Q5

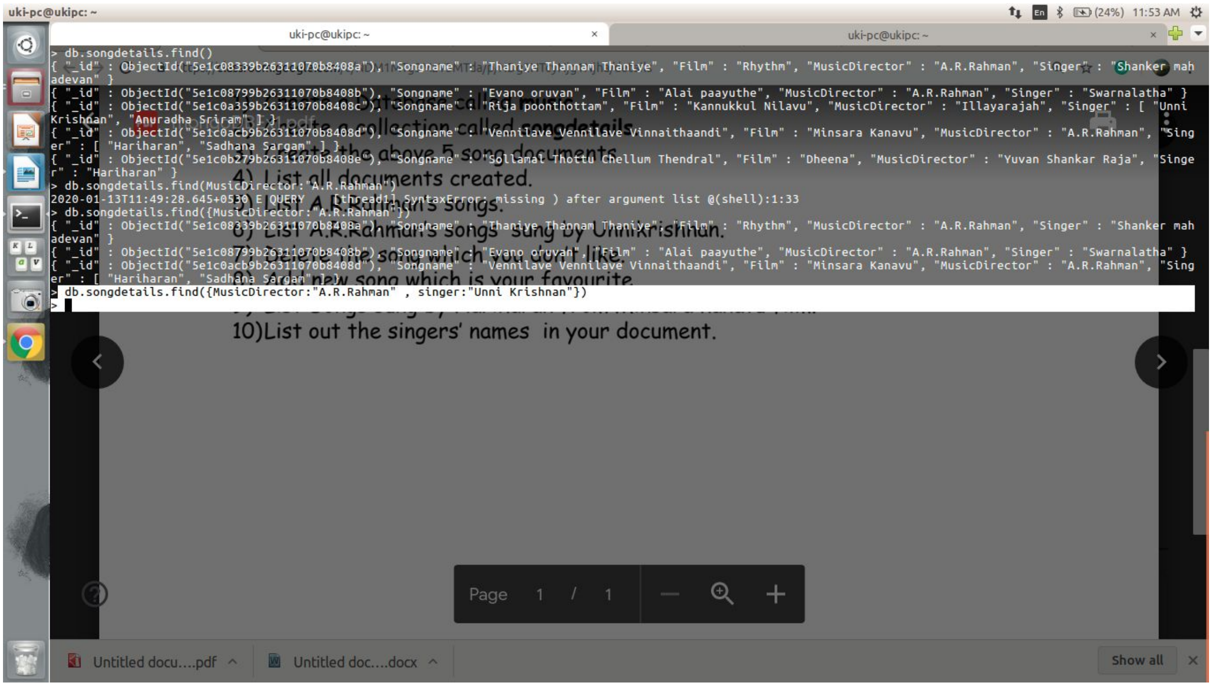


```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> 
```

Roja Poonthottam	Kannukkul Nilavu	Ilaiyaraaja	Unnikrishnan, Anuradha Sriram
Vennilavae Vennilavae Vinnaithaandi	Minsara Kanavu	A.R.Rahman	Hariharan Sadhana Sargam
Sollamal Thottu Chellum Thendral	Dheena	Yuvan Shankar Raja	Hariharan

Q5

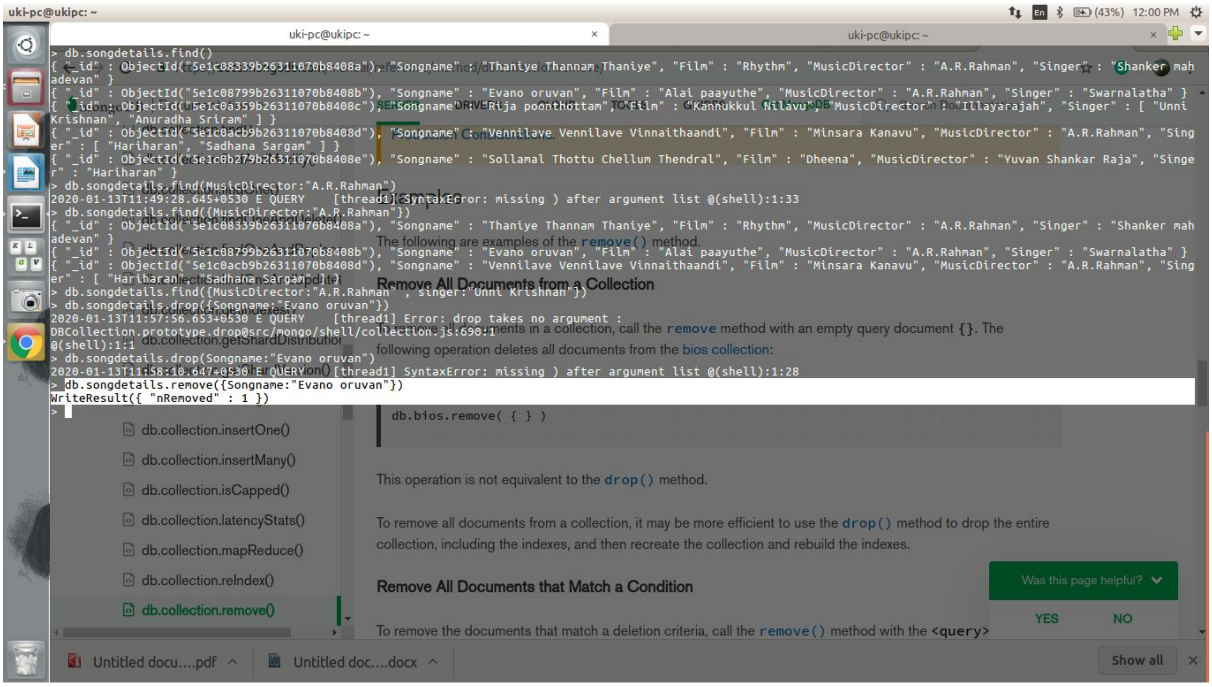
Q6



```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> db.songdetails.find({MusicDirector:"A.R.Rahman" , singer:"Unni Krishnan"})
> 
```

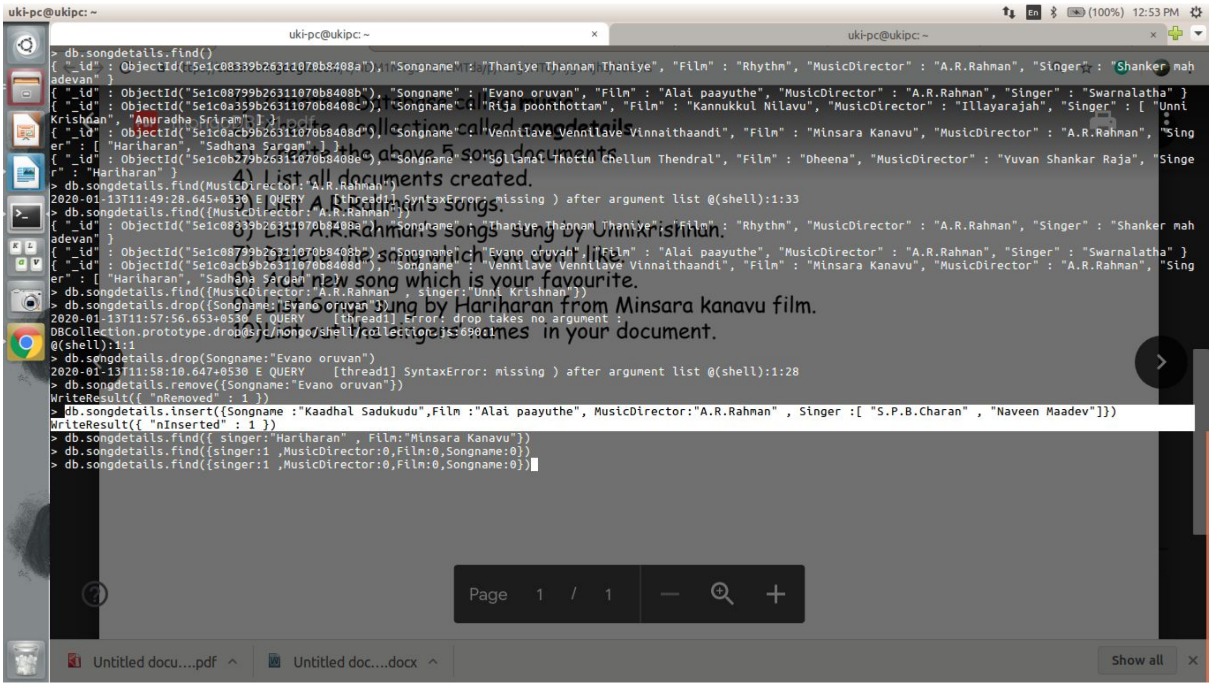
10)List out the singers' names in your document.

Q7



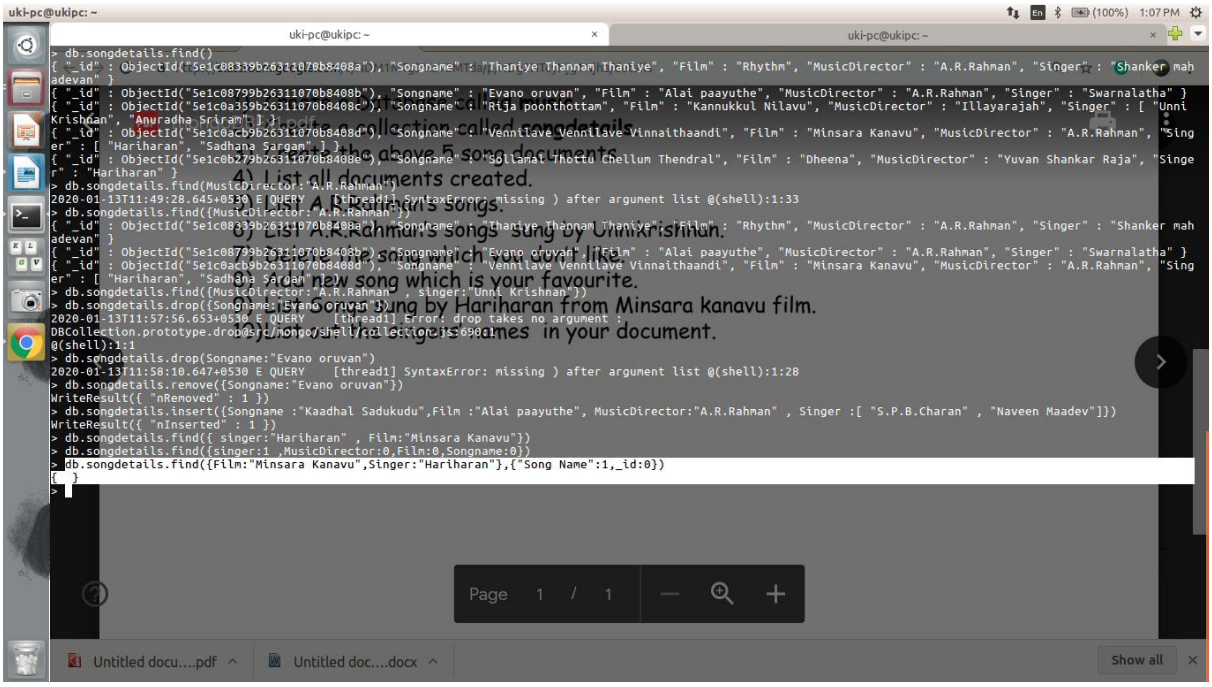
```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> db.songdetails.find({MusicDirector:"A.R.Rahman" , singer:"Unni Krishnan"})
> db.songdetails.drop({Songname:"Evano oruvan"})
2020-01-13T11:57:56.653+0530 E QUERY    [thread1] Error: drop takes no argument :
DBCollection.prototype.drop@src/mongo/shell/collection.js:690:1
@(shell):1:1
> db.songdetails.drop(Songname:"Evano oruvan")
2020-01-13T11:58:10.647+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:28
> db.songdetails.remove({Songname:"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> 
```

Q8



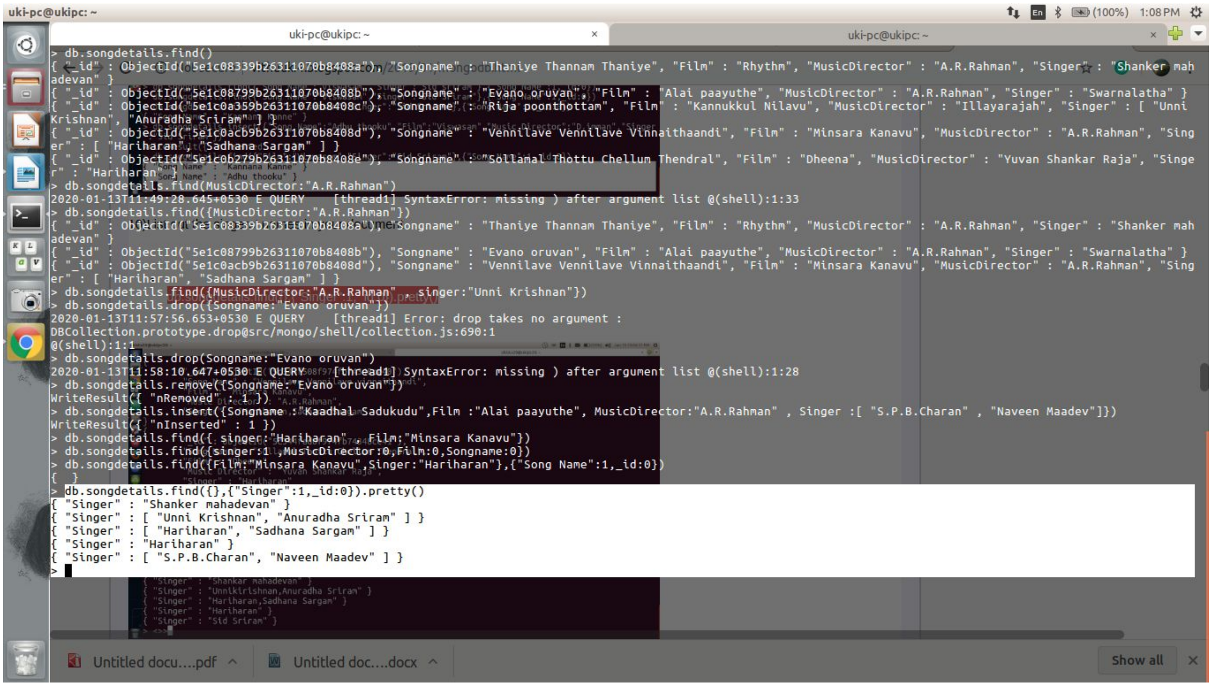
```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> db.songdetails.find({MusicDirector:"A.R.Rahman" , singer:"Unni Krishnan"})
> db.songdetails.drop({Songname:"Evano oruvan"})
2020-01-13T11:57:56.653+0530 E QUERY    [thread1] Error: drop takes no argument :
DBCollection.prototype.drop@src/mongo/shell/collection.js:690:1
@(shell):1:1
> db.songdetails.drop(Songname:"Evano oruvan")
2020-01-13T11:58:10.647+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:28
> db.songdetails.remove({Songname:"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> db.songdetails.insert({Songname :"Kaadhal Sadukudu",Film :"Alai paayuthe", MusicDirector:"A.R.Rahman" , Singer :[ "S.P.B.Charan" , "Naveen Maadev"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find({ singer:"Hariharan" , Film:"Minsara Kanavu"})
> db.songdetails.find({singer:1 ,MusicDirector:0,Film:0,Songname:0})
> db.songdetails.find({singer:1 ,MusicDirector:0,Film:0,Songname:0})
```

Q9



```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> db.songdetails.find({MusicDirector:"A.R.Rahman" , singer:"Unni Krishnan"})
> db.songdetails.drop({Songname:"Evano oruvan"})
2020-01-13T11:57:56.653+0530 E QUERY    [thread1] Error: drop takes no argument :
DBCollection.prototype.drop@src/mongo/shell/collection.js:690:1
@(shell):1:1
> db.songdetails.drop(Songname:"Evano oruvan")
2020-01-13T11:58:10.647+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:28
> db.songdetails.remove({Songname:"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> db.songdetails.insert({Songname :"Kaadhal Sadukudu",Film :"Alai paayuthe", MusicDirector:"A.R.Rahman" , Singer :[ "S.P.B.Charan" , "Naveen Maadev"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find({ singer:"Hariharan" , Film:"Minsara Kanavu"})
> db.songdetails.find({singer:1 ,MusicDirector:0,Film:0,Songname:0})
> db.songdetails.find({Film:"Minsara Kanavu",Singer:"Hariharan"},{"Song Name":1,_id:0})
{  }
> 
```

Q10



```
> db.songdetails.find()
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0a359b26311070b8408c"), "Songname" : "Rija poonthottam", "Film" : "Kannukkul Nilavu", "MusicDirector" : "Illayarajah", "Singer" : [ "Unni
Krishnan", "Anuradha Sriram" ] }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
{ "_id" : ObjectId("5e1c0b279b26311070b8408e"), "Songname" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "MusicDirector" : "Yuvan Shankar Raja", "Singe
r" : "Hariharan" }
> db.songdetails.find(MusicDirector:"A.R.Rahman")
2020-01-13T11:49:28.645+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:33
> db.songdetails.find({MusicDirector:"A.R.Rahman"})
{ "_id" : ObjectId("5e1c08339b26311070b8408a"), "Songname" : "Thaniye Thannam Thaniye", "Film" : "Rhythm", "MusicDirector" : "A.R.Rahman", "Singer" : "Shanker mah
adevan" }
{ "_id" : ObjectId("5e1c08799b26311070b8408b"), "Songname" : "Evano oruvan", "Film" : "Alai paayuthe", "MusicDirector" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "_id" : ObjectId("5e1c0acb9b26311070b8408d"), "Songname" : "Vennilave Vennilave Vinnaithaandi", "Film" : "Minsara Kanavu", "MusicDirector" : "A.R.Rahman", "Sing
er" : [ "Hariharan", "Sadhana Sargam" ] }
> db.songdetails.find({MusicDirector:"A.R.Rahman" , singer:"Unni Krishnan"})
> db.songdetails.drop({Songname:"Evano oruvan"})
2020-01-13T11:57:56.653+0530 E QUERY    [thread1] Error: drop takes no argument :
DBCollection.prototype.drop@src/mongo/shell/collection.js:690:1
@(shell):1:1
> db.songdetails.drop(Songname:"Evano oruvan")
2020-01-13T11:58:10.647+0530 E QUERY    [thread1] SyntaxError: missing ) after argument list @(shell):1:28
> db.songdetails.remove({Songname:"Evano oruvan"})
WriteResult({ "nRemoved" : 1 })
> db.songdetails.insert({Songname :"Kaadhal Sadukudu",Film :"Alai paayuthe", MusicDirector:"A.R.Rahman" , Singer :[ "S.P.B.Charan" , "Naveen Maadev"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find({ singer:"Hariharan" , Film:"Minsara Kanavu"})
> db.songdetails.find({singer:1 ,MusicDirector:0,Film:0,Songname:0})
> db.songdetails.find({Film:"Minsara Kanavu",Singer:"Hariharan"},{"Song Name":1,_id:0})
{  }
> db.songdetails.find({},{"Singer":1,_id:0}).pretty()
{ "Singer" : "Shanker mahadevan" }
{ "Singer" : [ "Unni Krishnan", "Anuradha Sriram" ] }
{ "Singer" : [ "Hariharan", "Sadhana Sargam" ] }
{ "Singer" : "Hariharan" }
{ "Singer" : [ "S.P.B.Charan", "Naveen Maadev" ] }
> 
```