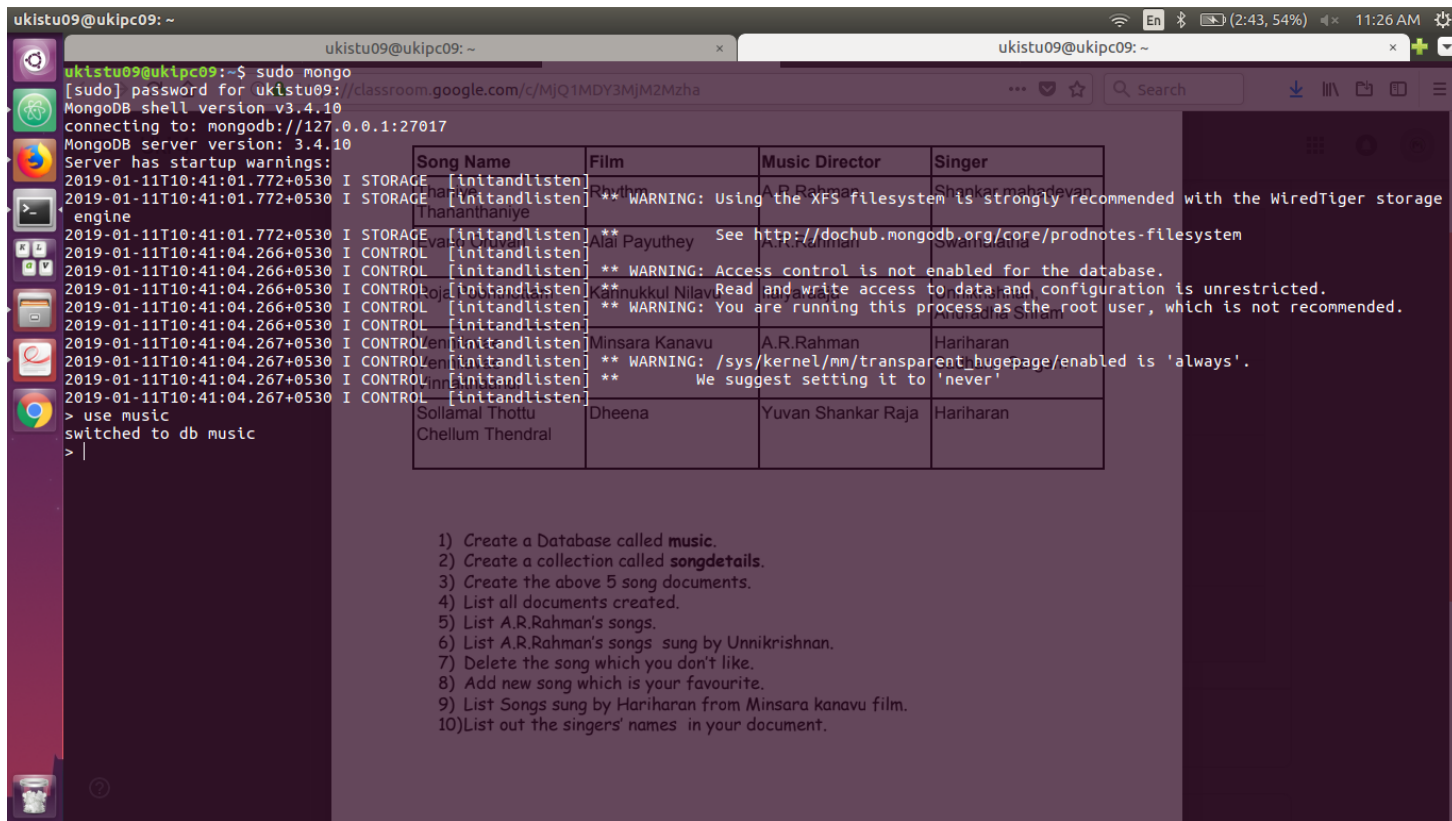


Mongo DB Exercise 1

- 1) Create a Database called music.
use music



```
ukistu09@ukipc09: ~  
ukistu09@ukipc09:~$ sudo mongo  
[sudo] password for ukistu09: /classroom.google.com/c/MjQ1MDY3MjM2Mzha  
MongoDB shell version v3.4.10  
connecting to: mongodb://127.0.0.1:27017  
MongoDB server version: 3.4.10  
Server has startup warnings:  
2019-01-11T10:41:01.772+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2019-01-11T10:41:01.772+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] Read and write access to data and configuration is unrestricted.  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] We suggest setting it to 'never'  
2019-01-11T10:41:04.267+0530 I CONTROL [initandlisten]  
2019-01-11T10:41:04.267+0530 I CONTROL [initandlisten]  
2019-01-11T10:41:04.267+0530 I CONTROL [initandlisten]  
> use music  
switched to db music  
> |
```

Song Name	Film	Music Director	Singer
Thaananthaniye	Rhythm	A.R.Rahman	Shreya Ghoshal
Vaa	Alai Payuthey	A.R.Rahman	Shreya Ghoshal
Kannukkul Nilavu	Kannukkul Nilavu	A.R.Rahman	Shreya Ghoshal
Minsara Kanavu	Minsara Kanavu	A.R.Rahman	Hariharan
Sollamalai Thottu	Dheena	Yuvan Shankar Raja	Hariharan
Chellum 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.

2) Create a collection called songdetails.

```
db.createCollection("Songdetails")
```

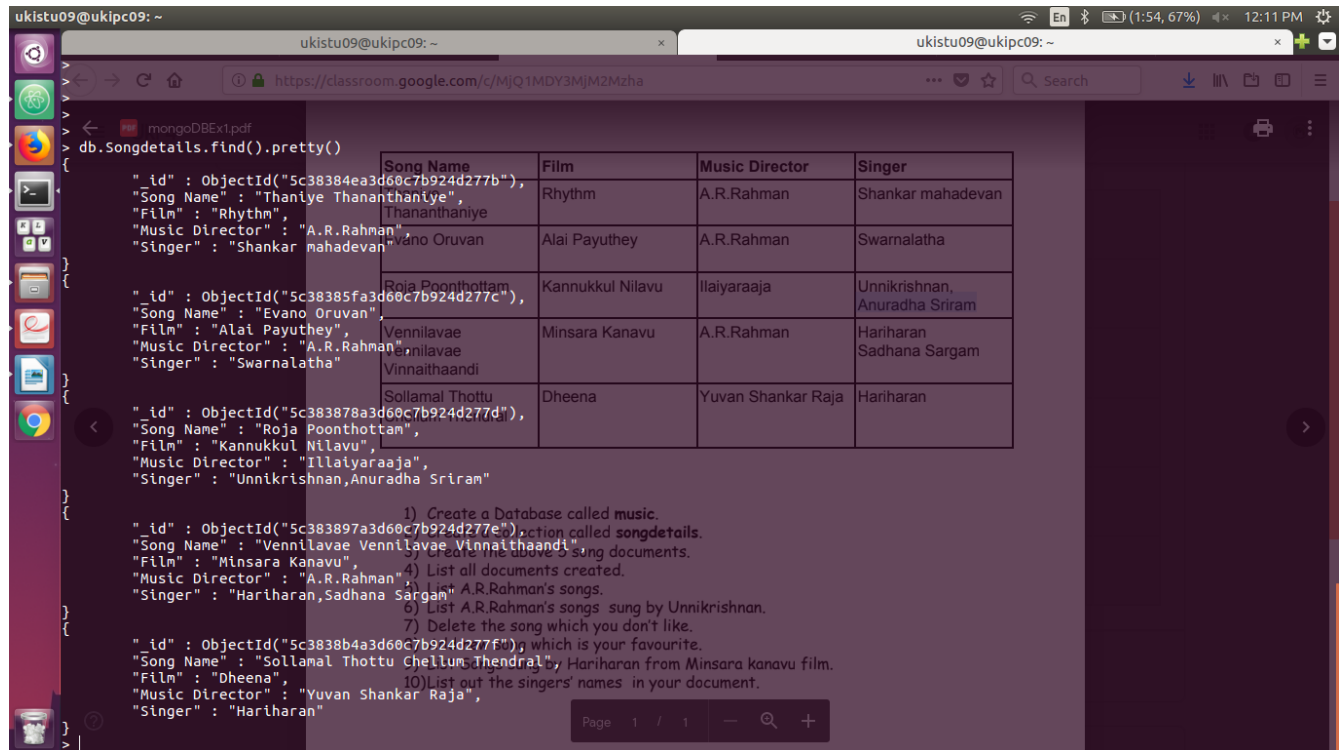
```
ukistu09@ukipc09: ~  
ukistu09@ukipc09:~$ sudo mongo  
[sudo] password for ukistu09:  
MongoDB shell version v3.4.10  
connecting to: mongod://127.0.0.1:27017  
MongoDB server version: 3.4.10  
Server has startup warnings:  
2019-01-11T10:41:01.772+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2019-01-11T10:41:01.772+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] Read and write access to data and configuration is unrestricted.  
2019-01-11T10:41:04.266+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2019-01-11T10:41:04.267+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2019-01-11T10:41:04.267+0530 I CONTROL [initandlisten] We suggest setting it to 'never'  
> use music  
switched to db music  
> db.createCollection("Songdetails")  
{ "ok" : 1 }  
>  
  
ukistu09@ukipc09:~$ mongo  
MongoDB shell version v3.4.10  
connecting to: mongod://127.0.0.1:27017  
MongoDB server version: 3.4.10  
ukistu09@ukipc09:~$ use music  
switched to db music  
ukistu09@ukipc09:~$ db.songdetails.find().pretty()  
[{"_id": "Songdetails", "Song Name": "Sollalai Thattu Chellum Thendral", "File": "Oheena", "Music Director": "Yuvan S", "Singer": "Shankar mahadevan"},  
{"_id": "Songdetails", "Song Name": "Rama Krishna", "File": "Rama Krishna", "Music Director": "A.R.Rahman", "Singer": "Swarnalatha"},  
{"_id": "Songdetails", "Song Name": "Roja Roosthottam", "File": "Roja Roosthottam", "Music Director": "Ilaiyaraaja", "Singer": "Unnikrishnan, Anuradha Sriram"},  
{"_id": "Songdetails", "Song Name": "Vennilave Vennilave Vinnathaandi", "File": "Rikhsara Kanava", "Music Director": "A.R.Rahman", "Singer": "Naiharan, Sathana Sargan"}]  
ukistu09@ukipc09:~$
```

4) List all documents created

The image shows a Linux desktop with a terminal window and a web browser. The terminal window is titled 'ukistu09@ukip09:~' and shows a series of MongoDB commands and their output. The commands include creating a database named 'music', creating a collection named 'songdetails', and inserting five song documents. The output shows the documents being inserted and a list of all documents in the collection. The web browser window is titled 'ukistu09@ukip09:~' and shows a search for 'ukistu09@ukip09:~'. The search results show a link to 'https://classroom.google.com/c/MjQ1MDY3MjM2Mzha' and a warning about XFS filesystem. The terminal window also shows a warning about access control not being enabled for the database. The desktop background is a dark purple color with a grid of application icons on the left side.

4) List all documents created.

db.Songdetails.find() or db.Songdetails.find().pretty()



The screenshot shows a terminal window with the following commands and output:

```
ukistu09@ukipc09: ~  
ukistu09@ukipc09: ~  
https://classroom.google.com/c/MJQ1MDY3MjM2Mzha  
mongoDBEx1.pdf  
> db.Songdetails.find().pretty()
```

Song Name	Film	Music Director	Singer
"_id" : ObjectId("5c38384ea3d60c7b924d277b"), "Song Name" : "Thaniye Thananthaniye", "Film" : "Rhythm", "Music Director" : "A.R.Rahman", "Singer" : "Shankar mahadevan"	Rhythm	A.R.Rahman	Shankar mahadevan
"_id" : ObjectId("5c38385fa3d60c7b924d277c"), "Song Name" : "Evano Oruvan", "Film" : "Alai Payuthey", "Music Director" : "A.R.Rahman", "Singer" : "Swarnalatha"	Alai Payuthey	A.R.Rahman	Swarnalatha
"_id" : ObjectId("5c383878a3d60c7b924d277d"), "Song Name" : "Raja Poonthottam", "Film" : "Kannukkul Nilavu", "Music Director" : "Ilaiyaraaja", "Singer" : "Unnikrishnan, Anuradha Sriram"	Kannukkul Nilavu	Ilaiyaraaja	Unnikrishnan, Anuradha Sriram
"_id" : ObjectId("5c383897a3d60c7b924d277e"), "Song Name" : "Vennilavae Vennilavae Vinnalthaandi", "Film" : "Minsara Kanavu", "Music Director" : "A.R.Rahman", "Singer" : "Hariharan, Sadhana Sargam"	Minsara Kanavu	A.R.Rahman	Hariharan, Sadhana Sargam
"_id" : ObjectId("5c3838b4a3d60c7b924d277f"), "Song Name" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "Music Director" : "Yuvan Shankar Raja", "Singer" : "Hariharan"	Dheena	Yuvan Shankar Raja	Hariharan

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) Create a document, which is your favourite.
9) Create a document, Hariharan from Minsara kanavu film.
10) List out the singers' names in your document.

Page 1 / 1

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

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

The screenshot shows a MongoDB web interface with a query result for A.R. Rahman's songs. The query is `db.Songdetails.find({"Music Director":"A.R.Rahman"}).pretty()`. The result is displayed as a table with 4 columns: Song Name, Film, Music Director, and Singer.

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 Vinnathaandi	Minsara Kanavu	A.R.Rahman	Hariharan Sadhana Sargam
Sollamalai Thottu	Dheena	Yuvan Shankar Raja	Hariharan

Below the table, the JSON representation of the first document is shown:

```
{
  "_id": "5c38384ea3d60c7b924d277b",
  "Song Name": "Thaniye Thananthaniye",
  "Film": "Rhythm",
  "Music Director": "A.R.Rahman",
  "Singer": "Shankar mahadevan"
}
```

On the left side of the interface, a list of tasks is visible:

- 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) List the song which is your favourite.
- 9) List out the singers' names in your document.
- 10) List out the singers' names in your document.

6) List Illaiyaraaja's songs sung by Unnikrishnan.

```
db.Songdetails.find({"Music Director":"Illaiyaraaja","Singer":"Unnikrishnan"},
{"Song Name":"1",_id:0}).pretty()
```

Terminal Output:

```

> db.Songdetails.find({"Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"}, {"Song Name": "1"})
{"_id": ObjectId("5c383878a3d60c7b924d277d"), "Song Name": "Sollamal Thottu Chellum Thendral", "Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"},
{"_id": ObjectId("5c3840c3a3d60c7b924d2780"), "Song Name": "Naan Un", "Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"},
{"_id": ObjectId("5c3840c3a3d60c7b924d2781"), "Song Name": "Raja Poonthottam", "Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"},
{"_id": ObjectId("5c3840c3a3d60c7b924d2782"), "Song Name": "Vennilavae", "Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"},
{"_id": ObjectId("5c3840c3a3d60c7b924d2783"), "Song Name": "Vinnaihaandi", "Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"}

```

Song Name	Music Director	Singer
Sollamal Thottu Chellum Thendral	Illaiyaraaja	Unnikrishnan
Naan Un	Illaiyaraaja	Unnikrishnan
Raja Poonthottam	Illaiyaraaja	Unnikrishnan
Vennilavae	Illaiyaraaja	Unnikrishnan
Vinnaihaandi	Illaiyaraaja	Unnikrishnan

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

```
db.Songdetails.remove({"Film":"Alai Payuthey"})
```

The screenshot shows a terminal window with the following commands and output:

```
> db.mycol.remove({"Song Name":"Alai Payuthey"})
WriteResult({"nRemoved": 1})
> db.Songdetails.find({"Music Director":"A.R.Rahman"}).pretty()
```

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	Minsara Kanavu	A.R.Rahman	Hariharan, Sadhana Sargam
Vennilavae Vennilavae Vinnai Thandil	Dheena	Yuvan Shankar Raja	Hariharan

```
> db.Songdetails.remove({"Film":"Alai Payuthey"})
WriteResult({"nRemoved": 1})
> db.Songdetails.find({"Music Director":"A.R.Rahman"}).pretty()
```

1) Create a Database called music.
2) Create the above 5 song documents.
3) List all documents created.
4) List A.R.Rahman's songs.
5) List A.R.Rahman's songs sung by Unnikrishnan.
6) Delete the song which you don't like.
7) Delete the song which is your favourite.
8) List out the singers' names in your document.

8) Add new song which is your favourite.

db.Songdetails.insert({"Song Name":"Naan un", "Film":24 , "Music Director" :
"A.R.Rahman", "Singer":"Arijit Singh,Chinmayi"})

The screenshot shows a web browser window displaying a MongoDB interface. The browser's address bar shows a URL starting with 'http://localhost:27020/'. The page title is 'mongoDBEx1.pdf'. The main content area features a table with four columns: 'Song Name', 'Film', 'Music Director', and 'Singer'. The table contains five rows of data. Below the table, there is a list of ten tasks. The terminal window at the bottom shows a MongoDB command being executed to insert a new song document, and the result is displayed as 'WriteResult({ "nInserted" : 1 })'.

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 Vinnaihaandi	Minsara Kanavu	A.R.Rahman	Hariharan Sadhana Sargam
Sollamal Thottu Chellum Thendral	Dheena	Yuvar Shankar Raja	Hariharan

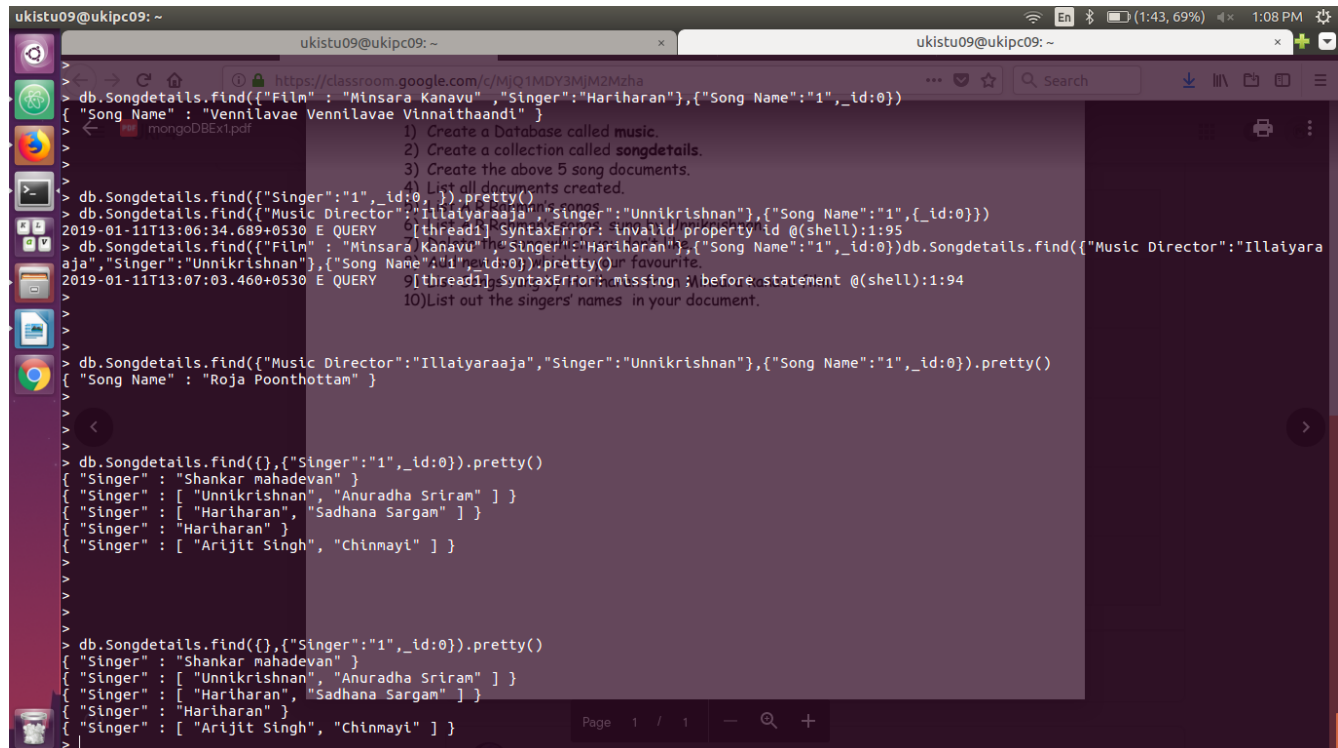
- 1) Create a Database called **music**.
- 2) Create a collection called **songdetails**.
- 3) Create the above 5 song documents.
- 4) List all documents created.
- 5) List A.R.Rahman's songs.
- 6) List A.R.Rahman's songs sung by Unnikrishnan.
- 7) Delete the song which you don't like.
- 8) Add new song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

```
> db.Songdetails.insert({"Song Name":"Naan un", "Film":24 , "Music Director" : "A.R.Rahman", "Singer":"Arijit Singh,Chinmayi"})
WriteResult({ "nInserted" : 1 })
```


[illegible]

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

```
db.Songdetails.find({}, {"Singer": "1", _id: 0}).pretty()
```



```
ukistu09@ukipc09: ~
> db.Songdetails.find({"Film": "Minsara Kanavu", "Singer": "Haritharan"}, {"Song Name": "1", _id: 0})
{ "Song Name" : "Vennilavae Vennilavae Vinnaitthaandi" }
1) Create a Database called music.
2) Create a collection called songdetails.
3) Create the above 5 song documents.
4) List all documents created.
> db.Songdetails.find({}, {"Singer": "1", _id: 0}).pretty()
{ "Singer" : "Shankar mahadevan" }
{ "Singer" : [ "Unnikrishnan", "Anuradha Sritam" ] }
{ "Singer" : [ "Haritharan", "Sadhana Sargam" ] }
{ "Singer" : "Haritharan" }
{ "Singer" : [ "Arijit Singh", "Chinmayi" ] }
> db.Songdetails.find({"Music Director": "Illaiyaraaja", "Singer": "Unnikrishnan"}, {"Song Name": "1", _id: 0}).pretty()
{ "Song Name" : "Raja Poonthottam" }
> db.Songdetails.find({}, {"Singer": "1", _id: 0}).pretty()
{ "Singer" : "Shankar mahadevan" }
{ "Singer" : [ "Unnikrishnan", "Anuradha Sritam" ] }
{ "Singer" : [ "Haritharan", "Sadhana Sargam" ] }
{ "Singer" : "Haritharan" }
{ "Singer" : [ "Arijit Singh", "Chinmayi" ] }
> db.Songdetails.find({}, {"Singer": "1", _id: 0}).pretty()
{ "Singer" : "Shankar mahadevan" }
{ "Singer" : [ "Unnikrishnan", "Anuradha Sritam" ] }
{ "Singer" : [ "Haritharan", "Sadhana Sargam" ] }
{ "Singer" : "Haritharan" }
{ "Singer" : [ "Arijit Singh", "Chinmayi" ] }
```