

1.Create a Database called music.

> use music

2.Create a collection called songdetails.

> db.createCollection("songdetails")

```
ukistu07@ukipc07: ~$ sudo mongo
MongoDB shell version v3.6.9
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("15ed328d-6a4c-40fc-8508-c3ccd787c325") }
MongoDB server version: 3.6.9
Server has startup warnings:
2019-01-08T20:21:24.311+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the Wired
Tiger storage engine
2019-01-08T20:21:24.311+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not rec
ommended.
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.
2019-01-08T20:21:26.073+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
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, start the
server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten]
> use music
switched to db music
> db.createCollection("songdetails")
{ "ok" : 1 }
>
```

3. Create the above 5 song documents

```
> db.songdetails.insert({"Song  
Name":"thaniye  
Thananthaniye","Flim":"Rythm","Music  
Director":"A.R.Rahman","Singer":"Shank  
ar Mahadevan"})
```

```
> db.songdetails.insert({"Song  
Name":"Evano Oruvan","Flim":"Alai  
Payuthey","Music  
Director":"A.R.Rahman","Singer":"Swam  
alatha"})
```

```
> db.songdetails.insert({"Song  
Name":"Roja  
Poonthottam","Flim":"KannukKul  
Nilavu","Music
```

```
Director":"Illaiyaraaja","Singer":  
"Unnikrishnan,Anuradha Sriram"})
```

```
> db.songdetails.insert({"Song  
Name":"Vennilave Venillave  
Vennaithanandi","Fim":"Minsara  
Kanavu","Music  
Director":"A.R.Rahman","Singer":  
"Hariharan Sadhana Sargam"})
```

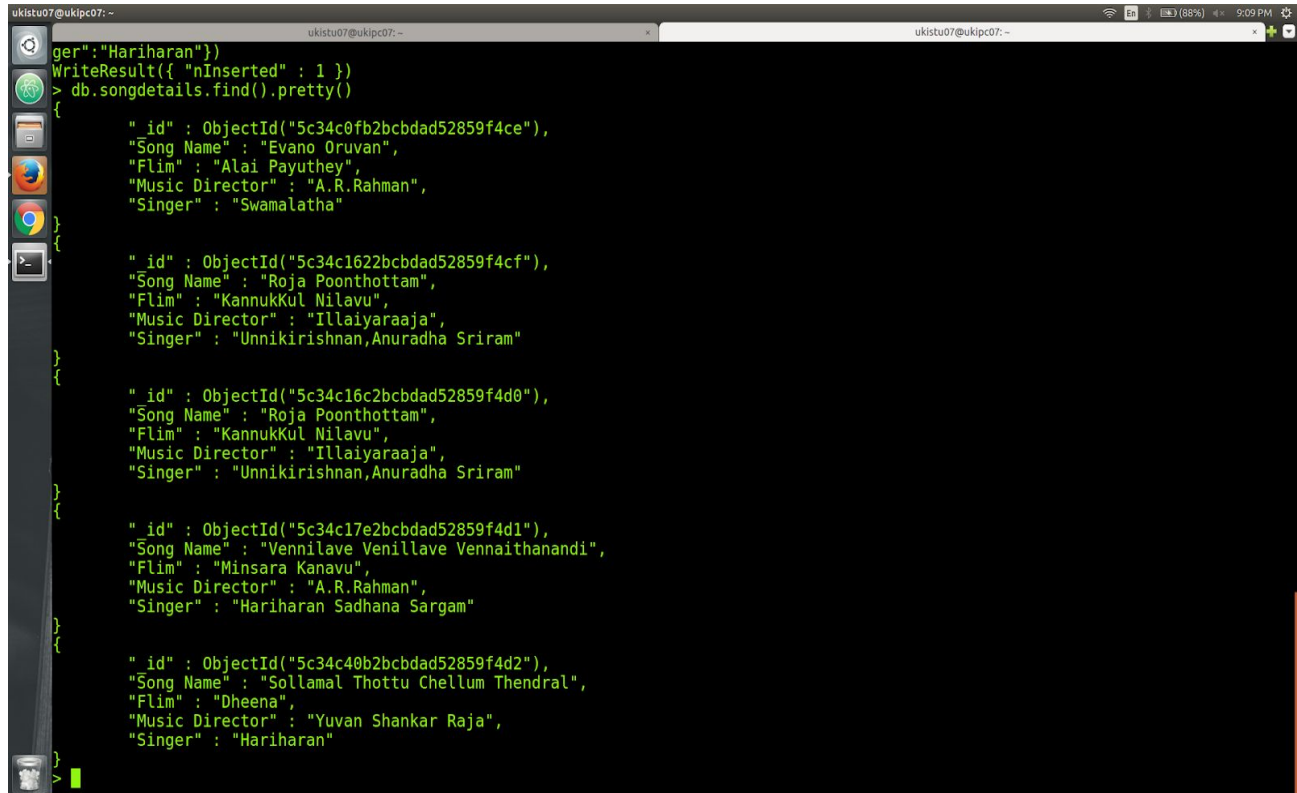
```
> db.songdetails.insert({"Song  
Name":"Roja  
Poonthottam","Film":"KannukKul  
Nilavu","Music  
Director":"Illaiyaraaja","Singer":  
["Unnikrishnan","Anuradha Sriram"]})
```

```
ukistu07@ukipc07: ~
ommeded.
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten]
2019-01-08T20:21:26.072+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify which IP
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --bind_ip_all
to
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, start the
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten]
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten]
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2019-01-08T20:21:26.073+0530 I CONTROL [initandlisten]
> use music
switched to db music
> db.createCollection("songdetails")
{ "ok" : 1 }
> db.songdetails.insert({"Song Name":"Thaniye Thananthaniye","Flim":"Rythm","Music Director":"A.R.Rahman","Singer":"Shankar Mahadeva
n"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({"Song Name":"Evano Oruvan","Flim":"Alai Payuthey","Music Director":"A.R.Rahman","Singer":"Swamalatha"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({"Song Name":"Roja Poonthottam","Flim":"KannukKul Nilavu","Music Director":"Illaiyaraaja","Singer":"Unnikir
ishnan,Anuradha Sriram"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({"Song Name":"Roja Poonthottam","Flim":"KannukKul Nilavu","Music Director":"Illaiyaraaja","Singer":"Unnikir
ishnan,Anuradha Sriram"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({"Song Name":"Vennilave Venillave Vennaithanandi","Flim":"Minsara Kanavu","Music Director":"A.R.Rahman","S
inger":"Hariharan Sadhana Sargam"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.insert({"Song Name":"Sollamal Thottu Chellum Thendral","Flim":"Dheena","Music Director":"Yuvan Shankar Raja","Sing
er":"Hariharan"})
...
...
> db.songdetails.insert({"Song Name":"Sollamal Thottu Chellum Thendral","Flim":"Dheena","Music Director":"Yuvan Shankar Raja","Sing
er":"Hariharan"})
...
...

```

4. List all documents created.

> db.songdetails.find().pretty()

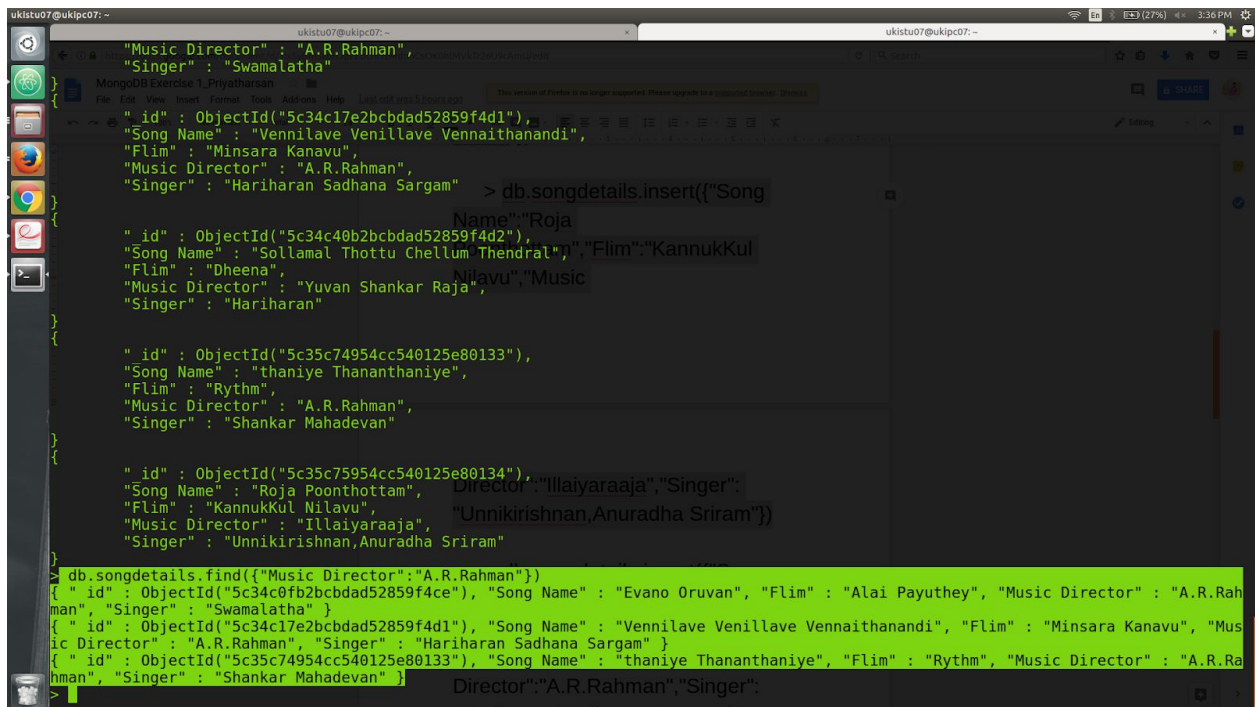
A terminal window with a dark background and light green text. The prompt is 'ukistu07@ukipc07: ~'. The command 'db.songdetails.find().pretty()' has been executed. The output is a JSON array of five song documents. Each document contains fields for 'id', 'Song Name', 'Flim', 'Music Director', and 'Singer'. The songs listed are 'Evano Oruvan', 'Roja Poonthottam', 'Roja Poonthottam', 'Vennilave Venillave Vennaithanandi', and 'Sollamal Thottu Chellum Thendral'.

```
ukistu07@ukipc07: ~
ger:"Hariharan"})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find().pretty()
{
  " id" : ObjectId("5c34c0fb2bcbdad52859f4ce"),
  "Song Name" : "Evano Oruvan",
  "Flim" : "Alai Payuthey",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Swamalatha"
}
{
  " id" : ObjectId("5c34c1622bcbdad52859f4cf"),
  "Song Name" : "Roja Poonthottam",
  "Flim" : "KannukKul Nilavu",
  "Music Director" : "Illaiyaraaja",
  "Singer" : "Unnikirishnan,Anuradha Sriram"
}
{
  " id" : ObjectId("5c34c16c2bcbdad52859f4d0"),
  "Song Name" : "Roja Poonthottam",
  "Flim" : "KannukKul Nilavu",
  "Music Director" : "Illaiyaraaja",
  "Singer" : "Unnikirishnan,Anuradha Sriram"
}
{
  " id" : ObjectId("5c34c17e2bcbdad52859f4d1"),
  "Song Name" : "Vennilave Venillave Vennaithanandi",
  "Flim" : "Minsara Kanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan Sadhana Sargam"
}
{
  " id" : ObjectId("5c34c40b2bcbdad52859f4d2"),
  "Song Name" : "Sollamal Thottu Chellum Thendral",
  "Flim" : "Dheena",
  "Music Director" : "Yuvan Shankar Raja",
  "Singer" : "Hariharan"
}
>
```

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

Method 1.

```
>db.songdetails.find({"Music  
Director":"A.R.Rahman"})
```

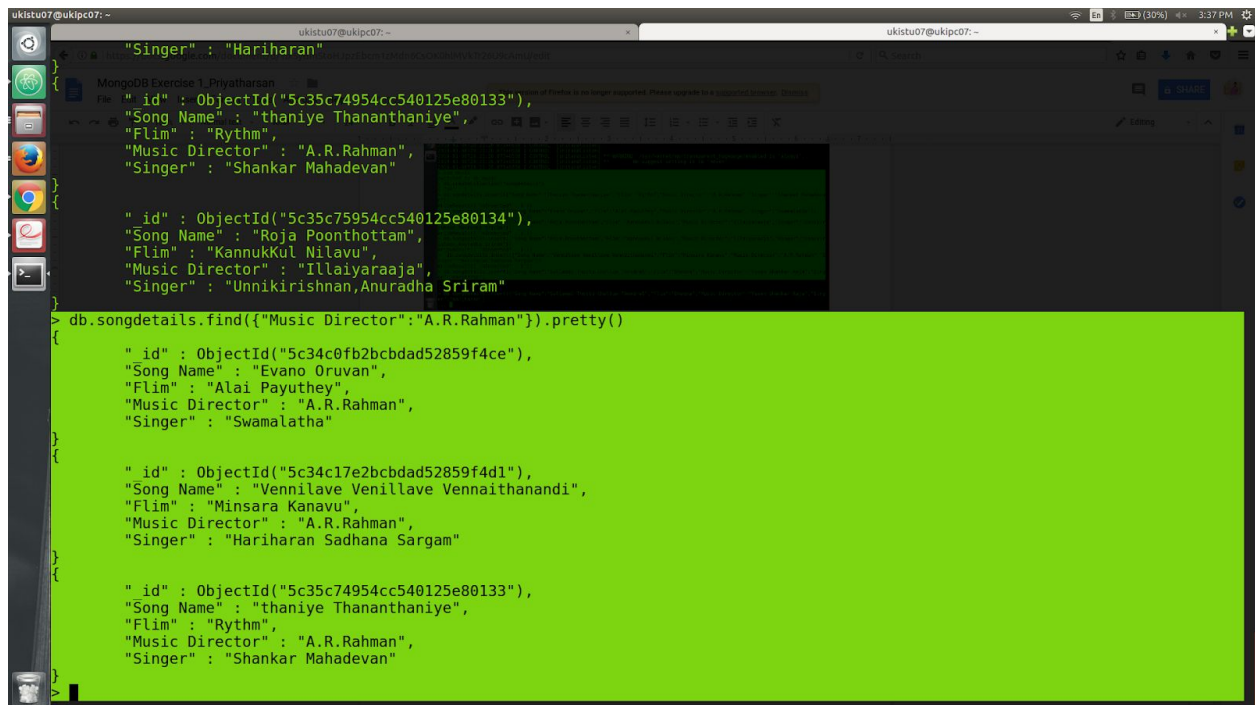


The screenshot shows a MongoDB command prompt window with the following content:

```
ukistu07@ukipc07: ~  
ukistu07@ukipc07: ~  
ukistu07@ukipc07: ~  
MongoDB Exercise 1_Priyatharan  
File Edit View Insert Format Tools Addons Help 1.0.0.0 (2015-08-04)  
This version of Firefox is no longer supported. Please upgrade to a supported version. Settings  
"Music Director" : "A.R.Rahman",  
"Singer" : "Swamalatha"  
" id" : ObjectId("5c34c17e2bcbdad52859f4d1"),  
"Song Name" : "Vennilave Venillave Vennaithanandi",  
"Flim" : "Minsara Kanavu",  
"Music Director" : "A.R.Rahman",  
"Singer" : "Hariharan Sadhana Sargam"  
> db.songdetails.insert({"Song  
Name":"Roja  
" id" : ObjectId("5c34c40b2bcbdad52859f4d2"),  
"Song Name" : "Sollamal Thottu Chellum Thendral",  
"Flim" : "Dheena",  
"Music Director" : "Yuvan Shankar Raja",  
"Singer" : "Hariharan"  
}  
}  
" id" : ObjectId("5c35c74954cc540125e80133"),  
"Song Name" : "thaniye Thananthaniye",  
"Flim" : "Rythm",  
"Music Director" : "A.R.Rahman",  
"Singer" : "Shankar Mahadevan"  
}  
}  
" id" : ObjectId("5c35c75954cc540125e80134"),  
"Song Name" : "Roja Poonthottam",  
"Flim" : "KannukKul Nilavu",  
"Music Director" : "Illaiyaraaja",  
"Singer" : "Unnikrishnan,Anuradha Sriram"  
}  
}  
> db.songdetails.find({"Music Director":"A.R.Rahman"})  
{ " id" : ObjectId("5c34c0fb2bcbdad52859f4ce"), "Song Name" : "Evano Oruvan", "Flim" : "Alai Payuthey", "Music Director" : "A.R.Rahman", "Singer" : "Swamalatha" }  
{ " id" : ObjectId("5c34c17e2bcbdad52859f4d1"), "Song Name" : "Vennilave Venillave Vennaithanandi", "Flim" : "Minsara Kanavu", "Music Director" : "A.R.Rahman", "Singer" : "Hariharan Sadhana Sargam" }  
{ " id" : ObjectId("5c35c74954cc540125e80133"), "Song Name" : "thaniye Thananthaniye", "Flim" : "Rythm", "Music Director" : "A.R.Rahman", "Singer" : "Shankar Mahadevan" }  
>
```

Method 2.

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

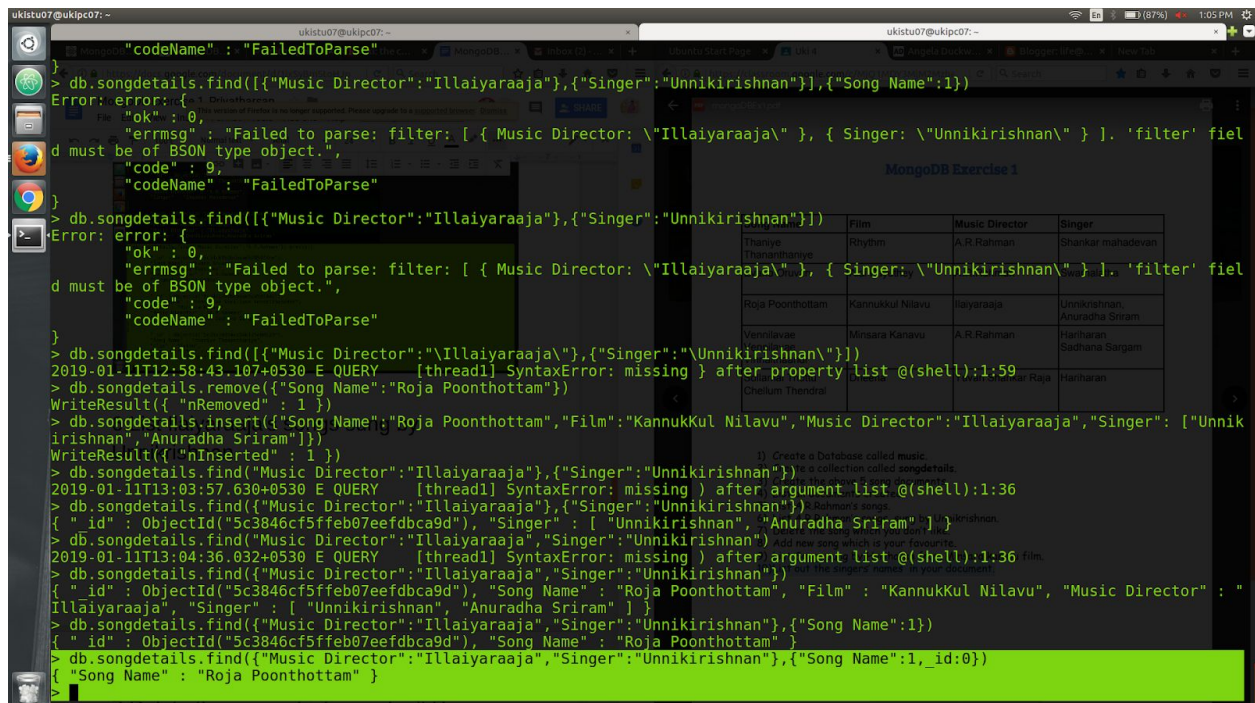


The screenshot shows a MongoDB GUI interface with a query result for the command `db.songdetails.find({"Music Director":"A.R.Rahman"}).pretty()`. The result is displayed in a green box and lists three songs:

```
{
  "_id" : ObjectId("5c34c0fb2bcbdad52859f4ce"),
  "Song Name" : "Evano Oruvan",
  "Flim" : "Alai Payuthey",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Swamalatha"
},
{
  "_id" : ObjectId("5c34c17e2bcbdad52859f4d1"),
  "Song Name" : "Vennilave Venillave Vennaithanandi",
  "Flim" : "Minsara Kanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan Sadhana Sargam"
},
{
  "_id" : ObjectId("5c35c74954cc540125e80133"),
  "Song Name" : "thaniye Thananthaniye",
  "Flim" : "Rythm",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Shankar Mahadevan"
}
```


6. List Ilaiyaraaja's songs sung by Unnikrishnan.

```
> db.songdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikirishnan"},{"Song Name":1})
```



The screenshot shows a terminal window with MongoDB commands and their output. The commands are as follows:

```
> db.songdetails.find({"Music Director":"Ilaiyaraaja"},{"Singer":"Unnikirishnan"},{"Song Name":1})
Error: error: {
  "ok" : 0,
  "errmsg" : "Failed to parse: filter: [ { Music Director: 'Ilaiyaraaja' }, { Singer: 'Unnikirishnan' } ]. 'filter' field must be of BSON type object.",
  "code" : 9,
  "codeName" : "FailedToParse"
}

> db.songdetails.find({"Music Director":"Ilaiyaraaja"},{"Singer":"Unnikirishnan"})
Error: error: {
  "ok" : 0,
  "errmsg" : "Failed to parse: filter: [ { Music Director: 'Ilaiyaraaja' }, { Singer: 'Unnikirishnan' } ]. 'filter' field must be of BSON type object.",
  "code" : 9,
  "codeName" : "FailedToParse"
}

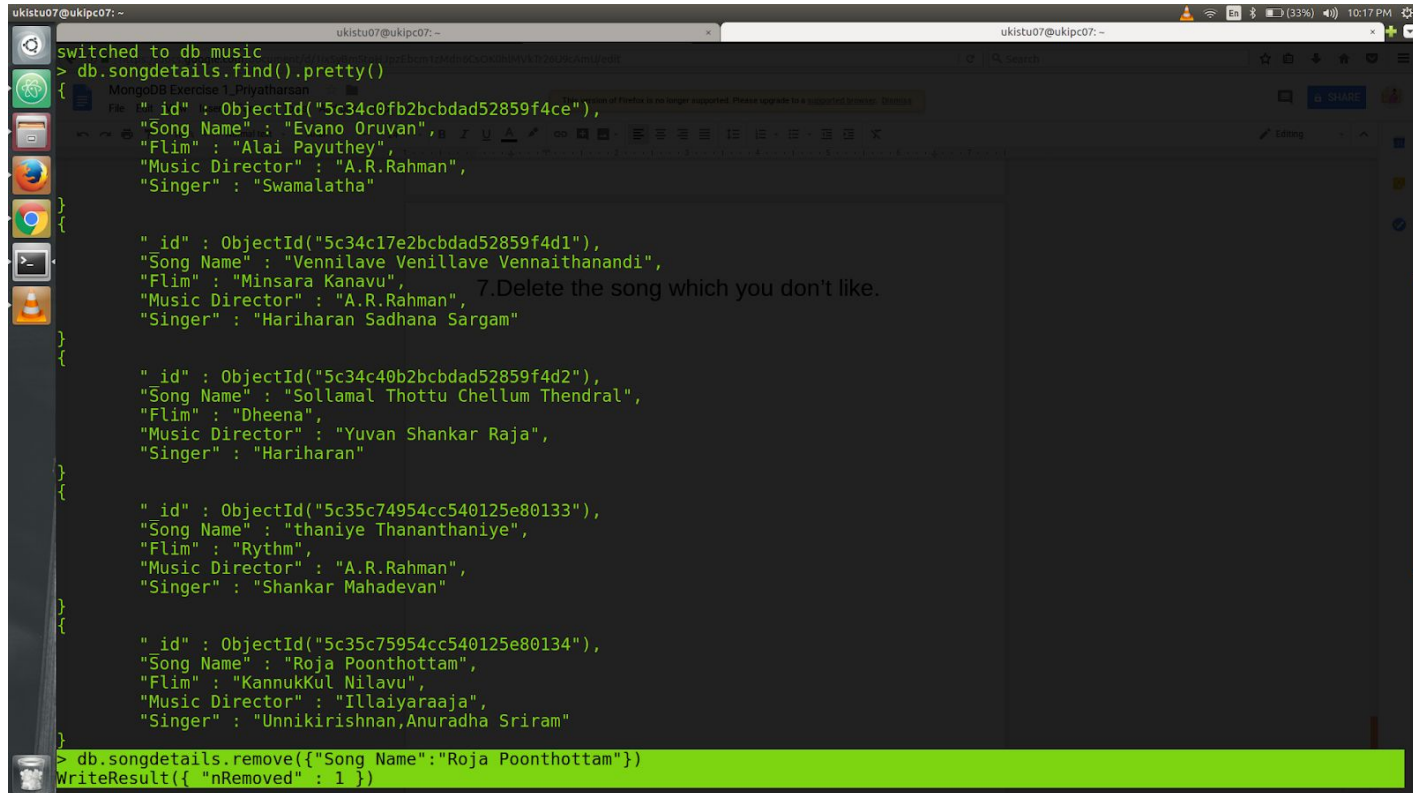
> db.songdetails.find({"Music Director":"Ilaiyaraaja"},{"Singer":"Unnikirishnan"})
2019-01-11T12:58:43.107+0530 E QUERY [thread1] SyntaxError: missing } after property list @(shell):1:59
> db.songdetails.remove({"Song Name":"Raja Poonthottam"})
WriteResult({ "nRemoved" : 1 })
> db.songdetails.insert({"Song Name":"Raja Poonthottam","Film":"Kannukkul Nilavu","Music Director":"Ilaiyaraaja","Singer":["Unnikirishnan","Anuradha Sriram"]})
WriteResult({ "nInserted" : 1 })
> db.songdetails.find("Music Director":"Ilaiyaraaja",{"Singer":"Unnikirishnan"})
2019-01-11T13:03:57.630+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:36
> db.songdetails.find({"Music Director":"Ilaiyaraaja"},{"Singer":"Unnikirishnan"})
{ "id" : ObjectId("5c3846cf5ffeb07eefdbca9d"), "Singer" : [ "Unnikirishnan", "Anuradha Sriram" ] }
> db.songdetails.find("Music Director":"Ilaiyaraaja","Singer":"Unnikirishnan")
2019-01-11T13:04:36.032+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:36
> db.songdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikirishnan"})
{ "id" : ObjectId("5c3846cf5ffeb07eefdbca9d"), "Song Name" : "Raja Poonthottam" }
> db.songdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikirishnan"},{"Song Name":1})
{ "id" : ObjectId("5c3846cf5ffeb07eefdbca9d"), "Song Name" : "Raja Poonthottam" }
> db.songdetails.find({"Music Director":"Ilaiyaraaja","Singer":"Unnikirishnan"},{"Song Name":1,_id:0})
{ "Song Name" : "Raja Poonthottam" }
```

On the right side of the terminal, there is a table titled "MongoDB Exercise 1" with the following data:

Singer	Film	Music Director	Singer
Shanmugam	Rhythm	A.R. Rahman	Shankar mahadevan
Raja Poonthottam	Kannukkul Nilavu	Ilaiyaraaja	Unnikrishnan, Anuradha Sriram
Venkatesh	Minsara Kanavu	A.R. Rahman	Harman, Sachana Sargam
Chellum Thendral			

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

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

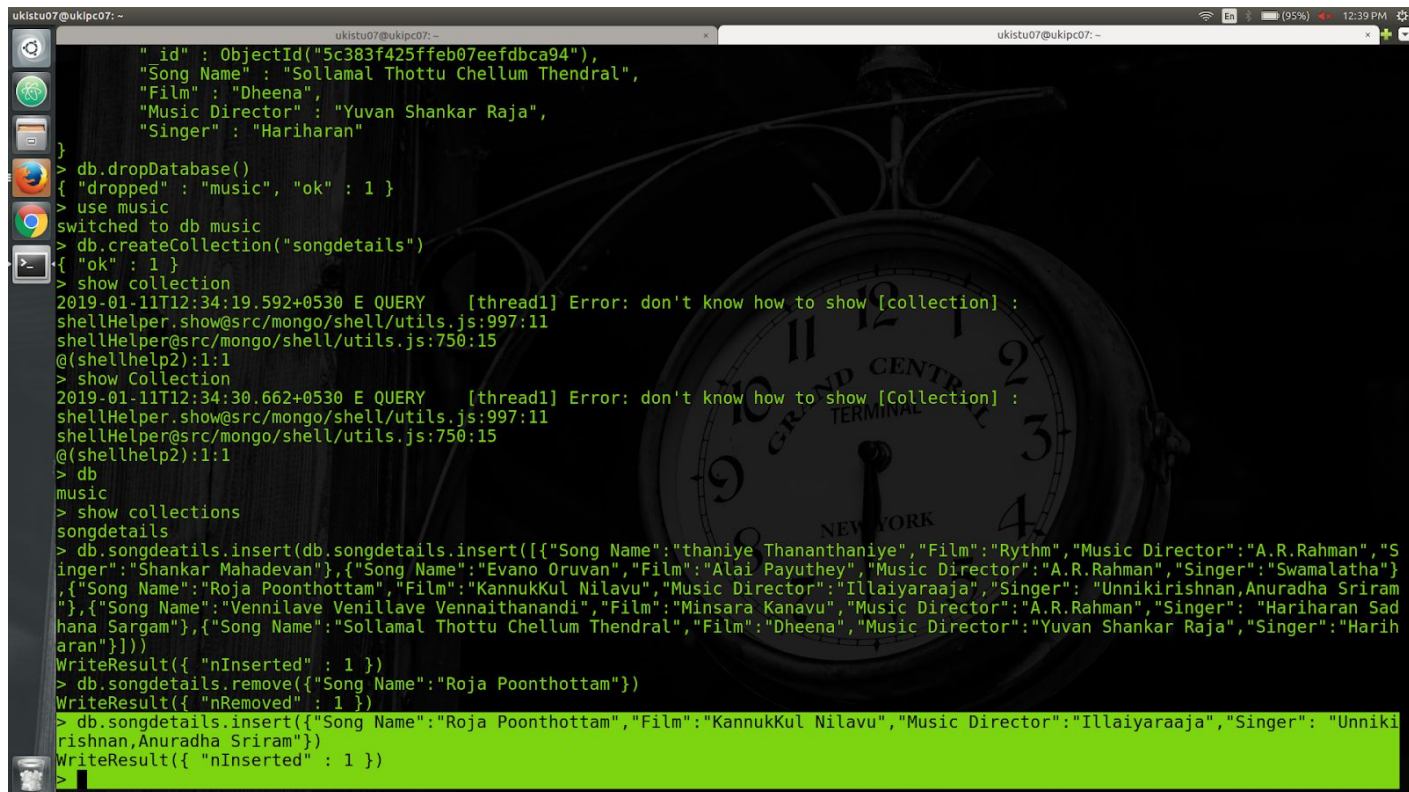


The screenshot shows a terminal window with a dark background. The prompt is 'ukistu07@ukipc07: ~'. The user has switched to the 'db music' database. They run the command 'db.songdetails.find().pretty()' which displays a list of five songs in JSON format. The songs are: 'Evano Oruvan', 'Vennilave Venillave Vennaithanandi', 'Minsara Kanavu', 'Sollamal Thottu Chellum Thendral', and 'thaniye Thananthaniye'. The song 'Roja Poonthottam' is also listed. The user then runs the command 'db.songdetails.remove({"Song Name":"Roja Poonthottam"})' and the output is 'WriteResult({ "nRemoved" : 1 })'.

```
ukistu07@ukipc07: ~
switched to db music
> db.songdetails.find().pretty()
{
  "_id" : ObjectId("5c34c0fb2bcbdad52859f4ce"),
  "Song Name" : "Evano Oruvan",
  "Flim" : "Alai Payuthey",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Swamalatha"
},
{
  "_id" : ObjectId("5c34c17e2bcbdad52859f4d1"),
  "Song Name" : "Vennilave Venillave Vennaithanandi",
  "Flim" : "Minsara Kanavu",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Hariharan Sadhana Sargam"
},
{
  "_id" : ObjectId("5c34c40b2bcbdad52859f4d2"),
  "Song Name" : "Sollamal Thottu Chellum Thendral",
  "Flim" : "Dheena",
  "Music Director" : "Yuvan Shankar Raja",
  "Singer" : "Hariharan"
},
{
  "_id" : ObjectId("5c35c74954cc540125e80133"),
  "Song Name" : "thaniye Thananthaniye",
  "Flim" : "Rythm",
  "Music Director" : "A.R.Rahman",
  "Singer" : "Shankar Mahadevan"
},
{
  "_id" : ObjectId("5c35c75954cc540125e80134"),
  "Song Name" : "Roja Poonthottam",
  "Flim" : "KannukKul Nilavu",
  "Music Director" : "Illaiyaraaja",
  "Singer" : "Unnikirishnan,Anuradha Sriram"
}
> db.songdetails.remove({"Song Name":"Roja Poonthottam"})
WriteResult({ "nRemoved" : 1 })
```

8. Add new song which is your favourite.

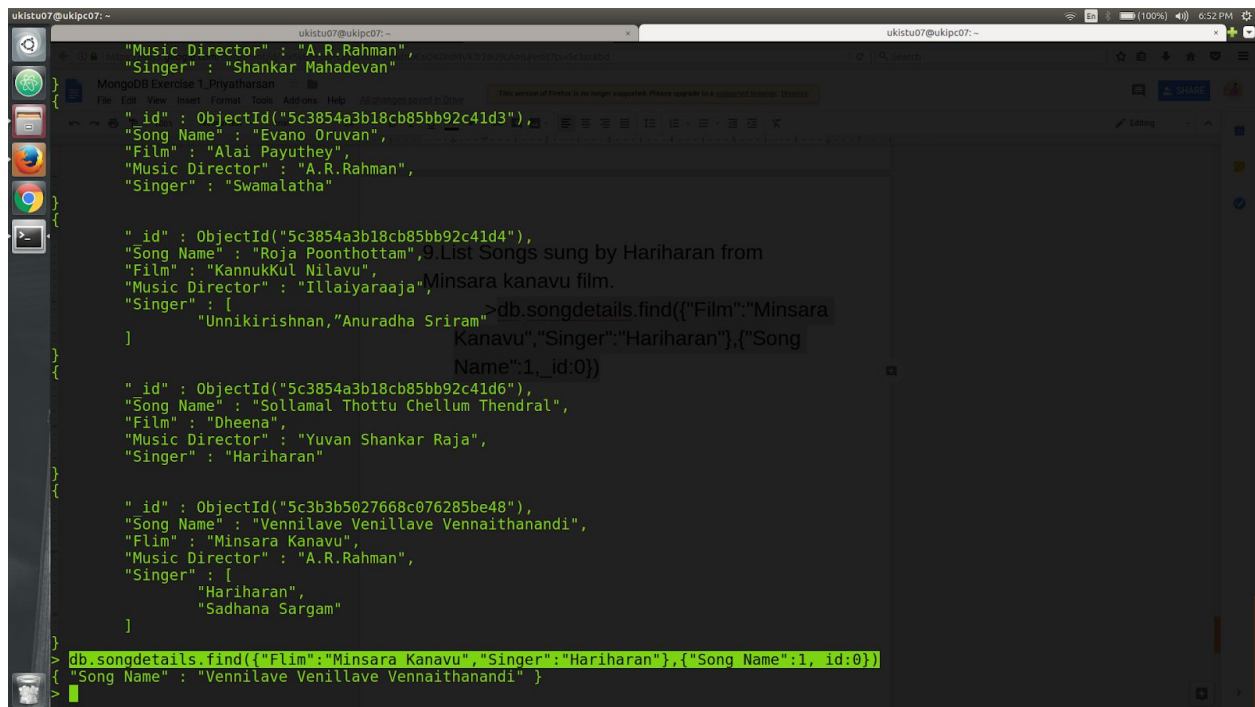
```
>db.songdetails.insert({"Song  
Name":"Roja  
Poonthottam","Film":"KannukKul  
Nilavu","Music  
Director":"Illaiyaraaja","Singer":  
"Unnikirishnan,Anuradha Sriram"})
```



```
ukistu07@ukipc07: ~  
ukistu07@ukipc07: ~  
ukistu07@ukipc07: ~  
" id" : ObjectId("5c383f425ffeb07eefdbca94"),  
"Song Name" : "Sollamal Thottu Chellum Thendral",  
"Film" : "Dheena",  
"Music Director" : "Yuvan Shankar Raja",  
"Singer" : "Hariharan"  
}  
> db.dropDatabase()  
{ "dropped" : "music", "ok" : 1 }  
> use music  
switched to db music  
> db.createCollection("songdetails")  
{ "ok" : 1 }  
> show collection  
2019-01-11T12:34:19.592+0530 E QUERY [thread1] Error: don't know how to show [collection] :  
shellHelper.show@src/mongo/shell/utils.js:997:11  
shellHelper@src/mongo/shell/utils.js:750:15  
@(shellhelp2):1:1  
> show Collection  
2019-01-11T12:34:30.662+0530 E QUERY [thread1] Error: don't know how to show [Collection] :  
shellHelper.show@src/mongo/shell/utils.js:997:11  
shellHelper@src/mongo/shell/utils.js:750:15  
@(shellhelp2):1:1  
> db  
music  
> show collections  
songdetails  
> db.songdeatils.insert(db.songdetails.insert([{"Song Name":"thaniye Thananthaniye","Film":"Rythm","Music Director":"A.R.Rahman","Singer":"Shankar Mahadevan"}, {"Song Name":"Evano Oruvan","Film":"Alai Payuthey","Music Director":"A.R.Rahman","Singer":"Swamalatha"}, {"Song Name":"Roja Poonthottam","Film":"KannukKul Nilavu","Music Director":"Illaiyaraaja","Singer": "Unnikirishnan,Anuradha Sriram"}, {"Song Name":"Vennilave Venillave Vennaithanandi","Film":"Minsara Kanavu","Music Director":"A.R.Rahman","Singer": "Hariharan Sadhana Sargam"}, {"Song Name":"Sollamal Thottu Chellum Thendral","Film":"Dheena","Music Director":"Yuvan Shankar Raja","Singer":"Hariharan"}]))  
WriteResult({ "nInserted" : 1 })  
> db.songdetails.remove({"Song Name":"Roja Poonthottam"})  
WriteResult({ "nRemoved" : 1 })  
> db.songdetails.insert({"Song Name":"Roja Poonthottam","Film":"KannukKul Nilavu","Music Director":"Illaiyaraaja","Singer": "Unnikirishnan,Anuradha Sriram"})  
WriteResult({ "nInserted" : 1 })  
>
```

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

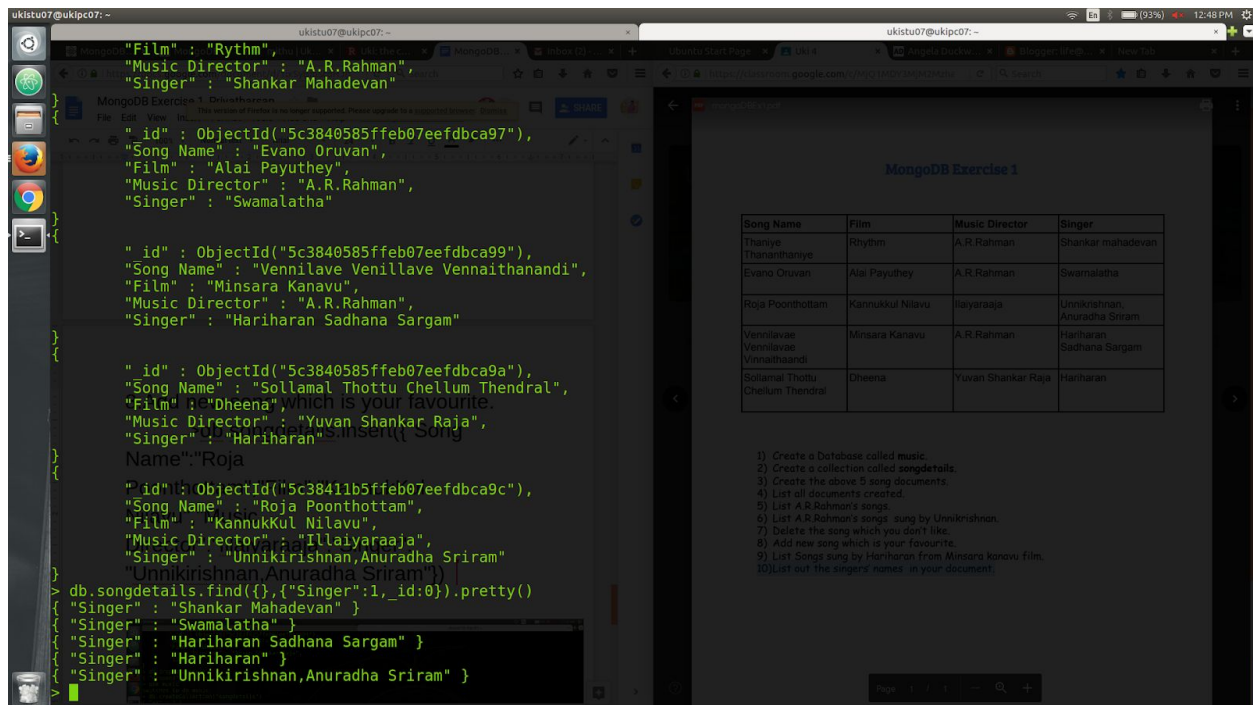
```
>db.songdetails.find({"Flim":"Minsara  
Kanavu","Singer":"Hariharan"},"Song  
Name":1,_id:0})
```



```
ukistu07@ukipc07: ~  
MongoDB Shell  
ukistu07@ukipc07: ~  
{"_id": "ObjectId('5c3854a3b18cb85bb92c41d3')",  
  "Song Name": "Evano Oruvan",  
  "Film": "Alai Payuthey",  
  "Music Director": "A.R. Rahman",  
  "Singer": "Swamalatha"  
}  
  
{"_id": "ObjectId('5c3854a3b18cb85bb92c41d4')",  
  "Song Name": "Raja Poonthottam",  
  "Film": "Kannukkul Nilavu",  
  "Music Director": "Illaiyaraaja",  
  "Singer": ["Unnikrishnan", "Anuradha Sriram"]  
}  
  
{"_id": "ObjectId('5c3854a3b18cb85bb92c41d6')",  
  "Song Name": "Sollamal Thottu Chellum Thendral",  
  "Film": "Dheena",  
  "Music Director": "Yuvan Shankar Raja",  
  "Singer": "Hariharan"  
}  
  
{"_id": "ObjectId('5c3b3b5027668c076285be48')",  
  "Song Name": "Vennilave Venillave Vennaithanandi",  
  "Film": "Minsara Kanavu",  
  "Music Director": "A.R. Rahman",  
  "Singer": ["Hariharan", "Sadhana Sargam"]  
}  
  
> db.songdetails.find({"Film": "Minsara Kanavu", "Singer": "Hariharan"}, {"Song Name": 1, _id: 0})  
{  
  "Song Name": "Vennilave Venillave Vennaithanandi"  
}
```

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

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



The screenshot shows a MongoDB command prompt on the left and a web browser on the right. The command prompt displays the output of the command `db.songdetails.find({}, {"Singer":1, "_id":0}).pretty()`, which lists the names of the singers for each song. The web browser shows a table titled "MongoDB Exercise 1" with columns for Song Name, Film, Music Director, and Singer. The table contains 10 rows of data, including songs like "Thaniye", "Evano Oruvan", "Raja Poonthottam", "Vennilave", "Sollamal Thottu Chellum Thendral", "Kannukkul Nilavu", "Minsara Kanavu", "Dheena", "Rajathi", and "Unnikirishnan, Anuradha Sriram".

Song Name	Film	Music Director	Singer
Thaniye	Rhythm	A.R.Rahman	Shankar mahadevan
Thaniye	Rhythm	A.R.Rahman	Shankar mahadevan
Evano Oruvan	Alai Payuthey	A.R.Rahman	Swamalatha
Raja Poonthottam	Kannukkul Nilavu	Isayaraaja	Unnikrishnan, Anuradha Sriram
Vennilave	Minsara Kanavu	A.R.Rahman	Hariharan, Anuradha Sriram
Vennilave	Minsara Kanavu	A.R.Rahman	Hariharan, Anuradha Sriram
Sollamal Thottu Chellum Thendral	Dheena	Yuvan Shankar Raja	Hariharan
Rajathi	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram
Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram
Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram	Unnikirishnan, Anuradha Sriram