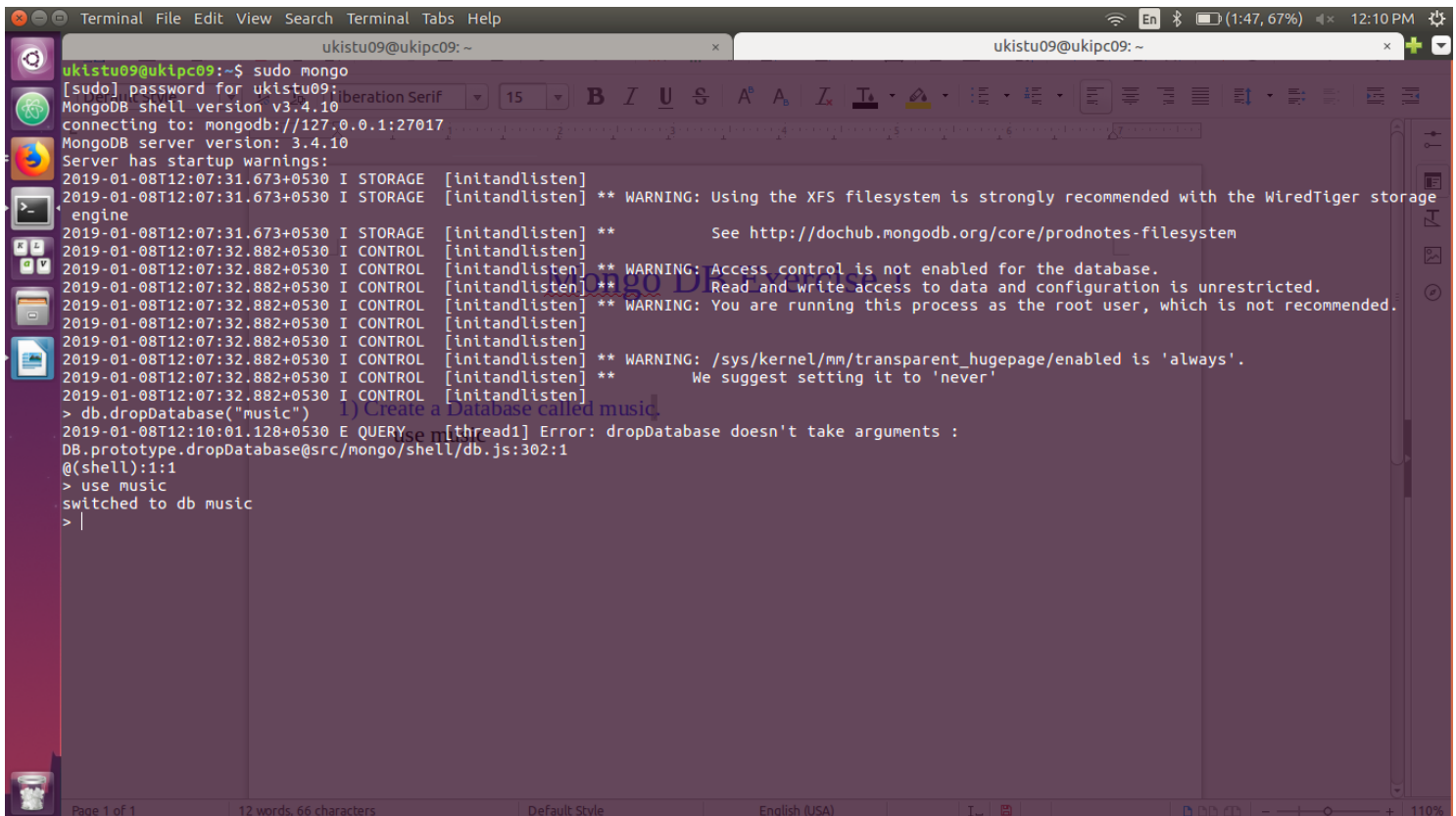


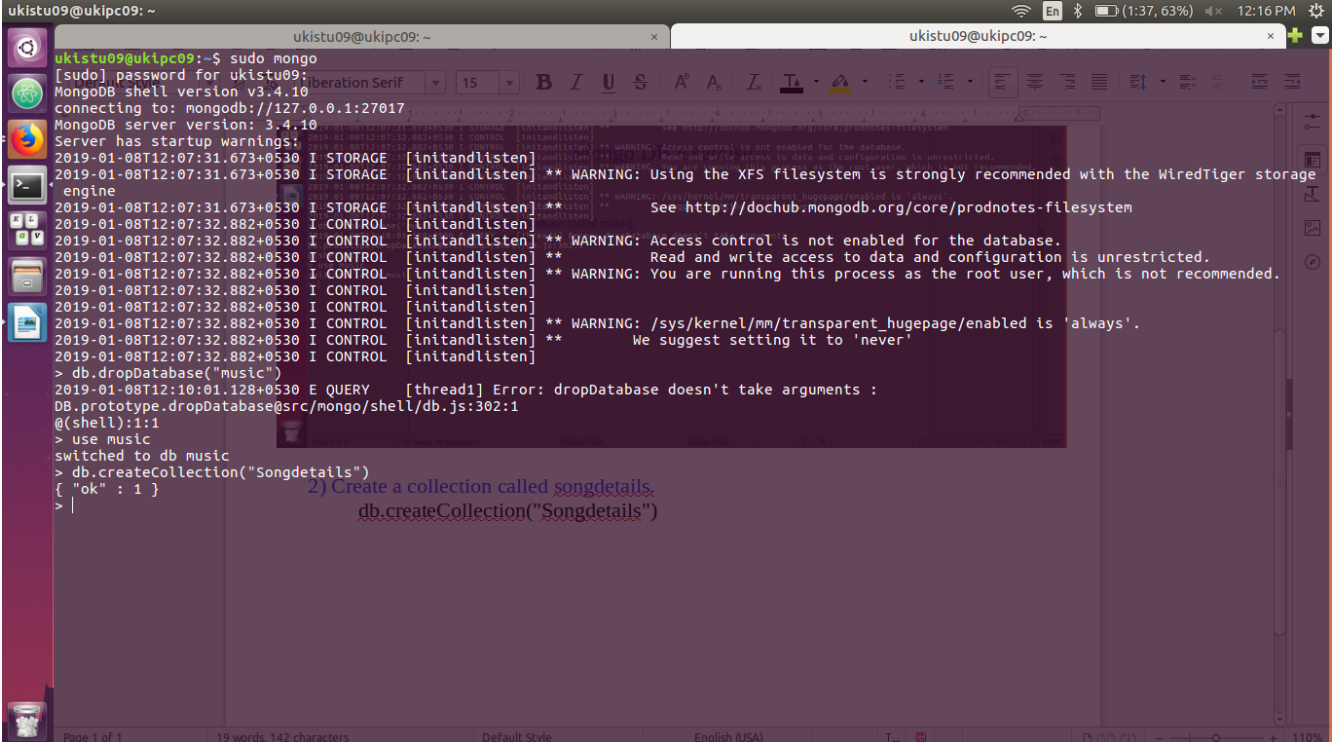
# Mongo DB Exercise 1

- 1) Create a Database called music.  
use music



```
ukistu09@ukipc09:~$ sudo mongo
[sudo] password for ukistu09:
MongoDB shell version v3.4.10
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.10
Server has startup warnings:
2019-01-08T12:07:31.673+0530 I STORAGE [initandlisten]
2019-01-08T12:07:31.673+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2019-01-08T12:07:31.673+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten]
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] Read and write access to data and configuration is unrestricted.
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten]
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten]
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] We suggest setting it to 'never'
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten]
> db.dropDatabase("music")
2019-01-08T12:10:01.128+0530 E QUERY [thread1] Error: dropDatabase doesn't take arguments :
DB.prototype.dropDatabase@src/mongo/shell/db.js:302:1
@(shell):1:1
> use music
switched to db music
> |
```

2) Create a collection called songdetails.  
`db.createCollection("Songdetails")`



```
ukistu09@ukipc09: ~  
ukistu09@ukipc09: ~  
ukistu09@ukipc09:~$ sudo mongo  
[sudo] password for ukistu09:  
MongoDB shell version v3.4.10  
connecting to: mongodb://127.0.0.1:27017  
MongoDB server version: 3.4.10  
Server has startup warnings:  
2019-01-08T12:07:31.673+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine  
2019-01-08T12:07:31.673+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] **  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2019-01-08T12:07:32.882+0530 I CONTROL [initandlisten]  
> db.dropDatabase("music")  
2019-01-08T12:10:01.128+0530 E QUERY [thread1] Error: dropDatabase doesn't take arguments :  
DB.prototype.dropDatabase@src/mongo/shell/db.js:302:1  
@(shell):1:1  
> use music  
switched to db music  
> db.createCollection("Songdetails")  
{ "ok" : 1 }  
> |
```

2) Create a collection called songdetails.  
`db.createCollection("Songdetails")`

Page 1 of 1 19 words, 142 characters Default Style English (USA) 110%

### 3) Create the above 5 song documents.

```
db.Songdetails.insert({"Song Name":"Thaniye Thananthaniye",  
"Film":"Rhythm" , "Music Director" : "A.R.Rahman", "Singer":"Shankar  
mahadevan"})
```

```
db.Songdetails.insert({"Song Name":"Evano Oruvan", "Film":"Alai  
Payuthey" , "Music Director" : "A.R.Rahman", "Singer":"Swarnalatha"})
```

```
db.Songdetails.insert({"Song Name":"Roja Poonthottam",  
"Film":"Kannukkul Nilavu" , "Music Director" : "Illaiyaraaja",  
"Singer":"Unnikrishnan,Anuradha Sriram"})
```

```
db.Songdetails.insert({"Song Name":"Vennilavae Vennilavae  
Vinnaihaandi", "Film":"Minsara Kanavu" , "Music Director" : "A.R.Rahman",  
"Singer":"Hariharan,Sadhana Sargam"})
```

```
db.Songdetails.insert({"Song Name":"Sollamal Thottu Chellum Thendral",  
"Film":"Dheena" , "Music Director" : "Yuvan Shankar Raja",  
"Singer":"Hariharan"})
```

The screenshot shows a document titled "MongoDB Exercise 1" with a table of song details and a list of commands to create and manage these documents in a MongoDB database.

| 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           | Yuvan Shankar Raja | Hariharan                     |

Commands:

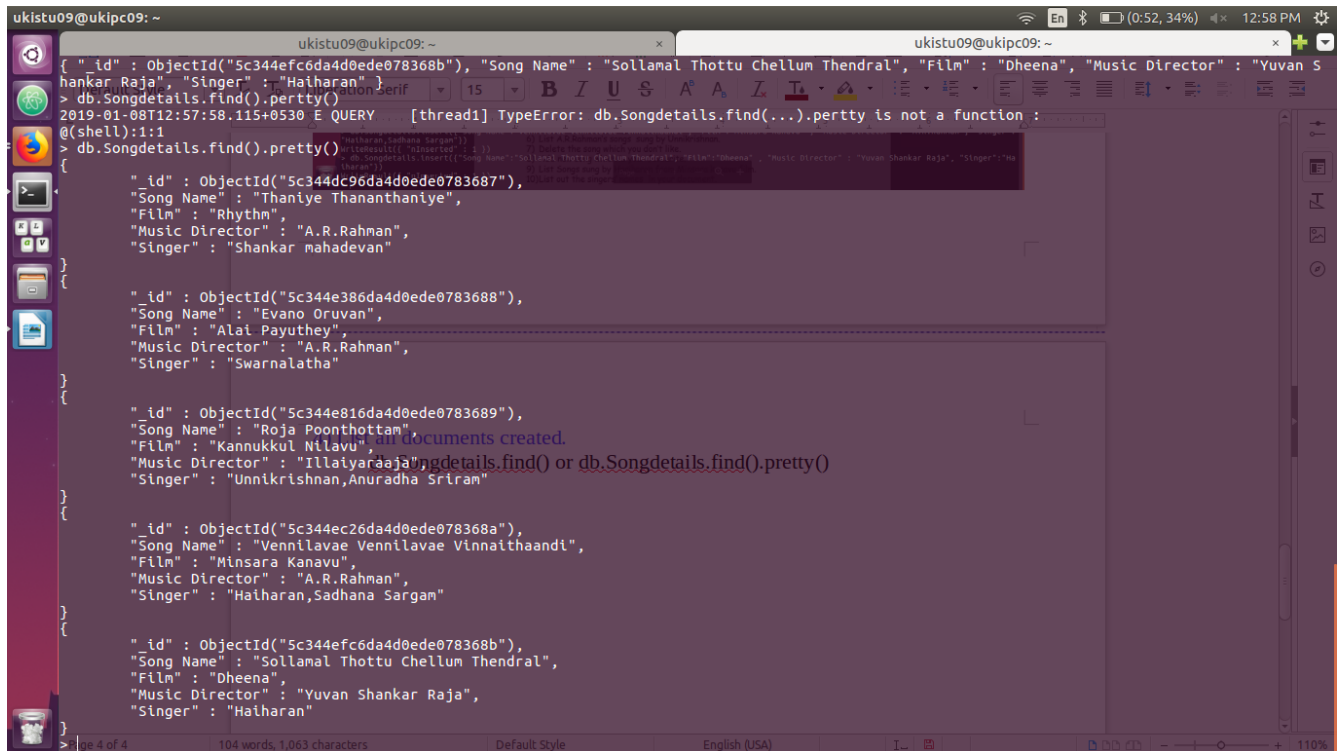
```
> db.Songdetails.insert({"Song Name":"Thaniye Thananthaniye", "Film":"Rhythm" , "Music Director" : "A.R.Rahman", "Singer":"Shankar mahadevan"})  
WriteResult({ "nInserted" : 1 })  
> db.Songdetails.insert({"Song Name":"Evano Oruvan", "Film":"Alai Payuthey" , "Music Director" : "A.R.Rahman", "Singer":"Swarnalatha"})  
WriteResult({ "nInserted" : 1 })  
> db.Songdetails.insert({"Song Name":"Roja Poonthottam", "Film":"Kannukkul Nilavu" , "Music Director" : "Illaiyaraaja", "Singer":"Unnikrishnan,A  
nuradha Sriram"})  
WriteResult({ "nInserted" : 1 })  
> db.Songdetails.insert({"Song Name":"Vennilavae Vennilavae Vinnaihaandi", "Film":"Minsara Kanavu" , "Music Director" : "A.R.Rahman", "Singer":  
"Hariharan, Sadhana Sargam"})  
WriteResult({ "nInserted" : 1 })  
> db.Songdetails.insert({"Song Name":"Sollamal Thottu Chellum Thendral", "Film":"Dheena" , "Music Director" : "Yuvan Shankar Raja", "Singer":"Ha  
riharan"})  
WriteResult({ "nInserted" : 1 })
```

Tasks:

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

#### 4) List all documents created.

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



```
ukistu09@ukipc09: ~  
ukistu09@ukipc09: ~  
{ "id" : ObjectId("5c344efc6da4d0ede078368b"), "Song Name" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "Music Director" : "Yuva S  
hankar Raja", "Singer" : "Hatharan",  
> db.Songdetails.find().pretty()  
2019-01-08T12:57:58.115+0530 E QUERY [thread1] TypeError: db.Songdetails.find(...).pretty is not a function: ...  
@shell:1:1  
> db.Songdetails.find().pretty()  
{  
  "id" : ObjectId("5c344dc96da4d0ede0783687"),  
  "Song Name" : "Thaniye Thananthaniye",  
  "Film" : "Rhythm",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Shankar mahadevan"  
}  
{  
  "id" : ObjectId("5c344e386da4d0ede0783688"),  
  "Song Name" : "Evano Oruvan",  
  "Film" : "Alai Payuthey",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Swarnalatha"  
}  
{  
  "id" : ObjectId("5c344e816da4d0ede0783689"),  
  "Song Name" : "Raja Poonthottam",  
  "Film" : "Kannukkul Nilavu",  
  "Music Director" : "Ilaiyaraaja",  
  "Singer" : "Unnikrishnan,Anuradha Sritam"  
}  
{  
  "id" : ObjectId("5c344ec26da4d0ede078368a"),  
  "Song Name" : "Vennilavae Vennilavae Vinnaihaandi",  
  "Film" : "Minsara Kanavu",  
  "Music Director" : "A.R.Rahman",  
  "Singer" : "Hatharan,Sadhana Sargam"  
}  
{  
  "id" : ObjectId("5c344efc6da4d0ede078368b"),  
  "Song Name" : "Sollamal Thottu Chellum Thendral",  
  "Film" : "Dheena",  
  "Music Director" : "Yuva Shankar Raja",  
  "Singer" : "Hatharan"  
}  
> |  
page 4 of 4 104 words, 1,063 characters Default Style English (USA) 110%
```

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

```
db.Songdetails.find().sort({"A.R.Rahman": 1 })
```

The screenshot shows a web browser window displaying a Stack Overflow page. The browser's address bar shows the URL: `https://stackoverflow.com/questions/33018048/how-does-mongodb-order-their-docs-in-...`. The page title is "MongoDB does not 'order' the documents at all, unless you ask it to." The question, asked by user "lonica Bizau", is answered by user "Paola G.". The answer explains that MongoDB does not order documents by default and provides a code snippet to sort by the "\_id" field.

**Question:** MongoDB does not "order" the documents at all, unless you ask it to.

**Answer:** The basic insertion will create an `ObjectId` in the `_id` primary key value unless you tell it to do otherwise. This `ObjectId` value is a special value with "monotonic" or "ever increasing" properties, which means each value created is guaranteed to be larger than the last.

If you want "sorted" then do an explicit "sort":

```
db.collection.find().sort({ "_id": 1 })
```

Or a "natural" sort means in the order stored on disk:

```
db.Songdetails.find().sort({"A.R.Rahman": 1 })
```

The answer also includes a list of songs by A.R. Rahman, sorted by their "\_id" field:

```
{ "id" : ObjectId("5c344dc96da4d0ede0783687"), "Song Name" : "Thaniye Thananthaniye", "Film" : "Rhythm", "Music Director" : "A.R.Rahman", "Singer" : "Shankar mahadevan" }
{ "id" : ObjectId("5c344e386da4d0ede0783688"), "Song Name" : "Evano Oruvan", "Film" : "Alai Payuthey", "Music Director" : "A.R.Rahman", "Singer" : "Swarnalatha" }
{ "id" : ObjectId("5c344e816da4d0ede0783689"), "Song Name" : "Raja Poonthottam", "Film" : "Kannukkul Nilavu", "Music Director" : "A.R.Rahman", "Singer" : "Unnikrishnan, Anuradha Srinam" }
{ "id" : ObjectId("5c344ec26da4d0ede078368a"), "Song Name" : "Vennilavae Vennilavae Vinnaitaandi", "Film" : "Minsara Kanavu", "Music Director" : "A.R.Rahman", "Singer" : "Hatharan, Sadhana Sargam" }
{ "id" : ObjectId("5c344efc6da4d0ede078368b"), "Song Name" : "Sollamal Thottu Chellum Thendral", "Film" : "Dheena", "Music Director" : "Yuvan Shankar Raja", "Singer" : "Hatharan" }
```