

Answers for MongoDB questions

1. And 2.

```
ukistu06@UkiPc06: ~
2020 03:23:25 UkiPc06 systemd[1]: Started MongoDB Database Server.
ukistu06@UkiPc06: ~$ mongo
MongoDB shell version v4.2.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "_id" : UUID("77492b31-68c5-4d7f-b038-62681da07ae8") }
MongoDB server version: 4.2.2
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2020-01-08T03:23:25.324+0530 I STORAGE [initandlisten] MongoDB Exercise 1
2020-01-08T03:23:25.324+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-01-08T03:23:32.110+0530 I CONTROL [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten]
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten]
MongoDB Enterprise > dbs
2020-01-08T03:23:55.809+0530 E QUERY [js] uncaught exception: ReferenceError: dbs is not defined :
@(shell):1:1
MongoDB Enterprise > show dbs
2020-01-08T03:24:00.013+0530 E QUERY [js] uncaught exception: ReferenceError: show dbs is not defined :
@(shell):1:1
MongoDB Enterprise > show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
MongoDB Enterprise > use music
switched to db music
MongoDB Enterprise > db.createCollection('songdetails')
{ "ok" : 1 }
MongoDB Enterprise > show collections
1) Create a Database called music
2) Create a collection
songdetails
MongoDB Enterprise > 
3) Create the above 5
4) List all documents created
```

3.

```
ukistu06@UkiPc06: ~
2020-01-08T03:23:25.324+0530 I STORAGE [initandlisten] ** warning Access control is not enabled for the database. See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-01-08T03:23:32.110+0530 I CONTROL [initandlisten]
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten]
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2020-01-08T03:23:32.111+0530 I CONTROL [initandlisten]
MongoDB Enterprise > dbs
2020-01-08T03:23:55.809+0530 E QUERY [js] uncaught exception: ReferenceError: dbs is not defined :
@(shell):1:1
MongoDB Enterprise > show dbs
2020-01-08T03:24:00.013+0530 E QUERY [js] uncaught exception: ReferenceError: show dbs is not defined :
@(shell):1:1
MongoDB Enterprise > show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
MongoDB Enterprise > use music
switched to db music
MongoDB Enterprise > db.createCollection('songdetails')
{ "ok" : 1 }
MongoDB Enterprise > show collections
songdetails
MongoDB Enterprise > db.songdetails.insert({ "Song Name": "Thaniye Thananthathaniya", "Film": "Rhythm", "Music Director": "A.R.Rahman", "Singer": "Shankar Mahadevan" })
2020-01-08T05:47:37.018+0530 E QUERY [js] uncaught exception: SyntaxError: " " literal not terminated before end of script :
@(shell):1:139
MongoDB Enterprise > db.songdetails.insert({ "Song Name": "Thaniye Thananthathaniya", "Film": "Rhythm", "Music Director": "A.R.Rahman", "Singer": "Shankar Mahadevan" })
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise > db.songdetails.find()
{ "_id" : ObjectId("5e5faa6ae0d0be719e1651"), "Song Name" : "Thaniye Thananthathaniya", "Film" : "Rhythm", "Music Director" : "A.R.Rahman", "Singer" : "Shankar Mahadevan" }
MongoDB Enterprise > db.songdetails.insert({ "Song Name": "Evano Oruvan", "Film": "Alai Payuthey", "Music Director": "A.R.Rahman", "Singer": "Swamalatha" })
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise > db.songdetails.insert({ "Song Name": "Roja Poonthoodam", "Film": "Kannukul Nilavu", "Music Director": "Illayaraaja", "Singer": ["Unnikrishnan", "Anuradha Sriram"] })
WriteResult({ "nInserted" : 1 })
3) Create the above 5
MongoDB Enterprise > 
4) List all documents created
```

4.

ukistu06@UkiPc06:~

```
MongoDB Enterprise > db.songdetails.find().pretty()
{
    "id" : ObjectId("5e151faa6ae6d0be719e1651"),
    "Song Name" : "Thantyea Thananthathanyea",
    "Film" : "Rhythm",
    "Music Director" : "A.R.Rahman",
    "Singer" : "ShankarMahadevan"
}
{
    "id" : ObjectId("5e15200d6ae6d0be719e1652"),
    "Song Name" : "Evano Oruvan",
    "Film" : "Alai Payuthey",
    "Music Director" : "A.R.Rahman",
    "Singer" : "Swamalatha"
}
{
    "id" : ObjectId("5e1520a06ae6d0be719e1653"),
    "Song Name" : "Roja Poonthoodam",
    "Film" : "Kannukul Nilavu",
    "Music Director" : "Illeyaraaja",
    "Singer" : [
        "Unnikrishnan",
        "Anuradha Sriram"
    ]
}
{
    "id" : ObjectId("5e1521116ae6d0be719e1654"),
    "Song Name" : "Vennilavae Vennilvae Vennaithaandi",
    "Film" : "Minsara Kanavu",
    "Music Director" : "A.R.Rahman",
    "Singer" : [
        "Haritharan",
        "Sadhana Sargam"
    ]
}
{
    "id" : ObjectId("5e1521680ae6d0be719e1655"),
    "Song Name" : "Sollamal Thottu Chelum Thendral",
    "Film" : "Dheena",
    "Music Director" : "Yuvan Shankar Raja",
    "Singer" : "Haritharan"
}
MongoDB Enterprise >
```

MongoDB Exercise 1

Song Name	Film	Music Director	Singer
Thantyea Thananthathanyea	Rhythm	A.R.Rahman	Shankar Mahadevan
Evano Oruvan	Alai Payuthey	A.R.Rahman	Swamalatha
Kannukul Nilavu		Illeyaraaja	Unnikrishnan, Anuradha Sriram
Vennilavae Vennilvae Vennaithaandi	Minsara Kanavu	A.R.Rahman	Haritharan, Sadhana Sargam
Sollamal Thottu Chelum Thendral	Dheena	Yuvan Shankar Raja	Haritharan

2) Create a collection
3) Create the above 5 documents
4) List all documents created

5.

ukistu06@UkiPc06:~

```
MongoDB Enterprise > db.songdetails.find({"Music Director": "A.R.Rahman"}).pretty()
{
    "id" : ObjectId("5e151faa6ae6d0be719e1651"),
    "Song Name" : "Thantyea Thananthathanyea",
    "Film" : "Rhythm",
    "Music Director" : "A.R.Rahman",
    "Singer" : "ShankarMahadevan"
}
{
    "id" : ObjectId("5e15200d6ae6d0be719e1652"),
    "Song Name" : "Evano Oruvan",
    "Film" : "Alai Payuthey",
    "Music Director" : "A.R.Rahman",
    "Singer" : "Swamalatha"
}
{
    "id" : ObjectId("5e1521116ae6d0be719e1654"),
    "Song Name" : "Vennilavae Vennilvae Vennaithaandi",
    "Film" : "Minsara Kanavu",
    "Music Director" : "A.R.Rahman",
    "Singer" : [
        "Haritharan",
        "Sadhana Sargam"
    ]
}
{
    "id" : ObjectId("5e1521680ae6d0be719e1655"),
    "Song Name" : "Sollamal Thottu Chelum Thendral",
    "Film" : "Dheena",
    "Music Director" : "Yuvan Shankar Raja",
    "Singer" : "Haritharan"
}
MongoDB Enterprise >
```

Analyze Query Performance

The cursor.explain("executionStats") and the db.collection.explain("executionStats") methods provide statistics about the performance of a query. This data output can be useful in measuring if and how a query uses an index.

db.collection.explain() provides information on the execution of other operations, such as db.collection.update(). See db.collection.explain() for details.

Evaluate the Performance of a Query

Consider a collection inventory with the following documents:

```
5e151faa6ae6d0be719e1651: { "_id": "5e151faa6ae6d0be719e1651", "item": "Milk", "type": "food", "quantity": 500 },
5e151faa6ae6d0be719e1652: { "_id": "5e151faa6ae6d0be719e1652", "item": "Butter", "type": "food", "quantity": 100 },
5e151faa6ae6d0be719e1654: { "_id": "5e151faa6ae6d0be719e1654", "item": "Oil", "type": "food", "quantity": 200 },
5e151faa6ae6d0be719e1655: { "_id": "5e151faa6ae6d0be719e1655", "item": "Salt", "type": "food", "quantity": 100 }
```

YES NO

6.

The screenshot shows a Linux desktop environment with a terminal window at the top and a browser window below it. The terminal window has the command 'MongoDB Enterprise > db.songdetails.find({\$and:[{"Music Director":"A.R.Rahman"}, {"Singer":"Unnikrishnan"}]}).pretty()' entered. The browser window displays the MongoDB Documentation for 'Specify Conditions Using Query Operators'. The page includes a sidebar with 'MONGODB MANUAL' and 'Version 4.2 (current)' sections. The main content area shows a query example: 'db.inventory.find({ status: { \$in: ["A", "D"] } })'. A note states: 'Although you can express this query using the \$or operator, use the \$in operator rather than the \$or operator when performing equality checks on the same field.' At the bottom, there's a note about the operation corresponding to an SQL statement and a 'View this page offline?' link.

7.

The screenshot shows a Linux desktop environment with a terminal window. The terminal history includes:

- Creation of a database: 'root@UkiPc06: ~
- Creation of a collection: 'use music'
- Insertion of documents into the 'songdetails' collection:
 - '{"id": ObjectId("5e151faa6ae6d0be719e1651"), "Song Name": "Thantyea Thananthathantyea", "Film": "Rhythm", "Music Director": "A.R.Rahman", "Singer": "Shankar Mahadevan"}'
 - '{"id": ObjectId("5e15200d6ae6d0be719e1652"), "Song Name": "Evano Oruvan", "Film": "Alai Payuthey", "Music Director": "A.R.Rahman", "Singer": "Swamalatha"}'
- Displaying the collection contents:

	Dheena	Yuvan Shankar Raja	Hariharan
"_id": ObjectId("5e15200d6ae6d0be719e1651")			
"_id": ObjectId("5e15200d6ae6d0be719e1652")			

- List of tasks:
 - Create a Database called music.
 - Create a collection called songdetails.
 - Insert 5 song documents.
 - List all documents created.
 - List A.R.Rahman's songs.
 - List Unnikrishnan's songs.
 - Delete the song which you don't like.
 - Add new song which is your favourite.
 - List Songs sung by Hariharan from Minsara Kanavu film.
 - List out the singers' names in your document.
- Removal of a document: 'MongoDB Enterprise > db.songdetails.remove({"Film": "Kannukkul Nilavu"})'
- Confirmation of removal: 'WriteResult({ "nRemoved": 1 })'
- Final state: 'MongoDB Enterprise > '

8.

```

ukistu06@UkiPc06: ~
MongoDB Enterprise > db.songdetails.find()
{
    "_id" : ObjectId("5e15200d6ae6d0be719e1652"),
    "Singer" : "ShankarMahadevan"
}
{
    "_id" : ObjectId("5e15200d6ae6d0be719e1653"),
    "Song Name" : "Evano Oruvan",
    "Film" : "Alai Payuthey", "Singer" : "Swamalatha"
}
{
    "_id" : ObjectId("5e15200d6ae6d0be719e1654"),
    "Song Name" : "Roja Poonthoodam",
    "Film" : "Kannukkul Nilavu",
    "Music Director" : "A.R.Rahman", "Singer" : "Unnikrishnan"
}
{
    "_id" : ObjectId("5e15200d6ae6d0be719e1655"),
    "Song Name" : "Vennilavae Vennilavae Vinnaithaandi",
    "Film" : "Minsara Kanavu",
    "Music Director" : "A.R.Rahman", "Singer" : [
        "Hariharan",
        "Sadhana Sargam"
    ]
}
{
    "_id" : ObjectId("5e1521116ae6d0be719e1654"),
    "Song Name" : "Sollamal Thottu Chellum Thendral",
    "Film" : "Dheena",
    "Music Director" : "Yuvan Shankar Raja",
    "Singer" : "Hariharan"
}
MongoDB Enterprise > db.songdetails.remove({"Film":"Kannukkul Nilavu"})
WriteResult({ "nRemoved" : 1 })
MongoDB Enterprise > db.songdetails.insert({"Song Name":"Chakarakady", "Film":"Meesaya Murukku", "Music Director":"Hip hop Tamizha", "Singer": "Hip hop Tamizha"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise > 

```

Terminal - [1] * Terminal

Singer(s)	Length		
Hiphop Tamizha	4:21	Producer	Hiphop Tamizha chronology
Hiphop Tamizha	2:29		
Kaushik Krish, Hip Hop Tamizha	4:05	Kavan?	Meesaya (2017)
Mahalingam, Hip Hop Tamizha	3:17	Nurukku	Karakalappu (2017)
Rajan Chelliah, Hip Hop Tamizha	3:05		Z (2016)
Khareema Revichandran, Hiphop Tamizha	3:42		
Tamiliza			
	3:09		
Hiphop Tamizha	4:30		
	Total length: 28:38		

9.

10.

```

ukistu06@UkiPc06: ~
MongoDB Enterprise > db.songdetails.find({}, {"Singer": true})
{
    "_id" : ObjectId("5e15faa6ae6d0be719e1651"),
    "Singer" : "ShankarMahadevan"
}
{
    "_id" : ObjectId("5e15200d6ae6d0be719e1652"),
    "Singer" : "Swamalatha"
}
{
    "_id" : ObjectId("5e1521116ae6d0be719e1654"),
    "Singer" : [
        "Hariharan",
        "Sadhana Sargam"
    ]
}
{
    "_id" : ObjectId("5e1521686ae6d0be719e1655"),
    "Singer" : "Hariharan"
}
{
    "_id" : ObjectId("5e1523c6ae6d0be719e1656"),
    "Singer" : "Hip hop Tamizha"
}
MongoDB Enterprise > 

```

Terminal - [1] * Terminal

Chellum Thendral	Shankar Mahadevan	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.N.Kannan'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.