

## Introduction

We will be creating a database called “M-Flix” database. It has the following collections:

- comments:

Name	Index	Description
_id_	{ "_id": 1 }	Primary key index on the _id field.

Sample Document ¶

```
{
  "_id": {
    "$oid": "5a9427648b0beeb69579cc"
  },
  "name": "Andrea Le",
  "email": "andrea_le@fakegmail.com",
  "movie_id": {
    "$oid": "573a1390f29313caabcd418c"
  },
  "text": "Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.",
  "date": {
    "$date": {
      "$numberLong": "1332804016000"
    }
  }
}
```

- movies:

This collection contains the following indexes:

Name	Index	Description
_id_	{ "_id": 1 }	Primary key index on the _id field.

Take a look at its sample document [here](#).

- sessions:

Name	Index	Description	Properties
_id_	{ "_id": 1 }	Primary key index on the _id field.	
user_id_1	{ "user_id" : 1}	Ascending index on the user_id field.	Unique

Sample Document

```
{
  "_id": {
    "$oid": "5a98348755593fdf68350932"
  },
  "user_id": "bfb9vc1zz@xhasq.5h9",
  "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..."
}
```

- theatres:

Name	Index	Description	Properties
_id_	{ "_id": 1 }	Primary key index on the _id field.	
geo index	{ "location.geo": "2dsphere" }	Geospatial index on the location.geo field.	Sparse

A sample document is available [here](#).

- users:

_id	{ "_id": 1 }	Primary key index on the _id field.	
email_1	{ "email": 1 }	Unique, ascending index on the email field.	Unique

#### Sample Document

```
{
  "_id": {
    "$oid": "59b99db4cfa9a34dcd7885b6"
  },
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crLJ1Vu"
}
```

## Setting Up the database

Right now, we do not have the database from which we can do the operations. We need to set up the database and its corresponding collections ( *remember, collections are just like tables from relational databases* ). First, go to the folder /data/mflix. These are your collection files. Now, we need to import these .json files into our database. To do so, open the terminal and type the following command:

```
mongoimport --uri <connection_url> --collection
<collectionName> --file <filePath><fileName>
--jsonArray
```

where:

- dbName is the database name. You can choose any name for your database. Right now, the chosen name is 'mflix'.
- collectionName is the name of the collection. Again, this can be anything that you like. The name should be relevant to the documents that the collection holds.
- filePath is the path of the file where your collection ( .json file) exists.
- fileName is the collection file (.json file).
- --jsonArray accepts the data import with multiple Mongo database documents within a single JSON array.

This will create a collection in our database.

Similarly, repeat this procedure for all the collections.

### **Some helpful commands**

- 1) use mflix: switches to the database mflix. So, this becomes our current database.
- 2) show collections: shows all the collections present in the current database.
- 3) show databases: shows all the databases present.
- 4) db.<collectionName>.find(): for querying in the collection.
- 5) db.<collectionName>.insertOne(): inserts a new document into the collection.
- 6) db.<collectionName>.insertMany(): inserts multiple documents into the collection.

- 7) `db.<collectionName>.updateOne()`: updates one existing document in the collection.
- 8) `db.<collectionName>.updateMany()`: updates multiple existing documents in the collection.
- 9) `db.<collectionName>.deleteOne()`: deletes an existing document from the collection.
- 10) `db.<collectionName>.deleteMany()`: deletes multiple existing documents from the collection.
- 11) `db.<collectionName>.drop()`: removes or drops an entire collection.
- 12) `db.<collectionName>.createIndex()`: creates an index on the collection.

## Questions

For the mflix database, using the mongo shell,

- 1) Show all the movies released after 1900. Only project the title and year of the movie.
- 2) Show all the movies with IMDB rating greater than or equal to 8. Only project the title of the movie.
- 3) Petyr Baelish is known for his witty remarks. Show all the comment texts that the user named "Petyr Baelish" made.
- 4) Determine the movie with the most number of comments.
- 5) Determine which state has the maximum number of theatres. Only project the state and the number of theatres.

6) Which 5 users made the maximum number of comments? Only project the usernames and the number of comments they made.

## References

#

[https://www.tutorialspoint.com/mongodb/mongodb\\_query\\_document.htm](https://www.tutorialspoint.com/mongodb/mongodb_query_document.htm)

#

<https://docs.mongodb.com/manual/tutorial/query-embedded-documents/>

#

[https://www.tutorialspoint.com/mongodb/mongodb\\_aggregation.htm](https://www.tutorialspoint.com/mongodb/mongodb_aggregation.htm)