Lab 09 | Working with MongoDB .2

In this lab, we do some more hands-on Mongo DB.
Here you are given three exercises. Exercise 1 and Exercise 2 are practice exercises.
Only exercise 3 requires submission.

Exercise #1

Perform all mongo db operations discussed at
https://docs.mongodb.com/manual/tutorial/aggregation-zip-code-data-set/

This may require you uploading the data into your atlas. For importing let us use "`mongoimport`" which is part of mongo db tools.

Download db tools from page: https://docs.mongodb.com/database-tools/installation/installation/.
Once downloaded and un-compressed in a folder, let us say `c:\mongodb-tools`.

Move to folder mongodb-tools\bin and issue following command from terminal command prompt (not from mongo shell):

```
mongoimport --uri "mongodb+srv://cluster0.abcde.mongodb.net"
--collection zips --file "c:\datasets\zips.json" --jsonArray
--username user_name --db mydb
```

Provide your own inputs for parameters in red.

Data file is given to you.

Exercise #2

Perform following operations on zips collection created in previous exercise. And appreciate the explain command of mongo db.

```
db.zips.explain().find({ "zip": "35004"})
db.zips.explain("executionStats").find({ "zip": "35004"})
```

```
db.zips.explain().find(
   { "state": "PA", pop: { $gte: 50000 } },{ "_id": 0, "city": 1 }
)

db.zips.explain("executionStats").find(
   { "state": "PA", pop: { $gte: 50000 } },{ "_id": 0, "city": 1 }
)

db.zips.explain("allPlansExecution").find(
   { "state": "PA", pop: { $gte: 50000 } },{ "_id": 0, "city": 1 }
)
```

```
db.zips.explain("executionStats").aggregate( [
   { $group: { _id: "$state", totalPop: { $sum: "$pop" } } },
   { $match: { totalPop: { $gte: 10*1000*1000 } } }
] )
```

Run all above explain commands after creating the following indexes as following

```
//before creating index, you can check what all indexes zips already has?
db.zips.getIndexes();

//create index on zip without name
db.zips.createIndex( { zip: 1 } )

//create index on state, population, and
//    give name "stateindex" to it
db.zips.createIndex(
   { state: 1, pop: 1 },
   { name: "stateindex"}
)
```

```
//Here is how you can drop indexes
//drop index that has no name
db.zips.dropIndex( { zip: 1 } )
//drop indexes that has name
db.zips.dropIndex( "stateindex" )
```

Exercise #3

Here you are given a movies dataset (description is available at the page https://docs.atlas.mongodb.com/sample-data/sample-mflix/). While there are a few collections here, let us only use "movies" that provides details like release year, director, reviews, etc.

You require uploading the data into your atlas. You can again use "`mongoimport`" for import this dataset.

Write Mongo DB "shell" queries for following questions:

1. Show all the movies released in 2015. 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. Count movies that have website ("tomatoes.website").
4. List distinct "type" in movies collection.
5. List movie titles in 2015 that are released in more than one language while one language remains is English.
6. List non-English movies. Only project the title and year of the movie.
7. Determine the movie with the greatest number of comments.
8. Count movies in which one of director is "King Vidor".
9. Give year-wise count the movies
10. List down (genre names, and number of movies of that genre).
    Hint: may require using `$unwind`.
11. Determine top 10 directors (project director names only), based number of movies they have directed.
12. Find movies that are comedies and dramas at the same time or have a imdb rating 8 or more