SOLUTIONS TO HW-4 PART-1

MONGODB-NOSQL DATABASE SYSTEM

First execute the following command from Command Prompt C:\Program Files\mongodb\Server\bin:

mongoimport --db locationdb --collection zips –file e:\zips.json

1. Find all the states that have a city called BOSTON.

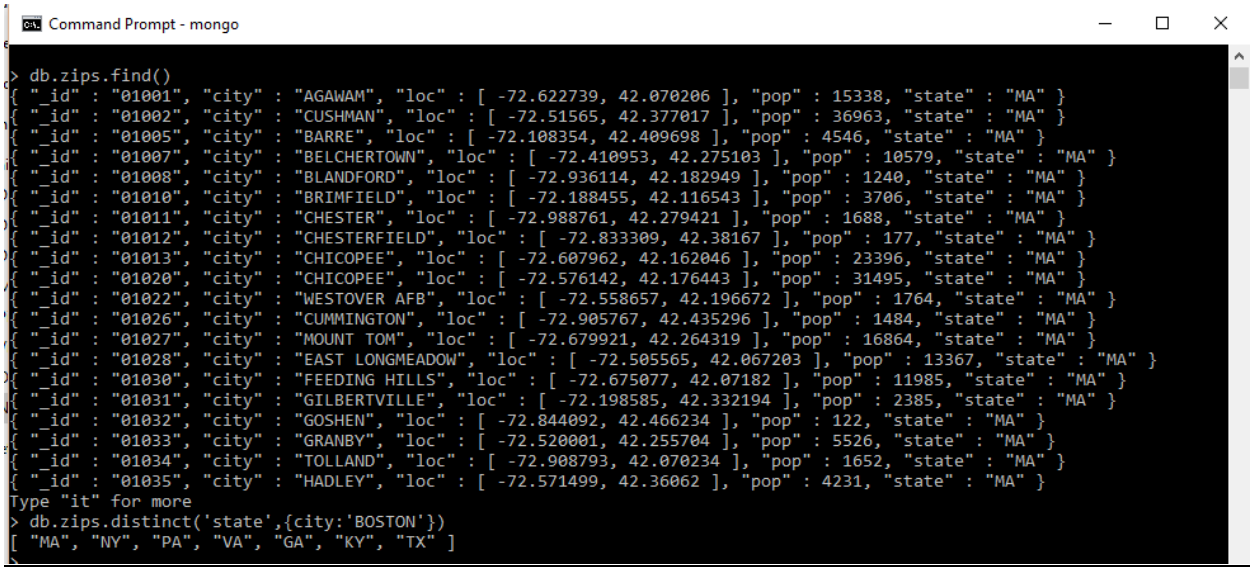
QUERY:

db.zips.distinct('state', { city : 'BOSTON' })

QUERY RESULT:

["MA", "NY", "PA", "VA", "GA", "KY", "TX"]

WINDOW:

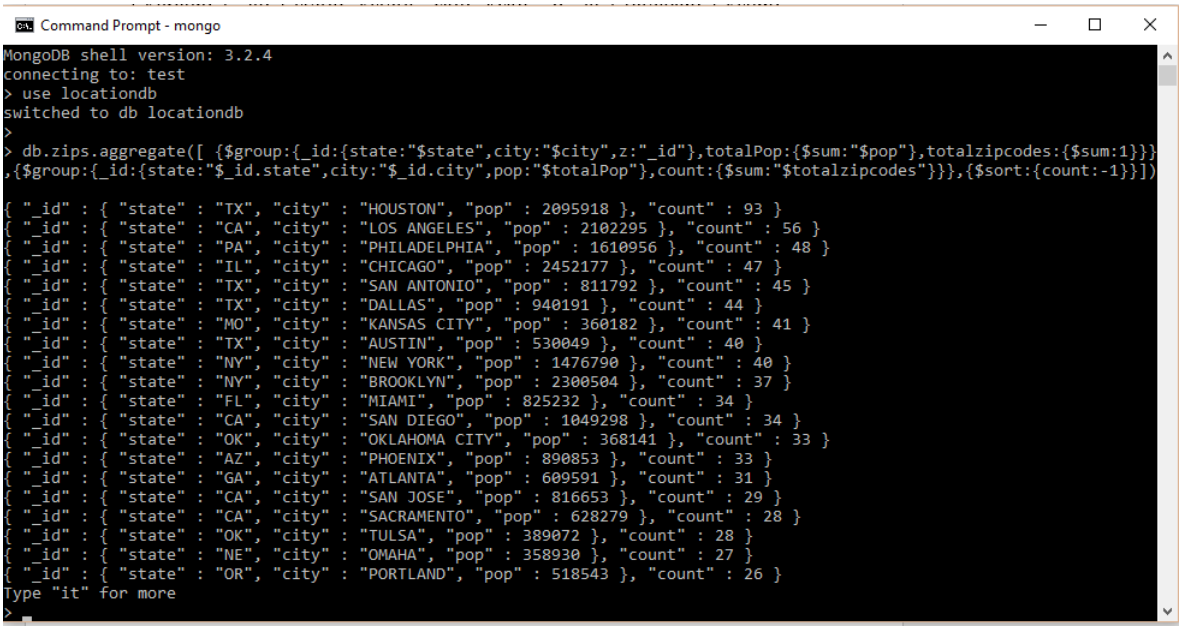


2. Each city may have several zip codes. Find the city in each state with the largest number of zip codes. Then print those cities along with their states using the city population for ordering (the cities should be ordered by their population.)

QUERY:

```
db.zips.aggregate([
{$group:{_id:{state:"$state",city:"$city",z:"_id"},totalPop:{$sum:"$pop"},totalzipcodes:{$sum:1}}},{$group:{_id:{state:"$_id.state",city:"$_id.city",pop:"$totalPop"},count:{$sum:"$totalzipcodes"}}},{$sort:{count:-1}}])
```

WINDOW:



3. MongoDB can query spatial information. Notice that for each zip code we use the latitude and longitude to store the location of the center of each zip code area. Now, consider the following location: [-72, 42] and the range (circle) of radius 2 around this point. Write a query to find all the states that intersect this range (circle). Then, you should return the total population and the number of cities for each of these states. Rank the states based on number of cities.

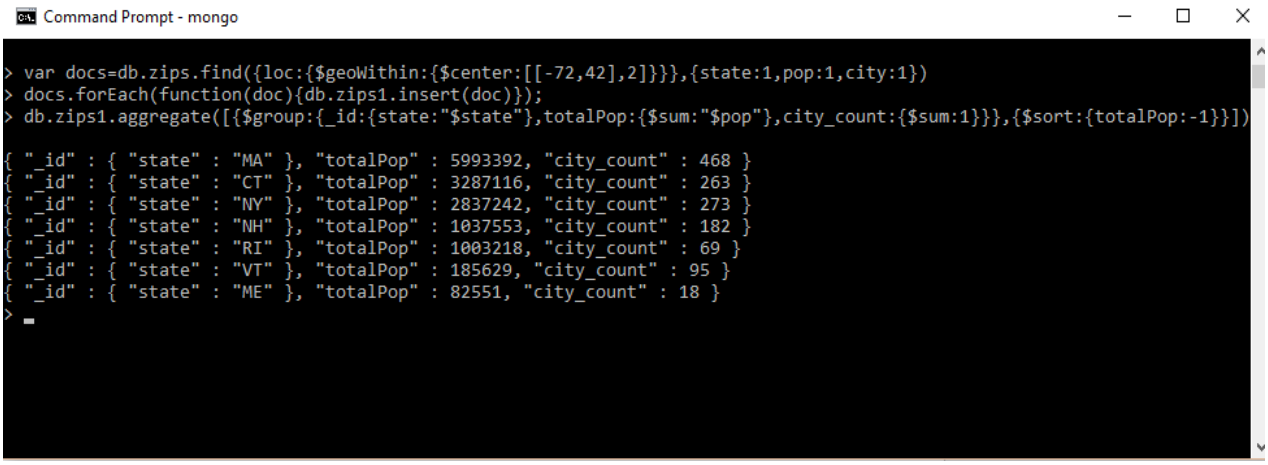
QUERY:

```
> var docs=db.zips.find({loc:{$geoWithin:{$center:[[-72,42],2]}}},{state:1,pop:1,city:1})

> docs.forEach(function(doc){db.zips1.insert(doc)});

> db.zips1.aggregate([{$group:{_id:{state:"$state"},totalPop:{$sum:"$pop"},city_count:{$sum:1}}},{$sort:{totalPop:-1}}])
```

WINDOW:



4. Consider a rectangular area with corners: [-80 , 30] , [-90 , 30] , [-90 , 40] and [-80 , 40]. Write a query to find the top 10 largest cities in this area.

QUERY:

```
db.zips.find({loc:{$geoWithin:{$box:[[-90,30],[-80,40]]}}}).sort({pop:-1}).limit(10)
```

WINDOW: