**MongoDB Lab Assignments -Day 1**

**MongoDB Exercise in mongo shell:**

Connect to a running mongo instance, use a database named mongo\_practice. Document all your queries in a javascript file to use as a reference.

Insert Documents

Insert the following documents into a movies collection

title : Fight Club

writer : Chuck Palahniuko

year : 1999

actors : [

Brad Pitt

Edward Norton

]

title : Pulp Fiction

writer : Quentin Tarantino

year : 1994

actors : [

John Travolta

Uma Thurman

]

title : Inglorious Basterds

writer : Quentin Tarantino

year : 2009

actors : [

Brad Pitt

Diane Kruger

Eli Roth

]

title : The Hobbit: An Unexpected Journey

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

title : The Hobbit: The Desolation of Smaug

writer : J.R.R. Tolkein

year : 2013

franchise : The Hobbit

title : The Hobbit: The Battle of the Five Armies

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.

title : Pee Wee Herman's Big Adventure

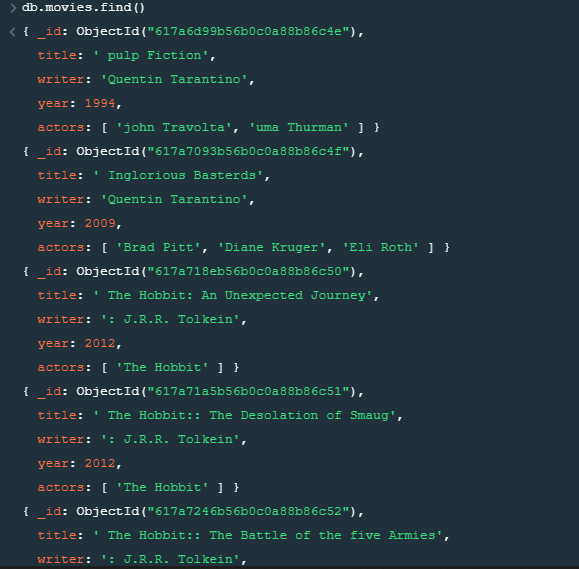
title : Avatar

Reference: <https://www.tutorialspoint.com/mongodb/mongodb_insert_document.htm>

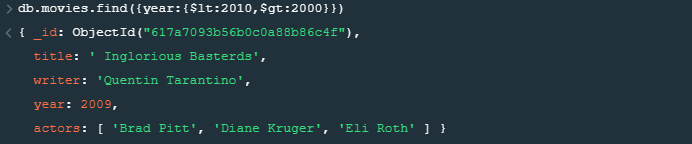
**Query / Find Documents**

query the **movies** collection to

1. get all documents.



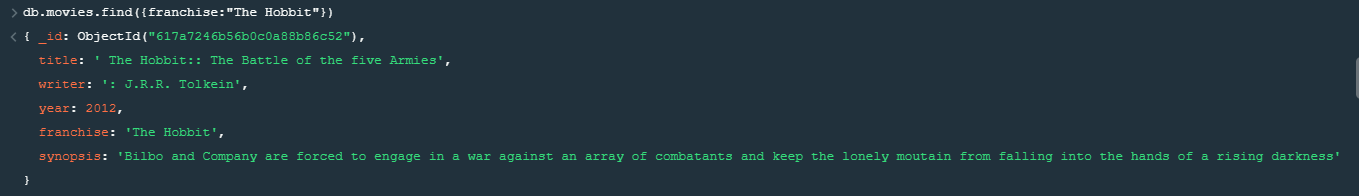
1. get all documents with writer set to "Quentin Tarantino"



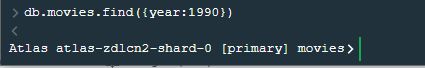
3. get all documents where actors include "Brad Pitt"



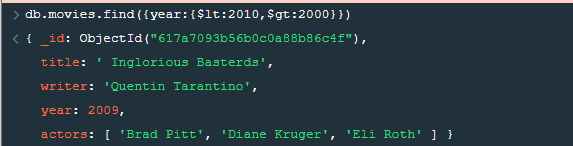
4.get all documents with franchise set to "The Hobbit".



5. . get all movies released in the 90s.



6.get all movies released before the year 2000 or after 2010.

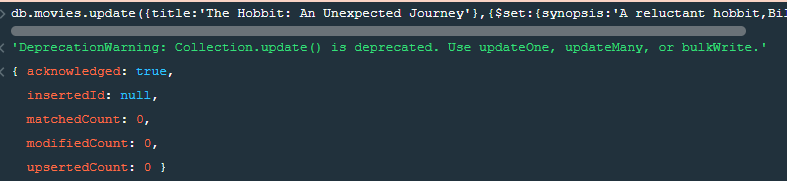


Reference:

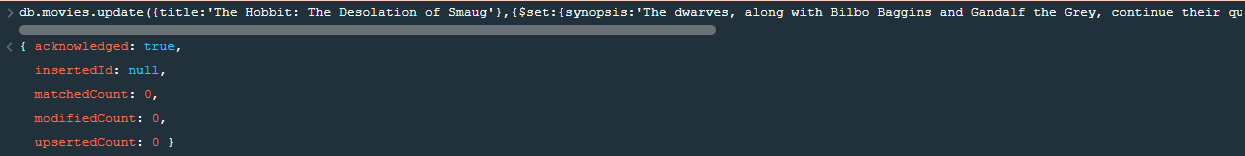
<https://www.tutorialspoint.com/mongodb/mongodb_query_document.htm>.

Update Documents

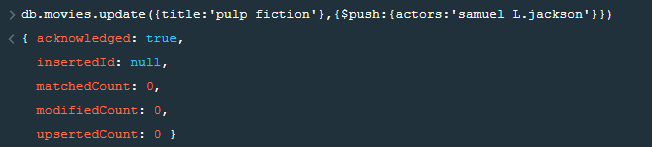
1.add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.



2.add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."



3.add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

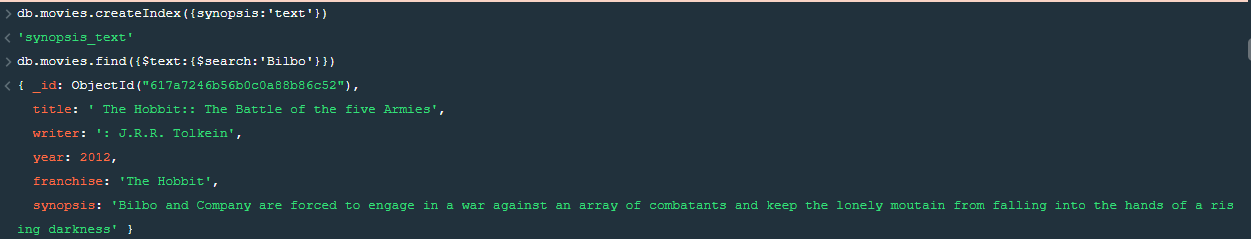


Reference:

<https://www.tutorialspoint.com/mongodb/mongodb_update_document.htm>.

Text Search

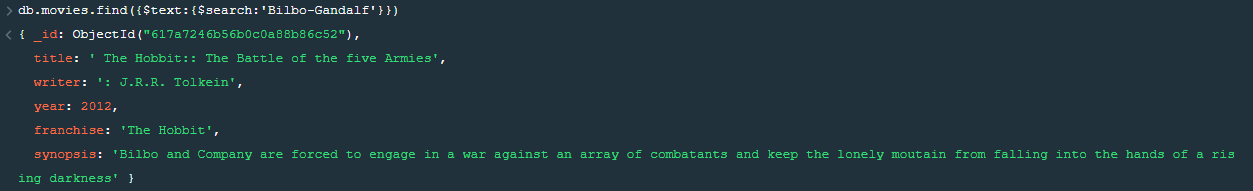
1.find all movies that have a synopsis that contains the word "Bilbo".



find all movies that have a synopsis that contains the word "Gandalf



find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"



find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

find all movies that have a synopsis that contains the word "gold" and "dragon"



Reference:

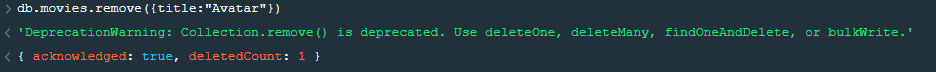
<https://www.tutorialspoint.com/mongodb/mongodb_text_search.htm>.

Delete Documents.

1.delete the movie "Pee Wee Herman's Big Adventure".



2.delete the movie "Avatar".

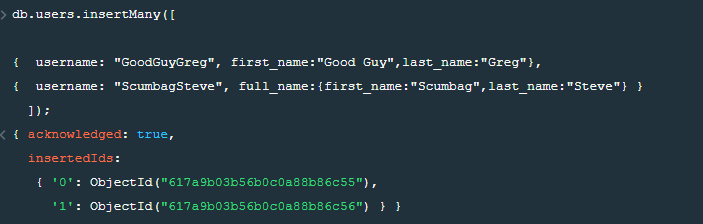


Reference:

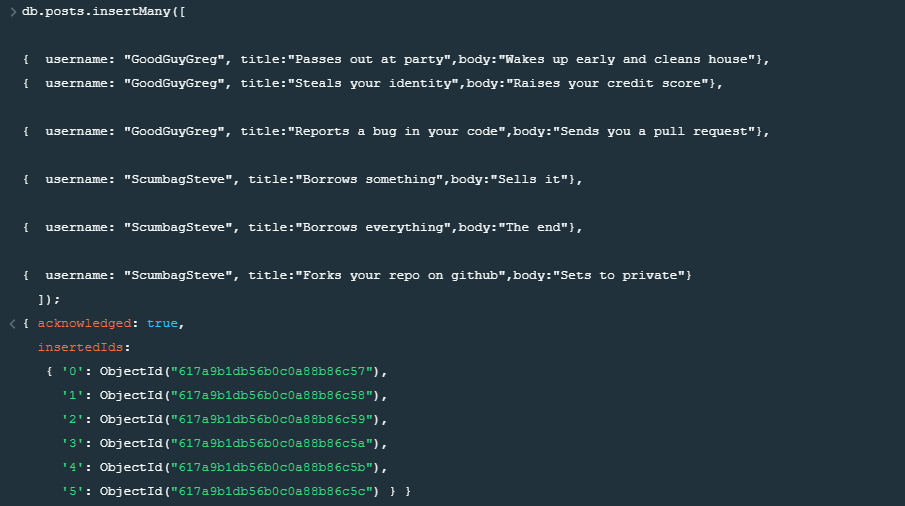
<https://www.tutorialspoint.com/mongodb/mongodb_delete_document.htm>’

**Relationships:**

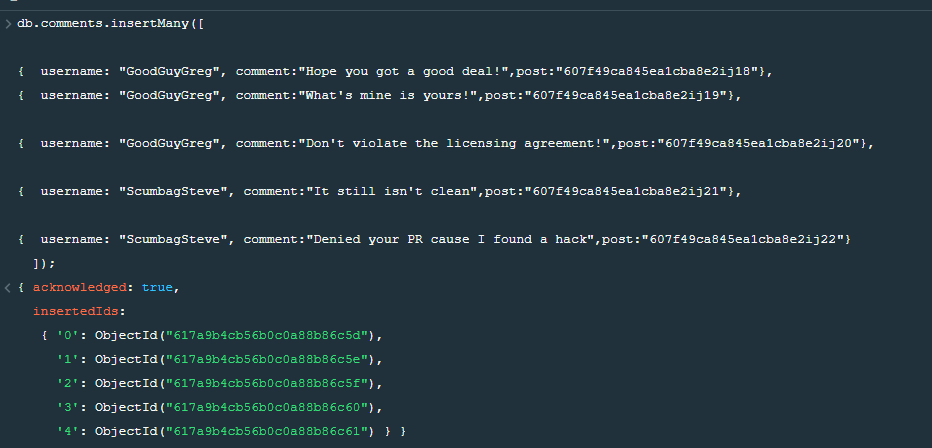
1,Insert the following documents into a users collection username : GoodGuyGreg first\_name : "Good Guy" last\_name : "Greg" username : ScumbagSteve full\_name : first : "Scumbag" last : "Steve".



2.Insert the following documents into a posts collection username : GoodGuyGreg title : Passes out at party body : Wakes up early and cleans house username : GoodGuyGreg title : Steals your identity body : Raises your credit score username : GoodGuyGreg title : Reports a bug in your code body : Sends you a Pull Request username : ScumbagSteve title : Borrows something body : Sells it username : ScumbagSteve title : Borrows everything body : The end username : ScumbagSteve title : Forks your repo on github body : Sets to private.

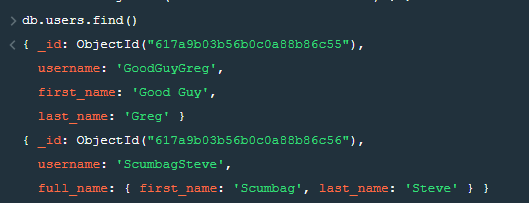


3.Insert the following documents into a comments collection username : GoodGuyGreg comment : Hope you got a good deal! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows something" username : GoodGuyGreg comment : What's mine is yours! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows everything" username : GoodGuyGreg comment : Don't violate the licensing agreement! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Forks your repo on github username : ScumbagSteve comment : It still isn't clean post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Passes out at party" username : ScumbagSteve comment : Denied your PR cause I found a hack post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Reports a bug in your code".

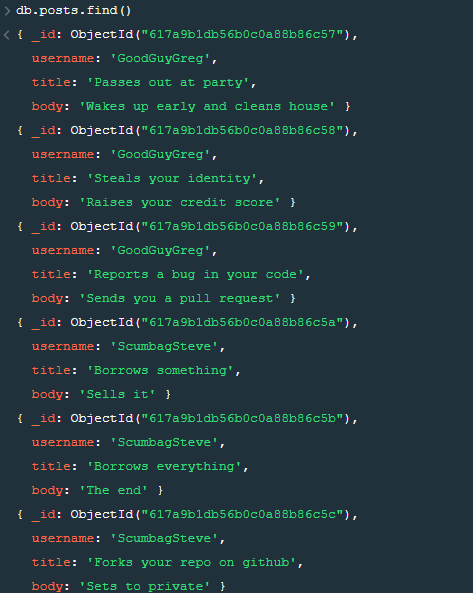


**Querying related collections.**

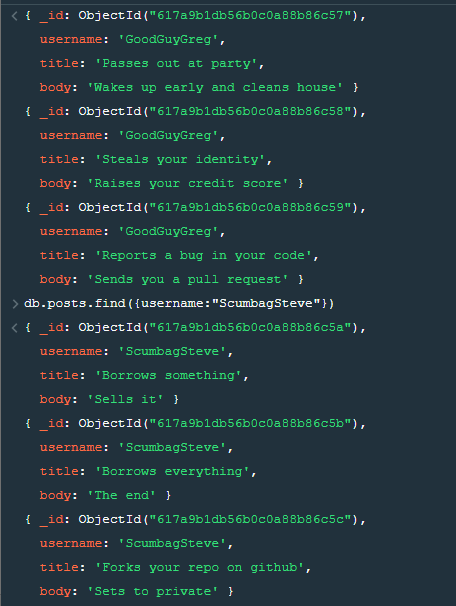
1.find all users.



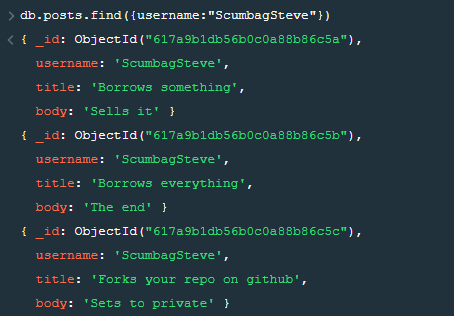
2.find all posts.



3.find all posts that was authored by "GoodGuyGreg"



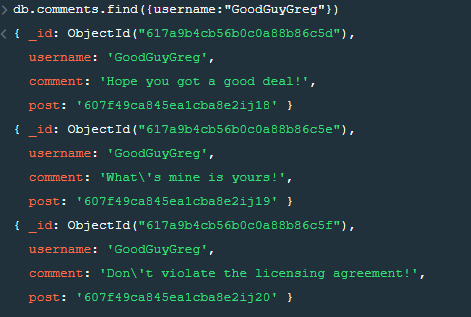
4. find all posts that was authored by "ScumbagSteve"



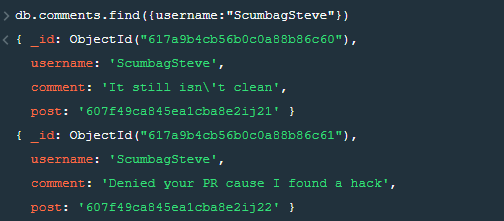
5.find all comments



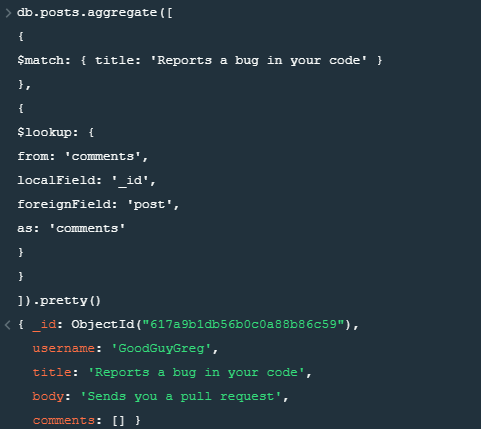
6.find all comments that was authored by "GoodGuyGreg"



7. find all comments that was authored by "ScumbagSteve"



8.find all comments belonging to the post "Reports a bug in your code"



References:

<https://docs.mongodb.com/manual/reference/method/db.collection.find/>

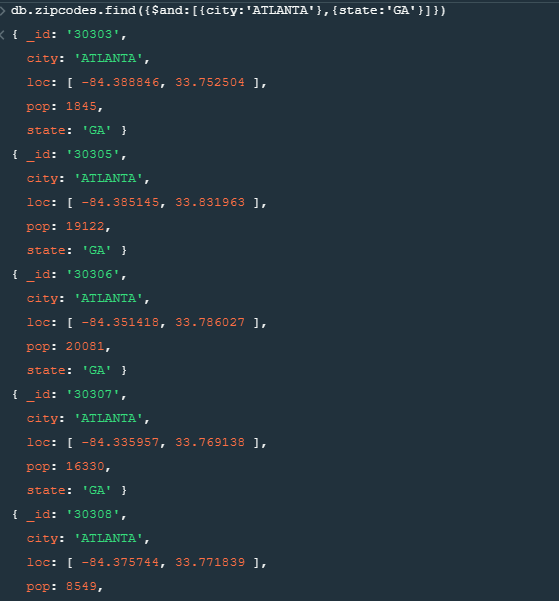
**MongoDB -Aggregation Exercises**

Import the zips.json file into your MongoDB. Database name is "population" and collection name is "zipcodes".

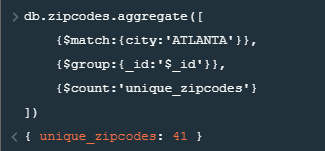
mongoimport --db population --collection zipcodes --file zips.json

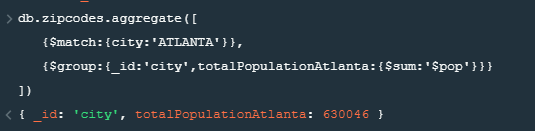
**Atlanta Population**

1.use db.zipcodes.find() to filter results toonly the results where city is ATLANTA and tate is GA.



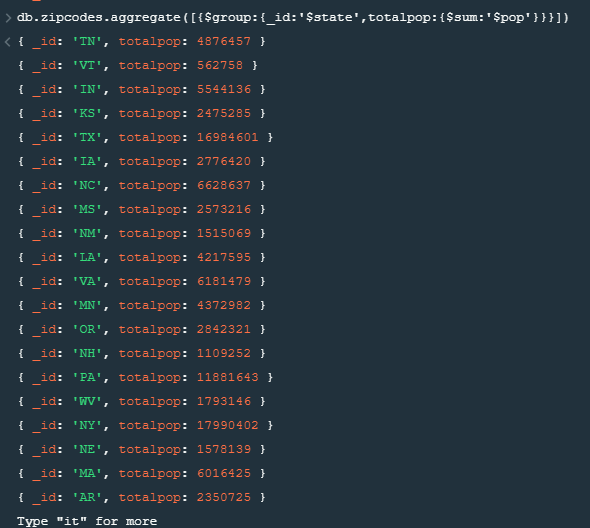
2.use db.zipcodes.aggregate with $match to do the same as above

3.use $group to count the number of zip codes in Atlanta

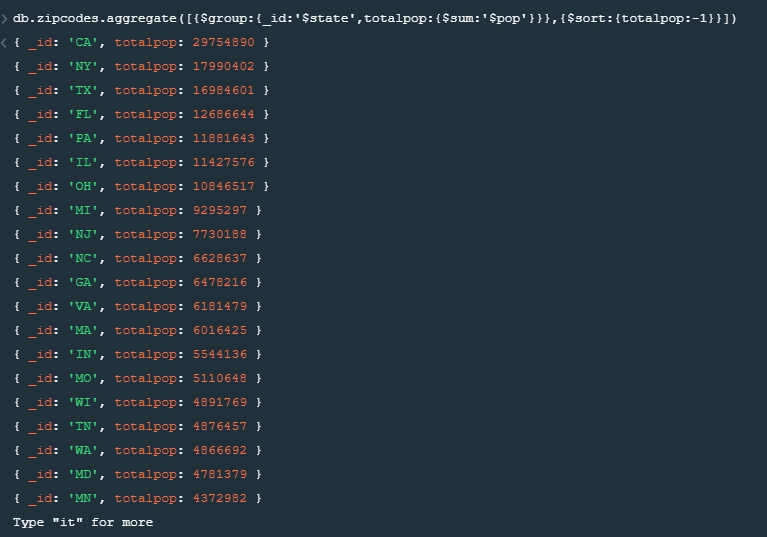
4.use $group to find the total population in Atlanta.

**Populations By state:**

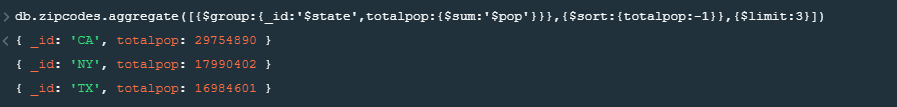
1. use aggregate to calculate the total population for each state



2.sort the results by population, highest first

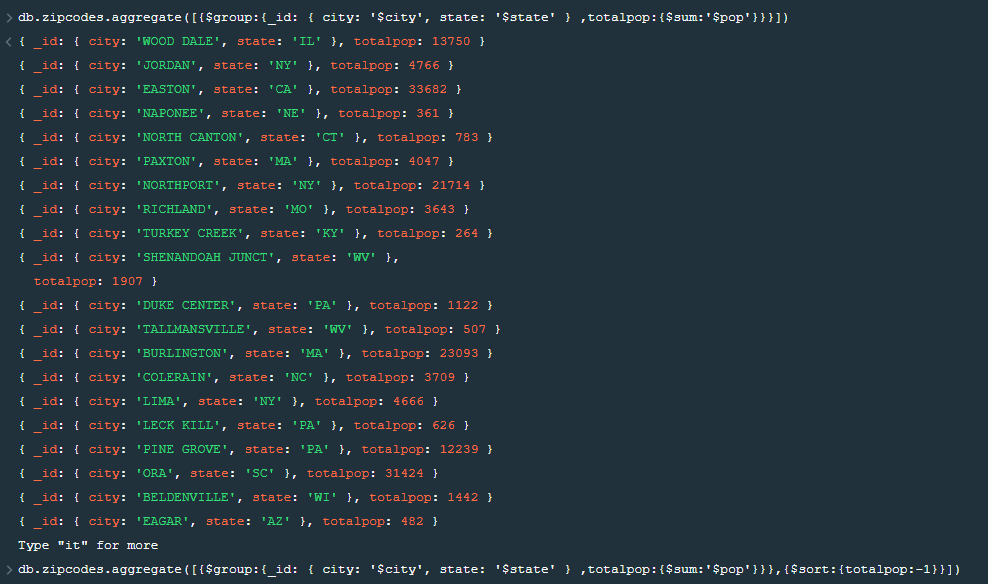


3. limit the results to just the first 3 results. What are the top 3 states in population?

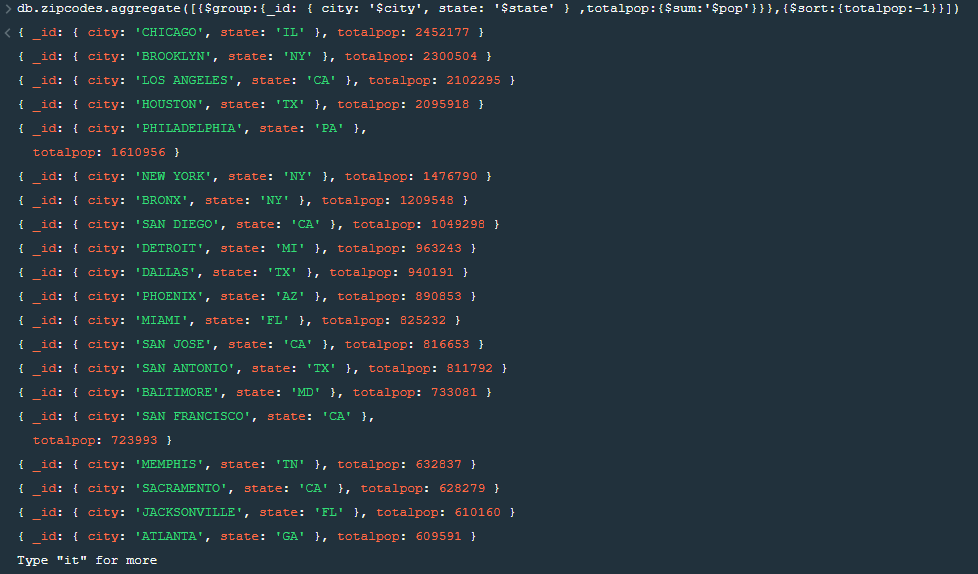


**Populations by City:**

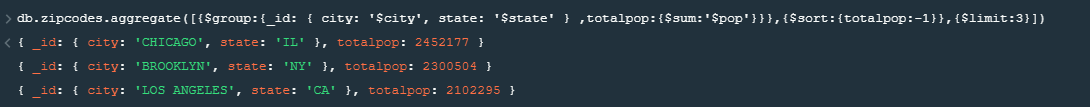
1. use aggregate to calculate the total population for each city (you have to use city/state combination). You can use a combination for the \_id of the $group: { city: '$city', state: '$state' }



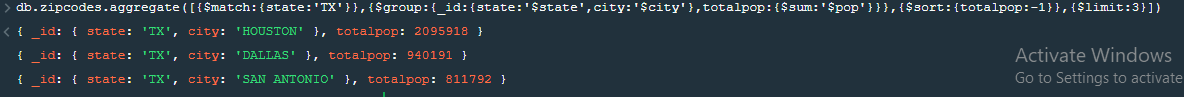
1. 2. sort the results by population, highest first



3.limit the results to just the first 3 results. What are the top 3 cities in population?

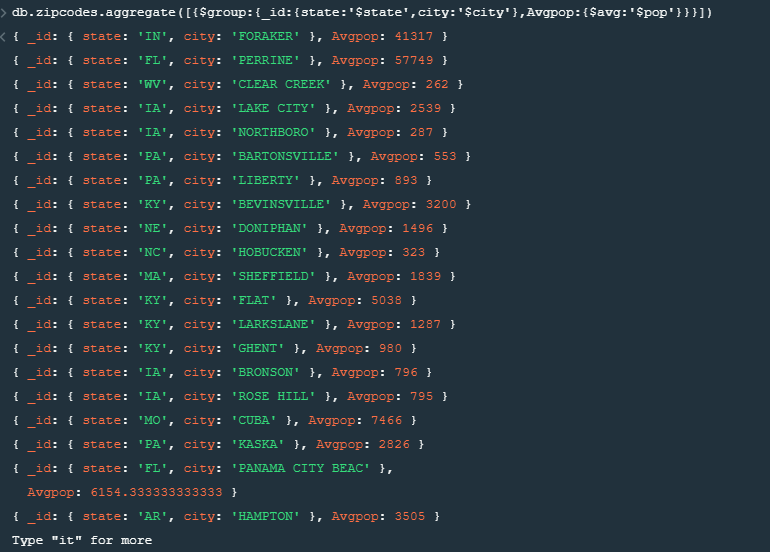


4.What are the top 3 cities in population in Texas?



**Bonus**

1.Write a query to get the average city population for each state



2.What are the top 3 states in terms of average city population\

