**MongoDB Assignment - 02**

**MongoDB -Aggregation Exercises**

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

C:\Users\trivet\Desktop\mongodb-win32-x86\_64-windows-4.4.6\bin>

mongoimport --db population --collection zipcodes --file C:\Users\trivet\Desktop\zip.json

2021-05-18T18:03:34.263+0530 connected to: mongodb://localhost/

2021-05-18T18:03:35.124+0530 29353 document(s) imported successfully. 0 document(s) failed to import.

**Atlanta Population**

1. **use db.zipcodes.find( to filter results to only the results where city is ATLANTA and state is GA).**

> db.zipcodes.find({city:"ATLANTA",state:"GA"})

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

> db.zipcodes.aggregate([

{$match :{city:"ATLANTA", state:"GA"}}

])

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

> db.zipcodes.count();

Result: 29353

1. **use $group to find the total population in Atlanta.**

> db.zipcodes.aggregate([

{$match :{city:"ATLANTA"}},

{$group :{\_id:null,"totalpopulation" : {$sum:"$pop"}}} ]);

Result: { "\_id" : null, "totalpopulation" : 630046 }

**Populations By State**

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

> db.zipcodes.aggregate([

{$group :{\_id:”$state”,"totalpopulation" : {$sum:"$pop"}}}

]);

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

> db.zipcodes.aggregate([

{$group :{\_id:"$state","totalpopulation" : {$sum:"$pop"}}},

{$sort : {totalpopulation:-1}}

])

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

> db.zipcodes.aggregate([

{$group :{\_id:"$state","totalpopulation" : {$sum:"$pop"}}},

{$sort : {totalpopulation:-1}}, {$limit:3}

])

**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)**

> db.zipcodes.aggregate([

{$group :{\_id:{city:”$city”,state:”$state”},

"totalpopulation" : {$sum:"$pop"}}}

]);

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

> db.zipcodes.aggregate([

{$group :{\_id:{city:”$city”,state:”$state”},

"totalpopulation" : {$sum:"$pop"}}},

{$sort : {totalpopulation:-1}}

])

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

> db.zipcodes.aggregate([

{$group :{\_id:{city:”$city”,state:”$state”},

"totalpopulation" : {$sum:"$pop"}}},

{$sort : {totalpopulation:-1}}, {$limit:3}

])

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

> db.zipcodes.aggregate([ {$match:{state:"TX"}},

{$group :{\_id:{city:"$city",state:"$state"}

"totalpopulation" : {$sum:"$pop"}}},

{$sort : {totalpopulation:-1}}, {$limit:3}

])

**Bonus**

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

> db.zipcodes.aggregate([

{ $group: { \_id: { state: "$state",

city: "$city" }, pop: { $sum: "$pop" } } },

{ $group: { \_id: "$\_id.state", avgCityPop: { $avg: "$pop" }}}

])

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

> db.zipcodes.aggregate([

{ $group: { \_id: { state: "$state",

city: "$city" }, pop: { $sum: "$pop" } } },

{ $group: { \_id: "$\_id.state", avgCityPop: { $avg: "$pop" }}},

{$sort : {avgCityPop:-1}}, {$limit:3}

])