# 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 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.aggregate([{$match: {"city": "ATLANTA"}}, {$group:{\_id:"$city", count:{$sum: 1}}}])

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

db.zipcodes.aggregate([{$match: {"city": "ATLANTA"}}, {$group: {\_id: "$city", totalPop: {$sum: "$pop"}}}])

# Populations By State

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

db.zipcodes.aggregate([{$group: {\_id: "$state", totalPop: {$sum: "$pop"}}}])

1. sort the results by population, highest first

db.zipcodes.aggregate([{$sort:{"pop": -1}}])

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

db.zipcodes.aggregate([{$sort:{"pop": -1}},{$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([{$group: {\_id: {state: "$state", city: "$city"}, pop: {$sum: "$pop"}}}])

1. sort the results by population, highest first

db.zipcodes.aggregate([{ $group: {\_id: {state: "$state", city: "$city"}, pop: {$sum: "$pop"}}}, {$sort: { pop: -1}}])

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

db.zipcodes.aggregate([{$group: {\_id: {state: "$state", city: "$city"}, pop: {$sum: "$pop"}}}, {$sort: { pop: -1}}, {$limit: 3}])

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

db.zipcodes.aggregate([{$match: {"state" : "TXS"}}, {$group: {\_id: {state: "$state", city: "$city"}, pop: {$sum:"$pop"}}}, {$sort: {pop: -1}}, {$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"}}}]).

1. 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}] )