



MOGODB-LAB2



BY: DOAA SHERIF ABDELSALAM

TRACK: PD-ALEX-43

MONGODB LAB2

1 - Download the following json file and import it into a collection named “zips” into “iti” database

```
CA Command Prompt
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mega Store>mongoimport --db iti_db --collection zips --file "C:\Users\Mega Store\Documents\MongoDBLabs\zips.json" --type json
2023-04-18T10:17:07.325+0200    connected to: mongodb://localhost/
2023-04-18T10:17:07.952+0200    29353 document(s) imported successfully. 0 document(s) failed to import.
```

2 – find all documents which contains data related to “NY” state
iti_db> db.zips.find({state: "NY"})

```
CA mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

iti_db> db.zips.find({state: "NY"})
[switched to db iti_db]
[
  {
    _id: '06390',
    city: 'FISHERS ISLAND',
    loc: [ -72.017834, 41.263934 ],
    pop: 329,
    state: 'NY'
  },
  {
    _id: '10001',
    city: 'NEW YORK',
    loc: [ -73.996705, 40.74838 ],
    pop: 18913,
    state: 'NY'
  },
  {
    _id: '10002',
    city: 'NEW YORK',
    loc: [ -73.987681, 40.715231 ],
    pop: 84143,
    state: 'NY'
  },
  {
    _id: '10003',
    city: 'NEW YORK',
    loc: [ -73.989223, 40.731253 ],
    pop: 51224,
    state: 'NY'
  },
  {
    _id: '10005',
    city: 'NEW YORK',
    loc: [ -74.008344, 40.705649 ],
    pop: 202,
    state: 'NY'
  },
  {
    _id: '10004',
    city: 'GOVERNORS ISLAND',

```

3 – find all zip codes whose population is greater than or equal to 1000

iti_db> db.zips.find({pop: {\$gte: 1000}})

```
type it for more
iti_db> db.zips.find({pop: {$gte: 1000}})
[
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [ -72.51565, 42.377017 ],
    pop: 36963,
    state: 'MA'
  },
  {
    _id: '01005',
    city: 'BARRE',
    loc: [ -72.108354, 42.409698 ],
    pop: 4546,
    state: 'MA'
  },
  {
    _id: '01007',
    city: 'BELCHERTOWN',
    loc: [ -72.410953, 42.275103 ],
    pop: 10579,
    state: 'MA'
  },
  {
    _id: '01001',
    city: 'AGAWAM',
    loc: [ -72.622739, 42.070206 ],
    pop: 15338,
    state: 'MA'
  },
  {
    _id: '01008',
    city: 'BLANDFORD',
    loc: [ -72.936114, 42.182949 ],
    pop: 1240,
    state: 'MA'
  }
]
```

4 – add a new boolean field called “check” and set its value to true for “PA” and “VA” state

```
iti_db> db.zips.updateMany({$or: [{state: "PA"}, {state: "VA"}]}, {$set: {check: "true"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2274,
  modifiedCount: 2274,
  upsertedCount: 0
}
iti_db> _
```

5 – using zip codes find all cities whose latitude is between 55 and 65 and show the population only.

```
iti_db> db.zips.find({ "loc.1": { $gt: 55, $lt: 65 } }, { _id: 0, city: 1, pop: 1 })
[
  { city: 'ANCHORAGE', pop: 14436 },
  { city: 'ANCHORAGE', pop: 32383 },
  { city: 'ANCHORAGE', pop: 12534 },
  { city: 'FORT RICHARDSON', pop: 7979 },
  { city: 'ELMENDORF AFB', pop: 7907 },
  { city: 'ANCHORAGE', pop: 20128 },
  { city: 'ANCHORAGE', pop: 29857 },
  { city: 'ANCHORAGE', pop: 17094 },
  { city: 'ANCHORAGE', pop: 8116 },
  { city: 'ANCHORAGE', pop: 15192 },
  { city: 'PORT HEIDEN', pop: 119 },
  { city: 'AKIACHAK', pop: 481 },
  { city: 'ANCHORAGE', pop: 18356 },
  { city: 'AKIAK', pop: 285 },
  { city: 'ANCHORAGE', pop: 15891 },
  { city: 'CHUATHBALUK', pop: 352 },
  { city: 'NIKOLAEVSK', pop: 1698 },
  { city: 'ALAKANUK', pop: 1186 },
  { city: 'ATMAUTLUAK', pop: 7188 },
  { city: 'ANVIK', pop: 296 }
]
```

6 – create index for states to be able to select it quickly and check any query explain using the index only.

```
iti_db> db.zips.createIndex({state:1})
state_1
iti_db>
```

7 – increase the population by 0.2 for all cities which doesn't located in “AK” nor “NY”

```
iti_db> db.zips.updateMany({state:{$nin:["AK","NY"]}},{$mul:{pop:1.2}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 27563,
  modifiedCount: 27563,
  upsertedCount: 0
}
iti_db> _
```

8 – update only one city whose longitude is lower than -71 and is not located in “MA” state, set its population to 0 if zipcode population less than 200.

```
iti_db> db.zips.updateOne({ state: { $ne: "MA" }, "loc.0": { $lt: -71 }, pop: { $lt: 200 } }, { $set: { pop: 0 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
iti_db> _
```

9 – update all documents whose city field is a string, rename its city field to be country and if there isn't any, add new document the same as the first document in the database but change the _id to avoid duplications.

```
iti_db> var firstDoc = db.zips.findOne();

iti_db> var numUpdated=db.zips.updateMany(
...   { city: { $type: "string" } },
...   { $rename: { city: "country" } }
... );

iti_db> if (numUpdated === 0) {
...   var newDoc = Object.assign({}, firstDoc);
...   newDoc._id = new ObjectId();
...   db.zips.insertOne(newDoc);
... }else{
...   console.log("UPDATED");
... }
UPDATED
iti_db> _
```

PART2

1. Get sum of population that state in PA, KA

```
iti_db> db.zips.aggregate([
...   {$match: {state: {$in: ["PA", "KA"]}}},
...   {$group: {_id: "$state", totalPopulation: {$sum: "$pop"}}}
... ])
[ { _id: 'PA', totalPopulation: 14257971.6 } ]
iti_db>
```

2. Get only 5 documents that state not equal to PA, KA

```
iti_db> db.zips.find({ state: { $nin: [ "PA", "KA" ] } }).limit(5)
[
  {
    _id: '99501',
    loc: [ -149.876077, 61.211571 ],
    pop: 14436,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99504',
    loc: [ -149.74467, 61.203696 ],
    pop: 32383,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99503',
    loc: [ -149.893844, 61.189953 ],
    pop: 12534,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99505',
    loc: [ -149.675454, 61.275256 ],
    pop: 7979,
    state: 'AK',
    country: 'FORT RICHARDSON'
  },
  {
    _id: '99506',
    loc: [ -149.812667, 61.251531 ],
    pop: 7907,
    state: 'AK',
    country: 'ELMENDORF AFB'
  }
]
iti_db>
```

3. Get sum of population that state equal to AK and their latitude between 55, 65

```
iti_db> db.zips.aggregate([{$match:{state:"AK",loc:{$elemMatch:{$gte:55,$lte:65}}}},{$group: {_id:"$state",totalPop:{$sum:"$pop"}}})
[ { _id: 'AK', totalPop: 524636 } ]
iti_db>
```

4. Sort Population of document that state in AK, PA and skip first 7 document

```
iti_db> db.zips.aggregate([{$match:{$state:{$in:['AK','PA']}}},{ $sort:{$pop:1}},{$skip:7}])
[
  {
    _id: '99773',
    loc: [ -157.613496, 66.958141 ],
    pop: 0,
    state: 'AK',
    country: 'SHUNGNAK'
  },
  {
    _id: '99770',
    loc: [ -158.534287, 65.713537 ],
    pop: 0,
    state: 'AK',
    country: 'SELANIK'
  },
  {
    _id: '15744',
    loc: [ -79.093987, 40.921432 ],
    pop: 0,
    state: 'PA',
    check: 'true',
    country: 'HAMILTON'
  },
  {
    _id: '19113',
    loc: [ -75.275196, 39.864998 ],
    pop: 0,
    state: 'PA',
    check: 'true',
    country: 'PHILADELPHIA'
  },
  {

```

5. Get smallest population and greatest population of each state and save the result in collection named "mypop" on your machine colleague

```
iti_db> db.zips.aggregate([{$group:{$id:"$state",smallestPop:{$min:"$pop"},greatestPop:{$max:"$pop"}},{ $out:"mypop"}])
```

6. Write an aggregation expression to calculate the average population of a zip code (postal code) by state

```

iti_db> db.zips.aggregate([{$group: {_id: "$state", avgPop: {$avg: "$pop"}}}])
[
  { _id: 'OR', avgPop: 8882.253125 },
  { _id: 'WY', avgPop: 3887.382857142857 },
  { _id: 'NY', avgPop: 11279.248902821317 },
  { _id: 'MT', avgPop: 3053.304458598726 },
  { _id: 'TN', avgPop: 10054.550515463916 },
  { _id: 'MN', avgPop: 5949.63537414966 },
  { _id: 'CT', avgPop: 14998.247908745247 },
  { _id: 'PA', avgPop: 9779.130041152262 },
  { _id: 'KS', avgPop: 4154.324475524476 },
  { _id: 'GA', avgPop: 12242.297952755904 },
  { _id: 'NC', avgPop: 11282.786382978722 },
  { _id: 'DC', avgPop: 30345 },
  { _id: 'NJ', avgPop: 17178.195555555554 },
  { _id: 'ND', avgPop: 1958.8910485933504 },
  { _id: 'LA', avgPop: 10907.573275862069 },
  { _id: 'OK', avgPop: 6441.470989761092 },
  { _id: 'WV', avgPop: 3280.145121951219 },
  { _id: 'DE', avgPop: 15083.049056603773 },
  { _id: 'MO', avgPop: 6169.796378269617 },
  { _id: 'NV', avgPop: 13867.303846153845 }
]
Type "it" for more

```

7. Write an aggregation query with just a sort stage to sort by (state, city), both ascending

City→ country after update in no 9 in part 1

```
iti_db> db.zips.aggregate([{$sort:{state:1,city:1}}])
[
  {
    _id: '99615',
    loc: [ -152.500169, 57.781967 ],
    pop: 13309,
    state: 'AK',
    country: 'AKHIOK'
  },
  {
    _id: '99551',
    loc: [ -161.39233, 60.891854 ],
    pop: 481,
    state: 'AK',
    country: 'AKIACHAK'
  },
  {
    _id: '99552',
    loc: [ -161.199325, 60.890632 ],
    pop: 285,
    state: 'AK',
    country: 'AKIAK'
  },
  {
    _id: '99553',
    loc: [ -165.785368, 54.143012 ],
    pop: 589,
    state: 'AK',
    country: 'AKUTAN'
  },
]
```

8. Write an aggregation query with just a sort stage to sort by (state, city), both descending

```
iti_db> db.zips.aggregate([{$sort:{state:-1,city:-1}}])
[
  {
    _id: '82244',
    loc: [ -104.353507, 41.912018 ],
    pop: 808.8,
    state: 'WY',
    country: 'YODER'
  },
  {
    _id: '82732',
    loc: [ -105.532327, 43.829349 ],
    pop: 2558.4,
    state: 'WY',
    country: 'WRIGHT'
  },
  {
    _id: '82401',
    loc: [ -107.95626, 44.013796 ],
    pop: 9231.6,
    state: 'WY',
  },
]
iti_db>
[
  {
    _id: '83014',
    loc: [ -110.874199, 43.49922 ],
    pop: 1318.8,
    state: 'WY',
  },
]
```

9. Calculate the average population of cities in California (abbreviation CA) and New York (NY) (taken together) with populations over 25,000

```
iti_db> db.zips.aggregate([
...   {$match: {state: {$in: ["CA", "NY"]}, pop: {$gte: 25000}}},
...   {$group: {_id: {state: "$state"}, avgPop: {$avg: "$pop"}}}
... ])
[
  { _id: { state: 'CA' }, avgPop: 46673.271224165335 },
  { _id: { state: 'NY' }, avgPop: 44494.818930041154 }
]
iti_db>
```

10. Return the average populations for cities in each state

```
iti_db> db.zips.aggregate([
...   {$group: {_id: {state: "$state"}, avgPop: {$avg: "$pop"}}},
...   {$sort: {"_id.state": 1}}
... ])
[
  { _id: { state: 'AK' }, avgPop: 2792.7128205128206 },
  { _id: { state: 'AL' }, avgPop: 8551.506878306878 },
  { _id: { state: 'AR' }, avgPop: 4880.397923875433 },
  { _id: { state: 'AZ' }, avgPop: 16289.902222222221 },
  { _id: { state: 'CA' }, avgPop: 23552.683377308706 },
  { _id: { state: 'CO' }, avgPop: 9547.115942028986 },
  { _id: { state: 'CT' }, avgPop: 14998.247908745247 },
  { _id: { state: 'DC' }, avgPop: 30345 },
  { _id: { state: 'DE' }, avgPop: 15083.049056603773 },
  { _id: { state: 'FL' }, avgPop: 18935.289552238806 },
  { _id: { state: 'GA' }, avgPop: 12242.297952755904 },
  { _id: { state: 'HI' }, avgPop: 16623.435 },
  { _id: { state: 'IA' }, avgPop: 3613.5618221258133 },
  { _id: { state: 'ID' }, avgPop: 4951.224590163934 },
  { _id: { state: 'IL' }, avgPop: 11085.764915117219 },
  { _id: { state: 'IN' }, avgPop: 9841.661538461538 },
  { _id: { state: 'KS' }, avgPop: 4154.324475524476 },
  { _id: { state: 'KY' }, avgPop: 5451.892212608158 },
  { _id: { state: 'LA' }, avgPop: 10907.573275862069 },
  { _id: { state: 'MA' }, avgPop: 15231.45569620253 }
]
```