1

Thursday, April 20, 2023 5:25 AM

Mongo DB

Lab 2

Name : Karim Ali Ramadan
Track : Professional Web Development & BI
Intake 43 – Alexandria Branch

Part1

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

```
Arabtech@DESKTOP-STJ4OC2 MINGW64 /d/Web/My Github/ITI/MongoDB/Lab 2 (main)
$ mongoimport --version
mongoimport version: 100.7.0
git version: 17946f45f5fabfcdd99f8960b9109f976d370631
Go version: go1.19.3
   os: windows
   arch: amd64
   compiler: gc
```

```
Arabtech@DESKTOP-STJ4OC2 MINGW64 /d/Web/My Github/ITI/MongoDB/Lab 2 (main)
$ mongoimport --uri "mongodb://127.0.0.1:27017/ITI" --collection 'zips' --file 'zips.json'
2023-04-20T05:35:41.332+0200    connected to: mongodb://127.0.0.1:27017/ITI
2023-04-20T05:35:42.212+0200    29353 document(s) imported successfully. 0 document(s) failed to import.
```

```
test> use ITI
switched to db ITI
ITI> show collections
employees
students
zips
```

2 – find all documents which contains data related to “NY” state

```
ITI> db.zips.find({state:"NY"})
[
  {
    _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: '10005',
    city: 'NEW YORK',
    loc: [ -74.008344, 40.705649 ],
    pop: 202,
    state: 'NY'
  },
  {
    _id: '10003',
    city: 'NEW YORK',
    loc: [ -73.989223, 40.731253 ],
    pop: 51224,
    state: 'NY'
  },
  {
    _id: '10004',
    city: 'GOVERNORS ISLAND',
    loc: [ -74.019025, 40.693604 ],
    pop: 3593,
    state: 'NY'
  },
```

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

```
ITI> db.zips.find({pop:{$gte:1000}})
[
  {
    _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: '01013',
    city: 'CHICOPEE',
    loc: [ -72.607962, 42.162046 ],
    pop: 23396,
    state: 'MA'
  },
  {
    _id: '01020',
    city: 'CHICOPEE',
    loc: [ -72.576142, 42.176443 ],
    pop: 31495,
    state: 'MA'
  },
  {
    _id: '01010',
    city: 'BRIMFIELD',
    loc: [ -72.188455, 42.116543 ],
    pop: 3706,
    state: 'MA'
  },
```

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

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

```
ITI> db.zips.find({$or : [{state:'VA'},{state:'PA'}]})
[
  {
    _id: '15003',
    city: 'FAIROAKS',
    loc: [ -80.219778, 40.595368 ],
    pop: 13449,
    state: 'PA',
    check: true
  },
  {
    _id: '15001',
    city: 'MACARTHUR',
    loc: [ -80.281567, 40.604424 ],
    pop: 36849,
    state: 'PA',
    check: true
  },
  {
    _id: '15005',
    city: 'BADEN',
    loc: [ -80.198471, 40.641595 ],
    pop: 10068,
    state: 'PA',
    check: true
  },
  {
    _id: '15007',
    city: 'BAKERSTOWN',
    loc: [ -79.930956, 40.647826 ],
    pop: 782,
    state: 'PA',
    check: true
  },
  {
    _id: '15009',
    city: 'BEAVER',
    loc: [ -80.336528, 40.697184 ],
    pop: 14968,
    state: 'PA',
    check: true
  },
  {
    _id: '15017',
    city: 'BRIDGEVILLE',
    loc: [ -80.115293, 40.347195 ],
    pop: 12705,
```

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

```
ITI> db.zips.find(
...     {
...         "loc.1": {$gte:55,$lte:66}
...     },
...     {
...         pop:1
...     }
... )
[
  { _id: '99503', pop: 12534 },
  { _id: '99502', pop: 15891 },
  { _id: '99501', pop: 14436 },
  { _id: '99505', pop: 7979 },
  { _id: '99507', pop: 20128 },
  { _id: '99506', pop: 7907 },
  { _id: '99504', pop: 32383 },
  { _id: '99516', pop: 18356 },
  { _id: '99508', pop: 29857 },
  { _id: '99518', pop: 8116 },
  { _id: '99517', pop: 15192 },
  { _id: '99515', pop: 17094 },
  { _id: '99549', pop: 119 },
  { _id: '99551', pop: 481 },
  { _id: '99552', pop: 285 },
  { _id: '99555', pop: 185 },
  { _id: '99554', pop: 1186 },
  { _id: '99558', pop: 296 },
  { _id: '99557', pop: 352 },
  { _id: '99556', pop: 1698 }
]
Type "it" for more
ITI>
```

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

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

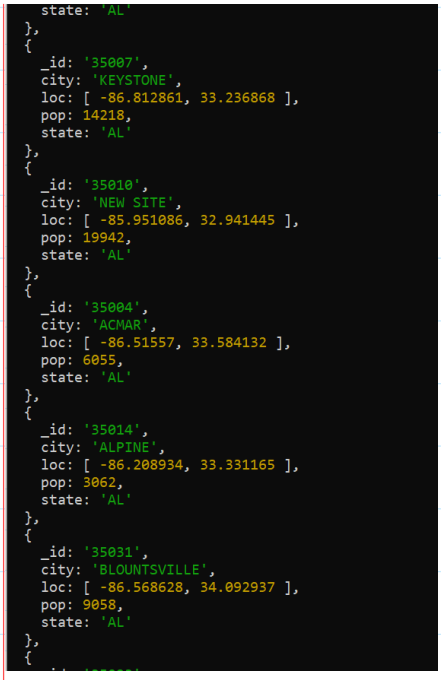
```
ITI> db.zips.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { state: 1 }, name: 'state_1' }
]
ITI>
```

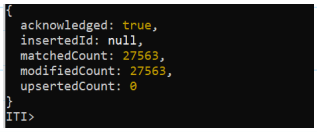
7 – increase the population by 0.2 for all cities which doesn't located in “AK” nor “NY”

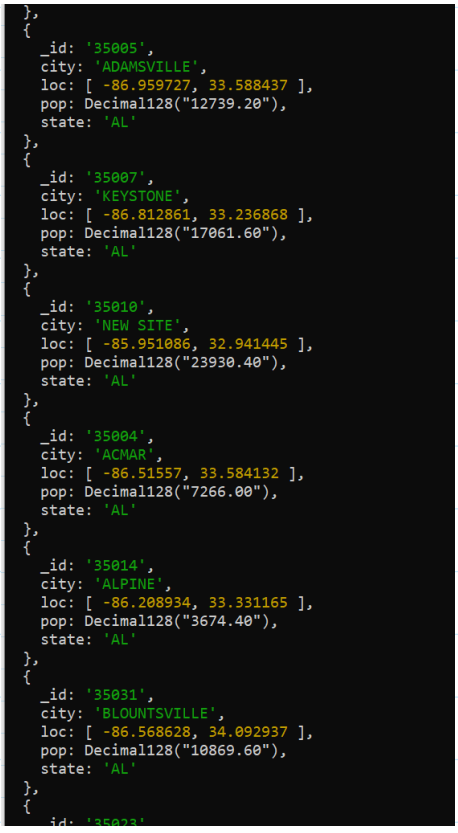
```
ITI> db.zips.find({ state : {$nin : ["AK","NY"]} })
[
  {
    _id: '35006',
    city: 'ADGER',
    loc: [ -87.167455, 33.434277 ],
    pop: 3205,
    state: 'AL'
  },
  {
    _id: '35005',
    city: 'ADAMSVILLE',
    loc: [ -86.959727, 33.588437 ],
    pop: 10616,
    state: 'AL'
  },
  {
    _id: '35007',
    city: 'KEYSTONE',
    loc: [ -86.812861, 33.236868 ],
```

```
ITI> db.zips.updateMany(
...     {
...         state : {$nin : ["AK","NY"]}
...     },
...     {
...         $mul : {
...             pop : Decimal128("1.20")
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 27563,
  modifiedCount: 27563,
  upsertedCount: 0
}
```

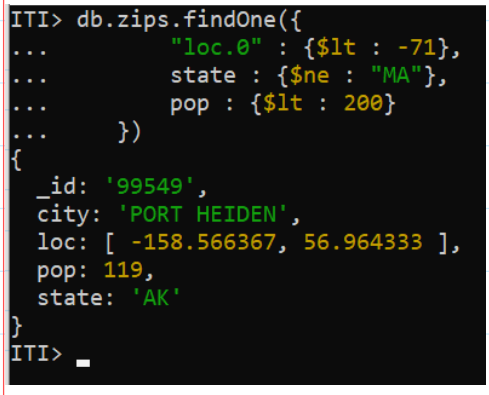
```
ITI> db.zips.find({ state : {$nin : ["AK","NY"]} })
[
  {
    _id: '35006',
    city: 'ADGER',
    loc: [ -87.167455, 33.434277 ],
    pop: Decimal128("3846.00"),
    state: 'AL'
  },
  {
    _id: '35005',
    city: 'ADAMSVILLE',
    loc: [ -86.959727, 33.588437 ],
```

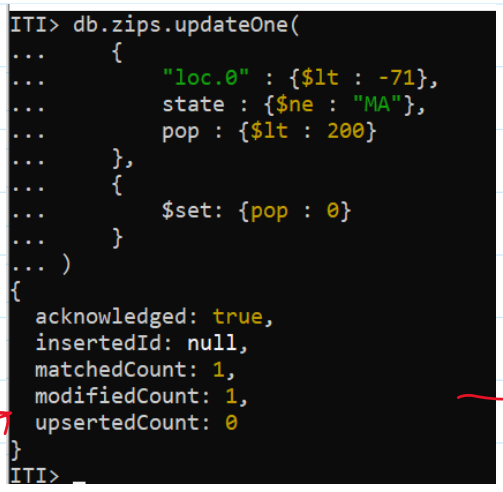






8 – update only one city whose longitude is lower than -71 and is not located in “MA” state, set its population to 0 if zipcode population less than 200.





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 documet in the database but change the _id to avoid duplications.

Hint: use Variables

****

First : **city field is string in all documents**

```
ITI> db.zips.find({}).count()
29353
ITI> db.zips.find({city : {$type : "string"}}).count()
29353
```

Second : **city field is string in all documents**

```
ITI> var requiredDocuments = db.zips.find({city : {$type : "string"}})
```

```
ITI> requiredDocuments
[
  {
    _id: '01007',
    city: 'BELCHERTOWN',
    loc: [ -72.410953, 42.275103 ],
    pop: Decimal128("12694.80"),
    state: 'MA'
  },
  {
    _id: '01001',
    city: 'AGAWAM',
    loc: [ -72.622739, 42.070206 ],
    pop: Decimal128("18405.60"),
    state: 'MA'
  },
  {
    _id: '01013',
    city: 'CHICOPEE',
    loc: [ -72.607962, 42.162046 ],
    pop: Decimal128("28075.20"),
    state: 'MA'
  },
  {
    _id: '01020',
    city: 'CHICOPEE',
    loc: [ -72.576142, 42.176443 ],
    pop: Decimal128("37794.00"),
    state: 'MA'
  },
  {
    _id: '01010',
    city: 'BRIMFIELD',
    loc: [ -72.188455, 42.116543 ],
    pop: Decimal128("4447.20"),
    state: 'MA'
  },
  {
    _id: '01022',
    city: 'WESTOVER AFB',
    loc: [ -72.558657, 42.196672 ],
    pop: Decimal128("2116.80"),
    state: 'MA'
  },
```

```
ITI> db.zips.updateMany (
...     {
...         city : {$type : "string"}
...     },
...     {
...         $rename : {'city':'country'}
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 29353,
  modifiedCount: 29353,
  upsertedCount: 0
}
```

```
I> db.zips.find({country : {$type : "string"}})

{
  _id: '01007',
  loc: [ -72.410953, 42.275103 ],
  pop: Decimal128("12694.80"),
  state: 'MA',
  country: 'BELCHERTOWN'
},
{
  _id: '01001',
  loc: [ -72.622739, 42.070206 ],
  pop: Decimal128("18405.60"),
  state: 'MA',
  country: 'AGAWAM'
},
{
  _id: '01013',
  loc: [ -72.607962, 42.162046 ],
  pop: Decimal128("28075.20"),
  state: 'MA',
  country: 'CHICOPEE'
},
{
  _id: '01020',
  loc: [ -72.576142, 42.176443 ],
  pop: Decimal128("37794.00"),
  state: 'MA',
  country: 'CHICOPEE'
},
{
  _id: '01010',
  loc: [ -72.188455, 42.116543 ],
  pop: Decimal128("4447.20"),
  state: 'MA',
  country: 'BRIMFIELD'
},
{
  _id: '01022',
  loc: [ -72.558657, 42.196672 ],
  pop: Decimal128("2116.80"),
  state: 'MA',
  country: 'WESTOVER AFB'
},
```

```
ITI> db.zips.find({city : {$type : "string"}})
```

Part2

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

```
ITI> db.zips.aggregate([
...     {$match: { state : {$in: ['PA','MA']} } },
...     {$group: { _id:"$state" , total :{$sum:"$pop"} }}
... ])
[
  { _id: 'MA', total: Decimal128("7219710.00") },
  { _id: 'PA', total: Decimal128("14257971.60") }
]
```

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

```
ITI> db.zips.aggregate([
...     { $match : { state : {$nin:['PA','MA']} } },
...     { $limit : 5}
... ])
[
  {
    _id: '99503',
    loc: [ -149.893844, 61.189953 ],
    pop: 12534,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99502',
    loc: [ -150.093943, 61.096163 ],
    pop: 15891,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99501',
    loc: [ -149.876077, 61.211571 ],
    pop: 14436,
    state: 'AK',
    country: 'ANCHORAGE'
  },
  {
    _id: '99505',
    loc: [ -149.675454, 61.275256 ],
    pop: 7979,
    state: 'AK',
    country: 'FORT RICHARDSON'
  },
  {
    _id: '99507',
    loc: [ -149.828912, 61.153543 ],
    pop: 20128,
    state: 'AK',
    country: 'ANCHORAGE'
  }
]
```

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

```
ITI> db.zips.aggregate([
...     { $match:{state:"AK" , "loc.1" : {$gte:55} , "loc.1" : {$lte:65} } },
...     { $group: {_id:"$state" , "Total Population" : {$sum : "$pop"}}}
... ])
[ { _id: 'AK', 'Total Population': 528427 } ]
```

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

```
ITI> db.zips.aggregate([
...     { $match: { state: {$in:['AK','PA']} } },
...     {$sort:{pop:1}},
...     {$skip:7}
... ])
[
  {
    _id: '99770',
    loc: [ -158.534287, 65.713537 ],
    pop: 0,
    state: 'AK',
    country: 'SELAWIK'
  },
  {
    _id: '99773',
    loc: [ -157.613496, 66.958141 ],
    pop: 0,
    state: 'AK',
    country: 'SHUNGNAK'
  },
  {
    _id: '15744',
    loc: [ -79.093987, 40.921432 ],
    pop: Decimal128("0.00"),
    state: 'PA',
    check: true,
    country: 'HAMILTON'
  },
  {
    _id: '19113',
    loc: [ -75.275196, 39.864998 ],
    pop: Decimal128("0.00"),
    state: 'PA',
    check: true,
    country: 'PHILADELPHIA'
  },
  {
    _id: '99575',
    loc: [ -158.002483, 61.818072 ],
    pop: 1,
    state: 'AK',
    country: 'CROOKED CREEK'
  },
  {
    _id: '99692',
    loc: [ -167.510656, 53.362757 ],
    pop: 3,
    state: 'AK',
    country: 'DUTCH HARBOR'
  },
```

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

```
ITI> db.zips.aggregate([
...     { $match: {} },
...     { $group: {  _id:"$state" , Minimum:{$min:"$pop"} , Maximum:{$max:"$pop"}  } },
...     { $out: "mypop"}
... ])
```

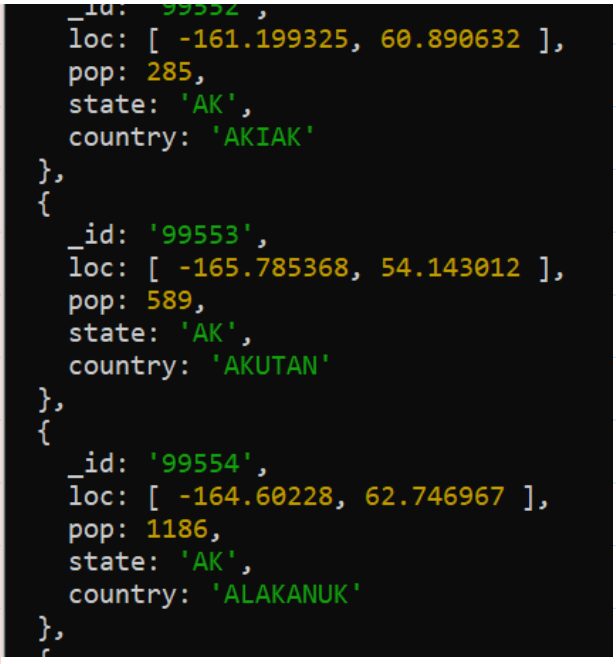
```
ITI> db.mypop.find()
[
  {
    _id: 'MT',
    Minimum: Decimal128("8.40"),
    Maximum: Decimal128("48145.20")
  },
  {
    _id: 'MO',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("65992.80")
  },
  {
    _id: 'TX',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("95355.60")
  },
  {
    _id: 'MN',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("61705.20")
  },
  {
    _id: 'ID',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("49094.40")
  },
  {
    _id: 'OR',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("57608.40")
  },
  {
    _id: 'GA',
    Minimum: Decimal128("0.00"),
    Maximum: Decimal128("70375.20")
  },
  { _id: 'AK', Minimum: 0, Maximum: 32383 },
  {
    _id: 'DC',
    Minimum: Decimal128("13.20"),
    Maximum: Decimal128("75508.80")
  },
  {
    _id: 'TN',
    Minimum: Decimal128("2.40"),
    Maximum: Decimal128("72609.60")
  },
```

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

```
ITI> db.zips.aggregate([
...     { $match: {} },
...     { $group : { _id:"$state" , "Average":{$avg:"$pop"} } },
...     { $project: { "Average Population": { $toLong:{$ceil:"$Average"} } } }
... ])
[
  { _id: 'MD', 'Average Population': Long("13662") },
  { _id: 'FL', 'Average Population': Long("18936") },
  { _id: 'CA', 'Average Population': Long("23553") },
  { _id: 'NY', 'Average Population': Long("11280") },
  { _id: 'ME', 'Average Population': Long("3591") },
  { _id: 'NJ', 'Average Population': Long("17179") },
  { _id: 'AL', 'Average Population': Long("8552") },
  { _id: 'HI', 'Average Population': Long("16624") },
  { _id: 'NH', 'Average Population': Long("6106") },
  { _id: 'WI', 'Average Population': Long("8199") },
  { _id: 'UT', 'Average Population': Long("10085") },
  { _id: 'MN', 'Average Population': Long("5950") },
  { _id: 'AK', 'Average Population': Long("2793") },
  { _id: 'ID', 'Average Population': Long("4952") },
  { _id: 'GA', 'Average Population': Long("12243") },
  { _id: 'DC', 'Average Population': Long("30345") },
  { _id: 'TN', 'Average Population': Long("10055") },
  { _id: 'MS', 'Average Population': Long("8507") },
  { _id: 'ND', 'Average Population': Long("1959") },
  { _id: 'IL', 'Average Population': Long("11086") }
]
```

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

```
ITI> db.zips.aggregate([
...     { $match: {} },
...     { $sort: {state:1,country: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,
```



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

```
ITI> db.zips.aggregate([
...     { $match: {} },
...     { $sort: {state:-1,country:-1} }
... ])
[
  {
    _id: '82244',
    loc: [ -104.353507, 41.912018 ],
    pop: Decimal128("808.80"),
    state: 'WY',
    country: 'YODER'
  },
  {
    _id: '82732',
    loc: [ -105.532327, 43.829349 ],
    pop: Decimal128("2558.40"),
    state: 'WY',
    country: 'WRIGHT'
  },
  {
    _id: '82401',
    loc: [ -107.95626, 44.013796 ],
    pop: Decimal128("9231.60"),
    state: 'WY',
    country: 'WORLAND'
  },
  {
    _id: '83014',
    loc: [ -110.874199, 43.49922 ],
    pop: Decimal128("1318.80"),
    state: 'WY',
    country: 'WILSON'
  },
  {
    _id: '82201',
    loc: [ -104.967852, 42.049467 ],
    pop: Decimal128("7142.40"),
    state: 'WY',
    country: 'WHEATLAND'
  },
  {
    _id: '82450',
    loc: [ -109.432629, 44.47967 ],
    pop: Decimal128("256.80"),
    state: 'WY',
    country: 'WAPITI'
  },
```

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

```
ITI> db.zips.aggregate([
...     { $match: { pop:{$gte:25000} , state:{$in:['CA','NY']} } },
...     { $project: { "Average Population": { $toLong:{$ceil:{$avg:"$pop"}} } } }
... ])
[
  { _id: '90001', 'Average Population': Long("62210") },
  { _id: '90002', 'Average Population': Long("48755") },
  { _id: '90006', 'Average Population': Long("76067") },
  { _id: '90004', 'Average Population': Long("76875") },
  { _id: '90003', 'Average Population': Long("64726") },
  { _id: '90007', 'Average Population': Long("56382") },
  { _id: '90008', 'Average Population': Long("39688") },
  { _id: '90005', 'Average Population': Long("43037") },
  { _id: '90011', 'Average Population': Long("115289") },
  { _id: '90012', 'Average Population': Long("34222") },
  { _id: '90019', 'Average Population': Long("77996") },
  { _id: '90018', 'Average Population': Long("57921") },
  { _id: '90020', 'Average Population': Long("41912") },
  { _id: '90022', 'Average Population': Long("78078") },
  { _id: '90017', 'Average Population': Long("26148") },
  { _id: '90023', 'Average Population': Long("56564") },
  { _id: '90024', 'Average Population': Long("46044") },
  { _id: '90026', 'Average Population': Long("89702") },
  { _id: '90027', 'Average Population': Long("60581") },
  { _id: '90028', 'Average Population': Long("36183") }
]
```

10.Return the average populations for cities in each state

```
ITI> db.zips.aggregate([
...     { $match: {} },
...     { $group : { _id:{state:"$state",city:"$country"} , "Average":{$avg:"$pop"} } },
...     { $project: { "Average Population": { $toLong:{$ceil:"$Average"} } } }
... ])
[
  {
    _id: { state: 'MD', city: 'HALETHORPE' },
    'Average Population': Long("54516")
  },
  {
    _id: { state: 'NJ', city: 'FAIR LAWN' },
    'Average Population': Long("36627")
  },
  {
    _id: { state: 'SD', city: 'BALTIC' },
    'Average Population': Long("2207")
  },
  {
    _id: { state: 'SD', city: 'WARNER' },
    'Average Population': Long("1272")
  },
  {
    _id: { state: 'ND', city: 'HURDSFIELD' },
    'Average Population': Long("233")
  },
  {
    _id: { state: 'CA', city: 'LAKEHEAD' },
    'Average Population': Long("2051")
  },
  {
    _id: { state: 'WA', city: 'HAZEL DELL' },
    'Average Population': Long("19786")
  },
  {
    _id: { state: 'NE', city: 'MILBURN' },
    'Average Population': Long("680")
  },
  {
    _id: { state: 'IL', city: 'HERRIN' },
    'Average Population': Long("16680")
  },
  {
    _id: { state: 'VA', city: 'JAVA' },
    'Average Population': Long("1334")
  },
  {
    _id: { state: 'NH', city: 'BEEBE RIVER' },
    'Average Population': Long("3363")
  },
  {
    _id: { state: 'MO', city: 'WEATHERBY' },
    'Average Population': Long("701")
  },
```

Agile day 3

02:00:30