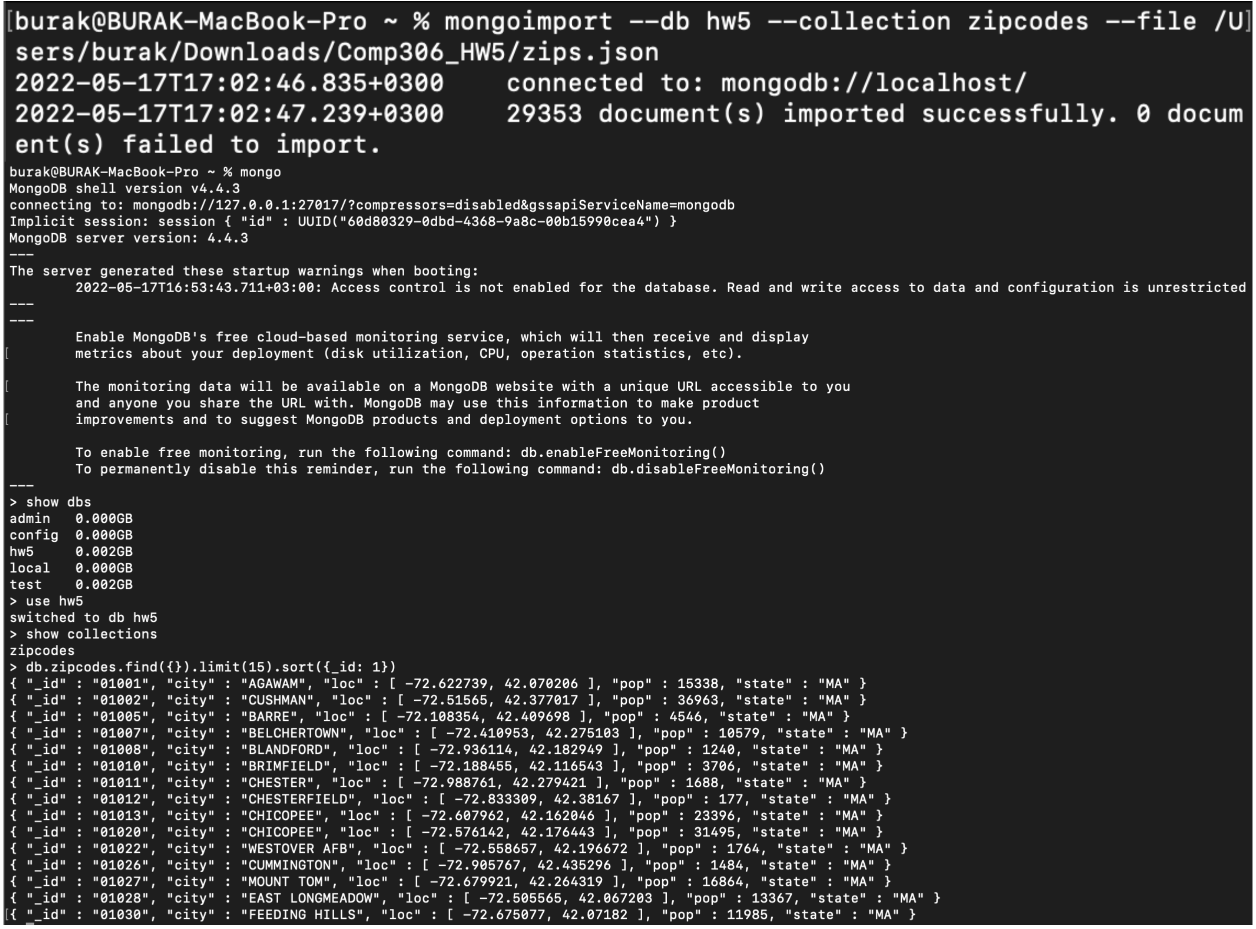
Comp 306 HW#5 Report
Burak Yıldırım 72849

1-

```
db.zipcodes.find({}).limit(15).sort({_id: 1})
```



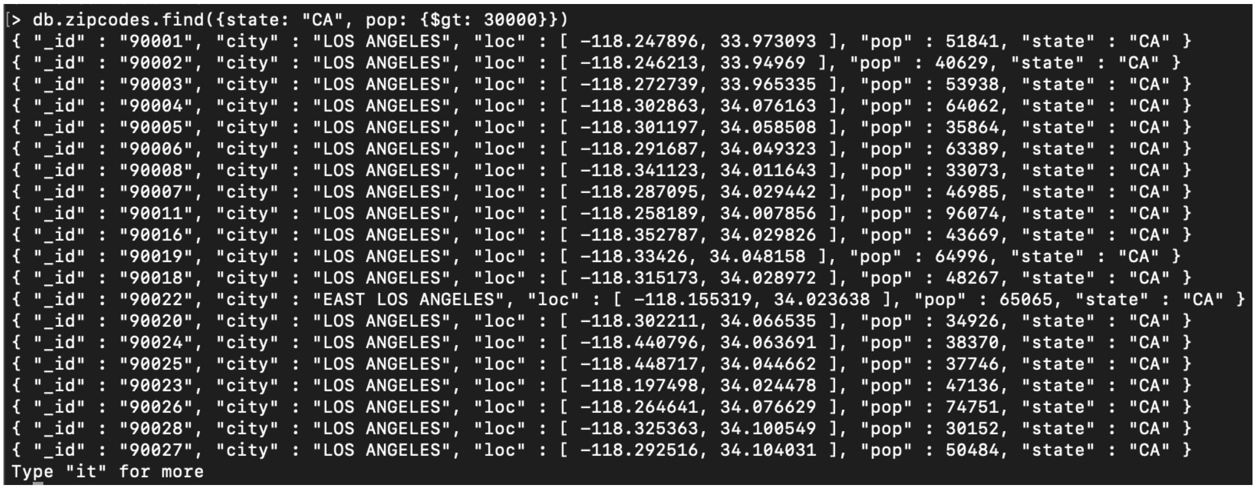
```
[burak@BURAK-MacBook-Pro ~ % mongoimport --db hw5 --collection zipcodes --file /U]
sers/burak/Downloads/Comp306_HW5/zips.json
2022-05-17T17:02:46.835+0300    connected to: mongodb://localhost/
2022-05-17T17:02:47.239+0300    29353 document(s) imported successfully. 0 docum
ent(s) failed to import.
burak@BURAK-MacBook-Pro ~ % mongo
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("60d80329-0dbd-4368-9a8c-00b15990cea4") }
MongoDB server version: 4.4.3
---
The server generated these startup warnings when booting:
        2022-05-17T16:53:43.711+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
[       metrics about your deployment (disk utilization, CPU, operation statistics, etc).

[       The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
[       improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin   0.000GB
config  0.000GB
hw5     0.002GB
local   0.000GB
test    0.002GB
> use hw5
switched to db hw5
> show collections
zipcodes
> db.zipcodes.find({}).limit(15).sort({_id: 1})
{ "_id" : "01001", "city" : "AGAWAM", "loc" : [ -72.622739, 42.070206 ], "pop" : 15338, "state" : "MA" }
{ "_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" : "01008", "city" : "BLANDFORD", "loc" : [ -72.936114, 42.182949 ], "pop" : 1240, "state" : "MA" }
{ "_id" : "01010", "city" : "BRIMFIELD", "loc" : [ -72.188455, 42.116543 ], "pop" : 3706, "state" : "MA" }
{ "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" }
{ "_id" : "01012", "city" : "CHESTERFIELD", "loc" : [ -72.833309, 42.38167 ], "pop" : 177, "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" : "01022", "city" : "WESTOVER AFB", "loc" : [ -72.558657, 42.196672 ], "pop" : 1764, "state" : "MA" }
{ "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
{ "_id" : "01027", "city" : "MOUNT TOM", "loc" : [ -72.679921, 42.264319 ], "pop" : 16864, "state" : "MA" }
{ "_id" : "01028", "city" : "EAST LONGMEADOW", "loc" : [ -72.505565, 42.067203 ], "pop" : 13367, "state" : "MA" }
[{ "_id" : "01030", "city" : "FEEDING HILLS", "loc" : [ -72.675077, 42.07182 ], "pop" : 11985, "state" : "MA" }
```

2-

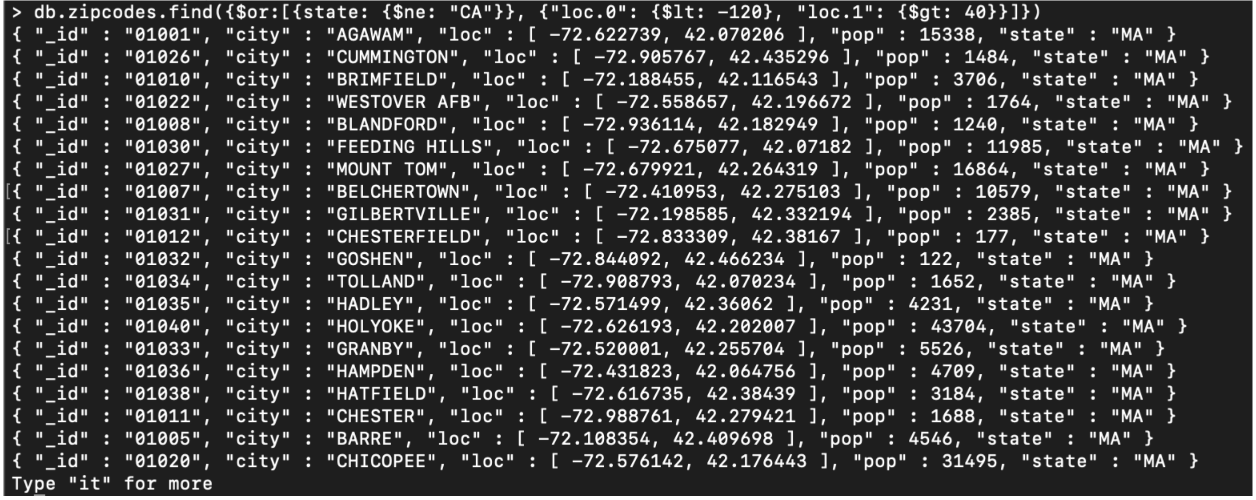
```
db.zipcodes.find({state: "CA", pop: {$gt: 30000}})
```



```
[> db.zipcodes.find({state: "CA", pop: {$gt: 30000}})
 { "_id" : "90001", "city" : "LOS ANGELES", "loc" : [ -118.247896, 33.973093 ], "pop" : 51841, "state" : "CA" }
 { "_id" : "90002", "city" : "LOS ANGELES", "loc" : [ -118.246213, 33.94969 ], "pop" : 40629, "state" : "CA" }
 { "_id" : "90003", "city" : "LOS ANGELES", "loc" : [ -118.272739, 33.965335 ], "pop" : 53938, "state" : "CA" }
 { "_id" : "90004", "city" : "LOS ANGELES", "loc" : [ -118.302863, 34.076163 ], "pop" : 64062, "state" : "CA" }
 { "_id" : "90005", "city" : "LOS ANGELES", "loc" : [ -118.301197, 34.058508 ], "pop" : 35864, "state" : "CA" }
 { "_id" : "90006", "city" : "LOS ANGELES", "loc" : [ -118.291687, 34.049323 ], "pop" : 63389, "state" : "CA" }
 { "_id" : "90008", "city" : "LOS ANGELES", "loc" : [ -118.341123, 34.011643 ], "pop" : 33073, "state" : "CA" }
 { "_id" : "90007", "city" : "LOS ANGELES", "loc" : [ -118.287095, 34.029442 ], "pop" : 46985, "state" : "CA" }
 { "_id" : "90011", "city" : "LOS ANGELES", "loc" : [ -118.258189, 34.007856 ], "pop" : 96074, "state" : "CA" }
 { "_id" : "90016", "city" : "LOS ANGELES", "loc" : [ -118.352787, 34.029826 ], "pop" : 43669, "state" : "CA" }
 { "_id" : "90019", "city" : "LOS ANGELES", "loc" : [ -118.33426, 34.048158 ], "pop" : 64996, "state" : "CA" }
 { "_id" : "90018", "city" : "LOS ANGELES", "loc" : [ -118.315173, 34.028972 ], "pop" : 48267, "state" : "CA" }
 { "_id" : "90022", "city" : "EAST LOS ANGELES", "loc" : [ -118.155319, 34.023638 ], "pop" : 65065, "state" : "CA" }
 { "_id" : "90020", "city" : "LOS ANGELES", "loc" : [ -118.302211, 34.066535 ], "pop" : 34926, "state" : "CA" }
 { "_id" : "90024", "city" : "LOS ANGELES", "loc" : [ -118.440796, 34.063691 ], "pop" : 38370, "state" : "CA" }
 { "_id" : "90025", "city" : "LOS ANGELES", "loc" : [ -118.448717, 34.044662 ], "pop" : 37746, "state" : "CA" }
 { "_id" : "90023", "city" : "LOS ANGELES", "loc" : [ -118.197498, 34.024478 ], "pop" : 47136, "state" : "CA" }
 { "_id" : "90026", "city" : "LOS ANGELES", "loc" : [ -118.264641, 34.076629 ], "pop" : 74751, "state" : "CA" }
 { "_id" : "90028", "city" : "LOS ANGELES", "loc" : [ -118.325363, 34.100549 ], "pop" : 30152, "state" : "CA" }
 { "_id" : "90027", "city" : "LOS ANGELES", "loc" : [ -118.292516, 34.104031 ], "pop" : 50484, "state" : "CA" }
 Type "it" for more
```

3-

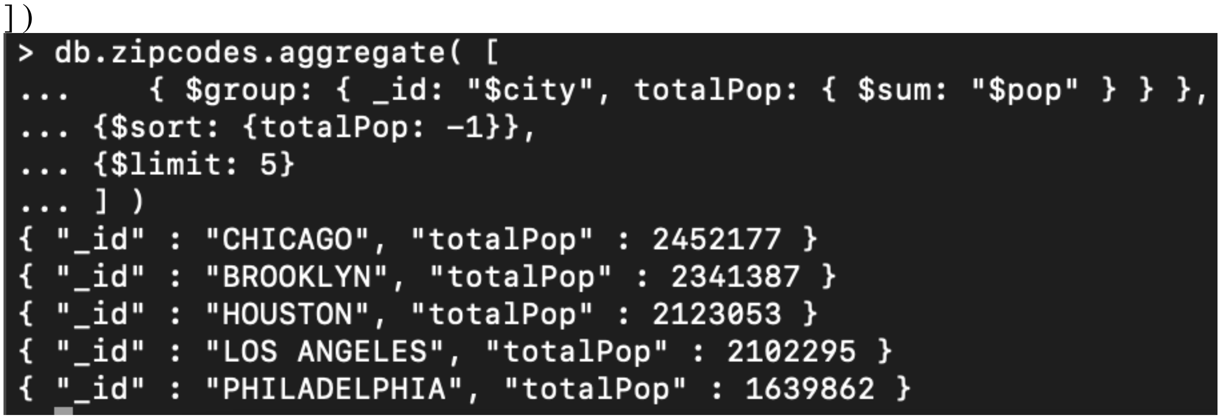
```
db.zipcodes.find({$or:[{state: {$ne: "CA"}}, {"loc.0": {$lt: -120},
"loc.1": {$gt: 40}}]})
```



```
 > db.zipcodes.find({$or:[{state: {$ne: "CA"}}, {"loc.0": {$lt: -120}, "loc.1": {$gt: 40}}]})
 { "_id" : "01001", "city" : "AGAWAM", "loc" : [ -72.622739, 42.070206 ], "pop" : 15338, "state" : "MA" }
 { "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
 { "_id" : "01010", "city" : "BRIMFIELD", "loc" : [ -72.188455, 42.116543 ], "pop" : 3706, "state" : "MA" }
 { "_id" : "01022", "city" : "WESTOVER AFB", "loc" : [ -72.558657, 42.196672 ], "pop" : 1764, "state" : "MA" }
 { "_id" : "01008", "city" : "BLANDFORD", "loc" : [ -72.936114, 42.182949 ], "pop" : 1240, "state" : "MA" }
 { "_id" : "01030", "city" : "FEEDING HILLS", "loc" : [ -72.675077, 42.07182 ], "pop" : 11985, "state" : "MA" }
 { "_id" : "01027", "city" : "MOUNT TOM", "loc" : [ -72.679921, 42.264319 ], "pop" : 16864, "state" : "MA" }
[{ "_id" : "01007", "city" : "BELCHERTOWN", "loc" : [ -72.410953, 42.275103 ], "pop" : 10579, "state" : "MA" }
 { "_id" : "01031", "city" : "GILBERTVILLE", "loc" : [ -72.198585, 42.332194 ], "pop" : 2385, "state" : "MA" }
[{ "_id" : "01012", "city" : "CHESTERFIELD", "loc" : [ -72.833309, 42.38167 ], "pop" : 177, "state" : "MA" }
 { "_id" : "01032", "city" : "GOSHEN", "loc" : [ -72.844092, 42.466234 ], "pop" : 122, "state" : "MA" }
 { "_id" : "01034", "city" : "TOLLAND", "loc" : [ -72.908793, 42.070234 ], "pop" : 1652, "state" : "MA" }
 { "_id" : "01035", "city" : "HADLEY", "loc" : [ -72.571499, 42.36062 ], "pop" : 4231, "state" : "MA" }
 { "_id" : "01040", "city" : "HOLYOKE", "loc" : [ -72.626193, 42.202007 ], "pop" : 43704, "state" : "MA" }
 { "_id" : "01033", "city" : "GRANBY", "loc" : [ -72.520001, 42.255704 ], "pop" : 5526, "state" : "MA" }
 { "_id" : "01036", "city" : "HAMPDEN", "loc" : [ -72.431823, 42.064756 ], "pop" : 4709, "state" : "MA" }
 { "_id" : "01038", "city" : "HATFIELD", "loc" : [ -72.616735, 42.38439 ], "pop" : 3184, "state" : "MA" }
 { "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" }
 { "_id" : "01005", "city" : "BARRE", "loc" : [ -72.108354, 42.409698 ], "pop" : 4546, "state" : "MA" }
 { "_id" : "01020", "city" : "CHICOPEE", "loc" : [ -72.576142, 42.176443 ], "pop" : 31495, "state" : "MA" }
 Type "it" for more
```

4-

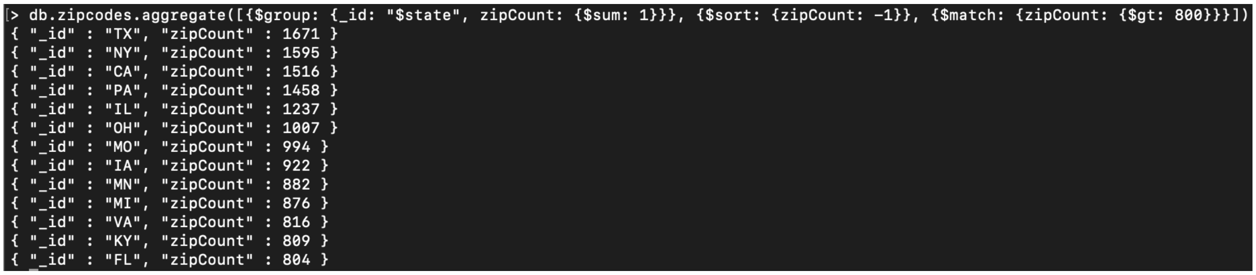
```
db.zipcodes.aggregate( [
   { $group: { _id: "$city", totalPop: { $sum: "$pop" } } },
{$sort: {totalPop: -1}},
{$limit: 5}
] )
```



```
> db.zipcodes.aggregate( [
...    { $group: { _id: "$city", totalPop: { $sum: "$pop" } } },
... {$sort: {totalPop: -1}},
... {$limit: 5}
... ] )
{ "_id" : "CHICAGO", "totalPop" : 2452177 }
{ "_id" : "BROOKLYN", "totalPop" : 2341387 }
{ "_id" : "HOUSTON", "totalPop" : 2123053 }
{ "_id" : "LOS ANGELES", "totalPop" : 2102295 }
{ "_id" : "PHILADELPHIA", "totalPop" : 1639862 }
```

5-

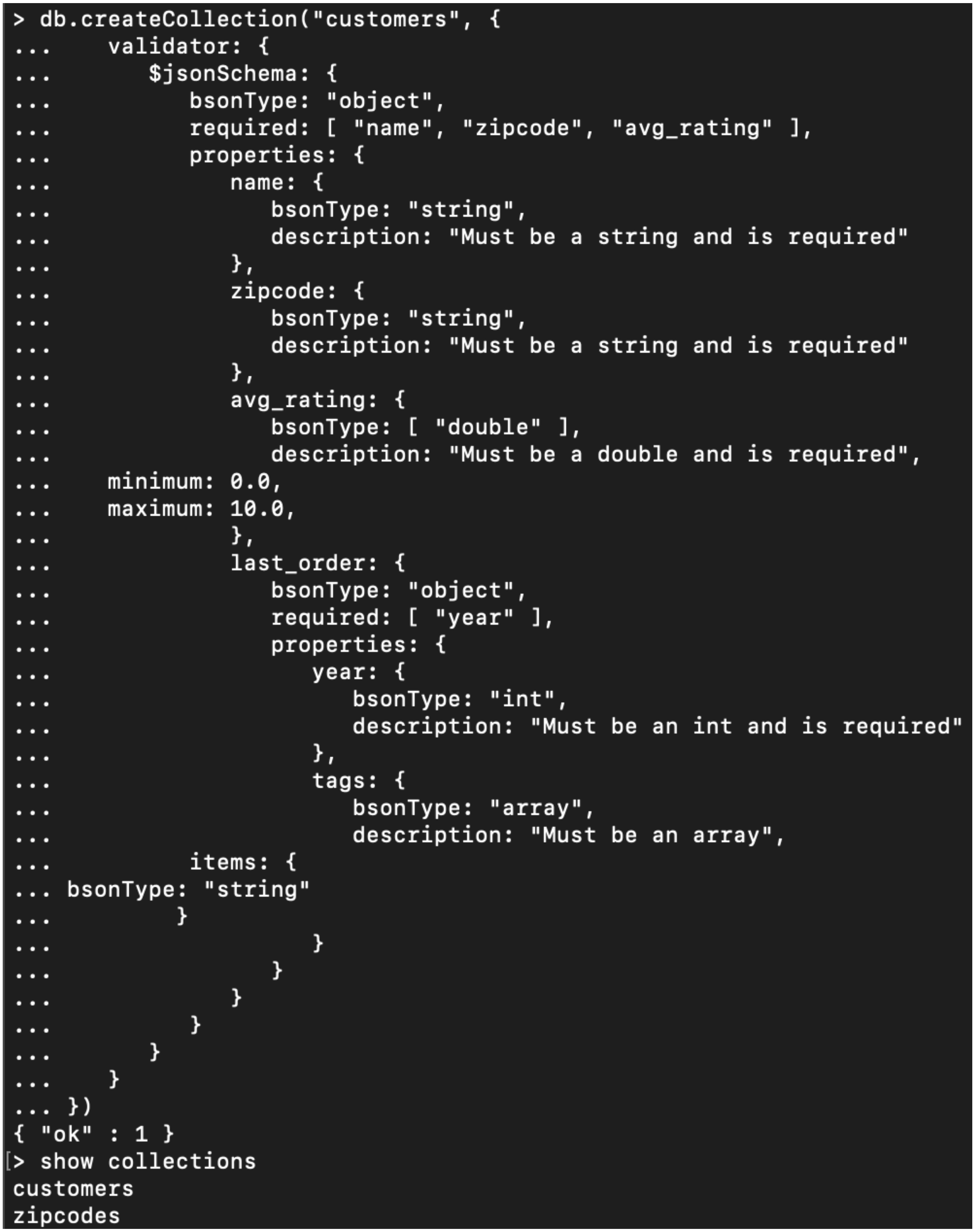
```
db.zipcodes.aggregate([{$group: {_id: "$state", zipCount: {$sum: 1}}},
{$sort: {zipCount: -1}}, {$match: {zipCount: {$gt: 800}}}])
```



```
> db.zipcodes.aggregate([{$group: {_id: "$state", zipCount: {$sum: 1}}}, {$sort: {zipCount: -1}}, {$match: {zipCount: {$gt: 800}}}])
{ "_id" : "TX", "zipCount" : 1671 }
{ "_id" : "NY", "zipCount" : 1595 }
{ "_id" : "CA", "zipCount" : 1516 }
{ "_id" : "PA", "zipCount" : 1458 }
{ "_id" : "IL", "zipCount" : 1237 }
{ "_id" : "OH", "zipCount" : 1007 }
{ "_id" : "MO", "zipCount" : 994 }
{ "_id" : "IA", "zipCount" : 922 }
{ "_id" : "MN", "zipCount" : 882 }
{ "_id" : "MI", "zipCount" : 876 }
{ "_id" : "VA", "zipCount" : 816 }
{ "_id" : "KY", "zipCount" : 809 }
{ "_id" : "FL", "zipCount" : 804 }
```

6-

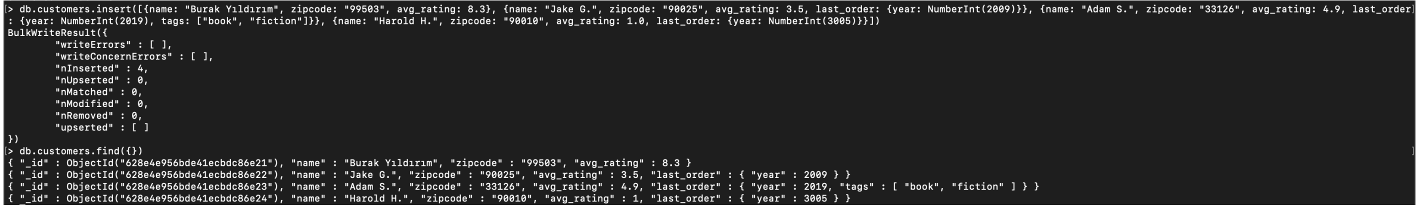
```
db.createCollection("customers", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "name", "zipcode", "avg_rating" ],
      properties: {
        name: {
          bsonType: "string",
          description: "Must be a string and is required"
        },
        zipcode: {
          bsonType: "string",
          description: "Must be a string and is required"
        },
        avg_rating: {
          bsonType: [ "double" ],
          description: "Must be a double and is required",
          minimum: 0.0,
          maximum: 10.0,
        },
        last_order: {
          bsonType: "object",
          required: [ "year" ],
          properties: {
            year: {
              bsonType: "int",
              description: "Must be an int and is required"
            },
            tags: {
              bsonType: "array",
              description: "Must be an array",
              items: {
                bsonType: "string"
      }
            }
          }
        }
      }
    }
  }
})
```



```
> db.createCollection("customers", {
...    validator: {
...       $jsonSchema: {
...          bsonType: "object",
...          required: [ "name", "zipcode", "avg_rating" ],
...          properties: {
...             name: {
...                bsonType: "string",
...                description: "Must be a string and is required"
...             },
...             zipcode: {
...                bsonType: "string",
...                description: "Must be a string and is required"
...             },
...             avg_rating: {
...                bsonType: [ "double" ],
...                description: "Must be a double and is required",
...    minimum: 0.0,
...    maximum: 10.0,
...             },
...             last_order: {
...                bsonType: "object",
...                required: [ "year" ],
...                properties: {
...                   year: {
...                      bsonType: "int",
...                      description: "Must be an int and is required"
...                   },
...                   tags: {
...                      bsonType: "array",
...                      description: "Must be an array",
...          items: {
... bsonType: "string"
...         }
...                   }
...                }
...             }
...          }
...       }
...    }
... })
{ "ok" : 1 }
[> show collections
customers
zipcodes
```

7-

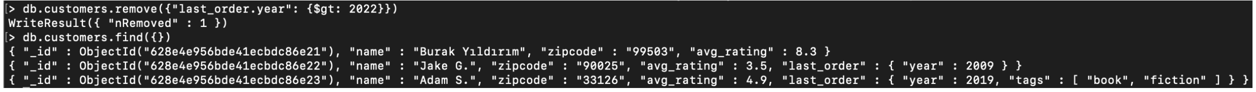
```
db.customers.insert([{name: "Burak Yıldırım", zipcode: "99503",
avg_rating: 8.3}, {name: "Jake G.", zipcode: "90025", avg_rating: 3.5,
last_order: {year: NumberInt(2009)}}, {name: "Adam S.", zipcode:
"33126", avg_rating: 4.9, last_order: {year: NumberInt(2019), tags:
["book", "fiction"]}}, {name: "Harold H.", zipcode: "90010",
avg_rating: 1.0, last_order: {year: NumberInt(3005)}}])
```



```
> db.customers.insert([{name: "Burak Yıldırım", zipcode: "99503", avg_rating: 8.3}, {name: "Jake G.", zipcode: "90025", avg_rating: 3.5, last_order: {year: NumberInt(2009)}}, {name: "Adam S.", zipcode: "33126", avg_rating: 4.9, last_order
: {year: NumberInt(2019), tags: ["book", "fiction"]}}, {name: "Harold H.", zipcode: "90010", avg_rating: 1.0, last_order: {year: NumberInt(3005)}}])
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 4,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.customers.find({})
{ "_id" : ObjectId("628e4e956bde41ecbdc86e21"), "name" : "Burak Yıldırım", "zipcode" : "99503", "avg_rating" : 8.3 }
{ "_id" : ObjectId("628e4e956bde41ecbdc86e22"), "name" : "Jake G.", "zipcode" : "90025", "avg_rating" : 3.5, "last_order" : { "year" : 2009 } }
{ "_id" : ObjectId("628e4e956bde41ecbdc86e23"), "name" : "Adam S.", "zipcode" : "33126", "avg_rating" : 4.9, "last_order" : { "year" : 2019, "tags" : [ "book", "fiction" ] } }
{ "_id" : ObjectId("628e4e956bde41ecbdc86e24"), "name" : "Harold H.", "zipcode" : "90010", "avg_rating" : 1, "last_order" : { "year" : 3005 } }
```

8-

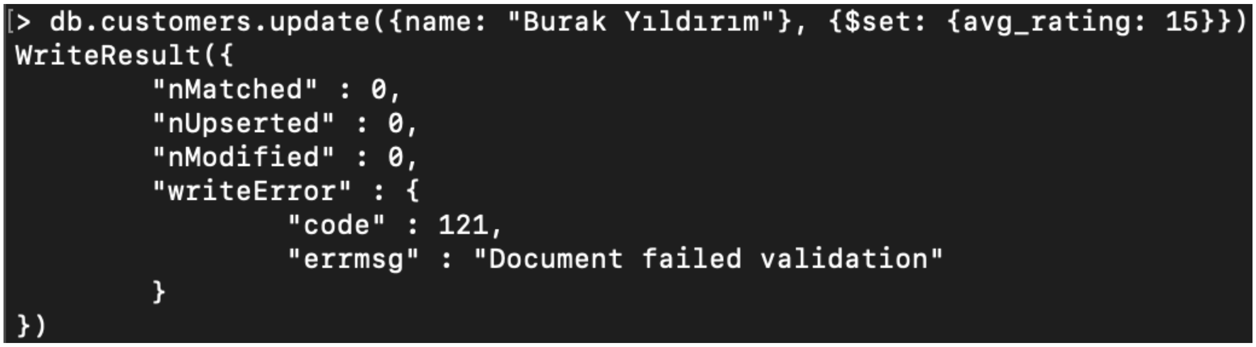
```
db.customers.remove({"last_order.year": {$gt: 2022}})
```



```
> db.customers.remove({"last_order.year": {$gt: 2022}})
WriteResult({ "nRemoved" : 1 })
> db.customers.find({})
{ "_id" : ObjectId("628e4e956bde41ecbdc86e21"), "name" : "Burak Yıldırım", "zipcode" : "99503", "avg_rating" : 8.3 }
{ "_id" : ObjectId("628e4e956bde41ecbdc86e22"), "name" : "Jake G.", "zipcode" : "90025", "avg_rating" : 3.5, "last_order" : { "year" : 2009 } }
{ "_id" : ObjectId("628e4e956bde41ecbdc86e23"), "name" : "Adam S.", "zipcode" : "33126", "avg_rating" : 4.9, "last_order" : { "year" : 2019, "tags" : [ "book", "fiction" ] } }
```

9-

```
db.customers.update({name: "Burak Yıldırım"}, {$set: {avg_rating:
15}})
```

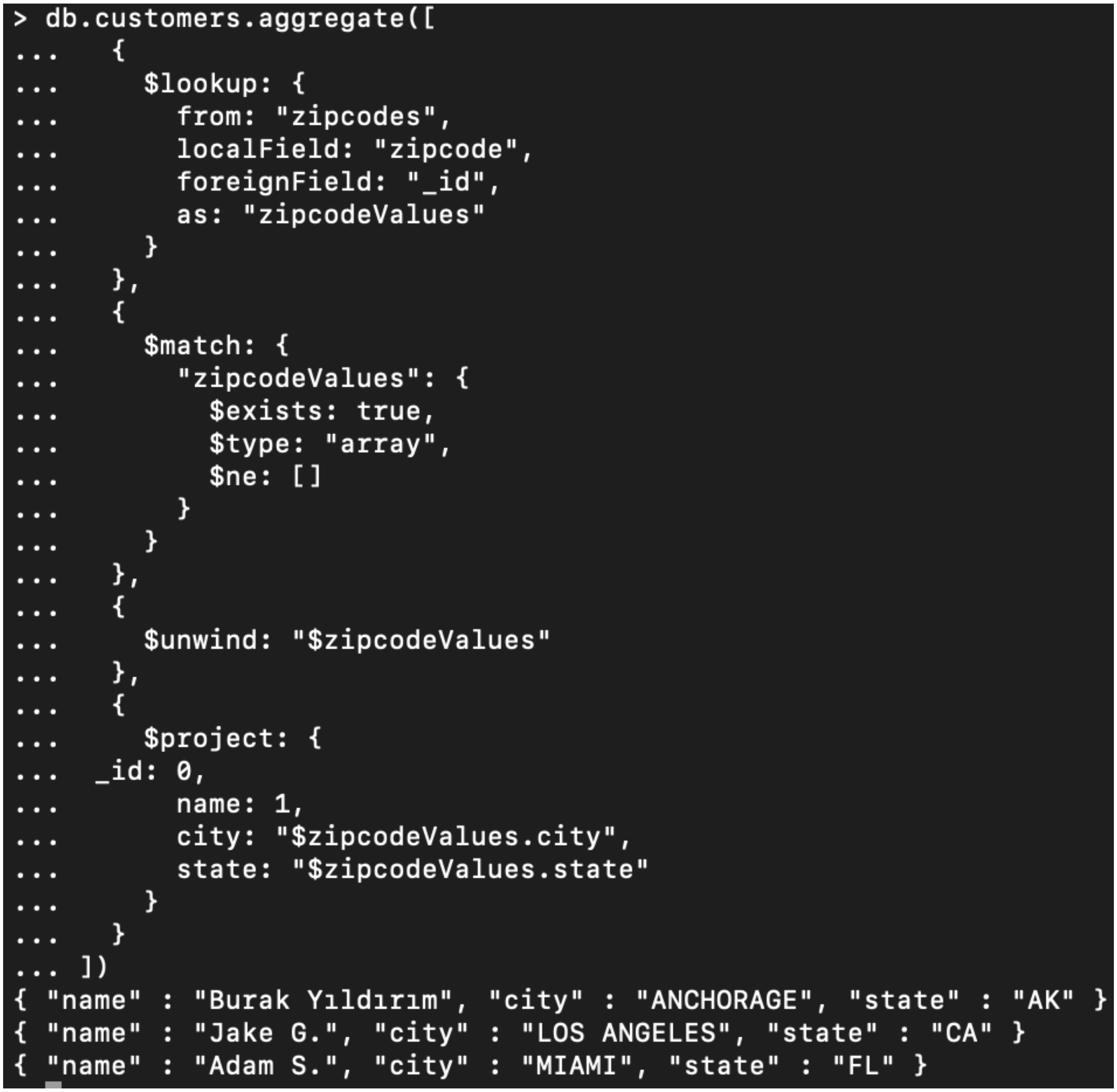


```
> db.customers.update({name: "Burak Yıldırım"}, {$set: {avg_rating: 15}})
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 0,
        "nModified" : 0,
        "writeError" : {
                "code" : 121,
                "errmsg" : "Document failed validation"
        }
})
```

When I tried to update the document with my name and set its average rating to 15 it gave Document failed validation error. Because it doesn’t satisfy the schema validation since average rating of 15 isn’t between 0 and 10.

10-

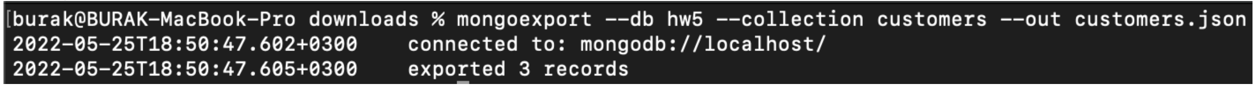
```
db.customers.aggregate([
  {
    $lookup: {
      from: "zipcodes",
      localField: "zipcode",
      foreignField: "_id",
      as: "zipcodeValues"
    }
  },
  {
    $match: {
      "zipcodeValues": {
        $exists: true,
        $type: "array",
        $ne: []
      }
    }
  },
  {
    $unwind: "$zipcodeValues"
  },
  {
    $project: {
      _id: 0,
      name: 1,
      city: "$zipcodeValues.city",
      state: "$zipcodeValues.state"
    }
  }
])
```



```
> db.customers.aggregate([
...   {
...     $lookup: {
...       from: "zipcodes",
...       localField: "zipcode",
...       foreignField: "_id",
...       as: "zipcodeValues"
...     }
...   },
...   {
...     $match: {
...       "zipcodeValues": {
...         $exists: true,
...         $type: "array",
...         $ne: []
...       }
...     }
...   },
...   {
...     $unwind: "$zipcodeValues"
...   },
...   {
...     $project: {
...  _id: 0,
...       name: 1,
...       city: "$zipcodeValues.city",
...       state: "$zipcodeValues.state"
...     }
...   }
... ])
{ "name" : "Burak Yıldırım", "city" : "ANCHORAGE", "state" : "AK" }
{ "name" : "Jake G.", "city" : "LOS ANGELES", "state" : "CA" }
{ "name" : "Adam S.", "city" : "MIAMI", "state" : "FL" }
```

11-

```
mongoexport --db hw5 --collection customers --out customers.json
```



```
[burak@BURAK-MacBook-Pro downloads % mongoexport --db hw5 --collection customers --out customers.json
2022-05-25T18:50:47.602+0300    connected to: mongodb://localhost/
2022-05-25T18:50:47.605+0300    exported 3 records
```