

HW5

COMP 306

Gülbarin Maçın

Q1)

```
C:\Program Files\MongoDB\Tools\100\bin>mongoimport --db hw5 --collection zipcodes --authenticationDatabase admin --username username --password 1234 --drop --file F:\zips.json
2023-06-04T19:12:33.381+0300    connected to: mongodb://localhost/
2023-06-04T19:12:33.485+0300    dropping: hw5.zipcodes
2023-06-04T19:12:34.386+0300    29353 document(s) imported successfully. 0 document(s) failed to import.
```

```
C:\Program Files\MongoDB\Tools\100\bin>mongosh mongodb://localhost:27017/authSource=admin --username username --password
Enter password: ****
Current Mongosh Log ID: 647cb84d58973453b2f8fd85
Connecting to:      mongodb://<credentials>@localhost:27017/authSource=admin?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.9.1
MongoServerError: Authentication failed.

C:\Program Files\MongoDB\Tools\100\bin>mongosh mongodb://localhost:27017/?authSource=admin --username username --password
Enter password: ****
Current Mongosh Log ID: 647cb86fb61fe88076e38b5f
Connecting to:      mongodb://<credentials>@localhost:27017/?authSource=admin&directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.9.1
Using MongoDB:      6.0.6
Using Mongosh:       1.9.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
  The server generated these startup warnings when booting
  2023-06-04T17:59:09.464+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  -----

test> show collection
MongoSHInvalidInputError: [COMMON-10001] 'collection' is not a valid argument for "show".
test> show collections
zips
test> use hw5
switched to db hw5
hw5> show collections
zipcodes
```

```

    @.zipcodes.find(0).limit(10)

    {
      "id": "91000",
      "city": "COLUMBIA",
      "loc": {
        "lat": 37.226978,
        "lon": 42.413296
      },
      "pop": 1844,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "BIRMINGHAM",
      "loc": {
        "lat": 33.520614,
        "lon": 42.182649
      },
      "pop": 1240,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "MEMPHIS",
      "loc": {
        "lat": 37.18845,
        "lon": 42.116541
      },
      "pop": 198,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.573342,
        "lon": 42.176441
      },
      "pop": 122,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "NASHVILLE",
      "loc": {
        "lat": 37.79921,
        "lon": 42.264319
      },
      "pop": 1884,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "KANSAS",
      "loc": {
        "lat": 37.749561,
        "lon": 42.067281
      },
      "pop": 1337,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "FARMINGTON HILLS",
      "loc": {
        "lat": 42.475877,
        "lon": 42.07182
      },
      "pop": 1108,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.129585,
        "lon": 42.332194
      },
      "pop": 385,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.248492,
        "lon": 42.466234
      },
      "pop": 52,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.51009,
        "lon": 42.255704
      },
      "pop": 528,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "COLUMBIA",
      "loc": {
        "lat": 37.247348,
        "lon": 42.377917
      },
      "pop": 1693,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "COLUMBIA",
      "loc": {
        "lat": 36.90973,
        "lon": 42.079234
      },
      "pop": 161,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "NASHVILLE",
      "loc": {
        "lat": 37.57459,
        "lon": 42.16062
      },
      "pop": 133,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.623373,
        "lon": 42.208071
      },
      "pop": 12,
      "state": "TN"
    },
    {
      "id": "91000",
      "city": "CHATTANOOGA",
      "loc": {
        "lat": 37.87341,
        "lon": 42.263361
      },
      "pop": 284,
      "state": "TN"
    }
  ]
}

```

Q2)

```
hw5> db.zipcodes.find({ state: "CA", pop: { $gt: 50000}, loc: {$elemMatch: {$gt: 35}}}).sort({pop: -1}).limit(5)
[
  {
    _id: '94501',
    city: 'COAST GUARD ISLA',
    loc: [ -122.260516, 37.764783 ],
    pop: 76110,
    state: 'CA'
  },
  {
    _id: '94110',
    city: 'SAN FRANCISCO',
    loc: [ -122.415344, 37.750858 ],
    pop: 70770,
    state: 'CA'
  },
  {
    _id: '95351',
    city: 'MODESTO',
    loc: [ -121.006033, 37.625022 ],
    pop: 69275,
    state: 'CA'
  },
  {
    _id: '95076',
    city: 'LA SELVA BEACH',
    loc: [ -121.763437, 36.920515 ],
    pop: 68295,
    state: 'CA'
  },
  {
    _id: '94533',
    city: 'FAIRFIELD',
    loc: [ -122.03565, 38.267084 ],
    pop: 65455,
    state: 'CA'
  }
]
```

Q3)

Function:

```
hw5> db.zipcodes.find({
... $or: [
... { state: {$ne: "CA"}},
... {$and: [
... {"loc.0": {$lt: -120}},
... {"loc.1": {$gt: 40}}
... ]
... }
... ]
... })
```

Output:

```
[
  {
    _id: '01026',
    city: 'CUMMINGTON',
    loc: [ -72.905767, 42.435296 ],
    pop: 1484,
    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: '01020',  
  
city: 'CHICOPEE',  
  
loc: [ -72.576142, 42.176443 ],  
  
pop: 31495,  
  
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'  
  
},  
  
{  
  
_id: '01031',  
  
city: 'GILBERTVILLE',
```

```
loc: [ -72.198585, 42.332194 ],

pop: 2385,

state: 'MA'

},

{

  _id: '01032',

  city: 'GOSHEN',

  loc: [ -72.844092, 42.466234 ],

  pop: 122,

  state: 'MA'

},

{

  _id: '01033',

  city: 'GRANBY',

  loc: [ -72.520001, 42.255704 ],

  pop: 5526,

  state: 'MA'

},

{

  _id: '01002',

  city: 'CUSHMAN',

  loc: [ -72.51565, 42.377017 ],

  pop: 36963,

  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: '01050',
    city: 'HUNTINGTON',
    loc: [ -72.873341, 42.265301 ],
    pop: 2084,
    state: 'MA'
  },
  {
    _id: '01038',
    city: 'HATFIELD',
    loc: [ -72.616735, 42.38439 ],
    pop: 3184,
    state: 'MA'
  },
  {
    _id: '01039',
    city: 'HAYDENVILLE',
    loc: [ -72.703178, 42.381799 ],
    pop: 1387,
    state: 'MA'
  },
  {
    _id: '01036',
    city: 'HAMPDEN',
    loc: [ -72.431823, 42.064756 ],
    pop: 4709,
    state: 'MA'
  },
  {
    _id: '01013',
    city: 'CHICOPEE',
    loc: [ -72.607962, 42.162046 ],
    pop: 23396,
    state: 'MA'
  },
  {
    _id: '01011',
    city: 'CHESTER',
    loc: [ -72.988761, 42.279421 ],
    pop: 1688,
    state: 'MA'
  }
}
```

```
}
```

Type "it" for more

Q4)

```
hw5> db.zipcodes.aggregate([{$group: {_id: "$city", population: {$sum: "$pop"}}}, {$sort: {population: -1}}, {$limit: 5}, {$project: {_id: 0, city: "$_id", population: 1}} ]])
[
  { population: 2452177, city: 'CHICAGO' },
  { population: 2341387, city: 'BROOKLYN' },
  { population: 2123053, city: 'HOUSTON' },
  { population: 2102295, city: 'LOS ANGELES' },
  { population: 1639862, city: 'PHILADELPHIA' }
]
hw5> _
```

Q5)

```
hw5> db.zipcodes.aggregate([{$group: {_id: "$city", population: {$sum: "$pop"}}}, {$sort: {population: -1}}, {$limit: 5}, {$project: {_id: 0, city: "$_id", population: 1}} ]])
[
  { population: 2452177, city: 'CHICAGO' },
  { population: 2341387, city: 'BROOKLYN' },
  { population: 2123053, city: 'HOUSTON' },
  { population: 2102295, city: 'LOS ANGELES' },
  { population: 1639862, city: 'PHILADELPHIA' }
]
hw5> db.zipcodes.aggregate([
... {$group: {_id: "$state", count: {$sum: 1}}},
... {$match: {count: {$gt: 300, $lt: 500}}},
... {$sort: {count: 1}},
... {$project: {_id: 0, state: "$_id", count: 1}}
... ])
ReferenceError: aggregate is not defined
hw5> db.zipcodes.aggregate([{$group: {_id: "$state", count: {$sum: 1}}}, {$match: {count: {$gt: 300, $lt: 500}}}, {$sort: {count: 1}}, {$project: {_id: 0, state: "$_id", count: 1}} ]])
[
  { count: 314, state: 'MT' },
  { count: 350, state: 'SC' },
  { count: 363, state: 'MS' },
  { count: 384, state: 'SD' },
  { count: 384, state: 'OR' },
  { count: 391, state: 'ND' },
  { count: 410, state: 'NE' },
  { count: 414, state: 'CO' },
  { count: 420, state: 'RI' },
  { count: 464, state: 'LA' },
  { count: 474, state: 'VA' },
  { count: 484, state: 'MA' }
]
```

Q6)

```
hw5> db.createCollection("customers", {
...   validator: {
...     $jsonSchema: {
...       bsonType: "object",
...       required: ["Name", "Zipcode", "AvgRating"],
...       properties: {
...         Name: {
...           bsonType: "string"
...         },
...         Zipcode: {
...           bsonType: "string"
...         },
...         AvgRating: {
...           bsonType: "double",
...           minimum: 0.0,
...           maximum: 10.0
...         },
...         LastOrder: {
...           bsonType: "object",
...           properties: {
...             year: {
...               bsonType: "int"
...             },
...             tags: {
...               bsonType: "array",
...               items: {
...                 bsonType: "string"
...               }
...             }
...           }
...         }
...       }
...     }
...   })
... })
{ ok: 1 }
```

Q7)

```
hw5> db.customers.insertMany([
...   {
...     Name: "Gülbarin Maçın",
...     Zipcode: "99503",
...     AvgRating: 8.3
...   },
...   {
...     Name: "Andrei T.",
...     Zipcode: "90025",
...     AvgRating: 3.5,
...     LastOrder: { year: 2009 }
...   },
...   {
...     Name: "Bela T.",
...     Zipcode: "33126",
...     AvgRating: 4.9,
...     LastOrder: { year: 2019, tags: ["art", "melancholy"] }
...   },
...   {
...     Name: "Nuri Bilge C.",
...     Zipcode: "90010",
...     AvgRating: 6.5,
...     LastOrder: { year: 3005 }
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("647cc9e9b61fe88076e38b6c"),
    '1': ObjectId("647cc9e9b61fe88076e38b6d"),
    '2': ObjectId("647cc9e9b61fe88076e38b6e"),
    '3': ObjectId("647cc9e9b61fe88076e38b6f")
  }
}
```

```
hw5> db.customers.find()
[
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6c"),
    Name: 'Gülbarin Maçın',
    Zipcode: '99503',
    AvgRating: 8.3
  },
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6d"),
    Name: 'Andrei T.',
    Zipcode: '90025',
    AvgRating: 3.5,
    LastOrder: { year: 2009 }
  },
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6e"),
    Name: 'Bela T.',
    Zipcode: '33126',
    AvgRating: 4.9,
    LastOrder: { year: 2019, tags: [ 'art', 'melancholy' ] }
  },
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6f"),
    Name: 'Nuri Bilge C.',
    Zipcode: '90010',
    AvgRating: 6.5,
    LastOrder: { year: 3005 }
  }
]
hw5> _
```



```

hw5> db.customers.find({}, { _id: 0 })
[
  { Name: 'Gülbarın Maçın', Zipcode: '99503', AvgRating: 8.3 },
  {
    Name: 'Andrei T.',
    Zipcode: '90025',
    AvgRating: 3.5,
    LastOrder: { year: 2009 }
  },
  {
    Name: 'Bela T.',
    Zipcode: '33126',
    AvgRating: 4.9,
    LastOrder: { year: 2019, tags: [ 'art', 'melancholy' ] }
  },
  {
    Name: 'Nuri Bilge C.',
    Zipcode: '90010',
    AvgRating: 6.5,
    LastOrder: { year: 3005 }
  }
]

```

Q8)

```

hw5> db.customers.deleteMany({
...   "LastOrder.year": { $gt: 2023 }
... })
{ acknowledged: true, deletedCount: 1 }
hw5>

```

```

hw5> db.customers.find()
[
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6c"),
    Name: 'Gülbarın Maçın',
    Zipcode: '99503',
    AvgRating: 8.3
  },
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6d"),
    Name: 'Andrei T.',
    Zipcode: '90025',
    AvgRating: 3.5,
    LastOrder: { year: 2009 }
  },
  {
    _id: ObjectId("647cc9e9b61fe88076e38b6e"),
    Name: 'Bela T.',
    Zipcode: '33126',
    AvgRating: 4.9,
    LastOrder: { year: 2019, tags: [ 'art', 'melancholy' ] }
  }
]
hw5> db.customers.find({}, { _id: 0 })
[
  { Name: 'Gülbarın Maçın', Zipcode: '99503', AvgRating: 8.3 },
  {
    Name: 'Andrei T.',
    Zipcode: '90025',
    AvgRating: 3.5,
    LastOrder: { year: 2009 }
  },
  {
    Name: 'Bela T.',
    Zipcode: '33126',
    AvgRating: 4.9,
    LastOrder: { year: 2019, tags: [ 'art', 'melancholy' ] }
  }
]

```

Q9)

```

hw5> db.customers.updateOne(
...   { Name: "Gülbarin Maçın" },
...   { $set: { AvgRating: 15 } }
... )
Uncaught:
MongoServerError: Document failed validation
Additional information: {
  failingDocumentId: ObjectId("647cc9e9b61fe88076e38b6c"),
  details: {
    operatorName: '$jsonSchema',
    schemaRulesNotSatisfied: [
      {
        operatorName: 'properties',
        propertiesNotSatisfied: [
          {
            propertyName: 'AvgRating',
            details: [ [Object], [Object] ]
          }
        ]
      }
    ]
  }
}

```

It gives error because average rating is between 0 and 10.0. We do not have to update rating to 15 which is bigger than 10.

Q10)

```

hw5> db.customers.aggregate([
...   {
...     $lookup: {
...       from: "zipcodes",
...       localField: "Zipcode",
...       foreignField: "_id",
...       as: "zipcodeInfo"
...     }
...   },
...   {
...     $unwind: "$zipcodeInfo"
...   },
...   {
...     $project: {
...       _id: 0,
...       Name: 1,
...       City: "$zipcodeInfo.city",
...       State: "$zipcodeInfo.state",
...       Zipcode: "$zipcodeInfo._id"
...     }
...   },
...   {
...     $sort: {
...       Name: 1
...     }
...   }
... ])
[
  {
    Name: 'Andrei T.',
    City: 'LOS ANGELES',
    State: 'CA',
    Zipcode: '90025'
  },
  { Name: 'Bela T.', City: 'MIAMI', State: 'FL', Zipcode: '33126' },
  {
    Name: 'Gülbarin Maçın',
    City: 'ANCHORAGE',
    State: 'AK',
    Zipcode: '99503'
  }
]

```

Q11)

```
C:\Program Files\MongoDB\Tools\100\bin>mongoexport --db hw5 --collection=customers --out "F:\customers.json" mongod://localhost:27017/?authSource=admin --username username --password 1234
2023-06-04T21:17:58.810+0300    connected to: mongod://localhost:27017/?authSource=admin
2023-06-04T21:17:58.851+0300    exported 3 records
C:\Program Files\MongoDB\Tools\100\bin>
```