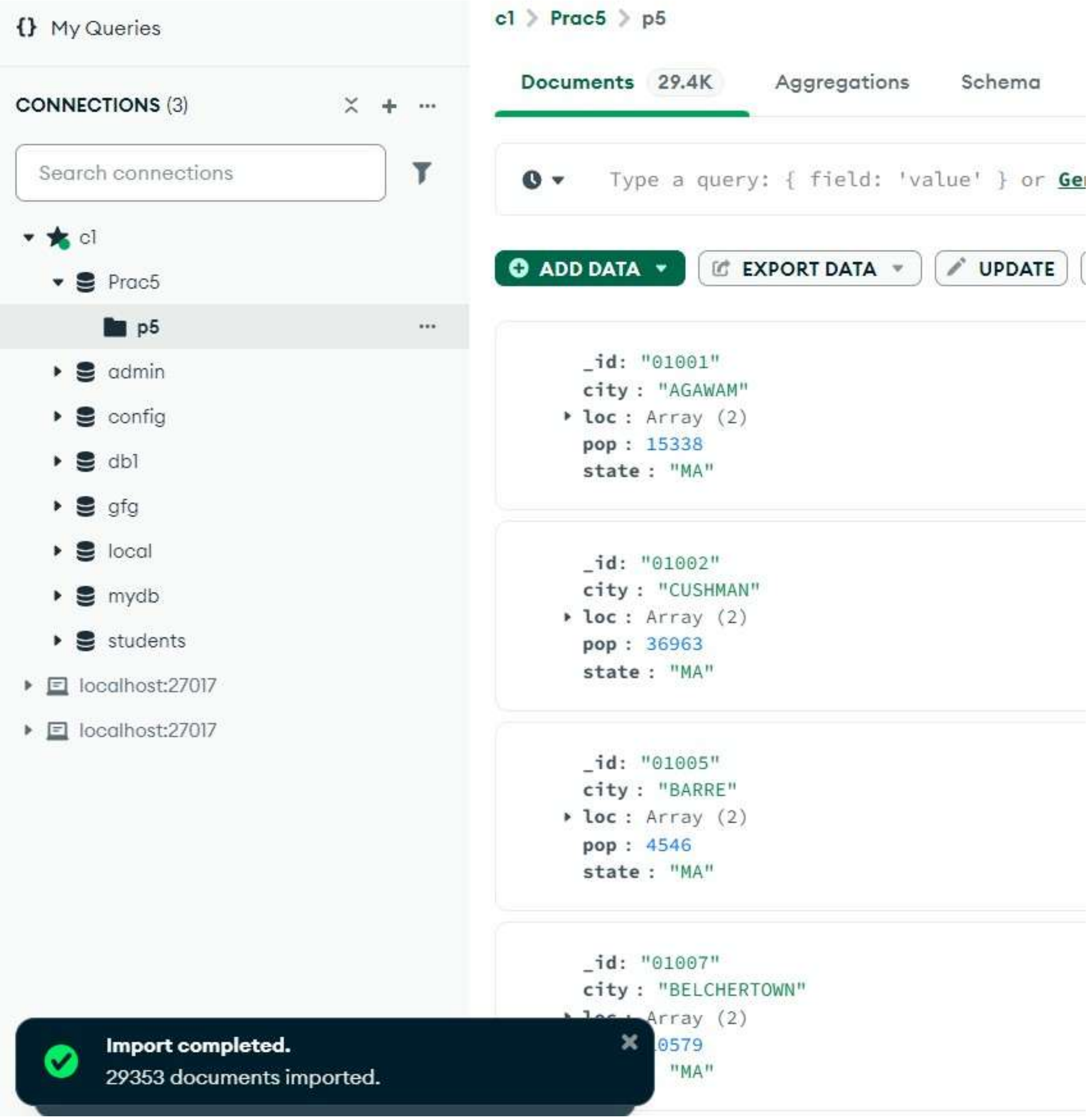
**Pratham Mehta 40778240026 MSCDSAI Div:L**

# Practical - 5

1. Download & import the file in MongoDb Compass



```
_id: "01001"
city : "AGAWAM"
loc : Array (2)
pop : 15338
state : "MA"

_id: "01002"
city : "CUSHMAN"
loc : Array (2)
pop : 36963
state : "MA"

_id: "01005"
city : "BARRE"
loc : Array (2)
pop : 4546
state : "MA"

_id: "01007"
city : "BELCHERTOWN"
loc : Array (2)
0579
"MA"
```

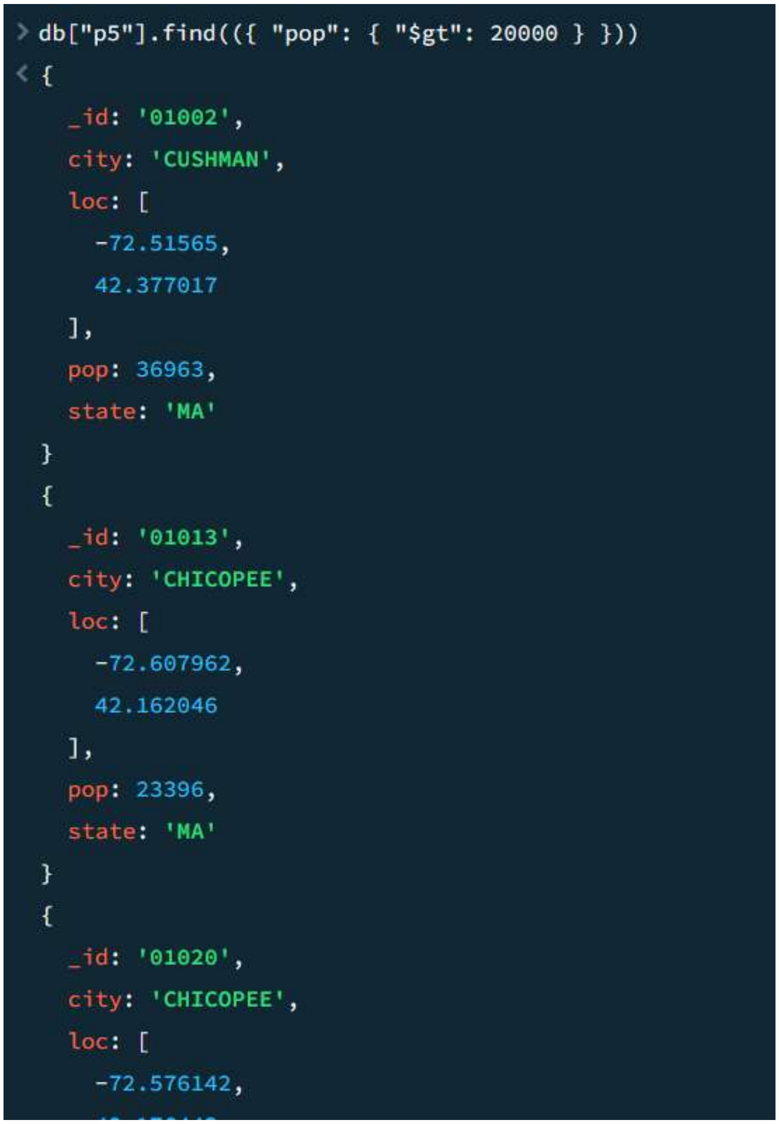
Import completed.
29353 documents imported.

# *Intermediate Queries*

## 1. Comparison Operators

a. Find cities with a population **greater** than 20,000

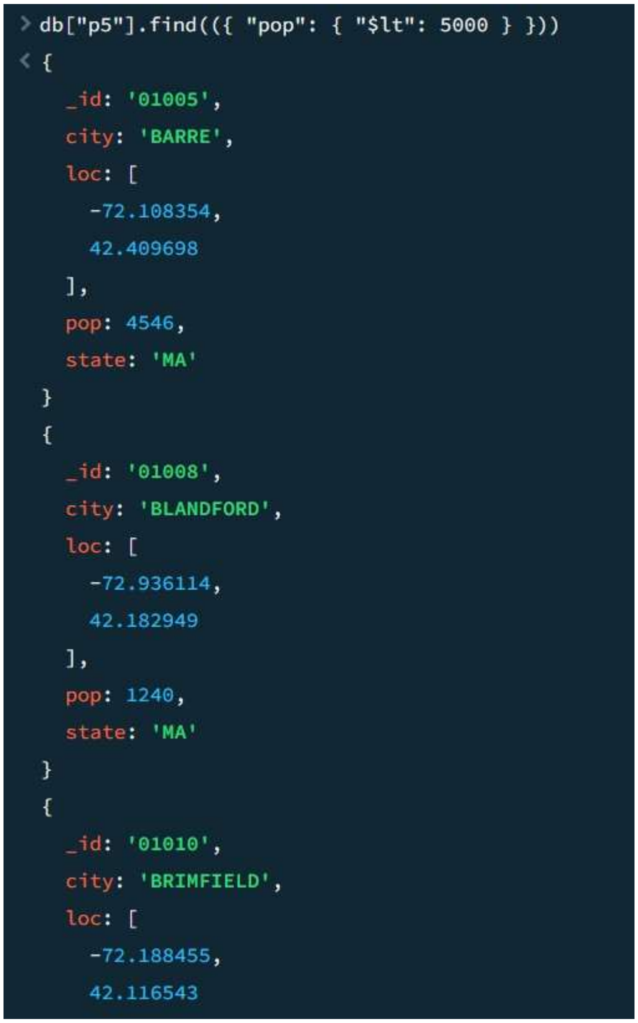
db.p5.find({ "pop": { "$gt": 20000 } })



```
> db["p5"].find(({ "pop": { "$gt": 20000 } }))
< {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01013',
    city: 'CHICOPEE',
    loc: [
      -72.607962,
      42.162046
    ],
    pop: 23396,
    state: 'MA'
  }
  {
    _id: '01020',
    city: 'CHICOPEE',
    loc: [
      -72.576142,
```

b. Find cities with a population **less than or equal to** 5,000

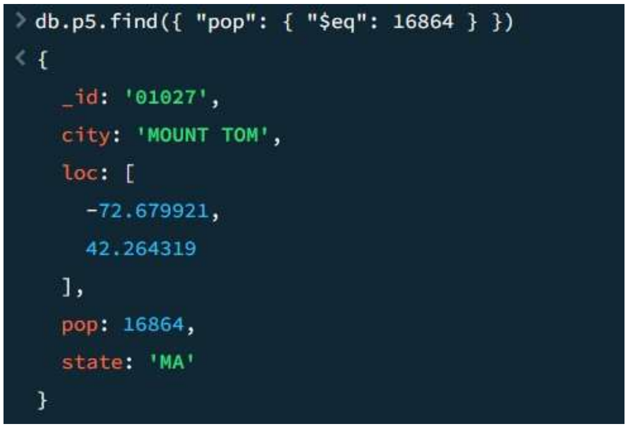
db.p5.find({ "pop": { "$lte": 5000 } })



```
> db["p5"].find(({ "pop": { "$lt": 5000 } }))
< {
    _id: '01005',
    city: 'BARRE',
    loc: [
      -72.108354,
      42.409698
    ],
    pop: 4546,
    state: 'MA'
  }
  {
    _id: '01008',
    city: 'BLANDFORD',
    loc: [
      -72.936114,
      42.182949
    ],
    pop: 1240,
    state: 'MA'
  }
  {
    _id: '01010',
    city: 'BRIMFIELD',
    loc: [
      -72.188455,
      42.116543
```

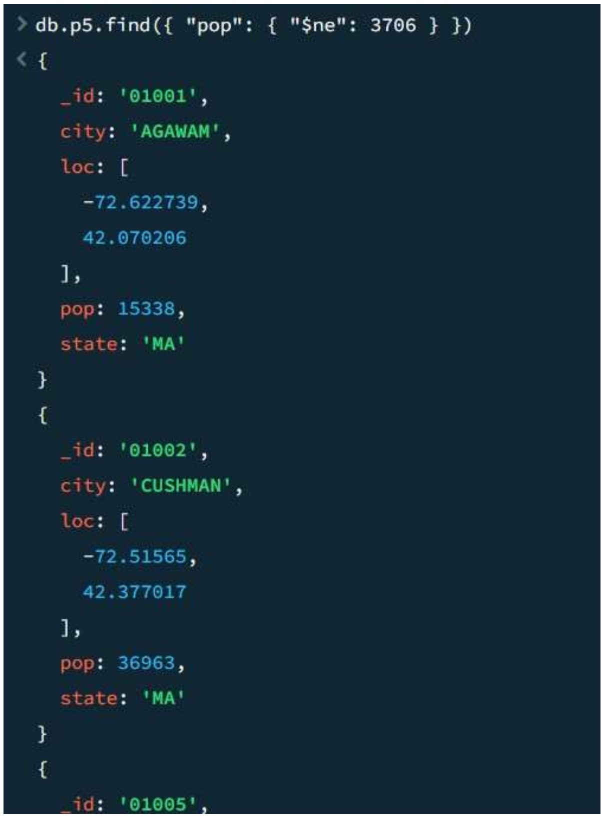
c. Find cities where population is **exactly** 16,864

```
db.p5.find({ "pop": { "$eq": 16864 } })
```



```
> db.p5.find({ "pop": { "$eq": 16864 } })
< {
    _id: '01027',
    city: 'MOUNT TOM',
    loc: [
      -72.679921,
      42.264319
    ],
    pop: 16864,
    state: 'MA'
  }
```

d. Find cities where population is **not equal to** 3,706 db.p5.find({ "pop": { "$ne": 3706 } })



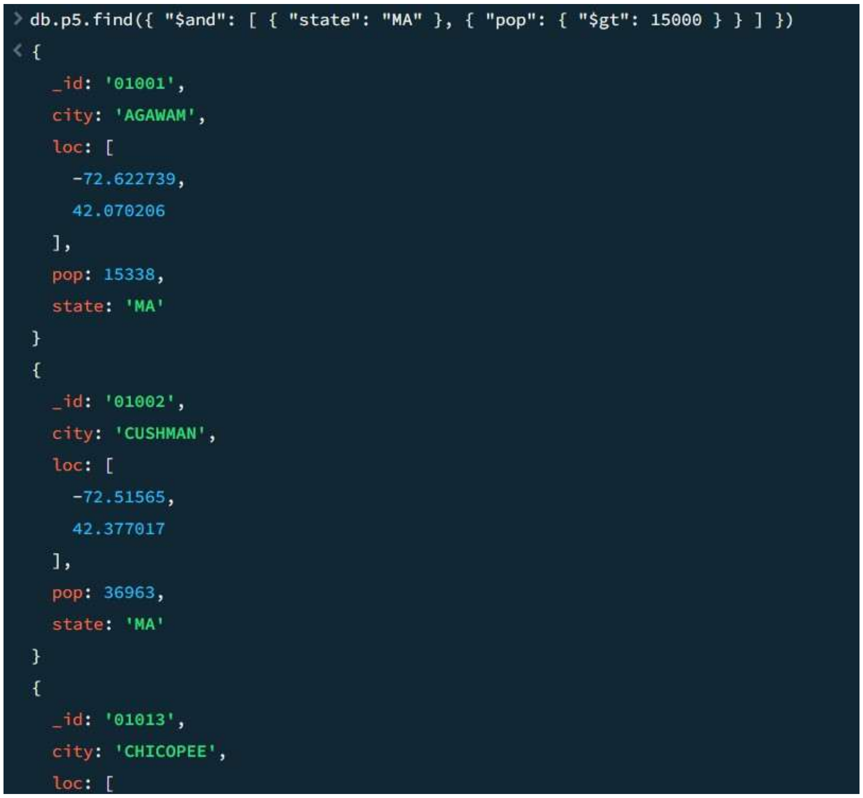
```
> db.p5.find({ "pop": { "$ne": 3706 } })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01005',
```

## 2. Logical Operators

a. AND operator

Find cities in Massachusetts (MA) with a population greater than 15,000

db.p5.find({ "$and": [ { "state": "MA" }, { "pop": { "$gt": 15000 } } ] })

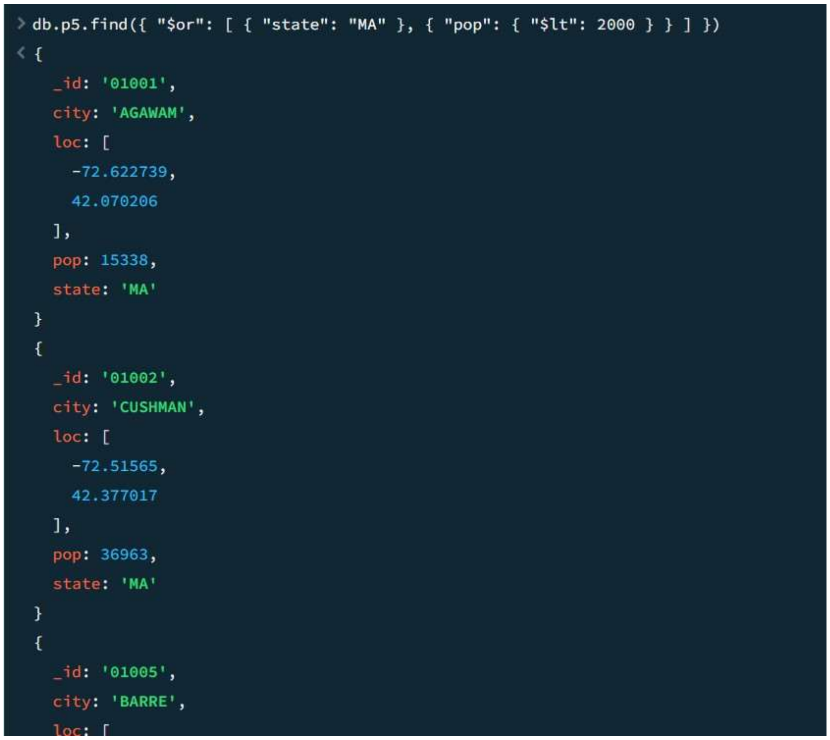


```
> db.p5.find({ "$and": [ { "state": "MA" }, { "pop": { "$gt": 15000 } } ] })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01013',
    city: 'CHICOPEE',
    loc: [
```

b. OR Operator

Find cities either in Massachusetts (MA) or with a population less than 2,000

```
db.p5.find({ "$or": [ { "state": "MA" }, { "pop": { "$lt": 2000 } } ] })
```

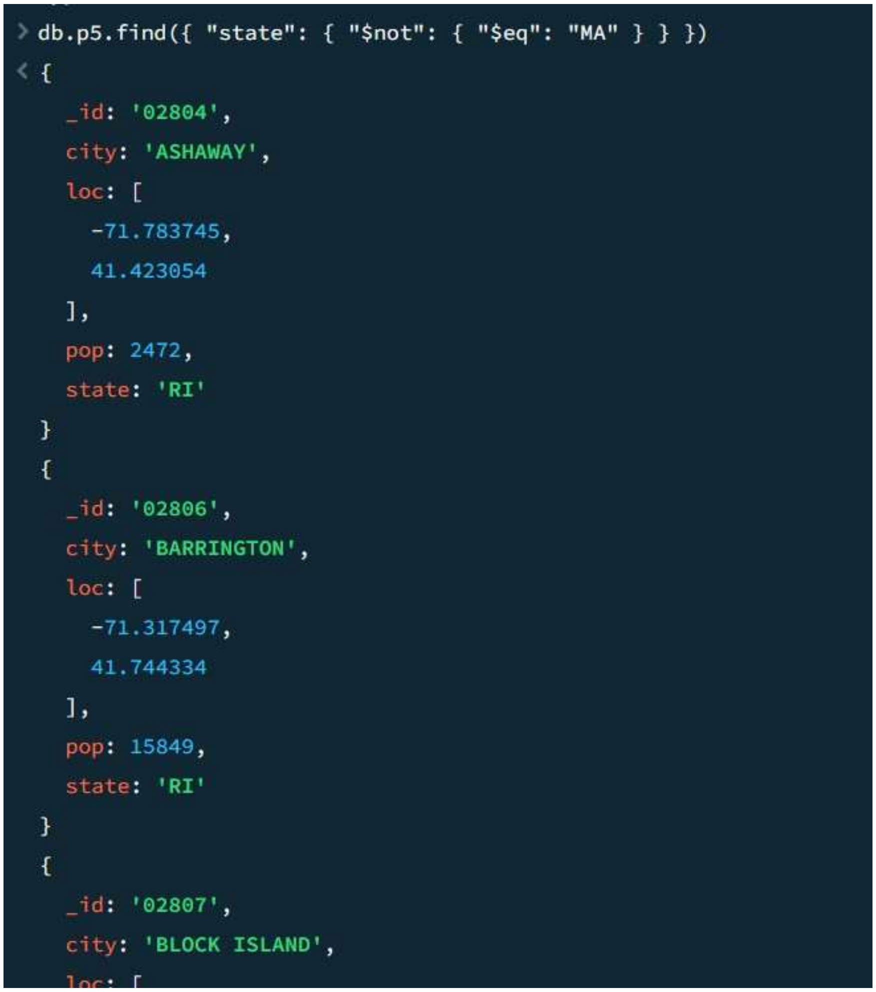


```
> db.p5.find({ "$or": [ { "state": "MA" }, { "pop": { "$lt": 2000 } } ] })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01005',
    city: 'BARRE',
    loc: [
```

c. NOT Operator

Find cities that are NOT in Massachusetts (MA)

```
db.p5.find({ "state": { "$not": { "$eq": "MA" } } })
```

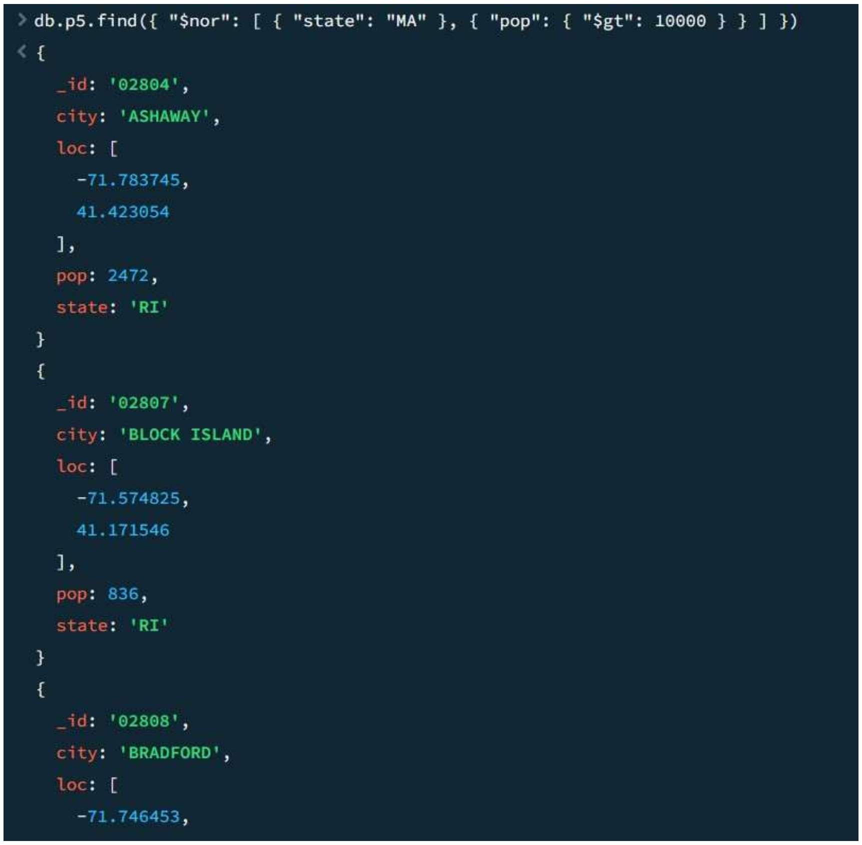


```
> db.p5.find({ "state": { "$not": { "$eq": "MA" } } })
< {
    _id: '02804',
    city: 'ASHAWAY',
    loc: [
      -71.783745,
      41.423054
    ],
    pop: 2472,
    state: 'RI'
  }
  {
    _id: '02806',
    city: 'BARRINGTON',
    loc: [
      -71.317497,
      41.744334
    ],
    pop: 15849,
    state: 'RI'
  }
  {
    _id: '02807',
    city: 'BLOCK ISLAND',
    loc: [
```

d. NOR Operator

Find cities that are neither in Massachusetts (MA) nor have a population greater than 10,000

```
db.p5.find({ "$nor": [ { "state": "MA" }, { "pop": { "$gt": 10000 } } ] })
```



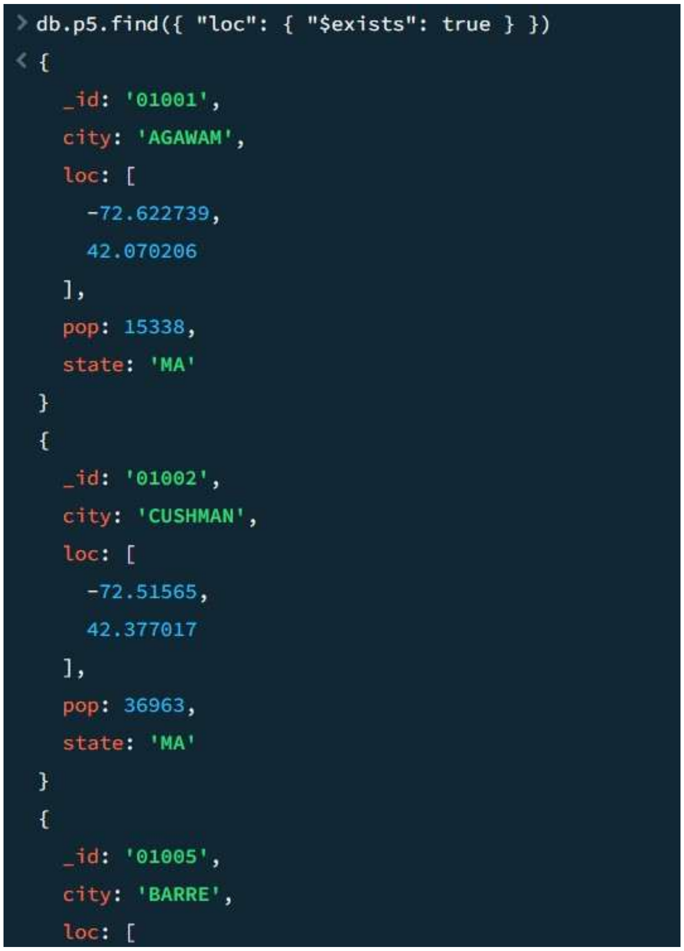
```
> db.p5.find({ "$nor": [ { "state": "MA" }, { "pop": { "$gt": 10000 } } ] })
< {
    _id: '02804',
    city: 'ASHAWAY',
    loc: [
      -71.783745,
      41.423054
    ],
    pop: 2472,
    state: 'RI'
  }
  {
    _id: '02807',
    city: 'BLOCK ISLAND',
    loc: [
      -71.574825,
      41.171546
    ],
    pop: 836,
    state: 'RI'
  }
  {
    _id: '02808',
    city: 'BRADFORD',
    loc: [
      -71.746453,
```

## 3. Element Operators

a. Exists

Find documents where the "loc" field exists

db.p5.find({ "loc": { "$exists": true } })

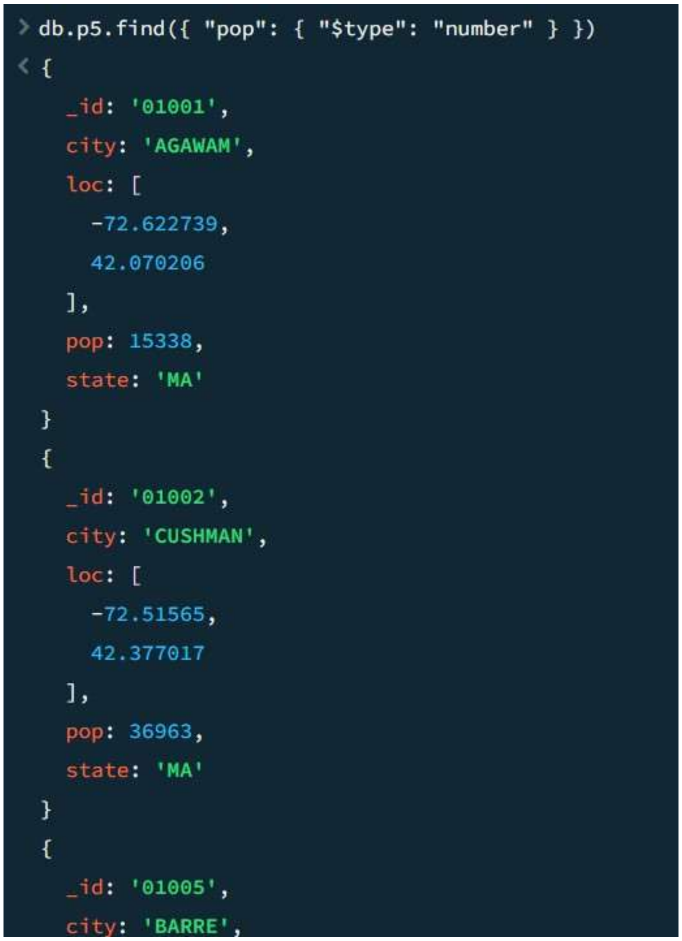


```
> db.p5.find({ "loc": { "$exists": true } })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01005',
    city: 'BARRE',
    loc: [
```

b. Type

Find documents where "pop" is of type number

db.p5.find({ "pop": { "$type": "number" } })

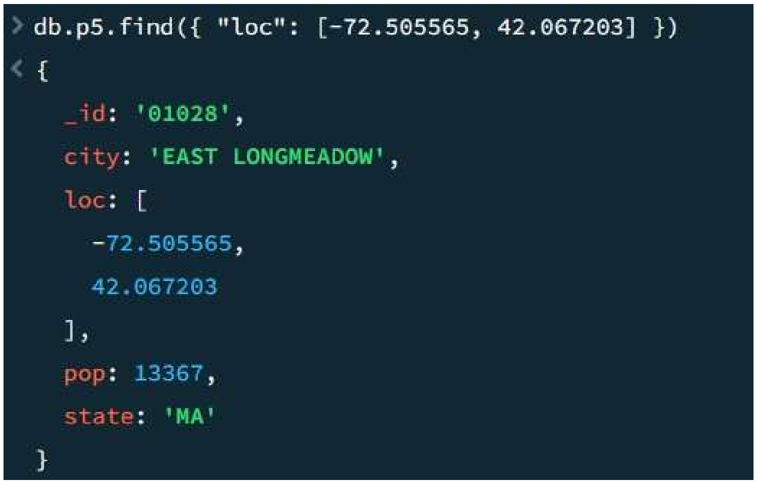


```
> db.p5.find({ "pop": { "$type": "number" } })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
  {
    _id: '01005',
    city: 'BARRE',
```

## 4. Array Operators

a. Find cities where "loc" contains exactly [-72.505565, 42.067203]

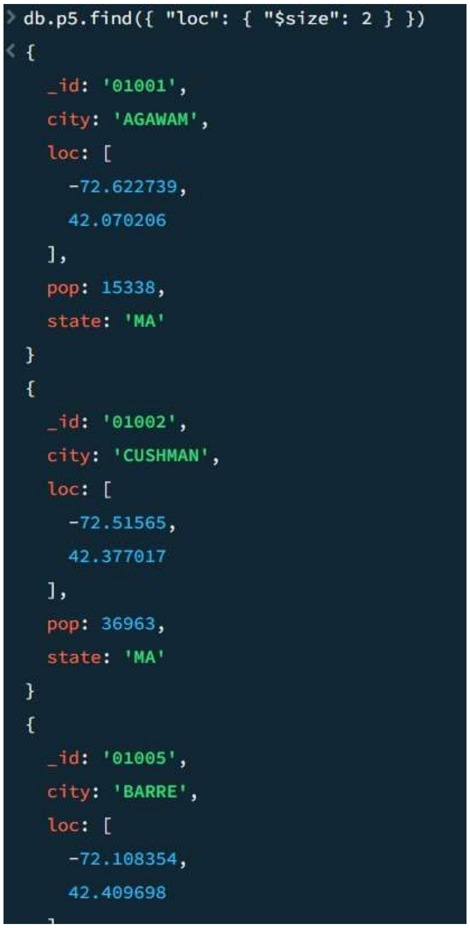
db.p5.find({ "loc": [-72.505565, 42.067203] })



```
> db.p5.find({ "loc": [-72.505565, 42.067203] })
< {
    _id: '01028',
    city: 'EAST LONGMEADOW',
    loc: [
      -72.505565,
      42.067203
    ],
    pop: 13367,
    state: 'MA'
  }
```

b. Find cities where "loc" array has exactly two elements

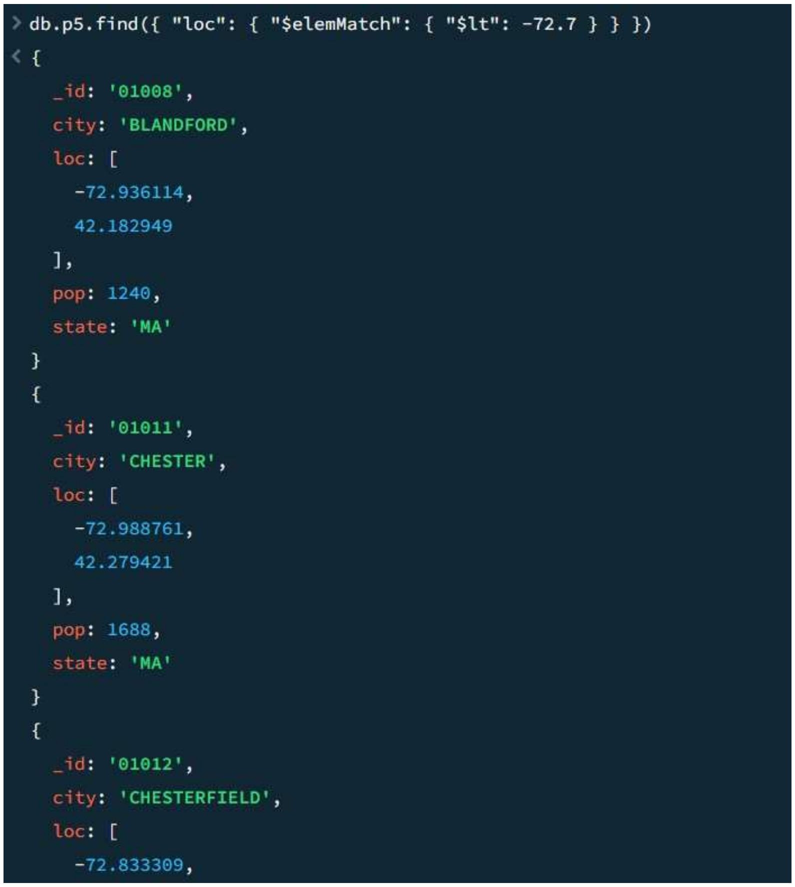
db.p5.find({ "loc": { "$size": 2 } })



```
> db.p5.find({ "loc": { "$size": 2 } })
< {
    _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
    ]
```

c. Find cities where longitude (first element of loc array) is less than -72.7

db.p5.find({ "loc": { "$elemMatch": { "$lt": -72.7 } } })



```
> db.p5.find({ "loc": { "$elemMatch": { "$lt": -72.7 } } })
< {
    _id: '01008',
    city: 'BLANDFORD',
    loc: [
      -72.936114,
      42.182949
    ],
    pop: 1240,
    state: 'MA'
  }
  {
    _id: '01011',
    city: 'CHESTER',
    loc: [
      -72.988761,
      42.279421
    ],
    pop: 1688,
    state: 'MA'
  }
  {
    _id: '01012',
    city: 'CHESTERFIELD',
    loc: [
      -72.833309,
```

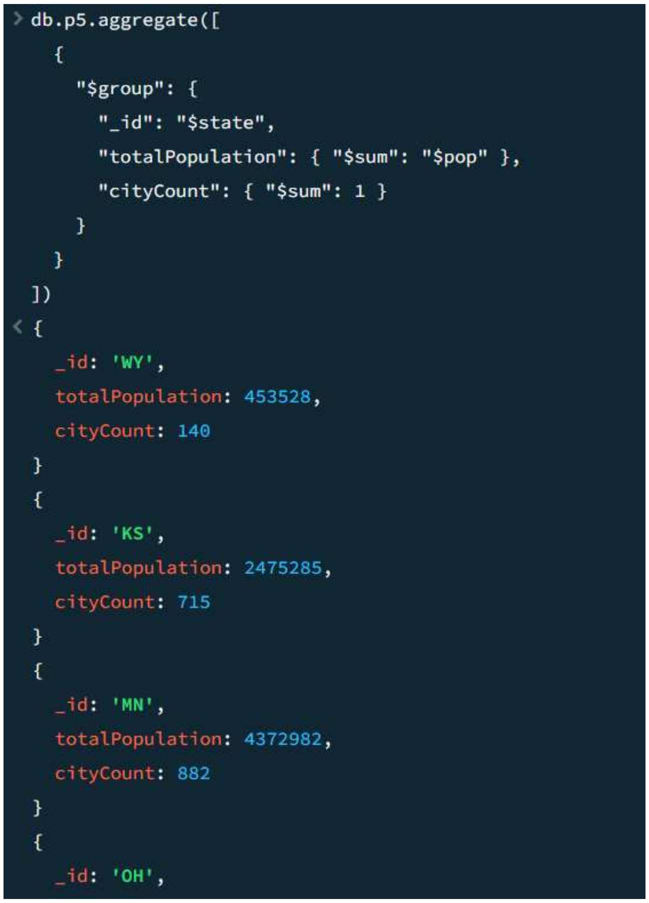
## ***Advanced Queries***

### 1. $group (Grouping Documents)

Group cities by state and calculate the total population per state

```
db.p5.aggregate([
 {
   "$group": {
     "_id": "$state",
     "totalPopulation": { "$sum": "$pop" },
     "cityCount": { "$sum": 1 }
```

}}])

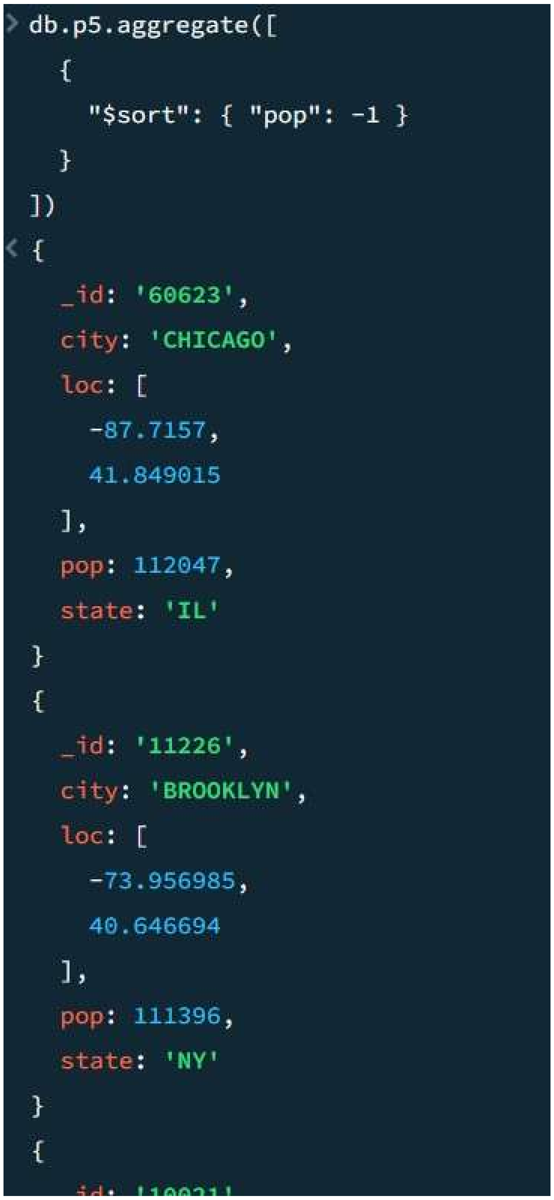


```
> db.p5.aggregate([
    {
      "$group": {
        "_id": "$state",
        "totalPopulation": { "$sum": "$pop" },
        "cityCount": { "$sum": 1 }
      }
    }
  ])
< {
    _id: 'WY',
    totalPopulation: 453528,
    cityCount: 140
  }
  {
    _id: 'KS',
    totalPopulation: 2475285,
    cityCount: 715
  }
  {
    _id: 'MN',
    totalPopulation: 4372982,
    cityCount: 882
  }
  {
    _id: 'OH',
```

## 2. $sort (Sorting Documents)

Sort cities by population in descending order

```
db.p5.aggregate([
 {
   "$sort": { "pop": -1 }
 }])
```

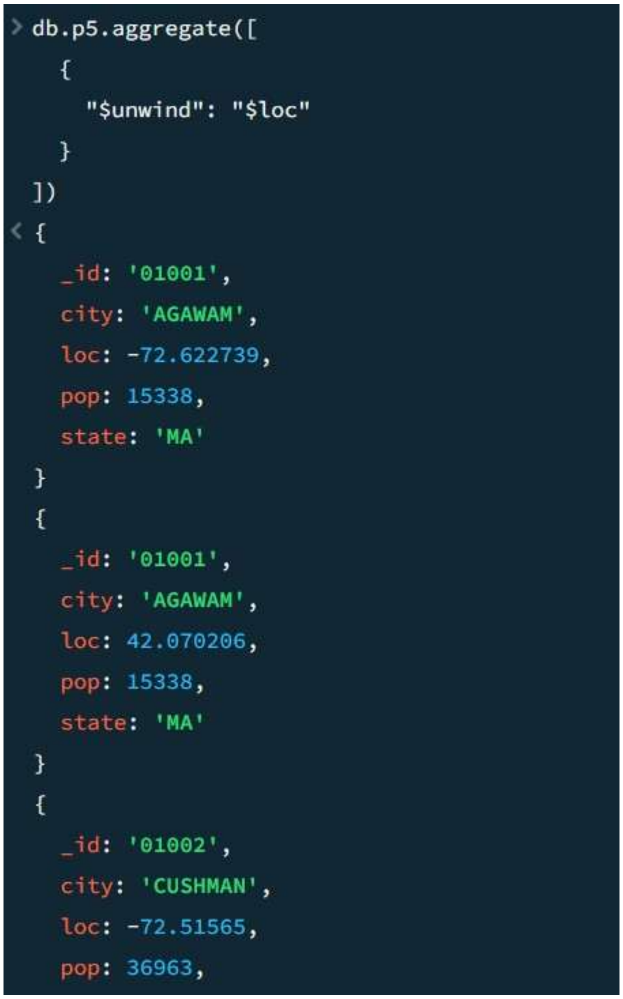


```
> db.p5.aggregate([
    {
      "$sort": { "pop": -1 }
    }
  ])
< {
    _id: '60623',
    city: 'CHICAGO',
    loc: [
      -87.7157,
      41.849015
    ],
    pop: 112047,
    state: 'IL'
  }
  {
    _id: '11226',
    city: 'BROOKLYN',
    loc: [
      -73.956985,
      40.646694
    ],
    pop: 111396,
    state: 'NY'
  }
  {
    _id: '10021'
```

### 3. $unwind (Deconstructing Arrays)

Unwind the "loc" array and display each coordinate separately

```
db.p5.aggregate([
 {
   "$unwind": "$loc"
 }])
```

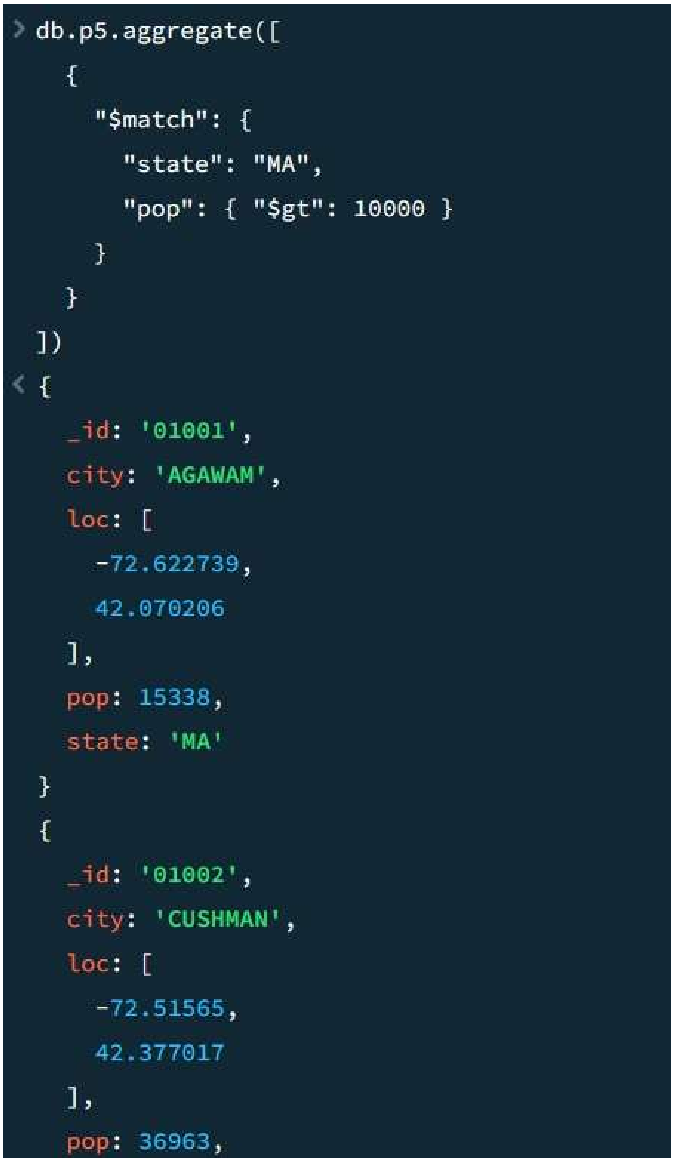


```
> db.p5.aggregate([
    {
      "$unwind": "$loc"
    }
  ])
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: -72.622739,
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01001',
    city: 'AGAWAM',
    loc: 42.070206,
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: -72.51565,
    pop: 36963,
```

## 4. $match (Filtering Documents)

Find cities in Massachusetts (MA) with a population greater than 10,000

```
db.p5.aggregate([
 {
   "$match": {
     "state": "MA",
     "pop": { "$gt": 10000 }
   }}])
```



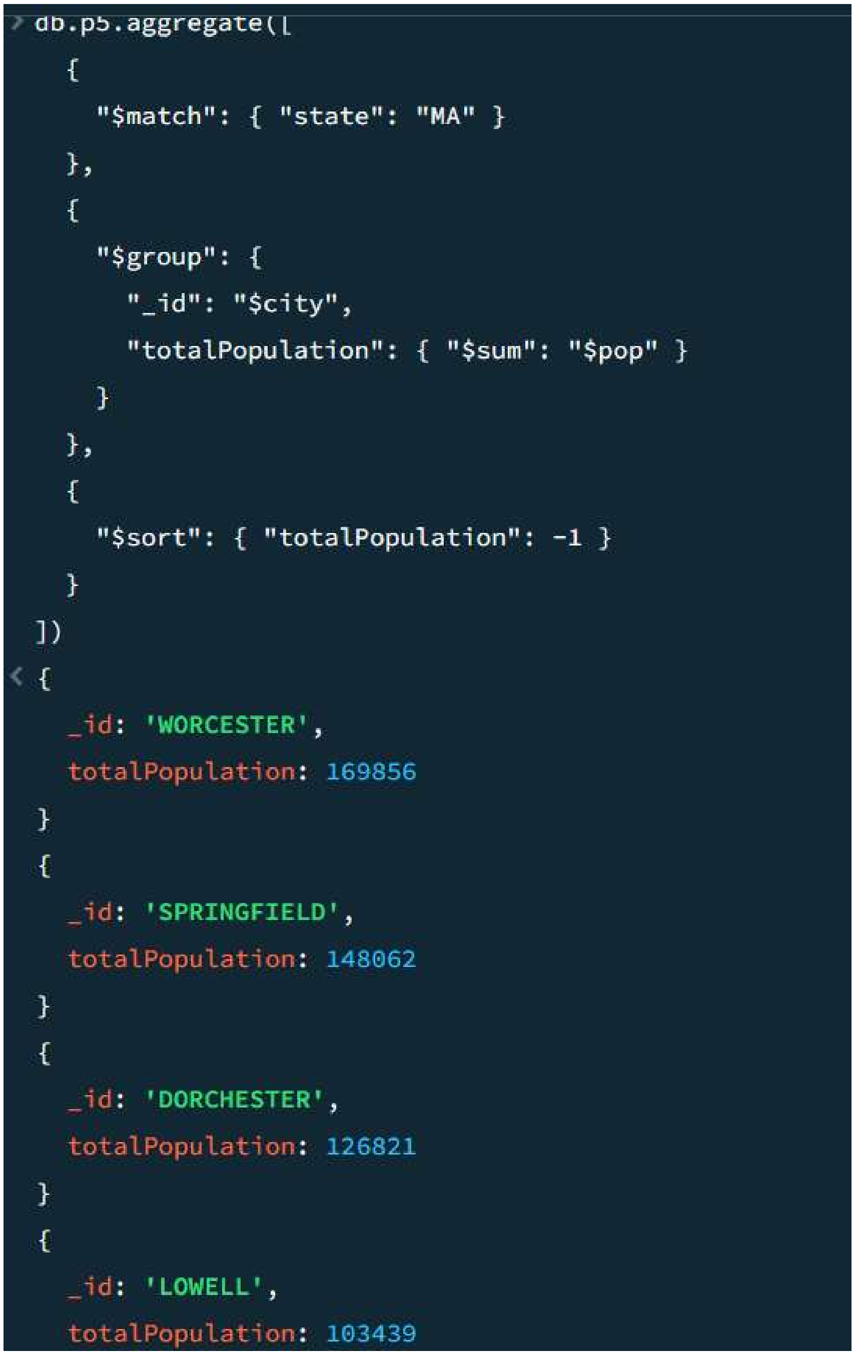
```
> db.p5.aggregate([
    {
      "$match": {
        "state": "MA",
        "pop": { "$gt": 10000 }
      }
    }
  ])
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
```

### *Combining Operators*

Find cities in MA, group by city, and sort by total population

```
db.p5.aggregate([
  {
    "$match": { "state": "MA" }
  },
  {
```

```
  "$group": {
    "_id": "$city",
    "totalPopulation": { "$sum": "$pop" }
  }},
{
  "$sort": { "totalPopulation": -1 }
}])
```



```
> db.p5.aggregate([
    {
      "$match": { "state": "MA" }
    },
    {
      "$group": {
        "_id": "$city",
        "totalPopulation": { "$sum": "$pop" }
      }
    },
    {
      "$sort": { "totalPopulation": -1 }
    }
  ])
< {
    _id: 'WORCESTER',
    totalPopulation: 169856
  }
  {
    _id: 'SPRINGFIELD',
    totalPopulation: 148062
  }
  {
    _id: 'DORCHESTER',
    totalPopulation: 126821
  }
  {
    _id: 'LOWELL',
    totalPopulation: 103439
```