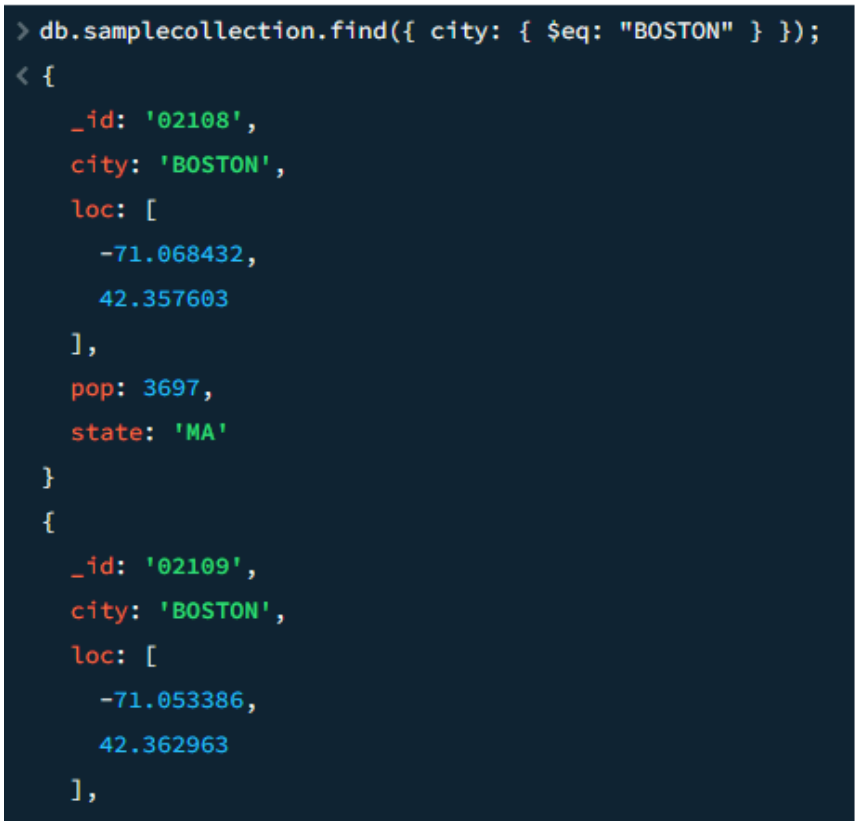
Practical 5
L030

Comparison Operators

Q1.Find all documents where the city is "BOSTON"
>db.samplecollection.find({ city: { $eq: "BOSTON" } });



Q2.Find all documents where the state is not "MA"
>db.samplecollection.find({ state: { $ne: "MA" } });

```
> db.samplecollection.find({ state: { $ne: "MA" } });
< {
    _id: '02804',
    city: 'ASHAWAY',
    loc: [
      -71.783745,
      41.423054
    ],
    pop: 2472,
    state: 'RI'
  }
```

Q3.Find all documents where the city is either "BOSTON" or "WORCESTER"
>db.samplecollection.find({ city: { $in: ["BOSTON", "WORCESTER"] } });

```
> db.samplecollection.find({ city: { $in: ["BOSTON", "WORCESTER"] } });
< {
    _id: '01602',
    city: 'WORCESTER',
    loc: [
      -71.841678,
      42.270251
    ],
    pop: 19988,
    state: 'MA'
  }
  {
    _id: '01603',
    city: 'WORCESTER',
    loc: [
      -71.837995,
      42.245033
```

Q4.Find all documents where the state is neither "MA" nor "NH"
>db.samplecollection.find({ state: { $nin: ["MA", "NH"] } });

```
> db.samplecollection.find({ state: { $nin: ["MA", "NH"] } });
< {
    _id: '02804',
    city: 'ASHAWAY',
    loc: [
      -71.783745,
      41.423054
    ],
    pop: 2472,
    state: 'RI'
  }
```

Q5.Find all documents where the population is less than or equal to 500:
>db.samplecollection.find({ pop: { $lte: 500 } });

```
> db.samplecollection.find({ pop: { $lte: 500 } });
< {
    _id: '01012',
    city: 'CHESTERFIELD',
    loc: [
      -72.833309,
      42.38167
    ],
    pop: 177,
    state: 'MA'
  }
  {
```

Logical Operators

Q1.Find all cities in **Massachusetts (MA)** where the population is **greater than 10,000** AND **less than 30,000**:

```
db.samplecollection.find({
  "$and": [
    { "state": "MA" },
    { "pop": { "$gt": 10000 } },
    { "pop": { "$lt": 30000 } }
  ]
```

})

```
db.samplecollection.find({
  "$and": [
    { "state": "MA" },
    { "pop": { "$gt": 10000 } },
    { "pop": { "$lt": 30000 } }
  ]
})
{
  _id: '01001',
  city: 'AGAWAM',
  loc: [
    -72.622739,
    42.070206
  ],
  pop: 15338,
  state: 'MA'
}
```

Q2.Find all cities in **Massachusetts (MA)** where the population is **either less than 1,000 OR greater than 50,000**:

```
db.samplecollection.find({
  "$or": [
    { "pop": { "$lt": 1000 } },
    { "pop": { "$gt": 50000 } }
  ]
})
```

Q3.Find all cities in **Massachusetts (MA)** where the population is **NOT greater than 30,000**:

```
db.samplecollection.find({
  "state": "MA",
  "pop": { "$not": { "$gt": 30000 } }
})
```

Q4F.ind all cities in **Massachusetts (MA)** where the population is **neither less than 1,000 NOR greater than 50,000** (opposite of `$or` above):

```
db.samplecollection.find({
  "$nor": [
    { "pop": { "$lt": 1000 } },
    { "pop": { "$gt": 50000 } }
  ]
})
```

```
> db.samplecollection.find({
    "$nor": [
      { "pop": { "$lt": 1000 } },
      { "pop": { "$gt": 50000 } }
    ]
  })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
  }
  {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
```

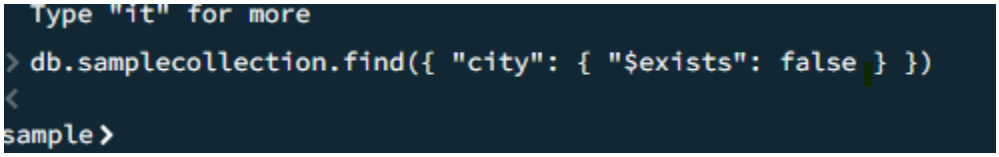
Element Operators

Q1.A.Find all documents where the `pop` (population) field **exists**:
db.samplecollection.find({ "pop": { "$exists": true } })

```
> db.samplecollection.find({ "pop": { "$exists": true } })
< {
    _id: '01001',
    city: 'AGAWAM',
    loc: [
      -72.622739,
      42.070206
    ],
    pop: 15338,
    state: 'MA'
```

Q1.B.Find all documents where the `city` field **does not exist**:

>db.samplecollection.find({ "city": { "$exists": false } })



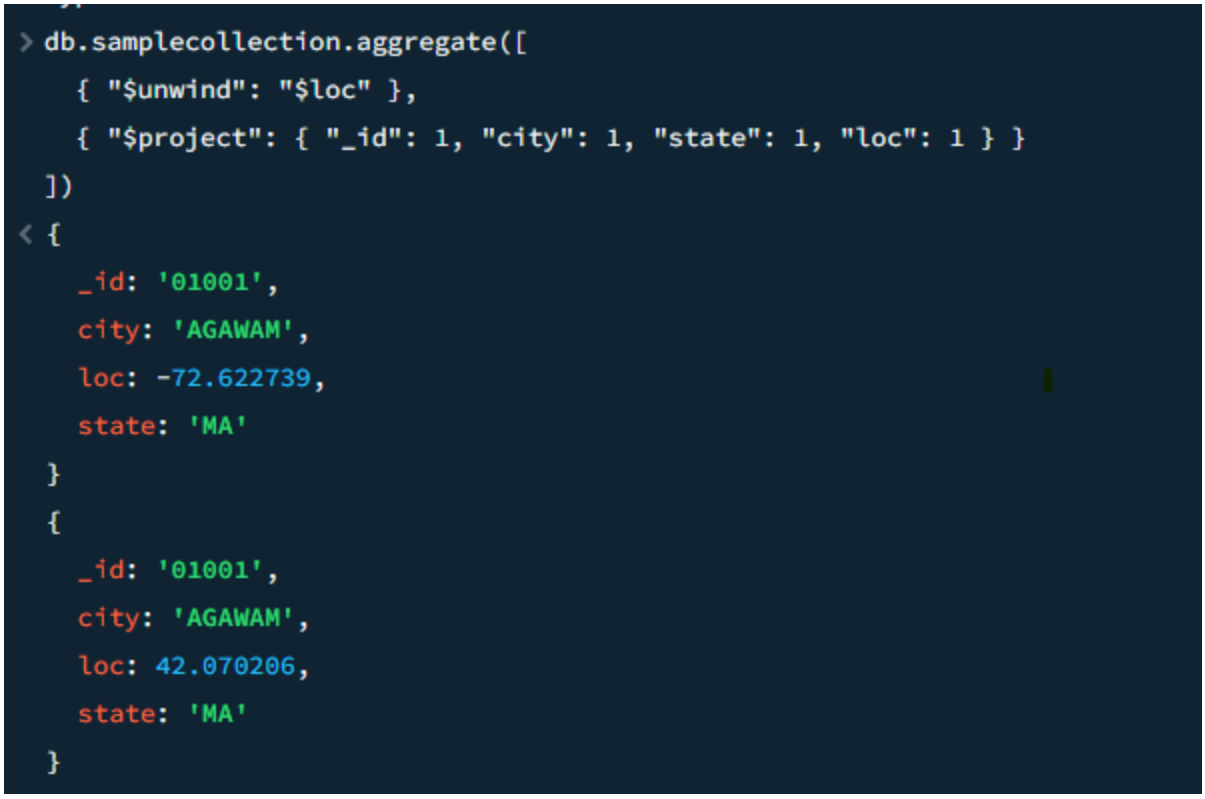
Q2.A.Find all documents where `pop` is stored as a **number**:
>db.samplecollection.find({ "pop": { "$type": "number" } })

Q2.B.Find all documents where `city` is stored as a **string**:
db.samplecollection.find({ "city": { "$type": "string" } })

Advanced queries

Q1. **$unwind (Deconstruct an array field)**
Since our dataset does not contain an array field, assume we modify loc (latitude and longitude) for unwinding:

```
>db.samplecollection.aggregate([
  { "$unwind": "$loc" },
  { "$project": { "_id": 1, "city": 1, "state": 1, "loc": 1 } }
])
```



Q2.**$group (Group by a field and perform aggregation)**
Find the total population of each city:

```
>db.samplecollection.aggregate([
  { "$group": {
      "_id": "$city",
      "total_population": { "$sum": "$pop" }
    }
  }
])
```

```
> db.samplecollection.aggregate([
    { "$group": {
        "_id": "$city",
        "total_population": { "$sum": "$pop" }
      }
    }
  ])
< {
    _id: 'OAKLEY',
    total_population: 46650
  }
  {
    _id: 'TURBOTVILLE',
    total_population: 3902
  }
```

Q3.**$match (Filter documents based on conditions)**
Find all cities in Massachusetts (MA) where the population is greater than 30,000:
db.samplecollection.aggregate([
  { "$match": { "state": "MA", "pop": { "$gt": 30000 } } }
])

```
> db.samplecollection.aggregate([
    { "$match": { "state": "MA", "pop": { "$gt": 30000 } } }
  ])
< {
    _id: '01002',
    city: 'CUSHMAN',
    loc: [
      -72.51565,
      42.377017
    ],
    pop: 36963,
    state: 'MA'
  }
```

Q4.**$sort (Sort documents in ascending/descending order)**
Sort cities in descending order of population:
db.samplecollection.aggregate([
  { "$sort": { "pop": -1 } }

])

```
> db.samplecollection.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'
```