

## Geospatial query:

Need to upload a new collection called "location" in json format Follow the same steps to switch this collection to database.

**Use db**

**Show dbs**

**Show collections**

```
Current Mongosh Log ID: 66640228cfc60363b8cdcdf5
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&se
erverSelectionTimeoutMS=2000&appName=mongosh+2.2.6
Using MongoDB:      7.0.11
Using Mongosh:      2.2.6

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

-----
The server generated these startup warnings when booting
2024-06-08T11:39:34.025+05:30: Access control is not enabled for the da
tabase. Read and write access to data and configuration is unrestricted
-----

test> use db
switched to db db
db> show dbs
admin    40.00 KiB
config  108.00 KiB
db       192.00 KiB
local    72.00 KiB
db> show collections
locations
students
students_permission
Please enter a MongoDB connection string (Default: mongodb://localhost/):
db> |
```

For find a location :

```
db> db.locations.find({
... location:{
... $geoWithin:{
... $centerSphere:[[-74.005,40.712],0.00621376]]}}});
[
  {
    _id: 1,
    name: 'Coffee Shop A',
    location: { type: 'Point', coordinates: [ -73.985, 40.748 ] }
  },
  {
    _id: 2,
    name: 'Restaurant B',
    location: { type: 'Point', coordinates: [ -74.009, 40.712 ] }
  },
  {
    _id: 5,
    name: 'Park E',
    location: { type: 'Point', coordinates: [ -74.006, 40.705 ] }
  }
]
db>
```

## Projection operators

**Projection** in mongodb is a powerful tool that can be used to extract only the fields you need from a document – not all fields.

- It enables you to project concise and transparent data.
- It filter the dataset without impacting the overall database performance.

### Adding new dataset called **PRODUCTS**

```
test> use db
switched to db db
db> show collections
candidates
locations
products
std
students_permission
```

here. The new dataset **product** is added

now we are retriving name and ratings of the product

```
db> db.products.find({}, {name:1,rating:1});
[
  {
    _id: 'ac3', name: 'AC3 Phone', rating: 3.8 },
  {
    _id: 'ac7', name: 'AC7 Phone', rating: 4 },
  {
    _id: ObjectId('507d95d5719dbef170f15bf9'),
    name: 'AC3 Series Charger',
    rating: 2.8
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfa'),
    name: 'AC3 Case Green',
    rating: 1
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfb'),
    name: 'Phone Extended Warranty',
    rating: 5
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfc'),
    name: 'AC3 Case Black',
    rating: 2
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfd'),
    name: 'AC3 Case Red',
    rating: 4
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfe'),
    name: 'Phone Service Basic Plan',
    rating: 3
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bff'),
    name: 'Phone Service Core Plan',
    rating: 3
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c00'),
    name: 'Phone Service Family Plan',
    rating: 4
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c01'),
    name: 'Cable TV Basic Service Package',
    rating: 3.9
  }
]
```

Product: refers to the collection named products within the current database.

To get excluding fields:

```
db> db.products.find({}, {_id:0,type:0,limits:0,data:0});
[
  {
    name: 'AC3 Phone',
    brand: 'ACME',
    price: 200,
    rating: 3.8,
    warranty_years: 1,
    available: true
  },
  {
    name: 'AC7 Phone',
    brand: 'ACME',
    price: 320,
    rating: 4,
    warranty_years: 1,
    available: false
  },
  {
    name: 'AC3 Series Charger',
    price: 19,
    rating: 2.8,
    warranty_years: 0.25,
    for: [ 'ac3', 'ac7', 'ac9' ]
  },
  {
    name: 'AC3 Case Green',
    color: 'green',
    price: 12,
    rating: 1,
    warranty_years: 0
  },
  {
    name: 'Phone Extended Warranty',
    price: 38,
    rating: 5,
    warranty_years: 2,
    for: [ 'ac3', 'ac7', 'ac9', 'qp7', 'qp8', 'qp9' ]
  },
  {
    name: 'AC3 Case Black',
    color: 'black',
    price: 12.5,
    rating: 2,
    warranty_years: 0.25,
    available: false,
    for: 'ac3'
  },
  {
    name: 'AC3 Case Red',
    color: 'red',
    price: 12,
    rating: 4,
    warranty_years: 0.25,
    available: true,
    for: 'ac3'
  }
],
[
  {
    name: 'AC3 Case Red',
    color: 'red',
    price: 12,
    rating: 4,
    warranty_years: 0.25,
    available: true,
    for: 'ac3'
  },
  {
    name: 'Phone Service Basic Plan',
    monthly_price: 40,
    rating: 3,
    term_years: 2
  },
  {
    name: 'Phone Service Core Plan',
    monthly_price: 60,
    rating: 3,
    term_years: 1
  },
  {
    name: 'Phone Service Family Plan',
    monthly_price: 90,
    rating: 4,
    sales_tax: true,
    term_years: 2
  },
  {
    name: 'Cable TV Basic Service Package',
    monthly_price: 50,
    rating: 3.9,
    term_years: 2,
    cancel_penalty: 25,
    sales_tax: true,
    additional_tarriffs: [
      { kind: 'federal tarriff', amount: { percent_of_service: 0.06 } },
      { kind: 'misc tarriff', amount: 2.25 }
    ]
  }
]
```

## \$elemMatch

This is a MongoDB operator that allows you to find documents within an array that contain an element that matches a specified condition

```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Physics"}}},{name:1,"courses,$":1});
[
  {
    _id: ObjectId('6657ff95946a866d8bb971e60'), name: 'Bob Johnson' },
  {
    _id: ObjectId('6657ff95946a866d8bb971e62'), name: 'Emily Jones' }
]
db> .
```

## \$slice operation:

This operator limits the number of elements returned from an array. It takes two arguments: o The first argument specifies the starting index of the slice (zero-based). The second argument (optional) specifies the number of elements to return. It returns all elements from the starting index to the end of the array.

```
db> db.candidates.find({}, {courses:{$slice:1}})
[
  {
    _id: ObjectId('6657ff95946a866d8bb971e5f'),
    name: 'Alice Smith',
    age: 20,
    courses: [ 'English' ],
    gpa: 2.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866d8bb971e60'),
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Computer Science' ],
    gpa: 4.0,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866d8bb971e61'),
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'History' ],
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866d8bb971e62'),
    name: 'Emily Jones',
    age: 21,
    courses: [ 'Mathematics' ],
    gpa: 3.6,
    home_city: 'Houston',
    blood_group: 'AB',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866d8bb971e63'),
    name: 'David Williams',
    age: 23,
    courses: [ 'English' ],
    gpa: 4,
    home_city: 'Phoenix',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866d8bb971e64'),
    name: 'Fatima Brown',
    age: 18,
    courses: [ 'Biology' ],
    gpa: 3.5,
    home_city: 'San Antonio',
    blood_group: 'B-'
  }
]
```

This part defines a projection document using curly braces {}.



```

1 db.candidates.find({}, {courses: {$slice: [1,3]}})
2 {
3   _id: ObjectId('6652ff64a9a9a9b971e5f'),
4   name: 'Alice Smith',
5   age: 28,
6   courses: [ 'Biology', 'Chemistry' ],
7   gpa: 3.4,
8   home_city: 'New York City',
9   blood_group: 'A+',
10  is_hotel_resident: true
11 }
12 {
13   _id: ObjectId('6652ff6594a9a9b971e60'),
14   name: 'Bob Johnson',
15   age: 22,
16   courses: [ 'Mathematics', 'Physics' ],
17   gpa: 3.8,
18   home_city: 'Los Angeles',
19   blood_group: 'O-',
20   is_hotel_resident: false
21 }
22 {
23   _id: ObjectId('6652ff6594a9a9b971e61'),
24   name: 'Charlie Lee',
25   age: 19,
26   courses: [ 'English', 'Psychology' ],
27   gpa: 3.3,
28   home_city: 'Chicago',
29   blood_group: 'B+',
30   is_hotel_resident: true
31 }
32 {
33   _id: ObjectId('6652ff6594a9a9b971e62'),
34   name: 'Emily Jones',
35   age: 24,
36   courses: [ 'Physics', 'Statistics' ],
37   gpa: 3.6,
38   home_city: 'Houston',
39   blood_group: 'AB-',
40   is_hotel_resident: false
41 }
42 {
43   _id: ObjectId('6652ff6594a9a9b971e63'),
44   name: 'David Williams',
45   age: 20,
46   courses: [ 'Literature', 'Philosophy' ],
47   gpa: 3.1,
48   home_city: 'Phoenix',
49   blood_group: 'A-',
50   is_hotel_resident: true
51 }
52 {
53   _id: ObjectId('6652ff64a9a9a9b971e64'),
54   name: 'Fatima Brown',
55   age: 18,
56   courses: [ 'Chemistry', 'Environmental Science' ],
57   gpa: 3.5,
58   home_city: 'San Antonio',
59   blood_group: 'B+',
60   is_hotel_resident: false
61 }
62 }

```

\$slice operation [1:3]

```

1 db.candidates.find([1], {courses: {$slice: 3}})
2 {
3   _id: ObjectId('6652ff64a9a9a9b971e5f'),
4   name: 'Alice Smith',
5   age: 28,
6   courses: [ 'English', 'Biology', 'Chemistry' ],
7   gpa: 3.4,
8   home_city: 'New York City',
9   blood_group: 'A+',
10  is_hotel_resident: true
11 }
12 {
13   _id: ObjectId('6652ff6594a9a9b971e60'),
14   name: 'Bob Johnson',
15   age: 22,
16   courses: [ 'Environmental Science', 'Mathematics', 'Physics' ],
17   gpa: 3.8,
18   home_city: 'Los Angeles',
19   blood_group: 'O-',
20   is_hotel_resident: false
21 }
22 {
23   _id: ObjectId('6652ff6594a9a9b971e61'),
24   name: 'Charlie Lee',
25   age: 19,
26   courses: [ 'History', 'English', 'Psychology' ],
27   gpa: 3.3,
28   home_city: 'Chicago',
29   blood_group: 'B+',
30   is_hotel_resident: true
31 }
32 {
33   _id: ObjectId('6652ff6594a9a9b971e62'),
34   name: 'Emily Jones',
35   age: 24,
36   courses: [ 'Mathematics', 'Physics', 'Statistics' ],
37   gpa: 3.6,
38   home_city: 'Houston',
39   blood_group: 'AB-',
40   is_hotel_resident: false
41 }
42 {
43   _id: ObjectId('6652ff6594a9a9b971e63'),
44   name: 'David Williams',
45   age: 20,
46   courses: [ 'English', 'Literature', 'Philosophy' ],
47   gpa: 3.1,
48   home_city: 'Phoenix',
49   blood_group: 'A-',
50   is_hotel_resident: true
51 }
52 {
53   _id: ObjectId('6652ff64a9a9a9b971e64'),
54   name: 'Fatima Brown',
55   age: 18,
56   courses: [ 'Biology', 'Chemistry', 'Environmental Science' ],
57   gpa: 3.5,
58   home_city: 'San Antonio',
59   blood_group: 'B+',
60   is_hotel_resident: false
61 }
62 }

```