



Course Overview



View Discussion

Chapter 2: Basic Aggregation - Utility Stages

Lab: Using Cursor-like Stages

< Back to the Question

 COPY

```
var favorites = [
  "Sandra Bullock",
  "Tom Hanks",
  "Julia Roberts",
  "Kevin Spacey",
  "George Clooney"]

db.movies.aggregate([
  {
    $match: {
      "tomatoes.viewer.rating": { $gte: 3 },
      countries: "USA",
      cast: {
        $in: favorites
      }
    }
  },
  {
    $project: {
      _id: 0,
      title: 1,
      "tomatoes.viewer.rating": 1,
      num_favs: {
        $size: {
          $setIntersection: [
            "$cast",
            favorites
          ]
        }
      }
    }
  },
  {
    $sort: { num_favs: -1, "tomatoes.viewer.rating": -1, title: -1 }
  },
  {
    $skip: 24
  },
  {
    $limit: 1
  }
])
```

We store our favorites in a variable for easy reference within the pipeline

[COPY](#)

```
var favorites = [  
  "Sandra Bullock",  
  "Tom Hanks",  
  "Julia Roberts",  
  "Kevin Spacey",  
  "George Clooney"]
```

We start by matching films that include at least one of our favorites in their **cast**

[COPY](#)

```
{  
  $match: {  
    "tomatoes.viewer.rating": { $gte: 3 },  
    countries: "USA",  
    cast: {  
      $in: favorites  
    }  
  }  
}
```

Then, we will be projecting the **num_favs** value by calculating the **\$size** of the array intersection, between the given set of favorites and the film **cast**:

[COPY](#)

```
{  
  $project: {  
    _id: 0,  
    title: 1,  
    "tomatoes.viewer.rating": 1,  
    starPower: {  
      $size: {  
        $setIntersection: favorites  
      }  
    }  
  }  
}
```

After that, we call the **\$sort** stage and **\$skip + \$limit** in the result to the element requested:

[COPY](#)

```
{  
  $sort: { num_favs: -1, "tomatoes.viewer.rating": -1, title: -1 }  
},  
{  
  $skip: 24  
},  
{  
  $limit: 1  
}  
)
```

[Proceed to next section](#)