CLASS 5

PROJECTORS OPERATORS

PROJECTION

MongoDB provides a special feature that is known as **Projection**. It allows you to select only the necessary data rather than selecting whole data from the document. For example, a document contains 5 fields, i.e.,\

```
{
name: "Roma",
age: 30,
branch: EEE,
department: "HR",
salary: 20000
}
```

But we only want to display the *name* and the *age* of the employee rather than displaying whole details. Now, here we use projection to display the name and age of the employee.

One can use projection with db.collection.find() method. In this method, the second parameter is the projection parameter, which is used to specify which fields are returned in the matching documents.

Syntax:

db.collection.find({}, {field1: value2, field2: value2, ..})

- If the value of the field is set to 1 or true, then it means the field will include in the return document.
- If the value of the field is set to 0 or false, then it means the field will not include in the return document.
- You are allowed to use projection operators, but find() method does not support following projection operators, i.e., $, $elemMatch, $slice, and $meta.
- There is no need to set _id field to 1 to return _id field, the find() method always return _id unless you set a _id field to 0.

Example:

Excluding specific fields:

```
Type "it" for more
db> db.Students.find({},{"name":1,"_id":0});
[
  { name: 'Student 948' }, { name: 'Student 157' },
  { name: 'Student 316' }, { name: 'Student 346' },
  { name: 'Student 930' }, { name: 'Student 305' },
  { name: 'Student 268' }, { name: 'Student 563' },
  { name: 'Student 440' }, { name: 'Student 536' },
  { name: 'Student 256' }, { name: 'Student 177' },
  { name: 'Student 871' }, { name: 'Student 487' },
  { name: 'Student 213' }, { name: 'Student 690' },
  { name: 'Student 368' }, { name: 'Student 172' },
  { name: 'Student 647' }, { name: 'Student 232' }
]
```

Retrive Name Age And GPA:

```
db> db.candidates.find({},{name:1,age:1,gpa:1});
[
  {
    _id: ObjectId('666490b3f378263ae82e00da'),
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4
  },
  {
    _id: ObjectId('666490b3f378263ae82e00db'),
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8
  },
  {
    _id: ObjectId('666490b3f378263ae82e00dc'),
    name: 'Charlie Lee',
    age: 19,
    gpa: 3.2
  },
  {
    _id: ObjectId('666490b3f378263ae82e00dd'),
    name: 'Emily Jones',
    age: 21,
    gpa: 3.6
  },
  {
    _id: ObjectId('666490b3f378263ae82e00de'),
    name: 'David Williams',
    age: 23,
    gpa: 3
  },
  {
    _id: ObjectId('666490b3f378263ae82e00df'),
    name: 'Fatima Brown',
    age: 18,
    gpa: 3.5
  },
```

To reduce the workload, MongoDB offers the below operators that can be used within a projection query:

- $elemMatch
- $slice

$elemMatch operator will limit the contents of an array to the first element that matches a given condition. This condition differs from the $ operator because the $elemMatch projection operator requires an explicit condition argument.

Let's see the syntax of using $elemMatch in find() method.

Syntax:

```
db.collection.find( { <array>: <condition> ... }, { "<array>.$elemMatch": ($elemMatch operator) } )
```

Limitations of the $elemMatch operator :

- Regardless of the ordering of fields in the document, the field to which $elemMatch projection is applied will be returned as the last field of the document.
- find() operations done on MongoDB views do not support $elemMatch projection operator.
- $text query expressions are not supported with the $elemMatch operator.

Example:

Finding Enrolled Candidates in English:

```
db> db.candidates.find({courses:{$elemMatch:{$eq:"English"}}},{name:1,"cours
es.$":1});
[
  {
    _id: ObjectId('666af15334bca5162beff948'),
    name: 'Alice Smith',
    courses: [ 'English' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94a'),
    name: 'Charlie Lee',
    courses: [ 'English' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94c'),
    name: 'David Williams',
    courses: [ 'English' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff950'),
    name: 'Isaac Clark',
    courses: [ 'English' ]
  }
]
```

$slice Projection Operator:

The $slice operator specifies the number of elements that should be returned as the output of a query.

Syntax:

db.collection.find(<query>, { <array Field>: { $slice: <number> } })

Limitations in $slice operator:

- With the introduction of MongoDB 4.4, the $slice operator will only return the sliced element. It will not return any other item in a nested array.
- The $slice operator does not support the find() operation done on MongoDB views.
- The $slice operator cannot be used in conjunction with the $ projection operator due to the MongoDB restriction, where top-level fields can't consist of $ (dollar sign) as a part of the field name.
- Queries can't contain the $slice of an array and a field embedded in the array as part of the same statement to eliminate path collisions from MongoDB 4.4.

Example:

Filter or remove First two index value

db.data.find({"Name":"Vinoth"},{"Skill":{$slice:-2}})

```
student> db.data.find({"Name":"Vinoth"},{"Skill":{$slice:-2}});
[
  {
    _id: ObjectId("6503626cb63da7242a8c6c29"),
    Name: 'Vinoth',
    Age: 23,
    Gender: 'Male',
    Skill: [ 'java', 'python' ],
    University: 'Anna University '
  }
]
student>
```

- **db.data.find({"Name":"Vinoth"}):** This part of the code performs a find operation in the "data" collection. It searches for documents that meet the specified criteria, where the "Name" field is equal to "Vinoth."
- **{"Skill": {$slice: -2}}:** After the find operation, the projection part of the query specifies which fields to include in the result. In this case, it's specifying that only the "Skill" field should be included. The $slice operator is applied to the "Skill" field with a negative value of -2.

When you use a negative value with $slice, it indicates that you want to include the last N elements of the array, where N is the absolute value of the negative number. In this case, it includes the last 2 elements of the "Skill" array.

So, when you run this code against your MongoDB database, it will return all documents where "Name" is "Vinoth," and for each of those documents, it will include only the last 2 elements of the "Skill" array in the result.

Example (2):

db.data.find({"Name":"Sara"},{"Skill":{$slice:[1,2]}})

```
student> db.data.find({"Name":"Sara"},{"Skill":{$slice:[1,2]}});
[
  {
    _id: ObjectId("6503626cb63da7242a8c6c2b"),
    Name: 'Sara',
    Age: 28,
    Gender: 'Female',
    Skill: [ 'express js', 'react js' ],
    University: 'Annamalai University '
  }
]
student>
```