Class 5: Projection Operators

Agenda:-

- Understand Projection operators:-

  Create and demonstrate how projection operators ($, $elematch and $slice) would be used in the MongoDB.

Projection Operators:-

Projection operators in MongoDB are used to control which fields you retrieve from a document when using the find() method. By default, find() returns all fields in a document. Projection operators allow you to specify which fields you want to see in the results, reducing the amount of data transferred and potentially improving query performance.

Here are some key points about projection operators in MongoDB:-

- **Benefits:-**
  - Reduced data transfer: By only returning the necessary fields, you can significantly reduce the amount of data transferred between the database server and your application.
  - Improved performance: Less data to process can lead to faster queries.

- **Common Projection Operators:-**
  - **$:-** Projects a specific field by name.
  - **$elemMatch:-** Projects the first element in an array that matches a specific condition.
  - **$slice:-** Limits the number of elements returned from an array.

- **Usage:-**
  - Projection operators are used within the second argument of the find() method.
  - You can specify which fields to include or exclude using a document where the field names are the keys and the values are either 1 (include) or 0 (exclude).

Name	Description
$	Projects the first element in an array that matches the query condition.
$elemMatch	Projects the first element in an array that matches the specified $elemMatch condition.
$meta	Projects the available per-document metadata.
$slice	Limits the number of elements projected from an array. Supports skip and limit slices.

1. $ (projection):-

The positional $ operator limits the contents of an <array> to return the first element that matches the query condition on the array.

Use $ in the projection document of the find() method or the findOne() method when you only need one particular array element in selected documents.

See the aggregation operator $filter to return an array with only those elements that match the specified condition.

Syntax:-

To return the first array element that matches the specified query condition on the array:

```
db.collection.find( { <array>: <condition> ... }, { "<array>.$": 1 } )
```

```
db.collection.find( { <array.field>: <condition> ...},{ "<array>.$": 1 } )
```

2. $elemMatch (projection):-

$elemMatch operator and its use in the given query. The $elemMatch operator is used in MongoDB to match documents that contain an array field with at least one element that matches all the specified query criteria.

Usage of $elemMatch :-

The $elemMatch operator is typically used in two contexts:-

1. Querying Arrays: To find documents where at least one element in an array field matches the specified criteria.

2. Projection: To project the first element in an array that matches the specified criteria.

```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Physics"}}},{name:1,"courses.$":1});
[
  { _id: ObjectId('6657ff95946a866dbb971e60'), name: 'Bob Johnson' },
  { _id: ObjectId('6657ff95946a866dbb971e62'), name: 'Emily Jones' }
]
db>
```

Ex:-

Db.candidates.find({ courses: { $elemMatch: { $eq: “Computer Science”}, {name: 1, “courses.$”: 1 });

Output:

{ "name": "Alice", "courses": "Computer Science" }

{ "name": "Bob", "courses": "Computer Science" }

2. Projection Operator ($elemMatch):

Example 2: Find Candidates Enrolled in "Computer Science" with Specific Projection

JavaScript

```
db.candidates.find({ courses: { $elemMatch: { $eq: "Computer Science" } }
  { name: 1, "courses.$": 1 }); // Include only matched course
```

Here’s a breakdown of the code:

1. db.candidates.find({}): This line uses the find method to query the candidates collection in the database. An empty curly brace {} specifies that all documents in the collection should be matched.
2. .courses: { $elemMatch: { $eq: "Computer Science" }}: This part of the query filters the documents based on the courses field. The $elemMatch operator checks for an array element in the courses field that matches the specified condition.

- In this case, the condition is $eq: "Computer Science", which means it's looking for an element in the courses array that is equal to the string "Computer Science".

3. .name: 1, "courses.$": 1: This part of the query specifies which fields to include in the projection using the projection operator.
   - name: 1 includes the name field of the candidate document.
   - "courses.$" is a special notation that refers to the matched element within the courses array. So it's including only the element that matched the $eq: "Computer Science" condition.

3. $slice:-

The $slice projection operator specifies the number of elements in an array to return in the query result.

Syntax:

The $slice has one of the following syntax forms:

```
db.collection.find(
<query>,
{ <arrayField>: { $slice: <number> } }
);
```

Or

```
db.collection.find(
<query>,
{ <arrayField>: { $slice: [ <number>, <number> ] } }
);
```

Usage of $slice:-

There are three common ways to use the $slice operator:

1.Retrieve the first n elements from an array:

```
db.collection.find({}, { arrayField: { $slice: n } })
```

This will return the first n elements of the array arrayField.

2.Retrieve the last n elements from an array:

```
db.collection.find({}, { arrayField: { $slice: -n } })
```

This will return the last n elements of the array arrayField.

3.Retrieve a subset starting at a specific position:

db.collection.find({}, { arrayField: { $slice: [start, length] } })

This will return length elements starting from the start position of the array arrayField.

$slice operation:-

The $slice operation in MongoDB is a powerful tool for limiting the number of elements returned from an array within a document. It can be used in projections to retrieve a subset of an array, which is particularly useful for applications like pagination, data previews, and efficient data retrieval.

```
db> db.candidates.find({},{courses:{$slice:1}})
[
  {
    _id: ObjectId('6657ff95946a866dbb971e5f'),
    name: 'Alice Smith',
    age: 20,
    courses: [ 'English' ],
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e60'),
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Computer Science' ],
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e61'),
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'History' ],
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e62'),
    name: 'Emily Jones',
    age: 21,
    courses: [ 'Mathematics' ],
    gpa: 3.6,
    home_city: 'Houston',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e63'),
    name: 'David Williams',
    age: 23,
    courses: [ 'English' ],
    gpa: 3,
    home_city: 'Phoenix',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e64'),
    name: 'Fatima Brown',
    age: 18,
    courses: [ 'Biology' ],
    gpa: 3.5,
    home_city: 'San Antonio',
    blood_group: 'B+',
```

db.candidates.find({}, {courses: {$slice: 1}})

This query does the following:

1. db.candidates.find({}): This part of the query is selecting all documents in the candidates collection.

2. {courses:{$slice: 1}}: This projection specifies that only the first element of the courses array should be included in the result.

Explanation:-

The query returns all documents in the candidates collection, but for each document, it only includes the first element of the courses array.

Before Slicing:-

Consider a document in the candidates collection before applying the $slice operation:


```
{
 "_id": ObjectId("6657f9f9946a866dbb971e5f"),
 "name": "Alice Smith",
 "age": 20,
 "courses": ["English", "Math", "Science"],
 "gpa": 3.4,
 "home_city": "New York City",
 "blood_group": "A+",
 "is_hotel_resident": true
}
```


After Slicing:-

After applying the $slice: 1 projection, the document would look like:


```
{
 "_id": ObjectId("6657f9f9946a866dbb971e5f"),
 "name": "Alice Smith",
 "age": 20,
 "courses": ["English"],
 "gpa": 3.4,
 "home_city": "New York City",
 "blood_group": "A+",
 "is_hotel_resident": true
}
```


Only the first element of the courses array ("English") is included in the output.

Result Set:-

The results in the image show several documents from the candidates collection after applying the $slice operation. Each document includes only the first element of the courses array.