5.Projection Operators

Projection:

Projection operators specify the fields returned by an operation.

find() operations on views do not support the following Query and Projection Operators operators:

- $
- $elemMatch
- $slice
- $meta

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. $ (Dollar Sign):

- **Function:** This symbol generally precedes actual query and projection operators. It signifies that the following expression is a MongoDB operator.

Example:

```
db.products.find({ price: { $gt: 10 } }) // $gt is a comparison operator
```

Output:

```
{ "_id": ObjectId("..."), "name": "Headphones", "price": 15.99 } { "_id":
ObjectId("..."), "name": "Laptop", "price": 799.99 } { "_id": ObjectId("..."), "name":
"Smartwatch", "price": 249.99 } ... (depending on your data)
```

2.$elemMatch (Element Match):

- **Function:** Filters documents based on an element within an array field.
- **Syntax:** { arrayField: { $elemMatch: { element_condition1: value1, ... } } }

Example:

```
db.orders.find({ items: { $elemMatch: { product_id: "123", quantity: { $gt: 1 } } } })
```

Output:

```
Order ID: ObjectId("...") // Assuming an ObjectId field named "_id" exists Items: -
product_id: "123", quantity: 2 // Example item matching the criteria - product_id:
"456", quantity: 1 // Other items in the order (optional) ... (multiple orders if they
exist)
```

3. $slice (Slice):

- **Function:** Projects a specific subset of elements from an array field.
- **Syntax:** { arrayField: { $slice: [start_index, number_of_elements] } }

Example:

```
db.posts.find({}, { projection: { comments: { $slice: 2 } } }) // Get only the first two
comments
```

Output:

```
Post ID: ObjectId("...") // Assuming an ObjectId field named "_id" exists Title:
<post title> (if a title field exists) Comments: - <comment 1 text> // First comment
from the array - <comment 2 text> // Second comment from the array ... (multiple
posts if they exist)
```

4.$meta (Metadata):

- **Function:** Accesses and returns metadata about the query execution. It's rarely used directly in queries.
- **Syntax:** { $meta: { field_name } } (field_name can be textScore, location, etc.)

Example:

While not used in typical queries, $meta can be helpful for debugging purposes. For instance, $meta: { textScore: 1 } within a $text search can show the relevance score for each document.

Explaining $meta and Projection Operators: https://www.bmc.com/blogs/mongodb-overview-getting-started-with-mongodb/