MONGODB

CLASS 5 : PROJECTION OPERATORS

In MongoDB, projection operators allow you to control which fields are returned from documents when you execute a query. This is a valuable feature for several reasons:

- **Reduced Data Transfer:** By specifying only the fields you need, you minimize the amount of data transferred between the database and your application, improving performance.
- **Optimized Processing:** Your application only needs to process the returned fields, leading to faster client-side or server-side processing.
- **Data Privacy:** You can exclude sensitive fields from the results, enhancing data security if needed.

Projection Operators:

MongoDB offers two primary projection operators:

- **`{ include: [field1, field2, ...] }:`** Includes only the specified fields in the results.
- **`{ exclude: [field1, field2, ...] }:`** Excludes the specified fields from the results. All other fields are included.

Example: Including Specific Fields

Suppose you have a collection named `products` with the following fields:

```
{
  "_id": ObjectId(...),
  "name": "T-Shirt",
  "price": 25.99,
  "description": "A comfortable cotton T-Shirt",
  "category": "Clothing"
}
```

Agenda

Understand Projection

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

Projection

Create new collection called candidates.

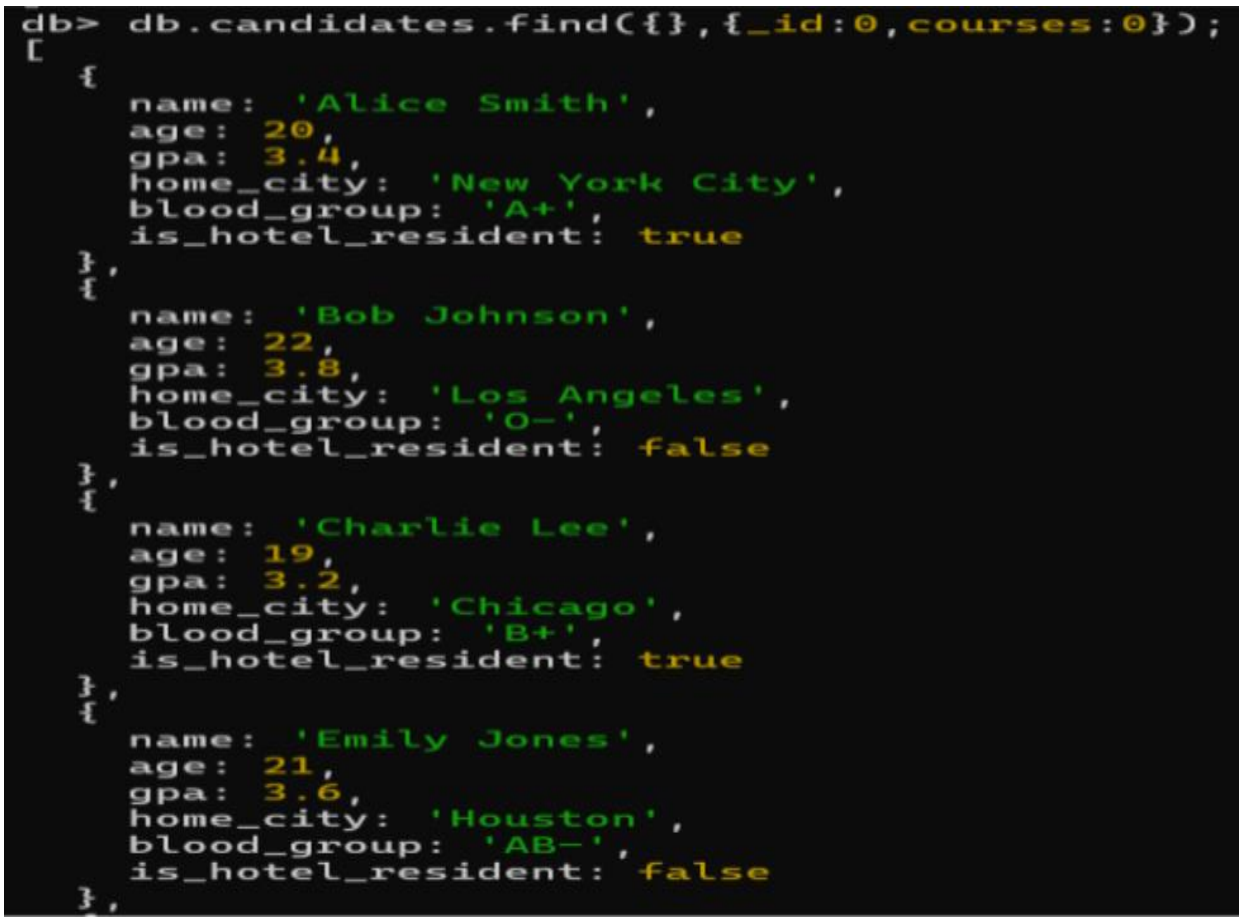
```
_id: ObjectId('665752830959f4120ac93d06')
name : "Emily Jones"
age : 21
courses : Array (3)
gpa : 3.6
home_city : "Houston"
blood_group : "AB-"
is_hotel_resident : false
```

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
  },
```

In this MongoDB query, `db.candidates.find({},{name:1,age:1,gpa:1})`, the `find` method is used to retrieve documents from the `candidates` collection. The first argument, `{}`, is an empty query filter, meaning all documents in the collection are selected. The second argument, `{name:1, age:1, gpa:1}`, is a projection that specifies only the `name`, `age`, and `gpa` fields should be included in the output, while excluding other fields. The result is a list of documents where each document contains only the `_id`, `name`, `age`, and `gpa` fields. The output includes details of candidates such as Alice Smith, Bob Johnson, and Charlie Lee, displaying their names, ages, and GPAs, providing a concise view of this specific subset of data from the collection.

VARIATION:EXCLUDE FIELDS:



In MongoDB, you can exclude specific fields from your query results using the `exclude` operator within the projection document. This is a valuable technique for:

Reduced Data Transfer: Minimize the amount of data transferred between the database and your application, improving performance.

Optimized Processing: Your application only needs to process the returned fields, leading to faster client-side or server-side processing.

Data Privacy: Exclude sensitive fields from the results, enhancing data security if needed.

PROJECTION OPERATOR($elemMatch):

In MongoDB, the `$elemMatch` projection operator is specifically used for projecting a single element from an array field based on a condition within the query. It's not directly applicable for general projection purposes (including or excluding fields).

Find candidates enrolled in “Computer Science” with specific projection:

```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1});
[
  {
    _id: ObjectId('666490b3f378263ae82e00db'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00e0'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00e4'),
    name: 'Kevin Lewis',
    courses: [ 'Computer Science' ]
  }
]
```

The MongoDB query retrieves documents from the `candidates` collection where the `courses` array includes the element "Computer Science". The query is structured to filter and project specific fields. The filter condition `{ courses: { $elemMatch: { $eq: "Computer Science" } } }` ensures that only documents with "Computer Science" as one of the courses are selected. The projection `{ name: 1, "courses.$": 1 }` includes only the `name` field and the specific array element from the `courses` array that matched the filter condition.

$elemMatch:

In MongoDB, the `$elemMatch` operator is used to filter documents based on conditions within an array field. It allows you to search for documents where at least one element in the array matches all the specified criteria.

PROJECTION OPERATOR($slice):

In MongoDB, the `$slice` projection operator is used to control how many elements are returned from an array field in your query results. It essentially slices a portion of the array for inclusion in the document you retrieve.

Retrieve all candidates with first two courses:

```
db> db.candidates.find({},{name:1,courses:{$slice:2}});
[
  {
    _id: ObjectId('666490b3f378263ae82e00da'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00db'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00dc'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00dd'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00de'),
    name: 'David Williams',
    courses: [ 'English', 'Literature' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00df'),
    name: 'Fatima Brown',
    courses: [ 'Biology', 'Chemistry' ]
  },
  {
    _id: ObjectId('666490b3f378263ae82e00e0'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science', 'Engineering' ]
  },
```

This MongoDB query retrieves documents from the `candidates` collection, projecting only the `name` and `courses` fields. However, it limits the number of elements returned in the `courses` array to 2 using the `$slice` projection operator.

$Slice:

The `$slice` projection operator is used to retrieve a specific portion of elements from an array field in your MongoDB query results. It essentially acts like slicing a part of the array for inclusion in the document you retrieve.

Examples:

1. **Retrieving First 3 Elements:**

db.posts.find({}, { "comments": { $slice: 3 } })

This query fetches all documents from the "posts" collection and includes the "comments" field, but only the first 3 elements from the comments array.

2. **Getting Last 5 Items:**

db.orders.find({}, { "items": { $slice: -5 } })

This query retrieves all documents from the "orders" collection and includes the "items" field, containing the last 5 elements of the items array.

3. **Skipping 10 and Taking Next 20:**

```
db.products.find({}, { "images": { $slice: [10, 20] } })
```

This query finds all documents from the "products" collection and includes the "images" field. However, it skips the first 10 elements and then returns the following 20 elements from the images array.

`$slice` only affects the targeted array field in the projection. It doesn't exclude other fields in the document.

When using `$slice` with negative values for skipping or limiting, the results are based on the zero-based index of the array. The first element has an index of 0, the second element has an index of 1, and so on.