Projection Operators

Projection:

In MongoDB, projections allow you to control which fields are included or excluded when retrieving data from a collection.

Syntax:

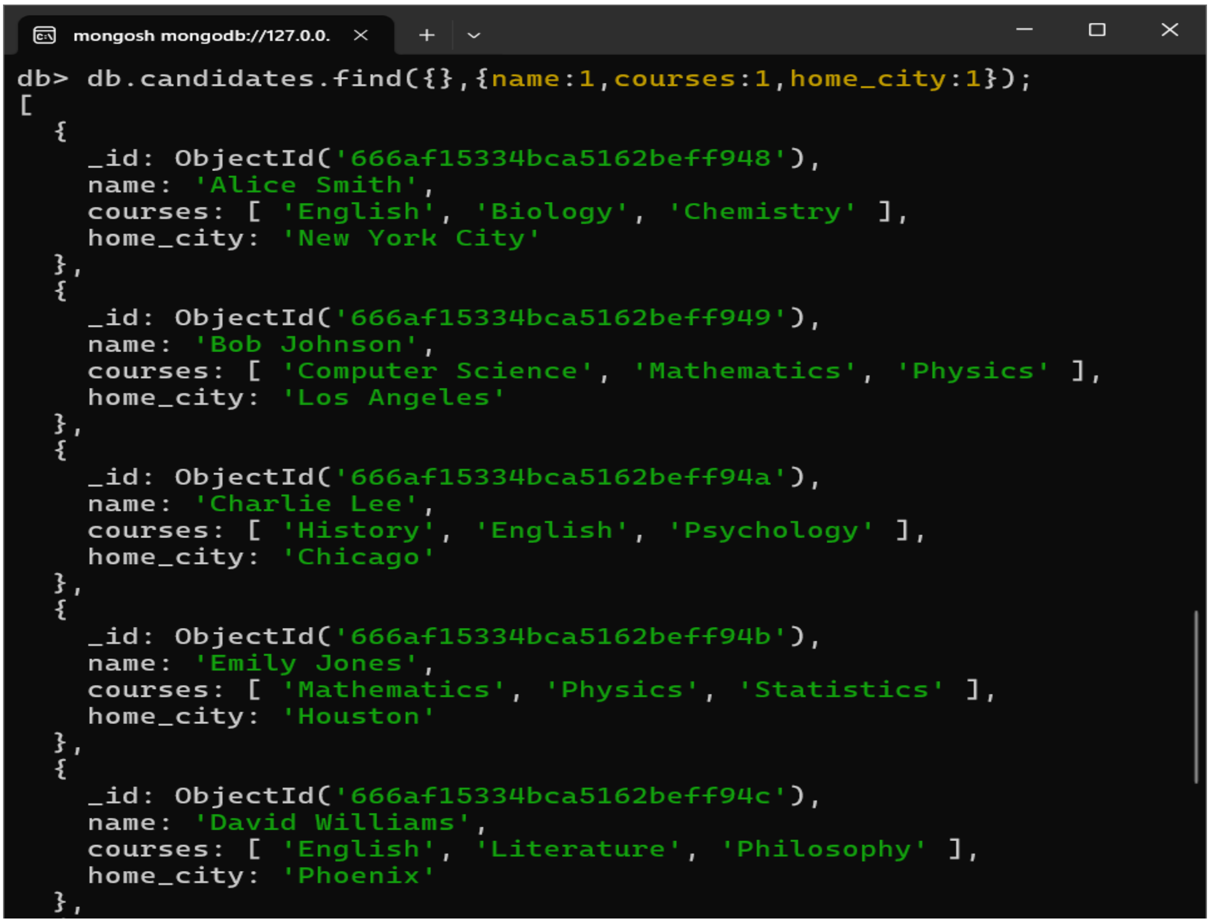
```
db.candidates.find(

{ /* query document */ },

 { /* projection document */ })
```

Retrieve Name, Courses and Home city:

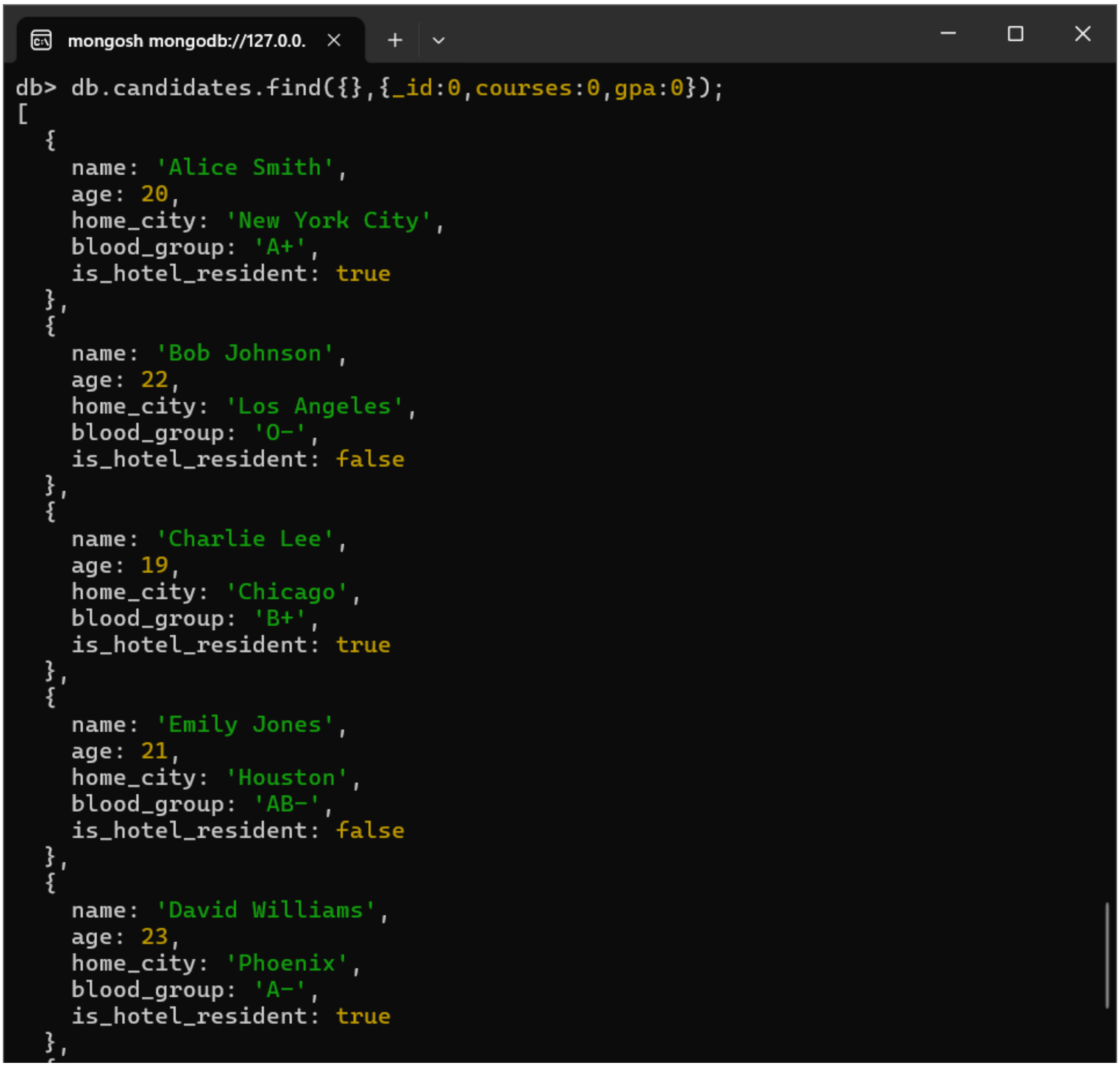
- **Inclusion:** Set a field's value to 1 in the projection document to include it in the returned results.



```
db> db.candidates.find({},{name:1,courses:1,home_city:1});
[
  {
    _id: ObjectId('666af15334bca5162beff948'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology', 'Chemistry' ],
    home_city: 'New York City'
  },
  {
    _id: ObjectId('666af15334bca5162beff949'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics', 'Physics' ],
    home_city: 'Los Angeles'
  },
  {
    _id: ObjectId('666af15334bca5162beff94a'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English', 'Psychology' ],
    home_city: 'Chicago'
  },
  {
    _id: ObjectId('666af15334bca5162beff94b'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics', 'Statistics' ],
    home_city: 'Houston'
  },
  {
    _id: ObjectId('666af15334bca5162beff94c'),
    name: 'David Williams',
    courses: [ 'English', 'Literature', 'Philosophy' ],
    home_city: 'Phoenix'
  },
```

Ignore Age and Gpa:

- **Exclusion:** Set a field's value to 0 to exclude it.



```
db> db.candidates.find({},{_id:0,courses:0,gpa:0});
[
  {
    name: 'Alice Smith',
    age: 20,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    name: 'Bob Johnson',
    age: 22,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    name: 'Charlie Lee',
    age: 19,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    name: 'Emily Jones',
    age: 21,
    home_city: 'Houston',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
    name: 'David Williams',
    age: 23,
    home_city: 'Phoenix',
    blood_group: 'A-',
    is_hotel_resident: true
  },
```

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

Collection Name:Candidates

Projection operators:

- $elemMatch
- $slice

$elemMatch:

- The $elemMatch operator in MongoDB is a powerful tool for querying documents that contain arrays and finding elements within those arrays that meet specific criteria.
- **Matches Documents:** $elemMatch targets documents where an array field holds at least one element that satisfies all the conditions you specify within the operator.
- **Filtering by Array Elements:** It essentially filters the documents based on the properties of individual elements within an array field.

Syntax:

```
{
<array_field>: { $elemMatch: { <condition1>, <condition2>, ... } }
 }
```

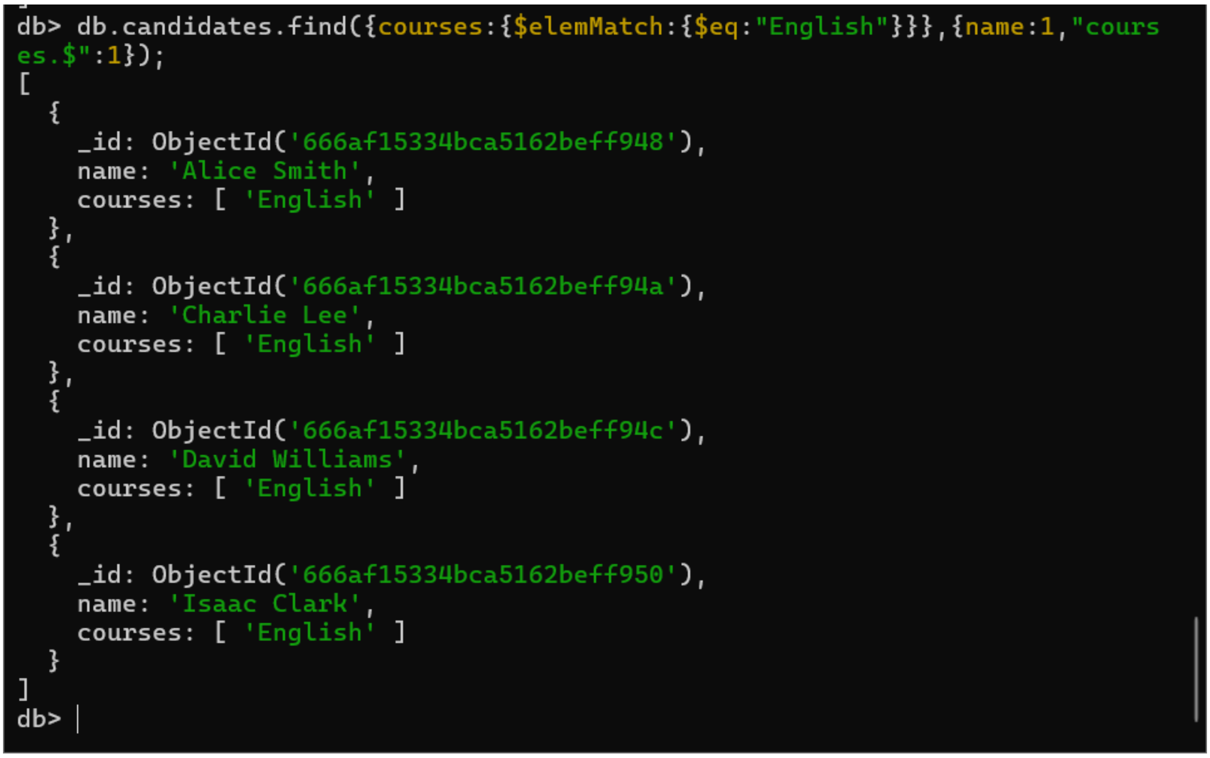
- ✓ <array_field>: The name of the field in your document that contains the array you want to filter.
- ✓ <condition1>, <condition2>: These are standard MongoDB query operators like $eq, $gt, $in, etc., used to define the criteria for matching elements within the array. You can specify multiple conditions for more complex filtering.

- **Matching at Least One Element:** $elemMatch finds documents with at least one element in the array that fulfills all the specified conditions. If no element matches, the entire document is excluded.

➢ **Single Condition Shortcut:** If you only have a single condition without $not or $ne operators, you can often omit the $elemMatch wrapper. MongoDB will automatically understand the intent.

➢ **Nesting and Combining:** The $elemMatch operator can be nested within other query operators for more intricate filtering scenarios.

//Finding Candidates Enrooled in English with specific Projection.



```
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' ]
  }
]
db> |
```

✓ name: 1: Includes the name field from each matching document.

✓ "courses.$": This is a special projection syntax using the $ wildcard operator.

✓ It instructs MongoDB to include all fields within the matched elements of the courses array.

✓ Since $elemMatch filters for documents with "Computer Science", this will effectively include only the "English" element from the courses array (if it exists) for each matching document.

✓ The $ operator projects the first matching array element from each document in a collection based on some condition from the query statement.

$slice:

The $slice projection operator in MongoDB is a powerful tool for extracting specific portions of arrays from documents during retrieval.

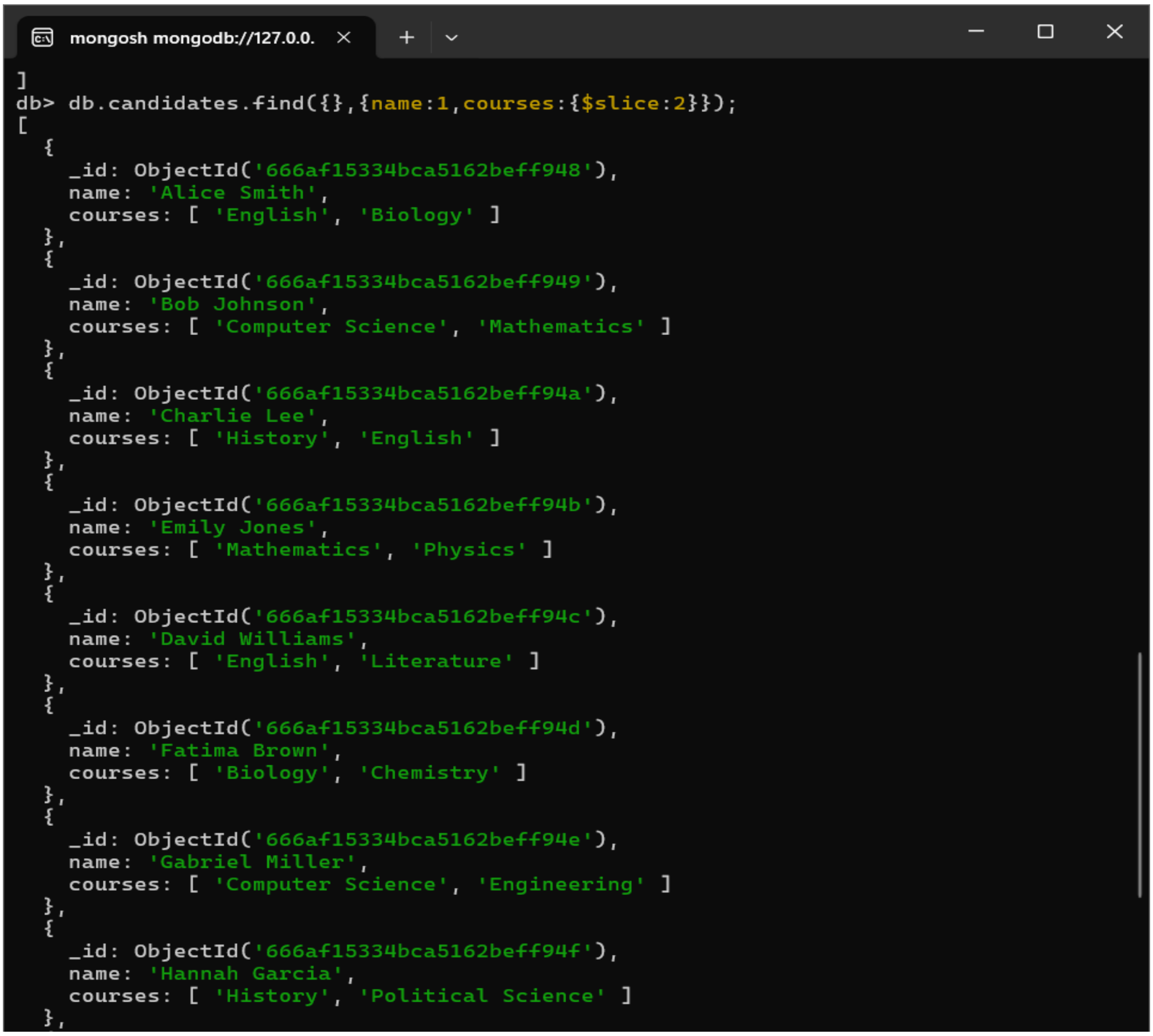
- **Extracting Subsets:** $slice allows you to select a designated number of elements from the beginning or end of an array field within a document.
- **Zero-Based Indexing:** It uses zero-based indexing, meaning the first element in the array has an index of 0, the second has an index of 1, and so on.
- Can specify both start_index and number_to_slice to extract a specific range of elements.
- If <number_to_slice> is greater than the remaining elements in the array after considering <start_index>, all elements from <start_index> to the end are returned.

Syntax:

```
{ <array_field>: { $slice: [<start_index>, <number_to_slice>] } }
```

- array_field>: The name of the array field you want to extract elements from.
- <start_index> (Optional): The starting index (inclusive) of the slice. Defaults to 0 (beginning of the array) if omitted.
- <number_to_slice> (Optional): The number of elements to include in the slice. Defaults to all remaining elements from the <start_index> to the end if omitted.

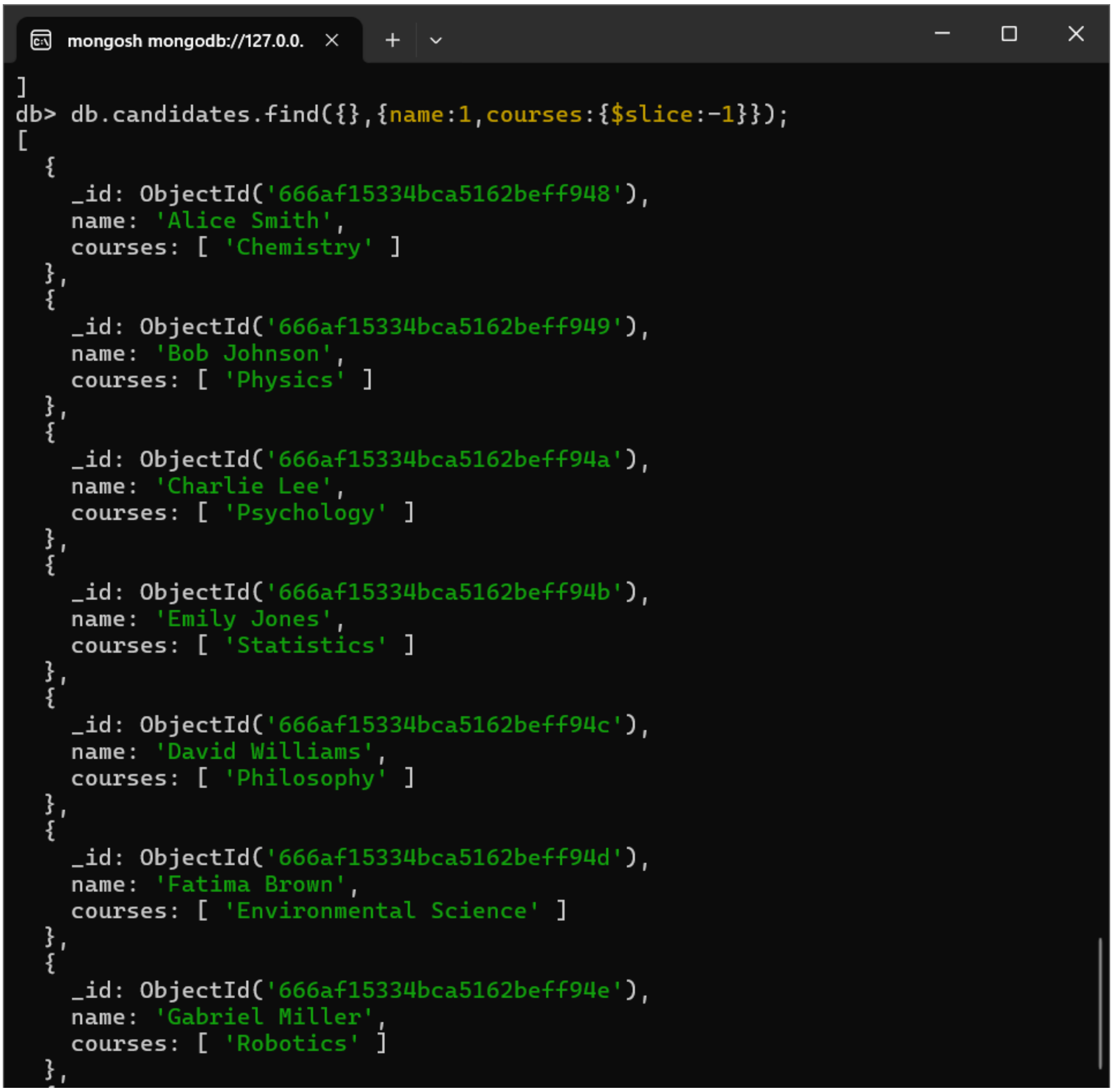
//Retrieving First Two Elements:



```
]
db> db.candidates.find({},{name:1,courses:{$slice:2}});
[
  {
    _id: ObjectId('666af15334bca5162beff948'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff949'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94a'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94b'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94c'),
    name: 'David Williams',
    courses: [ 'English', 'Literature' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94d'),
    name: 'Fatima Brown',
    courses: [ 'Biology', 'Chemistry' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94e'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science', 'Engineering' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94f'),
    name: 'Hannah Garcia',
    courses: [ 'History', 'Political Science' ]
  },
```

- ✓ courses: { $slice: 2 }: This part applies the $slice projection operator to the courses field.
- ✓ $slice extracts a specific portion of an array.
- ✓ In this case, 2 specifies that only the first two elements (zero-based indexing) from the courses array should be included in the projection.
- ✓ If a candidate has fewer than two courses, only the available courses will be returned.

//Getting Last One Element:



```
]
db> db.candidates.find({},{name:1,courses:{$slice:-1}});
[
  {
    _id: ObjectId('666af15334bca5162beff948'),
    name: 'Alice Smith',
    courses: [ 'Chemistry' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff949'),
    name: 'Bob Johnson',
    courses: [ 'Physics' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94a'),
    name: 'Charlie Lee',
    courses: [ 'Psychology' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94b'),
    name: 'Emily Jones',
    courses: [ 'Statistics' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94c'),
    name: 'David Williams',
    courses: [ 'Philosophy' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94d'),
    name: 'Fatima Brown',
    courses: [ 'Environmental Science' ]
  },
  {
    _id: ObjectId('666af15334bca5162beff94e'),
    name: 'Gabriel Miller',
    courses: [ 'Robotics' ]
  },
```

- ✓ The second curly braces {} define the projection document to control which fields are returned.
  - ○ name: 1: Includes the name field for each candidate.
  - ○ courses: { $slice: -1 }: Applies the $slice operator to the courses field.
- ✓ While negative values are allowed for start_index in $slice, -1 in this context has a special meaning.
- ✓ In MongoDB, -1 alone as the argument to $slice is generally treated as an instruction to return **only the last element** of the array.