PROJECTION OPERATORS

PROJECTION:

In MongoDB, projection refers to the process of specifying which fields should be included or excluded in the documents that are returned by a query. This is done to limit the amount of data that is retrieved.

When performing a query in MongoDB, we can use projection to:

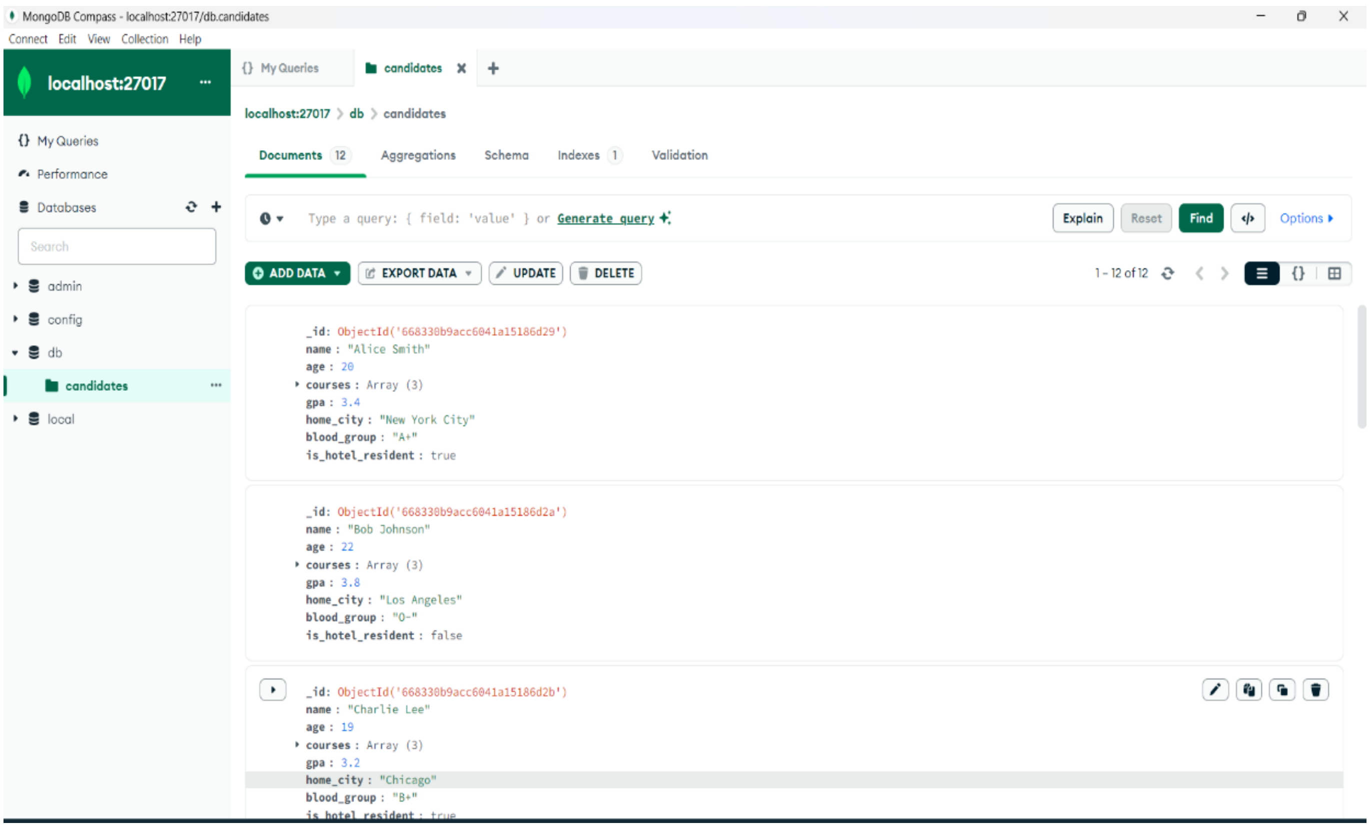
1. INCLUDE SPECIFIC FIELDS:

It specify which fields want to be included in the result set.

2. EXCLUDE SPECIFIC FIELDS:

It specify which fields want to be excluded from the result set.

First we need to import a collection called "**candidates**" to the mongocompass.



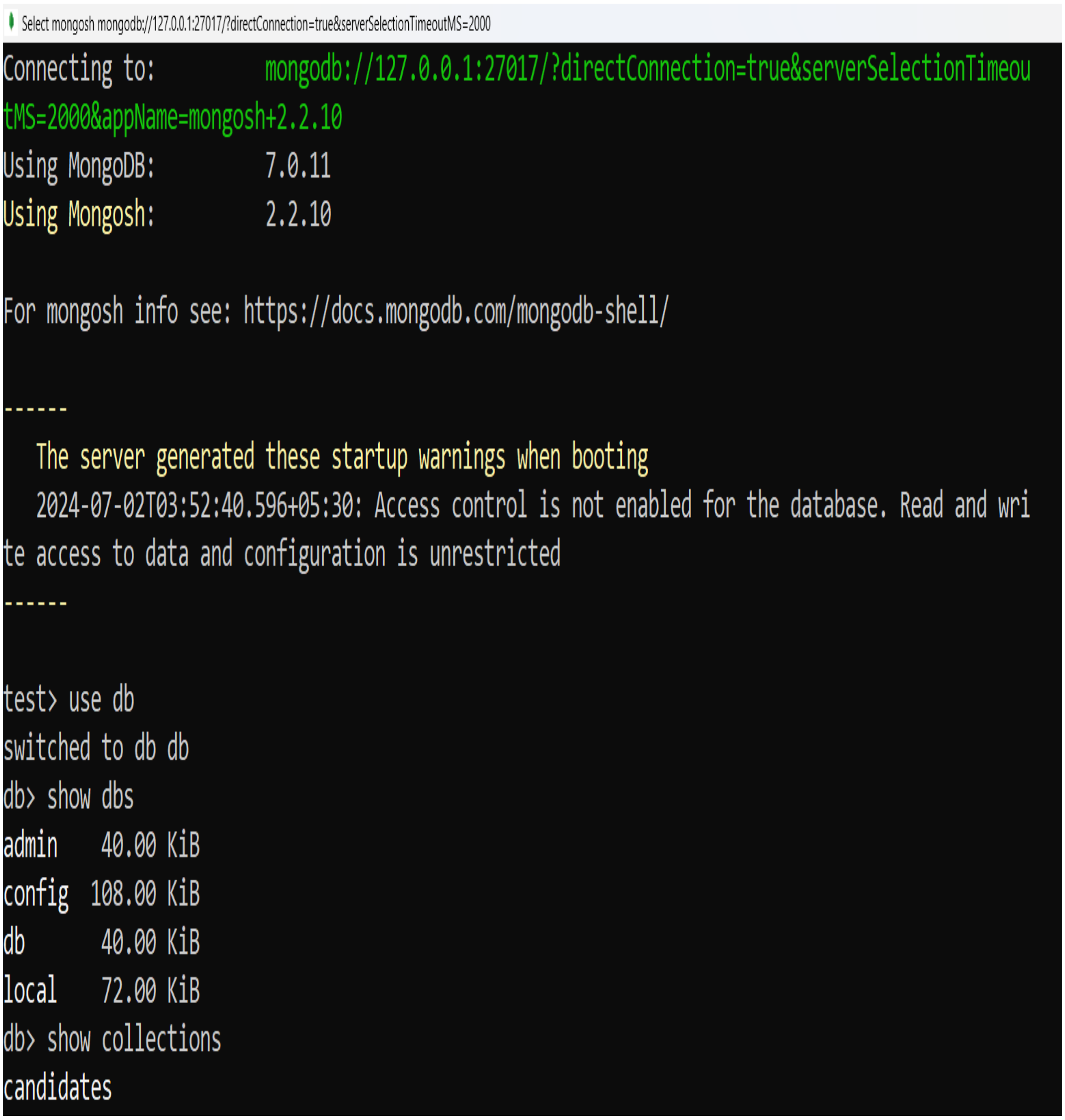
```
_id: ObjectId('668330b9acc6041a15186d29')
name : "Alice Smith"
age : 20
courses : Array (3)
gpa : 3.4
home_city : "New York City"
blood_group : "A+"
is_hotel_resident : true

_id: ObjectId('668330b9acc6041a15186d2a')
name : "Bob Johnson"
age : 22
courses : Array (3)
gpa : 3.8
home_city : "Los Angeles"
blood_group : "O-"
is_hotel_resident : false

_id: ObjectId('668330b9acc6041a15186d2b')
name : "Charlie Lee"
age : 19
courses : Array (3)
gpa : 3.2
home_city : "Chicago"
blood_group : "B+"
is_hotel_resident : true
```

To check the collection we use commands :

Use db

Show dbs

Show collections

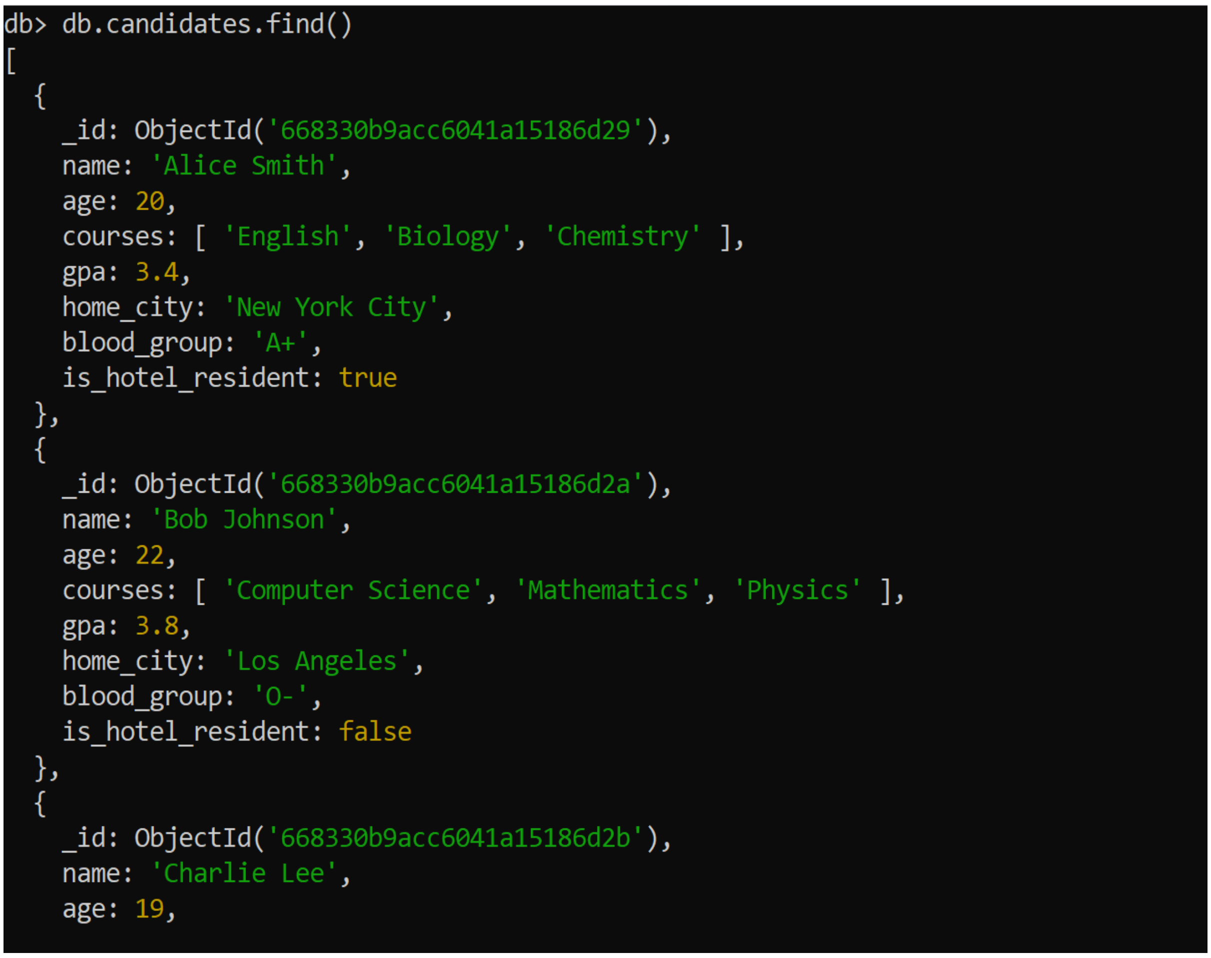


```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeou
tMS=2000&appName=mongosh+2.2.10
Using MongoDB:          7.0.11
Using Mongosh:          2.2.10

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


------
   The server generated these startup warnings when booting
   2024-07-02T03:52:40.596+05:30: Access control is not enabled for the database. Read and wri
te access to data and configuration is unrestricted
------


test> use db
switched to db db
db> show dbs
admin    40.00 KiB
config  108.00 KiB
db       40.00 KiB
local    72.00 KiB
db> show collections
candidates
```

Now the collection “candidates” is displayed on the command prompt.

Here to find candidates data which is used in collection we use command

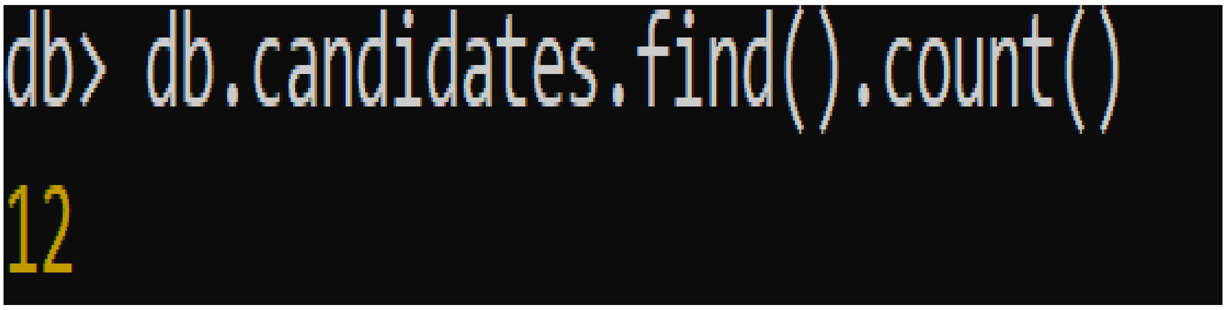
db.candidates.find()



```
db> db.candidates.find()
[
  {
    _id: ObjectId('668330b9acc6041a15186d29'),
    name: 'Alice Smith',
    age: 20,
    courses: [ 'English', 'Biology', 'Chemistry' ],
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Computer Science', 'Mathematics', 'Physics' ],
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2b'),
    name: 'Charlie Lee',
    age: 19,
```

To find number of counts of candidates we use

db.candidates.find().count()

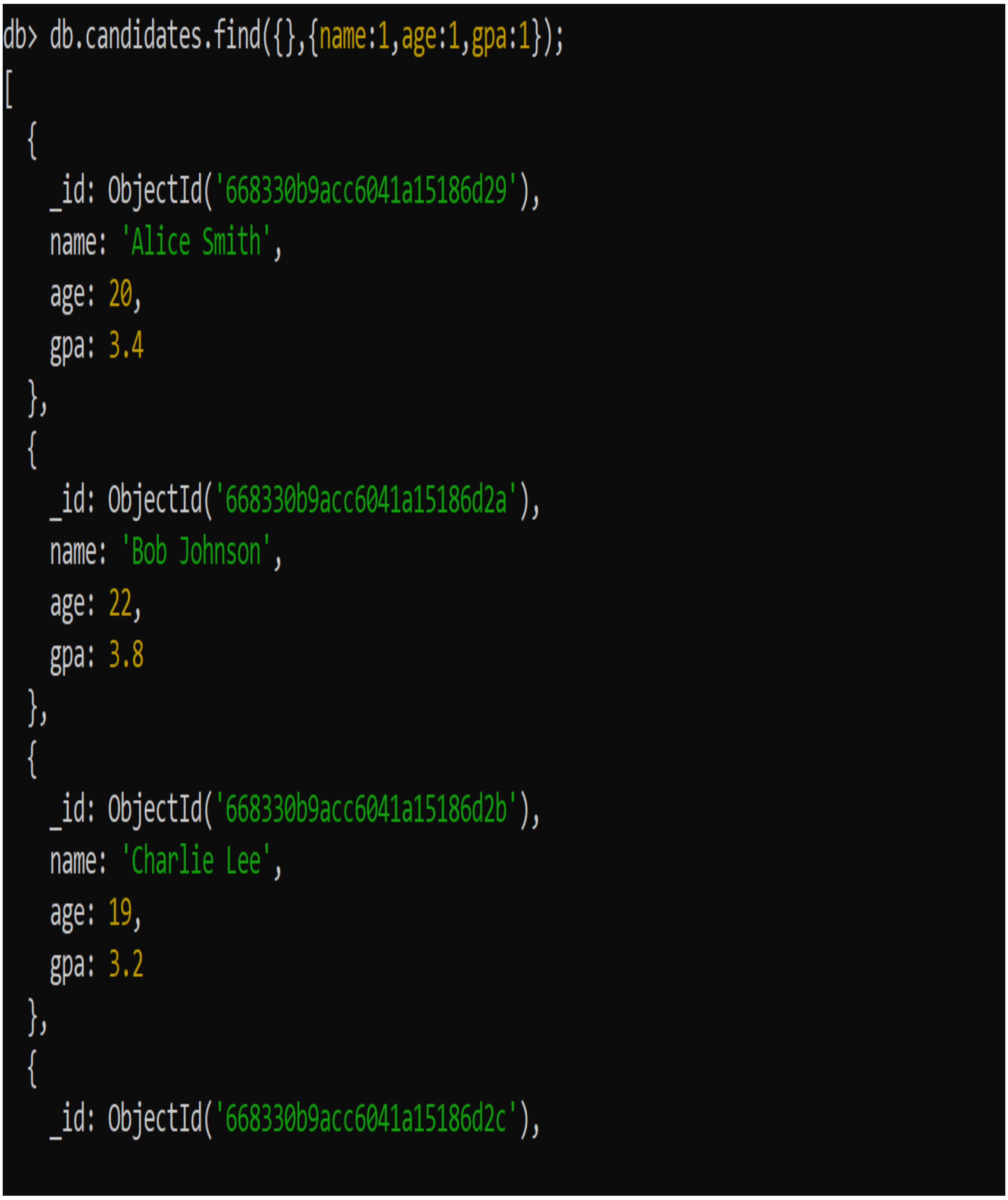


```
db> db.candidates.find().count()
12
```

Output displayed as 12 candidates.

To retrieve Name , Age and Gpa of candidates we use command

db.candidates.find({},{name:1,age:1,gpa:1});



```
db> db.candidates.find({},{name:1,age:1,gpa:1});
[
  {
    _id: ObjectId('668330b9acc6041a15186d29'),
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2b'),
    name: 'Charlie Lee',
    age: 19,
    gpa: 3.2
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2c'),
```

displays a candidates list only with name ,age and gpa with _id.

The most common projection operators in MongoDB are:

1. Inclusion (1): This operator is used to include specific fields in the query results.

2. Exclusion (0): This operator is used to exclude specific fields from the query results.

3. Slice ($slice): This operator limits the number of array elements that are returned.

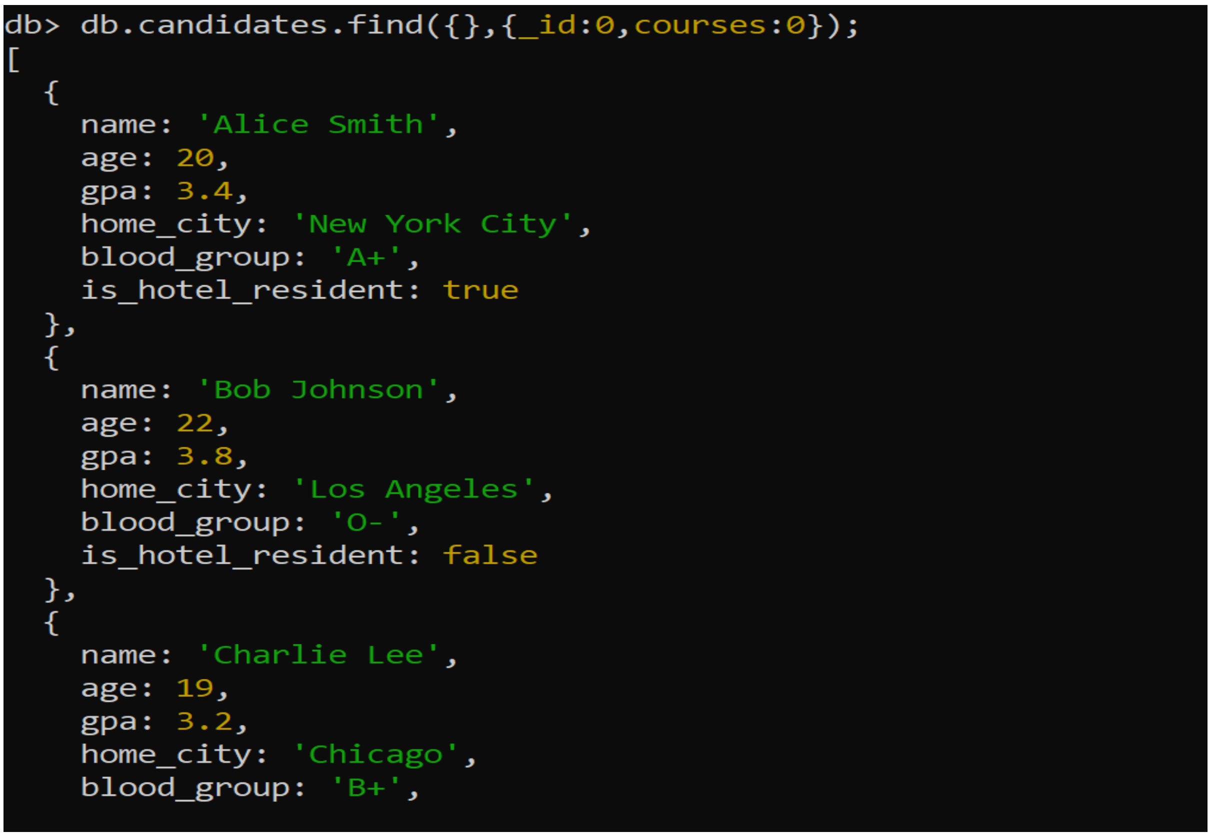
4. ElemMatch ($elemMatch): This operator projects only the first element from an array that matches the specified condition.

5. Meta ($meta): This operator can include metadata in the query results, such as text search scores.

1.Exclude fields(0):

Here to retrieve data of candidates excluding _id and course details we use

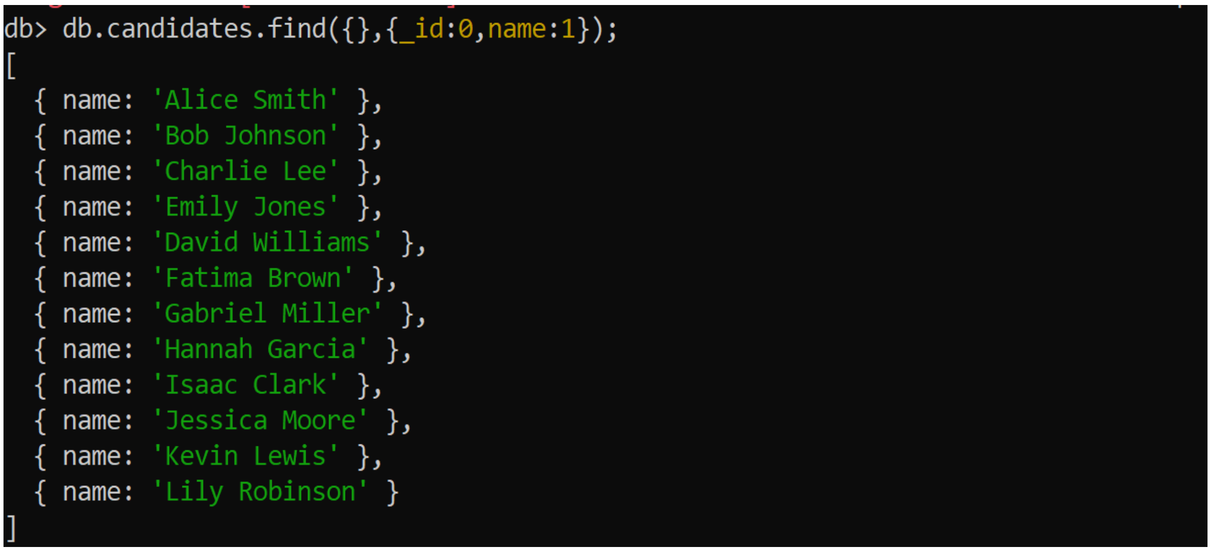
db.candidates.find({},{_id:0,courses:0});



```
db> db.candidates.find({},{_id:0,courses:0});
[
  {
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    name: 'Charlie Lee',
    age: 19,
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
```

here to retrieve including only name of candidates excluding _id we use

db.candidates.find({},{_id:0,name:1});

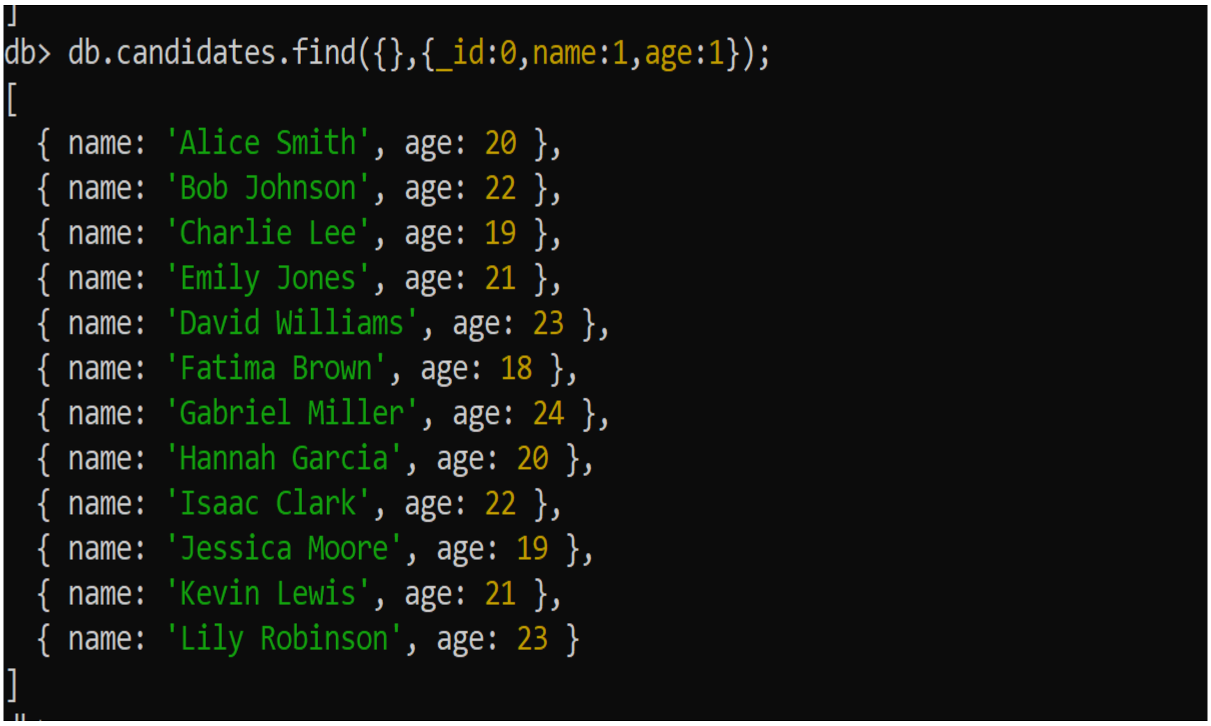


```
db> db.candidates.find({},{_id:0,name:1});
[
  { name: 'Alice Smith' },
  { name: 'Bob Johnson' },
  { name: 'Charlie Lee' },
  { name: 'Emily Jones' },
  { name: 'David Williams' },
  { name: 'Fatima Brown' },
  { name: 'Gabriel Miller' },
  { name: 'Hannah Garcia' },
  { name: 'Isaac Clark' },
  { name: 'Jessica Moore' },
  { name: 'Kevin Lewis' },
  { name: 'Lily Robinson' }
]
```

<u>2.Include fields(1):</u>

To include age and name of candidates without _id we use

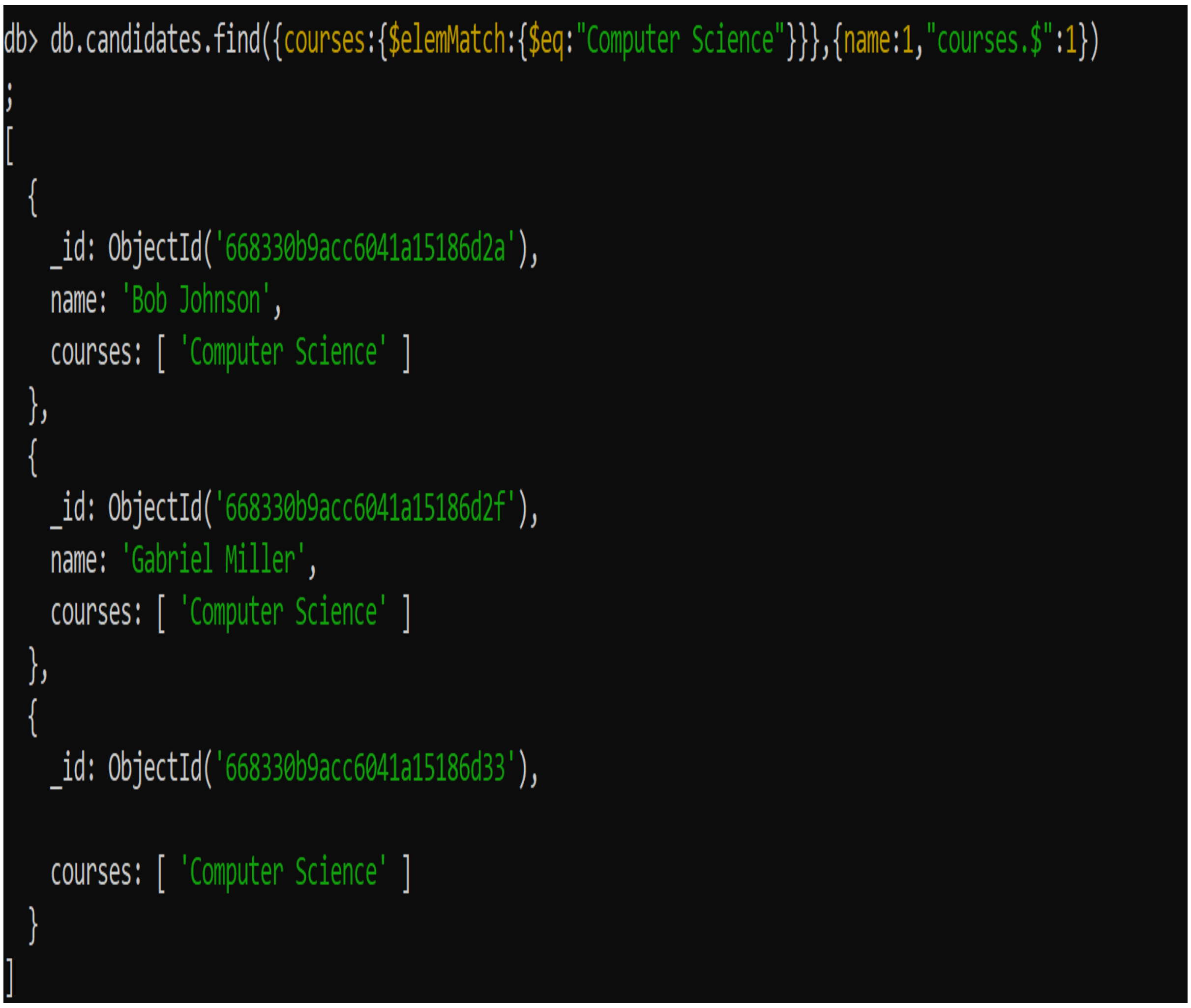
db.students.find({},{_id:0,name:1,age:1});



```
db> db.candidates.find({},{_id:0,name:1,age:1});
[
  { name: 'Alice Smith', age: 20 },
  { name: 'Bob Johnson', age: 22 },
  { name: 'Charlie Lee', age: 19 },
  { name: 'Emily Jones', age: 21 },
  { name: 'David Williams', age: 23 },
  { name: 'Fatima Brown', age: 18 },
  { name: 'Gabriel Miller', age: 24 },
  { name: 'Hannah Garcia', age: 20 },
  { name: 'Isaac Clark', age: 22 },
  { name: 'Jessica Moore', age: 19 },
  { name: 'Kevin Lewis', age: 21 },
  { name: 'Lily Robinson', age: 23 }
]
```

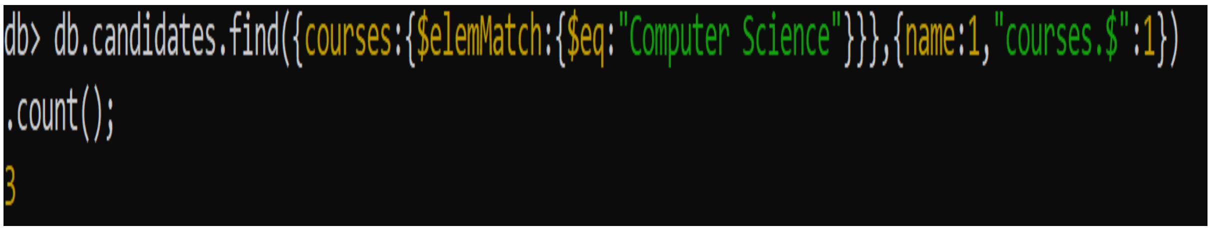
3.Elem operator($elemMatch):

To include matched fields ,here to find candidates who are enrolled in "Computer Science"with specific projection we use command **db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1});**



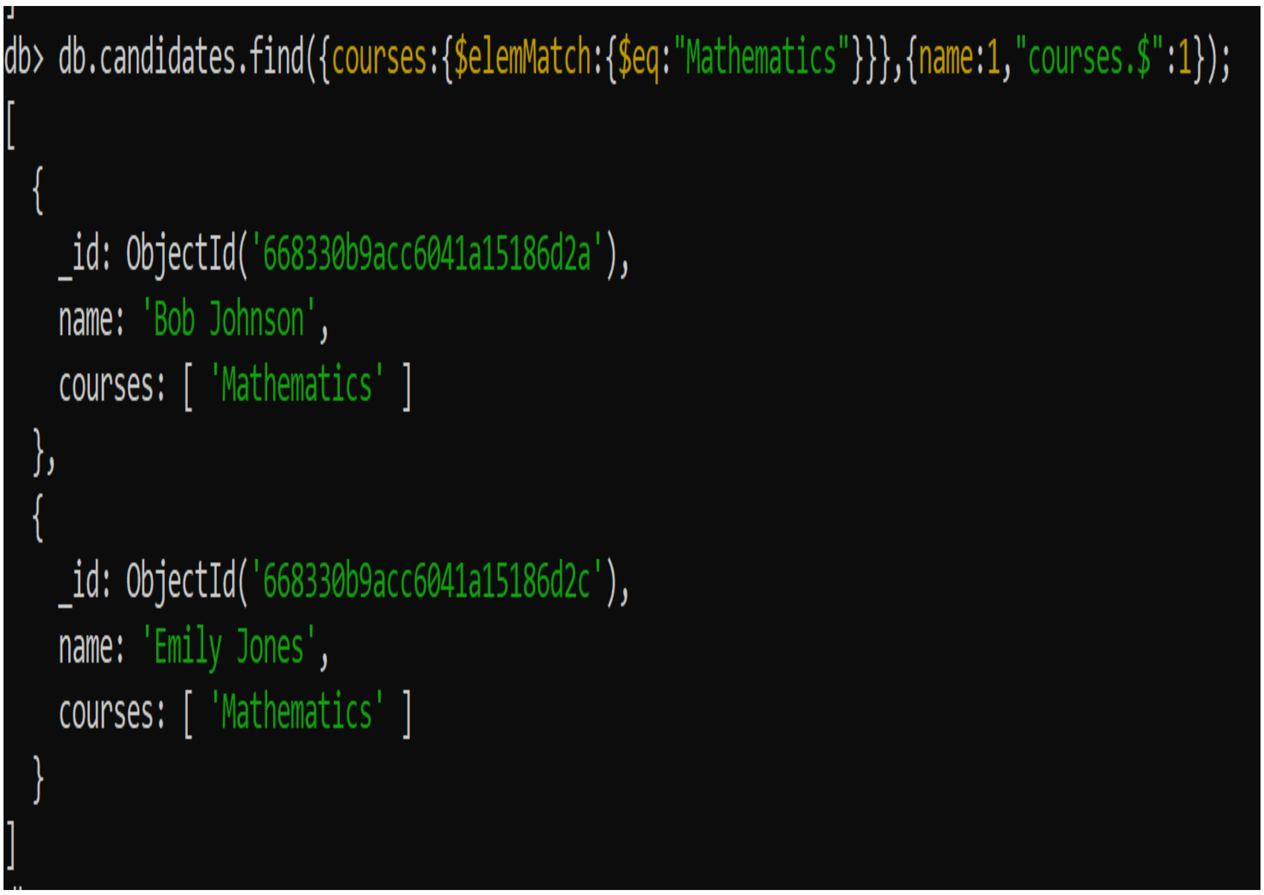
```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1})
;
[
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2f'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d33'),

    courses: [ 'Computer Science' ]
  }
]
```

To find count candidates who are enrolled in "Computer Science"with specific projection we use **db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1}).count()**



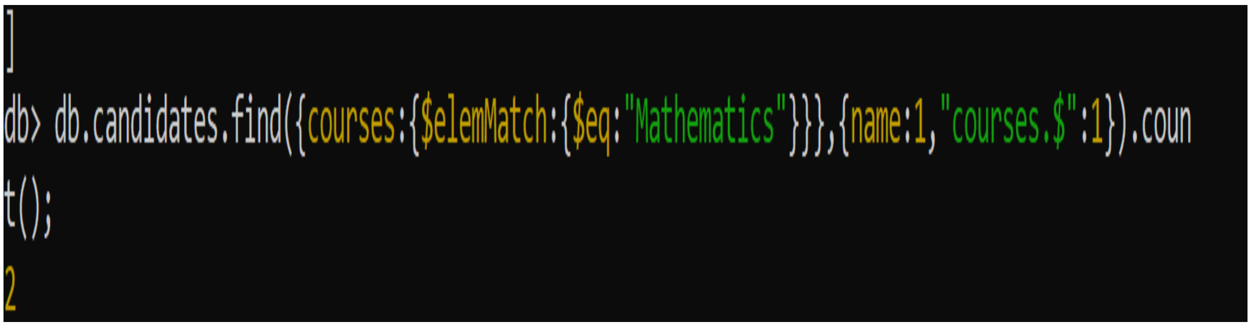
```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1})
.count();
3
```

Here to find candidates who are enrolled in “Mathematics”with specific projection we use
db.candidates.find({courses:{$elemMatch:{$eq:”Mathematics”}}},{name:1,”courses.$”:1});



```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Mathematics"}}},{name:1,"courses.$":1});
[
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    courses: [ 'Mathematics' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2c'),
    name: 'Emily Jones',
    courses: [ 'Mathematics' ]
  }
]
```

To find count of candidates who are enrolled in “Mathematics”with specific projection we use
db.candidates.find({courses:{$elemMatch:{$eq:”Mathematics”}}},{name:1,”courses.$”:1}).count();

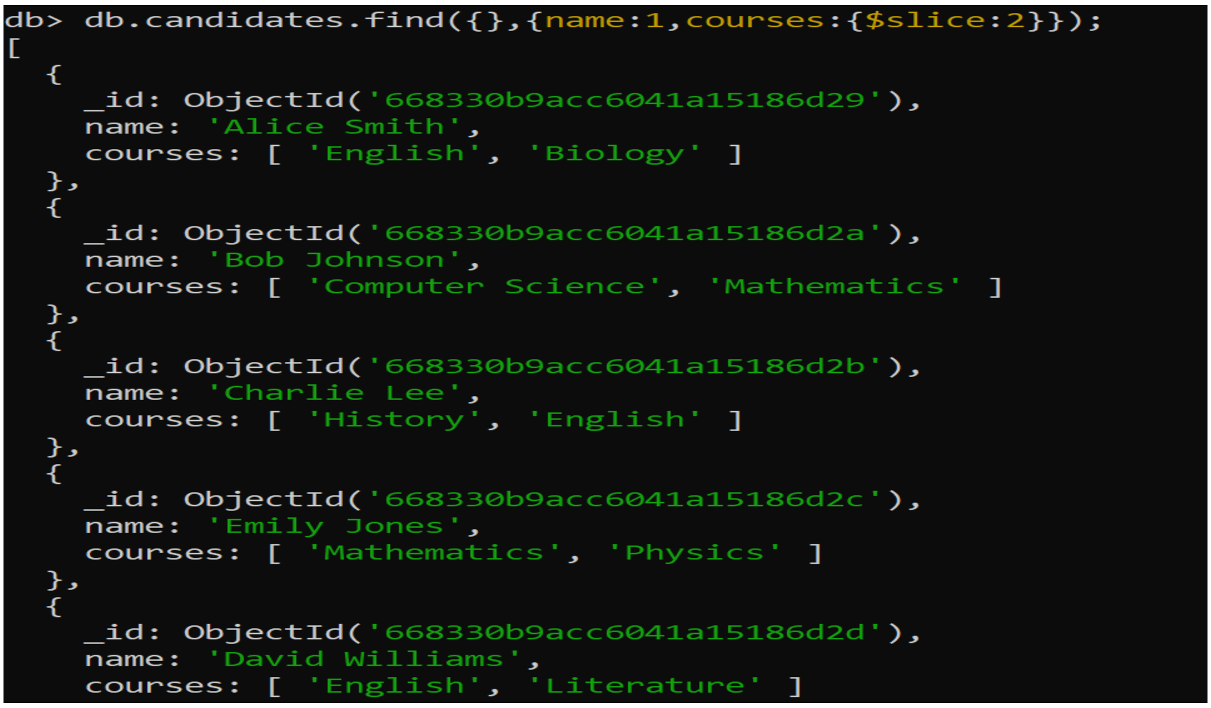


```
]
db> db.candidates.find({courses:{$elemMatch:{$eq:"Mathematics"}}},{name:1,"courses.$":1}).coun
t();
2
```

4.Slice operator($slice):

To retrieve all candidates with first two courses with name we use

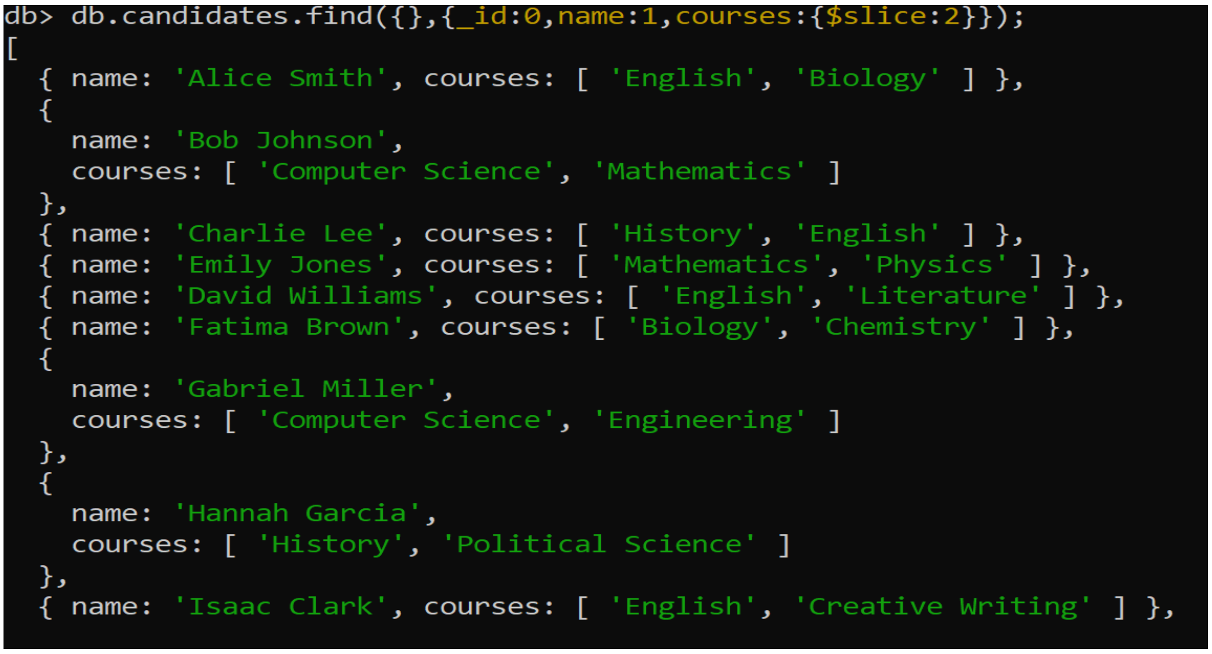
db.candidates.find({},{name:1,courses:{$slice:2}});



```
db> db.candidates.find({},{name:1,courses:{$slice:2}});
[
  {
    _id: ObjectId('668330b9acc6041a15186d29'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2b'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2c'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2d'),
    name: 'David Williams',
    courses: [ 'English', 'Literature' ]
```

One more without an _id to retrieve all candidates with first two courses we use

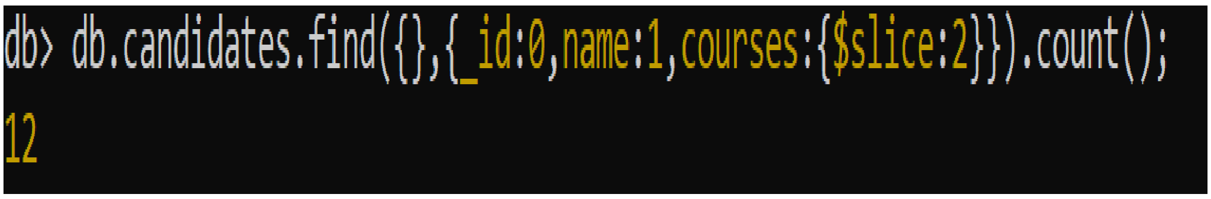
db.candidates.find({},_id:0,name:1,courses:{$slice:2}});



```
db> db.candidates.find({},{_id:0,name:1,courses:{$slice:2}});
[
  { name: 'Alice Smith', courses: [ 'English', 'Biology' ] },
  {
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  { name: 'Charlie Lee', courses: [ 'History', 'English' ] },
  { name: 'Emily Jones', courses: [ 'Mathematics', 'Physics' ] },
  { name: 'David Williams', courses: [ 'English', 'Literature' ] },
  { name: 'Fatima Brown', courses: [ 'Biology', 'Chemistry' ] },
  {
    name: 'Gabriel Miller',
    courses: [ 'Computer Science', 'Engineering' ]
  },
  {
    name: 'Hannah Garcia',
    courses: [ 'History', 'Political Science' ]
  },
  { name: 'Isaac Clark', courses: [ 'English', 'Creative Writing' ] },
```

To find count of candidates we use

db.candidates.find({},_id:0,name:1,courses:{$slice:2}});



```
db> db.candidates.find({},{_id:0,name:1,courses:{$slice:2}}).count();
12
```