CLASS 5: PROJECTION OPERATION

PROJECTION OPERATION:

In MongoDB, projection refers to the operation of selecting only the necessary data rather than retrieving the whole set of data of a document. When querying MongoDB, you can specify a projection to return only the necessary fields.

Using the $ Projection OperatorThe positional `$` operator is used in the projection document of the `find()` or `findOne()` method to limit the contents of an array to return the first element that matches the query condition on the array. This operator is particularly useful when you have an array of elements and you only need to retrieve the first element that matches your query criteria.

UNDERSTAND PROJECTION:

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

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
```

RETRIEVE NAME , AGE,AND GPA

```
mongosh mongodb://127.0.0.                                                          –  □  X

db> db.students.find({},{name:1,age:1,gpa:1});
[
  {
    _id: ObjectId('669a78a306e8da23cd90278f'),
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4
  },
  {
    _id: ObjectId('669a78a306e8da23cd902790'),
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8
  },
  {
    _id: ObjectId('669a78a306e8da23cd902791'),
    name: 'Charlie Lee',
    age: 19,
    gpa: 3.2
  },
  {
    _id: ObjectId('669a78a306e8da23cd902792'),
    name: 'Emily Jones',
    age: 21,
    gpa: 3.6
  },
  {
    _id: ObjectId('669a78a306e8da23cd902793'),
    name: 'David Williams',
    age: 23,
    gpa: 3
  },
  {
    _id: ObjectId('669a78a306e8da23cd902794'),
    name: 'Fatima Brown',
    age: 18,
    gpa: 3.5
```

In this MongoDB query, 'db.candidates.find({}.(name:l,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.

EXCLUDE FIELDS:

```
db> db.students.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+',
    is_hotel_resident: true
  },
  {
    name: 'Emily Jones',
    age: 21,
    gpa: 3.6,
    home_city: 'Houston',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
```
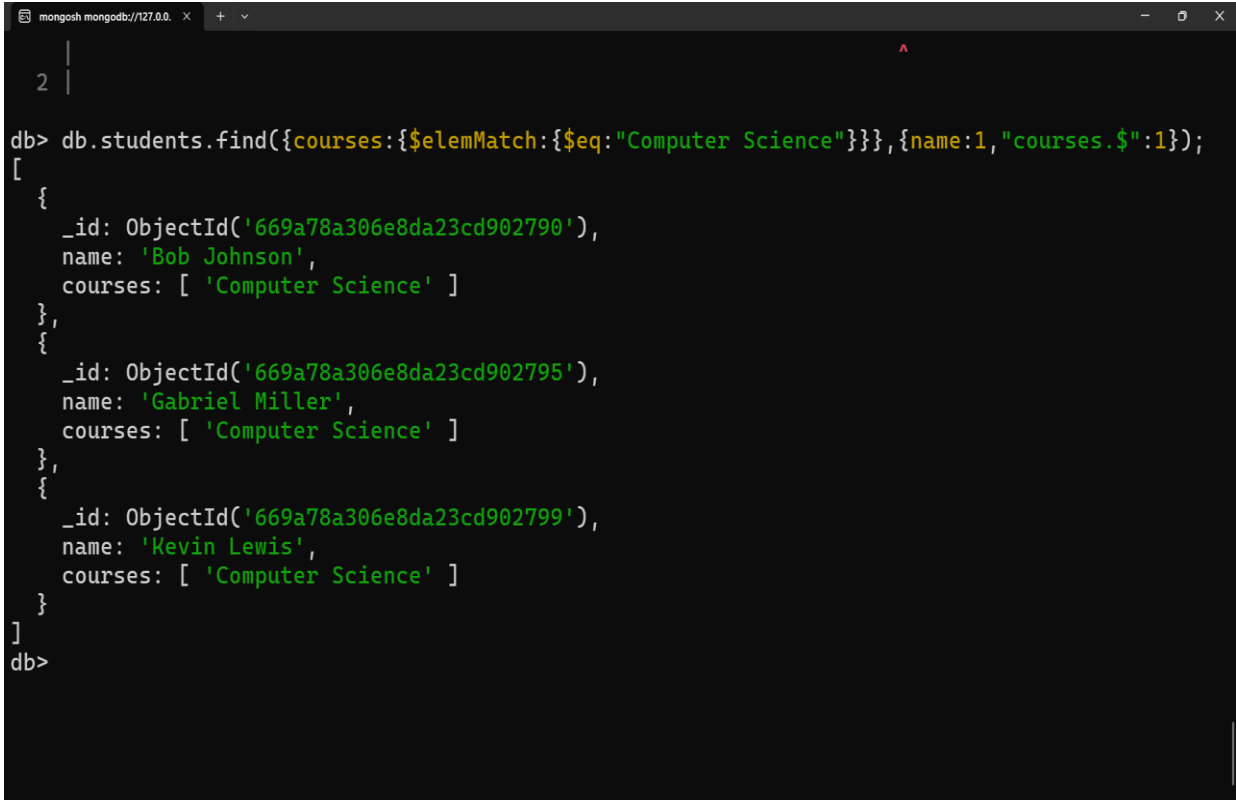
In this MongoDB query, 'db.candidates. find({},{_id:0,courses:O})', the 'find' method retrieves documents from the 'candidates' collection, and the projection part specifies the exclusion of the id' and 'courses' fields. The first argument, '{}', is an empty query filter, meaning all documents in the collection are selected. The second argument, '{_id:0, courses:0}, indicates that the ' id' and 'courses' fields should be excluded from the results. The resulting documents include fields such as 'name', 'age', 'gpa', 'home_city', 'blood_group', and 'is_hotel_resident, providing detailed information about each candidate without displaying their '_id' and 'courses' fields. This allow, for a more focused view of the relevant data, which can be useful for analysis or reporting purposes. The output includes candidates like Alice Smith, Bob Johnson, and Charlie Lee, showing their personal details and academic performance while omitting their unique identifiers and enrolled courses.

PROJECTION OPERATOR($elemMatch):

'$elemMatch' is a query operator in MongoDB that selects documents containing an array field with at least one element that matches all the specified query criteria.

Find candidates Enrolled in "Computer Science" with specific projection:

```
db> db.students.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1});
[
  {
    _id: ObjectId('669a78a306e8da23cd902790'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902795'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902799'),
    name: 'Kevin Lewis',
    courses: [ 'Computer Science' ]
  }
]
db>
```

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.

In the output, each document contains the candidate's "_id' (included by default), 'name', and an array with just the "Computer Science" course. This focused output is useful for identifying which candidates are enrolled in "Computer Science" while excluding other courses they might be taking. For example, the result includes candidates like Bob Johnson, Gabriel Miller, and Kevin Lewis, each showing "Computer Science" as the matched course. This query helps in pinpointing specific course enrollments efficiently within the dataset.

$elemMatch:

New player dataset.

```
db> db.players.find({},{games:{$elemMatch:{score:{$gt:5}}},joined:1,lastLogin:1})
[
  {
    _id: ObjectId('60bf1ad4366e071b0405a8f8'),
    joined: '2020-01-01',
    lastLogin: '2024-06-07',
    games: [ { game: 'game2', score: 6 } ]
  },
  {
    _id: ObjectId('60bf1ad4366e071b0405a8f9'),
    joined: '2021-02-15',
    lastLogin: '2024-06-06',
    games: [ { game: 'game2', score: 8 } ]
  }
]
db>
```

This MongoDB query searches the 'players' collection for documents where the 'games' array contains at least one subdocument where the 'score' is greater than 5. For each matching document, it projects the 'joined' and 'lastLogin' fields along with the 'games' array containing only the subdocument that matches the condition (score' greater than 5).

PROJECTION OPERATOR($slice):

'$slice' is a projection operator in MongoDB that limits the number of elements returned an array field projection.

Retrieve all candidates with first two courses:

```
]
db> db.students.find({},{name:1,courses:{$slice:2}});
[
  {
    _id: ObjectId('669a78a306e8da23cd90278f'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902790'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902791'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902792'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902793'),
    name: 'David Williams',
    courses: [ 'English', 'Literature' ]
  },
  {
    _id: ObjectId('669a78a306e8da23cd902794'),
```

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.