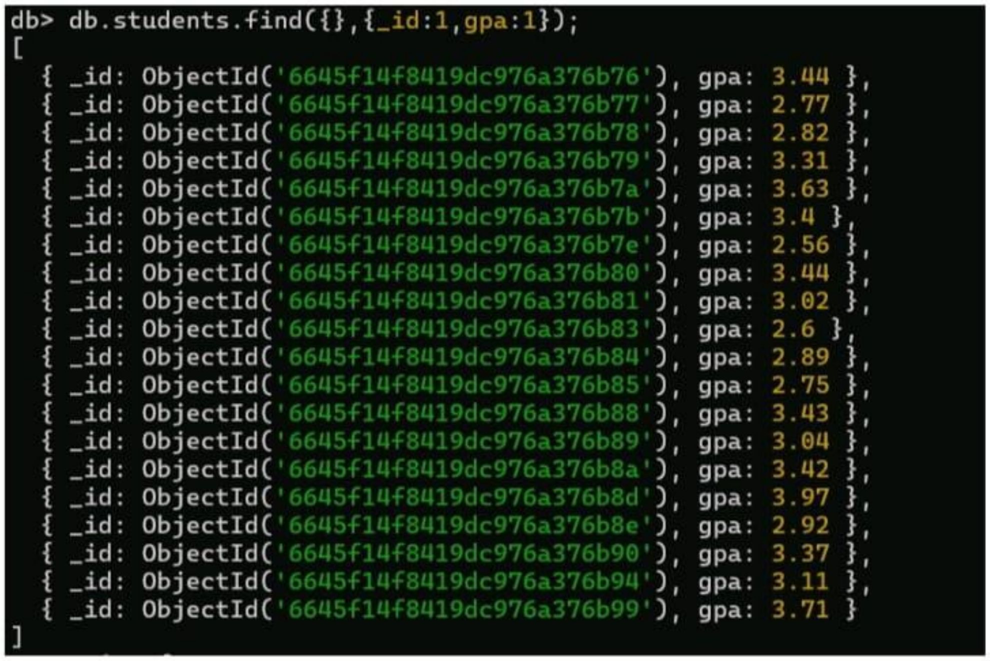
PROJECTION:

• MongoDB Projection is a special feature allowing you to select only the necessary data rather than selecting the whole set of data from the document.

EXAMPLE : db.students.find({}, {_id: 1, gpa: 1});



```
db> db.students.find({},{_id:1,gpa:1});
[
  { _id: ObjectId('6645f14f8419dc976a376b76'), gpa: 3.44 },
  { _id: ObjectId('6645f14f8419dc976a376b77'), gpa: 2.77 },
  { _id: ObjectId('6645f14f8419dc976a376b78'), gpa: 2.82 },
  { _id: ObjectId('6645f14f8419dc976a376b79'), gpa: 3.31 },
  { _id: ObjectId('6645f14f8419dc976a376b7a'), gpa: 3.63 },
  { _id: ObjectId('6645f14f8419dc976a376b7b'), gpa: 3.4 },
  { _id: ObjectId('6645f14f8419dc976a376b7e'), gpa: 2.56 },
  { _id: ObjectId('6645f14f8419dc976a376b80'), gpa: 3.44 },
  { _id: ObjectId('6645f14f8419dc976a376b81'), gpa: 3.02 },
  { _id: ObjectId('6645f14f8419dc976a376b83'), gpa: 2.6 },
  { _id: ObjectId('6645f14f8419dc976a376b84'), gpa: 2.89 },
  { _id: ObjectId('6645f14f8419dc976a376b85'), gpa: 2.75 },
  { _id: ObjectId('6645f14f8419dc976a376b88'), gpa: 3.43 },
  { _id: ObjectId('6645f14f8419dc976a376b89'), gpa: 3.04 },
  { _id: ObjectId('6645f14f8419dc976a376b8a'), gpa: 3.42 },
  { _id: ObjectId('6645f14f8419dc976a376b8d'), gpa: 3.97 },
  { _id: ObjectId('6645f14f8419dc976a376b8e'), gpa: 2.92 },
  { _id: ObjectId('6645f14f8419dc976a376b90'), gpa: 3.37 },
  { _id: ObjectId('6645f14f8419dc976a376b94'), gpa: 3.11 },
  { _id: ObjectId('6645f14f8419dc976a376b99'), gpa: 3.71 }
]
```

BENEFITS OF PROJECTION:

1. Data Reduction: By specifying which fields to include or exclude, you can reduce the amount of data sent to applications. This optimization is crucial for bandwidth efficiency.

2. Performance Optimization: Limiting the fields returned can improve query performance. Instead of retrieving all data, you only fetch what's necessary.

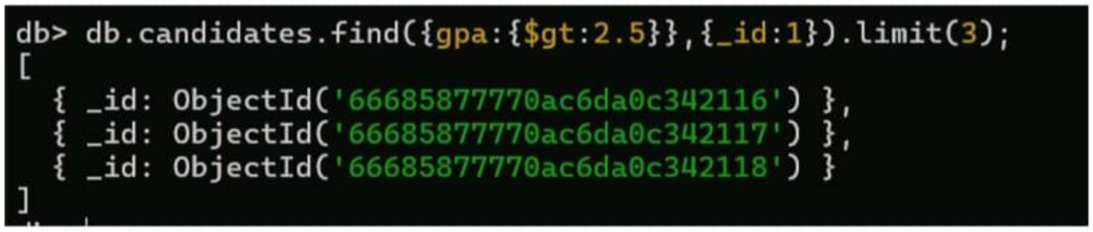
3. User-Centric Data: Projection helps extract specific fields relevant to user needs from large datasets12 .

For example, you can select only the name and age of employees rather than displaying all details.

LIMIT:

• We can use this method after the find() method and find() will give you all the records or documents in the collection. You can also use some conditions inside the find to give you the result that you want.

EXAMPLE : db.candidates.find({gpa: {$gt:2.5}} , {_id:1}).limit(3);

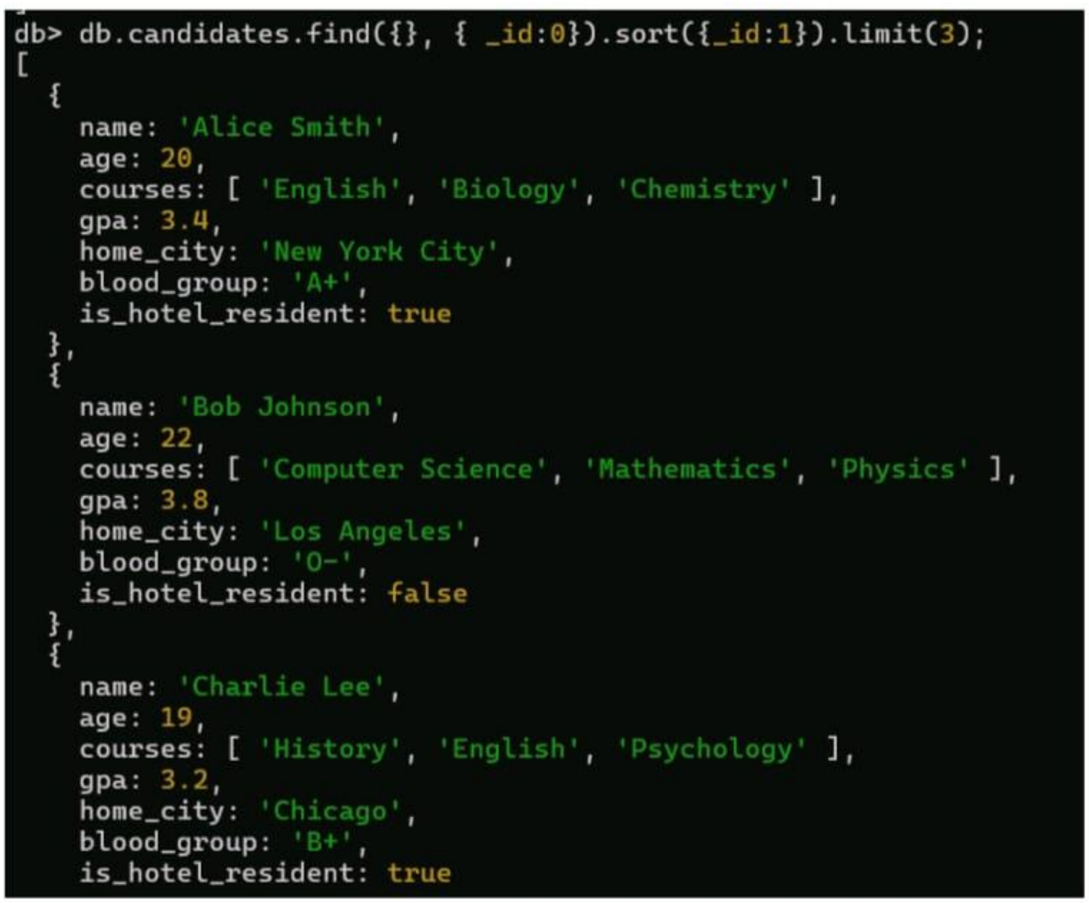


```
db> db.candidates.find({gpa:{$gt:2.5}},{_id:1}).limit(3);
[
  { _id: ObjectId('66685877770ac6da0c342116') },
  { _id: ObjectId('66685877770ac6da0c342117') },
  { _id: ObjectId('66685877770ac6da0c342118') }
]
```

Use of sort():

• We can use the limit() method with the sort() method, it will return the first m documents, where m is the given limit.

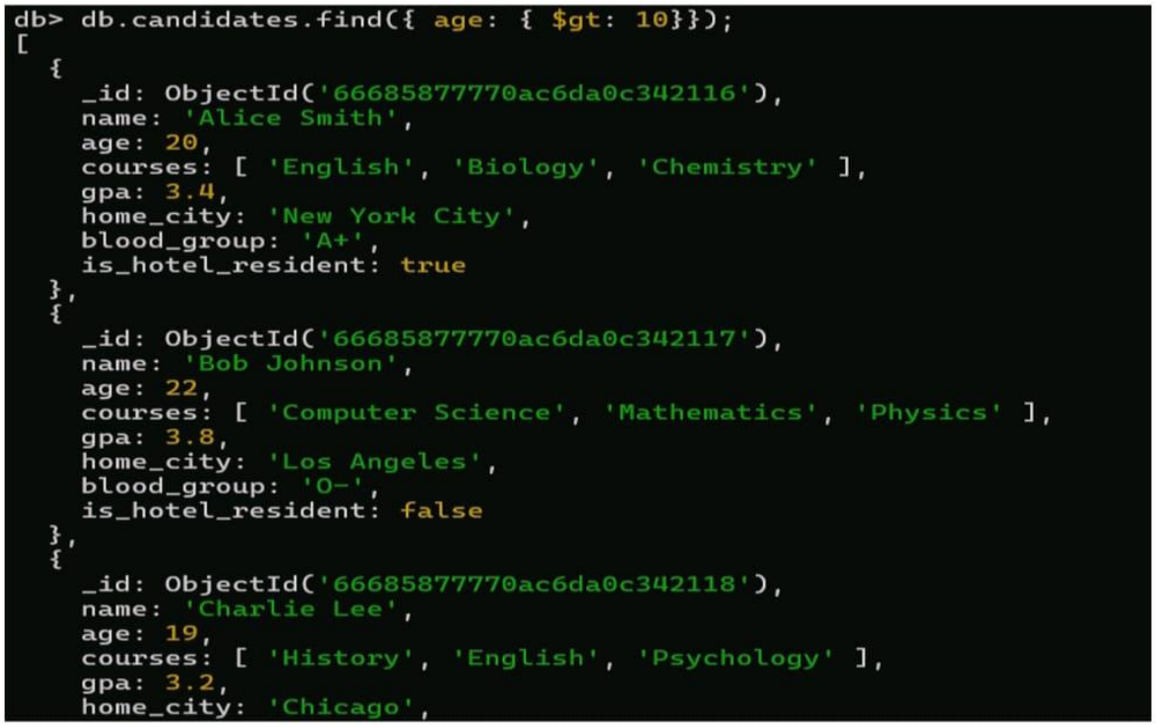
EXAMPLE : db.students.find({}, { _id:1}).sort({_id:-1}).limit(4);



```
db> db.candidates.find({}, { _id:0}).sort({_id:1}).limit(3);
[
  {
    name: 'Alice Smith',
    age: 20,
    courses: [ 'English', 'Biology', 'Chemistry' ],
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Computer Science', 'Mathematics', 'Physics' ],
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'History', 'English', 'Psychology' ],
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
```

SELECTORS:

db.candidates.find({ age: { $gt: 10}});



```
db> db.candidates.find({ age: { $gt: 10}});
[
  {
    _id: ObjectId('66685877770ac6da0c342116'),
    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('66685877770ac6da0c342117'),
    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('66685877770ac6da0c342118'),
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'History', 'English', 'Psychology' ],
    gpa: 3.2,
    home_city: 'Chicago',
```

SELECTORS:

Grater than: The $gt operator in MongoDB is used to specify a query condition where the field value must be greater than a specified value. It stands for "greater than."

Lesser than: the less than operation is performed using the $lt operator. This operator allows you to query documents where a specific field's value is less than a given value.