EXPERIMENT 2

Projection & Limit

a. Develop a MongoDB query to select certain fields and ignore some fields of the documents from any collection.

b. Develop a MongoDB query to display the first 5 documents from the results obtained in a.

PROJECTION:

In MongoDB, projection is the process of specifying which fields of a document should be returned in the query result. It allows you to retrieve only the necessary data, reducing network traffic and improving performance. By default, queries return all fields in matching documents, but projection allows you to limit the fields returned to only those you specify.

```
db> db.students.find({},{name:1,gpa:1,_id:0});
[
  { name: 'Student 948', gpa: 3.44 },
  { name: 'Student 157', gpa: 2.77 },
  { name: 'Student 316', gpa: 2.82 },
  { name: 'Student 346', gpa: 3.31 },
  { name: 'Student 930', gpa: 3.63 },
  { name: 'Student 305', gpa: 3.4 },
  { name: 'Student 440', gpa: 2.56 },
  { name: 'Student 256', gpa: 3.44 },
  { name: 'Student 177', gpa: 3.02 },
  { name: 'Student 487', gpa: 2.6 },
  { name: 'Student 213', gpa: 2.89 },
  { name: 'Student 690', gpa: 2.75 },
  { name: 'Student 647', gpa: 3.43 },
  { name: 'Student 232', gpa: 3.04 },
  { name: 'Student 328', gpa: 3.42 },
  { name: 'Student 468', gpa: 3.97 },
  { name: 'Student 504', gpa: 2.92 },
  { name: 'Student 915', gpa: 3.37 },
  { name: 'Student 367', gpa: 3.11 },
  { name: 'Student 969', gpa: 3.71 }
]
```

The MongoDB query `db.students.find({}, { name: 1, gpa: 1, _id: 0 })` retrieves documents from the `students` collection but includes only the `name` and `gpa` fields in the result. The `_id` field, which is included in all documents by default, is excluded (`_id: 0`) from the result. This projection results in a list of objects where each object contains only the `name` and `gpa` fields for each student, without the `_id` field.

FIELD PROJECTION:

Use the projection document as the second argument to the find method.Include field names with a value of 1 to specify fields to be returned.Omit fields or set them to 0 to exclude them from the results.

BENEFITS:

Reduces data transferred between the database and your application.Improves query performance by retrieving only necessary data.Simplifies your code by focusing on the specific information you need.

LIMIT:

The limit operator is used with the find method.It's chained after the filter criteria or any sorting operations.

Syntax: db.collection.find({filter}, {projection}).limit(number)

`name` and `gpa` fields for each student, without the `_id` field.

Getting first 5 elements:

```
db> db.students.find({},{_id:0}).limit(5);
[
  {
    name: 'Student 948',
    age: 19,
    courses: "['English', 'Computer Science', 'Physics', 'Mathematics']",
    gpa: 3.44,
    home_city: 'City 2',
    blood_group: 'O+',
    is_hotel_resident: true
  },
  {
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.77,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    name: 'Student 316',
    age: 20,
    courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
    gpa: 2.82,
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    name: 'Student 346',
    age: 25,
    courses: "['Mathematics', 'History', 'English']",
    gpa: 3.31,
    home_city: 'City 8',
    blood_group: 'O-',
    is_hotel_resident: true
  },
```

The query `db.students.find({},{_id:0}).limit(5);` is used to retrieve documents from the `students` collection in MongoDB. The query returns up to five student documents but excludes the `_id` field in the results. The output includes details such as the student's name, age, courses, GPA, home city, blood group, and whether they are a hotel resident. This demonstrates the use of the `find` method with a projection to exclude specific fields and the `limit` method to restrict the number of documents returned.

LIMITING RESULTS:

```
db>  db.students.find({gpa:{$gt:3.5}},{_id:0}).limit(2);
[
  {
    name: 'Student 930',
    age: 25,
    courses: "['English', 'Computer Science', 'Mathematics', 'History']",
    gpa: 3.63,
    home_city: 'City 3',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    name: 'Student 468',
    age: 21,
    courses: "['Computer Science', 'Physics', 'Mathematics', 'History']",
    gpa: 3.97,
    blood_group: 'A-',
    is_hotel_resident: true
  }
]
db>
```

query `db.students.find({gpa:{$gt:3.5}},{_id:0}).limit(2);` retrieves documents from the `students` collection in MongoDB where the `gpa` field is greater than 3.5. It excludes the `_id` field from the results and limits the output to two documents. The resulting documents include details such as the student's name, age, courses, GPA, home city, blood group, and whether they are a hotel resident, showcasing the use of the `find` method with filtering criteria, projection, and limit.

TO GET TOP 10 RESULTS:

```
db>  db.students.find({},{_id:0}).sort({_id:-1}).limit(5);
[
  {
    name: 'Student 933',
    age: 18,
    courses: "['Mathematics', 'English', 'Physics', 'History']",
    gpa: 3.04,
    home_city: 'City 10',
    blood_group: 'B-',
    is_hotel_resident: true
  },
  {
    name: 'Student 831',
    age: 20,
    courses: "['Mathematics', 'Computer Science']",
    gpa: 3.49,
    home_city: 'City 3',
    blood_group: 'AB+',
    is_hotel_resident: true
  },
  {
    name: 'Student 143',
    age: 21,
    courses: "['Mathematics', 'Computer Science', 'English', 'History']",
    gpa: 2.78,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    name: 'Student 718',
    age: 21,
    courses: "['Computer Science', 'English']",
    gpa: 2.75,
    home_city: 'City 5',
    blood_group: 'O-',
    is_hotel_resident: true
  },
```

The query `db.students.find({},{_id:0}).sort({_id:-1}).limit(5);` retrieves documents from the `students` collection, excluding the `_id` field from the results, and sorts them in descending order by the `_id` field. The `sort({_id:-1})` part ensures that the documents are ordered from the most recent to the oldest based on their `_id` values. The `limit(5)` clause restricts the output to the first five documents that meet these criteria. The resulting documents include various student details such as name, age, courses, GPA, home city, blood group, and whether they are hotel residents, demonstrating how sorting and limiting can be applied to refine query results.