

Github Link: (<https://github.com/soukhyakj/MongoDB/tree/main>)

INTRODUCTION:

MongoDB is a popular, open-source, NoSQL database that allows you to store and manage large amounts of data in a flexible and scalable way. It is designed to handle the demands of modern web and mobile applications, and is widely used in industries such as finance, healthcare, and e-commerce.

Experiment 1

a. Illustration of Where Clause, AND, OR operations in MongoDB:

1.Where clause:

The most common use of the **WHERE** clause is to filter data based on a condition.

Input:

// find all students with GPA lesser than 3

```
db.stu.find({gpa:{<3}});
```

```
db.stu.find({gpa:{<3}});
```

Output:

```
{
  "_id": ObjectId("66af17231d6eb7f7f70165cb"),
  "name": "Student 157",
  "age": 20,
  "courses": ["Physics", "English"],
  "gpa": 2.27,
  "home_city": "City 4",
  "blood_group": "O-",
  "is_hotel_resident": true
},
{
  "_id": ObjectId("66af17231d6eb7f7f70165cc"),
  "name": "Student 316",
  "age": 20,
  "courses": ["Physics", "Computer Science", "Mathematics", "History"],
  "gpa": 2.32,
  "blood_group": "B+",
  "is_hotel_resident": true
},
{
  "_id": ObjectId("66af17231d6eb7f7f70165d1"),
  "name": "Student 563",
  "age": 18,
  "courses": ["Mathematics", "English"],
  "gpa": 2.25,
  "blood_group": "AB+",
  "is_hotel_resident": false
},
{
  "_id": ObjectId("66af17231d6eb7f7f70165d2"),
  "name": "Student 440",
  "age": 21,
  "courses": ["History", "Physics", "Computer Science"],
  "gpa": 2.1,
  "blood_group": "A+",
  "is_hotel_resident": false
}
```

2.AND clause:

\$and clause is used to combine multiple conditions in a query and filter documents based on multiple criteria.

Input:

// find all students who live in “City 5” AND have blood group “A+”

```
db> db.students.find({
... $and:[
... { home_city:"City 5"},
... { blood_group:"A+"}
... ]
... });
```

Output:

```
[
  {
    _id: ObjectId('66a215ba046ad7f15a24ca84'),
    name: 'Student 142',
    age: 24,
    courses: ["'History'", "'English'", "'Physics'", "'Computer Science'"],
    gpa: 3.41,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('66a215ba046ad7f15a24cba4'),
    name: 'Student 947',
    age: 20,
    courses: ["'Physics'", "'History'", "'English'", "'Computer Science'"],
    gpa: 2.86,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('66a215ba046ad7f15a24cc16'),
    name: 'Student 567',
    age: 22,
    courses: ["'Computer Science'", "'History'", "'English'", "'Mathematics'"],
    gpa: 2.01,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: true
  }
]
```

3. OR clause:

The **\$or** clause is used to filter documents that match at least one of the conditions specified in the clause.

Input:

//find all students who are hotel resident OR have GPA less than 3.5.

```
db.students.find({ $or:[ {is_hotel_resident: true}, {gpa :{$lt: 3.5}} ]});
```

Output:

```

{
  _id: ObjectId('66af17231d6eb7f7f70165ca'),
  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
},
{
  _id: ObjectId('66af17231d6eb7f7f70165cb'),
  name: 'Student 157',
  age: 20,
  courses: "['Physics', 'English']",
  gpa: 2.27,
  home_city: 'City 4',
  blood_group: 'O-',
  is_hotel_resident: true
},
{
  _id: ObjectId('66af17231d6eb7f7f70165cc'),
  name: 'Student 316',
  age: 20,
  courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
  gpa: 2.32,
  blood_group: 'B+',
  is_hotel_resident: true
},
{
  _id: ObjectId('66af17231d6eb7f7f70165cd'),
  name: 'Student 346',
  age: 25,

```

- b. Execute the Commands of MongoDB and operations in MongoDB: Insert, Query, Update, Delete and Projection.

1. Insert document

Input:

```

db> const studentData={
...   "name": "Alice Smith",
...   "age": 22,
...   "courses": ["Mathematics", "Computer Science", "English"],
...   "gpa": 3.8,
...   "home_city": "New York",
...   "blood_group": "A+",
...   "is_hotel_resident": false
... };
db> db.students.insertOne(studentData);

```

Output:

```

{
  acknowledged: true,
  insertedId: ObjectId('6661da38b0d232162dcdcdf6')
}
db>

```

2. Query document

Input:

```

db> db.students.find({age:{$gt:23}});

```

Output:

```
{
  _id: ObjectId('66a215ba046ad7f15a24ca50'),
  name: 'Student 346',
  age: 25,
  courses: "['Mathematics', 'History', 'English']",
  gpa: 3.31,
  home_city: 'City 8',
  blood_group: 'O-',
  is_hotel_resident: true
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca51'),
  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
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca52'),
  name: 'Student 305',
  age: 24,
  courses: "['History', 'Physics', 'Computer Science', 'Mathematics']",
  gpa: 3.4,
  home_city: 'City 6',
  blood_group: 'O+',
  is_hotel_resident: true
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca5e'),
  name: 'Student 172',
  age: 25,
  courses: "['English', 'History', 'Physics', 'Mathematics']",
  gpa: 2.46,
  home_city: 'City 3',
  blood_group: 'A+',
  is_hotel_resident: false
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca62'),
  name: 'Student 690',
  age: 24,
  courses: "['Computer Science', 'English', 'History']",
  gpa: 2.71,
```

3. Update document**Input:**

```
db> db.students.updateOne({name:"Alice Smith"},{$set:{gpa:3.8}});
```

Output:

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
```

4.Delete document**Input:**

```
db> db.students.deleteOne({name:"John Doe"});
{ acknowledged: true, deletedCount: 0 }
```

Output:

```
{ acknowledged: true, deletedCount: 0 }
```

5.Projection

Input:

```
db> db.students.find({}, {name:1,gpa:1});
```

Output:

```
{
  _id: ObjectId('66a215ba046ad7f15a24ca4d'),
  name: 'Student 948',
  gpa: 3.44
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca4e'),
  name: 'Student 157',
  gpa: 2.27
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca4f'),
  name: 'Student 316',
  gpa: 2.32
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca50'),
  name: 'Student 346',
  gpa: 3.31
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca51'),
  name: 'Student 930',
  gpa: 3.63
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca52'),
  name: 'Student 305',
  gpa: 3.4
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca53'),
  name: 'Student 268',
  gpa: 3.98
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca54'),
  name: 'Student 563',
  gpa: 2.25
},
{
  _id: ObjectId('66a215ba046ad7f15a24ca55'),
  name: 'Student 440',
  gpa: 2.06
},
},
```

Experiment 2

- a. Develop a MongoDB query to select certain fields and ignore some fields of the documents:

1. Get only name & age for all students

Input:

```
db> db.students.find({}, {name:1, age:1});
```

Output:

```
{
  "_id": ObjectId("66a215ba046ad7f15a24ca4d"),
  "name": "Student 948",
  "age": 19
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca4e"),
  "name": "Student 157",
  "age": 20
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca4f"),
  "name": "Student 316",
  "age": 20
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca50"),
  "name": "Student 346",
  "age": 25
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca51"),
  "name": "Student 930",
  "age": 25
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca52"),
  "name": "Student 305",
  "age": 24
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca53"),
  "name": "Student 268",
  "age": 21
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca54"),
  "name": "Student 563",
  "age": 18
},
{
  "_id": ObjectId("66a215ba046ad7f15a24ca55"),
  "name": "Student 440",
  "age": 21
}
```

2. Select certain fields and ignore some fields

Input:

```
db> db.students.find({}, {name:1, gpa:1, _id:0});
```

Output:

```
[
  { 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 }
]
```

3. Ignoring attributes

input:

```
db> db.students.find({}, {_id:0});
```

Output:

```
[
  {
    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: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    name: 'Student 316',
    age: 20,
    courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
    gpa: 2.32,
    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
  },
  {
    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 305',
    age: 24,
```

- b. Develop a MongoDB query to display the first 5 documents from the results obtained in a. [use of limit and find]

Input:

// Display the first 5 documents

```
db> db.students.find({}, {_id:0}).limit(5);
```

Output:

```
[
  {
    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
  },
]
```

Experiment 3

a. Execute query selectors (comparison selectors, logical selectors) and list out the results on any collection

1.Comparison selectors

Input:

```
db> db.students.find({age:{$gt:20}});
```

Output:

```
[
  {
    _id: ObjectId('6649bb89b51b15a423b44ad0'),
    name: 'Student 346',
    age: 25,
    courses: "['Mathematics', 'History', 'English']",
    gpa: 3.31,
    home_city: 'City 8',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6649bb89b51b15a423b44ad1'),
    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
  },
  {
    _id: ObjectId('6649bb89b51b15a423b44ad2'),
    name: 'Student 305',
    age: 24,
    courses: "['History', 'Physics', 'Computer Science', 'Mathematics']",
    gpa: 3.4,
    home_city: 'City 6',
    blood_group: 'O+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6649bb89b51b15a423b44ad5'),
    name: 'Student 440',
    age: 21,
    courses: "['History', 'Physics', 'Computer Science']",
    gpa: 2.56,
    home_city: 'City 10',
    blood_group: 'O-',
    is_hotel_resident: true
  },
]
```

2. Logical selectors

Input:

```
db> db.students.find({
... $and:[
... {home_city:"City 2"},
... {blood_group:"B+"}
... ]
... });
```

Output:

```
[
  {
    _id: ObjectId('6649bb89b51b15a423b44ae5'),
    name: 'Student 504',
    age: 21,
    courses: "['Physics', 'Computer Science', 'English', 'Mathematics']",
    gpa: 2.92,
    home_city: 'City 2',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6649bb89b51b15a423b44c93'),
    name: 'Student 872',
    age: 24,
    courses: "['English', 'Mathematics', 'History']",
    gpa: 3.36,
    home_city: 'City 2',
    blood_group: 'B+',
    is_hotel_resident: true
  }
]
```

b. Execute query selectors (Geospatial selectors, Bitwise selectors) and list out the results on any collection

1. Geospatial selectors**Input:**

```
db> db.locations.find({
... location:{
... $geoWithin:{
... $centerSphere:[[-74.005,40.712],0.00621376]
... }
... }
... });
```

Output:

```
[
  {
    _id: 1,
    name: 'Coffee Shop A',
    location: { type: 'Point', coordinates: [ -73.985, 40.748 ] }
  },
  {
    _id: 2,
    name: 'Restaurant B',
    location: { type: 'Point', coordinates: [ -74.009, 40.712 ] }
  },
  {
    _id: 5,
    name: 'Park E',
    location: { type: 'Point', coordinates: [ -74.006, 40.705 ] }
  }
]
```

2. Bitwise selectors

Input:

```
db> const LOBBY_PERMISSION=1;
db> const CAMPUS_PERMISSION=2;
db> db.students_permission.find({
... permissions:{$bitsAllSet:[LOBBY_PERMISSION,CAMPUS_PERMISSION]}
... });
```

Output:

```
[
  {
    _id: ObjectId('66635182d29d811170a4e560'),
    name: 'George',
    age: 21,
    permissions: 6
  },
  {
    _id: ObjectId('66635182d29d811170a4e561'),
    name: 'Henry',
    age: 27,
    permissions: 7
  },
  {
    _id: ObjectId('66635182d29d811170a4e562'),
    name: 'Isla',
    age: 18,
    permissions: 6
  }
]
```

Experiment 4

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

1.\$ projection operator

Input:

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

Output:

```
[
  {
    _id: ObjectId('66a3861aa0bde03994b0ee75'),
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee76'),
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee77'),
    name: 'Charlie Lee',
    age: 19,
    gpa: 3.2
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee78'),
    name: 'Emily Jones',
    age: 21,
    gpa: 3.6
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee79'),
    name: 'David Williams',
    age: 23,
    gpa: 3
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee7a'),
    name: 'Fatima Brown',
    age: 18,
    gpa: 3.5
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee7b'),
    name: 'Gabriel Miller',
    age: 24,
    gpa: 3.9
  },
  {
    _id: ObjectId('66a3861aa0bde03994b0ee7c'),
    name: 'Hannah Garcia',

```

2.\$elemMatch projection operator

Input:

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

Output:

```
[
  { _id: ObjectId('66a3861aa0bde03994b0ee76'), name: 'Bob Johnson' },
  { _id: ObjectId('66a3861aa0bde03994b0ee7b'), name: 'Gabriel Miller' },
  { _id: ObjectId('66a3861aa0bde03994b0ee7f'), name: 'Kevin Lewis' }
]
```

3.\$slice

Input:

```
db> db.candidates.find({}, {courses:{$slice:1}})
[
  {

```

Output:

```
[
  {
    _id: ObjectId('6657ff95946a866dbb971e5f'),
    name: 'Alice Smith',
    age: 20,
    courses: [ 'English' ],
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e60'),
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Computer Science' ],
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e61'),
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'History' ],
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e62'),
    name: 'Emily Jones',
    age: 21,
    courses: [ 'Mathematics' ],
    gpa: 3.6,
    home_city: 'Houston',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e63'),
    name: 'David Williams',
    age: 23,
    courses: [ 'English' ],
    gpa: 3,
    home_city: 'Phoenix',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e64'),
    name: 'Fatima Brown',
    age: 18,
    courses: [ 'Biology' ],
    gpa: 3.5,
    home_city: 'San Antonio',
    blood_group: 'B+',
  }
]
```

Output:

```
db> db.candidates.find({}, {courses: {$slice: [1,3]}})
[
  {
    _id: ObjectId('6657ff95946a866dbb971e5f'),
    name: 'Alice Smith',
    age: 20,
    courses: [ 'Biology', 'Chemistry' ],
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e60'),
    name: 'Bob Johnson',
    age: 22,
    courses: [ 'Mathematics', 'Physics' ],
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e61'),
    name: 'Charlie Lee',
    age: 19,
    courses: [ 'English', 'Psychology' ],
    gpa: 3.2,
    home_city: 'Chicago',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e62'),
    name: 'Emily Jones',
    age: 21,
    courses: [ 'Physics', 'Statistics' ],
    gpa: 3.6,
    home_city: 'Houston',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e63'),
    name: 'David Williams',
    age: 23,
    courses: [ 'Literature', 'Philosophy' ],
    gpa: 3,
    home_city: 'Phoenix',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6657ff95946a866dbb971e64'),
    name: 'Fatima Brown',
    age: 18,
    courses: [ 'Chemistry', 'Environmental Science' ],
    gpa: 3.5,
    home_city: 'San Antonio',
    blood_group: 'B+',
    is_hotel_resident: false
  }
]
```

Experiment 5

Execute Aggregation operations (\$avg, \$min, \$max, \$push, \$addToSet etc.).

1. \$avg

Input:

```
db> db.students.aggregate([
...   {$group:
...   {_id:null,avgAge:{$avg:"$age"}}
... ]})
```

Output:

```
[ { _id: null, avgAge: 21.534 } ]
```

2.\$min

Input:

```
db> db.students.aggregate([ { $group: { _id: null, minAge: { $min: "$age" } } } ]])
```

Output:

```
[ { _id: null, minAge: 18 } ]
```

3.\$max

Input:

```
db> db.students.aggregate([ { $group: { _id: null, maxAge: { $max: "$age" } } } ]])
```

Output:

```
[ { _id: null, maxAge: 25 } ]
```

4.\$push

Input1:

```
db> db.students.aggregate([ { $group: { _id: null, allNames: { $push: "$name" } } } ]])
```

Output1:

```
[
  {
    _id: null,
    allNames: [
      'Student 948', 'Student 157', 'Student 316', 'Student 346', 'Student 930',
      'Student 305', 'Student 268', 'Student 563', 'Student 440', 'Student 536',
      'Student 256', 'Student 177', 'Student 871', 'Student 487', 'Student 213',
      'Student 690', 'Student 368', 'Student 172', 'Student 647', 'Student 232',
      'Student 328', 'Student 690', 'Student 499', 'Student 468', 'Student 504',
      'Student 652', 'Student 915', 'Student 127', 'Student 239', 'Student 770',
      'Student 367', 'Student 268', 'Student 384', 'Student 372', 'Student 771',
      'Student 969', 'Student 502', 'Student 504', 'Student 440', 'Student 201',
      'Student 814', 'Student 901', 'Student 676', 'Student 533', 'Student 347',
      'Student 404', 'Student 111', 'Student 873', 'Student 871', 'Student 425',
      'Student 610', 'Student 539', 'Student 165', 'Student 959', 'Student 860',
      'Student 142', 'Student 457', 'Student 990', 'Student 522', 'Student 789',
      'Student 895', 'Student 592', 'Student 219', 'Student 722', 'Student 159',
      'Student 833', 'Student 909', 'Student 632', 'Student 994', 'Student 515',
      'Student 746', 'Student 402', 'Student 193', 'Student 925', 'Student 833',
      'Student 301', 'Student 984', 'Student 378', 'Student 886', 'Student 367',
      'Student 781', 'Student 430', 'Student 887', 'Student 314', 'Student 517',
      'Student 357', 'Student 871', 'Student 508', 'Student 803', 'Student 989',
      'Student 291', 'Student 502', 'Student 220', 'Student 175', 'Student 857',
      'Student 229', 'Student 352', 'Student 212', 'Student 148', 'Student 558',
      'Student 880', 'Student 110', 'Student 908', 'Student 920', 'Student 880',
      'Student 543', 'Student 129', 'Student 262', 'Student 710', 'Student 422',
      'Student 821', 'Student 465', 'Student 703', 'Student 243', 'Student 517',
      'Student 780', 'Student 932', 'Student 902', 'Student 638', 'Student 724',
      'Student 753', 'Student 628', 'Student 799', 'Student 876', 'Student 920',
      'Student 617', 'Student 900', 'Student 949', 'Student 525', 'Student 450',
      'Student 514', 'Student 549', 'Student 634', 'Student 610', 'Student 586',
      'Student 181', 'Student 252', 'Student 767', 'Student 988', 'Student 361',
      'Student 737', 'Student 205', 'Student 239', 'Student 276', 'Student 402',
      'Student 779', 'Student 480', 'Student 909', 'Student 336', 'Student 322',
      'Student 236', 'Student 869', 'Student 266', 'Student 443', 'Student 985',
      'Student 279', 'Student 128', 'Student 953', 'Student 223', 'Student 346',
      'Student 885', 'Student 246', 'Student 197', 'Student 267', 'Student 378',
      'Student 796', 'Student 311', 'Student 580', 'Student 683', 'Student 388',
      'Student 949', 'Student 530', 'Student 728', 'Student 287', 'Student 868',
      'Student 488', 'Student 702', 'Student 618', 'Student 457', 'Student 392',
      'Student 682', 'Student 743', 'Student 216', 'Student 766', 'Student 463',
      'Student 665', 'Student 267', 'Student 831', 'Student 859', 'Student 156',
      'Student 298', 'Student 973', 'Student 766', 'Student 763', 'Student 371',
      'Student 653', 'Student 912', 'Student 918', 'Student 104', 'Student 646',
      'Student 726', 'Student 261', 'Student 305', 'Student 863', 'Student 315',
      'Student 167', 'Student 784', 'Student 801', 'Student 369', 'Student 585',
      'Student 321', 'Student 325', 'Student 921', 'Student 619', 'Student 376',
      'Student 654', 'Student 424', 'Student 578', 'Student 601', 'Student 255',
      'Student 425', 'Student 913', 'Student 601', 'Student 199', 'Student 548'
    ]
  }
]
```

Input2:

```
db> db.sstudents.aggregate([
...  {$group:{
...    _id:null,collectAGE:{$push:"$age"},
...    collectGPA:{$push:"$gpa"}
...  }}
... ]);
```

Output2:

```
[
  {
    _id: null,
    collectAGE: [
      20, 22, 19, 21, 23,
      18, 24, 20, 22, 19,
      21, 23
    ],
    collectGPA: [
      3.4, 3.8, 3.2, 3.6, 3,
      3.5, 3.9, 3.3, 3.7, 3.1,
      4, 3.5
    ]
  }
]
```

5.\$addToSet**Input1:**

```
db> db.students.aggregate([ { $group: { _id: null, uniqueNames: { $addToSet: "$name" } } } ])
```

Output1:


```
{
  "_id": null,
  "uniqueNames": [
    'Student 102', 'Student 211', 'Student 246', 'Student 800', 'Student 647',
    'Student 635', 'Student 835', 'Student 143', 'Student 924', 'Student 973',
    'Student 948', 'Student 575', 'Student 885', 'Student 180', 'Student 678',
    'Student 457', 'Student 125', 'Student 305', 'Student 156', 'Student 273',
    'Student 721', 'Student 369', 'Student 969', 'Student 804', 'Student 539',
    'Student 514', 'Student 617', 'Student 529', 'Student 159', 'Student 177',
    'Student 267', 'Student 631', 'Student 921', 'Student 859', 'Student 190',
    'Student 853', 'Student 460', 'Student 722', 'Student 367', 'Student 469',
    'Student 171', 'Student 615', 'Student 920', 'Student 440', 'Student 201',
    'Student 252', 'Student 809', 'Student 361', 'Student 502', 'Student 289',
    'Student 488', 'Student 234', 'Student 872', 'Student 594', 'Student 839',
    'Student 570', 'Student 291', 'Student 255', 'Student 458', 'Student 592',
    'Student 500', 'Student 710', 'Student 675', 'Student 567', 'Student 166',
    'Student 690', 'Student 630', 'Student 135', 'Student 422', 'Student 352',
    'Student 499', 'Student 620', 'Student 432', 'Student 493', 'Student 763',
    'Student 784', 'Student 327', 'Student 116', 'Student 401', 'Student 857',
    'Student 762', 'Student 165', 'Student 239', 'Student 821', 'Student 353',
    'Student 707', 'Student 162', 'Student 337', 'Student 504', 'Student 298',
    'Student 426', 'Student 867', 'Student 320', 'Student 679', 'Student 879',
    'Student 938', 'Student 164', 'Student 640', 'Student 540', 'Student 876',
    'Student 199', 'Student 946', 'Student 161', 'Student 766', 'Student 576',
    'Student 543', 'Student 402', 'Student 526', 'Student 953', 'Student 682',
    'Student 984', 'Student 197', 'Student 325', 'Student 236', 'Student 814',
    'Student 706', 'Student 100', 'Student 185', 'Student 820', 'Student 819',
    'Student 351', 'Student 496', 'Student 411', 'Student 925', 'Student 729',
    'Student 624', 'Student 491', 'Student 764', 'Student 495', 'Student 492',
    'Student 212', 'Student 398', 'Student 213', 'Student 237', 'Student 602',
    'Student 961', 'Student 386', 'Student 933', 'Student 733', 'Student 517',
    'Student 431', 'Student 154', 'Student 990', 'Student 404', 'Student 828',
    'Student 998', 'Student 344', 'Student 466', 'Student 131', 'Student 980',
    'Student 988', 'Student 370', 'Student 142', 'Student 104', 'Student 219',
    'Student 892', 'Student 216', 'Student 767', 'Student 372', 'Student 632',
    'Student 726', 'Student 730', 'Student 714', 'Student 145', 'Student 892',
    'Student 167', 'Student 684', 'Student 788', 'Student 210', 'Student 824',
    'Student 965', 'Student 100', 'Student 563', 'Student 450', 'Student 251',
    'Student 873', 'Student 718', 'Student 613', 'Student 468', 'Student 895',
    'Student 901', 'Student 676', 'Student 833', 'Student 276', 'Student 505',
    'Student 789', 'Student 913', 'Student 990', 'Student 701', 'Student 831',
    'Student 341', 'Student 536', 'Student 223', 'Student 378', 'Student 444',
    'Student 425', 'Student 320', 'Student 596', 'Student 651', 'Student 601',
    'Student 908', 'Student 578', 'Student 652', 'Student 549', 'Student 975',
    'Student 136', 'Student 262', 'Student 580', 'Student 484', 'Student 748',
  ]
}
```

Input2:

```
db> db.candidates.aggregate([{$unwind: "$courses"},
... {$group: {_id: null, uniqueCourses: {$addToSet: "$courses"}}}
... ]]);
```

Output2:

```
{
  "_id": null,
  "uniqueCourses": [
    'Literature',
    'Film Studies',
    'Mathematics',
    'Biology',
    'Ecology',
    'Marine Science',
    'Environmental Science',
    'Computer Science',
    'Philosophy',
    'Statistics',
    'Engineering',
    'Chemistry',
    'Robotics',
    'History',
    'Political Science',
    'Sociology',
    'English',
    'Cybersecurity',
    'Artificial Intelligence',
    'Art History',
    'Psychology',
    'Physics',
    'Creative Writing',
    'Music History'
  ]
}
```

Experiment 6

Execute Aggregation Pipeline and its operations (pipeline must contain \$match, \$group, \$sort, \$project, \$skip etc.

1.\$match:**Input:**


```
db> db.sstudent.aggregate([
...  {$match:{age:{$gt:23}}}
...  ])
```

Output:

```
[
  {
    _id: 1,
    name: 'Alice',
    age: 25,
    major: 'Computer Science',
    scores: [ 85, 92, 78 ]
  },
  {
    _id: 3,
    name: 'Charlie',
    age: 28,
    major: 'English',
    scores: [ 75, 82, 89 ]
  }
]
```

2.\$sort:

Input:

```
db> db.sstudent.aggregate([ {$sort:{age:-1}} ] )
```

Output:

```
[
  {
    _id: 3,
    name: 'Charlie',
    age: 28,
    major: 'English',
    scores: [ 75, 82, 89 ]
  },
  {
    _id: 1,
    name: 'Alice',
    age: 25,
    major: 'Computer Science',
    scores: [ 85, 92, 78 ]
  },
  {
    _id: 5,
    name: 'Eve',
    age: 23,
    major: 'Biology',
    scores: [ 80, 77, 93 ]
  },
  {
    _id: 2,
    name: 'Bob',
    age: 22,
    major: 'Mathematics',
    scores: [ 90, 88, 95 ]
  },
  {
    _id: 4,
    name: 'David',
    age: 20,
    major: 'Computer Science',
    scores: [ 98, 95, 87 ]
  }
]
```

3.\$project:

Input:

```
db> db.sstudent.aggregate([ {$project:{_id:0,name:1,age:1}} ] )
```

Output:

```
[
  { name: 'Alice', age: 25 },
  { name: 'Bob', age: 22 },
  { name: 'Charlie', age: 28 },
  { name: 'David', age: 20 },
  { name: 'Eve', age: 23 }
]
```

4.\$group:**Input:**

```
db> db.sstudent.aggregate([
... {$group:{_id:"$major",averageAge:{$avg:"$age"},
... totalStudents:{$sum:1}}}
... ])
```

Output:

```
[
  { _id: 'Mathematics', averageAge: 22, totalStudents: 1 },
  { _id: 'English', averageAge: 28, totalStudents: 1 },
  { _id: 'Biology', averageAge: 23, totalStudents: 1 },
  { _id: 'Computer Science', averageAge: 22.5, totalStudents: 2 }
]
```

5.\$skip:**Input:**

```
db> db.sstudent.aggregate([{$project: { _id: 0, name: 1, averageScore: { $avg: "$scores" } } }, {$match: { averageScore: { $gt: 65 } } }, { $skip: 1 } ] )
```

Output:

```
[
  { name: 'Bob', averageScore: 91 },
  { name: 'Charlie', averageScore: 82 },
  { name: 'David', averageScore: 93.33333333333333 },
  { name: 'Eve', averageScore: 83.33333333333333 }
]
```