Tharun Atithya
May-26, Assement-1

Collection students

```
db.createCollection("students")
```

CURD Operations

1. Insert a new student record with embedded courses and address data.

```
[
 {
  "name": "Eren Yeager",
  "age": 22,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
   { "name": "MongoDB", "score": 87 },
   { "name": "Python", "score": 91 }
  ],
  "address": {
   "city": "Hyderabad",
   "state": "Telangana",
   "pincode": 500032
  },
  "enrollmentDate": ISODate("2024-07-15T00:00:00Z"),
  "isActive": true
 },
 {
  "name": "Levi",
  "age": 24,
  "gender": "Male",
  "department": "Mechanical",
  "courses": [
   { "name": "C++", "score": 78 },
   { "name": "Python", "score": 88 }
  ],
  "address": {
   "city": "Chennai",
   "state": "Tamil Nadu",
   "pincode": 600001
  },
  "enrollmentDate": ISODate("2024-08-01T00:00:00Z"),
  "isActive": false
 },
 {
  "name": "John Snow",
  "age": 23,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
   { "name": "MongoDB", "score": 82 },
   { "name": "Python", "score": 77 }
  ],
  "address": {
   "city": "Mumbai",
   "state": "Maharashtra",
   "pincode": 400001
  },
  "enrollmentDate": ISODate("2024-07-25T00:00:00Z"),
  "isActive": true
 },
 {
  "name": "Patrick Bateman",
  "age": 25,
  "gender": "Male",
  "department": "Mathematics",
```

```
  "courses": [
    { "name": "Statistics", "score": 88 },
    { "name": "Python", "score": 85 }
  ],
  "address": {
    "city": "Delhi",
    "state": "Delhi",
    "pincode": 110001
  },
  "enrollmentDate": ISODate("2024-06-20T00:00:00Z"),
  "isActive": true
},
{
  "name": "Erwin Smith",
  "age": 26,
  "gender": "Male",
  "department": "Mechanical",
  "courses": [
    { "name": "AutoCAD", "score": 81 },
    { "name": "Thermodynamics", "score": 79 }
  ],
  "address": {
    "city": "Pune",
    "state": "Maharashtra",
    "pincode": 411001
  },
  "enrollmentDate": ISODate("2024-08-05T00:00:00Z"),
  "isActive": false
},
{
  "name": "Ichigo Kurosagi",
  "age": 22,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "Python", "score": 92 },
    { "name": "AI", "score": 89 }
  ],
  "address": {
    "city": "Bangalore",
    "state": "Karnataka",
    "pincode": 560001
  },
  "enrollmentDate": ISODate("2024-07-30T00:00:00Z"),
  "isActive": true
},
{
  "name": "Souske Izen",
  "age": 21,
  "gender": "Male",
  "department": "Mathematics",
  "courses": [
    { "name": "MongoDB", "score": 86 },
    { "name": "Python", "score": 90 }
  ],
  "address": {
    "city": "Hyderabad",
    "state": "Telangana",
    "pincode": 500034
  },
  "enrollmentDate": ISODate("2024-07-20T00:00:00Z"),
  "isActive": true
},
{
  "name": "William Butcher",
  "age": 24,
  "gender": "Male",
```

```
  "department": "Computer Science",
  "courses": [
    { "name": "Data Structures", "score": 84 },
    { "name": "MongoDB", "score": 89 }
  ],
  "address": {
    "city": "Kolkata",
    "state": "West Bengal",
    "pincode": 700001
  },
  "enrollmentDate": ISODate("2024-07-10T00:00:00Z"),
  "isActive": false
},
{
  "name": "Steve Rogers",
  "age": 22,
  "gender": "Male",
  "department": "Mechanical",
  "courses": [
    { "name": "Machine Design", "score": 75 },
    { "name": "C++", "score": 80 }
  ],
  "address": {
    "city": "Delhi",
    "state": "Delhi",
    "pincode": 110002
  },
  "enrollmentDate": ISODate("2024-08-08T00:00:00Z"),
  "isActive": true
},
{
  "name": "Ben Tennison",
  "age": 20,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "MongoDB", "score": 85 },
    { "name": "Python", "score": 92 }
  ],
  "address": {
    "city": "Ahmedabad",
    "state": "Gujarat",
    "pincode": 380001
  },
  "enrollmentDate": ISODate("2024-08-01T00:00:00Z"),
  "isActive": true
},
{
  "name": "Monkey D Luffy",
  "age": 23,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "MongoDB", "score": 83 },
    { "name": "Python", "score": 86 }
  ],
  "address": {
    "city": "Hyderabad",
    "state": "Telangana",
    "pincode": 500032
  },
  "enrollmentDate": ISODate("2024-08-01T00:00:00Z"),
  "isActive": true
},
{
  "name": "Ronora Zoro",
  "age": 22,
```

```
  "gender": "Male",
  "department": "Mechanical",
  "courses": [
    { "name": "AutoCAD", "score": 73 },
    { "name": "C++", "score": 84 }
  ],
  "address": {
    "city": "Chennai",
    "state": "Tamil Nadu",
    "pincode": 600002
  },
  "enrollmentDate": ISODate("2024-06-28T00:00:00Z"),
  "isActive": false
},
{
  "name": "Yami Sukihero",
  "age": 21,
  "gender": "Male",
  "department": "Mathematics",
  "courses": [
    { "name": "Statistics", "score": 80 },
    { "name": "Python", "score": 85 }
  ],
  "address": {
    "city": "Hyderabad",
    "state": "Telangana",
    "pincode": 500033
  },
  "enrollmentDate": ISODate("2024-07-17T00:00:00Z"),
  "isActive": true
},
{
  "name": "Asta",
  "age": 20,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "MongoDB", "score": 89 },
    { "name": "Python", "score": 93 }
  ],
  "address": {
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": 400002
  },
  "enrollmentDate": ISODate("2024-07-21T00:00:00Z"),
  "isActive": true
},
{
  "name": "Yhwach",
  "age": 26,
  "gender": "Male",
  "department": "Mathematics",
  "courses": [
    { "name": "Algebra", "score": 90 },
    { "name": "Python", "score": 87 }
  ],
  "address": {
    "city": "Pune",
    "state": "Maharashtra",
    "pincode": 411002
  },
  "enrollmentDate": ISODate("2024-07-22T00:00:00Z"),
  "isActive": false
},
{
  "name": "Harry Poter",
```

```
  "age": 23,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "MongoDB", "score": 84 },
    { "name": "Python", "score": 91 }
  ],
  "address": {
    "city": "Hyderabad",
    "state": "Telangana",
    "pincode": 500032
  },
  "enrollmentDate": ISODate("2024-07-19T00:00:00Z"),
  "isActive": true
},
{
  "name": "Mahendra Singh Dhoni",
  "age": 24,
  "gender": "Male",
  "department": "Mechanical",
  "courses": [
    { "name": "C++", "score": 76 },
    { "name": "Thermodynamics", "score": 81 }
  ],
  "address": {
    "city": "Ranchi",
    "state": "Jharkhand",
    "pincode": 834001
  },
  "enrollmentDate": ISODate("2024-08-10T00:00:00Z"),
  "isActive": true
},
{
  "name": "Virat Kholi",
  "age": 25,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
    { "name": "MongoDB", "score": 86 },
    { "name": "Python", "score": 92 }
  ],
  "address": {
    "city": "Delhi",
    "state": "Delhi",
    "pincode": 110003
  },
  "enrollmentDate": ISODate("2024-08-03T00:00:00Z"),
  "isActive": true
},
{
  "name": "Rohit Sharma",
  "age": 24,
  "gender": "Male",
  "department": "Mathematics",
  "courses": [
    { "name": "Statistics", "score": 85 },
    { "name": "Python", "score": 89 }
  ],
  "address": {
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": 400003
  },
  "enrollmentDate": ISODate("2024-07-26T00:00:00Z"),
  "isActive": true
},
{
```

```
  "name": "Sachin Tendulkar",
  "age": 26,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
   { "name": "MongoDB", "score": 87 },
   { "name": "Python", "score": 94 }
  ],
  "address": {
   "city": "Mumbai",
   "state": "Maharashtra",
   "pincode": 400004
  },
  "enrollmentDate": ISODate("2024-07-18T00:00:00Z"),
  "isActive": true
 },
 {
  "name": "AB De Villars",
  "age": 25,
  "gender": "Male",
  "department": "Computer Science",
  "courses": [
   { "name": "MongoDB", "score": 90 },
   { "name": "Python", "score": 93 }
  ],
  "address": {
   "city": "Bangalore",
   "state": "Karnataka",
   "pincode": 560002
  },
  "enrollmentDate": ISODate("2024-08-02T00:00:00Z"),
  "isActive": true
 }
]
```

2. Update score for a course (Python) inside the courses array.

```
db.students.updateMany(
{"courses.name": "Python"},
{$inc: {"courses.$[course].score": -5}},
{arrayFilters: [{"course.name": "Python"}]}
)
```

3. Delete a student whose name is "John Doe".

```
db.students.deleteOne({name: "John Doe"})
```

4. Find all students in the "Computer Science" department.

```
db.students.find({department: "Computer Science"})
```

Query Operators

5. Find students where age is greater than 20.

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

Tharun Atithya
May-26, Assement-1

6. Find students enrolled between two dates.

```
db.students.find({enrollmentDate: {$gt: new Date("2024-01-01"), $lt: new Date("2024-12-01")}})
```

7. Find students who are either in "Computer Science" or "Mathematics".

```
db.students.find({$or: [{department: "Computer Science"}, {department: "Mathematics"}]})
```

8. Find students not in the "Mechanical" department.

```
db.students.find({department: {$nin: ["Mechanical"]}})
```

9. Find students whose courses.score is greater than 80.

```
db.students.find({"courses.score": {$gt: 80}})
```

Aggregation Framework

10. Group by department and count students.

```
db.students.aggregate(
[
{$group: {_id: "$department", counts: {$sum: 1}}}
]
)
```

11. Calculate average age of students per department.

```
db.students.aggregate(
[
{$group: {
_id: "$department",
averageAge: {$avg: "$age"}
}}
]
)
```

12. Sort students by total course score (computed using $sum inside $project).

```
db.students.aggregate(
[
{$project: {_id: 0, name: 1, totalCourseScore: {$sum: "$courses.score"}}},
{$sort: {totalCourseScore: -1}}
]
)
```

13. Filter only active students before aggregation.

```
db.students.aggregate(
[
{$match: {isActive: true}},
{$project: {_id:0, name:1, isActive: 1}}
]
)
```

Tharun Atithya
May-26, Assement-1

14. Group and list unique cities from the address field.

```
db.students.aggregate(
[
{$group: {_id: "$address.city"}},
{$project: {_id: 0, uniqueCity: "$_id"}}
]
)
```

Projections

15. Find students with only name , department , and city fields shown.

```
db.students.find({}, {_id: false, name: true, department: true, "address.city": true})
```

16. Exclude the _id field from output.

```
db.students.find({}, {_id: false})
```

17. Show each student's name and total score using $project .

```
 db.students.aggregate(
[
{$project: {_id: 0, name: 1, totalScore: {$sum: "$courses.score"}}}
]
)
```

Embedded Documents

18. Query students where address.city = "Hyderabad".

```
db.students.find({"address.city": "Hyderabad"})
```

19. Update address.pincode for a student.

```
db.students.updateOne({name: "Eren Yeager"}, {$set: {"address.pincode": 638701}})
```

20. Add a new field landmark to all address objects.

```
db.students.updateMany({}, {$set: {"address.landmark": null}})
```

Array Operations

21. Add a new course "Node.js" to a student's courses array.

```
db.students.updateOne({name: "Monkey D Luffy"}, {$push: {courses: {name: "Node.js", score: null}}})
```

Tharun Atithya
May-26, Assement-1

22. Remove a course by name "MongoDB" from the array.

```
db.students.updateOne({"courses.name": "MongoDB"}, {$pull: {courses: {name: "MongoDB"}}})
```

23. Find students who have enrolled in both Python and MongoDB.

```
db.students.find({"courses.name": {$all: ["Python", "MongoDB"]}})
```

24. Use $elemMatch to query students where score in MongoDB > 80.

```
db.students.find({courses: {$elemMatch: {name: "MongoDB", score: {$gt: 80}}}})
```