May 26 Assessment

Records inserted:

```
db.students.insertMany([
...   {
...     "name": "Adrian Saravanan",
...     "age": 21,
...     "gender": "Male",
...     "department": "Artificial Intelligence and Data Science",
...     "courses": [
...       { "name": "MongoDB", "score": 87 },
...       { "name": "Python", "score": 89 },
...       { "name": "Azure", "score": 78 }],
...     "address": {
...       "city": "Chennai",
...       "state": "Tamil Nadu",
...       "pincode": 600754},
...     "enrollmentDate": ISODate("2025-05-10T00:00:00Z"),
...     "isActive": true},
...   {"name": "Fathima Zahira",
...     "age": 21,
...     "gender": "Female",
...     "department": "Computer Science",
...     "courses": [
...       { "name": "Java", "score": 88 },
...       { "name": "MongoDB", "score": 92 },
...       { "name": "Python", "score": 85 }],
...     "address": {
...       "city": "Bangalore",
...       "state": "Karnataka",
...       "pincode": 643001},
...     "enrollmentDate": ISODate("2025-02-15T00:00:00Z"),
...     "isActive": true},
...   {"name": "John Doe",
...     "age": 19,
...     "gender": "Male",
...     "department": "Mechanical",
...     "courses": [
...       { "name": "Thermodynamics", "score": 75 },
...       { "name": "CAD", "score": 82 }],
...     "address": {
...       "city": "Chennai",
...       "state": "Tamil Nadu",
...       "pincode": 600001
...     "enrollmentDate": ISODate("2024-12-10T00:00:00Z"),
...     "isActive": false},
...   {
...     "name": "Reshma Parag",
...     "age": 20,
...     "gender": "Female",
...     "department": "Computer Science",
...     "courses": [
...       { "name": "MongoDB", "score": 92 },
...       { "name": "Python", "score": 88 },
...       { "name": "JavaScript", "score": 85 }],
...     "address": {
...       "city": "Hyderabad",
...       "state": "Telangana",
...       "pincode": 600123},
...     "enrollmentDate": ISODate("2024-06-20T00:00:00Z"),
...     "isActive": true}])
```

CRUD Operations

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

```
db.students.insertOne({
...   "name": "Cristiano De Santos",
...   "age": 24,
...   "gender": "Male",
...   "department": "Computer Science",
...   "courses": [
...     { "name": "Node.js", "score": 80 },
...     { "name": "React", "score": 85 }],
...   "address": {
```

```
...    "city": "Mumbai",
...    "state": "Maharashtra",
...    "pincode": 400001},
...   "enrollmentDate": ISODate("2025-01-10T00:00:00Z"),
...   "isActive": true})
```

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

```
db.students.updateOne(
...   { "name": "Adrian Saravanan", "courses.name": "Python" },
...   { $set: { "courses.$.score": 95 } })
```

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 }})
```

6. Find students enrolled between two dates.

```
db.students.find({
...   "enrollmentDate": {
...     $gte: ISODate("2025-01-01"),
...     $lte: ISODate("2025-06-30")}})
```

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

```
db.students.find({
...   "department": {
```

```
...    $in: ["Computer Science", "Mathematics"]}}
...     )
```

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

```
db.students.find({ "department": { $ne: "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", count: { $sum: 1 } } }])
```

11. Calculate average age of students per department.

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

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

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

13. Filter only active students before aggregation.

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

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

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

Projections

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

```
db.students.find({}, { name: 1, department: 1, "address.city": 1, _id: 0 })
```

16. Exclude the _id field from output.

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

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

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

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

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

19. Update address.pincode for a student.

```
db.students.updateOne(
...   { "name": "Reshma Parag" },
...   { $set: { "address.pincode": 500032 } })
```

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

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

Array Operations

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

```
db.students.updateOne(
 … { "name": "Adrian Saravanan" },
 … { $push: { "courses": { "name": "Node.js", "score": 80 } } })
```

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

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

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

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

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

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