**Experiment 4:**

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

**Insert a Document with a Single Embedded Document**

A **single embedded document** means one field contains another document as its value.

db.students.insertOne({

\_id: 57,

name: "Sanjay",

age: 22,

department: "AIML",

address: {

street: "MG Road",

city: "Bangalore",

state: "Karnataka",

pincode: 560001

}

})

**Insert a Document with Multiple Embedded Documents**

A **document containing multiple embedded documents** is useful for structured data like subjects, projects, etc.

db.students.insertOne({

\_id: 58,

name: "Meera",

age: 23,

department: "CSE",

subjects: [

{ name: "Math", score: 90 },

{ name: "AI", score: 95 },

{ name: "DBMS", score: 88 }

],

address: {

street: "Brigade Road",

city: "Bangalore",

state: "Karnataka",

pincode: 560025

}

})

**Insert a Document with Multi-Level Embedded Documents**

A **multi-level embedded document** means nested documents inside other documents.

db.students.insertOne({

\_id: 59,

name: "Arjun",

age: 24,

department: "AIML",

contact: {

phone: "9876543210",

email: "arjun@example.com",

emergency\_contact: {

name: "Ravi Kumar",

relation: "Father",

phone: "8765432109"

}

},

projects: [

{

title: "AI Chatbot",

year: 2024,

team: [

{ name: "Arjun", role: "Developer" },

{ name: "Meera", role: "Designer" }

]

},

{

title: "Big Data Analysis",

year: 2023,

team: [

{ name: "Arjun", role: "Data Engineer" },

{ name: "Sanjay", role: "Analyst" }

]

}

]

})

**Adding an Array While Inserting a New Document**

db.students.insertOne({

\_id: 56,

name: "Rohit",

age: 24,

department: "AIML",

subjects: [

{ name: "Math", score: 88 },

{ name: "AI", score: 91 },

{ name: "DBMS", score: 82 }

],

skills: ["Python", "Machine Learning", "Data Science"]

})

db.students.insertMany([

{

\_id: 60,

name: "Amit",

age: 24,

department: "AIML",

subjects: [

{ name: "Math", score: 85 },

{ name: "AI", score: 90 },

{ name: "DBMS", score: 75 }

],

attendance: [92, 88, 85, 90, 95]

},

{

\_id: 61,

name: "Priya",

age: 23,

department: "CSE",

subjects: [

{ name: "Math", score: 95 },

{ name: "AI", score: 92 },

{ name: "DBMS", score: 88 }

],

attendance: [98, 99, 97, 95, 96]

}

])

**Using $ Projection Operator**

**Retrieve only the first matching subject where the student studied AI.**

db.students.find(

{ "subjects.name": "AI" },

{ name: 1, "subjects.$": 1, \_id: 0 }

)

**Using $elemMatch Projection Operator**

**Retrieve only the subject where Priya scored more than 90.**

db.students.find(

{ name: "Priya" },

{ name: 1, subjects: { $elemMatch: { score: { $gt: 90 } } }, \_id: 0 }

)

**Using $slice Projection Operator**

**Retrieve only the first 3 attendance records for Amit.**

db.students.find(

{ name: "Amit" },

{ name: 1, attendance: { $slice: 3 }, \_id: 0 }

)