// Insert Employee documents with embedded fields for aggregation examples

db.Employee.insertMany([

{

Emp\_id: 3,

Name: { FName: "John", LName: "Doe" },

CompanyName: "Infosys",

Salary: 80000,

Designation: "DBA",

Age: 34,

Expertise: ["MongoDB", "MySQL"],

DOB: "1989-04-12",

Email: "john.doe@infosys.com",

Contact: "9876543210",

Address: [{ PAddr: "789 Main St", LAddr: "456 Lane St" }]

},

// More documents as required

]);

// 1. Aggregation to get Designations with total salary above 200000

db.Employee.aggregate([

{ $group: { \_id: "$Designation", totalSalary: { $sum: "$Salary" } } },

{ $match: { totalSalary: { $gt: 200000 } } }

]);

// 2. Aggregation to return Name and \_id in uppercase, sorted alphabetically

db.Employee.aggregate([

{ $project: { Name: { FName: { $toUpper: "$Name.FName" }, LName: { $toUpper: "$Name.LName" } }, \_id: 1 } },

{ $sort: { "Name.FName": 1, "Name.LName": 1 } }

]);

// 3. Total Salary for each City with Designation "DBA"

db.Employee.aggregate([

{ $match: { Designation: "DBA" } },

{ $group: { \_id: "$Address.city", totalSalary: { $sum: "$Salary" } } }

]);

// 4. Create Single Field Index on Designation

db.Employee.createIndex({ Designation: 1 });

// 5. Create Multikey Index on Expertise field

db.Employee.createIndex({ Expertise: 1 });

// 6. Create an index on Emp\_id, and compare search time before and after

// Note: Add 10,000 documents before this test for noticeable time comparison.

db.Employee.createIndex({ Emp\_id: 1 });

// 7. List all indexes created on Employee collection

db.Employee.getIndexes();