db.Employee.insertMany([

{

Emp\_id: 4,

Name: { FName: "Mary", LName: "Green" },

CompanyName: "Wipro",

Salary: 60000,

Designation: "Developer",

Age: 30,

Expertise: ["JavaScript", "MongoDB", "Cassandra"],

DOB: "1992-07-19",

Email: "mary.green@wipro.com",

Contact: "4561237890",

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

},

// Additional documents as needed

]);

// 1. Split values in Expertise array and sum occurrences

db.Employee.aggregate([

{ $unwind: "$Expertise" },

{ $group: { \_id: "$Expertise", total: { $sum: 1 } } }

]);

// 2. Return Max and Min Salary for each company

db.Employee.aggregate([

{ $group: { \_id: "$CompanyName", maxSalary: { $max: "$Salary" }, minSalary: { $min: "$Salary" } } }

]);

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

db.Employee.aggregate([

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

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

]);

// 4. Separate values in Expertise array for employee with Name "Swapnil Jadhav"

db.Employee.aggregate([

{ $match: { "Name.FName": "Swapnil", "Name.LName": "Jadhav" } },

{ $unwind: "$Expertise" },

{ $group: { \_id: "$Emp\_id", Expertise: { $addToSet: "$Expertise" } } }

]);

// 5. Create Compound Index on Name (ascending) and Age (descending)

db.Employee.createIndex({ "Name.FName": 1, Age: -1 });

// 6. Create Index on Emp\_id and compare search time before/after

// Add 10,000+ documents for significant results

db.Employee.find({ Emp\_id: 5000 }).explain("executionStats");

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

db.Employee.find({ Emp\_id: 5000 }).explain("executionStats");

// 7. Return a list of indexes on Employee collection

db.Employee.getIndexes();