B2

db.Employee.insertOne({

EmpId: 101,

Name: { FName: "John", LName: "Doe" }, // You can modify the Name

CompanyName: "TCS",

Salary: 30000, // You can modify Salary

Designation: "Tester",

Age: 25,

Expertise: ["Java", "MongoDB"], // You can add more fields as required

DOB: "1998-05-12",

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

Contract: "Permanent",

Address: [{ PAddr: "Mumbai", LAddr: "Pune" }]

})

1. Return Designation with Total Salary Above 200000

db.Employee.aggregate([

{

$group: {

\_id: "$Designation",

totalSalary: { $sum: "$Salary" }

}

},

{

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

}

])

2) Find Employee with Total Salary for Each City with Designation="DBA"

db.Employee.aggregate([

{

$match: { Designation: "DBA" }

},

{

$group: {

\_id: "$Address.PAddr", // Assuming Address contains PAddr as City

totalSalary: { $sum: "$Salary" }

}

}

])

3) Find Total Salary of Employee with Designation="DBA" for Each Company

db.Employee.aggregate([

{

$match: { Designation: "DBA" }

},

{

$group: {

\_id: "$CompanyName",

totalSalary: { $sum: "$Salary" }

}

}

])

4) Returns Names and \_id in Upper Case and in Alphabetical Order

db.Employee.aggregate([

{

$project: {

nameUpperCase: { $toUpper: "$Name.FName" },

\_id: 1

}

},

{

$sort: { nameUpperCase: 1 }

}

])

5) Count All Records from Collection

db.Employee.countDocuments()

6) Return Values in the Expertise Array where Name of Employee="Swapnil"

db.Employee.aggregate([

{

$group: {

\_id: "$Designation",

avgSalary: { $avg: "$Salary" }

}

},

{

$sort: { avgSalary: 1 }

}

])

7) Return Values in the Expertise Array where Name of Employee="Swapnil"

db.Employee.find(

{ "Name.FName": "Swapnil" },

{ Expertise: 1, \_id: 0 }

)

8) Return Values in the Expertise Array and Sum Each Element of Array

db.Employee.aggregate([

{ $unwind: "$Expertise" },

{

$group: {

\_id: "$Expertise",

totalCount: { $sum: 1 }

}

}

])

9) Return Array for Designation whose Address is "Pune"

db.Employee.find(

{ "Address.PAddr": "Pune" },

{ Designation: 1, \_id: 0 }

)

10) Return Max and Min Salary for Each Company

db.Employee.aggregate([

{

$group: {

\_id: "$CompanyName",

maxSalary: { $max: "$Salary" },

minSalary: { $min: "$Salary" }

}

}

])

B…

**B. Indexing Operations**

1. **To Create Single Field Index on Designation**

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

1. **To Create Compound Index on Name (Ascending) and Age (Descending)**

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

1. **To Create Multikey Index on Expertise Array**

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

1. **Return a List of All Indexes on Collection**

db.Employee.getIndexes()