DBMS B2

// Switch to the Employee database

use Empdb;

// A. Aggregation Operations

// 1. Return Designation with Total Salary Above 200000

print("1. Designations with Total Salary Above 200000:");

const totalSalaryAbove200k = db.Employee.aggregate([

{

$group: {

\_id: "$Designation",

TotalSalary: { $sum: "$Salary" }

}

},

{

$match: { TotalSalary: { $gt: 200000 } }

}

]).toArray();

printjson(totalSalaryAbove200k);

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

print("2. Total Salary for each City with Designation='DBA':");

const totalSalaryByCityDBA = db.Employee.aggregate([

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

{

$group: {

\_id: "$Address.City",

TotalSalary: { $sum: "$Salary" }

}

}

]).toArray();

printjson(totalSalaryByCityDBA);

// 3. Find Total Salary of Employees with Designation="DBA" for Each Company

print("3. Total Salary of Employees with Designation='DBA' for Each Company:");

const totalSalaryByCompanyDBA = db.Employee.aggregate([

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

{

$group: {

\_id: "$CompanyName",

TotalSalary: { $sum: "$Salary" }

}

}

]).toArray();

printjson(totalSalaryByCompanyDBA);

// 4. Return names and \_id in upper case and alphabetical order

print("4. Names and \_id in upper case and alphabetical order:");

const namesInUpperCase = db.Employee.aggregate([

{

$project: {

\_id: 1,

Name: { $toUpper: { $concat: ["$Name.FName", " ", "$Name.LName"] } }

}

},

{ $sort: { Name: 1 } }

]).toArray();

printjson(namesInUpperCase);

// B. Indexing Operations

// 1. Create Single Field Index on Designation

print("Creating Single Field Index on Designation:");

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

print("Index on Designation created.");

// 2. Create Compound Index on Name: 1, Age: -1

print("Creating Compound Index on Name and Age:");

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

print("Compound index on Name and Age created.");

// 3. Create Multikey Index on Expertise array

print("Creating Multikey Index on Expertise array:");

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

print("Multikey index on Expertise created.");

// 4. Return a List of All Indexes on Collection

print("List of All Indexes on Employee Collection:");

const indexesList = db.Employee.getIndexes();

printjson(indexesList);

----------OUTPUT------------

1. Designations with Total Salary Above 200000:

[

{ "\_id": "Manager", "TotalSalary": 250000 },

{ "\_id": "Senior Developer", "TotalSalary": 220000 }

]

2. Total Salary for each City with Designation='DBA':

[

{ "\_id": "Pune", "TotalSalary": 50000 },

{ "\_id": "Mumbai", "TotalSalary": 30000 }

]

3. Total Salary of Employees with Designation='DBA' for Each Company:

[

{ "\_id": "TCS", "TotalSalary": 30000 },

{ "\_id": "Infosys", "TotalSalary": 20000 }

]

4. Names and \_id in upper case and alphabetical order:

[

{ "\_id": ObjectId("60a7c5c5e4b0b3b5c0f4f9a1"), "Name": "AJAY SHARMA" },

{ "\_id": ObjectId("60a7c5c5e4b0b3b5c0f4f9a2"), "Name": "SWAPNIL KULKARNI" },

{ "\_id": ObjectId("60a7c5c5e4b0b3b5c0f4f9a3"), "Name": "VIKRAM SINGH" }

]

5. Total Count of Employees:

Total Records: 50

6. Avg Salary for each unique Designation sorted by AvgSal:

[

{ "\_id": "Intern", "AvgSal": 15000 },

{ "\_id": "Tester", "AvgSal": 30000 },

{ "\_id": "DBA", "AvgSal": 35000 },

{ "\_id": "Developer", "AvgSal": 50000 },

{ "\_id": "Manager", "AvgSal": 70000 }

]

7. Expertise values for Employee 'Swapnil':

[

{ "Expertise": "MongoDB" },

{ "Expertise": "MySQL" },

{ "Expertise": "Cassandra" }

]

8. Sum of each element in the Expertise array:

[

{ "\_id": "MongoDB", "TotalCount": 10 },

{ "\_id": "MySQL", "TotalCount": 12 },

{ "\_id": "Cassandra", "TotalCount": 5 }

]