DBMS – Assignment-B2 – Softcopy

Queries

Group-A

1. Return Designation with Total Salary Above 20000:

```
db.Employee.aggregate([
    {
        $group: {
            _id: "$Designation",
            TotalSalary: { $sum: "$Salary" }
        }
    },
    {
        $match: {
            TotalSalary: { $gt: 20000 }
        }
    }
])
[
  { _id: 'HR', TotalSalary: 39000 },
  { _id: 'Tester', TotalSalary: 120000 },
  { _id: 'Project Manager', TotalSalary: 70000 },
  { _id: 'Programmer', TotalSalary: 84000 },
  { _id: 'Designer', TotalSalary: 50000 },
  { _id: 'Architect', TotalSalary: 80000 }
]
```

2. Find Employee with Total Salary for Each City with Designation "Tester":

```
db.Employee.aggregate([
    {
        $match: { Designation: "Tester" }
    },
    {
        $group: {
            _id: { $arrayElemAt: ["$Address.PAddr", 0] },
            TotalSalary: { $sum: "$Salary" }
        }
    }
])
[
  { _id: '789, Street G, Pune', TotalSalary: 45000 },
  { _id: '345, Street C, Hyderabad', TotalSalary: 45000 },
  { _id: '123 Street', TotalSalary: 30000 }
]
```

3. Find Total Salary of Employee with Designation "Tester" for Each Company:

```
db.Employee.aggregate([
    {
        $match: { Designation: "Tester" }
    },
    {
        $group: {
            _id: "$Company_name",
            TotalSalary: { $sum: "$Salary" }
        }
    }
])
[
  { _id: 'Lex Corp.', TotalSalary: 45000 },
  { _id: 'Y-Space', TotalSalary: 75000 }
]
```

4. Returns Names and _id in Upper Case and in Alphabetical Order:

```
db.Employee.aggregate([
    {
        $project: {
            _id: 1,
            Name: { $toUpper: { $concat: ["$Name.FName", " ",
"$Name.LName"] } }
        }
    },
    {
        $sort: { Name: 1 }
    }
])
[
  { _id: ObjectId('671118cb722906af4efe6918'), Name: 'ADITYA GUNDETI' },
  { _id: ObjectId('671118cb722906af4efe691a'), Name: 'AFAN SHAIKH' },
  { _id: ObjectId('6711195c7b947b9ca67f6f0b'), Name: 'ANIL SALVI' },
  { _id: ObjectId('671118cb722906af4efe6911'), Name: 'AYUSH KALASKAR' },
  { _id: ObjectId('671118cb722906af4efe6913'), Name: 'MEHUL PATIL' },
  {
    _id: ObjectId('671118cb722906af4efe6917'),
    Name: 'PRASHANT KALASKAR'
  },
  { _id: ObjectId('671118cb722906af4efe6915'), Name: 'RAJENDRA PATIL' },
  { _id: ObjectId('671118cb722906af4efe6916'), Name: 'RAJESH PATIL' },
  { _id: ObjectId('671118cb722906af4efe6914'), Name: 'TANMAY MACHKAR' }
]
```

5. Count All Records from Collection:

```
db.Employee.countDocuments() -> 9
```

6. For Each Unique Designation, Find Avg Salary and Output Sorted by AvgSal:

```
db.Employee.aggregate([
    {
        $group: {
            _id: "$Designation",
            AvgSalary: { $avg: "$Salary" }
        }
    },
    {
        $sort: { AvgSalary: 1 }
    }
])
[
  { _id: 'HR', AvgSalary: 39000 },
  { _id: 'Tester', AvgSalary: 40000 },
  { _id: 'Programmer', AvgSalary: 42000 },
  { _id: 'Designer', AvgSalary: 50000 },
  { _id: 'Project Manager', AvgSalary: 70000 },
  { _id: 'Architect', AvgSalary: 80000 }]
```

7. Return Separate Value in the Expertise Array Where Name of Employee is "Aditya":

```
db.Employee.aggregate([
    {
        $match: { "Name.FName": "Aditya" }
    },
    {
        $unwind: "$Expertise"
    },
    {
        $project: { Expertise: 1 }
    }
])
[
  { _id: ObjectId('671118cb722906af4efe6918'), Expertise: 'Cloud' },
  {
    _id: ObjectId('671118cb722906af4efe6918'),
    Expertise: 'Microservices' }]
```

8. Return Separate Value in the Expertise Array and Return Sum of Each Element of Array:

```
db.Employee.aggregate([
    {
        $unwind: "$Expertise"
    },
    {
        $group: {
            _id: "$Expertise",
            TotalCount: { $sum: 1 }
        }
    }
])
[
  { _id: 'Agile', TotalCount: 1 },
  { _id: 'MongoDB', TotalCount: 1 },
  { _id: 'Illustrator', TotalCount: 1 },
  { _id: 'Angular', TotalCount: 1 },
  { _id: 'Cloud', TotalCount: 1 },
  { _id: 'Java', TotalCount: 3 },
  { _id: 'Microservices', TotalCount: 1 },
  { _id: 'Python', TotalCount: 1 },
  { _id: 'Scrum', TotalCount: 1 },
  { _id: 'Selenium', TotalCount: 2 },
  { _id: 'Spring', TotalCount: 1 },
  { _id: 'Photoshop', TotalCount: 1 },
  { _id: 'Recruitment', TotalCount: 1 },
  { _id: 'Employee Relations', TotalCount: 1 }
]
```

9. Return Array for Designation Whose Address is "Pune":

```
db.Employee.aggregate([
    {
        $match: { "Address.PAddr": { $regex: "Pune" } }
    },
    {
        $project: { Designation: 1 }
    }
])
[
  {
    _id: ObjectId('671118cb722906af4efe6911'),
    Designation: 'Programmer'
 },
  { _id: ObjectId('671118cb722906af4efe6917'), Designation: 'Tester' }
]
```

10. Return Max and Min Salary for Each Company:

```
db.Employee.aggregate([
    {
        $group: {
            _id: "$Company_name",
            MaxSalary: { $max: "$Salary" },
            MinSalary: { $min: "$Salary" }
        }
    }
])
[
  { _id: 'Roxonn', MaxSalary: 50000, MinSalary: 50000 },
  { _id: 'SNASA', MaxSalary: 80000, MinSalary: 80000 },
  { _id: 'Stark Industries', MaxSalary: 32000, MinSalary: 32000 },
 { _id: 'Lex Corp.', MaxSalary: 45000, MinSalary: 45000 },
  { _id: 'Y-Space', MaxSalary: 45000, MinSalary: 30000 },
  { _id: 'Wayne Industries', MaxSalary: 70000, MinSalary: 70000 },
  { _id: 'Vought', MaxSalary: 39000, MinSalary: 39000 },
  { _id: 'Oscorp', MaxSalary: 52000, MinSalary: 52000 }
]
```

Group-B

1. Create Single Field Indexes on Designation:

```
db.Employee.createIndex({ Designation: 1 })
Designation_1
```

2. Create Compound Indexes on Name and Age:

```
db.Employee.createIndex({ "Name.FName": 1, Age: -1 })
Name.FName_1_Age_-1
```

3. Create Multikey Indexes on Expertise Array:

```
db.Employee.createIndex({ Expertise: 1 })
Expertise_1
```

4. Return a List of All Indexes on Collection:

```
db.Employee.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { Designation: 1 }, name: 'Designation_1' },
  {
    v: 2,
    key: { 'Name.FName': 1, Age: -1 },
    name: 'Name.FName_1_Age_-1'
  },
  { v: 2, key: { Expertise: 1 }, name: 'Expertise_1' }
]
```

5. Rebuild Indexes:

```
db.Employee.reIndex()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { Designation: 1 }, name: 'Designation_1' },
  {
    v: 2,
    key: { 'Name.FName': 1, Age: -1 },
    name: 'Name.FName_1_Age_-1'
  },
  { v: 2, key: { Expertise: 1 }, name: 'Expertise_1' }
]
empDB> db.Employee.reIndex()
{
  nIndexesWas: 4,
  nIndexes: 4,
  indexes: [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { Designation: 1 }, name: 'Designation_1' },
    {
      v: 2,
      key: { 'Name.FName': 1, Age: -1 },
      name: 'Name.FName_1_Age_-1'
    },
    { v: 2, key: { Expertise: 1 }, name: 'Expertise_1' }
  ],
  ok: 1
}
```

6. Drop Index on Remove Specific Index:

```
db.Employee.dropIndex("Designation_1")
{ nIndexesWas: 4, ok: 1 }
```

7. Remove All Indexes Except for the _id Index from a Collection:

```
db.Employee.dropIndexes()
{
  nIndexesWas: 3,
  msg: 'non-_id indexes dropped for collection',
  ok: 1 }
```

--- END OF DOCUMENT ---