Queries-B2

Group A

> *[!NOTE] Use Employee database created in Assignment B-01 and perform following aggregation operation Refer Queries-B1 (https://git.kska.io/sppu-te-comp-content/DatabaseManagementSystems/src/branch/main/Practical/Assignment-B1/Queries-B1.md)*

1. Return Designation with Total Salary Above 20000:

```
db.Employee.aggregate([
    {
        $group: {
            _id: "$Designation",
            TotalSalary: { $sum: "$Salary" }
        }
    },
    {
        $match: {
            TotalSalary: { $gt: 20000 }
        }
    }
])
```

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

```
db.Employee.aggregate([
  {
    $match: {
      Designation: "Developer"
    }
  },
  {
    $group: {
      _id: "$Address.PAddr",
      Total: { $sum: "$Salary" }
    }
  }
])
```

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" }
        }
    }
])
```

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 }
    }
])
```

5. Count All Records from Collection:

```
db.Employee.countDocuments()
```

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 }
    }
])
```

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 }
    }
])
```

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 }
        }
    }
])
```

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

```
db.Employee.aggregate([
    {
        $match: { "Address.PAddr": { $regex: "Pune" } }
    },
    {
        $project: { Designation: 1 }
    }
])
```

10. Return Max and Min Salary for Each Company:

```
db.Employee.aggregate([
    {
        $group: {
            _id: "$Company_name",
            MaxSalary: { $max: "$Salary" },
            MinSalary: { $min: "$Salary" }
        }
    }
])
```

Group B

> *[!NOTE] Use Employee database created in Assignment B-01 and perform following aggregation operation Refer Queries-B1 (https://git.kska.io/sppu-te-comp-content/DatabaseManagementSystems/src/branch/main/Practical/Assignment-B1/Queries-B1.md)*

1. Create Single Field Indexes on Designation:

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

2. Create Compound Indexes on Name and Age:

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

3. Create Multikey Indexes on Expertise Array:

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

4. Return a List of All Indexes on Collection:

```
db.Employee.getIndexes()
```

5. Rebuild Indexes:

```
db.Employee.reIndex()
```

6. Drop Index on Remove Specific Index:

```
db.Employee.dropIndex("Designation_1")
```

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

```
db.Employee.dropIndexes()
```