DBMS Assignment B3

1. Return Designation with Total Salary is Above 100000

```
>db.empqw.aggregate([{$group:{_id:"$Designation",total:{$sum:"$Salary"}}},{$match:{total:{$gt
:100000}}}])
{ "_id" : "Developer", "total" : 110000 }
{ "_id" : "Designer", "total" : 115000 }
```

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

```
> db.empqw.aggregate([{$group:{_id:"$Designation", "Total_Salary":{$sum:"$Salary"}}}]);
{ "_id" : "Developer", "Total_Salary" : 110000 }
{ "_id" : "Designer", "Total_Salary" : 115000 }
{ "_id" : "Programmer", "Total_Salary" : 100000 }
{ "_id" : "Tester", "Total_Salary" : 45000 }
```

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

```
>db.empqw.aggregate([{$group:{_id:"$Designation",_id:"$Company_Name","Total_Salary":{$su
m:"$Salary"}}}]);
{ "_id" : "TCS", "Total_Salary" : 95000 }
{ "_id" : "Infosys", "Total_Salary" : 45000 }
{ "_id" : "GB lab", "Total_Salary" : 20000 }
{ "_id" : "capgemini", "Total_Salary" : 60000 }
{ "_id" : "IBM", "Total_Salary" : 35000 }
{ "_id" : "Nvidia", "Total_Salary" : 50000 }
{ "_id" : "VM ware", "Total_Salary" : 65000 }
```

4. Returns names and _id in upper case and in alphabetical order.

```
>
db.empqw.aggregate([{$project:{Fname:{$toUpper:"$Name.Fname"},Lname:{$toUpper:"$Name
.Lname"}}}]).pretty()
{
        "_id" : ObjectId("5fe11ede0b4d2650a1422ce9"),
        "Fname" : "SHREYAS",
        "Lname" : "CHAUDHARI"
}
{
        "_id" : ObjectId("5fe11ede0b4d2650a1422cea"),
        "Fname" : "DHRUVIL",
        "Lname" : "SHAH"
}
{
        "_id" : ObjectId("5fe11ede0b4d2650a1422ceb"),
        "Fname" : "GAURAV",
        "Lname" : "VERMA"
```

```
}
{
	"_id" : ObjectId("5fe11ede0b4d2650a1422cec"),
	"Fname" : "SUDESH",
	"Lname" : "PAWAR"
}
{
	"_id" : ObjectId("5fe11edf0b4d2650a1422ced"),
	"Fname" : "RAM",
	"Lname" : "PATEL"
}
{
	"_id" : ObjectId("5fe11edf0b4d2650a1422cee"),
	"Fname" : "POOJA",
	"Lname" : "PATEL"
}
{
	"_id" : ObjectId("5fe11edf0b4d2650a1422cef"),
	"Fname" : "VIKAS",
	"Lname" : "GUPTA"
}
{
	"_id" : ObjectId("5fe11edf0b4d2650a1422cf0"),
	"Fname" : "GITA",
	"Lname" : "RAO"
}
```


5. Count all records from collection

```
> db.empqw.find().count()
8
```

6. For each unique Designation, find avg Salary and output is sorted by AvgSal

```
>db.empqw.aggregate([{$group:{_id:"$Designation",AvgAmount:{$avg:"$Salary"}}},{$sort:{AvgAmount:1}}]).pretty()
{ "_id" : "Tester", "AvgAmount" : 22500 }
{ "_id" : "Programmer", "AvgAmount" : 50000 }
{ "_id" : "Developer", "AvgAmount" : 55000 }
{ "_id" : "Designer", "AvgAmount" : 57500 }
```

7. Return separates value in the Expertise array where Name of Employee="Swapnil"

```
> db.empqw.find({"Name.Fname":"Gaurav"},{Expertise:1,_id:0}).pretty()
{ "Expertise" : [ "Mysql", "R language", "UI/UX" ] }
```

8. Return separates value in the Expertise array and return sum of each element of array
>db.empqw.aggregate([{$match:{Expertise:{$not:{$size:0}}}},{$unwind:"$Expertise"},{$group:{_id:"$Expertise",count:{$sum:1}}}])
{ "_id" : "YACC", "count" : 1 }
{ "_id" : "Cloud", "count" : 1 }
{ "_id" : "Java", "count" : 1 }
{ "_id" : "mongoDB", "count" : 4 }
{ "_id" : "Cpp", "count" : 2 }
{ "_id" : "python", "count" : 1 }
{ "_id" : "MySQL", "count" : 1 }
{ "_id" : "Sqlite", "count" : 1 }
{ "_id" : "Mysql", "count" : 3 }
{ "_id" : "R language", "count" : 2 }
{ "_id" : "DSA", "count" : 1 }
{ "_id" : "Cassandra", "count" : 1 }
{ "_id" : "JavaScript", "count" : 2 }
{ "_id" : "Html", "count" : 1 }
{ "_id" : "scala", "count" : 1 }
{ "_id" : "UI/UX", "count" : 1 }

9. Return Array for Designation whose address is "Pune"
> db.empqw.distinct("Designation",{"Address.city":"pune"})
["Developer"]

10. Return Max and Min Salary for each company.
> db.empqw.aggregate([{$group:{_id:"$Company_Name",minSalary:{$min:"$Salary"}}}]).pretty()
{ "_id" : "TCS", "minSalary" : 25000 }
{ "_id" : "GB lab", "minSalary" : 20000 }
{ "_id" : "Infosys", "minSalary" : 45000 }
{ "_id" : "capgemini", "minSalary" : 60000 }
{ "_id" : "IBM", "minSalary" : 35000 }
{ "_id" : "VM ware", "minSalary" : 65000 }
{ "_id" : "Nvidia", "minSalary" : 50000 }

B. Use Employee database created in Assignment 01 and perform following indexing operation

1. To Create Single Field Indexes on Designation

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

{

"createdCollectionAutomatically" : true,

"numIndexesBefore" : 1,

```
 "numIndexesAfter" : 2,
 "ok" : 1
}
```

2. To Create Compound Indexes on Name: 1, Age: -1

```
> db.Employee.createIndex({Name:1,Age:-1})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 2,
    "numIndexesAfter" : 3,
    "ok" : 1
}
```

3. To Create Multikey Indexes on Expertise array

```
> db.Employee.createIndex({Expertise:1})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 3,
    "numIndexesAfter" : 4,
    "ok" : 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" : 1,
                "Age" : -1
            },
            "name" : "Name_1_Age_-1"
        },
        {
            "v" : 2,
            "key" : {
                "Expertise" : 1
            },
            "name" : "Expertise_1"
        }
    ]
```

5. Rebuild Indexes

```
    > 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" : 1,
		"Age" : -1
	},
	"name" : "Name_1_Age_-1"
},
{
	"v" : 2,
	"key" : {
		"Expertise" : 1
	},
	"name" : "Expertise_1"
}
```

```
        ],
        "ok" : 1
}
```

6. Drop Index on Remove Specific Index

```
> db.Employee.dropIndex({Salary:1})
{ "nIndexesWas" : 5, "ok" : 1 }
```

7. Remove All Indexes except for the _id index from a collection

```
> db.Employee.dropIndexes()
{
        "nIndexesWas" : 4,
        "msg" : "non-_id indexes dropped for collection",
        "ok" : 1
}
```