**Assignment 09**

1. Consider each document in the zipcode collection has the following form:

{

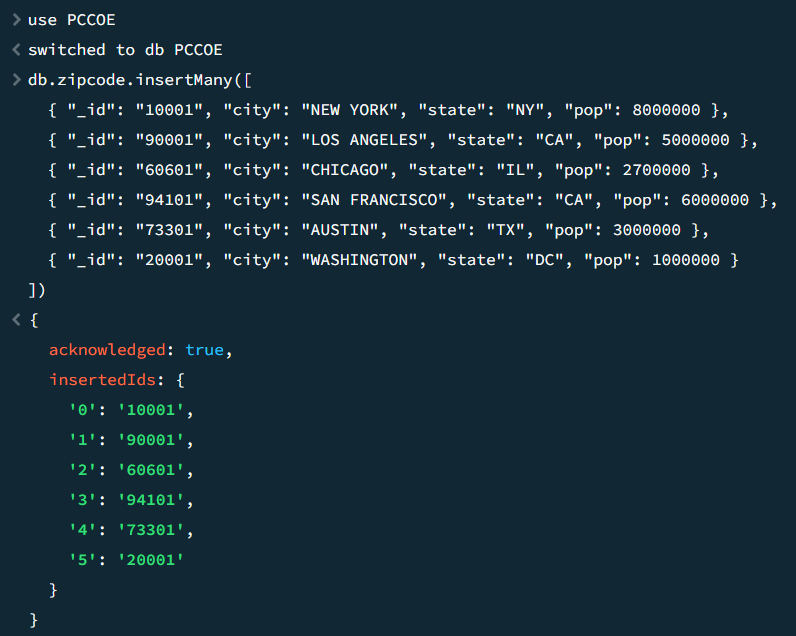
"\_id": "10180",

"city": "NEW YORK",

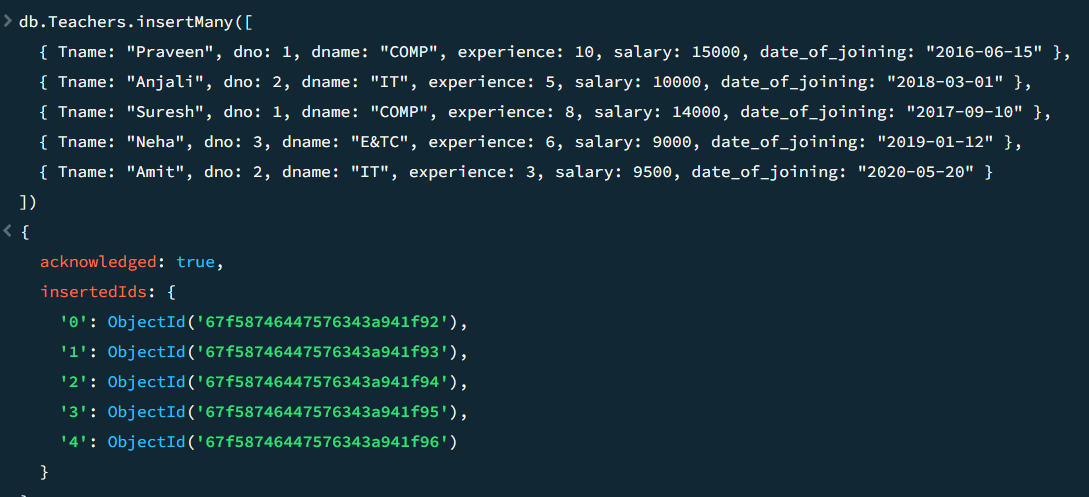
"state": "NY", "pop": 5574,

}

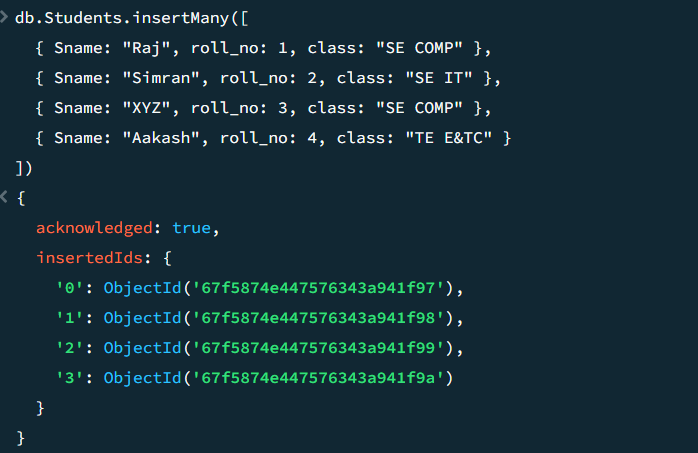
Return States with Populations above 10 Million



* **Insert sample data into Teachers collection**



* **Insert data in Students collection**

****

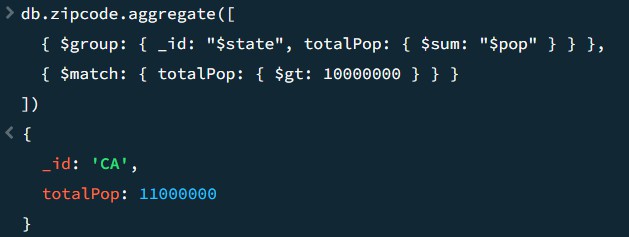
**Questions:**

1. Return States with Populations above 10 Million

Db.zipcode.aggregate([

{ $group: { \_id: “$state”, totalPop: { $sum: “$pop” } } },

{ $match: { totalPop: { $gt: 10000000 } } }

] )

1. Display the department wise average salary

{ $group: {\_id: “$dname”, avgSalary: {$avg: “$salary”}}}

] )



1. display the no. Of employees working in each department

{ $group: {\_id: “$dname”, empcount: {$sum: 1 }}}

])



1. Display the department wise total salary of departments having total salary greater than or equals to

5000/-

{ $group: {\_id: “$dname”, totalSalary: {$sum: “,salary”}}},

{ $match: { totalSalary: { $gte: 5000 } } }

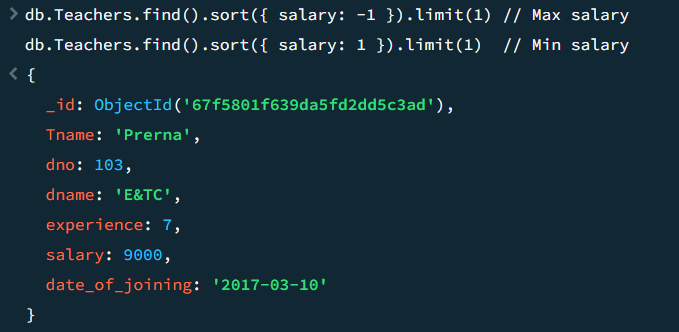
] )



1. Write the queries using the different operators like max, min. Etc.

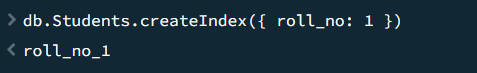
db.Teachers.fin().sort({ salary: -1 }).limit(1)

db.Teachers.fin().sort({ salary: 1 }).limit(1)



1. Create the simple index on roll\_no field

db.Students.createIndex ({roll\_no: 1 })



1. create unique index on any field for above given collections

db.Teachers.createIndex({ Tname: 1 }, { unique: true })



1. create compound index on any fields for above given collections

db.Teachers.createIndex({ dname: 1 , salary: -1 })



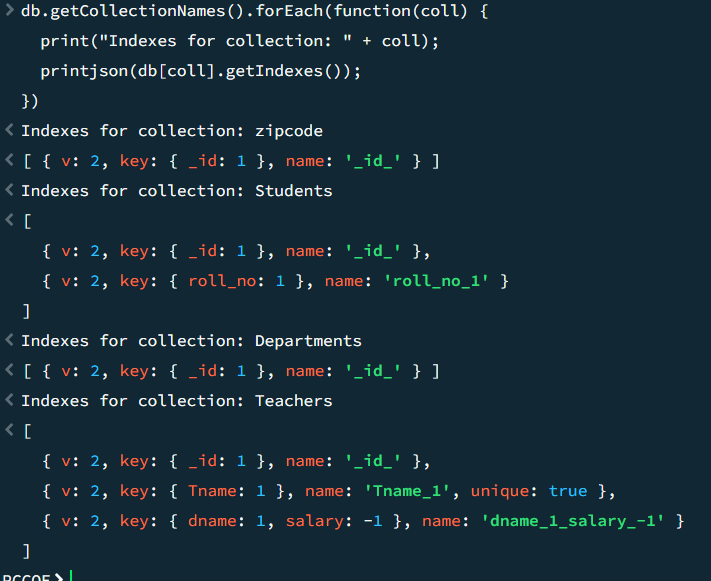
1. Show all the indexes created in the database PCCOE

db.getCollectionNames().forEach(function(coll) {

print(“Indexes for collection: “ + coll);

printjson(db[coll].getIndexes());

})



1. Show all the indexes created in above collections.

db.Teachers.getIndexes()

db.Students.getIndexes()

