Assignment -1

1. db.movies.find()
2. db.movies.find({writer: "Quentin Tarantino"})
3. db.movies.find({actors: "Brad Pitt"})
4. db.movies.find({franchise: "The Hobbit"})
5. db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]})
6. db.movies.find({$or: [{year: {$lt: 2000}}, {year: {$gt: 2010}}]})
7. db.movies.update({title: "The Hobbit: An Unexpected Journey"}, {synopsis: "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."})
8. db.movies.update({title: "The Hobbit: The Desolation of Smaug"}, {synopsis: "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."})
9. db.movies.update({title: "Pulp Fiction"}, {$push: {actors: "Samuel L. Jackson"}})
10. db.movies.find({$text: {$search: "Bilbo"}})
11. db.movies.find({$text: {$search: "Gandalf"}})
12. db.movies.find({$text: {$search: "Bilbo -Gandalf"}})
13. db.movies.find({$text: {$search: "dwarves hobbit"}})
14. db.movies.find({$text: {$search: "gold dragon"}})

1. db.movies.remove({title: "Pee Wee Herman's Big Adventure"})
2. db.movies.remove({title: "Avatar"})
3. db.users.find()
4. db.posts.find()
5. db.posts.find({username: "GoodGuyGreg"})
6. db.posts.find({username: "ScumbagSteve"})
7. db.comments.find()
8. db.comments.find({username: "GoodGuyGreg"})
9. db.comments.find({username: "ScumbagSteve"})

Assignment 2

1)db.zipcodes.find({ $and: [{city :"ATLANTA"},{state : "GA"}]})

2) db.zipcodes.aggregate([

{$match:{city:"ATLANTA"}},{$match:{state:"GA"}}])

3) db.zipcodes.aggregate([ {$group:{\_id:"$city"}},{$count:"zipcodes"}])

4) db.zipcodes.aggregate ( {$match:{city:'ATLANTA'}},{$group: { \_id: '$city', population: {$sum: '$pop'} } } )

1) db.zipcodes.aggregate ( {$group: { \_id: '$state', population: {$sum: '$pop'} } } )

2) db.zipcodes.aggregate ( {$group: { \_id: '$state', population: {$sum: '$pop'} } } ,{$sort:{population:-1}})

3) db.zipcodes.aggregate ( {$group: { \_id: '$state', population: {$sum: '$pop'} } } ,{$sort:{population:-1}},{$limit:3})

1)db.zipcodes.aggregate ( {$group: { \_id: '$city', population: {$sum: '$pop'} } } )

2) db.zipcodes.aggregate ( {$group: { \_id: '$city', population: {$sum: '$pop'} } } ,{$sort:{population:-1}})

3)db.zipcodes.aggregate ( {$group: { \_id: '$city', population: {$sum: '$pop'} } } ,{$sort:{population:-1}},{$limit:3})

4) db.zipcodes.aggregate ( {$match:{state:"TX"}},{$group: { \_id: {city: '$city', state: '$state'}, population: {$sum: '$pop'} } } ,{$sort:{population:-1}},{$limit:3})

1. db.zipcodes.aggregate ( {$group: { \_id: {city:'$city',state:'$state'}, population: {$avg: '$pop'} } },{$sort:{population:-1}} )
2. db.zipcodes.aggregate ( {$group: { \_id: '$state', population: {$avg: '$pop'} }},{$sort:{population:-1}} ,{$limit:3} )

ASSIGNMENT 3

1. db.addresses.find()
2. db.addresses.aggregate([ {$project:{name:1,borough:1,cuisine:1}}])
3. db.addresses.aggregate([ {$project:{\_id:0,name:1,borough:1,cuisine:1}}])
4. db.addresses.aggregate([ {$project:{\_id:0,name:1,borough:1,cuisine:1,"address.zipcode":1}}])
5. db.addresses.aggregate([ {$match:{borough:"Bronx"}},{$sort:{name:1}},{$limit:5},{$project:{borough:1,name:1}}])
6. db.addresses.aggregate([ {$match:{borough:"Bronx"}},{$project:{borough:1,name:1}}])
7. db.addresses.aggregate([ {$match:{borough:"Bronx"}},{$sort:{name:1}},{$skip:5},{$limit:5},{$project:{borough:1,name:1}}])
8. db.addresses.find({"grades.score":{$gto:90}}).pretty()
9. db.addresses.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}}).pretty()
10. db.restaurants.find({ 'address.coord.0': { $lt: -95.754168 } })
11. db.addresses.find({$and:[ {'address.coord.0': { $lt: - 65.754168 }},{cuisine:{$ne:"American "}},{grades : { $elemMatch:{"score":{$gt : 70}}}}]}).pretty()
12. db.addresses.find({$and:[ {'address.coord.1': { $lt: - 65.754168 }},{cuisine:{$ne:"American "}},{grades : { $elemMatch:{"score":{$gt : 70}}}}]}).pretty()
13. db.addresses.find({$and:[ {"grades.grade": "A" },{cuisine:{$ne:"American "}},{borough:{$ne:"Brooklyn"}}]}).sort({cuisine:-1}).pretty()
14. db.addresses.find( {name: /^Wil/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } )
15. db.addresses.find( {name: /ces$/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } )
16. db.addresses.find( {name: /.\*Reg.\*/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } )
17. db.addresses.find({borough:"Bronx" , $or:[ {cuisine:"American "},{cuisine:"Chinese"}]},{"name":1,"\_id":0,"cuisine":1}).sort({cuisine:-1}) .pretty()
18. db.addresses.find({$or:[{borough:"Staten Island"},{borough:"Queens"},{borough:"Bronxor Brooklyn"}]},{"name":1,"\_id":1,"cuisine":1,"borough":1}).sort({cuisine:-1}) .pretty()
19. db.addresses.find({$nor:[{borough:"Staten Island"},{borough:"Queens"},{borough:"Bronxor Brooklyn"}]},{"name":1,\_id:1,"borough":1,”cuisine”:1}).sort({cuisine:-1}) .pretty()
20. db.addresses.find({"grades.score":{$lte:10}},{"name":1,"\_id":1,"cuisine":1,"borough":1,"grades.score":1}).pretty()
21. db.addresses.find({$and:[{name: /^Wil/},{$nor:[ {cuisine:"American "},{cuisine:"Chinese"}]}]}, {"borough":1, "\_id" : 1,"name":1, "cuisine" :1 } )
22. db.addresses.find({"grades.date": ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score" : 11},{"restaurant\_id" : 1,"name":1,"grades":1}).pretty()
23. db.addresses.find({"grades.date": ISODate("2014-08-11T00:00:00Z"),"grades.1.grade":"A","grades.score" : 9},{"restaurant\_id" : 1,"name":1,"grades":1}).pretty()
24. db.addresses.find({"address.coord.1":{$gt:42,$lte:52}},{"restaurant\_id" : 1,"name":1,"address":1,"coord":1}).pretty()
25. db.addresses.find().sort({"name":1}).pretty()
26. db.addresses.find().sort({"name":-1}).pretty()
27. db.addresses.find().sort({"cuisine":1,"borough" : -1,}).pretty()
28. db.addresses.find({"address.street":{$exists:true}}).pretty()
29. db.addresses.find({"address.coord" :{$type : 1}})
30. db.addresses.find({"grades.score" :{$mod : [7,0]}},{"restaurant\_id" : 1,"name":1,"grades":1}).pretty();
31. db.addresses.find( {name: /.\*mon.\*/}, { "address.coord" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()
32. db.addresses.find( {name: /^Mad/}, { "address.coord" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()