**MongoDB – Complex Queries Assignment -3**

**Exercise Questions: -**

1. db.addresses.find().pretty()
2. db.addresses.aggregate([{$project: {restaurant\_id:1, name:1, borough:1, cuisine:1 }}]).pretty()
3. db.addresses.aggregate([{$project: {restaurant\_id:1, name:1, borough:1, cuisine:1, \_id:0 }}]).pretty()
4. db.addresses.aggregate([{$project: {restaurant\_id:1, name:1, borough:1, cuisine:1, "address.zipcode": 1, \_id:0 }}]).pretty()
5. db.addresses.find({"borough":"Bronx"}).limit(5).pretty()
6. db.addresses.find({"borough":"Bronx"}).pretty()
7. db.addresses.find({"borough":"Bronx"}).skip(5).limit(5).pretty()
8. db.addresses.find({"grades.score":{$gt:90}}).pretty()
9. db.addresses.find({"grades.score":{$gt:80, $lt:100}}).pretty()
10. db.addresses.find({"address.coord":{$lt : -95.754168}}).pretty()
11. db.addresses.find({ "cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70},"address.coord" : {$lt : -65.754168}}).pretty()
12. db.addresses.find({"cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70}, "address.coord" : {$lt : -65.754168}}).pretty()
13. db.addresses.find({ "cuisine" : {$ne : "American "}, "grades.grade" :"A", "borough": {$ne : "Brooklyn"}}).sort({"cuisine":-1}).pretty()
14. db.addresses.find( {name: /^Wil/}, {"restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1}).pretty()
15. db.addresses.find( {name: /ces$/}, {"restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1}).pretty()
16. db.addresses.find( {name: /.\*Reg\*./}, {"restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1}).pretty()
17. db.addresses.find({ "borough": "Bronx", $or: [ { "cuisine" : "American " }, { "cuisine" : "Chinese" }]}).pretty()
18. db.addresses.find( {"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn" ]}}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1}).pretty()
19. db.addresses.find( {"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn" ]}}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1}).pretty()
20. db.addresses.find({"grades.score": {$not: {$gt: 10}}},{"restaurant\_id": 1, "name": 1, "borough": 1, "cuisine": 1}).pretty()
21. db.addresses.find({$or: [{name: /^Wil/}, {"$and": [{"cuisine": {$ne :"American "}}, {"cuisine" : {$ne :"Chinees"}}]}]}, {"restaurant\_id" : 1, "name":1, "borough":1, "cuisine" :1 }).pretty()
22. db.addresses.find({ "grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade": "A" , "grades.score" : 11}, {"restaurant\_id" : 1, "name":1, "grades":1 }).pretty()
23. db.addresses.find({ "grades.1.date": ISODate("2014-08-11T00:00:00Z"), "grades.1.grade": "A" , "grades.1.score" : 9}, {"restaurant\_id" : 1, "name":1, "grades":1}).pretty()
24. db.restaurants.find({ "address.coord.1": {$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" : {$type : 1}}).pretty()
30. db.addresses.find({"grades.score" : {$mod : [7,0]}}, {"restaurant\_id" : 1,"name": 1, "grades": 1}).pretty()
31. db.addresses.find({ name : { $regex : "mon.\*"}}, { "name":1, "borough": 1, "address.coord":1, "cuisine" :1}).pretty()
32. db.addresses.find({name : { $regex : /^Mad/}}, {"name":1, "borough":1, "address.coord":1, "cuisine" :1 }).pretty()