MongoDB – Complex Queries

MongoDB Exercises – With the Restaurants Data Set

1. Downloaded the file
2. Unzipped the file
3. Running server
4. Importing json file
5. Mongod, mongo
6. Show dbs
7. Use restaurants
8. Db.addresses.find().pretty()

Exercise Questions

1. Db.addresses.find()
2. Db.addresses.find(_id:1, name:1,cuisine:1, borough:1, restaurant_id:1)
3. Db.addresses.find(_id:0, name:1, cuisine:1, borough:1, restaurant_id:1)
4. db.addresses.find({}, {_id:0, name:1, cuisine: 1, borough: 1, restaurant_id: 1, "address.zipcode": 1})
5. db.addresses.find({"borough":"Bronx"}).pretty().limit(5)
6. db.addresses.find({"borough":"Bronx"}).pretty()
7. db.addresses.find({"borough":"Bronx"}).pretty().skip(5).limit(5)
8. db.addresses.find({"grades.score":{$gt:90}})
9. db.addresses.find({$and: [{"grades.score":{$gt:80}},{"grades.score":{$lt:100}}]})
10. db.addresses.find({"address.coord.0":{$lt:-95.754168}})
11. db.addresses.find ({$and:[{"cuisine":{$ne:"American"}},{"address.coord.0":{$lt:-65.754168}},{"grades.score":{$gt:70}}]}).pretty()
12. db.addresses.find({$and : [{"cuisine" : {$ne : "American "}}, {"address.coord.1" : {$lt : -65.754168}}, {"grades.score" : {$gt : 70}}]})
13. db.addresses.find({$and:[{"cuisine":{$ne:"American"}}, {"grades.grade":"A"}, {"borough":{$ne: "Brooklyn"}}]}).sort({cuisine:-1}).pretty()
14. db.addresses.find({"name" : { $regex: /^Wil.*/}}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
15. db.addresses.find({"name" : { $regex: /.*ces$/}}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
16. db.addresses.find({"name" : { $regex: /Reg/}}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
17. db.addresses.find({borough: "Bronx", cuisine: {$in: ["American ","Chinese"]}}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
18. db.addresses.find({$or: [{"borough": "Staten Island"}, {"borough": "Bronxor Brooklyn"}, {"borough": "Queens"}]}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
19. db.addresses.find({borough: {$nin: ["Staten Island","Queens","Bronx","Brooklyn"]}} , {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})

20. db.addresses.find({"grades.score": {$lte: 10}}, {_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
21. db.addresses.find({$nor: [{cuisine: {$in: ["American ","Chinese"]}},{name: /^Wil.*/}]},{_id:0, restaurant_id:1, name:1, borough:1, cuisine:1})
22. db.addresses.find({"grades" : {$elemMatch: {"date": ISODate("2014-08-11T00:00:00Z"), "grade":"A", "score":11}}}, {_id:0, restaurant_id:1, name:1, grades:1})
23. db.addresses.find({$and: [{"grades.1.grade":"A"}, {"grades.1.score": 9}, {"grades.1.date": ISODate("2014-08-11T00:00:00Z")}]},{_id:0, restaurant_id:1, name:1, grades:1}).pretty()
24. db.addresses.find({$and : [{"address.coord.1": {$gt : 42}},{"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
25. db.addresses.find({},{_id:0, name:1}).sort({name: 1})
26. db.addresses.find({},{_id:0, name:1}).sort({name: -1})
27. db.addresses.find({}, {_id:0, cuisine:1, borough:1}).sort({cuisine: 1, borough: -1})
28. db.addresses.find({“address.street”:{$regex:/Street/}}).pretty() – With Street
    db.addresses.find({“address.street”:{$ne:{$regex:/Street/}}}).pretty() – Without Street.
29. db.addresses.find({"address.coord": {$type: "double"}}, {_id:0, address:1})
30. db.addresses.find({"grades": {$elemMatch: {"score": {$mod: [7,0]}}}},{_id:0, restaurant_id:1, name:1, grades:1})
31. db.addresses.find({name: {$regex: /mon/}},{_id:0, name:1, borough:1, "address.coord":1, cuisine:1})
32. db.addresses.find({name: {$regex: /^Mad.*/}},{_id:0, name:1, borough:1, "address.coord":1, cuisine:1})