**Assignment No3**

MongoDB Complex Queries

1)

db.addresses.find()

2)

db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

3)

db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1,"\_id":0});

4)

db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"\_id":0});

5)

db.addresses.find({"borough": "Bronx"}).limit(5);

6)

db.addresses.find({"borough": "Bronx"});

7)

db.addresses.find({"borough": "Bronx"}).skip(5).limit(5);

8)

db.addresses.find({grades : { $elemMatch:{"score":{$gt : 90}}}});

9)

db.addresses.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});

10)

db.addresses.find({"address.coord" : {$lt : -95.754168}});

11)

db.addresses.find(

{$and:

[

{"cuisine" : {$ne :"American "}},

{"grades.score" : {$gt : 70}},

{"address.coord" : {$lt : -65.754168}}

]

}

);

12)

db.addresses.find(

{

"cuisine" : {$ne : "American "},

"grades.score" :{$gt: 70},

"address.coord" : {$lt : -65.754168}

}

);

13)

db.addresses.find( {

"cuisine" : {$ne : "American "},

"grades.grade" :"A",

"borough": {$ne : "Brooklyn"}

}

).sort({"cuisine":-1});

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" }

]

}

);

18)

db.addresses.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

19)

db.addresses.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

20)

db.addresses.find(

{"grades.score" :

{ $not:

{$gt : 10}

}

},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

21)

db.addresses.find(

{$or: [

{name: /^Wil/},

{"$and": [

{"cuisine" : {$ne :"American "}},

{"cuisine" : {$ne :"Chinees"}}

]}

]}

,{"restaurant\_id" : 1,"name":1,"borough":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}

);

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}

);

24)

db.addresses.find(

{

"address.coord.1": {$gt : 42, $lte : 52}

},

{"restaurant\_id" : 1,"name":1,"address":1,"coord":1}

);

25)

db.addresses.find().sort({"name":1});

26)

db.addresses.find().sort(

{"name":-1}

);

27)

db.addresses.find().sort(

{"cuisine":1,"borough" : -1,}

);

28)

db.addresses.find(

{"address.street" :

{ $exists : true }

}

);

29)

db.addresses.find(

{"address.coord" :

{$type : 1}

}

);

30)

db.addressess.find(

{"grades.score" :

{$mod : [7,0]}

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);

31)

db.addresses.find(

{ name :

{ $regex : "mon.\*", $options: "i" }

},

{

"name":1,

"borough":1,

"address.coord":1,

"cuisine" :1

}

);

32)

db.addresses.find(

{ name :

{ $regex : /^Mad/i, }

},

{

"name":1,

"borough":1,

"address.coord":1,

"cuisine" :1

}

);

Screenshots





























