{

"address": {

"building": "1007",

"coord": [ -73.856077, 40.848447 ],

"street": "Morris Park Ave",

"zipcode": "10462"

},

"borough": "Bronx",

"cuisine": "Bakery",

"grades": [

{ "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },

{ "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },

{ "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },

{ "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },

{ "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }

],

"name": "Morris Park Bake Shop",

"restaurant\_id": "30075445"

}

PFA the above sample data in “restaurants.json” attached with this document.

Import it in mongo test database

> mongoimport restaurants.json

1. Write a MongoDB query to display all the documents in the collection restaurants.

db.restaurants.find().pretty()

2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.

db.restaurants.aggregate({$project:{restaurant\_id:1,name:1,borough:1,cuisine:1}}).pretty()

3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.

db.restaurants.aggregate({$project:{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1}}).pretty()

4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.

db.restaurants.aggregate({$project:{\_id:0,restaurant\_id:1,name:1,borough:1,'address.zipcode':1}}).pretty()

5. Write a MongoDB query to display all the restaurant which is in the borough Bronx.

db.restaurants.find({borough:'Bronx'}).pretty()

6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

db.restaurants.find({borough:'Bronx'}).pretty().limit(5)

7.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

db.restaurants.find({borough:'Bronx'}).pretty().skip (5)

8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

db.restaurants.find({'grades.score':{$gt:90}}).pretty()

9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

db.restaurants.find({'grades.score':{'$in':[80,10]}}).pretty()

10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.

db.restaurants.find({'address.coord':{'$elemMatch':{'$lt':-95.754168}}}).pretty()

11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

db.restaurants.find({$and:[{$and:[{'address.coord':{$elemMatch:{$lt:-65.754168}}},{'grades.score':{$gt:70}}]},{cuisine:{$ne:'American'}}]}).pretty()

12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

Note : Do this query without using $and operator.

db.restaurants.find({"cuisine":{$ne:"American"},"grades.score":{$gt:70},"address.coord.0":{$lt:-65.754168}}).pretty();

13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order. db.restaurants.find({$and:[{"cuisine":{$ne:"American"}},{"grades.grade":"A"},{"borough":{$ne:"Brooklyn"}}]}).sort({cuisine:-1}).pretty()

14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

db.restaurants.find({name:/^Wil/},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();

15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

db.restaurants.find({name:/ces$/},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();

16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

db.restaurants.find({name:/.Reg./},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();

17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

db.restaurants.find({$and:[{"borough":"Bronx"},{$or:[{"cuisine":"American"},{"cuisine":"Chinese"}]}]}).pretty();

18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

db.restaurants.find({$or:[{"borough":"Staten Island"},{"borough":"Queens"},{"borough":"Bronx"},{"borough":"Brooklyn"}]},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();

19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

db.restaurants.find({$and:[{"borough":{$ne:"Staten Island"}},{"borough":{$ne:"Queens"}},{"borough":{$ne:"Bronx"}},{"borough":{$ne:"Brooklyn"}}]},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();

20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

db.restaurants.find({‘grades.score’:{$ne:10}},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).pretty();