Practical -1

1. Display details of employee staying in mulund

db.records.find({"city":"mulund"})

1. Display details of employee staying in dadar and salary greater than 10000.

db.records.find({"city":"dadar","sal":{$gt:"10000"}})

1. Display details of employee whose salary is  greater than equal 12000 but less than 18000

db.records.find({"sal":{$gte:"12000",$lt:"18000"}})

1. Display details of employees staying in either thane or mulund and salary greater than 14000

db.records.find({"sal":{$gt:"14000"}, $or:[{"city":"mulund"},{"address":"thane"}] })

1. Display details of employee whose salary is not equal to 14000 and working in either testing or sales department.

db.records.find({$or:[{"dname":"testing"},{"dname":"sales"}],"sal":{$ne:"14000"}})

Practical 2 A

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

db.rest.find().pretty()

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

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

1. 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 rest

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

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

1. Write a MongoDB query to display all the rest which is in the borough Bronx

db.rest.find({"borough": "Bronx"}).pretty()

1. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.Ans:- db.rest.find({"borough": "Bronx"}).limit(5).pretty

db.rest.find({"borough": "Bronx"}).limit(5).pretty()

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

db.rest.find({"borough":"Bronx"}).skip(5).limit(5).pretty()

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

db.rest.find({grades : {$elemMatch:{"score":{$gt : 90}}}}).pretty()

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

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

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

db.rest.find({"address.coord" : {$lt : -95.754168}}).pretty()

1. 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.rest.find( {$and: [{"cuisine" : {$ne :"American "}}, {"grades.score" : {$gt : 70}},{"address.coord" : {$lt : -65.754168}} ] } ).pretty()

1. 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.rest.find({"cuisine":{$ne : "American "}, "grades.score" :{$gt: 70},"address.coord" : {$lt : -65.754168}}).pretty()

1. 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.rest.find( {"cuisine" : {$ne : "American "}, "grades.grade" :"A", "borough": "Brooklyn" }).sort({"cuisine":-1}).pretty()

1. 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.rest.find( {name: /^Wil/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()

1. 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.rest.find({name: /ces$/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()

1. 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.rest.find( {"name": /.\*Reg.\*/}, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()

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

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

1. 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 Bronx or Brooklyn

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

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

1. . 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.rest.find( {"grades.score" : { $not: {$gt : 10} } }, { "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'

db.rest.find( {$or: [ {name: /^Wil/}, {"$and":[ {"cuisine" : {$ne :"American "}}, {"cuisine" :{$ne :"Chinees"}} ]} ]} ,{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1} ).pretty()

1. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014- 08-11T00:00:00Z" among many of survey dates.

db.rest.find({"grades.date": ISODate("2014-0811T00:00:00Z"),"grades.grade":"A" ,"grades.score" : 11 },{"restaurant\_id" : 1,"name":1,"grades":1}).pretty()

1. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

db.rest.find({ "grades.1.date": ISODate("2014-0811T00:00:00Z"),"grades.1.grade":"A" ,"grades.1.score" : 9},{"restaurant\_id" : 1,"name":1,"grades":1} ).pretty()

1. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52..

db.rest.find({"address.coord.1": {$gt : 42, $lte : 52}},{"restaurant\_id" : 1,"name":1,"address":1,"coord":1}).pretty()

1. . Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns

db.rest.find().sort({"name":1}).pretty()

1. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns

db.rest.find().sort( {"name":-1} ).pretty()

1. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order

db.rest.find().sort({"cuisine":1,"borough" : -1,} ).pretty()

1. Write a MongoDB query to know whether all the addresses contains the street or not.

db.rest.find({"address.street" :{ $exists :true } } ).pretty()

1. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double

db.rest.find( {"address.coord" :{$type :1}}).pretty()

1. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

db.rest.find( {"grades.score" :{$mod : [7,0]}},{"restaurant\_id" : 1,"name":1,"grades":1} ).pretty()

1. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name

db.rest.find( { name : { $regex : "mon.\*", $options: "i" }},{ "name":1,"borough":1,"address.coord":1,"cuisine":1}).pretty()

1. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name

db.rest.find( { name :{ $regex : /^Mad/i, }},{"name":1,"borough":1,"address.coord":1,"cuisine" :1}).pretty()

Practical 2B

1. Update the country to USA for all female users

Ans:- db.users.update({"Gender": "F"}, {$set:{"Country":"USA"}}, {multi:true})

1. Add the new filed company to all the documents

Ans:- db.users.update({},{$set:{"Company":"AP Enterprises"}},{multi:true})

1. Delete all the documents where Gender=’M ‘

Ans:- db.users.remove({"Gender":"M"})

1. Find out a count of female users who stay in either India or USA

Ans:- db.users.find({"Gender":"F",$or:[{"Country":"India"},{"Country":"USA"}]}).count()

1. Display the first name and age of all female employees

Ans:- db.users.find({"Gender":"F"}, {"FName":1,"Age":1})