

Assignmnet_1

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

Ans: `db.restaurant.find().pretty()`

```
restaurant_id: 40350000 }
> db.restaurants.find().limit(2)
< { _id: ObjectId("63504c6c6e6934db446a5723"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
      { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],
  name: 'Morris Park Bake Shop',
```

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

Ans: `db.restaurant.find({}, {restaurant_id:1,name:1,borough:1,cuisine:1})`

```

{ _id: ObjectId("63504c6c6e6934db446a572a"),
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  name: 'Wilken\'S Fine Food',
  restaurant_id: '40356483' }
{ _id: ObjectId("63504c6c6e6934db446a572b"),
  borough: 'Brooklyn',
  cuisine: 'American ',
  name: 'Regina Caterers',
  restaurant_id: '40356649' }

```

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.

Ans: db.restaurant.find({}, {restaurant_id:1,name:1,_id:0,borough:1,cuisine:1})

```

> db.restaurants.find({}, {_id:0,restaurant_id:1, name:1, borough:1, cuisine:1})
< { borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
{ borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: 'Wendy\'S',
  restaurant_id: '30112340' }

```

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.

Ans: db.restaurant.find({}, {address: {zipcode:1},restaurant_id:1,name:1,_id:0})

```
> db.restaurants.find({}, {_id:0,address:{zipcode:1},restaurant_id:1, name:1}).
< { address: { zipcode: '10462' },
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445' }
{ address: { zipcode: '11225' },
  name: 'Wendy\'S',
  restaurant_id: '30112340' }
Practice\
```

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

Ans: db.restaurant.find({borough:'Bronx'})

```
> db.restaurants.find({borough:"Bronx"}).limit(1)
< { _id: ObjectId("63504c6c6e6934db446a5723"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
      { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445' }
```

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

Ans: db.restaurant.find({borough:'Bronx'}).limit(5)

```

restaurant_id: '30075445' }
> db.restaurants.find({borough:"Bronx"}).limit(5)
< { _id: ObjectId("63504c6c6e6934db446a5723"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
      { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445' }

```

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

Ans: `db.restaurant.find({borough:'Bronx'}).limit(5).skip(5)`

```

restaurant_id: '40364363' }
> db.restaurants.find({borough:"Bronx"}).skip(5).limit(5)
< { _id: ObjectId("63504c6c6e6934db446a5760"),
  address:
    { building: '658',
      coord: [ -73.81363999999999, 40.829411000000001 ],
      street: 'Clarence Ave',
      zipcode: '10465' },
  borough: 'Bronx',
  cuisine: 'American ',
  grades:
    [ { date: 2014-06-21T00:00:00.000Z, grade: 'A', score: 5 },
      { date: 2012-07-11T00:00:00.000Z, grade: 'A', score: 10 } ],
  name: 'Manhem Club',
  restaurant_id: '40364363' }
{ _id: ObjectId("63504c6c6e6934db446a5778"),
  address:
    { building: '2222',
      coord: [ -73.84071750000000, 40.8204011 ]

```

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

Ans: db.restaurant.find({'grades.score':{\$gt:90}}).pretty()

```

> db.restaurants.find({"grades.score":{$gt:90}})
< { _id: ObjectId("63504c6c6e6934db446a5881"),
  address:
    { building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019' },
  borough: 'Manhattan',
  cuisine: 'American ',
  grades:
    [ { date: 2014-08-22T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2014-03-28T00:00:00.000Z, grade: 'C', score: 131 },
      { date: 2013-09-25T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2013-04-08T00:00:00.000Z, grade: 'B', score: 25 },
      { date: 2012-10-15T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2011-10-19T00:00:00.000Z, grade: 'A', score: 13 } ],
  name: 'Murals On 54/Randolphs\'S',

```

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

Ans: `db.restaurant.find({$and:[{'grades.score':{$gt:90}},{'grades.score':{$lt:100}}]}).pretty()`

```

> db.restaurants.find({"grades.score": {$gt:90, $lt:100}})
< { _id: ObjectId("63504c6c6e6934db446a5881"),
  address:
    { building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019' },
  borough: 'Manhattan',
  cuisine: 'American ',
  grades:
    [ { date: 2014-08-22T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2014-03-28T00:00:00.000Z, grade: 'C', score: 131 },
      { date: 2013-09-25T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2013-04-08T00:00:00.000Z, grade: 'B', score: 25 },
      { date: 2012-10-15T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2011-10-19T00:00:00.000Z, grade: 'A', score: 13 } ],
  name: 'Murals On 54/Randolphs\'S',

```

Assignmnet_2

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

Ans: db.restaurant.find({'address.coord.0': {\$lt:-95.754168}}).pretty()

```
> db.restaurants.find({"address.coord.0":{"$lt:-95.75}})
< { _id: ObjectId("63504c6d6e6934db446a5d6b"),
  address:
    { building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372' },
  borough: 'Queens',
  cuisine: 'American ',
  grades:
    [ { date: 2014-06-04T00:00:00.000Z, grade: 'A', score: 12 },
      { date: 2013-11-07T00:00:00.000Z, grade: 'B', score: 19 },
      { date: 2013-05-17T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2012-08-29T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2012-04-03T00:00:00.000Z, grade: 'A', score: 12 },
      { date: 2011-11-16T00:00:00.000Z, grade: 'A', score: 7 } ],
  name: 'Burger King',
```

2. 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.

Ans: db.restaurant.find({\$and:[{cuisine:{\$ne:'American'}},{address.coord.0:{\$lt:-65.754168}}]}).pretty()

```
> db.restaurants.find({cuisine:{$ne:"American"}, "address.coord.0":{"$lt:-65.75}})
< { _id: ObjectId("63504c6c6e6934db446a5723"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
```


3. 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.

Ans:

```
> db.restaurants.find({cuisine:{$ne:"American"}, "address.coord.0":{$lt:-65.75}})
< { _id: ObjectId("63504c6c6e6934db446a5723"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
```

4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those

db.restaurant.find({name:/^Wil/},{restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()

```
> db.restaurants.find({name:{$regex:"^Wil"}},{restaurant_id:1,name:1, borough:1, cuisine:1})
< { _id: ObjectId("63504c6c6e6934db446a572a"),
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  name: 'Wilken\'S Fine Food',
  restaurant_id: '40356483' }
{ _id: ObjectId("63504c6c6e6934db446a572d"),
  borough: 'Bronx',
  cuisine: 'American ',
  name: 'Wild Asia',
  restaurant_id: '40357217' }
{ _id: ObjectId("63504c6d6e6934db446a6532"),
  borough: 'Bronx',
  cuisine: 'Pizza',
  name: 'Wilbel Pizza',
  restaurant_id: '40871979' }
```

5. 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.

Ans: db.restaurant.find({name:/ces\$/},{restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()

```
> db.restaurants.find({name:{$regex:"ces$"}},{restaurant_id:1,name:1, borough:1, cuisine:1})
< { _id: ObjectId("63504c6d6e6934db446a5bb6"),
  borough: 'Manhattan',
  cuisine: 'American ',
  name: 'Pieces',
  restaurant_id: '40399910' }
{ _id: ObjectId("63504c6d6e6934db446a5c75"),
  borough: 'Queens',
  cuisine: 'American ',
  name: 'S.M.R Restaurant Services',
  restaurant_id: '40403857' }
{ _id: ObjectId("63504c6d6e6934db446a5c7b"),
```

6. 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.

Ans: `db.restaurant.find({name:/.*/}, {restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()`

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

Ans: `db.restaurant.find({$and:[{borough:'Bronx'}],{$or:[{cuisine:'American'},{cuisine:'Chinese'}]}]).pretty()`

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

Ans: `db.restaurant.find().sort({name:1}).pretty()`

```
{ _id: ObjectId("63504c6d6e6934db446a63b9"),
  address:
    { building: '525',
      coord: [ -73.9728136, 40.75546629999999 ],
      street: 'Lexington Ave',
      zipcode: '10017' },
  borough: 'Manhattan',
  cuisine: 'American ',
  grades:
    [ { date: 2014-07-22T00:00:00.000Z, grade: 'C', score: 41 },
      { date: 2014-01-24T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2013-05-15T00:00:00.000Z, grade: 'B', score: 24 },
      { date: 2012-09-12T00:00:00.000Z, grade: 'A', score: 7 },
      { date: 2012-04-12T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2011-10-24T00:00:00.000Z, grade: 'C', score: 33 } ],
  name: '525 Lex Restaurant & Bar',
  restaurant_id: '40806489' }
Type "it" for more
practice> db.restaurants.find().sort({name:1})
```

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

Ans: `db.restaurant.find().sort({name:-1}).pretty()`

```
{ _id: ObjectId("63504c6c6e6934db446a5a7b"),
  address:
    { building: '404',
      coord: [ -73.97635439999999, 40.6438997 ],
      street: 'Church Avenue',
      zipcode: '11218' },
  borough: 'Brooklyn',
  cuisine: 'Chinese',
  grades:
    [ { date: 2014-10-29T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2014-03-24T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2013-10-25T00:00:00.000Z, grade: 'C', score: 3 },
      { date: 2012-10-01T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2012-05-09T00:00:00.000Z, grade: 'C', score: 2 } ],
  name: 'Yen Yen Restaurant',
  restaurant_id: '40393181' }
Type "it" for more
Practice > db.restaurants.find().sort({name:-1})
```

10. 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.

Ans: `db.restaurant.find().sort({name:1,borough:-1}).pretty()`

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

Ans: `db.restaurant.find({'address.street':{'$exists:true'}}).count()`

-
1. Write a MongoDB query to display all the documents in the collection restaurants.
 2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.
 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.
 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.
 5. Write a MongoDB query to display all the restaurant which is in the borough Bronx.
 6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.
 7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.
 8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

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

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

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.

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.

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.

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.

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.

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.

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

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 Bronx or Brooklyn.

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 Bronx or Brooklyn.

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.

21. 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'.

22. 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..

23. 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".

24. 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..

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

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

27. 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.

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

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

30. 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.

31. 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.

32. 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.