MongoDB Exercises

Q1. Write a MongoDB query to display all the documents in the collection restaurants.

Solution :-

```
>  db.restaurants.find().toArray()
```

Q2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.

Solution :-

```
> db.restaurants.find({},{"restaurant_id":1, "name":1,"borough":1, "cuisine":1} ).pretty()
```

Q**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.

Solution :-

```
> db.restaurants.find({},{"restaurant_id":1, "name":1, "borough":1, "cuisine":1, "_id":0} ).pretty()
```

Q4. 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.

Solution :-

```
>db.restaurants.find({},{"restaurant_id":1,"name":1,"borough":1,"address.zipcode":1, "_id":0} ).pretty()
```

Q5. Write a MongoDB query to display all the restaurant which is in the borough Bronx.

Solution :-

```
> db.restaurants.find({"borough":"Bronx"}).pretty()
```

Q6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

Solution :-

```
> db.restaurants.find({"borough":"Bronx"}).limit(5).pretty()
```

Q7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

Solution :-

```
> db.restaurants.find( {"borough":"Bronx"}
).skip(5).limit(5).pretty()
```

Q8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

Solution :-

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

Q9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

Solution :-

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

Q10. Write a MongoDB query to find the restaurants, which locate in latitude value less than -95.754168.

Solution :-

```
> db.restaurants.find({"address.coord.0": {$lt: -95.754168} } )
```

Q11. 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.

Solution :-

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

Q12. 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 thecuisine in descending order.

Solution :-

```
> db.restaurants.find( {$and: [ {"cuisine":{$ne:"American "}}, {"grades.0.grade": "A"}, {"address.coord.0":{$lt:-65.754168}} ] } ).pretty()
```

Q13. 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.

Solution :-

```
> db.restaurants.find({ name: /^Wil/ },{ "restaurant_id":1,"name":1,"borough":1, "cuisine":1 } ).pretty()
```

Q14. 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.

Solution :-

```
> db.restaurants.find({ "grades.0.score":{ $not:{$gt:10} } },{ "restaurant_id":1, "name":1, "borough":1, "cuisine":1 } ).pretty()
```

Q15. Write a MongoDB query to find the restaurant Id, name, borough andcuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

Solution :-

```
> db.restaurants.find({$or: [ {name: /^Wil/ },{"$and" : [ {"cuisine":{$ne:"American "} }, {"cuisine":{$ne:"American "} } ] } ] },{"restaurant_id":1, "name":1, "borough":1, "cuisine":1 } ).pretty()
```

Q16. 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".

Solution :-

```
> db.restaurants.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 } ).pretty()
```

Q17. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

Solution :-

```
> db.restaurants.find().sort( {"name":1} ).pretty()
```

Q18. Write a MongoDB query to know whether all the addresses contains the street or not.

Solution :-

```
> db.restaurants.find( {"address.street":{$exists:true} } ).pretty()
```

Q19. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants, which contains 'mon' as threeletters somewhere in its name.

Solution :-

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

Q20. 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.

Solution :-

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