Mongo Assignment 3

MongoDB – Complex Queries

Exercise Questions

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

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find()
[
  {
    _id: ObjectId("61815303eaaa17f1252daded"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
```

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

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({},{restaurant_id:1, name:1,borough:1, cuisine:1 })
[
  {
    _id: ObjectId("61815303eaaa17f1252daded"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61815303eaaa17f1252dadee"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
```

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.

```
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({}, {restaurant_id:1, _id:0, 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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({}, { restaurant_id: 1, _id: 0, name: 1, borough: 1, "address.zipcode": 1 })
[
  {
    address: { zipcode: '10462' },
    borough: 'Bronx',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    address: { zipcode: '11225' },
    borough: 'Brooklyn',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
```

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

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants>  db.addresses.find({"borough":"Bronx"}).limit(5)
[
  {
    _id: ObjectId("61815303eaaa17f1252daded"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
```

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

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"borough":"Bronx"})
[
  {
    _id: ObjectId("61815303eaaa17f1252daded"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
```

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

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"borough":"Bronx"}).limit(5).skip(5)
[
  {
    _id: ObjectId("61815303eaaa17f1252dae2a"),
    address: {
      building: '658',
      coord: [ -73.81363999999999, 40.82941100000001 ],
      street: 'Clarence Ave',
      zipcode: '10465'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
```

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

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{$gt:90}})
[
  {
    _id: ObjectId("61815303eaaa17f1252daf4b"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West   54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades: [
```

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

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{$gt:80, $lt:100}})
[
  {
    _id: ObjectId("61815303eaaa17f1252daf4b"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West   54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades: [
```

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

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"address.coord":{$lt : -95.754168}})
[
  {
    _id: ObjectId("61815303eaaa17f1252db435"),
    address: {
      building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372'
    },
    borough: 'Queens',
    cuisine: 'American ',
    grades: [
```

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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ $and: [{ "cuisine": { $ne: "American " } }, { "address.coord.0": { $lt: -65.754168 } }, { "grades.scor
e": { $gt: 70 } }] })
[
  {
    _id: ObjectId("61815303eaaa17f1252dafec"),
    address: {
      building: '345',
```

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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ $and: [{ "cuisine": { $ne: "American " } }, { "address.coord.1": { $lt: -65.754168 } }, { "grades.scor
e": { $gt: 70 } }] })
```

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.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ $and: [{ "cuisine": { $ne: "American " } }, { "grades.grade": "A" }, { "borough": { $ne: "Brooklyn " } }
 ] }).sort({ cuisine: -1 })
[
  {
    _id: ObjectId("61815303eaaa17f1252db4f9"),
    address: {
```

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.

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ "name": { $regex: /^Wil.*/ } }, { _id: 0, restaurant_id: 1, name: 1, borough: 1, cuisine: 1 })
[
  {
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  },
  {
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    borough: 'Bronx',
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
```

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.

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ "name": { $regex: /.*ces$/ } }, { _id: 0, restaurant_id: 1, name: 1, borough: 1, cuisine: 1 })
[
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Pieces',
    restaurant_id: '40399910'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'S.M.R Restaurant Services',
    restaurant_id: '40403857'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Good Shepherd Services',
    restaurant_id: '40403989'
  },
  {
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: "The Ice Box-Ralph'S Famous Italian Ices",
    restaurant_id: '40690899'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Alices',
    restaurant_id: '40782042'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Re: Sources',
    restaurant_id: '40876068'
  }
]
```

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.

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ "name": { $regex: /Reg/ } }, { _id: 0, restaurant_id: 1, name: 1, borough: 1, cuisine: 1 })
[
  {
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Café/Coffee/Tea',
    name: 'Caffe Reggio',
    restaurant_id: '40369418'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
```

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

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ borough: "Bronx", cuisine: { $in: ["American ", "Chinese"] } }, { _id: 0, restaurant_id: 1, name: 1, b
orough: 1, cuisine: 1 })
[
  {
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'Happy Garden',
    restaurant_id: '40363289'
  },
```

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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({$or: [{"borough": "Staten Island"}, {"borough": "BronxorBrooklyn"}, {"borough": "Queens"}]}, {_id:0, re
restaurant_id:1, name:1, borough:1, cuisine:1})
[
  {
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356068'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
```

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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find( {borough: {$nin: ["Staten Island","Queens","Bronx","Brooklyn"]}} , {_id:0, restaurant_id:1, name:1,
... borough:1, cuisine:1})
[
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
```

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.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ "grades.score": { $lte: 10 } }, { _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'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
```

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'.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ $nor: [{ cuisine: { $in: ["American ", "Chinese"] } }, { name: /^Wil.*/ }] }, { _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'
```

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.

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"grades" : {$elemMatch: {"date": ISODate("2014-08-11T00:00:00Z"), "grade":"A", "score":11}}}, {_id:0, r
restaurant_id:1, name:1,grades:1})
[
  {
    grades: [
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
```

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"

```
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({$and: [{"grades.1.grade":"A"}, {"grades.1.score": 9},{"grades.1.date": ISODate("2014-08-11T00:00:00Z")}
]},{_id:0, restaurant_id:1, name:1, grades:1})
[
  {
    grades: [
      {
        date: ISODate("2015-01-12T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
```

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..

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ $and: [{ "address.coord.1": { $gt: 42 } }, { "address.coord.1": { $lte: 52 } }] }, { _id: 0, restauran
t_id: 1, name: 1, address: 1 })
[
  {
    address: {
      building: '47',
      coord: [ -78.877224, 42.89546199999999 ],
      street: 'Broadway @ Trinity Pl',
      zipcode: '10006'
    },
```

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

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({},{_id:0, name:1}).sort( {name: 1})
[
  { name: '(Lewis Drug Store) Locanda Vini E Olii' },
  { name: '1 East 66Th Street Kitchen' },
  { name: '101 Deli' },
  { name: '101 Restaurant And Bar' },
  { name: '1020 Bar' },
  { name: '104-01 Foster Avenue Coffee Shop(Ups)' },
  { name: '10Th Avenue Pizza & Cafe' },
  { name: '111 Restaurant' },
  { name: '15 East Restaurant' },
  { name: '200 Fifth Avenue Restaurant & Sports Bar' },
  { name: '21 Club' },
  { name: '2A' },
  { name: '3 Deli & Grill' },
  { name: '3 Guys' },
  { name: '3 Guys Resturant' },
  { name: '42Nd Street Pizza Diner' },
  { name: "44 & X Hell'S Kitchen" },
  { name: '44 Sw Ristorante & Bar' },
  { name: '5 Burro Cafe' },
  { name: '525 Lex Restaurant & Bar' }
]
```

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

```
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({},{_id:0, name:1}).sort( {name: -1})
[
  { name: 'Zum Stammtisch' },
  { name: 'Zum Schneider' },
  { name: "Zorba'S" },
  { name: 'Zebu Grill' },
  { name: "Zaro'S Bread Basket" },
  { name: "Zaro'S Bread Basket" },
  { name: "Zaro'S Bread Basket" },
  { name: "Zaro'S Bread Basket" },
  { name: "Zaro'S Bread Basket" },
  { name: "Zaro'S Bakery" },
  { name: "Zaro'S Bakery" },
  { name: "Zafi'S Luncheonette" },
  { name: 'Yvonne Yvonne Restaurant' },
  { name: 'Yura & Company On Madison' },
  { name: 'Yummy Kitchen' },
  { name: 'Your Bakery' },
  { name: 'Yonah Shimmels Knishes' },
  { name: 'Yolanda Pizzeria Restaurant' },
  { name: "Yip'S" },
  { name: 'Yen Yen Restaurant' }
]
```

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.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({}, { _id: 0, cuisine: 1, borough: 1 }).sort({ cuisine: 1, borough: - 1 })
[
  { borough: 'Manhattan', cuisine: 'Afghan' },
  { borough: 'Manhattan', cuisine: 'Afghan' },
  { borough: 'Manhattan', cuisine: 'Afghan' },
  { borough: 'Manhattan', cuisine: 'Afghan' },
  { borough: 'Queens', cuisine: 'African' },
  { borough: 'Brooklyn', cuisine: 'African' },
  { borough: 'Bronx', cuisine: 'African' },
  { borough: 'Bronx', cuisine: 'African' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' },
  { borough: 'Staten Island', cuisine: 'American ' }
]
```

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

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"address.street": {$regex: /Street/}})
[
  {
    _id: ObjectId("61815303eaaa17f1252dadef"),
    address: {
      building: '351',
      coord: [ -73.98513559999999, 40.7676919 ],
      street: 'West   57 Street',
      zipcode: '10019'
    },
```

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"address.street": {$ne: {$regex: /Street/}}})
[
  {
    _id: ObjectId("61815303eaaa17f1252daded"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
```

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

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({"address.coord": {$type: "double"}}, {_id:0, address:1})
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    }
```

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.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ "grades": { $elemMatch: { "score": { $mod: [7, 0] } } } }, { _id: 0, restaurant_id: 1, name: 1, grades
: 1 })
[
  {
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
```

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.

```
]
Type "it" for more
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ name: { $regex: /mon/ } }, { _id: 0, name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
[
  {
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
```

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.

```
]
Atlas atlas-b1w4mn-shard-0 [primary] restaurants> db.addresses.find({ name: { $regex: /^Mad.*/ } }, { _id: 0, name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
[
  {
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
```