Mongo DB Assignment

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

```
>db.restaurant.find().pretty()
{
	"_id" : ObjectId("5ed9da1efde20b3a44880f38"),
	"address" : {
		"building" : "1007",
		"coord" : [
			-73.856077,
			40.848447
		],
		"street" : "Morris Park Ave",
		"zipcode" : "10462"
	},
	"borough" : "Bronx",
	"cuisine" : "Bakery",
	"grades" : [
		{
			"date" : ISODate("2014-03-03T00:00:00Z"),
			"grade" : "A",
			"score" : 2
		},
		{
			"date" : ISODate("2013-09-11T00:00:00Z"),
			"grade" : "A",
			"score" : 6
		},
		{
			"date" : ISODate("2013-01-24T00:00:00Z"),
			"grade" : "A",
			"score" : 10
		},
		{
			"date" : ISODate("2011-11-23T00:00:00Z"),
			"grade" : "A",
			"score" : 9
		},
		{
			"date" : ISODate("2011-03-10T00:00:00Z"),
			"grade" : "B",
			"score" : 14
		}
	],
	"name" : "Morris Park Bake Shop",
	"restaurant_id" : "30075445"
}
```

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

```
> db.restaurant.find({},{"_id":0,"restaurant_id":1,"name":1,"borough":1,"cuisine":1}).pretty()
{
	"borough" : "Bronx",
	"cuisine" : "Bakery",
	"name" : "Morris Park Bake Shop",
	"restaurant_id" : "30075445"
}
{
	"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.

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

```
> db.restaurant.find({},{"_id":0,"restaurant_id":1,"name":1,"borough":1,
"address.zipcode":1}).pretty()
{
	"address" : {
		"zipcode" : "10462"
	},
	"borough" : "Bronx",
	"name" : "Morris Park Bake Shop",
	"restaurant_id" : "30075445"
}
```

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

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

```
        },
        "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.

```
> db.restaurant.find({},{"_id":0,"restaurant_id":1,"name":1,
"address.borough":"Bronx"}).pretty().limit(5)
{
        "address" : {

        },
        "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.

```
> db.restaurant.find({},{"_id":0,"restaurant_id":1,"name":1,
"address.borough":"Bronx"}).pretty().skip(5).limit(5)
{
        "address" : {

        },
        "name" : "Brunos On The Boulevard",
        "restaurant_id" : "40356151"
}
```

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

```
> db.restaurant.find({"grades.score":{$gt:90}}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44881096"),
        "address" : {
                "building" : "65",
                "coord" : [
                        -73.9782725,
                        40.7624022
                ],
                "street" : "West   54 Street",
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-08-22T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
```

```
            {
                "date" : ISODate("2014-03-28T00:00:00Z"),
                "grade" : "C",
                "score" : 131
            },
            {
                "date" : ISODate("2013-09-25T00:00:00Z"),
                "grade" : "A",
                "score" : 11
            },
            {
                "date" : ISODate("2013-04-08T00:00:00Z"),
                "grade" : "B",
                "score" : 25
            },
            {
                "date" : ISODate("2012-10-15T00:00:00Z"),
                "grade" : "A",
                "score" : 11
            },
            {
                "date" : ISODate("2011-10-19T00:00:00Z"),
                "grade" : "A",
                "score" : 13
            }
        ],
        "name" : "Murals On 54/Randolphs'S",
        "restaurant_id" : "40372466"
}
```


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


```
> db.restaurant.find({"grades.score":{$gt:80}, "grades.score":{$lt:100}}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44880f38"),
        "address" : {
            "building" : "1007",
            "coord" : [
                -73.856077,
                40.848447
            ],
            "street" : "Morris Park Ave",
            "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "grades" : [
            {
                "date" : ISODate("2014-03-03T00:00:00Z"),
                "grade" : "A",
                "score" : 2
```

```
            },
            {
                "date" : ISODate("2013-09-11T00:00:00Z"),
                "grade" : "A",
                "score" : 6
            },
            {
                "date" : ISODate("2013-01-24T00:00:00Z"),
                "grade" : "A",
                "score" : 10
            },
            {
                "date" : ISODate("2011-11-23T00:00:00Z"),
                "grade" : "A",
                "score" : 9
            },
            {
                "date" : ISODate("2011-03-10T00:00:00Z"),
                "grade" : "B",
                "score" : 14
            }
        ],
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
```


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


```
> db.restaurant.find({"address.coord":{$lt:-95.754168}}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44881582"),
        "address" : {
            "building" : "3707",
            "coord" : [
                -101.8945214,
                33.5197474
            ],
            "street" : "82 Street",
            "zipcode" : "11372"
        },
        "borough" : "Queens",
        "cuisine" : "American ",
        "grades" : [
            {
                "date" : ISODate("2014-06-04T00:00:00Z"),
                "grade" : "A",
                "score" : 12
            },
            {
                "date" : ISODate("2013-11-07T00:00:00Z"),
                "grade" : "B",
```

```
                "score" : 19
            },
            {
                "date" : ISODate("2013-05-17T00:00:00Z"),
                "grade" : "A",
                "score" : 11
            },
            {
                "date" : ISODate("2012-08-29T00:00:00Z"),
                "grade" : "A",
                "score" : 11
            },
            {
                "date" : ISODate("2012-04-03T00:00:00Z"),
                "grade" : "A",
                "score" : 12
            },
            {
                "date" : ISODate("2011-11-16T00:00:00Z"),
                "grade" : "A",
                "score" : 7
            }
        ],
        "name" : "Burger King",
        "restaurant_id" : "40534067"
}
```


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.restaurant.find({$and:[{"cuisine":{$ne:"American "}},{'grades.score':{$gte:70}},
{'address.coord.0':{$lte:-65.754168}}]}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44881137"),
        "address" : {
                "building" : "345",
                "coord" : [
                        -73.9864626,
                        40.7266739
                ],
                "street" : "East 6 Street",
                "zipcode" : "10003"
        },
        "borough" : "Manhattan",
        "cuisine" : "Indian",
        "grades" : [
                {
                        "date" : ISODate("2014-09-15T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
```

```
                "date" : ISODate("2014-01-14T00:00:00Z"),
                "grade" : "A",
                "score" : 8
            },
            {
                "date" : ISODate("2013-05-30T00:00:00Z"),
                "grade" : "A",
                "score" : 12
            },
            {
                "date" : ISODate("2013-04-24T00:00:00Z"),
                "grade" : "P",
                "score" : 2
            },
            {
                "date" : ISODate("2012-10-01T00:00:00Z"),
                "grade" : "A",
                "score" : 9
            },
            {
                "date" : ISODate("2012-04-06T00:00:00Z"),
                "grade" : "C",
                "score" : 92
            },
            {
                "date" : ISODate("2011-11-03T00:00:00Z"),
                "grade" : "C",
                "score" : 41
            }
        ],
        "name" : "Gandhi",
        "restaurant_id" : "40381295"
}
```


12. 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.restaurant.find({$and:[{"cuisine":{$ne:"American"}}, {"grades.grade":"A"}, {"borough":
{$ne:"Brooklyn"}}]}).sort({"cuisine":-1}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44881644"),
        "address" : {
                "building" : "89",
                "coord" : [
                        -73.9995899,
                        40.7168015
                ],
                "street" : "Baxter Street",
                "zipcode" : "10013"
        },
        "borough" : "Manhattan",
```

```
    "cuisine" : "Vietnamese/Cambodian/Malaysia",
    "grades" : [
        {
            "date" : ISODate("2014-08-21T00:00:00Z"),
            "grade" : "A",
            "score" : 13
        },
        {
            "date" : ISODate("2013-08-31T00:00:00Z"),
            "grade" : "A",
            "score" : 13
        },
        {
            "date" : ISODate("2013-04-11T00:00:00Z"),
            "grade" : "C",
            "score" : 3
        },
        {
            "date" : ISODate("2012-10-17T00:00:00Z"),
            "grade" : "A",
            "score" : 4
        },
        {
            "date" : ISODate("2012-05-15T00:00:00Z"),
            "grade" : "A",
            "score" : 10
        }
    ],
    "name" : "Thai Son",
    "restaurant_id" : "40559606"
}
```


13. 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.restaurant.find({"name":{"$regex":"^Wil"}}, {"_id":0, "restaurant_id":1, "name":1,
"borough":1, "cuisine":1}).pretty()
{
    "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"
}
```

14. 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.restaurant.find({"grades.score":{$lte:10}}, {"_id":0,"restaurant_id":1, "name":1, "borough":1, "cuisine":1}).pretty()
{
    "borough" : "Bronx",
    "cuisine" : "Bakery",
    "name" : "Morris Park Bake Shop",
    "restaurant_id" : "30075445"
}
```

15. 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.restaurant.find({$and:[{"name":{"$regex":"^Wil"}}, {$or:[{"cuisine":{$ne:"American"}}, {"cuisine":{$ne:"Chinees"}}]}]}, {"_id":0,"restaurant_id":1, "name":1, "borough":1, "cuisine":1}).pretty()
{
    "borough" : "Brooklyn",
    "cuisine" : "Delicatessen",
    "name" : "Wilken'S Fine Food",
    "restaurant_id" : "40356483"
}
{
    "borough" : "Bronx",
    "cuisine" : "American ",
    "name" : "Wild Asia",
    "restaurant_id" : "40357217"
}
```

16. 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.restaurant.find({$and: [{"grades.date" : ISODate("2014-08-16T00:00:00Z"), "grades.score":9, "grades.grade":"A"}]},{"_id":0, "restaurant_id":1,name:1,"grades.grade":1}).pretty()
{
    "grades" : [
        {
            "grade" : "A"
        },
        {
            "grade" : "A"
        },
        {
            "grade" : "A"
```

```
            },
            {
                "grade" : "A"
            }
        ],
        "name" : "Sal'S Deli",
        "restaurant_id" : "40361618"
}
```


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


```
> db.restaurant.find().sort({"name":1}).pretty()
{
        "_id" : ObjectId("5ed9da1ffde20b3a44881bcb"),
        "address" : {
            "building" : "129",
            "coord" : [
                -73.962943,
                40.685007
            ],
            "street" : "Gates Avenue",
            "zipcode" : "11238"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Italian",
        "grades" : [
            {
                "date" : ISODate("2014-03-06T00:00:00Z"),
                "grade" : "A",
                "score" : 5
            },
            {
                "date" : ISODate("2013-08-29T00:00:00Z"),
                "grade" : "A",
                "score" : 2
            },
            {
                "date" : ISODate("2013-03-08T00:00:00Z"),
                "grade" : "A",
                "score" : 7
            },
            {
                "date" : ISODate("2012-06-27T00:00:00Z"),
                "grade" : "A",
                "score" : 7
            },
            {
                "date" : ISODate("2011-11-17T00:00:00Z"),
                "grade" : "A",
                "score" : 12
            }
```

```
        ],
        "name" : "(Lewis Drug Store) Locanda Vini E Olii",
        "restaurant_id" : "40804423"
}
```

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

```
> db.restaurant.find({"address.street":{$exists:true}}).pretty()
{
        "_id" : ObjectId("5ed9da1efde20b3a44880f38"),
        "address" : {
                "building" : "1007",
                "coord" : [
                        -73.856077,
                        40.848447
                ],
                "street" : "Morris Park Ave",
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "grades" : [
                {
                        "date" : ISODate("2014-03-03T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
                        "date" : ISODate("2013-09-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 6
                },
                {
                        "date" : ISODate("2013-01-24T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2011-11-23T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2011-03-10T00:00:00Z"),
                        "grade" : "B",
                        "score" : 14
                }
        ],
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
```

19. 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.restaurant.find({"name":/mon/}, {"_id":0, "restaurant_id":1, "name":1, "borough":1,
"longitude":1, "attitude":1, "cuisine":1}).pretty()
{
      "borough" : "Manhattan",
      "cuisine" : "American ",
      "name" : "Desmond'S Tavern",
      "restaurant_id" : "40366396"
}
```

20. 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.restaurant.find({"name":/^Mad/}, {"_id":0, "name":1, "borough":1, "longitude":1, "latitude":1,
"cuisine":1}).pretty()
{
      "borough" : "Manhattan",
      "cuisine" : "American ",
      "name" : "Madison Square"
}
```