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 restaurant

Ans :   >db.restaurants.find({}, { _id: 0 })

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

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

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

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

```
rails@rails-System: ~                                              ↑↓ En ◄)) 7:12 PM ⚙

> db.restaurants.find({"grades.score": {$gt:90}}).pretty()
{
        "_id" : ObjectId("5947bc1fb60c37c179375d9d"),
        "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"
}
{
```
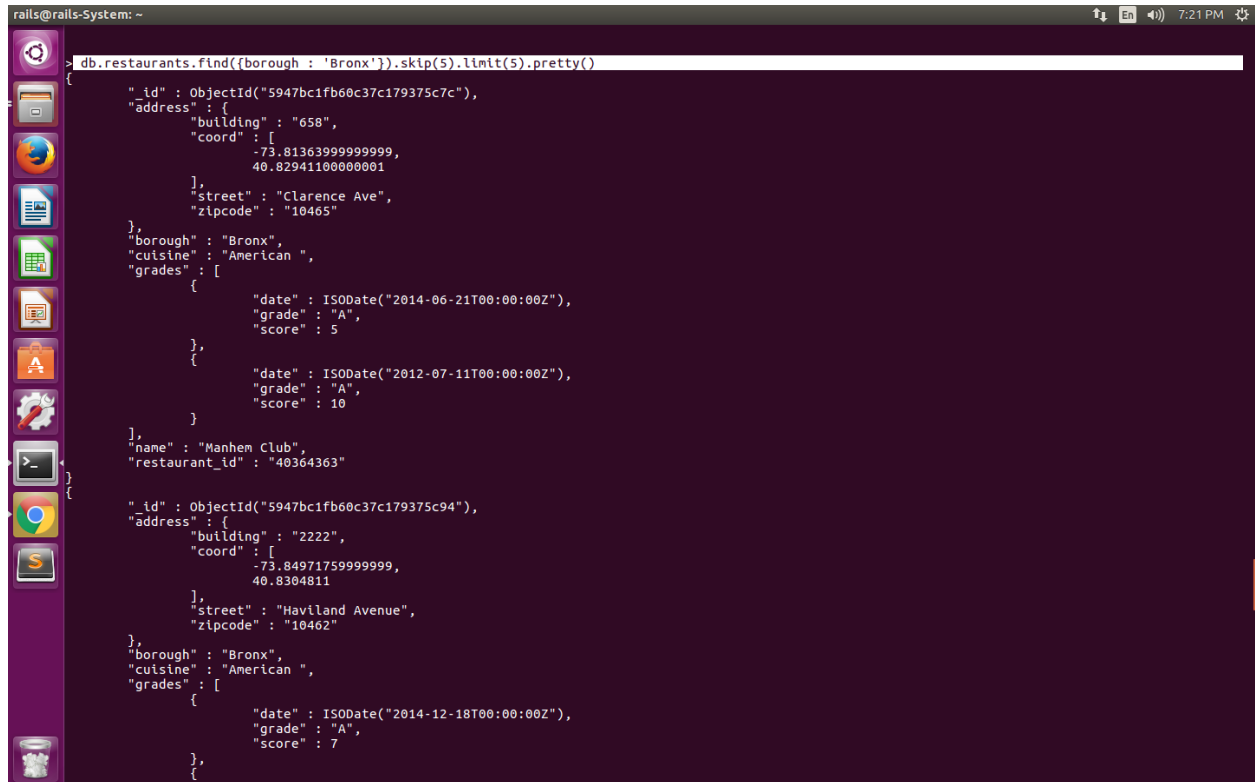
4. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx

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

rails@rails-System: ~

> db.restaurants.find({borough : 'Bronx'}).limit(5)
{ "_id" : ObjectId("5947bc1fb60c37c179375c3f"), "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" }
{ "_id" : ObjectId("5947bc1fb60c37c179375c49"), "address" : { "building" : "2300", "coord" : [-73.8786113, 40.8502883], "street" : "Southern Boulevard", "zipcode" : "10460" }, "borough" : "Bronx", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-05-28T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-06-19T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-06-15T00:00:00Z"), "grade" : "A", "score" : 3 }], "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "_id" : ObjectId("5947bc1fb60c37c179375c5e"), "address" : { "building" : "1006", "coord" : [-73.84856870000002, 40.8903781], "street" : "East 233 Street", "zipcode" : "10466" }, "borough" : "Bronx", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "grades" : [{ "date" : ISODate("2014-04-24T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-09-05T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-02-21T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-07-03T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-07-11T00:00:00Z"), "grade" : "A", "score" : 5 }], "name" : "Carvel Ice Cream", "restaurant_id" : "40363093" }
{ "_id" : ObjectId("5947bc1fb60c37c179375c62"), "address" : { "building" : "1236", "coord" : [-73.8893654, 40.81376179999999], "street" : "238 Spofford Ave", "zipcode" : "10474" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [{ "date" : ISODate("2013-12-30T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-01-08T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-06-12T00:00:00Z"), "grade" : "B", "score" : 15 }], "name" : "Happy Garden", "restaurant_id" : "40363289" }
{ "_id" : ObjectId("5947bc1fb60c37c179375c74"), "address" : { "building" : "277", "coord" : [-73.8941893, 40.8634684], "street" : "East Kingsbridge Road", "zipcode" : "10458" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [{ "date" : ISODate("2014-03-03T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-09-26T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-03-19T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-08-29T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-08-17T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "Happy Garden", "restaurant_id" : "40364296" }
>

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

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

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

Ans : > db.restaurants.find({ $and : [{ "grades.score" : {$gt:80}}, { "grades.score" : {$lt:100}}] }).pretty()


```
rails@rails-System: ~                                    ↑↓  En  ◄))  7:44 PM  ⚙
        "restaurant_id" : "40393488"
}
> cls

> db.restaurants.find( { $and : [ { "grades.score" : {$gt:80}}, { "grades.score" : {$lt:100}} ] } ).pretty()
{
        "_id" : ObjectId("5947bc1fb60c37c179375d9d"),
        "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",
```

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

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

8. 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.restaurants.find({ $and: [{ "cuisine": {$ne: "American "}},{ "grades.score" : {$gt:70}}, { "address.coord.0" : {$lt:-65.754168}}] })
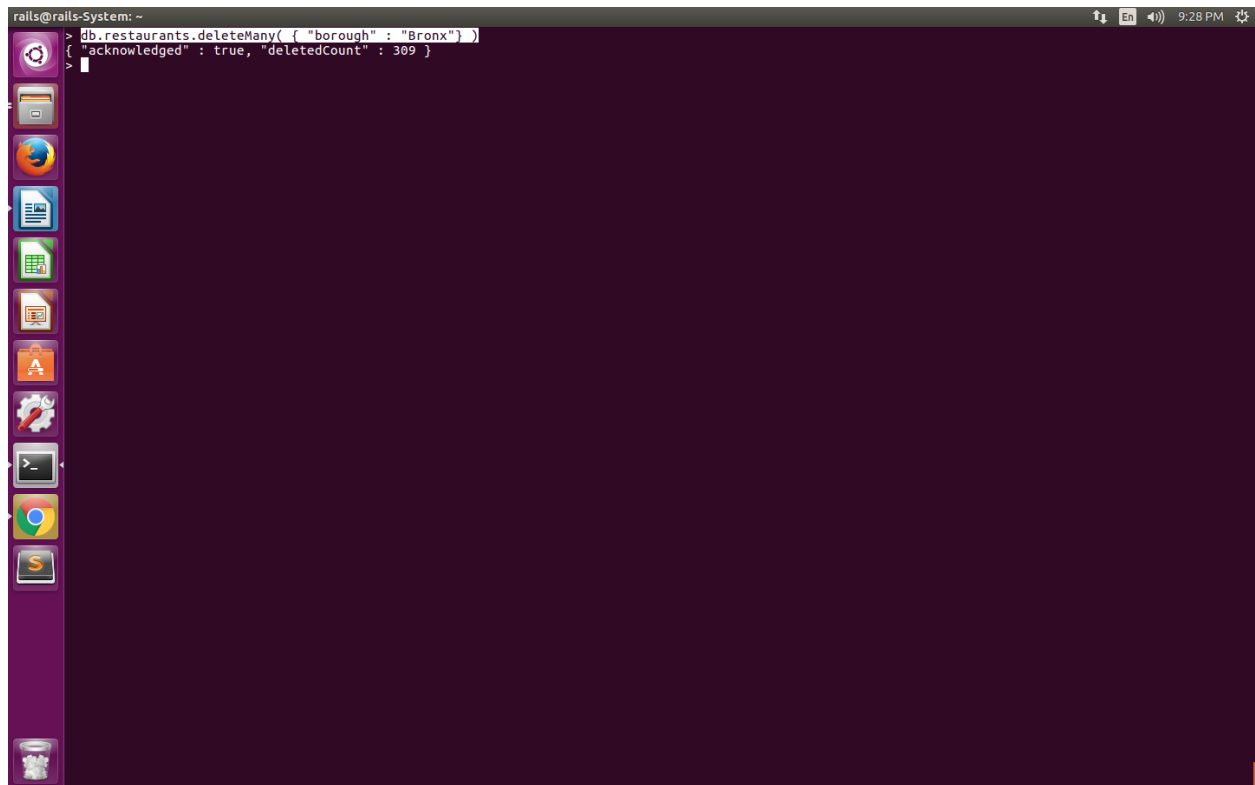
9. Write a MongoDB query to update the restaurant's grade to 'B' whose score is more than or equal to 10(update only the first grade).

Ans : >db.restaurants.update({"grades.score" : {$gte:10}}, { $set: {"grades.$.grade" : "B"} }, { multi : true })

10. Write a MongoDB query to delete the restaurants in borough Bronx.

Ans : >db.restaurants.deleteMany({ "borough" : "Bronx"})