

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 })

```
rails@rails-system: ~
> show dbs
admin          0.000GB
local          0.000GB
nydb           0.000GB
restaurants    0.001GB
test           0.000GB
testDB         0.000GB
> use restaurants
switched to db restaurants
> db.restaurants.find({}, { _id: 0 })
{ "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" }
{ "address": { "building": "469", "coord": [ -73.961704, 40.662942 ], "street": "Flatbush Avenue", "zipcode": "11225", "borough": "Brooklyn", "cuisine": "Hamburgers", "grades": [ { "date": ISODate("2014-12-30T00:00:00Z"), "grade": "A", "score": 8 }, { "date": ISODate("2014-07-01T00:00:00Z"), "grade": "B", "score": 23 }, { "date": ISODate("2013-04-30T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2012-05-08T00:00:00Z"), "grade": "A", "score": 12 } ], "name": "Wendy's", "restaurant_id": "30112340" }
{ "address": { "building": "351", "coord": [ -73.98513559999999, 40.7676919 ], "street": "West 57 Street", "zipcode": "10019", "borough": "Manhattan", "cuisine": "Irish", "grades": [ { "date": ISODate("2014-09-06T00:00:00Z"), "grade": "A", "score": 2 }, { "date": ISODate("2013-07-22T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2012-07-31T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2011-12-29T00:00:00Z"), "grade": "A", "score": 12 } ], "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841" }
{ "address": { "building": "2780", "coord": [ -73.98241999999999, 40.579505 ], "street": "Stillwell Avenue", "zipcode": "11224", "borough": "Brooklyn", "cuisine": "American", "grades": [ { "date": ISODate("2014-06-10T00:00:00Z"), "grade": "A", "score": 5 }, { "date": ISODate("2013-06-05T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2012-04-13T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2011-10-12T00:00:00Z"), "grade": "A", "score": 12 } ], "name": "Riviera Caterer", "restaurant_id": "40356018" }
{ "address": { "building": "97-22", "coord": [ -73.8601152, 40.7311739 ], "street": "63 Road", "zipcode": "11374", "borough": "Queens", "cuisine": "Jewish/Kosher", "grades": [ { "date": ISODate("2014-11-24T00:00:00Z"), "grade": "Z", "score": 20 }, { "date": ISODate("2013-01-17T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-08-02T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2011-12-15T00:00:00Z"), "grade": "B", "score": 25 } ], "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }
{ "address": { "building": "8825", "coord": [ -73.8863827, 40.7643124 ], "street": "Astoria Boulevard", "zipcode": "11369", "borough": "Queens", "cuisine": "American", "grades": [ { "date": ISODate("2014-11-15T00:00:00Z"), "grade": "Z", "score": 38 }, { "date": ISODate("2014-05-02T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2013-03-02T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2012-02-10T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Brunos On The Boulevard", "restaurant_id": "40356151" }
{ "address": { "building": "2206", "coord": [ -74.1377286, 40.6119572 ], "street": "Victory Boulevard", "zipcode": "10314", "borough": "Staten Island", "cuisine": "Jewish/Kosher", "grades": [ { "date": ISODate("2014-10-06T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2014-05-20T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2013-04-04T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2012-01-24T00:00:00Z"), "grade": "A", "score": 9 } ], "name": "Kosher Island", "restaurant_id": "40356442" }
{ "address": { "building": "7114", "coord": [ -73.9068506, 40.6199034 ], "street": "Avenue U", "zipcode": "11234", "borough": "Brooklyn", "cuisine": "Delicatessen", "grades": [ { "date": ISODate("2014-05-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2014-01-14T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-03-09T00:00:00Z"), "grade": "A", "score": 8 }, { "date": ISODate("2012-07-18T00:00:00Z"), "grade": "A", "score": 9 } ], "name": "Wilken S Fine Food", "restaurant_id": "40356483" }
{ "address": { "building": "6409", "coord": [ -74.00528999999999, 40.628886 ], "street": "11 Avenue", "zipcode": "11219", "borough": "Brooklyn", "cuisine": "American", "grades": [ { "date": ISODate("2014-07-18T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2013-07-30T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2013-02-13T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2012-08-16T00:00:00Z"), "grade": "A", "score": 2 }, { "date": ISODate("2011-08-17T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Regina Caterers", "restaurant_id": "40356649" }
{ "address": { "building": "1839", "coord": [ -73.9482609, 40.6408271 ], "street": "Nostrand Avenue", "zipcode": "11226", "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "grades": [ { "date": ISODate("2014-07-14T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISO
```

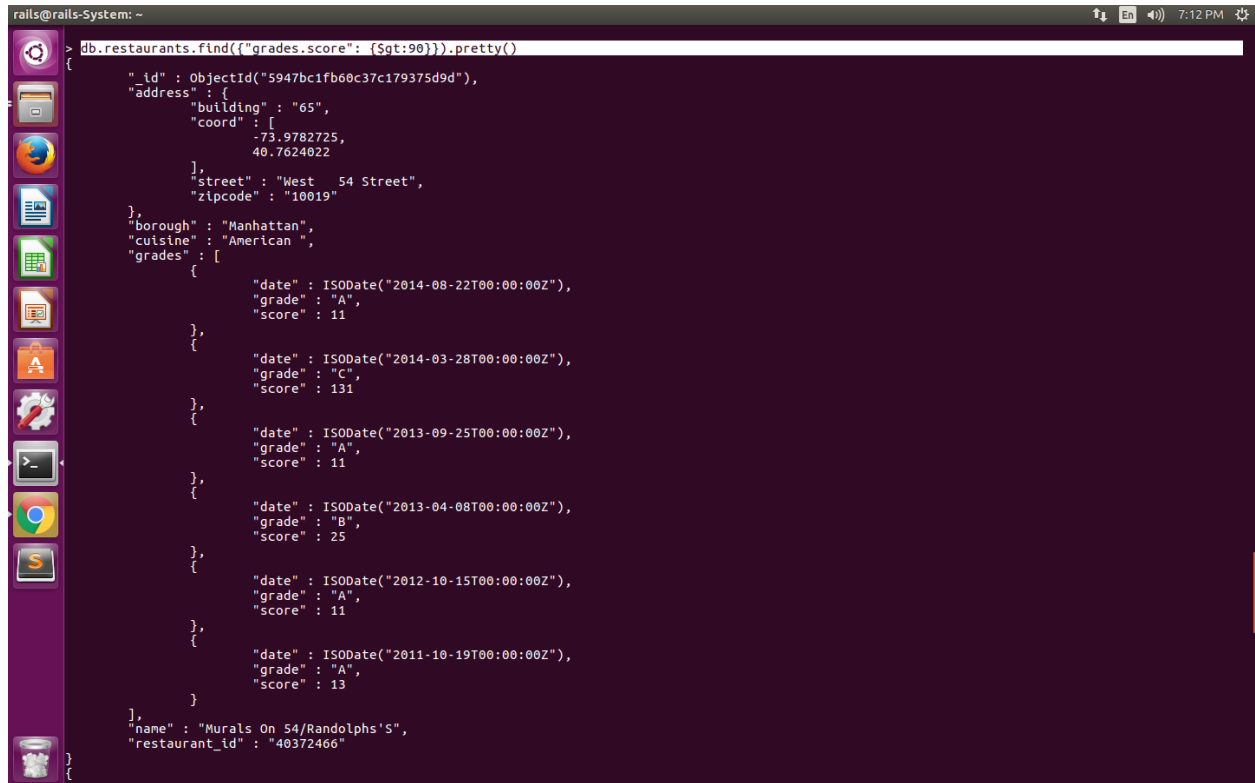
2. Write a MongoDB query to display all the restaurant which is in the borough Bronx

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

```
rails@rails-System: ~  
n", "cuisine": "American", "grades": [ { "date": ISODate("2014-11-19T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2013-11-14T00:00:00Z"), "grade": "A", "score": 2 }, { "date": ISODate("2012-12-05T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-05-17T00:00:00Z"), "grade": "A", "score": 11 } ], { "name": "The Movable Feast", "restaurant_id": "40361606" }  
> db.restaurants.find({'borough' : 'Bronx'})  
Type it for more  
{ "_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": "38075445" }  
{ "_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": "Wld Asia", "restaurant_id": "40357217" }  
{ "_id": ObjectId("5947bc1fb60c37c179375c5e"), "address": { "building": "1006", "coord": [ -73.84850870000002, 40.8903781 ], "street": "East 233 S", "zipcode": "10466" }, "borough": "Bronx", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "grades": [ { "date": ISODate("2014-04-24T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2013-02-21T00:00:00Z"), "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("2012-08-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2013-03-19T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Happy Garden", "restaurant_id": "40364296" }  
{ "_id": ObjectId("5947bc1fb60c37c179375c7c"), "address": { "building": "658", "coord": [ -73.81363999999999, 40.82941100000001 ], "street": "Clarke Ave", "zipcode": "10465" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-06-21T00:00:00Z"), "grade": "A", "score": 5 }, { "date": ISODate("2012-07-11T00:00:00Z"), "grade": "A", "score": 10 } ], "name": "Manhem Club", "restaurant_id": "40364363" }  
{ "_id": ObjectId("5947bc1fb60c37c179375c94"), "address": { "building": "2222", "coord": [ -73.84971759999999, 40.8304811 ], "street": "Haviland Avenue", "zipcode": "10462" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-12-18T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2014-05-01T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2013-03-14T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2012-09-20T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 19 } ], "name": "The New Starling Athletic Club Of The Bronx", "restaurant_id": "40364956" }  
{ "_id": ObjectId("5947bc1fb60c37c179375cac"), "address": { "building": "72", "coord": [ -73.92506, 40.8275556 ], "street": "East 161 Street", "zipcode": "10451" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-04-15T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2013-11-14T00:00:00Z"), "grade": "A", "score": 4 }, { "date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-12-31T00:00:00Z"), "grade": "B", "score": 15 }, { "date": ISODate("2012-05-30T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-01-09T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "C", "score": 37 } ], "name": "Yankee Tavern", "restaurant_id": "40365499" }  
{ "_id": ObjectId("5947bc1fb60c37c179375cbf"), "address": { "building": "331", "coord": [ -73.87786539999999, 40.8724377 ], "street": "East 204 Street", "zipcode": "10467" }, "borough": "Bronx", "cuisine": "Irish", "grades": [ { "date": ISODate("2014-08-26T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2014-03-26T00:00:00Z"), "grade": "B", "score": 23 }, { "date": ISODate("2013-09-11T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-12-18T00:00:00Z"), "grade": "B", "score": 27 }, { "date": ISODate("2011-10-20T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Mcdwysers Pub", "restaurant_id": "40365893" }  
{ "_id": ObjectId("5947bc1fb60c37c179375cd8"), "address": { "building": "5820", "coord": [ -73.9002615, 40.885186 ], "street": "Broadway", "zipcode": "10463" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-02-26T00:00:00Z"), "grade": "A", "score": 5 }, { "date": ISODate("2013-10-09T00:00:00Z"), "grade": "B", "score": 19 }, { "date": ISODate("2013-05-15T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2012-11-20T00:00:00Z"), "grade": "B", "score": 18 }, { "date": ISODate("2011-10-17T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2011-07-11T00:00:00Z"), "grade": "A", "score": 10 } ], "name": "The Movable Feast", "restaurant_id": "40361606" }
```

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()



The screenshot shows a terminal window titled 'rails@rails-System: ~' with a dark purple background. The command 'db.restaurants.find({"grades.score": {\$gt:90}}).pretty()' is entered at the prompt. The output is a JSON document for a restaurant named 'Murals On 54/Randolphs'S' with restaurant_id '40372466'. The document includes an ObjectId, address (West 54 Street, Manhattan, 10019), cuisine (American), and a grades array. The grades array contains five entries with dates, grades, and scores. The scores are 11, 131, 11, 25, and 11, all of which are greater than 90.

```
> 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"
}
```

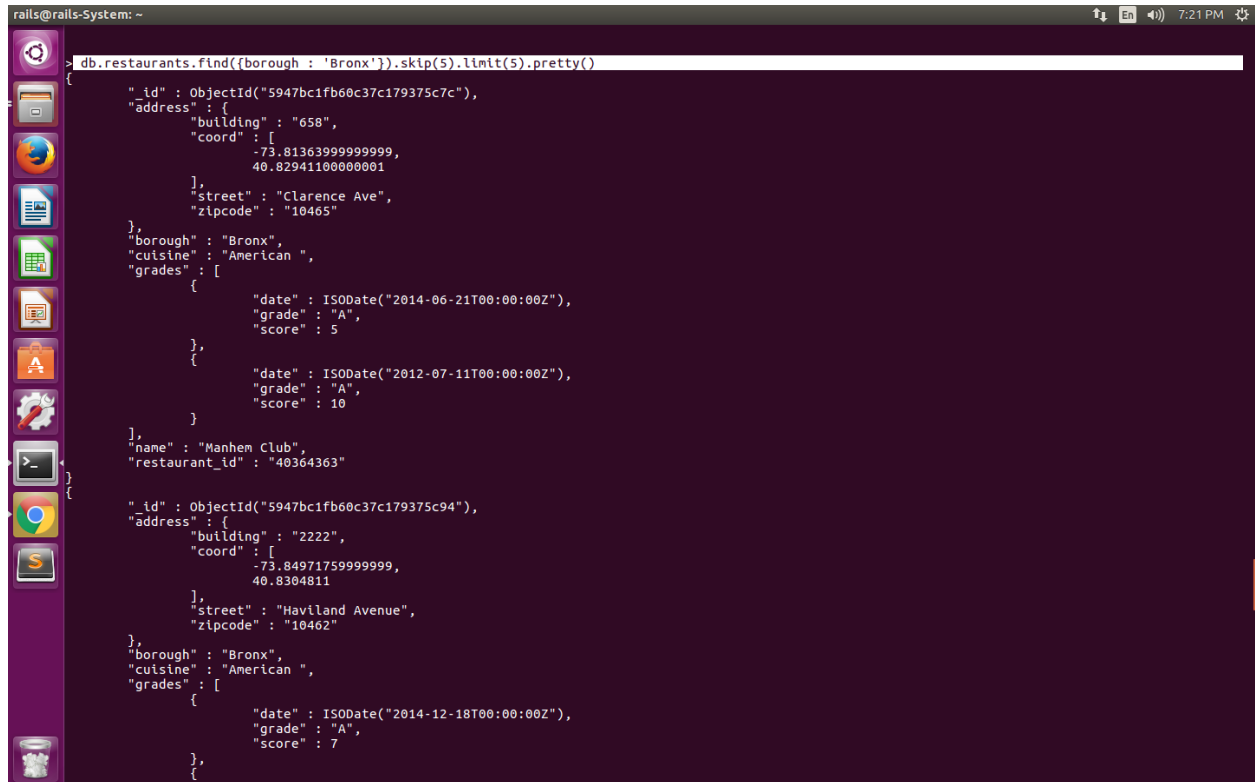
4. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx

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" : 9 }, { "date" : ISODate("2013-09-05T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-02-21T00:00:00Z"), "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()

A terminal window titled 'rails@rails-System: ~' shows the execution of a MongoDB query. The command 'db.restaurants.find({borough : 'Bronx'}).skip(5).limit(5).pretty()' is entered. The output displays two restaurant objects in a pretty-printed JSON format. The first object represents a restaurant with ID '5947bc1fb60c37c179375c7c', located at 'Clarence Ave' in the Bronx, with a score of 5. The second object represents a restaurant with ID '5947bc1fb60c37c179375c94', located at 'Haviland Avenue' in the Bronx, with a score of 7. The terminal window includes a sidebar with application icons and a system tray at the top right showing the time as 7:21 PM.

```
rails@rails-System: ~  
$ db.restaurants.find({borough : 'Bronx'}).skip(5).limit(5).pretty()  
{  
  "_id" : ObjectId("5947bc1fb60c37c179375c7c"),  
  "address" : {  
    "building" : "658",  
    "coord" : [  
      -73.81363999999999,  
      40.82941100000001  
    ],  
    "street" : "Clarence Ave",  
    "zipcode" : "10465"  
  },  
  "borough" : "Bronx",  
  "cuisine" : "American",  
  "grades" : [  
    {  
      "date" : ISODate("2014-06-21T00:00:00Z"),  
      "grade" : "A",  
      "score" : 5  
    },  
    {  
      "date" : ISODate("2012-07-11T00:00:00Z"),  
      "grade" : "A",  
      "score" : 10  
    }  
  ],  
  "name" : "Manhem Club",  
  "restaurant_id" : "40364363"  
},  
{  
  "_id" : ObjectId("5947bc1fb60c37c179375c94"),  
  "address" : {  
    "building" : "2222",  
    "coord" : [  
      -73.84971759999999,  
      40.8304811  
    ],  
    "street" : "Haviland Avenue",  
    "zipcode" : "10462"  
  },  
  "borough" : "Bronx",  
  "cuisine" : "American",  
  "grades" : [  
    {  
      "date" : ISODate("2014-12-18T00:00:00Z"),  
      "grade" : "A",  
      "score" : 7  
    },  
    {  
      "date" : ISODate("2014-12-18T00:00:00Z"),  
      "grade" : "A",  
      "score" : 7  
    }  
  ],  
  "name" : "Manhem Club",  
  "restaurant_id" : "40364363"  
}
```

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

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

```
rails@rails-System: ~  
> db.restaurants.find( { "grades.score" : { $gt : 80 , $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" : "B",  
      "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"  
}
```

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}})

```
rails@rails-System: ~
> db.restaurants.find({"address.coord.0" : {$lt:-95.754168}})
{ "_id" : ObjectId("5947bc1fb60c37c179376288"), "address" : { "building" : "3707", "coord" : [ -101.8945214, 33.5197474 ], "street" : "82 Street", "zip
code" : "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" }
{ "_id" : ObjectId("5947bc1fb60c37c1793765f4"), "address" : { "building" : "15259", "coord" : [ -119.6368672, 36.2504996 ], "street" : "10 Avenue", "zi
pcode" : "11357" }, "borough" : "Queens", "cuisine" : "Italian", "grades" : [ { "date" : ISODate("2014-09-04T00:00:00Z"), "grade" : "A", "score" : 11 }
], { "date" : ISODate("2014-03-26T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-03-04T00:00:00Z"), "grade" : "A", "score" : 10 },
{ "date" : ISODate("2012-09-27T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-04-20T00:00:00Z"), "grade" : "A", "score" : 7 }, {
"date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "C", "score" : 34 }, { "name" : "Cascarino'S", "restaurant_id" : "40668681" }
{ "_id" : ObjectId("5947bc1fb60c37c179376a9b"), "address" : { "building" : "60", "coord" : [ -111.9975205, 42.0970258 ], "street" : "West Side Highway"
, "zipcode" : "10006" }, "borough" : "Manhattan", "cuisine" : "Japanese", "grades" : [ { "date" : ISODate("2014-03-20T00:00:00Z"), "grade" : "A", "score"
: 9 }, { "date" : ISODate("2013-06-28T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-07-05T00:00:00Z"), "grade" : "A", "score"
: 13 }, { "date" : ISODate("2011-07-27T00:00:00Z"), "grade" : "A", "score" : 2 } ], { "name" : "Sports Center At Chelsea Pters (Susht Bar)", "restaurant
_id" : "40802356" }
>
```

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({ "cuisine": {\$ne: "American "}, "grades.score" : {\$gt:70}, "address.coord.0" : {\$lt:-65.754168} }).pretty()

```
rails@rails-System: ~  
> db.restaurants.find({ "cuisine": {$ne: "American "}, "grades.score" : {$gt:70}, "address.coord.0" : {$lt:-65.754168} }).pretty()  
{  
  "_id" : ObjectId("5947bc1fb60c37c179375e3e"),  
  "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" : "B",  
      "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  
    }  
  ]  
}
```

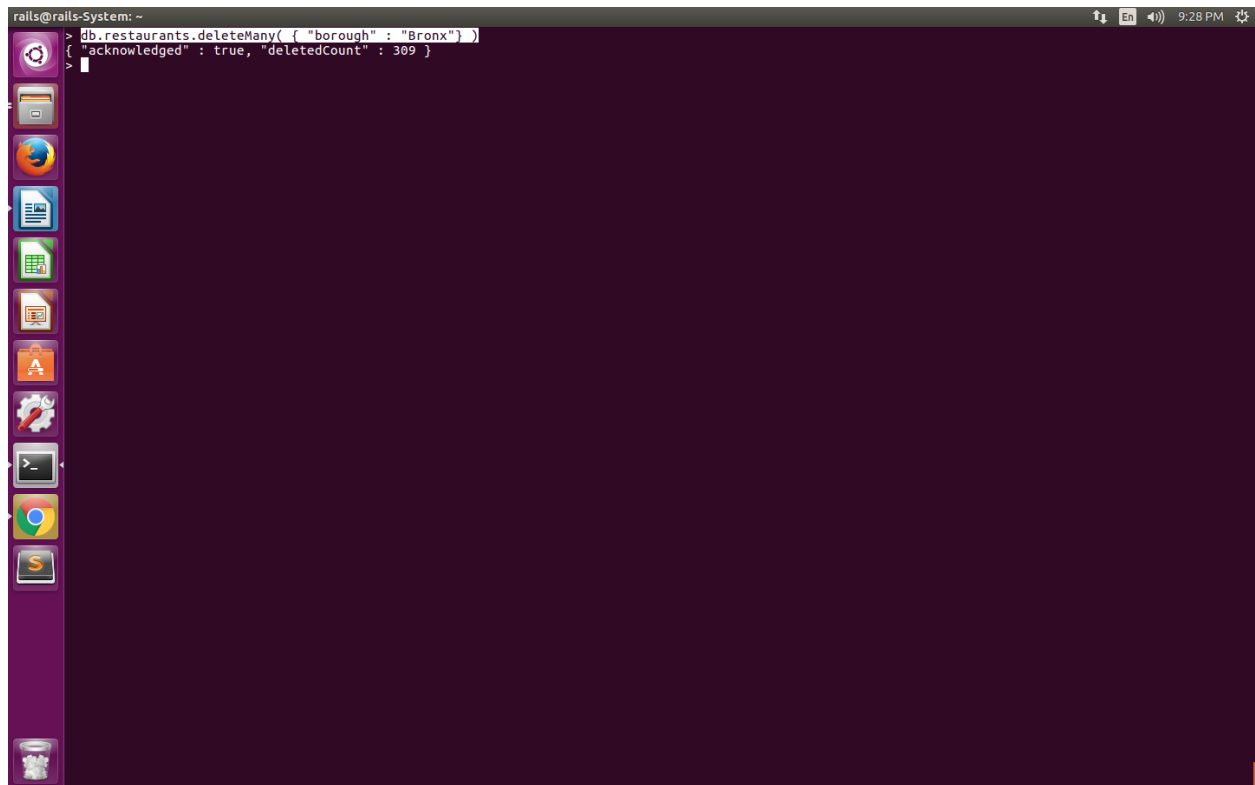

9. Write a MongoDB query to update the restaurant's grade to 'B' whose score is more than or equal to 10(update only the first grade).

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

```
rails@rails-System: ~  
{"address": {"building": "231",  
  "coord": [-73.9772294,  
    40.7527262  
  ],  
  "street": "Grand Central Station",  
  "zipcode": "10017"},  
"borough": "Manhattan",  
"cuisine": "Italian",  
"grades": [{"date": ISODate("2015-01-07T00:00:00Z"),  
  "grade": "Z",  
  "score": 20  
},  
{"date": ISODate("2014-07-03T00:00:00Z"),  
  "grade": "B",  
  "score": 14  
},  
{"date": ISODate("2013-12-21T00:00:00Z"),  
  "grade": "A",  
  "score": 13  
},  
{"date": ISODate("2013-05-17T00:00:00Z"),  
  "grade": "C",  
  "score": 76  
},  
{"date": ISODate("2012-04-20T00:00:00Z"),  
  "grade": "A",  
  "score": 12  
}],  
"name": "Two Boots Grand Central",  
"restaurant_id": "40725591"}  
> db.restaurants.update({"grades.score" : {$gte:10}}, {"grades.$grade" : "B"}, {multi : true })  
2017-07-17T21:20:04.950+0530 E QUERY [thread1] Error: can't have . in field names [grades.$grade] :  
DBCollection.prototype.validateForStorage@src/mongo/shell/collection.js:244:1  
DBCollection.prototype.validateUpdateDoc@src/mongo/shell/collection.js:501:9  
Bulk/findOperations.update@src/mongo/shell/bulk_api.js:673:17  
DBCollection.prototype.update@src/mongo/shell/collection.js:577:13  
@(shell):1:1  
= db.restaurants.update({"grades.score" : {$gte:10}}, { $set: {"grades.$grade" : "B"} }, { multi : true })  
WriteResult({ "nMatched" : 3600, "nUpserted" : 0, "nModified" : 3147 })
```

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

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

A terminal window titled 'rails@rails-System: ~' with a dark purple background. The prompt is '>'. The command 'db.restaurants.deleteMany({ "borough" : "Bronx"})' is entered and highlighted. The output is '{ "acknowledged" : true, "deletedCount" : 309 }'. The prompt '>' is visible on the next line. On the left side of the terminal, there is a vertical dock with various application icons including a file manager, web browser, and office applications. The top right corner of the terminal shows system icons for network, volume, and battery, along with the time '9:28 PM'.