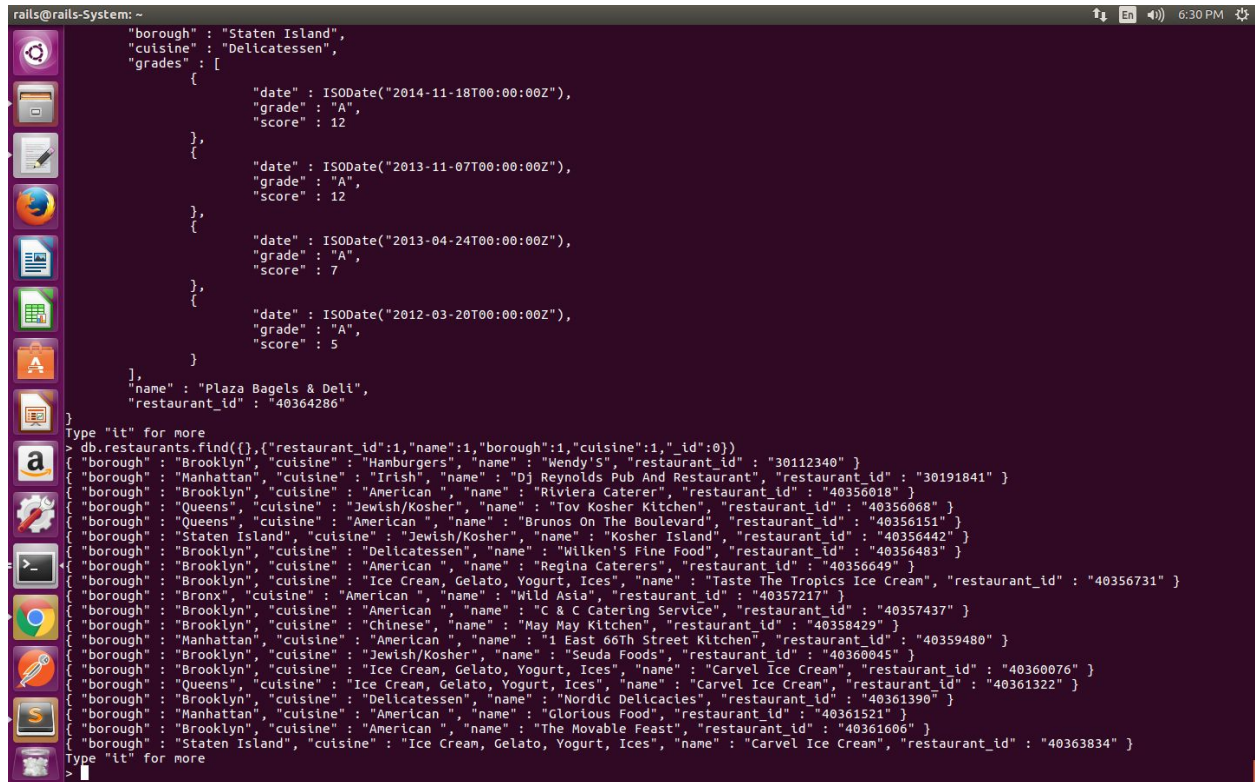


MongoDB Queries and Screenshots

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



The screenshot shows a terminal window with a dark background. The prompt is 'rails@rails-System: ~'. The user has entered a MongoDB query to find all documents in the 'restaurants' collection, displaying only 'restaurant_id', 'name', 'borough', and 'cuisine'. The output shows a list of restaurants, including 'Plaza Bagels & Deli' and many others. The user then enters a second query to find all restaurants in the 'Bronx' borough, and the output shows a list of restaurants in the Bronx, including 'Hanburgers', 'Wendy'S', 'Dj Reynolds Pub And Restaurant', etc.

```
rails@rails-System: ~
"borough" : "Staten Island",
"cuisine"  : "Delicatessen",
"grades"   : [
  {
    "date" : ISODate("2014-11-18T00:00:00Z"),
    "grade" : "A",
    "score" : 12
  },
  {
    "date" : ISODate("2013-11-07T00:00:00Z"),
    "grade" : "A",
    "score" : 12
  },
  {
    "date" : ISODate("2013-04-24T00:00:00Z"),
    "grade" : "A",
    "score" : 7
  },
  {
    "date" : ISODate("2012-03-20T00:00:00Z"),
    "grade" : "A",
    "score" : 5
  }
],
"name"     : "Plaza Bagels & Deli",
"restaurant_id" : "40364286"
}
Type "it" for more
> db.restaurants.find({}, {"restaurant_id":1,"name":1,"borough":1,"cuisine":1,"_id":0})
{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Brooklyn", "cuisine" : "American", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "borough" : "Queens", "cuisine" : "American", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
{ "borough" : "Brooklyn", "cuisine" : "American", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "borough" : "Bronx", "cuisine" : "American", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "borough" : "Brooklyn", "cuisine" : "American", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "borough" : "Manhattan", "cuisine" : "American", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "borough" : "Manhattan", "cuisine" : "American", "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "borough" : "Brooklyn", "cuisine" : "American", "name" : "The Movable Feast", "restaurant_id" : "40361606" }
{ "borough" : "Staten Island", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40363834" }
Type "it" for more
>
```

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

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

```
rails@rails-System: ~
{ "borough" : "Staten Island", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40363834" }
Type "it" for more
> db.restaurants.find({"borough":"Bronx"}).pretty()
{
  "_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("5947bc1fb60c37c179375c94"),
  "address" : {
    "building" : "2222",
    "coord" : [
      -73.84971759999999,
      40.8304811
    ],
    "street" : "Haviland Avenue",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "American ",
  "grades" : [
    {
```

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

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

```
rails@rails-System: ~
> db.restaurants.find({"grades.score":{$gt:90}}).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" : "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
    }
  ]
}
```

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

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

```
rails@rails-System: ~$ mongo
> use restaurants
switched to db restaurants
> db.restaurants.find({"borough": "Bronx"}).limit(5).pretty()
{
  "_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"
}
> db.restaurants.find({"borough": "Bronx"}).skip(5).pretty()
{
  "_id" : ObjectId("5947bc1fb60c37c179375c94"),
  "address" : {
    "building" : "2222",
    "coord" : [
      -73.84971759999999,
      40.8304811
    ],
    "street" : "Haviland Avenue",
    "zipcode" : "10462"
  },
  "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" : "Haviland",
  "restaurant_id" : "40357218"
}
```

- Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx
`db.restaurants.find({"borough": "Bronx"}).limit(5).skip(5)`

```
rails@rails-System: ~$ mongo
MongoDB shell version: 2.6.10
connecting to localhost:27020
> use restaurant
switched to db restaurant
> db.restaurants.find({"borough": "Bronx"})
{"_id": "5947bc1fb60c37c179375cea", "address": {"building": "1191", "coord": [-73.8513114, 40.8316981], "street": "Castle Hill Avenue", "zipcode": "10462"}, "borough": "Bronx", "cuisine": "Caf\u00e9/Coffee/Tea", "grades": [{"date": ISODate("2014-07-03T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2013-07-08T00:00:00Z"), "grade": "A", "score": 9}, {"date": ISODate("2013-02-11T00:00:00Z"), "grade": "A", "score": 11}, {"date": ISODate("2012-08-14T00:00:00Z"), "grade": "B", "score": 14}, {"date": ISODate("2011-08-23T00:00:00Z"), "grade": "A", "score": 12}], "name": "Lulu's Coffee Shop", "restaurant_id": "40366938"}, {"_id": "5947bc1fb60c37c179375cf0", "address": {"building": "2894", "coord": [-73.815175, 40.8137649], "street": "Schurz Ave", "zipcode": "10465"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-10-31T00:00:00Z"), "grade": "B", "score": 18}, {"date": ISODate("2014-02-20T00:00:00Z"), "grade": "A", "score": 4}, {"date": ISODate("2013-06-07T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2012-05-17T00:00:00Z"), "grade": "A", "score": 2}], "name": "Marina Delray", "restaurant_id": "40367161"}, {"_id": "5947bc1fb60c37c179375d3a", "address": {"building": "3533", "coord": [-73.9100082, 40.8863835], "street": "Johnson Avenue", "zipcode": "10463"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-02-20T00:00:00Z"), "grade": "A", "score": 5}, {"date": ISODate("2013-09-09T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2013-03-26T00:00:00Z"), "grade": "A", "score": 9}, {"date": ISODate("2013-03-08T00:00:00Z"), "grade": "P", "score": 2}, {"date": ISODate("2012-09-26T00:00:00Z"), "grade": "A", "score": 9}, {"date": ISODate("2012-04-13T00:00:00Z"), "grade": "A", "score": 11}, {"date": ISODate("2011-07-26T00:00:00Z"), "grade": "A", "score": 13}, {"date": ISODate("2011-02-26T00:00:00Z"), "grade": "A", "score": 13}], "name": "Blue Bay Restaurant", "restaurant_id": "40369887"}, {"_id": "5947bc1fb60c37c179375d42", "address": {"building": "591", "coord": [-73.790804, 40.8541848], "street": "City Island Ave", "zipcode": "10464"}, "borough": "Bronx", "cuisine": "Seafood", "grades": [{"date": ISODate("2014-12-11T00:00:00Z"), "grade": "A", "score": 13}, {"date": ISODate("2014-06-07T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2013-11-09T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2013-04-16T00:00:00Z"), "grade": "B", "score": 22}, {"date": ISODate("2012-02-29T00:00:00Z"), "grade": "A", "score": 9}], "name": "Seashore Restaurant", "restaurant_id": "40369299"}, {"_id": "5947bc1fb60c37c179375d4c", "address": {"building": "2375", "coord": [-73.85534559999999, 40.8426433], "street": "East Tremont Avenue", "zipcode": "10462"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-10-14T00:00:00Z"), "grade": "A", "score": 13}, {"date": ISODate("2014-04-08T00:00:00Z"), "grade": "A", "score": 8}, {"date": ISODate("2013-11-04T00:00:00Z"), "grade": "C", "score": 45}, {"date": ISODate("2013-05-23T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2012-10-18T00:00:00Z"), "grade": "B", "score": 15}, {"date": ISODate("2012-10-05T00:00:00Z"), "grade": "P", "score": 12}, {"date": ISODate("2012-05-14T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2011-12-21T00:00:00Z"), "grade": "A", "score": 11}], "name": "Bronx Grill", "restaurant_id": "40369601"}
>
```

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

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



```
rails@rails-System: ~$ mongo
> use restaurants
switched to db restaurants
> db.restaurants.find({"grades.score":{$gt:80,$lt:100}}).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" : "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"),

```

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

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

```
rails@rails-System: ~$ mongo
> use restaurants
switched to db restaurants
> db.restaurants.find({"address.coord":{"lt:-95.754168}}).pretty()
{
  "_id" : ObjectId("5947bc1fb60c37c179376288"),
  "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
    }
  ]
}
```

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


```
rails@rails-System: ~
"zipcode" : "10033"
},
"borough" : "Manhattan",
"cuisine" : "American ",
"grades" : [
  {
    "date" : ISODate("2014-11-15T00:00:00Z"),
    "grade" : "B",
    "score" : 20
  },
  {
    "date" : ISODate("2014-04-25T00:00:00Z"),
    "grade" : "A",
    "score" : 10
  },
  {
    "date" : ISODate("2014-03-28T00:00:00Z"),
    "grade" : "P",
    "score" : 15
  },
  {
    "date" : ISODate("2013-08-19T00:00:00Z"),
    "grade" : "B",
    "score" : 23
  },
  {
    "date" : ISODate("2013-02-20T00:00:00Z"),
    "grade" : "A",
    "score" : 3
  },
  {
    "date" : ISODate("2012-08-22T00:00:00Z"),
    "grade" : "B",
    "score" : 23
  },
  {
    "date" : ISODate("2012-01-30T00:00:00Z"),
    "grade" : "C",
    "score" : 48
  }
],
"name" : "Reynold'S Bar",
"restaurant_id" : "40365423"
}
Type "it" for more
> db.restaurants.update({"grades.0.score":{"$gte:10}},{"$set":{"grades.0.grade": "B"}})
2017-09-05T18:57:53.666+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:56
> db.restaurants.update({'grades.0.score':10},
...   {'$set':{'grades.0.grade':'B'}},{multi:true})
WriteResult({ "nMatched" : 407, "nUpserted" : 0, "nModified" : 407 })
>
```

- Write a MongoDB query to delete the restaurents in borough Bronx.

db.restaurants.remove({"borough":"Bronx"})

```
rails@rails-System: ~$ cat /dev/null
{
  "_id" : ObjectId("5947bc1fb60c37c1793765f4"),
  "address" : {
    "building" : "15259",
    "coord" : [
      -119.6368672,
      36.2504996
    ],
    "street" : "10 Avenue",
    "zipcode" : "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"
}
> db.restaurants.remove({"borough":"Bronx"})
WriteResult({"nRemoved" : 309 })
>
```