# Bài 2. Thực hiện các câu truy vấn các document trong collection như sau:

## 2.1. Thực hiện restore database qlsv lên server Mongodb

## Bài 2.2. Thực hiện các câu truy vấn các document trong collection như sau: Text Description automatically generated

### Text Description automatically generateddb.sinhvien.find()

### Text Description automatically generateddb.lophoc.find()

### Text Description automatically generateddb.sinhvien.find().count()

### db.lophoc.find().count()

Text

Description automatically generated with medium confidence

### db.sinhvien.find().limit(2)

Text

Description automatically generated

### Text Description automatically generateddb.lophoc.find().skip(2)

### db.sinhvien.find().skip(2).limit(2)

Text

Description automatically generated

### db.sinhvien.find({"ten":"Tí"})

Text

Description automatically generated

### db.sinhvien.find({"tuoi":{$gt:20}})

Text

Description automatically generated

### db.sinhvien.find({"lienLac.email":"ti@gmail.com"})

Text

Description automatically generated

### db.sinhvien.find({"monHoc":"Tin Hoc"})

Text

Description automatically generated

### db.lophoc.find({"Name":"DHKTPM16"})

Text

Description automatically generated

# Bài 4. Truy vấn collection restaurants

## 1. Hiển thị tất cả các documents có trong collection restaurants.

Text

Description automatically generated

## 2. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant\_id, name, borough and cuisine.

Text

Description automatically generated

## 3. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant\_id, name, borough and cuisine và không xuất field \_id.

Text

Description automatically generated

## 4. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant\_id, name, borough and zip code và không xuất field \_id.

Text

Description automatically generated

## 5. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị là Bronx.

Text

Description automatically generated

## 6. Hiển thị 5 documents đầu tiên có trong collection restaurants với field borough có giá trị là Bronx.

Text

Description automatically generated

## 7. Hiển thị 5 documents tiếp theo sau khi bỏ qua 5 documents đầu tiên có trong collection restaurants với field borough có giá trị là Bronx.

Text

Description automatically generated

## 8. Hiển thị tất cả các documents có trong collection restaurants với điều kiện score trong field grades lớn hơn 90

Text

Description automatically generated

## 9. Hiển thị tất cả các documents có trong collection restaurants với điều kiện score trong field grades lớn hơn 80 và nhỏ hơn 100.

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

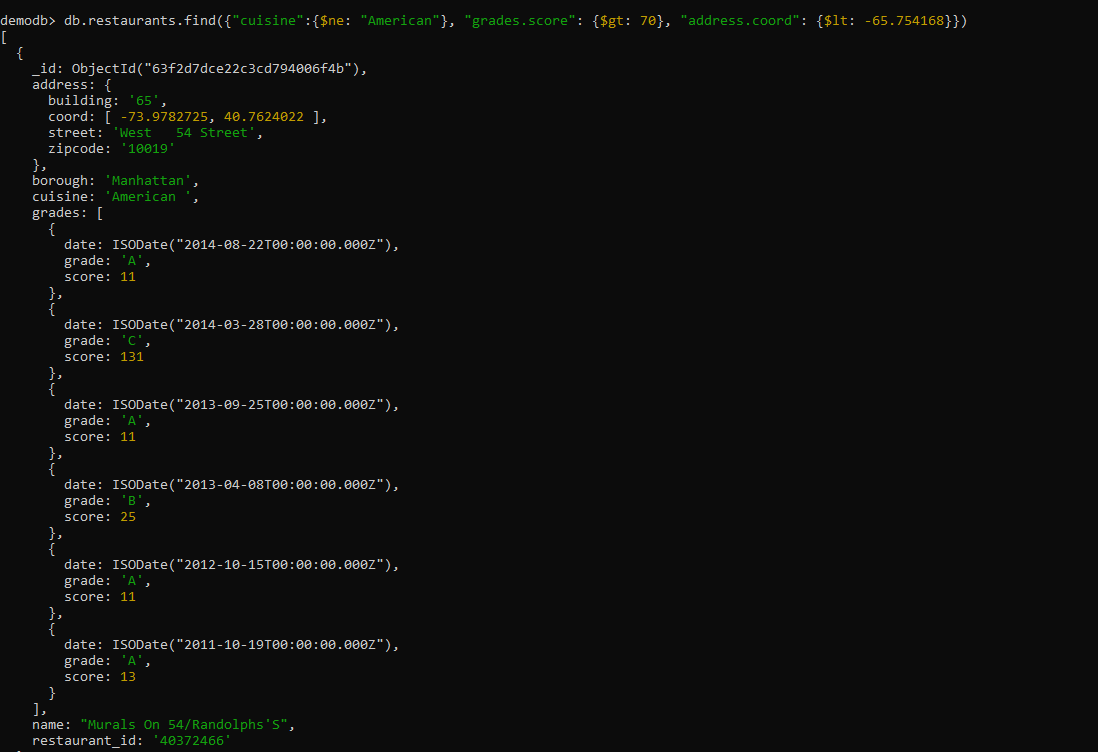


## 10. Hiển thị tất cả các documents có trong collection restaurants với điều kiện giá trị coord trong field address nhỏ hơn -95.754168.

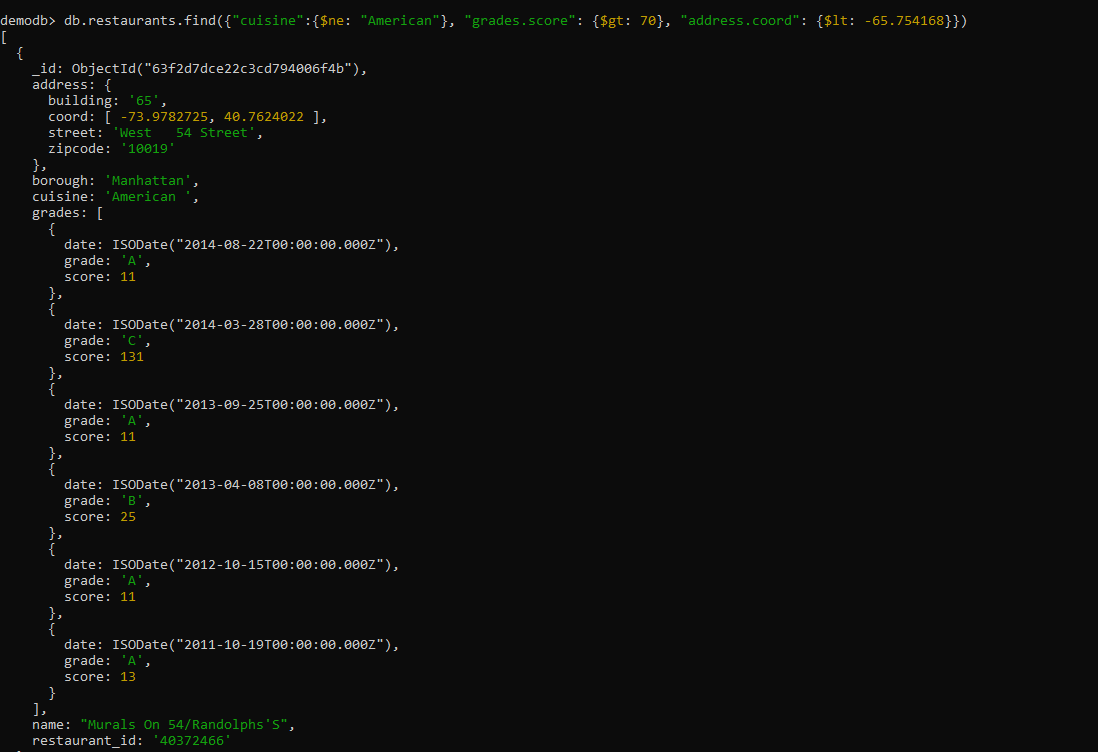


## 11. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và score của field grade lớn hơn 70 và giá trị coord trong field address nhỏ hơn -65.754168. \*\*\*\*\*\*

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

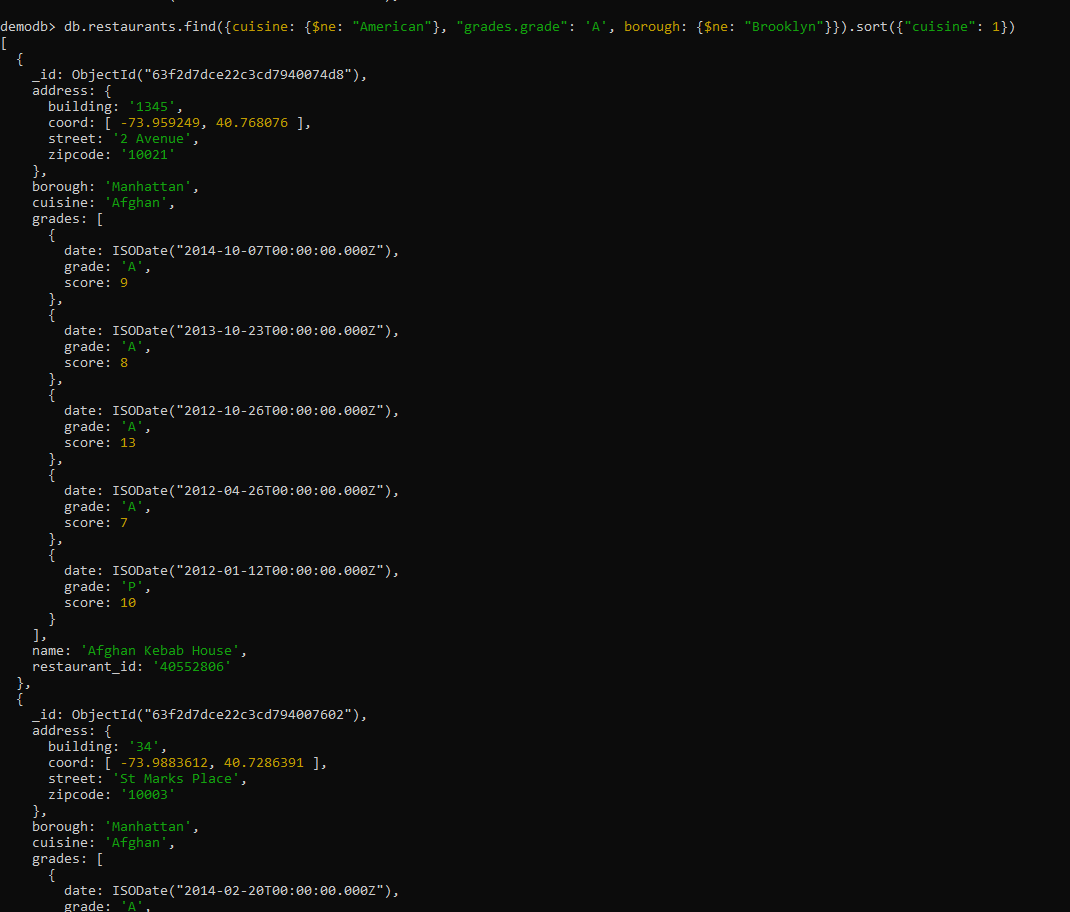


## 12. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và score của field grade lớn hơn 70 và giá trị coord trong field address nhỏ hơn -65.754168.



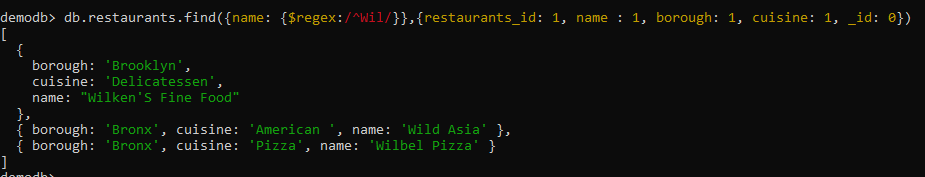
## 13. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và giá trị grade của field grade là 'A', và fiels borough không là Brooklyn. Sau đó sắp xếp các document theo thứ tự tăng dần của field cuisine.

db.restaurants.find({cuisine: {$ne: "American"}, "grades.grade": 'A', borough: {$ne: "Brooklyn"}}).sort({"cuisine": 1})



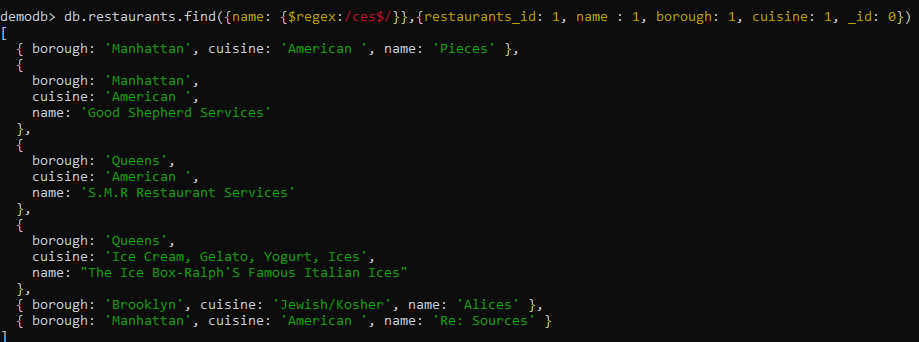
## 14. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự bắt đầu là 'Wil'.

db.restaurants.find({name: {$regex:/^Wil/}},{restaurants\_id: 1, name : 1, borough: 1, cuisine: 1, \_id: 0})

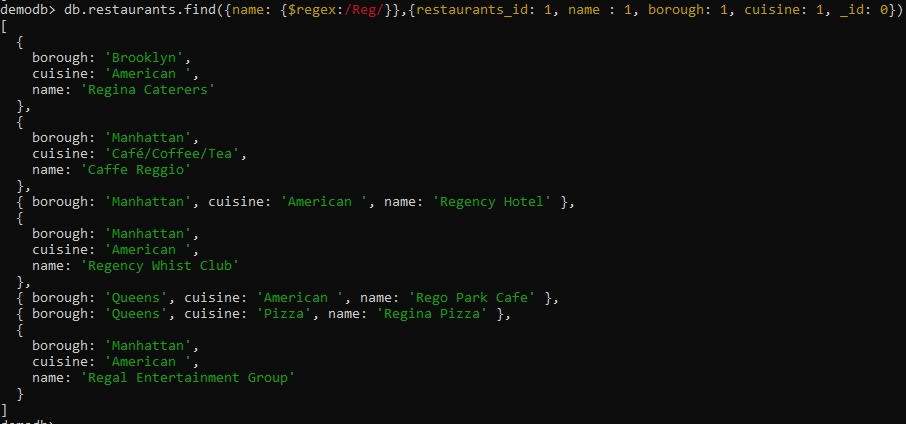


## 15. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự cuối cùng là 'ces'.

db.restaurants.find({name: {$regex:/ces$/}},{restaurants\_id: 1, name : 1, borough: 1, cuisine: 1, \_id: 0})



## 16. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự 'Reg'.

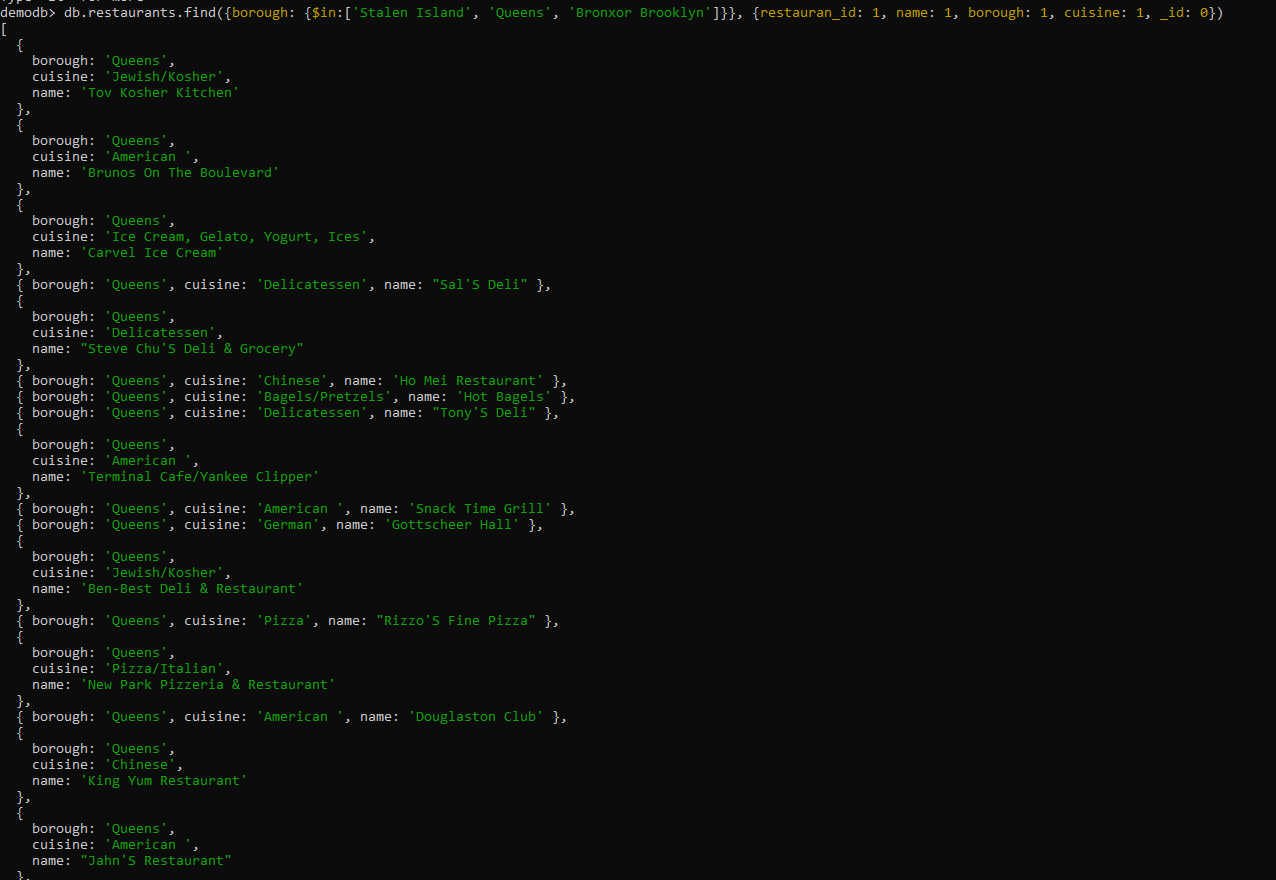


## 17. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị là Bronx và field cusine có giá trị là American hoặc Chinese.

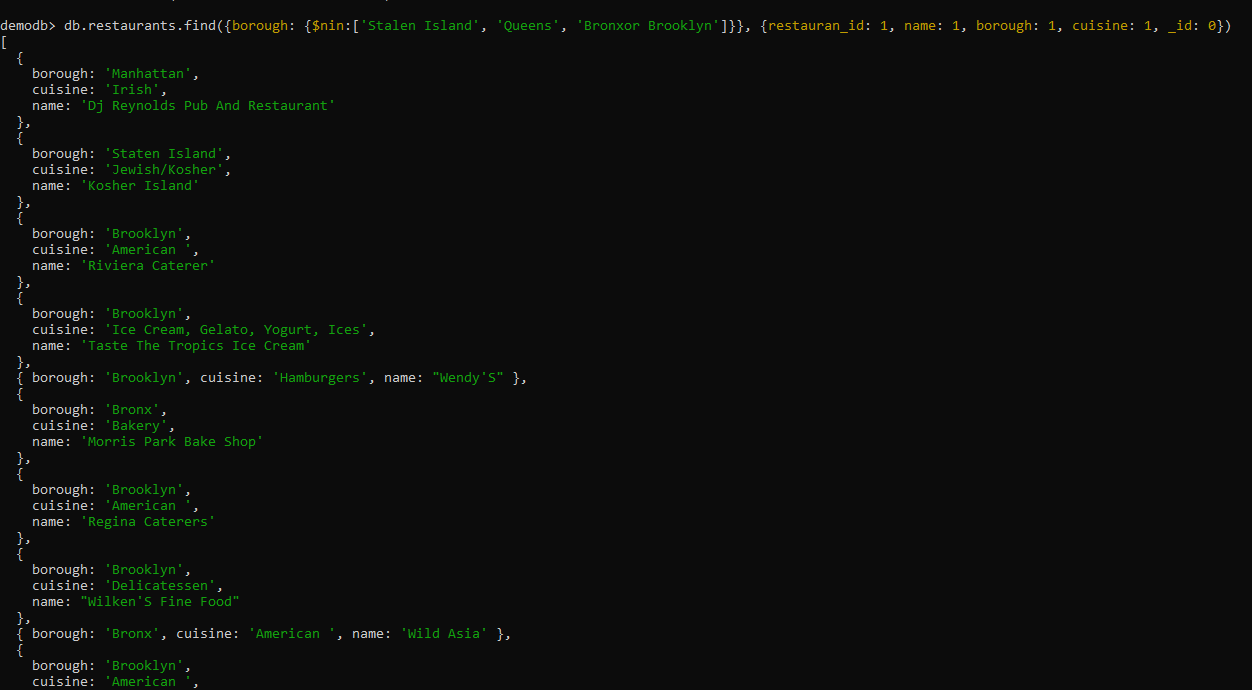


## 18. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị Staten Island or Queens or Bronxor Brooklyn, chỉ xuất các field restaurant Id, name, borough, cuisine.

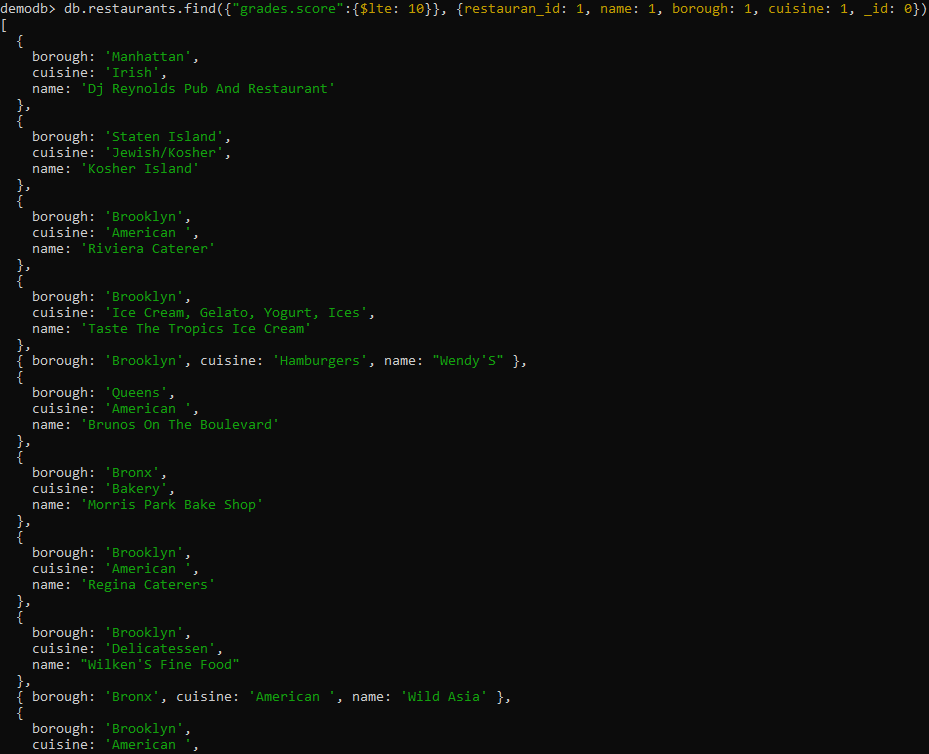
db.restaurants.find({borough: {$in:['Stalen Island', 'Queens', 'Bronxor Brooklyn']}}, {restauran\_id: 1, name: 1, borough: 1, cuisine: 1, \_id: 0})



## 19. Hiển thị tất cả các documents có trong collection restaurants với field borough có không là các giá trị Staten Island or Queens or Bronxor Brooklyn, chỉ xuất các field restaurant Id, name, borough, cuisine.



## 20. Hiển thị tất cả các documents có trong collection restaurants với giá trị score của field grades không lớn hơn 10, chỉ xuất các field restaurant Id, name, borough, cuisine.



# Bài 5: Truy vấn collection xe

1. Xuất toàn bộ danh sách xe máy.



2. Xuất ra danh sách xe là ô tô.

## 

## 3. Xuất danh sách xe máy Yamaha.

Text

Description automatically generated

## 4. Xuất danh sách xe máy Honda.

Text

Description automatically generated

## 5. Xuất danh sách xe máy Yamaha sản xuất năm 1992.

Text

Description automatically generated

## 6. Xuất danh sách xe máy Honda có giá từ 15.000.000 trở lên.



## 7. Xuất ra các danh sách các xe có Namsx là 1999.

Text

Description automatically generated

## 8. Xuất ra danh sách xe ô tô có sản xuất từ năm 2000.

Text

Description automatically generated

## 9. Đếm có bao nhiêu xe có năm sản xuất 1992.



## 10. Xuất ra các xe có giá trên 25.000.000 dưới 655.000.000.

Text

Description automatically generated

## 11. Cập nhật mã loại của ô tô thành ‘001CAR’.

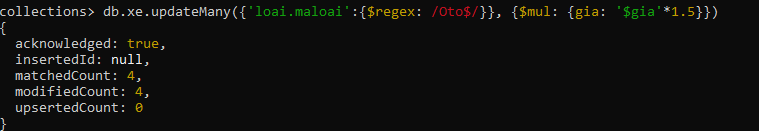
Text

Description automatically generated

## 12. Cập nhật giá của tất cả giá xe máy tăng thêm 1.000.000.



## 13. Cập nhật giá của tất cả giá ô tô lên 0.5 lần.

14. Cập nhật các document có “ten”:”Honda” với các Namsx: 2000

db.xe.updateMany({ten: 'Honda'}, {$set: {namsx: '2000'}})

Text

Description automatically generated

## 15. Thêm 1 thuộc tính màu xe có các giá trị đỏ, đen, trắng cho tất cả xe ô tô.

db.xe.updateMany({'loai.tenloai': 'Ô tô'}, {$set: {mauXe: ['Đỏ', 'Đen', 'Trắng']}})

Graphical user interface, text

Description automatically generated

## 16. Xuất ra tên và giá của tất cả xe ‘Yamaha’.

db.xe.find({ten: 'Yamaha'}, {ten: 1, gia: 1})

Text

Description automatically generated

# Bài 6. Truy vấn collection sinhvien, lop

## 1. Tìm kiếm theo mã: Tìm lớp khi biết mã lớp, tìm sinh viên khi biết mã sinh viên…

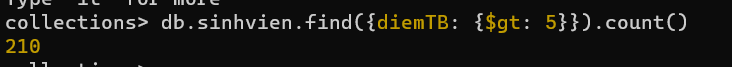
Text

Description automatically generated

Text

Description automatically generated

## 2. Đếm số sinh viên có điểm trung bình từ 5 trở lên.



## 3. Đếm số sinh viên có điểm trung bình từ 3.0 đến nhỏ hơn 6.5.



## 4. Liệt kê danh sách sinh viên không có số điện thoại hoặc email.

db.sinhvien.find({$or: [{'dsDienthoai.0': {$exists: false}}, {email: {$exists: false}}]})

Text

Description automatically generated

## 5. Liệt kê danh sách sinh viên có từ 2 số điện thoại trở lên.

db.sinhvien.find({'dsDienthoai.1': {$exists: true}})



6. Liệt kê danh sách sinh viên có lót chữ “Văn”, không phân biệt chữ hoa chữ thường.

Text

Description automatically generated

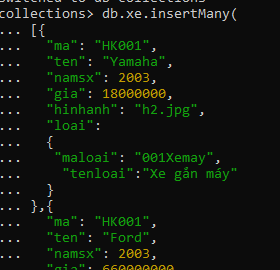
7. Liệt kê danh sách sinh viên có họ tên chứa chữ “Minh”, không phân biệt chữ hoa chữ thường.

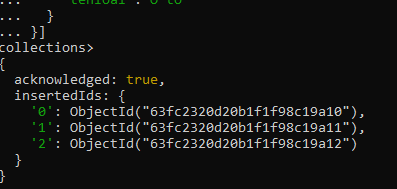
Text

Description automatically generated

# Bài 7. Insert/Update/Delete collection xe

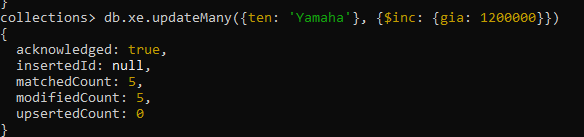
## 1. Thêm vào 3 document với các thuộc tính là xe của sv (tùy ý).





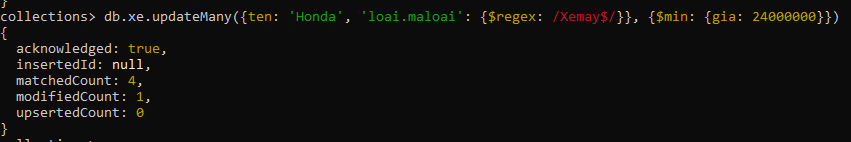
## 2. Tăng giá của tất cả xe Yamaha lên 1200000.

db.xe.updateMany({ten: 'Yamaha'}, {$inc: {gia: 1200000}})



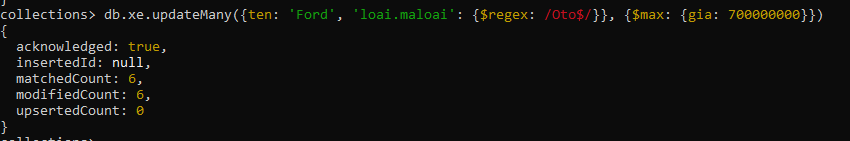
## 3. Cập nhật xe máy Honda có giá thấp nhất 24000000

db.xe.updateMany({ten: 'Honda', 'loai.maloai': {$regex: /Xemay$/}}, {$min: {gia: 24000000}})



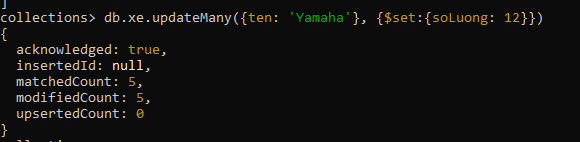
## 4. Cập nhật giá xe ô tô Ford có giá cao nhất 700000000.

db.xe.updateMany({ten: 'Ford', 'loai.maloai': {$regex: /Oto$/}}, {$max: {gia: 700000000}})



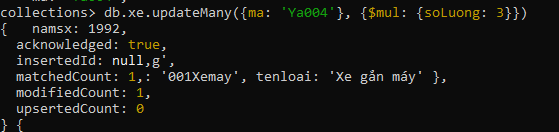
## 5. Cập nhật thêm thuộc tính số lượng là 12 cho xe yamaha

db.xe.updateMany({ten: 'Yamaha'}, {$set:{soLuong: 12}})



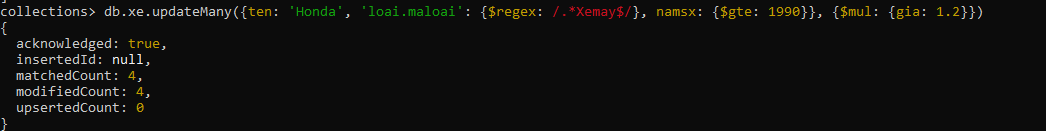
## 6. Cập nhật số lượng của xe Ya004 số lượng tăng gấp 3 lần.

db.xe.updateMany({ma: 'Ya004'}, {$mul: {soLuong: 3}})



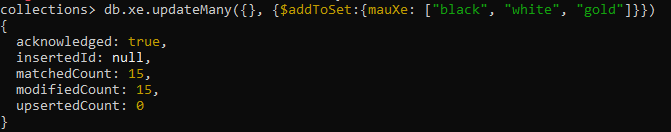
## 7. Cập nhật giá tăng 20% cho xe máy Honda sản xuất từ năm 1990 cho đến nay.

db.xe.updateMany({ten: 'Honda', 'loai.maloai': {$regex: /.\*Xemay$/}, namsx: {$gte: 1990}}, {$mul: {gia: 1.2}})



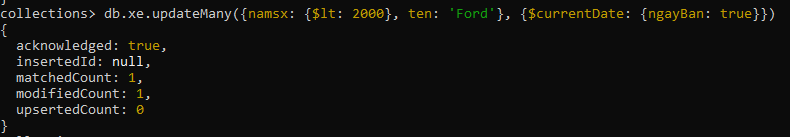
## 8. Tất cả xe cập nhật thêm thuộc tính mauxe(black,white,gold).

db.xe.updateMany({}, {$addToSet:{mauXe: ["black", "white", "gold"]}})



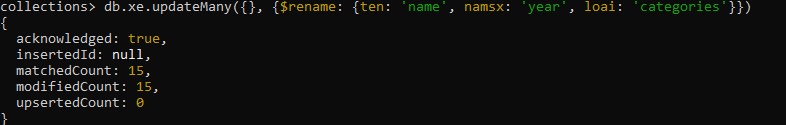
9. Tất cả xe Ford có năm sản xuất từ trước năm 2000 được cập nhật ngày bán là ngày hiện hành.

db.xe.update({namsx: {$lt: 2000}, ten: 'Ford'}, {$currentDate: {ngayBan: true}})



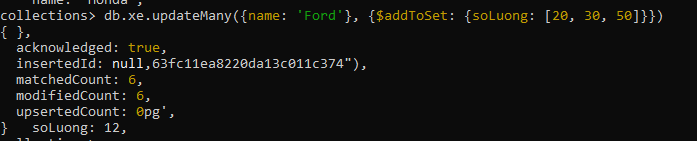
## 10. Đổi tất cả tên thuộc tính của tất cả các document trong collection Xe sang English (Ví dụ: tenname,namsxyear, loai categories, …)

db.xe.updateMany({}, {$rename: {ten: 'name', namsx: 'year', loai: 'categories'}})



## 11. Các xe Ford được cập nhật thêm thuộc tính như sau soluong(20,30,50)

db.xe.updateMany({name: 'Ford'}, {$addToSet: {soLuong: [20, 30, 50]}})



# Bài 8. Insert/Update/Delete collection restaurants

## 1. Thêm 2 documents restaurants với thông tin như sau:

db.restaurants.insertMany(

[{

address: {street: '63 Ky Dong', zipcode: '11374'},

cuisine: 'Vietnamese',

name: 'Lobster Bay',

restaurant\_id: '40356666',

like:23000,

share:4000

},

{

address: {street: '73rd Floor Landmark 81', zipcode: '11374'},

cuisine: 'Vietnamese',

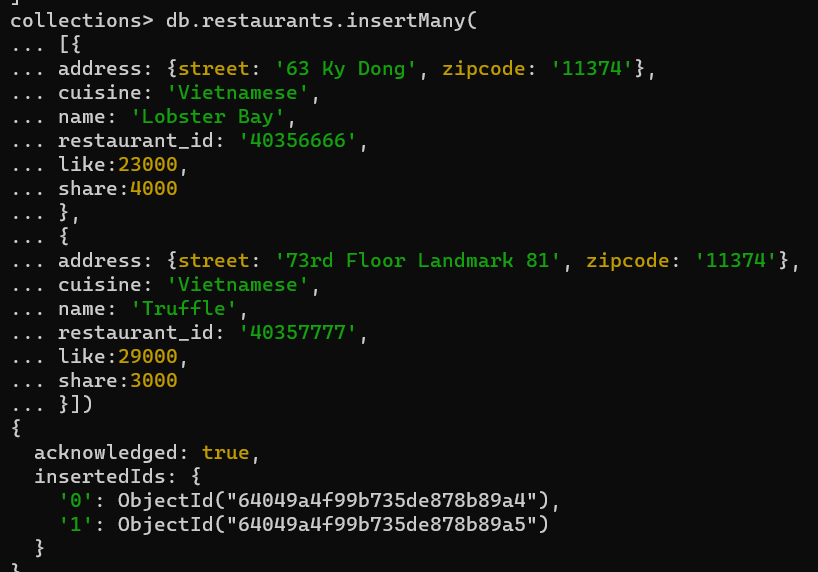
name: 'Truffle',

restaurant\_id: '40357777',

like:29000,

share:3000

}])



## 2. Cập nhật thuộc tính assigndate là ngày hiện hành cho các document có zipcode: ’11374’.

db.restaurants.updateMany({'address.zipcode':'11374'}, {$currentDate: {assigndate: true}})

Text

Description automatically generated

## 3. Cập nhật giá trị ‘Euro’ cho thuộc tính cuisine của các document có thuộc tính borough là 'Queens'.

db.restaurants.updateMany({borough: 'Queens'}, {$set:{cuisine: 'Euro'}})

Text

Description automatically generated

## 4. Tăng lượt like lên 100, lượt share lên 200 cho các nhà hàng Vietnamese.

collections> db.restaurants.updateMany({cuisine: 'Vietnamese'}, {$inc: {like: 100, share: 200}})

Text

Description automatically generated

## 5. Cập nhật điểm đánh giá cao nhất là 28 cho tất cả các document có zipcode là ‘10002’.

db.restaurants.updateMany({'address.zipcode': '10002'},{$min:{'grades.$[].score':28}})

Text

Description automatically generated

## 6. Trong document zipcode, cập nhật dân số (pop) tăng 10% của các city HADLEY thuộc state MA.