



Banco de Dados II

Prática - Atividade de Fixação de Conteúdo

Profª. Vanessa Souza

Assunto: MongoDB

Para essa lista de exercícios, considere o arquivo nomeado restaurants.json **Questão 1:**

Crie um novo banco de dados no servidor MongoDB.

Criação de um base dados chamada AT_PRATICA

```
> use AT_PRATICA
switched to db AT_PRATICA
>
```

Questão 2: Crie uma nova coleção no banco de dados criado na questão 1.

Criação da coleção mycollection e mostrando essa coleção.

```
switched to db AT_PRATICA
> db.createCollection("mycollection")
{ "ok" : 1 }
> show collections
mycollection
>
```

Questão 3: Importe os documentos arquivo restaurants.json para a coleção criada na questão 2. Use o mongoimport

Importação do arquivos json para base de dados AT_PRAT e coleção mycollection

```
C:\Users\giovany\95>mongoimport --collection=mycollection --db=AT_PRATICA --file "C:/Users/giovany\95/Desktop/restaurants.json"
2020-11-17T18:09:44.560-0300 connected to: mongodb://localhost/
2020-11-17T18:09:44.942-0300 3772 document(s) imported successfully. 0 document(s) failed to import.
```

Questão 4: Dê um count na coleção depois de executar a questão 3.

O count indica que há 3772 documentos na collection

```
> db.mycollection.count()
3772
```

Questão 5: Com base na coleção criada nas questões anteriores responda as questões: a) Listar todos os documentos da coleção.

```
db.mycollection.find().pretty()

  "_id" : ObjectId("5fb43c18931f57c01ae3e7de"),
  "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"
```

O comando `find()` (sem nenhuma passagem de parâmetros) retorna todos os documentos , acima temos o primeiro documento da lista.o comando `pretty()` disponibiliza uma visualização melhorada do documento.

b) Listar o id, nome, bairro (borough) e tipo de cozinha de todos os documentos da coleção. Devemos marcar os atributos que queremos marcar como `true`:

```
> db.mycollection.find({}, {"_id" : true , "name" : true , "borough" : true, "cuisine" : true}).pretty()
{
  "_id" : ObjectId("5fb43c18931f57c01ae3e7de"),
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "name" : "Riviera Caterer"
}
{
  "_id" : ObjectId("5fb43c18931f57c01ae3e7df"),
  "borough" : "Brooklyn",
  "cuisine" : "Hamburgers",
  "name" : "Wendy'S"
}
{
  "_id" : ObjectId("5fb43c18931f57c01ae3e7e0"),
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "name" : "Tov Kosher Kitchen"
}
{
  "_id" : ObjectId("5fb43c18931f57c01ae3e7e1"),
  "borough" : "Queens",
  "cuisine" : "American ",
  "name" : "Brunos On The Boulevard"
}
{
  "_id" : ObjectId("5fb43c18931f57c01ae3e7e2"),
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken'S Fine Food"
}
```

c) Listar o nome, bairro (borough) e tipo de cozinha de todos os documentos da coleção, excluindo-se o id.

Devemos setar id como false:

```
db.mycollection.find({}, {"_id" : false , "name" : true , "borough" : true, "cuisine" : true}).pretty()
{
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "name" : "Riviera Caterer"
}
{
  "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S" }
{
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "name" : "Tov Kosher Kitchen"
}
{
  "borough" : "Queens",
  "cuisine" : "American ",
  "name" : "Brunos On The Boulevard"
}
{
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken'S Fine Food"
}
{
  "borough" : "Staten Island",
  "cuisine" : "Jewish/Kosher",
  "name" : "Kosher Island"
}
{
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "name" : "Morris Park Bake Shop"
}
```

d) Listar todos os restaurantes do bairro Bronx.

```
> db.mycollection.find({"borough": "Bronx"}, {"_id" : false, "name" : true}).pretty()
{ "name" : "Morris Park Bake Shop" }
{ "name" : "Wild Asia" }
{ "name" : "Carvel Ice Cream" }
{ "name" : "Happy Garden" }
{ "name" : "Happy Garden" }
{ "name" : "Manhem Club" }
{ "name" : "The New Starling Athletic Club Of The Bronx" }
{ "name" : "Yankee Tavern" }
{ "name" : "Mcdwyers Pub" }
{ "name" : "The Punch Bowl" }
{ "name" : "Munchtime" }
{ "name" : "Ihop" }
{ "name" : "Lulu'S Coffee Shop" }
{ "name" : "Marina Delray" }
{ "name" : "The Lark'S Nest" }
{ "name" : "Terrace Cafe" }
{ "name" : "African Terrace" }
{ "name" : "Cool Zone" }
{ "name" : "Beaver Pond" }
{ "name" : "African Market (Baboon Cafe)" }
```

e) Listar os primeiros 5 restaurantes do bairro Bronx.

```
> db.mycollection.find({"borough": "Bronx"}, {"_id" : false, "name" : true}).pretty().limit(5)
{ "name" : "Morris Park Bake Shop" }
{ "name" : "Wild Asia" }
{ "name" : "Carvel Ice Cream" }
{ "name" : "Happy Garden" }
{ "name" : "Happy Garden" }
>
```

f) Listar os restaurantes que tenham alguma nota maior que 90.

```
> db.mycollection.find({"grades.score" : {$gt :90}}, {"name" : true , "_id" : false}).pretty()
{ "name" : "Murals On 54/Randolphs'S" }
{ "name" : "Gandhi" }
{ "name" : "Bella Napoli" }
>
```

g) Listar os restaurantes que tenham alguma nota maior que 80 e menor que 100.

```
> db.mycollection.find({"grades.score" : {$gt :80,$lt :100}}, {"name" : true , "_id" : false}).pretty()
{ "name" : "Murals On 54/Randolphs'S" }
{ "name" : "Gandhi" }
{ "name" : "Bella Napoli" }
{ "name" : "West 79Th Street Boat Basin Cafe" }
>
```

h) Listar os restaurantes que estejam localizados numa latitude menor que - 95.754168.

```
> db.mycollection.find({"address.coord.1" : {$lt : -95.754168}}, {"name" : true , "_id" : false}).pretty()
```

O mongo não retornou nenhum restaurante com essa consulta

i) Listar os restaurantes que não preparem cozinha 'American', tenham nota menor que 70 e latitude menor que -65.754168. Os documentos devem ser mostrados de acordo com o tipo de cozinha, em ordem descendente.

```
> db.mycollection.find({"cuisine" : "American ", "grades.score" : {$lt :70}, "address.coord.1" : {$lt : -65.754168}}, {"name" : true, "_id" : false}).sort({"cuisine" : -1}).pretty()
```

O mongo não retornou nenhum restaurante com essa consulta

j) Listar o id, nome, bairro e cozinha dos restaurantes que não estejam nos bairros 'Staten Island',

‘Queens’ ou ‘Brooklyn’

```
db.mycollection.find({borough:{$not: {$regex: "Staten Island"}},$not: {$regex: "Queens"},$not: {$regex: "Brooklyn"}},{ "_id" : true,"name" : true,"borough" : true}).pretty()
```

k) Listar a quantidade de restaurantes por tipo de cozinha;

```
> db.mycollection.aggregate([{$group :{ id:"$cuisine",count:{$sum : 1}}}]])
```

```
{ "_id" : "Salads", "count" : 1 }
{ "_id" : "Peruvian", "count" : 2 }
{ "_id" : "Moroccan", "count" : 1 }
{ "_id" : "Café/Coffee/Tea", "count" : 1 }
{ "_id" : "Brazilian", "count" : 7 }
{ "_id" : "Tapas", "count" : 4 }
```

l) Listar a quantidade de restaurantes de comida “Mediterranean” por bairro;

```
> db.mycollection.aggregate([{$match: {"cuisine": "Mediterranean"}},{$group: {_id:"$borough",total:{$sum: 1}}}]])
```

```
{ "_id" : "Queens", "total" : 2 }
{ "_id" : "Manhattan", "total" : 11 }
{ "_id" : "Brooklyn", "total" : 3 }
```

m) Alterar o nome do atributo ‘borough’ para ‘bairro’ em todos os documentos

```
> db.mycollection.update({},{$rename: {'borough' : 'bairro'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

n) Alterar o zipcode dos restaurantes cujo valor é “11215” para “112215”

```
db.mycollection.update({"zipcode" : "11215"},{$set : {"zipcode" : "112215"}})
```

o) Apagar os documentos referentes a cozinha ‘Barbecue’

```
> db.mycollection.remove({cuisine: "Barbecue"})
WriteResult({ "nRemoved" : 8 })
```