

Ficha prática 6

2)

a) Devolver todos os produtos

```
>_MONGOSH
> db.inventory.find({})
< {
  _id: ObjectId('671c2775d2d124e6dc508da3'),
  Marca: 'Samsung',
  Modelo: 'Galaxy',
  Tipo: 'Tablet'
}
{
  _id: ObjectId('671c2775d2d124e6dc508da4'),
  Marca: 'HUAWEI',
  Modelo: 'P40',
  Tipo: 'Smartphone',
  Cores: 'Azul, Preto, Vermelho'
}
{
  _id: ObjectId('671c29afd2d124e6dc508da5'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: '6.4',
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  }
}
}
Collection>
```

b) Devolver os produtos do tipo smartphone

```
}
> db.inventory.find({ Tipo: "Smartphone"})
< {
  _id: ObjectId('671c2775d2d124e6dc508da4'),
  Marca: 'HUAWEI',
  Modelo: 'P40',
  Tipo: 'Smartphone',
  Cores: 'Azul, Preto, Vermelho'
}
{
  _id: ObjectId('671c29afd2d124e6dc508da5'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: '6.4',
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  }
}
}
Collection>
```

c) Devolver os produtos do tipo smartphone e da marca Huawei

```
db.inventory.find({ Tipo: "Smartphone", Marca: "HUAWEI"})
{
  _id: ObjectId('671c2775d2d124e6dc508da4'),
  Marca: 'HUAWEI',
  Modelo: 'P40',
  Tipo: 'Smartphone',
  Cores: 'Azul, Preto, Vermelho'
}
{
  _id: ObjectId('671c29afd2d124e6dc508da5'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: '6.4',
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  }
}
```

d) Devolver os produtos que possuem um ecrã: LCD (consulta sobre um documento embutido)

```
db.inventory.find({"Ecrã.Tecnologia": "LCD"})
{
  _id: ObjectId('671c29afd2d124e6dc508da5'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: '6.4',
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  }
}
```

e) Devolver os produtos que possuem um ecrã com mais de 5.5 polegadas (consulta sobre um documento embutido)

```
db.inventory.find({"Ecrã.Diagonal": {$gt: 5.5}})
{
  _id: ObjectId('671c2ff3d2d124e6dc508da6'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: 6.4,
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  }
}
```

f) Devolver os produtos que possuem a cor Azul (consulta sobre um array)

```
> db.inventory.find({Cores: /Azul/})
< {
  _id: ObjectId('671c301bd2d124e6dc508da8'),
  Marca: 'HUAWEI',
  Modelo: 'P40',
  Tipo: 'Smartphone',
  Cores: [
    'Azul, Preto, Vermelho'
  ]
}
```

3. Utilizando a mongo Shell, atualize os documentos da coleção para:

a) Adicionar um novo campo a todos os documentos para a quantidade de stock disponível para cada um;

```
< {
  _id: ObjectId('671c2775d2d124e6dc508da3'),
  Marca: 'Samsung',
  Modelo: 'Galaxy',
  Tipo: 'Tablet',
  Stock: 50
}
{
  _id: ObjectId('671c2ff3d2d124e6dc508da6'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: 6.4,
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  },
  Stock: 50
}
```

```
db.inventory.updateMany(
  { },
  { $set: {"Stock": 50} }
);
```

b) Atualize os documentos com valores de stock à sua escolha;

```
{
  _id: ObjectId('671c2775d2d124e6dc508da3'),
  Marca: 'Samsung',
  Modelo: 'Galaxy',
  Tipo: 'Tablet',
  Stock: 20
}
{
  _id: ObjectId('671c2ff3d2d124e6dc508da6'),
  Marca: 'HUAWEI',
  Modelo: 'P40 lite',
  Tipo: 'Smartphone',
  'Ecrã': {
    Diagonal: 6.4,
    Tecnologia: 'LCD',
    'Resolução': 'FULL HD+'
  },
  Stock: 0
}
```

```
db.inventory.updateOne(
  {"Modelo": "P40 " },
  { $set: {"Stock": 5} }
);
```

c) Atualize os documentos de forma a adicionar um novo campo: `nível_stock` que deverá conter os valores: “sem stock”, “pouco stock” ou “em stock”. O valor deverá ser atribuído de acordo as seguintes regras:

o “sem stock”: 0

o “pouco stock”: entre 1 e 10

o “em stock”: mais do que 10.

```
{
  _id: ObjectId('671c2775d2d124e6dc508da3'),
  Marca: 'Samsung',
  Modelo: 'Galaxy',
  Tipo: 'Tablet',
  Stock: 20,
  'nível_stock': 'em stock'
}
[
  {
    _id: ObjectId('671c2ff3d2d124e6dc508da6'),
    Marca: 'HUAWEI',
    Modelo: 'P40 lite',
    Tipo: 'Smartphone',
    'Ecrã': {
      Diagonal: 6.4,
      Tecnologia: 'LCD',
      'Resolução': 'FULL HD+'
    },
    Stock: 0,
    'nível_stock': 'sem stock'
  }
]
```

```
> db.inventory.find().forEach(function(doc) {
  let nível_stock = "";

  if (doc.Stock === 0) {
    nível_stock = "sem stock";
  } else if (doc.Stock > 0 && doc.Stock <= 10) {
    nível_stock = "pouco stock";
  } else if (doc.Stock > 10) {
    nível_stock = "em stock";
  }

  db.inventory.updateOne(
    { _id: doc._id },
    { $set: { "nível_stock": nível_stock } }
  );
});
```

Exe 4. X

Exe 5

- a) Todos os documentos integrados na coleção de dados;

```
> db.collection.find({})
< {
  _id: ObjectId('671ced2d315b268a1bd7c9b'),
  address: {
    building: '1987',
    coord: [
      -73.856877,
      40.848447
    ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: 2014-03-03T00:00:00.000Z,
      grade: 'A',
      score: 2
    },
    {
      date: 2013-09-11T00:00:00.000Z,
      grade: 'A',
      score: 6
    }
  ],
}
```

- b) Os valores dos campos “restaurant_id”, “name”, “borough” e “cuisine”, para todos os documentos contidos na coleção.

```
db.collection.find({}, {restaurant_id: 1, name: 1, cuisine: 1, borough: 1})
{
  _id: ObjectId('671ced2d315b268a1bd7c9b'),
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
{
  _id: ObjectId('671ced2d315b268a1bd7c9c'),
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: 'Wendy'S',
  restaurant_id: '30112340'
}
```

- c) Todos os restaurantes que estão localizados no bairro (“borough”) “Bronx”.

```
db.collection.find({}, {borough: "Bronx"})
{
  _id: ObjectId('671ced2d315b268a1bd7c9b'),
  borough: 'Bronx'
}
{
  _id: ObjectId('671ced2d315b268a1bd7c9c'),
  borough: 'Bronx'
}
{
  _id: ObjectId('671ced2d315b268a1bd7c9d'),
  borough: 'Bronx'
}
{
  _id: ObjectId('671ced2d315b268a1bd7c9e'),
  borough: 'Bronx'
}
{
  _id: ObjectId('671ced2d315b268a1bd7c9f'),
  borough: 'Bronx'
}
{
  _id: ObjectId('671ced2d315b268a1bd7ca0'),
  borough: 'Bronx'
}
```

- d) Os restaurantes (“Id”, “name”, “borough” e “cuisine”) que contenham 'er' nas duas últimas letras do seu nome e que estejam situados em “Queens”.

```
db.collection.find({name: {$regex: "er$", $options: "i"}, borough: "Queens"},{
  restaurant_id: 1, borough: 1, name: 1, cuisine: 1})
{
  _id: ObjectId('671ced2d315b268a1b1d7cce'),
  borough: 'Queens',
  cuisine: 'American',
  name: 'Terminal Cafe/Yankee Clipper',
  restaurant_id: '40364262'
}
{
  _id: ObjectId('671ced2d315b268a1b1d7ddd'),
  borough: 'Queens',
  cuisine: 'American',
  name: 'Sizzler',
  restaurant_id: '40370946'
}
```

- e) Os restaurantes que não têm definidas as suas coordenadas geográficas (“coord”).

```
db.collection.find({$or: [
  { "address.coord": { $size: 0 } }
]})
{
  _id: ObjectId('671ced2e315b268a1b1d8d1b'),
  address: {
    building: '0',
    coord: [],
    street: 'Wards Island/2Fl',
    zipcode: '10057'
  },
}
```

Exe 6

Aplique o *schema* disponibilizado na coleção criada no exercício anterior. Teste a inserção de documentos inválidos.