MongoDB

Aula 7:
Aggregate Framework - Arrays

Banco de Dados 2 (BD2)

Instituto J&F Germinare TECH

Lembrando nosso techBank

Coleções com documentos de Clientes, Contas e Endereços.

```
techBank> show collections
clientes
contas
enderecos
techBank> db.clientes.count()
250
techBank> db.contas.count()
262
techBank> db.enderecos.count()
258
```

Problema

Quantos clientes tem seguros contratados na nossa base?

```
techBank> db.clientes.find({seguros:{$exists:true}}).count()
36
```

E se a pergunta agora mudar, quantos seguros foram vendidos? Veja, a pergunta não é contar quantos clientes tem o campo seguros, mas sim quantos seguros tem ao todo, tem alguns clientes que tem mais de um seguro contrato.

$unwind

Você sabe o que significa essa palavra?

Esse operador desconstrói um campo de matriz dos documentos de entrada para produzir um documento para cada elemento. Cada documento de saída é o documento de entrada com o valor do campo da matriz substituído pelo elemento.

```
techBank> db.clientes.aggregate([{$unwind:"$seguros"}])
[
  {
    _id: 6,
```

Percebeu que esse **_id: 6** aparece mais de uma vez?

$unwind e $count

Ok, você teve a saída, mas eu quero saber quantos seguros foram ao todo, como fazer?

```
techBank> db.clientes.aggregate([{$unwind:"$seguros"},
... {$count:"Quantidade de seguros contratados"}])
[ { 'Quantidade de seguros contratados': 75 } ]
```

Query:

db.clientes.aggregate([{**$unwind:**"$seguros"},
... {**$count:**"Quantidade de seguros contratados"}])

$unwind e $sortByCount

Quantos desses seguros foram vendidos para Mulheres e quantos para Homens?

```
techBank> db.clientes.aggregate([{$unwind:"$seguros"},
... {$sortByCount:"$genero"}])
[ { _id: 'Masculino', count: 46 }, { _id: 'Feminino', count: 29 } ]
```

Query:

db.clientes.aggregate([{$unwind:"$seguros"},
... {$sortByCount:"$genero"}])

$match

Um operador de filtragem, ou seja, ele filtra os documentos para passar apenas os documentos que correspondem à(s) condição(ões) especificada(s) para o próximo estágio do pipeline.

Por exemplo, quais são os clientes da agencia **1066**?

```
techBank> db.contas.aggregate([{$match:{agencia:1066}}])
[
  {
    _id: 154,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta corrente',
    cpf: '170.029.883-60',
    valor: ''
  },
  {
    _id: 153,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta salário',
    cpf: '170.029.883-60',
    valor: 7846.66
  }
]
```

$match

Ok, a query com o **$match** é igual ao find() então qual a vantagem?

```
techBank> db.contas.aggregate([{$match:{agencia:1066}}])
[
  {
    _id: 154,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta corrente',
    cpf: '170.029.883-60',
    valor: ''
  },
  {
    _id: 153,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta salário',
    cpf: '170.029.883-60',
    valor: 7846.66
  }
]
```

```
techBank> db.contas.find({agencia:1066})
[
  {
    _id: 154,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta corrente',
    cpf: '170.029.883-60',
    valor: ''
  },
  {
    _id: 153,
    id_cliente: 145,
    numero_conta: '261090-6',
    agencia: 1066,
    tipo: 'Conta salário',
    cpf: '170.029.883-60',
    valor: 7846.66
  }
]
```

$match e $count

Se quiser saber quais são as contas do tipo “Conta salário” que tem mais que 8000 no campo valor?

```
techBank> db.contas.find({tipo:"Conta salário", valor:{$gt:8500}}).count()
12
```

Tente pensar em ter esse resultado, porem agora usando o operador $match e o $count.

```
techBank> db.contas.aggregate([{$match: {$and:[ {tipo:"Conta salário"},{valor:{$gt:8000}} ]}},
... {$count:"Total de Contas Salários com mais de R$8000 de saldo"}])
[ { 'Total de Contas Salários com mais de R$8000 de saldo': 16 } ]
```

$match e $group

Tem até uma maneira mais correta e mais difícil, que é. Imagine que você quer na resposta já o que é buscado, então devemos usar o **$group** também.

```
techBank> db.contas.aggregate([
... {$match: {$and:[{tipo:"Conta salário"},{valor:{$gt:8000}}]}},
... {$group:{_id:"$tipo", "Quantidade":{$count:{}}}}
... ])
[ { _id: 'Conta salário', Quantidade: 16 } ]
```

Legal né?

☺

$match e $group

Mas se a gente quiser saber de cada conta, quantas tem mais do que 8000 no valor?

```
techBank> db.contas.aggregate([
... {$match:{valor:{$gt:8000}}},
... {$group:{_id:"$tipo", "Quantidade":{$count:{}}}}
... ])
[
  { _id: 'Conta salário', Quantidade: 16 },
  { _id: 'Conta poupança', Quantidade: 26 },
  { _id: 'Conta corrente', Quantidade: 19 }
]
```

Pulo do grilo

Ok! Aprendeu o **$match**? Agora vamos ver algo muito necessários, que é a junção de campos em diferentes coleções.