

MongoDB

Aula 8: Lookup e Project

Banco de Dados 2 (BD2)



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
```

Join



Na nossa base tem a collection contas que tem um campo id_cliente, certo?

Se quiser informações sobre o cliente de uma conta como fazer? No SQL a gente poderia fazer um JOIN, usando as relações através das chaves, mas e aqui?

Quem aqui usa o EXCEL, vai lembrar do PROCV ou PROCX, quem usa o PowerBI vai lembrar do LOOKUPVALUE, quem usa mongo vai lembrar do **LOOKUP**.

\$lookup



Sintaxe

```
db.clientes.aggregate( [  
  {$lookup: {  
    from: "contas",  
    localField: "_id",  
    foreignField: "id_cliente",  
    as: "clientesContas"  
  }  
},  
  ...  
] )
```

{ \$lookup: }

- from:** collection que vai buscar ,
- localField:** campo na collection que usa o aggregate,
- foreignField:** campo que vai buscar o mesmo valor,
- as:** alias para usar em outros estágios ou na saída.

}...

\$lookup



Faça a query abaixo e entenda como vai vir a saída.

```
techBank> db.clientes.aggregate([
... {$lookup: {from:"contas", localField: "_id", foreignField: "id_cliente", as: "clientesContas"}}
... ])
```

```
db.clientes.aggregate( [
  {$lookup: {
    from: "contas",
    localField: "_id",
    foreignField: "id_cliente",
    as: "clientesContas"
  }
} ] )
```

\$lookup



Saída: vai aparecer um novo campo (com o nome do alias), como um array contendo todas as informações presente na outra collection.

```
{
  _id: 2,
  nome: 'Calebe Danilo Roberto Figueiredo',
  cpf: '028.796.232-60',
  data_nascimento: ISODate("1986-10-01T21:00:00.000Z"),
  genero: 'Masculino',
  profissao: 'Supervisor de vendas comercial',
  status_civil: 'Viúvo(a)',
  clientesContas: [
    {
      _id: 2,
      id_cliente: 2,
      numero_conta: '200693-6',
      agencia: 1545,
      tipo: 'Conta corrente',
      cpf: '028.796.232-60',
      valor: 8288.56
    }
  ]
},
```

\$lookup



Faça novamente essa query, só que agora, ao invés de buscar pelo **id_cliente**, busque pelo **CPF**, veja se muda algo na saída.

```
techBank> db.clientes.aggregate([ { $lookup: { from: "contas", localField: "cpf", foreignField: "cpf", as: "clientesContas" } }])
[
  {
    _id: 2,
    nome: 'Calebe Danilo Roberto Figueiredo',
    cpf: '028.796.232-60',
    data_nascimento: ISODate("1986-10-01T21:00:00.000Z"),
    genero: 'Masculino',
    profissao: 'Supervisor de vendas comercial',
    status_civil: 'Viúvo(a)',
    clientesContas: [
      {
        _id: 2,
        id_cliente: 2,
        numero_conta: '200693-6',
        agencia: 1545,
        tipo: 'Conta corrente',
        cpf: '028.796.232-60',
        valor: 8288.56
      }
    ]
  }
]
```

\$lookup



Se quisermos saber quantas contas a cliente com **_id:30** tem?

Gerar a saída conforme abaixo!

```
[
  {
    nome: 'Nicole Heloise Débora Pinto',
    cpf: '780.986.023-20',
    data_nascimento: ISODate("2001-04-27T21:00:00.000Z"),
    genero: 'Feminino',
    profissao: 'Operador de máquinas',
    status_civil: 'Separado(a)',
    numero_de_contas: 3
  }
]
```




\$lookup + \$addFields

Para esse resultado podemos usar o estágio de **\$addFields**.

Mas mesmo assim, se você usar apenas isso vai sair o resultado anterior?

```
db.clientes.aggregate([
  { $match: { _id: 30 } },
  { $lookup: {
    from: "contas",
    localField: "_id",
    foreignField: "id_cliente",
    as: "contas_do_cliente"
  } },
  { $addFields: {
    numero_de_contas: { $size: "$contas_do_cliente" }
  } }, ....
])
```

\$project

NÃO!

Precisamos adicionar
umas restrições na
saída, para isso
podemos usar o
\$project

```
as: contas_do_cliente
} },
{ $addField: {
  numero_de_contas: { $size: "$contas_do_cliente" }
} },
{ $project: {
  _id: 0,
  numero_de_contas: 1,
  nome: 1,
  cpf: 1,
  data_nascimento: 1,
  genero: 1,
  profissao: 1,
  status_civil: 1
} } ])
```

\$project



Então, uma das possibilidades de uso do **\$project** é para restringir campos a serem exibidos.

Observe essa query abaixo, o que ela faz?

```
techBank> db.clientes.aggregate([
...  {$lookup:{from:"contas", localField:"cpf", foreignField:"cpf", as:"clientesConta"}},
...  {$project:{_id:0, data_nascimento:0, genero:0, profissao:0, status_civil:0}}
... ])
```

\$project



Vamos exibir os 4 clientes que tem mais endereços cadastrados e são os mais velhos.

Mostre apenas:

- Nome do cliente:
- Data de nascimento
- Cidade e estado do endereço.

\$project

Vamos exibir os 4 clientes

Mostre apenas:

- Nome do cliente:
- Data de nascimento
- Cidade e estado do en

Saída ->

```
[
  {
    nome: 'Emanuel Tomás Campos',
    data_nascimento: ISODate('1944-08-21T21:00:00.000Z'),
    endereco: [
      { cidade: 'Recife', estado: 'PE' },
      { cidade: 'Recife', estado: 'PE' }
    ]
  },
  {
    nome: 'Noah Pedro Barros',
    data_nascimento: ISODate('1958-05-22T21:00:00.000Z'),
    endereco: [
      { cidade: 'Belford Roxo', estado: 'RJ' },
      { cidade: 'Bagé', estado: 'RS' }
    ]
  },
  {
    nome: 'Bento Thales Matheus Araújo',
    data_nascimento: ISODate('1975-05-06T21:00:00.000Z'),
    endereco: [
      { cidade: 'Campo Grande', estado: 'MS' },
      { cidade: 'Macapá', estado: 'AP' }
    ]
  },
  {
    nome: 'Augusto Otávio Iago Jesus',
    data_nascimento: ISODate('1979-04-24T21:00:00.000Z'),
    endereco: [
      { cidade: 'Brasília', estado: 'DF' },
      { cidade: 'Bayeux', estado: 'PB' }
    ]
  }
]
```

mais velhos.



\$project



Vamos exibir os 4 clientes que tem mais endereços cadastrados e são os mais velhos.

Mostre apenas:

- Nome do cliente:
- Data de nascimento
- Cidade e estado do endereço.

```
techBank> db.clientes.aggregate([
...     {$lookup:{from:"enderecos", localField:"_id", foreignField:"id_cliente", as:"endereco"}},
...     {$addFields:{"endereços":{$size:"$endereco"}}},
...     {$sort:{"endereços":-1, "data_nascimento":1}},
...     {$limit:4},
...     {$project:{
...         _id:0, nome:1, data_nascimento:1, "endereco.cidade":1, "endereco.estado":1
...     }}
... ])
```

Operadores do \$project



Junto com o `$project`, também podemos usar vários operadores, sendo eles distribuídos entre:

- Aritméticos;
- Strings;
- Data;
- Acumuladores;
- Condicional e outros.