

Banco de dados não relacionais



mongoDB®

Aggregations

Estágio \$lookup

Prof. Me. Ricardo Resende de Mendonça
ricardo.mendonca@online.uscs.edu.br

Prof. Me. Renato Carioca Duarte
renato.duarte@online.uscs.edu.br

Introdução

Aggregations

O pipeline de agregação é uma sequência de estágios que processam os documentos de entrada de acordo com operações específicas, passando os resultados de um estágio para o próximo. Cada estágio do pipeline pode executar uma variedade de operações, como filtragem, projeção, agrupamento, ordenação ou junções.



\$match: Filtra os documentos na coleção com base em critérios específicos, semelhante à operação `find()`. Reduzindo o número de documentos que serão processados nos estágios subsequentes.

\$project: Projeta os campos dos documentos de entrada, permitindo que você inclua, exclua ou renomeie campos, bem como crie novos campos calculados usando expressões de agregação.

\$lookup: Executa uma operação de junção (`join`) entre documentos de várias coleções, combinando documentos com base em campos correspondentes e incluindo os documentos correspondentes em um array aninhado.

\$group: Agrupa os documentos com base em uma chave específica e executa operações de agregação nos grupos resultantes. Permitindo o cálculo estatístico como soma, média, mínimo, máximo e contabilização por grupo.

\$unwind: Desconstrói um campo de array em vários documentos, criando um documento separado para cada elemento do array. Isso é útil quando você precisa operar em elementos individuais de um array.

\$sort: Realiza a ordenação do resultado.

\$limit: Limita o número de documentos que serão processados ou retornados na consulta. Isso é útil para restringir o tamanho dos resultados ou para otimizar consultas.

Para acompanhar os exemplos e testar os comandos você deve considerar as seguintes collections:

```
db.funcionarios.insertMany([
  { _id: 1, nome: "João",    nascimento: ISODate("2000-03-15"), cargo_id: 10, departamento_id: 91 },
  { _id: 2, nome: "Maria",  nascimento: ISODate("2002-07-25"), cargo_id: 20, departamento_id: 91 },
  { _id: 3, nome: "Pedro",  nascimento: ISODate("2005-11-10"), cargo_id: 30, departamento_id: 92 }
])
```

```
db.cargos.insertMany([
  { _id: 10, nome: "Vendedor",          salario_base: 4000 },
  { _id: 20, nome: "Gerente de Vendas",  salario_base: 8000 },
  { _id: 30, nome: "Analista de Marketing", salario_base: 6000 }
])
```

```
db.departamentos.insertMany([
  { _id: 91, nome: "Vendas",    telefones: ["95984-7894", "3651-4512"], andar: 2, sala: "Sala 201" },
  { _id: 92, nome: "Marketing", telefones: ["95986-3649", "2461-3894"], andar: 3, sala: "Sala 301" }
])
```

\$match

O estágio **\$match** filtra os documentos na coleção com base em critérios específicos, semelhante à operação **find()**. Reduzindo o número de documentos que serão processados nos estágios subsequentes.

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   }
5 ])
```

```
[
  {
    _id: 1,
    nome: 'João',
    nascimento: ISODate('2000-03-15T00:00:00.000Z'),
    cargo_id: 10,
    departamento_id: 91
  },
  {
    _id: 2,
    nome: 'Maria',
    nascimento: ISODate('2002-07-25T00:00:00.000Z'),
    cargo_id: 20,
    departamento_id: 91
  }
]
```



\$lookup

O estágio \$lookup executa uma operação de junção (*join*) entre documentos de várias coleções, combinando documentos com base em campos correspondentes e incluindo os documentos correspondentes em um array aninhado.

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   },
5   {
6     $lookup: {
7       from: "departamentos",
8       localField: "departamento_id",
9       foreignField: "_id",
10      as: "departamento"
11    }
12  }
13 ])
```



```
[
  {
    _id: 1,
    nome: 'João',
    nascimento: ISODate('2000-03-15T00:00:00.000Z'),
    cargo_id: 10,
    departamento_id: 91,
    departamento: [
      {
        _id: 91,
        nome: 'Vendas',
        telefones: [ '95984-7894', '3651-4512' ],
        andar: 2,
        sala: 'Sala 201'
      }
    ]
  },
  {
    _id: 2,
    nome: 'Maria',
    nascimento: ISODate('2002-07-25T00:00:00.000Z'),
    cargo_id: 20,
    departamento_id: 91,
    departamento: [
      {
        _id: 91,
        nome: 'Vendas',
        telefones: [ '95984-7894', '3651-4512' ],
        andar: 2,
        sala: 'Sala 201'
      }
    ]
  }
]
```



\$lookup

O estágio \$lookup também permite que vários joins sejam executados em uma única consulta.

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   },
5   {
6     $lookup: {
7       from: "departamentos",
8       localField: "departamento_id",
9       foreignField: "_id",
10      as: "departamento"
11    }
12  },
13  {
14    $lookup: {
15      from: "cargos",
16      localField: "cargo_id",
17      foreignField: "_id",
18      as: "cargo"
19    }
20  }
21 ])
```



```
[
  {
    _id: 1,
    nome: 'João',
    nascimento: ISODate('2000-03-15T00:00:00.000Z'),
    cargo_id: 10,
    departamento_id: 91,
    departamento: [
      {
        _id: 91,
        nome: 'Vendas',
        telefones: [ '95984-7894', '3651-4512' ],
        andar: 2,
        sala: 'Sala 201'
      }
    ],
    cargo: [ { _id: 10, nome: 'Vendedor', salario_base: 4000 } ]
  },
  {
    _id: 2,
    nome: 'Maria',
    nascimento: ISODate('2002-07-25T00:00:00.000Z'),
    cargo_id: 20,
    departamento_id: 91,
    departamento: [
      {
        _id: 91,
        nome: 'Vendas',
        telefones: [ '95984-7894', '3651-4512' ],
        andar: 2,
        sala: 'Sala 201'
      }
    ],
    cargo: [ { _id: 20, nome: 'Gerente de Vendas', salario_base: 8000 } ]
  }
]
```

\$project

O estágio **\$project** exibe os campos dos documentos, permitindo que você inclua, exclua ou renomeie campos, assim como criar novos campos calculados usando expressões de agregação (Soma, Média, etc).

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   },
5   {
6     $lookup: {
7       from: "departamentos",
8       localField: "departamento_id",
9       foreignField: "_id",
10      as: "departamento"
11    }
12  },
13  {
14    $project: { nome:1, departamento: 1}
15  }
16 ])
```




```
[
  {
    _id: 1,
    nome: 'João',
    departamento: [
      {
        _id: 91,
        nome: 'Vendas',
        telefones: [ '95984-7894', '3651-4512' ],
        andar: 2,
        sala: 'Sala 201'
      }
    ]
  },
  { ... }
]
```


\$project

O estágio **\$Project** permite que você renomeie atributos no documento de saída. O uso do **\$** no nome original do atributo dentro do operador indica que o MongoDB deve utilizar o valor do atributo.

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   },
5   {
6     $lookup: {
7       from: "departamentos",
8       localField: "departamento_id",
9       foreignField: "_id",
10      as: "departamento"
11    }
12  },
13  {
14    $project: {
15      "nome_funcionario": "$nome",
16      "departamento.nome": 1
17    }
18  }
19 ])
```



```
[
  {
    _id: 1,
    departamento: [ { nome: 'Vendas' } ],
    nome_funcionario: 'João'
  },
  {
    _id: 2,
    departamento: [ { nome: 'Vendas' } ],
    nome_funcionario: 'Maria'
  }
]
```

O estágio **\$unwind** desconstrói um campo de array em vários documentos, criando um documento separado para cada elemento do array. Isso é útil quando você precisa operar em elementos individuais de um array.

```
1 db.funcionarios.aggregate([
2   {
3     $match: { departamento_id: 91 }
4   },
5   {
6     $lookup: {
7       from: "departamentos",
8       localField: "departamento_id",
9       foreignField: "_id",
10      as: "departamento"
11    }
12  },
13  {
14    $unwind: "$departamento"
15  },
16  {
17    $project: {
18      "nome_funcionario": "$nome",
19      "nome_departamento": "$departamento.nome"
20    }
21  }
22 ])
```

Resultado utilizando o \$unwind

```
[
  { _id: 1,
    nome_funcionario: 'João',
    nome_departamento: 'Vendas'
  },
  { _id: 2,
    nome_funcionario: 'Maria',
    nome_departamento: 'Vendas'
  }
]
```

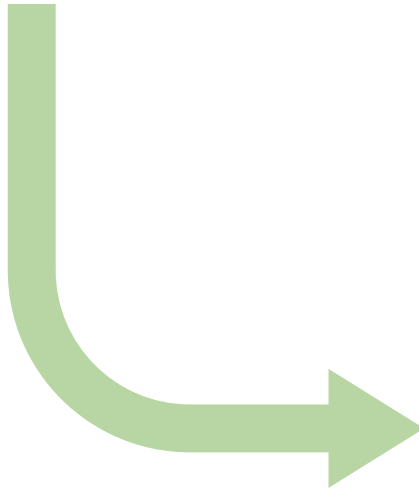
Resultado sem o \$unwind

```
[
  { _id: 1,
    nome_funcionario: 'João',
    nome_departamento: [ 'Vendas' ]
  },
  { _id: 2,
    nome_funcionario: 'Maria',
    nome_departamento: [ 'Vendas' ]
  }
]
```

\$sort

O estágio **\$sort** permite realizar a ordenação do resultado.

```
1 db.funcionarios.aggregate([
2   {$project: {nome:true, nascimento: true}},
3   {$sort: {nascimento: -1}}
4 ])
```

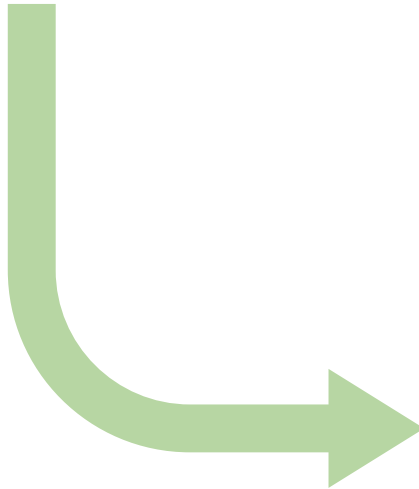


```
1 /* 1 */
2 {
3   "_id" : 3,
4   "nome" : "Pedro",
5   "nascimento" : ISODate("2005-11-09T22:00:00.000-02:00")
6 },
7
8 /* 2 */
9 {
10   "_id" : 2,
11   "nome" : "Maria",
12   "nascimento" : ISODate("2002-07-24T21:00:00.000-03:00")
13 },
14
15 /* 3 */
16 {
17   "_id" : 1,
18   "nome" : "João",
19   "nascimento" : ISODate("2000-03-14T21:00:00.000-03:00")
20 },
21
```

\$limit

O estágio \$limit limita o número de documentos que serão processados ou retornados na consulta. Isso é útil para restringir o tamanho dos resultados ou para otimizar consultas.

```
1 db.funcionarios.aggregate([
2   {$project: {nome:true, nascimento: true}},
3   {$sort: {nascimento: -1}}
4 ])
```



```
1 /* 1 */
2 {
3   "_id" : 3,
4   "nome" : "Pedro",
5   "nascimento" : ISODate("2005-11-09T22:00:00.000-02:00")
6 },
7
8 /* 2 */
9 {
10   "_id" : 2,
11   "nome" : "Maria",
12   "nascimento" : ISODate("2002-07-24T21:00:00.000-03:00")
13 },
14
15 /* 3 */
16 {
17   "_id" : 1,
18   "nome" : "João",
19   "nascimento" : ISODate("2000-03-14T21:00:00.000-03:00")
20 },
21
```



Dúvidas?

