

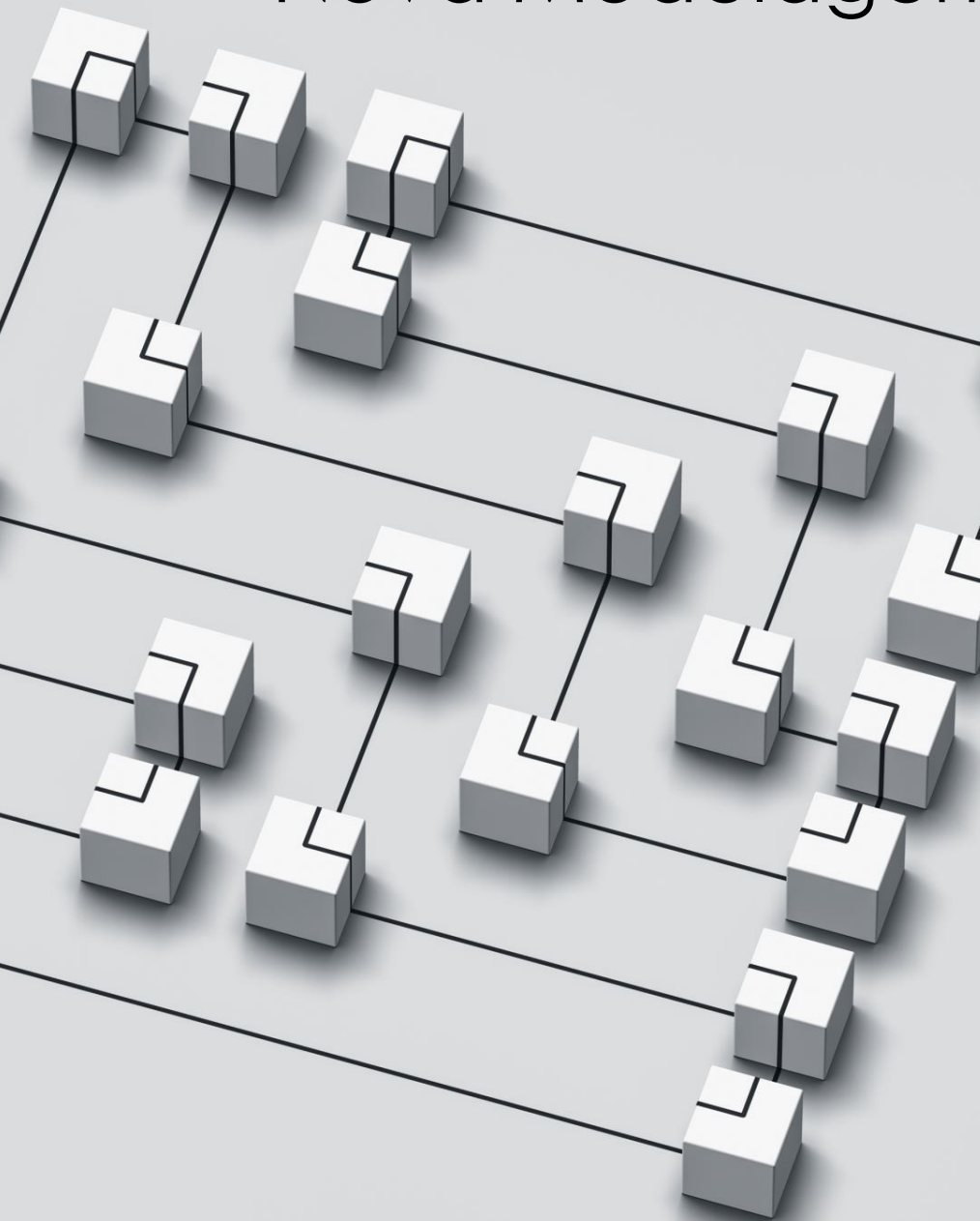


# MongoDB

Trabalhando com Arrays

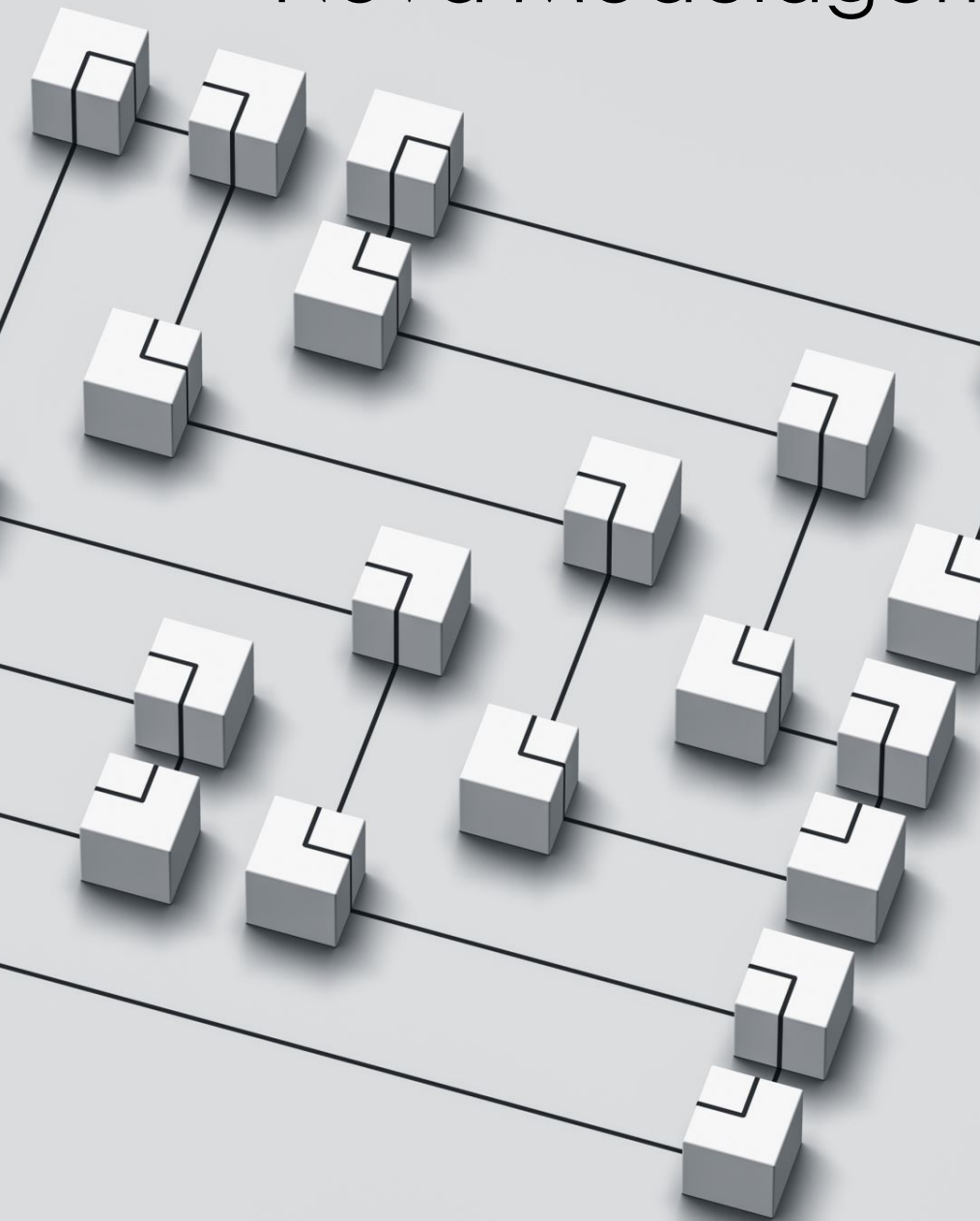


# Nova Modelagem



```
db.nome_da_colecao.aggregate([
  { $match: { "cod_venda": "999999" } },
  {
    $group: {
      _id: "$cod_venda",
      DATA_VENDA: { $first: "$DATA_VENDA" },
      COD_CLIENTE: { $first: "$COD_CLIENTE" },
      NOME_CLIENTE: { $first: "$NOME_CLIENTE" },
      produtos: {
        $push: {
          cod_produto: "$cod_produto",
          DESCRICAO: "$DESCRICAO",
          ....
          ..
          COD_VENDA: "$COD_VENDA"
        }
      }
    }
  }
])
```

# Nova Modelagem



```
db.nome_da_colecao.aggregate([
  { $match: { "cod_venda": { $ne: "20478" } } },
  {
    $group: {
      _id: "$cod_venda",
      DATA_VENDA: { $first: "$DATA_VENDA" },
      COD_CLIENTE: { $first: "$COD_CLIENTE" },
      NOME_CLIENTE: { $first: "$NOME_CLIENTE" },
      produtos: {
        $push: {
          cod_produto: "$cod_produto",
          DESCRICAO: "$DESCRICAO",
          Nome: "$Nome",
          cod_cliente: "$cod_cliente",
          COD_PRODUTO: "$COD_PRODUTO",
          COD_VENDA: "$COD_VENDA"
        }
      }
    }
  },
  {
    $out: "nova_colecao"
  }
])
```



# “Insert”

```
db.vendas.update(  
  { "_id": "17567" },  
  {  
    $push: {  
      "produtos": {  
        "cod_produto": "2000",  
        "DESCRICAO": "Mouse Gamer HyperX Pulsefire FPS Pro RGB - Preto",  
        "Nome": "Mouse Gamer HyperX Pulsefire FPS Pro RGB - Preto",  
        "Categoria": "Acessórios de Computador",  
        "MARCAS": "HYPERX",  
        "valor": 60,  
        "qtd": "10",  
        "VALOR": 0,  
        "QTD": 10,  
        "cod_cliente": 6918,  
        "COD_PRODUTO": 2000,  
        "COD_VENDA": 17567  
      }  
    }  
  }  
)
```



“Select”

```
db.vendas.findOne(  
  { "_id": "17567",  
    "produtos.cod_produto": "1299" },  
  { "produtos.$": 1 }  
)
```



**“update”**

```
db.vendas.update(  
  { "_id": "17567",  
    "produtos.cod_produto": "1299" },  
  {  
    $set: {  
      "produtos.$.valor": 90,  
      "produtos.$.qtd": "4"  
    }  
  }  
)
```



“update”

```
db.vendas.updateOne(  
  { "_id": "17567",  
    "produtos.cod_produto": "1299" },  
  { $set: { "produtos.$.valor": 100 } }  
)
```

“delete”

```
db.vendas.update(  
  { "_id": "17567" },  
  {  
    $pull: {  
      "produtos": { "cod_produto": "1299" }  
    }  
  }  
)
```





# \$unwind

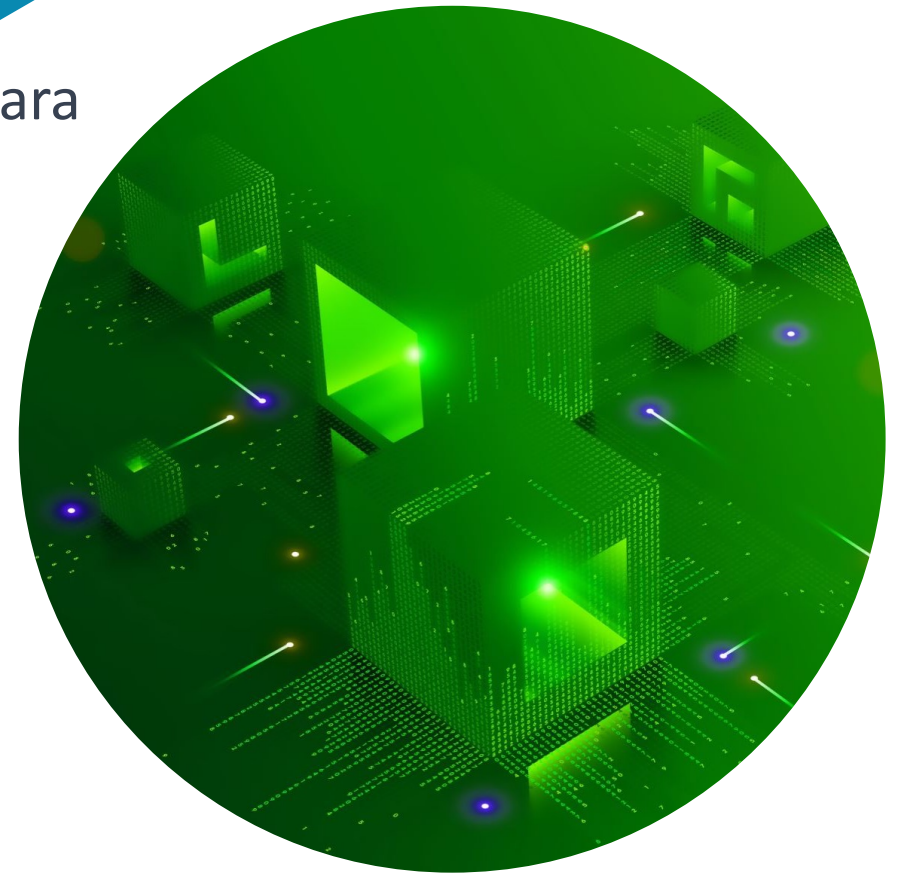
1. Transforma um campo de matriz em vários documentos, um para cada elemento da matriz. É útil para "desnormalizar" documentos com campos de matriz e realizar operações em elementos individuais da matriz.

```
db.vendas.aggregate([  
  {  
    $unwind: "$itens"  
  }  
])
```

# Exercícios

- 1 - Escreva uma consulta de agregação no MongoDB para calcular o total de vendas na coleção "vendas", considerando o campo "valor" em cada documento.

```
db.vendas_array.aggregate([
  {
    $unwind: "$produtos"
  },
  {
    $group: {
      _id: "$_id",
      totalVendas: { $sum: { $multiply: [ "$produtos.valor", "$produtos.QTD" ] } }
    }
  }
])
```



1. `$unwind`: Esta parte da consulta é usada para transformar arrays em documentos individuais. Cada documento resultante substitui o array pelo elemento. No caso de sua coleção de dados, o campo "produtos" é um array que contém vários produtos, cada um sendo um documento separado. O operador `$unwind` "desenrola" esse array em vários documentos, cada um contendo um único produto.
2. `$group`: Este estágio da consulta é usado para agrupar documentos por algum valor de campo especificado. No caso da consulta que forneci, estamos agrupando por `_id`, que é o ID de cada venda/documento em sua coleção. O resultado de cada grupo é então processado de acordo com as expressões de acumulação fornecidas, neste caso `$sum`.



1. `_id: "$_id"`: Isso indica que queremos agrupar os documentos por seu campo `_id`. Como o campo `_id` é único para cada venda/documento, isso efetivamente calcula o total de vendas para cada venda/documento individualmente.
2. `totalVendas: { $sum: { $multiply: [ "$produtos.valor", "$produtos.QTD" ] } }`: Esta é uma expressão de acumulação que calcula o total de vendas para cada grupo. `$sum` adiciona os valores para cada documento no grupo. A expressão `{ $multiply: [ "$produtos.valor", "$produtos.QTD" ] }` multiplica o valor de cada produto pela quantidade correspondente, e o resultado é então somado pelo `$sum` para obter o total de vendas.

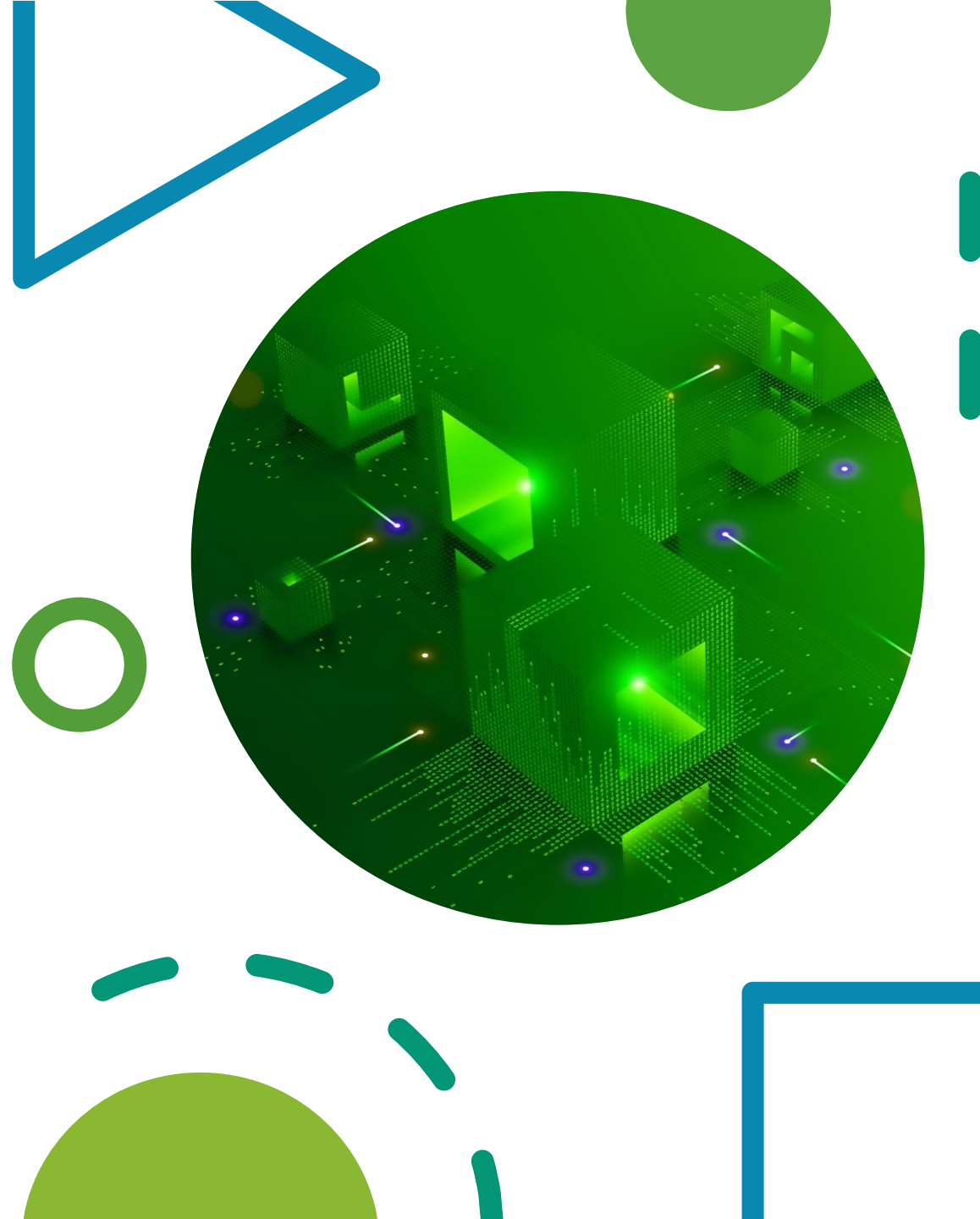




# Exercícios

- 2 - Escreva uma consulta de agregação no MongoDB para contar o número de vendas na coleção "vendas" agrupadas por categoria.

```
db.vendas.aggregate([
  {
    $unwind: "$produtos"
  },
  {
    $group: {
      _id: "$produtos.Categoria",
      numeroDeVendas: { $sum: 1 }
    }
  }
])
```

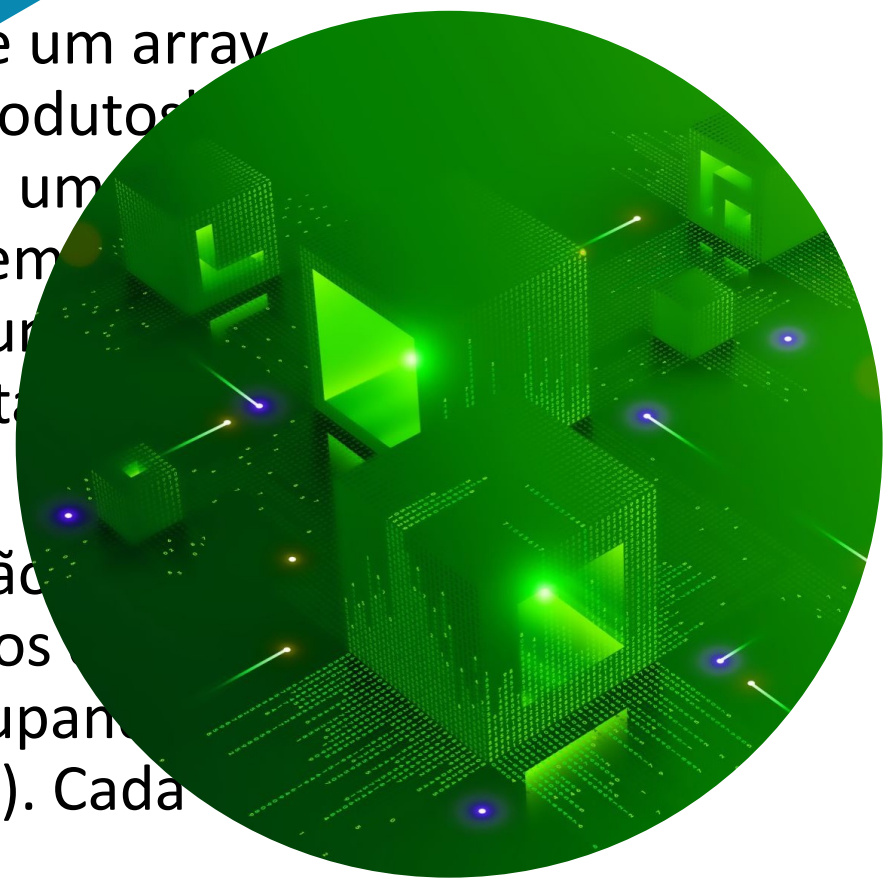




# Exercícios

**\$unwind:** Este é o primeiro estágio do pipeline de agregação. O operador **\$unwind** é usado para separar os elementos de um array em documentos individuais. No nosso caso, o array é "produtos". Quando usamos **\$unwind: "\$produtos"**, estamos criando um documento para cada produto em cada venda. Se tivéssemos uma venda com três produtos, por exemplo, após o estágio **\$unwind** teríamos três documentos separados, cada um representando um produto.

**\$group:** Este é o segundo estágio do pipeline de agregação. O operador **\$group** é usado para agrupar documentos juntos baseados em algum campo comum. Neste caso, estamos agrupando documentos por "Categoria" (`_id: "$produtos.Categoria"`). Cada grupo único de documentos terá um novo campo, "numeroDeVendas", que é calculado adicionando 1 para cada documento no grupo (`numeroDeVendas: { $sum: 1 }`). O `$sum: 1` é uma maneira de contar o número de documentos no grupo, ou seja, o número de vendas em cada categoria.



# Exercícios

- 3 - Escreva uma consulta de agregação no MongoDB para calcular o valor médio de vendas na coleção "vendas" agrupadas por marca.



# Exercícios

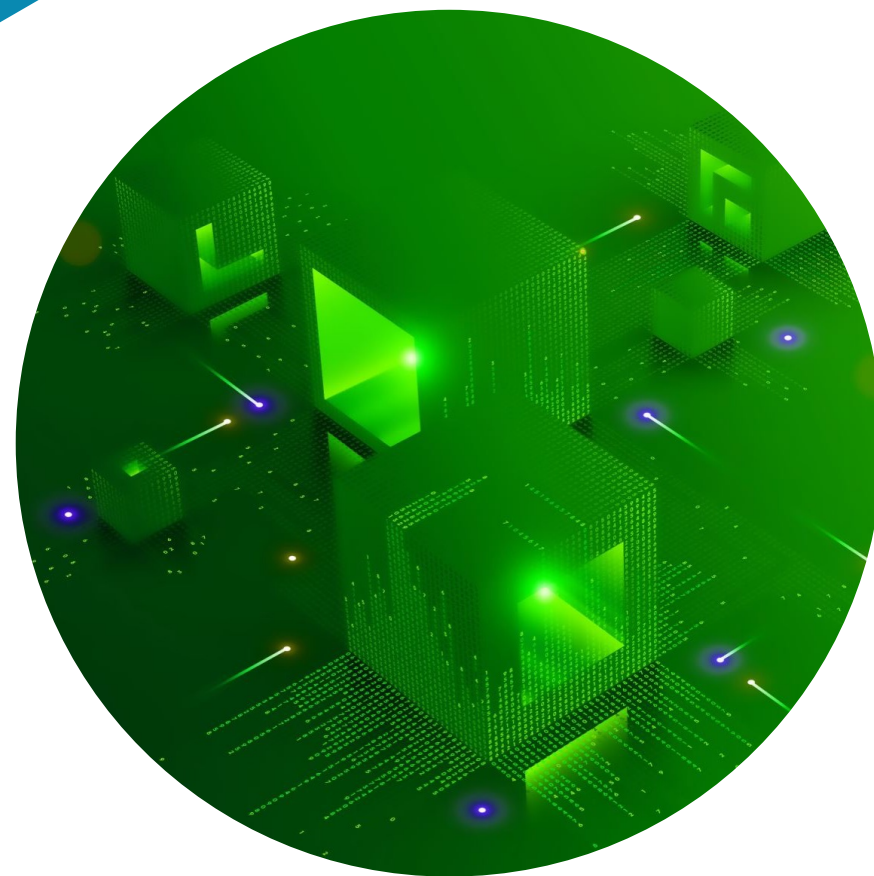
- 4 - Escreva uma consulta de agregação no MongoDB para calcular o total de produtos vendidos na coleção "vendas" agrupadas por código do cliente



- 5 Escreva uma consulta de agregação no MongoDB para identificar o produto mais vendido na coleção "vendas", com base no campo "QTD" em cada documento.



- 6 - Escreva uma consulta de agregação no MongoDB para calcular o total de vendas na coleção "vendas" agrupadas por data de venda.





# Exercícios

- 7 - Escreva uma consulta de agregação no MongoDB para calcular o total de produtos vendidos na coleção "vendas" agrupadas por código do cliente e ordene os resultados em ordem decrescente do total de produtos vendidos.



# Exercícios

- 8 - Escreva uma consulta de agregação no MongoDB para calcular o valor médio de vendas na coleção "vendas" agrupadas por código do cliente.



# Exercícios

- 9 - Escreva uma consulta de agregação no MongoDB para contar o número de vendas na coleção "vendas" agrupadas por descrição do produto.



# Exercícios

- 10 - Escreva uma consulta de agregação no MongoDB para obter o total de produtos vendidos agrupados por descrição e categoria na coleção "vendas".



11. Média do valor das vendas por categoria.
12. Valor mínimo e máximo das vendas por categoria.
13. Soma do valor das vendas por marca e categoria.
14. Quantidade total de produtos vendidos por marca e categoria.
15. Ordenar as vendas por data de venda em ordem crescente.
16. Selecione as vendas com quantidade maior ou igual a 5.
17. Valor total das vendas por ano.
18. Média do valor das vendas por ano.
19. Selecione as vendas com valor entre 10 e 30.
20. Selecione as 5 vendas com maior valor.
21. Selecione as 5 vendas com menor valor.





- 22. Quantidade total de produtos vendidos por mês.
- 23. Valor total das vendas por mês.
- 24. Valor total das vendas por cliente em ordem decrescente.
- 25. Média do valor das vendas por mês.
- 26. Selecione as vendas com valor maior ou igual a 50.
- 27. Valor total das vendas por dia da semana.
- 28. Média do valor das vendas por dia da semana.
- 29. Quantidade total de produtos vendidos por dia da semana.
- 30. Média do valor das vendas por marca.



- 31. Selecione as vendas que ocorreram em fevereiro.
  - 32. Ordene as vendas por valor e pelo código do cliente em ordem crescente.
  - 33. Soma do valor das vendas para cada combinação de categoria e marca.
  - 34. Selecione as vendas que ocorreram no primeiro semestre do ano.
  - 35. Média do valor das vendas no primeiro semestre do ano.
  - 36. Quantidade total de produtos vendidos no primeiro semestre do ano.
  - 37. Ordene as vendas por categoria em ordem alfabética e pelo valor em ordem decrescente.
- 
- 31. Selecione as vendas com valor igual a 10, 20 ou 30.
  - 32. Selecione as vendas que ocorreram no mês de dezembro.

