

Questão 1 – Consulta com Operadores

Lógicos Liste os produtos da categoria "Periféricos" cujo preço seja superior a R\$ 300 e estoque inferior a 10 unidades. Resultado esperado: Retorna apenas o Teclado Mecânico K500.

```
db.produtos.find({
  $and: [
    { categoria: "Periféricos" },
    { preco: { $gt: 300 } },
    { estoque: { $lt: 10 } }
  ]
})
```

```
> db.produtos.find({
  $and: [
    { categoria: "Periféricos" },
    { preco: { $gt: 300 } },
    { estoque: { $lt: 10 } }
  ]
})
< {
  _id: ObjectId('69092d5053de57865388c061'),
  nome: 'Teclado Mecânico K500',
  categoria: 'Periféricos',
  preco: 450,
  estoque: 8
}
```

Questão 2 – Inserção com Operadores de Atualização

Adicione o campo "desconto": 0 em todos os produtos e, depois, atualize o "Monitor UltraWide" para aplicar desconto de 10%. Resultado esperado: todos os produtos têm desconto = 0, exceto o Monitor UltraWide, que tem 10.

```
db.produtos.updateMany({}, { $set: { desconto: 0 } })
```

```
db.produtos.updateOne(
  { nome: "Monitor UltraWide" },
  { $set: { desconto: 10 } })
```

)

```
> db.produtos.updateMany({}, { $set: { desconto: 0 } })

< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}

> db.produtos.updateOne(
  { nome: "Monitor UltraWide" },
  { $set: { desconto: 10 } }
)

< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Resultado:

```
< {  
  _id: ObjectId('69092d5053de57865388c060'),  
  nome: 'Mouse Gamer X7',  
  categoria: 'Periféricos',  
  preco: 250,  
  estoque: 15,  
  desconto: 0  
}  
{  
  _id: ObjectId('69092d5053de57865388c061'),  
  nome: 'Teclado Mecânico K500',  
  categoria: 'Periféricos',  
  preco: 450,  
  estoque: 8,  
  desconto: 0  
}  
{  
  _id: ObjectId('69092d5053de57865388c062'),  
  nome: 'Monitor UltraWide',  
  categoria: 'Monitores',  
  preco: 1800,  
  estoque: 5,  
  desconto: 10  
}
```

```
{
  _id: ObjectId('69092d5053de57865388c063'),
  nome: 'Console Zeta',
  categoria: 'Consoles',
  preco: 3200,
  estoque: 3,
  desconto: 0
}
{
  _id: ObjectId('69092d5053de57865388c064'),
  nome: 'Headset ProSound',
  categoria: 'Áudio',
  preco: 600,
  estoque: 12,
  desconto: 0
}
```

Questão 3 – Incremento de Valores

Aumente o estoque de todos os produtos da categoria “Áudio” em 3 unidades.
Resultado esperado: Antes: estoque = 12 → Depois: 15.

```
db.produtos.updateMany(
  { categoria: "Áudio" },
  { $inc: { estoque: 3 } }
)
```

```

{
  _id: ObjectId('69092d5053de57865388c064'),
  nome: 'Headset ProSound',
  categoria: 'Áudio',
  preco: 600,
  estoque: 12,
  desconto: 0
}
> db.produtos.updateMany(
  { categoria: "Áudio" },
  { $inc: { estoque: 3 } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.produtos.find({categoria: "Áudio"})
< {
  _id: ObjectId('69092d5053de57865388c064'),
  nome: 'Headset ProSound',
  categoria: 'Áudio',
  preco: 600,
  estoque: 15,
  desconto: 0
}

```

Questão 4 – Atualização e Substituição

Substitua completamente o documento do produto “Console Zeta” por uma nova versão, mantendo o mesmo nome, mas com preço atualizado para 3500 e campo novo “garantia”: “2 anos”.

```

db.produtos.replaceOne(
  { nome: "Console Zeta" },
  {
    nome: "Console Zeta",
    preco: 3500,
    categoria: "Games",

```

```
    garantia: "2 anos",  
    estoque: 4  
  }  
)
```

Antes:

```
{  
  _id: ObjectId('69092d5053de57865388c063'),  
  nome: 'Console Zeta',  
  categoria: 'Consoles',  
  preco: 3200,  
  estoque: 3,  
  desconto: 0  
}
```

Depois:

```

> db.produtos.replaceOne(
  { nome: "Console Zeta" },
  {
    nome: "Console Zeta",
    preco: 3500,
    categoria: "Games",
    garantia: "2 anos",
    estoque: 4
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.produtos.find({categoria: "Games"})
< {
  _id: ObjectId('69092d5053de57865388c063'),
  nome: 'Console Zeta',
  preco: 3500,
  categoria: 'Games',
  garantia: '2 anos',
  estoque: 4
}

```

Questão 5 – Remoção Condicional

Remova todos os produtos cujo estoque seja menor que 5. Resultado esperado: Console Zeta será removido.

```
db.produtos.deleteMany({ estoque: { $lt: 5 } })
```

```

> db.produtos.deleteMany({ estoque: { $lt: 5 } })
< {
  acknowledged: true,
  deletedCount: 1
}
> db.produtos.find({ categoria: "Games" })
<

```

Questão 6 – Agregação e Contagem

Conte quantos produtos existem em cada categoria.

```

db.produtos.aggregate([
  { $group: { _id: "$categoria", totalProdutos: { $sum: 1 } } }
])

```

```

> db.produtos.aggregate([
  { $group: { _id: "$categoria", totalProdutos: { $sum: 1 } } }
])
< {
  _id: 'Áudio',
  totalProdutos: 1
}
{
  _id: 'Periféricos',
  totalProdutos: 2
}
{
  _id: 'Monitores',
  totalProdutos: 1
}

```

Questão 7 – Média de Preços por Categoria

Calcule o preço médio dos produtos de cada categoria e ordene do maior para o menor.

```

db.produtos.aggregate([
  { $group: { _id: "$categoria", mediaProdutoPreco: { $avg: "$preco" } } },
  { $sort: { mediaProdutoPreco: -1 } }
])

```



```

> db.produtos.aggregate([
  { $group: { _id: "$categoria", mediaProdutoPreco: { $avg: "$preco" } } },
  { $sort: { mediaProdutoPreco: -1 } }
])
< {
  _id: 'Monitores',
  mediaProdutoPreco: 1800
}
{
  _id: 'Áudio',
  mediaProdutoPreco: 600
}
{
  _id: 'Periféricos',
  mediaProdutoPreco: 350
}

```

Questão 8 – Paginação

Liste os produtos exibindo apenas 2 resultados por página, mostrando a segunda página. Resultado esperado: Página 2 = registros 3 e 4.

```
db.produtos.find().skip(2).limit(2)
```

```

> db.produtos.find().skip(2).limit(2)
< {
  _id: ObjectId('69092d5053de57865388c062'),
  nome: 'Monitor UltraWide',
  categoria: 'Monitores',
  preco: 1800,
  estoque: 5,
  desconto: 10
}
{
  _id: ObjectId('69092d5053de57865388c064'),
  nome: 'Headset ProSound',
  categoria: 'Áudio',
  preco: 600,
  estoque: 15,
  desconto: 0
}

```

Como foi deletado um dos itens, sendo ele o Console Zeta os 3 e 4 foram Monitor e Headset, conforme lista:

```
db.produtos.insertMany([
  { nome: "Mouse Gamer X7", categoria: "Periféricos", preco: 250, estoque: 15 },
  { nome: "Teclado Mecânico K500", categoria: "Periféricos", preco: 450, estoque: 8 },
  { nome: "Monitor UltraWide", categoria: "Monitores", preco: 1800, estoque: 5 },
  { nome: "Console Zeta", categoria: "Consoles", preco: 3200, estoque: 3 },
  { nome: "Headset ProSound", categoria: "Áudio", preco: 600, estoque: 12 }
]);
```

Questão 9 – Indexação e Otimização

Crie um índice no campo "categoria" e, em seguida, verifique se está sendo utilizado nas consultas.

```
db.produtos.createIndex({ categoria: 1 })
```

```
db.produtos.getIndexes()
```

```
db.produtos.find({ categoria: "Periféricos" }).explain("executionStats")
db.produtos.find({ categoria: "Monitores" }).explain("executionStats")
db.produtos.find({ categoria: "Consoles" }).explain("executionStats")
db.produtos.find({ categoria: "Áudio" }).explain("executionStats")
```

Criando index:

```
> db.produtos.createIndex({ categoria: 1 })

< categoria_1
> db.produtos.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { categoria: 1 }, name: 'categoria_1' }
]
```

Verificando

```
executionTimeMillisEstimate: 0,
works: 3,
advanced: 2,
needTime: 0,
needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
docsExamined: 2,
alreadyHasObj: 0,
inputStage: {
  stage: 'IXSCAN',
  nReturned: 2,
  executionTimeMillisEstimate: 0,
  works: 3,
  advanced: 2,
  needTime: 0,
  needYield: 0,
  saveState: 0,
  restoreState: 0,
  isEOF: 1,
  keyPattern: {
    categoria: 1
  },
  indexName: 'categoria_1',
  isMultiKey: false,
  multiKeyPaths: {
    categoria: []
  },
```

Questão 10 – Integração com API

Crie uma rota em Node.js/Express que exiba todos os produtos com apenas nome e preço, limitando a 5 resultados

