

Questão 1 – Consulta com Operadores Lógicos

```
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('69093a9b8ba87292d288775c'),  
    nome: 'Teclado Mecânico K500',  
    categoria: 'Periféricos',  
    preco: 450,  
    estoque: 8  
}
```

Questão 2 – Inserção com Operadores de Atualização

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

```
db.produtos.updateOne({ nome: "Monitor UltraWide" }, { $set: { desconto: 10 } });
db.produtos.updateMany({}, { $set: { desconto: 0 } });
db.produtos.updateOne({ nome: "Monitor UltraWide" }, { $set: { desconto: 10 } });

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

Questão 3 – Incremento de Valores

```
db.produtos.updateMany(
```

```
  { categoria: "Áudio" },
```

```
  { $inc: { estoque: 3 } }
```

```
);
```

```
db.produtos.updateMany(
  { categoria: "Áudio" },
  { $inc: { estoque: 3 } }
);
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Questão 4 – Atualização e Substituição

```
db.produtos.replaceOne(
```

```
{ nome: "Console Zeta" },  
  
{ nome: "Console Zeta", categoria: "Consoles", preco: 3500, garantia: "2 anos" }  
  
);  
  
db.produtos.replaceOne(  
  { nome: "Console Zeta" },  
  { nome: "Console Zeta", categoria: "Consoles", preco: 3500, garantia: "2 anos" }  
);  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

Questão 5 – Remoção Condisional

```
db.produtos.deleteMany({ estoque: { $lt: 5 } });  
db.produtos.deleteMany({ estoque: { $lt: 5 } })  
{  
  acknowledged: true,  
  deletedCount: 0  
}
```

Questão 6 – Agregação e Contagem

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

```
db.produtos.aggregate([
  { $group: { _id: "$categoria", total_produtos: { $sum: 1 } } }
]);
{
  _id: 'Áudio',
  total_produtos: 1
}
{
  _id: 'Periféricos',
  total_produtos: 2
}
{
  _id: 'Consoles',
  total_produtos: 1
}
{
  _id: 'Monitores',
  total_produtos: 1
}
```

Questão 7 – Média de Preços por Categoria

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

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

{
  _id: 'Consoles',
  media_preco: 3500
}
{
  _id: 'Monitores',
  media_preco: 1800
}
{
  _id: 'Áudio',
  media_preco: 600
}
{
  _id: 'Periféricos',
  media_preco: 350
}
```

Questão 8 – Paginação

```
db.produtos.find().skip(2).limit(2);  
  
db.produtos.find().skip(2).limit(2);  
{  
  _id: ObjectId('69093a9b8ba87292d288775d') ,  
  nome: 'Monitor UltraWide' ,  
  categoria: 'Monitores' ,  
  preco: 1800 ,  
  estoque: 5 ,  
  desconto: 10  
}  
  
{  
  _id: ObjectId('69093a9b8ba87292d288775e') ,  
  nome: 'Console Zeta' ,  
  categoria: 'Consoles' ,  
  preco: 3500 ,  
  garantia: '2 anos'  
}
```

Questão 9 – Indexação e Otimização

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

```
db.produtos.find({ categoria: "Periféricos" }).explain("executionStats");
db.produtos.createIndex({ categoria: 1 });

// Verificar índices criados
db.produtos.getIndexes();

// Forçar uso do índice (opcional)
db.produtos.find({ categoria: "Periféricos" }).explain("executionStats");
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'rede_games.produtos',
    parsedQuery: {
      categoria: {
        '$eq': 'Periféricos'
      }
    },
    indexFilterSet: false,
    queryHash: '8FC8741C',
    planCacheShapeHash: '8FC8741C',
    planCacheKey: '637486FA',
    optimizationTimeMillis: 2,
    maxIndexedOrSolutionsReached: false,
```

```
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
prunedSimilarIndexes: false,
winningPlan: {
  isCached: false,
  stage: 'FETCH',
  inputStage: {
    stage: 'IXSCAN',
    keyPattern: {
      categoria: 1
    },
    indexName: 'categoria_1',
    isMultiKey: false,
    multiKeyPaths: {
      categoria: []
    },
    isUnique: false,
    isSparse: false,
    isPartial: false,
    indexVersion: 2,
    direction: 'forward',
    indexBounds: {
```

```
        categoria: [
            '["Periféricos", "Periféricos"]'
        ]
    }
}

},
rejectedPlans: []

},
executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 12,
    totalKeysExamined: 2,
    totalDocsExamined: 2,
    executionStages: {
        isCached: false,
        stage: 'FETCH',
        nReturned: 2,
        executionTimeMillisEstimate: 8,
        works: 3,
        advanced: 2,
        needTime: 0,
```

```
needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
docsExamined: 2,
alreadyHasObj: 0,
inputStage: {
  stage: 'IXSCAN',
  nReturned: 2,
  executionTimeMillisEstimate: 8,
  works: 3,
  advanced: 2,
  needTime: 0,
  needYield: 0,
  saveState: 0,
  restoreState: 0,
  isEOF: 1,
  keyPattern: {
    categoria: 1
  },
  indexName: 'categoria_1',
  isMultiKey: false,
  multiKeyPaths: {
```

```
multiKeyPaths: {
    categoria: []
},
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: {
    categoria: [
        '["Periféricos", "Periféricos"]'
    ]
},
keysExamined: 2,
seeks: 1,
dupsTested: 0,
dupsDropped: 0
},
},
queryShapeHash: '298EF84B1C6F8CB11F469EEEDE5F9C9413646E69D2DF80A5D1E68D6E168A13D',
command: {
    find: 'produtos',
```

```
        categoria: 'Periféricos'
    },
    '$db': 'rede_games'
},
serverInfo: {
    host: 'DESKTOP-I3FQSDE',
    port: 27017,
    version: '8.0.5',
    gitVersion: 'cb9e2e5e552ee39de1e39d7859336456d0c9820'
},
serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted',
    internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
},
ok: 1
```

Questão 10 – Integração com API (Node.js + Express + MongoDB)

```
const express = require("express");
```

```
const mongoose = require("mongoose");
```

```
const app = express();
```

```
mongoose.connect("mongodb://localhost:27017/rede_games");
```

```
const produtoSchema = new mongoose.Schema({
```

```
    nome: String,
```

```
    preco: Number
```

```
});
```

```
const Produto = mongoose.model("Produto", produtoSchema);
```

```
app.get("/produtos", async (req, res) => {
  const produtos = await Produto.find({}, { nome: 1, preco: 1, _id: 0 }).limit(5);
  res.json(produtos);
});
```

```
app.listen(3000, () => console.log("Servidor rodando em http://localhost:3000"));
```

```
C:\Users\fatec258\Documents\GitHub\Banco-de-Dados-Nao-Relacional\rede_games - Atv 2>
C:\Users\fatec258\Documents\GitHub\Banco-de-Dados-Nao-Relacional\rede_games - Atv 2>node index.js
Servidor rodando em http://localhost:3000
```