

Prova I
Disciplina: Banco de Dados Não Relacional

Professor: Diego Ramos Inácio
Curso de Engenharia de Software

Aluno: Leandro Lima Cardoso
Matricula: 202323366

PARTE 1 QUESTÕES TEÓRICAS:

Questão 1:

De acordo com a linha do tempo apresentada nas aulas, qual foi o principal marco que iniciou a era NoSQL e suas motivações técnicas?

- ☐ a) O surgimento do modelo relacional de Edgar Codd na IBM em 1970, que estabeleceu SQL e normalização
- ☒ b) A publicação dos papers do Google (GFS, MapReduce, Bigtable) e do Amazon Dynamo entre 2003 -2 007, que apresentaram soluções para problemas de escala web
- ☐ c) O desenvolvimento de Data Warehouses e sistemas OLAP nos anos 90 para Business Intelligence
- ☐ d) A criação dos bancos hierárquicos e de rede (IMS, CODASYL) nos anos 60 -8 0 com navegação por ponteiros
- ☐ e) O amadurecimento do ecossistema Hadoop com MongoDB, Neo4j e Cassandra em 2010

Questão 2:

Sobre o Teorema CAP e as filosofias ACID vs BASE, analise as afirmativas:

- I. Teorema CAP estabelece que em sistemas distribuídos é possível garantir simultaneamente Consistência,
- II. ACID (Atomicity, Consistency, Isolation, Durability) é característico de bancos relacionais e prioriza consistência forte e transações complexas
- III. BASE (Basically Available, Soft state, Eventual consistency) ganhou espaço quando throughput e disponibilidade tornaram-se mais críticos que consistência imediata
- IV. Na realidade prática, os trade-offs do CAP são binários e permanentes, aplicando-se igualmente a todas as operações

Estão corretas apenas:

- ☐ a) I e II
- ☒ b) II e III
- ☐ c) I, II e IV
- ☐ d) II, III e IV
- ☐ e) I, III e IV

Questão 3:

Sobre a evolução histórica dos bancos de dados apresentada nas aulas, analise as afirmativas:

- I. Na era pré-relacional (1960-1970), os modelos hierárquico (IBM IMS) e de rede (CODASYL) apresentavam limitações como navegação complexa e dependência de aplicação
- II. A Revolução Relacional iniciada por Edgar F. Codd em 1970 trouxe o modelo de tabelas com propriedades ACID, consolidando-se entre 1990-2000 com 95% do mercado
- III. O termo "NoSQL" significa "No SQL" (Não SQL) e foi criado para substituir completamente os bancos relacionais em todas as aplicações
- IV. O crescimento da Internet (2000-2005) trouxe novos desafios como explosão de dados, necessidade de escalabilidade para milhões de usuários simultâneos e variedade de tipos de dados

Estão corretas:

- ☐ a) I, II e III apenas
- ☒ b) I, II e IV apenas
- ☐ c) II, III e IV apenas
- ☐ d) I, III e IV apenas
- ☐ e) Todas as afirmativas

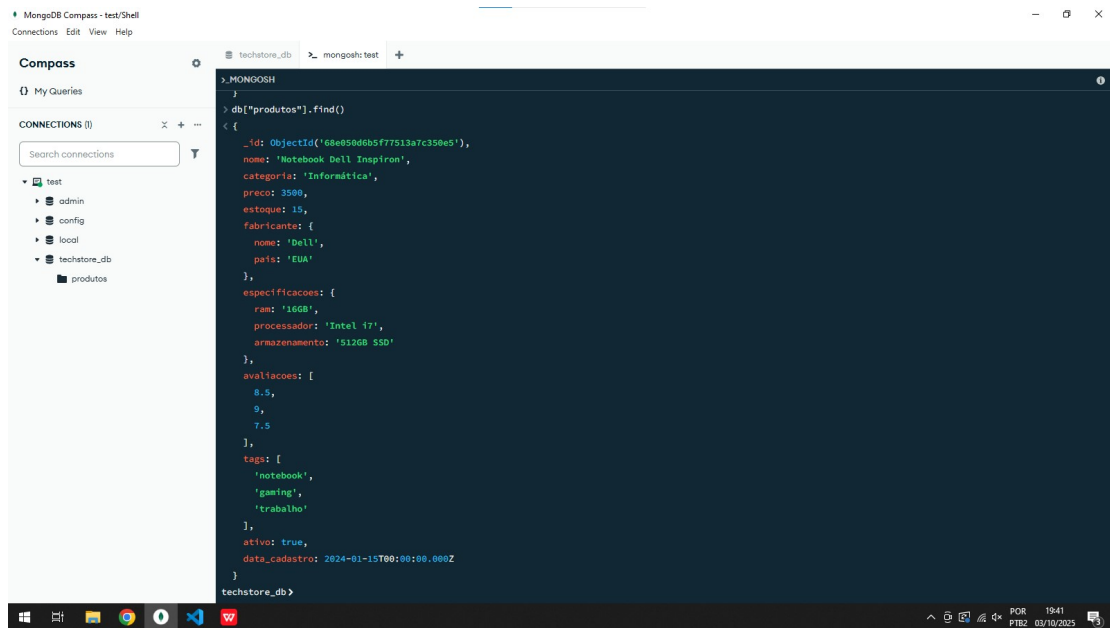
Questão 4:

Sobre os tipos de bancos NoSQL e seus casos de uso, assinale a alternativa que faz a correspondência CORRETA:

- ☐ a) Chave-valor (Redis) - ideal para redes sociais e detecção de fraudes com relacionamentos complexos
- ☐ b) Documento (MongoDB) - otimizado para data warehousing e consultas analíticas com compressão eficiente
- ☒ c) Coluna (Cassandra, HBase) - estrutura de dados organizados por famílias de colunas, vantajoso para analytics e séries temporais
- ☐ d) Grafo (Neo4j) - armazena pares chave-valor simples, ideal para cache e contadores de alta performance
- ☐ e) Documento (CouchDB) - utiliza nós e arestas, adequado para consultas de roteamento em mapas

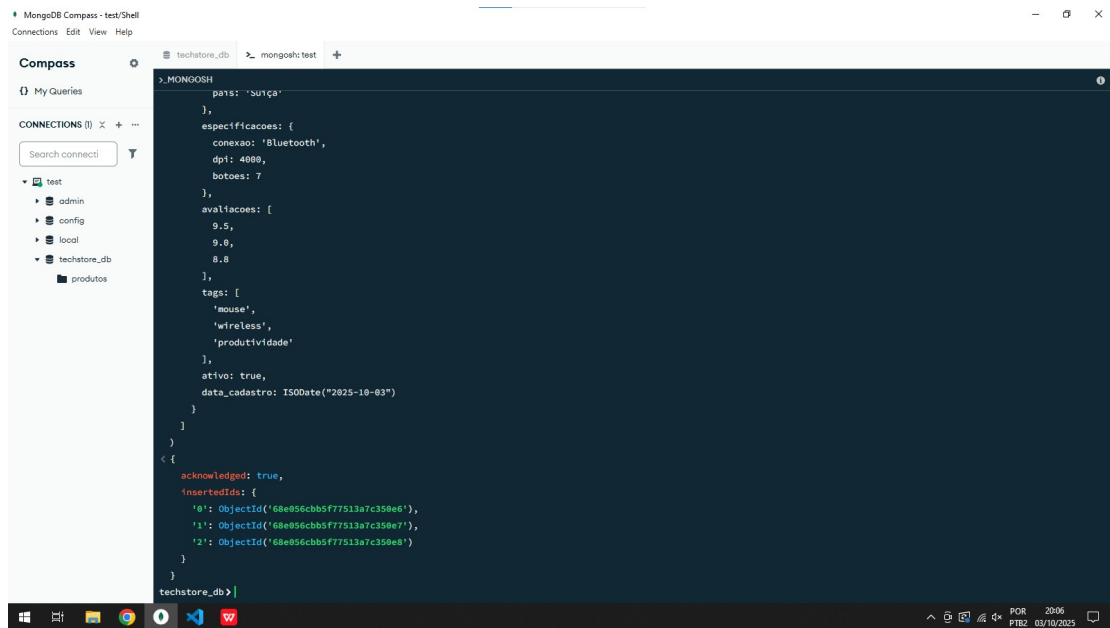
PARTE 2 QUESTÕES PRÁTICAS:

Contexto:



Questão 5 - CREATE:

Resposta: A)



The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists 'test', 'admin', 'config', 'local', 'techstore_db', and 'produtos'. The 'techstore_db' database is selected, and the 'produtos' collection is visible. The main window displays a document from the 'produtos' collection. The document is a JSON object with the following fields: '_id' (ObjectId), 'ativo' (false), 'data_cadastro' (2025-10-03T00:00:00.000Z), 'categoria' ('Periféricos'), 'preco' (650), 'estoque' (50), 'fabricante' (object with 'nome' 'Logitech' and 'pais' 'Suíça'), 'especificacoes' (object with 'conexao' 'Bluetooth', 'dpi' 4000, and 'botões' 7), 'avaliacoes' (array of 3 ratings: 9.5, 9, 8.8), 'tags' (array of 'mouse', 'wireless', 'produtividade'), and 'ativo' (true).

```
>_MONGOSH
{
  ativo: false,
  data_cadastro: 2025-10-03T00:00:00.000Z
}
{
  _id: ObjectId('68e956cb5f77513a7c350e8'),
  categoria: 'Periféricos',
  preco: 650,
  estoque: 50,
  fabricante: {
    nome: 'Logitech',
    pais: 'Suíça'
  },
  especificacoes: {
    conexao: 'Bluetooth',
    dpi: 4000,
    botoes: 7
  },
  avaliacoes: [
    9.5,
    9,
    8.8
  ],
  tags: [
    'mouse',
    'wireless',
    'produtividade'
  ],
  ativo: true,
  data_cadastro: 2025-10-03T00:00:00.000Z
}
techstore_db>
```

Questão 5 - READ:

Resposta: A)

The screenshot shows the MongoDB Compass interface. The 'CONNECTIONS' panel is the same as in the previous image. The main window displays a query result for the 'produtos' collection. The query is a find operation on the 'produtos' collection, filtering for documents where the 'categoria' is in the array ['Smartphones', 'Monitores'], the 'preco' is between 1000 and 5000, and the 'estoque' is greater than or equal to 10. The result is a single document with the following fields: 'nome' (1), 'preco' (1), 'estoque' (1), 'fabricante.nome' (1), '_id' (0), 'preco' (4200), 'estoque' (25), and 'fabricante' (object with 'nome' 'Samsung').

```
>_MONGOSH
{
  tags: [
    'mouse',
    'wireless',
    'produtividade'
  ],
  ativo: true,
  data_cadastro: 2025-10-03T00:00:00.000Z
}
> db["produtos"].find(
  {
    categoria: {$in: ["Smartphones", "Monitores"]}, preco: {$gte: 1000, $lte: 5000},
    estoque: {$gte: 10},
    ativo: true
  },
  {
    nome: 1,
    preco: 1,
    estoque: 1,
    "fabricante.nome": 1,
    _id: 0
  }
)
< {
  preco: 4200,
  estoque: 25,
  fabricante: {
    nome: 'Samsung'
  }
}
techstore_db>
```

Questão 5 - UPDATE:

Resposta: D)

MongoDB Compass - test/Shell
Connections Edit View Help

Compass

My Queries

CONNECTIONS (1) x + ...

Search connect

- test
- admin
- config
- local
- techstore_db
 - produtos

```
> .MONGOSSH
> db["produtos"].find(
  {
    categoria: {$in: ["Smartphones", "Monitores"]}, preco: {$gte: 1000, $lte: 5000},
    estoque: {$gt: 10},
    ativo: true
  },
  {
    nome: 1,
    preco: 1,
    estoque: 1,
    "fabricante.nome": 1,
    _id: 0
  }
)
< {
  preco: 4200,
  estoque: 25,
  fabricante: {
    nome: 'Samsung'
  }
}
> db["produtos"].update({categoria: "Periféricos"}, {$inc: {estoque: 20}})
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
techstore_db>
```

MongoDB Compass - test/Shell
Connections Edit View Help

Compass

My Queries

CONNECTIONS (1) x + ...

Search connect

- test
- admin
- config
- local
- techstore_db
 - produtos

```
> .MONGOSSH
  "fabricante.nome": 1,
  _id: 0
}
)
< {
  preco: 4200,
  estoque: 25,
  fabricante: {
    nome: 'Samsung'
  }
}
> db["produtos"].update({categoria: "Periféricos"}, {$inc: {estoque: 20}})
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db["produtos"].update({estoque: {$lt: 10}}, {$multiply: {preco: 0.85}})
> db["produtos"].updateMany({estoque: {$lt: 10}, ativo: true}, {$mul: {preco: 0.85}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
techstore_db>
```

MongoDB Compass - test/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1) < + ...

Search connect

test

- admin
- config
- local
- techstore_db
 - produtos

```
> .MONGOSH
nome: 'sausage'
}
}
> db["produtos"].update({categoria: "Periféricos"}, {$inc: {estoque: 20}})
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db["produtos"].update({estoque: {$lt: 10}}, {$multiply: {preco: 0.85}})
< MongoServerError: Unknown modifier: $multiply. Expected a valid update modifier or pipeline-style update specified as an array
> db["produtos"].updateMany({estoque: {$lt: 10}, ativo: true}, {$mul: {preco: 0.85}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
> db["produtos"].updateOne({nome: "Mouse Logitech MX Master 3"}, {$push: {avaliacoes: 9.2}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
techstore_db>
```

Windows taskbar: 20:48, 03/10/2023

Questão 8 - DELETE:

Resposta: A)

MongoDB Compass - test/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1) < + ...

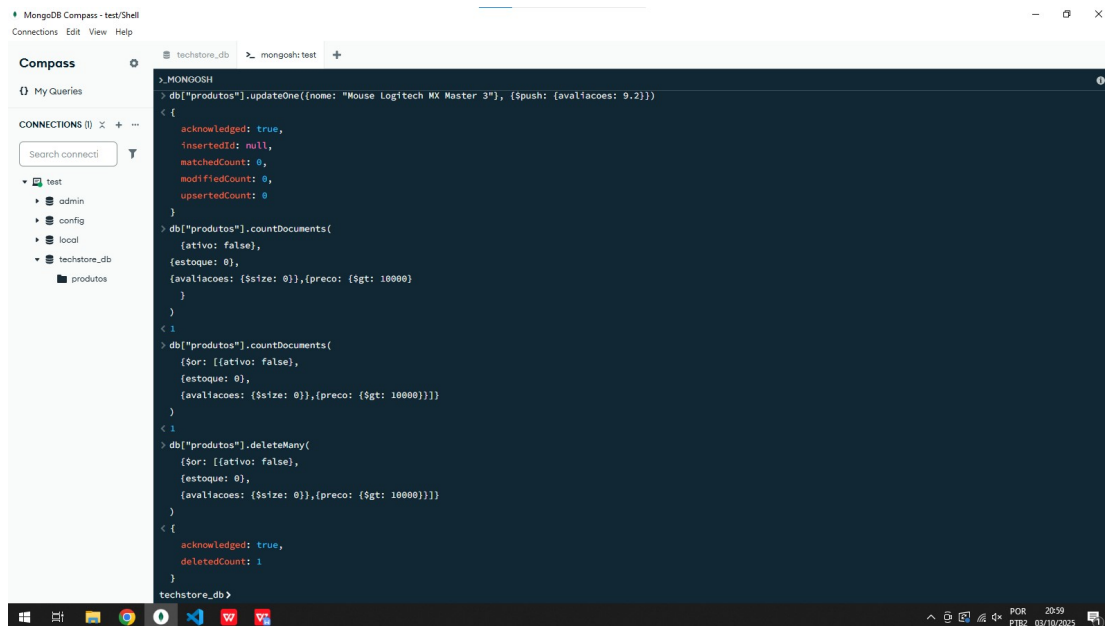
Search connect

test

- admin
- config
- local
- techstore_db
 - produtos

```
> .MONGOSH
< MongoServerError: Unknown modifier: $multiply. Expected a valid update modifier or pipeline-style update specified as an array
> db["produtos"].updateMany({estoque: {$lt: 10}, ativo: true}, {$mul: {preco: 0.85}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
> db["produtos"].updateOne({nome: "Mouse Logitech MX Master 3"}, {$push: {avaliacoes: 9.2}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
> db["produtos"].countDocuments(
  {ativo: false},
  {estoque: 0},
  {avaliacoes: {$size: 0}}, {preco: {$gt: 10000}}
)
< 1
> db["produtos"].countDocuments(
  {$or: [{ativo: false},
  {estoque: 0},
  {avaliacoes: {$size: 0}}, {preco: {$gt: 10000}}]}
)
< 1
techstore_db>
```

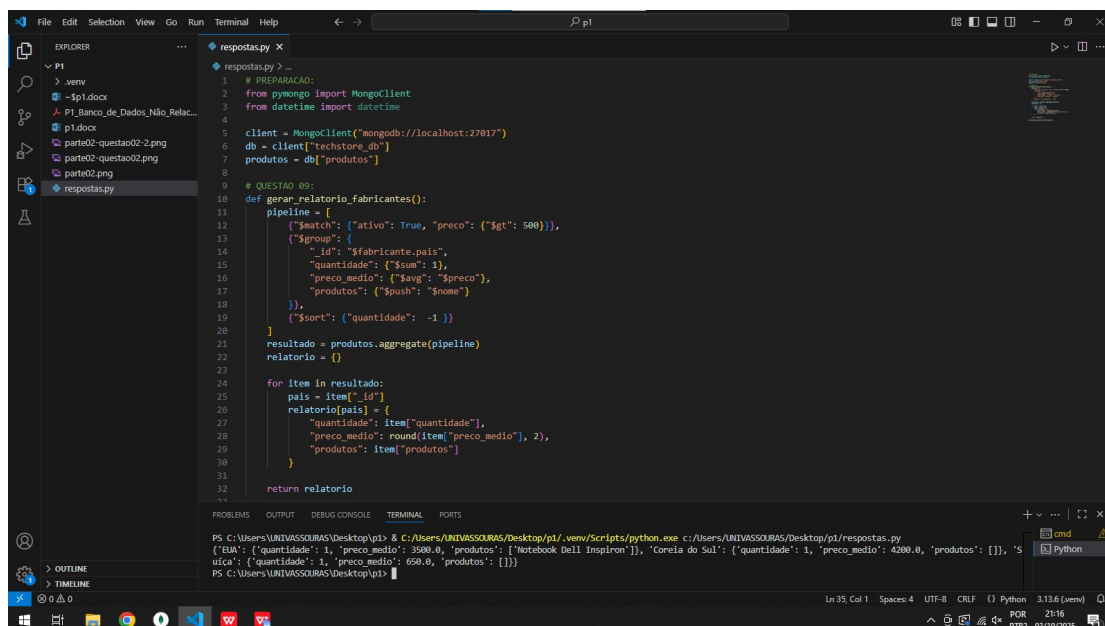
Windows taskbar: 20:59, 03/10/2023



PARTE 2 QUESTÕES DO PYTHON:

Questão 9 - READ:

Resposta: A)



Questão 10 - UPDATE, DELETE E VALIDAÇÃO:

Resposta: B)

