

# Unidade 1:

## Pergunta 1: Resumo da Evolução dos Bancos de Dados

A evolução dos bancos de dados tem sido um marco da era da informação, começando com sistemas de arquivos simples que eram difíceis de gerenciar. Surgiram os bancos de dados hierárquicos e em rede, proporcionando uma forma mais estruturada de gerenciamento de dados. No entanto, foi o advento dos bancos de dados relacionais, conforme proposto por E.F. Codd em 1970, que revolucionou o armazenamento de dados, permitindo que os dados fossem armazenados em formatos de tabelas com relações entre essas tabelas definidas por chaves estrangeiras.

A introdução do SQL como uma linguagem de consulta padrão solidificou ainda mais a dominância dos bancos de dados relacionais. Com o tempo, a necessidade de lidar com tipos de dados mais complexos e grandes volumes de dados levou ao desenvolvimento de bancos de dados orientados a objetos e, mais tarde, a bancos de dados NoSQL, incluindo lojas de documentos como o MongoDB, lojas de colunas, lojas de chave-valor e bancos de dados gráficos.

Atualmente, o mundo corporativo depende fortemente de bancos de dados para análise de dados, gestão de relacionamento com clientes, gestão da cadeia de suprimentos e muito mais. Eles servem como a espinha dorsal para processos de tomada de decisão e operações diárias, com soluções baseadas na nuvem como AWS RDS e Azure SQL Database fornecendo serviços de banco de dados escaláveis e confiáveis.

## Pergunta 2: Função dos SGBDs, Principais Características, Estrutura e Teorema CAP

Sistemas de Gerenciamento de Bancos de Dados (SGBDs) atuam como intermediários entre usuários e bancos de dados, fornecendo uma maneira sistemática de criar, recuperar, atualizar e gerenciar dados. SGBDs garantem a integridade, consistência e segurança dos dados. Eles suportam modelos de dados, como o modelo relacional, e fornecem uma linguagem de consulta como SQL.

As principais características de um SGBD incluem abstração de dados, suporte a transações, controle de concorrência e independência de dados. A estrutura típica de um SGBD compreende o motor do banco de dados, que lida com o armazenamento, recuperação e atualização de dados; um processador de consultas, que interpreta e executa consultas ao banco de dados; e um gerenciador de transações, que garante a integridade dos dados em transações multioperações.

O Teorema CAP, formulado por Eric Brewer, afirma que um sistema de banco de dados distribuído só pode garantir duas das seguintes três propriedades: Consistência (cada leitura recebe a escrita mais recente), Disponibilidade (cada solicitação recebe uma resposta) e Tolerância a Partições (o sistema continua operando apesar de partições na rede). O MongoDB, como um banco de dados NoSQL, oferece opções de configuração que permitem ajustar o equilíbrio entre essas propriedades. Por exemplo, ele pode ser configurado para forte

consistência ou para consistência eventual, a fim de melhorar a disponibilidade, refletindo diferentes aspectos do Teorema CAP dependendo das necessidades da aplicação.

### **Pergunta 3: Baixando o Conjunto de Dados do Kaggle**

Acessei o Kaggle e procurei pelo conjunto de dados fornecido no link. Após localizar o conjunto de dados, segui o processo para baixá-lo, que exigiu que eu fizesse login na minha conta do Kaggle. Uma vez logado, encontrei a página específica do conjunto de dados e cliquei no botão "Download", que baixou um arquivo ZIP contendo os dados. Extraí o conteúdo usando uma ferramenta de descompressão de arquivos e revisei os arquivos CSV incluídos para me familiarizar com a estrutura e conteúdos dos dados em preparação para análise ou importação para um sistema de banco de dados para manipulação adicional.

## **Unidade 2:**

### **Pergunta 1: Criando um Novo Banco de Dados e Coleção**

Para criar um novo banco de dados e uma coleção dentro dele para receber um conjunto de dados, segui estes passos:

#### **Fazer login no MongoDB:**

Acessei a instância do MongoDB onde queria criar o novo banco de dados e coleção.

#### **Criar Banco de Dados:**

Usando o shell do MongoDB ou uma ferramenta de gerenciamento do MongoDB como o MongoDB Compass, executei o seguinte comando para criar um novo banco de dados chamado Musical\_Instruments\_Database:

**Código:** use Musical\_Instruments\_Database

Este comando muda para o contexto de um banco de dados. Se o banco de dados não existir, o MongoDB o cria quando você armazena dados pela primeira vez nesse banco de dados (como criar uma nova coleção no próximo passo).

#### **Criar Coleção:**

Dentro do contexto do banco de dados Musical\_Instruments\_Database, criei uma nova coleção chamada Musical\_Instruments\_Data com o seguinte comando:

**Código:** db.createCollection("Musical\_Instruments\_Data")

Após executar esses comandos, eu tinha um novo banco de dados e uma coleção prontos para receber o conjunto de dados.

### **Pergunta 2: Importando o Novo Conjunto de Dados**

Com a nova coleção no lugar, o próximo passo foi importar o conjunto de dados para a coleção

Musical\_Instruments\_Data. Segui estes passos:

### **Preparar o Conjunto de Dados:**

Certifiquei-me de que o conjunto de dados estava em um formato compatível com o MongoDB, como JSON ou CSV. Se estivesse em um formato diferente, converti-o usando uma ferramenta de transformação de dados ou script.

### **Usar a Ferramenta mongoimport:**

O MongoDB fornece uma ferramenta de linha de comando chamada mongoimport para importar dados para uma coleção. Usei essa ferramenta para importar o conjunto de dados para a coleção recém-criada. O comando seria algo assim:

**Código:** mongoimport --db Musical\_Instruments\_Database --collection Musical\_Instruments\_Data --file /caminho/para/o/arquivo/do/conjunto/de/dados.json --jsonArray

--db: Especifica o nome do banco de dados.

--collection: Especifica o nome da coleção.

--file: Especifica o caminho para o arquivo do conjunto de dados.

--jsonArray: Indica que o arquivo de entrada contém um array JSON (se aplicável).

### **Executar o Comando de Importação:**

Após preparar o comando com o caminho correto para o conjunto de dados, executei o comando mongoimport a partir do terminal.

### **Verificar a Importação:**

Para confirmar que os dados foram importados corretamente, executei um comando find no shell do MongoDB:

**Código:** db.Musical\_Instruments\_Data.find().limit(5)

Este comando exibe os primeiros cinco documentos na coleção Musical\_Instruments\_Data para verificar a importação.

Seguindo esses passos, criei com sucesso um novo banco de dados e coleção e importei o conjunto de dados para o MongoDB, populando a coleção Musical\_Instruments\_Data com os dados.

•

Filter  { "overall": 5 }

ResetFind

ADD DATAEXPORT DATA

1 - 20 of 6938 

```
_id: ObjectId('65a931c486508c3766ba12cf')
reviewerID: "A2IBPI20UZIR0U"
asin: "1384719342"
reviewerName: "cassandra tu "Yeah, well, that's just like, u..."
helpful: Array
reviewText: "Not much to write about here, but it does exactly what it's supposed t..."
overall: 5
summary: "good"
unixReviewTime: 1393545600
reviewTime: "02 28, 2014"
```

```
_id: ObjectId('65a931c486508c3766ba12d0')
reviewerID: "A14VAT5EAX3D9S"
asin: "1384719342"
reviewerName: "Jake"
helpful: Array
reviewText: "The product does exactly as it should and is quite affordable.I did no..."
overall: 5
summary: "Jake"
unixReviewTime: 1363392000
reviewTime: "03 16, 2013"
```

## Segundo Objetivo de Consulta:

### Analisar o Engajamento dos Avaliadores Baseado em Votos 'Úteis':

- Objetivo: Identificar avaliações que receberam um número notável de votos como 'úteis'.
- Justificativa: Avaliações consideradas úteis por outros clientes podem conter feedback mais detalhado ou perspicaz sobre o produto.
- Exemplo de Consulta: `db.collection.find({ "helpful.1": { "$gt": 10 } })` // Avaliações onde mais de 10 pessoas consideraram a avaliação útil.

Filter
▼
{ "helpful.1": { "\$gt": 10 }}
Reset
Find

+ ADD DATA
EXPORT DATA
1 - 20 of 343

```

_id: ObjectId('65a931c486508c3766ba12d0')
reviewerID: "A14VAT5EAX3D9S"
asin: "1384719342"
reviewerName: "Jake"
helpful: Array
reviewText: "The product does exactly as it should and is quite affordable.I did no..."
overall: 5
summary: "Jake"
unixReviewTime: 1363392000
reviewTime: "03 16, 2013"

```

```

_id: ObjectId('65a931c486508c3766ba12fd')
reviewerID: "A13A81NN0NRDIS"
asin: "B000068NW5"
reviewerName: "aspiring saint 'TATITTLE'"
helpful: Array
reviewText: "Cheap and good texture rubber that does not get stiff. Only time will..."
overall: 4
summary: "flexible, soft rubber is great."
unixReviewTime: 1290038400
reviewTime: "11 18, 2010"

```

## Terceiro Objetivo de Consulta:

### Análise Baseada em Tempo das Avaliações:

- Objetivo: Extrair avaliações dentro de um período de tempo específico (por exemplo, avaliações escritas em 2014).
- Justificativa: Na captura de tela abaixo, você pode ver informações sobre como as percepções dos produtos podem ter mudado ao longo do tempo ou durante períodos específicos (como após uma atualização do produto ou durante uma campanha promocional).
- Exemplo de Consulta: `db.collection.find({ "unixReviewTime": { "$gte": UnixTimeStart, "$lte": UnixTimeEnd } })`
  - Aqui, `UnixTimeStart` e `UnixTimeEnd` representam o tempo Unix no início e no final de 2014, respectivamente.

Filter 

 { "unixReviewTime": { "\$gte": 1388534400, "\$lte": 1419984000 } }

ResetFind

ADD DATA

EXPORT DATA

1 - 20 of 2679 

```
_id: ObjectId('65a931c486508c3766ba12cf')
reviewerID: "A2IBPI20UZIR0U"
asin: "1384719342"
reviewerName: "cassandra tu "Yeah, well, that's just like, u..."
helpful: Array
reviewText: "Not much to write about here, but it does exactly what it's supposed t..."
overall: 5
summary: "good"
unixReviewTime: 1393545600
reviewTime: "02 28, 2014"
```

```
_id: ObjectId('65a931c486508c3766ba12d2')
reviewerID: "A2C00NNG1ZQG2"
asin: "1384719342"
reviewerName: "RustyBill "Sunday Rocker"
helpful: Array
reviewText: "Nice windscreen protects my MXL mic and prevents pops. Only thing is t..."
overall: 5
summary: "GOOD WINDSCREEN FOR THE MONEY"
unixReviewTime: 1392336000
reviewTime: "02 14, 2014"
```

**Pergunta 2: Execute as consultas sem criar índices e observe o desempenho (em termos de tempo de execução) e o número de documentos acessados.**

**Resposta: Execute a Consulta Sem um Índice:**

Podemos executar a consulta sem um índice da seguinte maneira: { "overall": 5 } ou { "unixReviewTime": { "\$gte": 1388534400, "\$lte": 1419984000 } }.

Filter  { "overall": 5 }

ResetFind

ADD DATAEXPORT DATA

1 - 20 of 6938 

```
_id: ObjectId('65a931c486508c3766ba12cf')
reviewerID: "A2IBPI20UZIR0U"
asin: "1384719342"
reviewerName: "cassandra tu "Yeah, well, that's just like, u..."
helpful: Array
reviewText: "Not much to write about here, but it does exactly what it's supposed t..."
overall: 5
summary: "good"
unixReviewTime: 1393545600
reviewTime: "02 28, 2014"
```

```
_id: ObjectId('65a931c486508c3766ba12d0')
reviewerID: "A14VAT5EAX3D9S"
asin: "1384719342"
reviewerName: "Jake"
helpful: Array
reviewText: "The product does exactly as it should and is quite affordable.I did no..."
overall: 5
summary: "Jake"
unixReviewTime: 1363392000
reviewTime: "03 16, 2013"
```

## Medição de Desempenho:

- O MongoDB Compass não mostra diretamente o tempo de execução ou o número de documentos escaneados para cada consulta em sua interface padrão.
- Para medir essas métricas, normalmente você utilizaria o método `explain()` em um shell do MongoDB (Veja a Captura de Tela Anexada).
- O método `explain("executionStats")` anexado à sua consulta fornecerá detalhes sobre o tempo de execução e o número de documentos acessados.

```
> db.Musaical_Instruments_Data.find({ "overall": 5 }).explain("executionStats");
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'Musaical_Instruments_Database.Musaical_Instruments_Data',
    indexFilterSet: false,
    parsedQuery: {
      overall: {
        '$eq': 5
      }
    },
    queryHash: '4DCC5541',
    planCacheKey: '4DCC5541',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: {
        overall: {
          '$eq': 5
        }
      },
      direction: 'forward'
    },
  },
}
```



```
executionStats: {
  executionSuccess: true,
  nReturned: 6938,
  executionTimeMillis: 16,
  totalKeysExamined: 0,
  totalDocsExamined: 10261,
  executionStages: {
    stage: 'COLLSCAN',
    filter: {
      overall: {
        '$eq': 5
      }
    },
  },
  nReturned: 6938,
  executionTimeMillisEstimate: 0,
  works: 10263,
  advanced: 6938,
  needTime: 3324,
  needYield: 0,
  saveState: 10,
  restoreState: 10,
  isEOF: 1,
  direction: 'forward',
  docsExamined: 10261
},
```

```

command: {
  find: 'Musaical_Instruments_Data',
  filter: {
    overall: 5
  },
  '$db': 'Musaical_Instruments_Database'
},
serverInfo: {
  host: 'DESKTOP-2JCQUI9',
  port: 27017,
  version: '6.0.6',
  gitVersion: '26b4851a412cc8b9b4a18cdb6cd0f9f642e06aa7'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600
},
ok: 1
}

```

## Usando o Shell do MongoDB para Estatísticas Detalhadas de Execução:

- O primeiro passo que realizei foi executar sua consulta anexada a `.explain("executionStats")`.
- Por exemplo: `db.collection.find({ "overall": 5 }).explain("executionStats")`.
- Isso retornará um relatório detalhado, incluindo o tempo de execução (`executionTimeMillis`) e o número de documentos examinados (`totalDocsExamined`).

```
executionStats: {
  executionSuccess: true,
  nReturned: 6938,
  executionTimeMillis: 16,
  totalKeysExamined: 0,
  totalDocsExamined: 10261,
  executionStages: {
    stage: 'COLLSCAN',
    filter: {
      overall: {
        '$eq': 5
      }
    },
    nReturned: 6938,
    executionTimeMillisEstimate: 0,
    works: 10263,
    advanced: 6938,
    needTime: 3324,
    needYield: 0,
    saveState: 10,
    restoreState: 10,
    isEOF: 1,
    direction: 'forward',
    docsExamined: 10261
  }
}
```

## Impacto Sem Índice:

- Quando as consultas são executadas sem índices, o MongoDB examina cada documento na coleção em busca de correspondências, o que pode ser demorado, especialmente para coleções grandes.