MongoDB

Teoria e Prática

Team

João Vitor
Desenvolvedor Back-end

Cursando ADS - UNOESTE

Henrique Ricci
Desenvolvedor Back-end

Bacharel em Ciência da Computação
Especialista em Inteligência Artificial e ML
Mestrando em Ciência da Computação

Tulio Olivieri
Desenvolvedor Back-end

Bacharel em Ciência da Computação
Especialista em Engenharia de Software
Mestrando em Ciência da Computação

NoSQL?

"Não SQL"?

Not Only SQL.

Como funciona um banco de dados NoSQL?

- Os modelos de dados armazenados não são tabelas;
- Possuem modelos de dados específicos e esquemas flexíveis;
- Garantem disponibilidade, mas não garantem consistência imediata;

Por que usar?

Flexibilidade: os bancos de dados NoSQL geralmente fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e iterativo. O modelo de dados flexível torna os bancos de dados NoSQL ideais para dados semiestruturados e não estruturados.

Escalabilidade: os bancos de dados NoSQL geralmente são projetados para serem escalados horizontalmente usando clusters distribuídos de hardware, em vez de escalá-los verticalmente adicionando servidores caros e robustos. Alguns provedores de nuvem lidam com essas operações nos bastidores como um serviço totalmente gerenciado.

Alta performance: o banco de dados NoSQL é otimizado para modelos de dados específicos e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com bancos de dados relacionais.

Altamente funcional: os bancos de dados NoSQL fornecem APIs e tipos de dados altamente funcionais criados especificamente para cada um de seus respectivos modelos de dados.

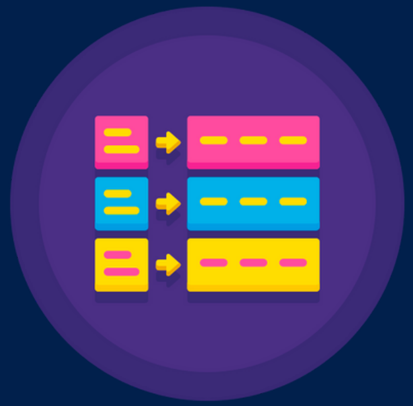
Fonte: AWS

Tipos de bancos de dados NoSQL

Documento

Grafo

Memória



Colunar

Documento

Documento

```
// Este documento representa um usuário da Netflix
{
  "_id": "1234567890",
  "nome": "Jubileu",
  "email": "jubileu@email.com",
  "senha": "123456",
  "plano": "Premium",
  "histórico": [
    {
      "id_titulo": "1234567890",
      "titulo": "O Poderoso Chefão",
      "data_visualizacao": "2023-07-20T20:00:00Z",
      "avaliacao": 5
    },
    {
      "id_titulo": "1234567891",
      "titulo": "Matrix",
      "data_visualizacao": "2023-07-21T10:00:00Z",
      "avaliacao": 4
    }
  ],
  "preferencias": {
    "generos": ["ação", "drama"],
    "atores": ["Al Pacino", "Keanu Reeves"]
  }
}
```

mongoDB

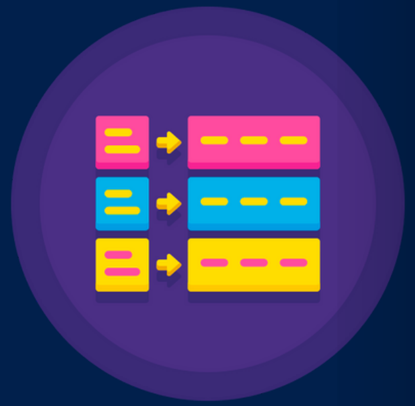
Grafo

Grafo

```
// Cria o nó para Henrique
CREATE (Henrique:Usuario {
  id: "1234567890",
  nome: "João da Silva"
})

// Cria o nó para Tulio
CREATE (Tulio:Usuario {
  id: "9876543210",
  nome: "Maria da Silva"
})

// Cria a aresta entre Henrique e Tulio
CREATE (Henrique)-[:AMIGO {tipo: "amizade"}]->(Tulio)

// Recupera a lista de amigos de Henrique
MATCH (Henrique:Usuario {id: "1234567890"})-[:AMIGO]->(amigos)
RETURN amigos
```

Memória

Memória



```
// Cria um novo cliente Redis
const redis = new Redis();

// Cria um novo registro de sessão para o usuário com o ID
1234567890
const dados_sessao = {
  id: "1234567890",
  nome_usuario: "Jubileu",
  estado_sessao: "ativo",
  itens_carrinho: [
    "1234567890",
    "9876543210"
  ]
};

// Armazena o registro de sessão no Redis
redis.set("sessões:1234567890", JSON.stringify(dados_sessao));

// Recupera os dados da sessão do usuário com o ID 1234567890
const dados_sessao_recuperado =
JSON.parse(redis.get("sessões:1234567890"));
```

Colunar

Colunar

```sql
// Cria uma tabela colunar para armazenar dados de log
CREATE TABLE logs (
  data_hora DATE,
  ip_origem VARCHAR,
  url VARCHAR,
  metodo VARCHAR,
  codigo_status INT
);

// Insere um novo registro de dados de log
INSERT INTO logs (data_hora, ip_origem, url, metodo, codigo_status)
VALUES ('2023-07-20', '192.168.0.1', '/home', 'GET', 200);

// Recupera as solicitações HTTP recebidas do IP 192.168.0.1
SELECT COUNT(*) AS solicitacoes_http
FROM logs
WHERE ip_origem = '192.168.0.1';

// Recupera as solicitações HTTP com código de status 200
SELECT COUNT(*) AS solicitacoes_sucesso
FROM logs
WHERE codigo_status = 200;
```

Google Analytics

```
// Dados de log
data_hora | ip_origem | url | metodo | codigo_status
--------- | --------- | --------- | --------- | ---------
2023-07-20 | 192.168.0.1 | /home | GET | 200
2023-07-20 | 192.168.0.2 | /about | GET | 200
2023-07-21 | 192.168.0.3 | /contact | GET | 200
```

SQL vs NoSQL

- Estrutura;
- Mecanismo de integridade dos dados;
- Performance;
- Escala.

Quando usar bancos relacionais ou não relacionais

Bancos relacionais:

- são previsíveis em termos de tamanho, estrutura e frequência de acesso;
- os relacionamentos entre entidades forem importantes.

Bancos não relacionais:

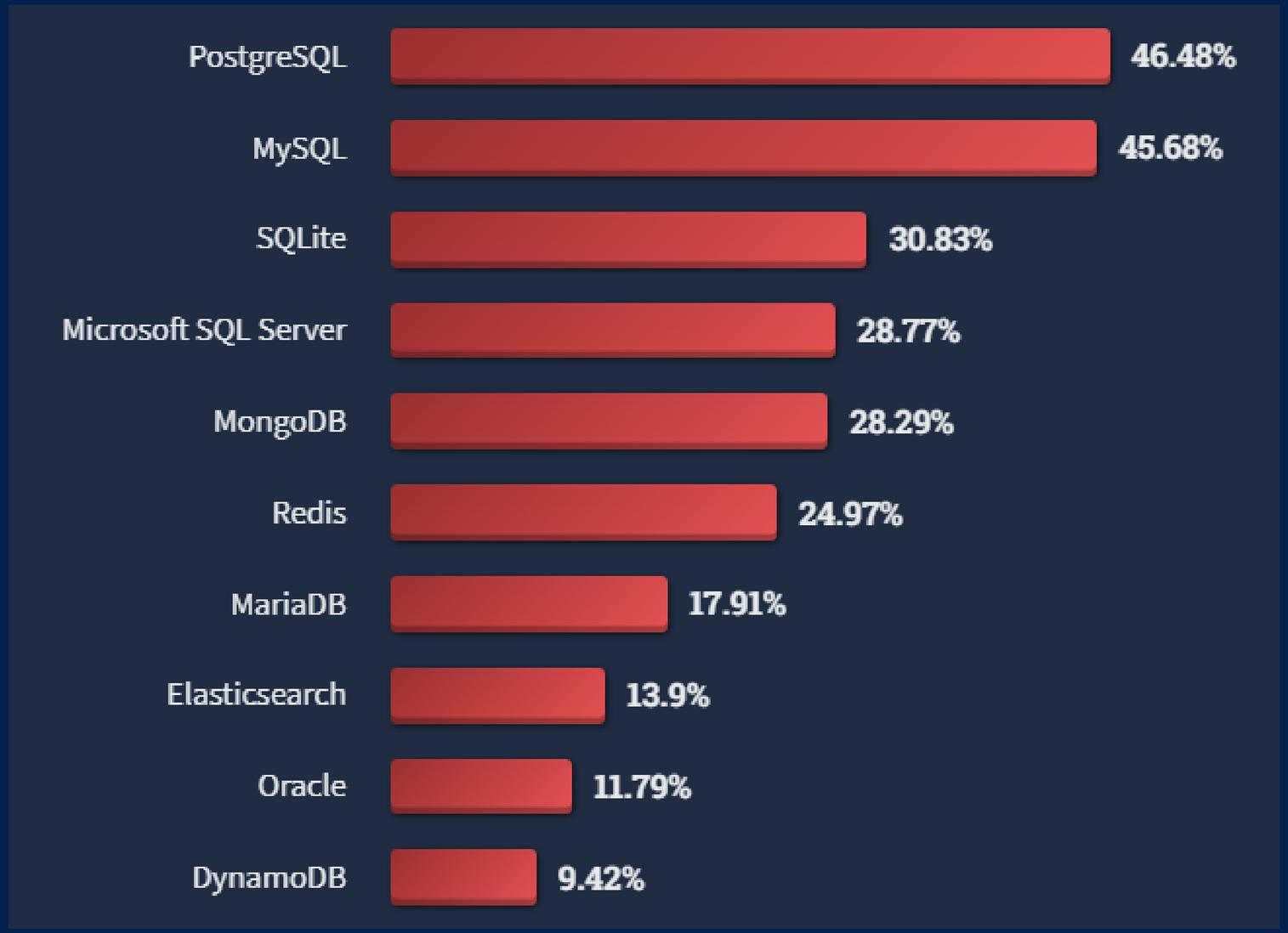
- funciona melhor para armazenar dados flexíveis em forma ou tamanho, ou que possam mudar no futuro;
- as relações de dados simplesmente não se encaixam bem no formato tabular de chaves primárias e estrangeiras.

Resumo

Categoria	Banco relacional	Banco não relacional
Modelo de dados	Tabular	Chave-valor, documento ou grafo
Tipo de dados	Estruturado	Dados estruturados, semi estruturados e não estruturados
Integridade dos dados	Total conformidade com ACID	Modelo de consistência eventual.
Performance	Aprimorada com a adição de mais recursos ao servidor	Melhorada com a adição de mais nós de servidor
Escalabilidade	O ajuste de escala horizontal requer estratégias adicionais de gerenciamento de dados	O ajuste de escala horizontal é simples e direto

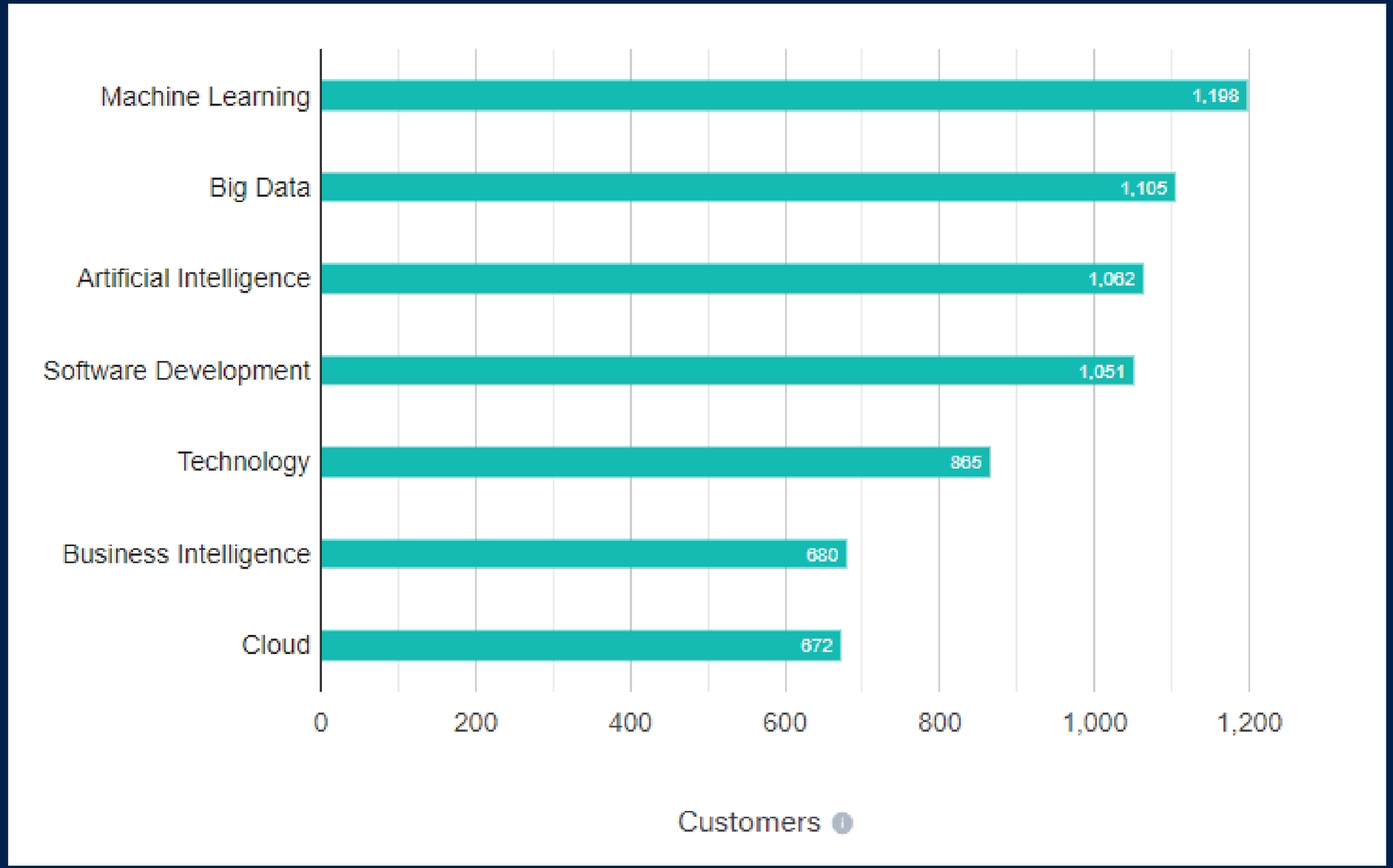
Fonte: AWS

NoSQL no Mercado



Fonte: stackoverflow

NoSQL no Mercado



Fonte: 6sense.com

NoSQL no Mercado

- **Chave-valor:** cache, flags, jogos...
- **Documento:** catálogos, perfis em redes sociais, logs, big data...
- **Grafos:** redes sociais, sistemas de recomendação, detecção de fraude...
- **Em memória:** armazenamento de sessão, cache, flags...
- **Colunas**: Analytics, logs, data processing, data warehouse...

Fonte: AWS

Um pouco de história...

Década de 1960

Primeiros bancos não relacionais
Ainda não eram chamados de NoSQL

1998

Carlo Strozzi
NoSQL
Projeto Open Source
Primeira referência a NoSQL

2009

Eric Evans
Johan Oskarsson
Open Source
Bancos de dados distribuídos

Dias atuais

Big Data
Logs
Eventos
Grandes registros

Fonte: Awari, AWS, DevMedia, Strozzi.it,

MongoDB

2007

10gen
Componente de um PaaS

2009

Open-source
Suporte comercial

2013

10gen se torna MongoDB Inc.

2017

IPO
Listada na NASDAQ

Fonte: MongoDB

Estrutura

- Cluster
- Database → Banco de dados
- Collection → Tabela
- Document → Linha da tabela
- Indexes
- Constraints

BSON

- Binary JSON
- Formato de serialização binário
- Leve
- Fácil de ser transportado
- Eficiente

Fonte: MongoDB, bsonspec,

BSON x JSON

- JSON:
  - Legível por humanos
  - Utilizado por outros bancos de dados
  - Tipos suportados:
    - Strings
    - Números
    - Arrays
    - Objetos
    - null

Fonte: MongoDB, geeksforgeeks

BSON x JSON

- BSON:
  - Binary file format
  - Utilizado pelo MongoDB
  - Alguns tipos suportados além do JSON:
    - Date
    - BinData → Array de Bytes
    - Regex
    - MinKey, MaxKey, Timestamp e ObjectId → MongoDB

Fonte: MongoDB, bsonspec, geeksforgeeks

BSON x JSON

```json
{
    "hello": "world"
}

```

```
\x16\x00\x00\x00                  // tamanho do documento
\x02                              // 0x02 = tipo String
hello\x00                         // nome do campo
\x06\x00\x00\x00world\x00         // valor do campo
\x00                              // 0x00 = type EOO (Final do objeto)

```

Fonte: stackoverflow

Ferramentas

- **MongoDB Shell** - CLI que permite maipulação do banco de dados
- **MongoDB Compass** - Interface que permite manipulação do banco, porém de forma visual
- **MongoDB Community Server** - Sistema Gerenciados de Banco de Dados (SGBD) - Permite ter um servidor MongoDB localmente

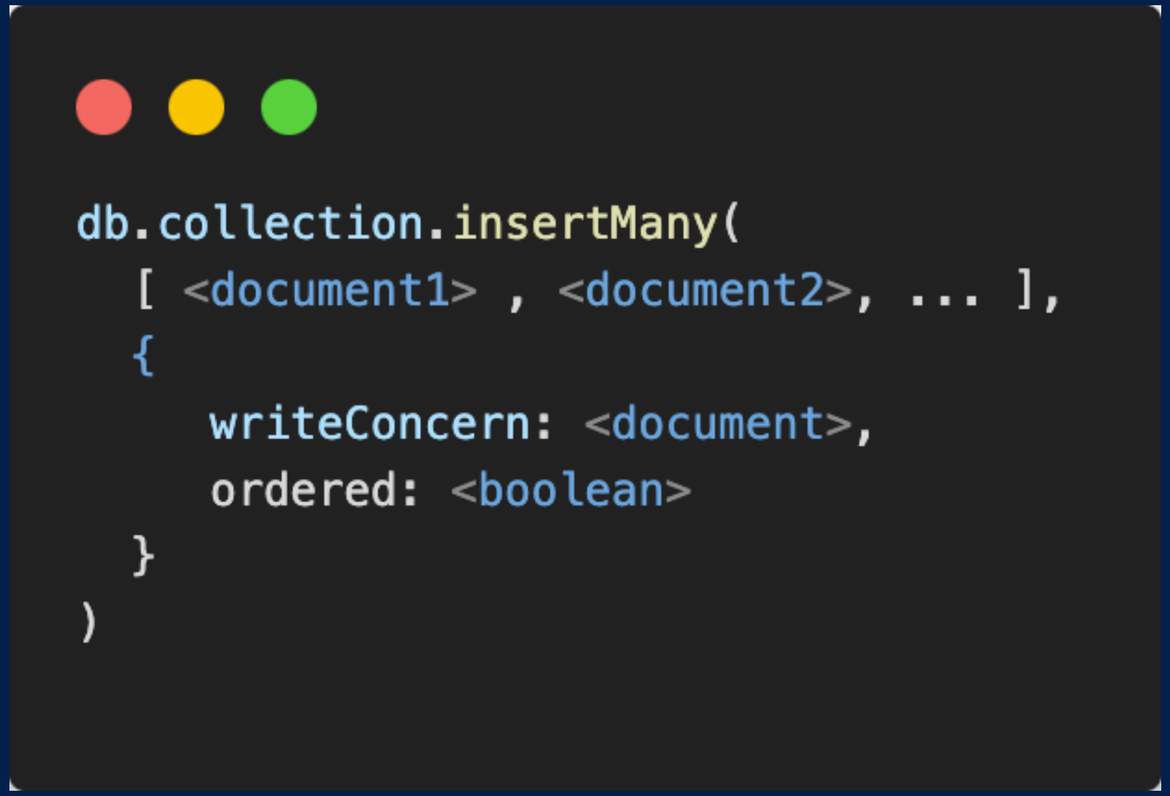
Conexão

- mongodb://localhost:27017/netflix

- mongodb+srv://aula:infoeste2023@aula.fnxovog.mongodb.net/netflix

db.collection.insertOne()

db.collection.insertMany()



```
db.collection.insertMany(
  [ <document1> , <document2>, ... ],
  {
     writeConcern: <document>,
     ordered: <boolean>
  }
)
```

Exercícios Insert

- Crie uma nova Collection dentro do seu database com o nome de insert_exercises
  - Insira um documento que contenha as informações básicas de uma pessoa: nome, telefone, idade, fazAcademia, usaSuco
  - Insira um documento com informações de um carro: marca, modelo, anoFabricacao, qtdeMultas, dataCompra, donos (array com pelo menos 3 pessoas com nome e cpf)
  - Insira um documento que contenha as informações de um estado brasileiro: nome, uf, cidades (array com pelo menos 4 cidades contendo nome, população, temPraia)

Exercícios Insert

- Crie uma nova Collection dentro do seu database com o nome de insert_exercises
  - Insira um documento que contenha as informacoes básicas de uma pessoa nome, cpf, telefone. Além disso, deverá haver um campo endereco, contendo rua, cidade, estado, cep)
  - Insira 10 documentos de produtos da amazon, contendo nome, preco, qtdeEstoque e nota

db.collection.deleteOne()

```
db.collection.deleteOne(
  <filter>,
  {
    writeConcern: <document>,
    collation: <document>,
  }
)
```

db.collection.deleteMany()

```
db.collection.deleteMany(
  <filter>,
  {
    writeConcern: <document>,
    collation: <document>,
  }
)
```

Exercícios Delete

- Importe a collection shopping para dentro do seu database. Um erro no sistema acabou gerando inconsistência nos documentos dessa collection. Faça a remoção dos registros incorretos
  - Nada da cor Lavanda foi vendida
  - A coleção de verão ainda não chegou para venda
  - O serviço de desconto está com problema desde o dia que foi lançado. Nenhum cupom pode ser usado
  - A máquina de cartões não está funcionando. Não deveriam existir compras no crédito ou débito.

db.collection.updateOne()

```
db.collection.updateOne(
  <filter>,
  <update>,
  {
    upsert: <boolean>,
    writeConcern: <document>,
    collation: <document>,
    arrayFilters: [ <filterdocument1>, ... ],
  }
)
```

db.collection.updateMany()

```
db.collection.updateMany(
  <filter>,
  <update>,
  {
    upsert: <boolean>,
    writeConcern: <document>,
    collation: <document>,
    arrayFilters: [ <filterdocument1>, ... ],
  }
)
```

Update

Operadores

- set - Atribui o valor ao campo
- unset - Remove um campo do documento
- rename - Renomeia um campo
- inc - Incrementa o campo de acordo com o valor passado
- mul - Multiplica o campo de acordo com o valor passado

- addToSet
- pop
- pull
- push

- each
- slice
- sort
- $[<identifier>]

Exercícios Update

- Importe a collection bugs_update para dentro do seu database e corrja todas as inconsistências encontradas
  - Dicas:
    - Fique de olho no nome dos campos, seja significado ou escrita
    - Confirme se os valores estão condizentes com todo o resto do documento, alguns valores podem estar muito maiores ou menores
    - Alguns registros possuem exatamente o mesmo erro, preste atenção

db.collection.find()

Query

Comparação

- eq - Igual
- gt - Maior
- gte - Maior ou Igual
- lt - Menor
- lte - Menor ou Igual
- ne - Diferente

```
{ field: { $op: value } }
```

- in - Está em
- nin - Não está em

```
{ field: { $op: [ value1, value2, ..., valueN] } }
```

Query
Lógicos

- and - e
- or - ou
- nor - não ou

{ $op: [{exp1}, {exp2}, ..., {expN}] }

- not - não

{ $not: { exp } }

Query
Array

- $all - encontra documentos que contenha o mesmo array passado na busca { field: { $all: [elem1, elem2, ..., elemN] } }
- $elemMatch - Encontra documentos se os itens do array satisfazem a condição { field: { $elemMatch: { exp1, ..., expN } } }
- $size - retorna documentos que o tamanho do array seja o mesmo passado na busca { field: { $size: N } }

Projection

- **{ field: true | 1 | false | 0 }**

Options

- **sort - Ordena o resultado. Caso 1, ordena o campo de forma crescente. Caso -1, ordena de forma decrescente**
- **limit - Limite a quantidade de registros retornados**
- **skip - Pula uma determinada quantidade de registros a partir do primeiro.**

Exercícios Find

- Importe a collection netflix para dentro do seu database.
  - Quais os filmes com nota acima de 9?
  - Quais os filmes lançados antes de 2006?
  - Quais filmes possuem orçamento maior que 100 milhões de reais?
  - Quais filmes não possuem idioma original em inglês?
  - Quais filmes não estão traduzidos nem em inglês nem em espanhol?
  - Qual o filme com mais palavras chave?
  - Qual o filme com maior popularidade? E com a menor?
  - Qual o filme da Disney com maior orçamento? E o menor?
  - Quais os filmes lançados entre 2010 e maio de 2014?

Exercícios Find

- Importe a collection netflix para dentro do seu database.
  - Quais filmes falam sobre flashback, possuem nota entre 4 e 6.9 e não possuem site?
  - Qual o 64 filme com menor receita?
  - Quais os 3 filmes com menos avaliações e nota inferior a 4? Os filmes deve conter ao menos 10 avaliação
  - Quais os 10 filmes mais longos que possuem tradução para italiano lançados depois de 24 de Fevereiro de 2015
  - Quais filmes foram originalmente gravados em inglês ou francês, ou possuem mais de 100 minutos de duração e não são do ano de 1999 e a nota não é menor que 7.2 nem possui menos de 300 votos

Exercícios Find

- Importe a collection songs para dentro do seu database.
  - Qual a 12 música não explicita com maior dançabilidade e menor instrumentalidade e com energia maior que 0.8
  - Quais músicas estão sem informações de artistas?
  - Quais as músicas que Vintage Culture gravou sozinho?

Aggregation Framework

- Processamento dos registros salvos
- Processamento / Análise de dados
- 28 estágios diferentes
- Pipeline:
  - Começa com um conjunto de documentos
  - Cada estágio recebe documentos, os processa e retorna documentos
  - Sintaxe do estágio igual a dos documentos

Fonte: MongoDB

Aggregation Framework

```
{
  $match: {
    "src_airport": "PDX",
    "stops": 0
  }
}
```

Fonte:MongoDB

Aggregation Framework

```
[{
    "$match": {
      "src_airport": "PDX",
      "stops": 0
    }
  },
  {
    "$group": {
      "_id": {
        "airline name": "$airline.name"
      },
      "count": {
        "$sum": 1
      }
    }
  },
  {
    "$sort": {
      "count": -1
    }
}]
```

Adaptado de: MongoDB

Aggregation Framework

- Poderosa ferramenta incorporada dentro do MongoDB que permite o agrupamento de dados
- Aggregation Pipelines
- Diversos estágios filtram, convertem e transformam os dados, gerando um resultado mais completo que para algumas buscas
- O resultado do estágio N é o retorno do estágio N - 1

Aggregation Framework

Operadores de estágio

- $match - Filtra os registros, como o find
- $group - Permite o agrupamento. Nesse estágio, é possível utilizar avg, max, min, sum
- $sort - Permite ordenar o resultado
- $project - Usado para "remodelar" a saída do estágio
- $unwind - Extrai elementos de um documento

Exercícios Aggregation

- Dentro da collection songs:
  - Quais as músicas que Vintage Culture gravou sozinho?
  - Quais as 10 músicas mais novas?
  - Qual a média de duração das músicas?
  - Qual a média de duração das músicas por ano?
  - Quais as músicas não foram lançadas no mês de Março?
  - Quantas músicas foram lançadas em 2017?
  - Quantas músicas explicitas e normais a base possui?

Links

- https://github.com/riccihenrique/mongo_databases

- https://www.mongodb.com/try/download/compass