



Banco dados para Big Data

Carlos Eduardo Rossi Cubas da Silva



Agenda

- Introdução
 - Dados estruturados
 - Dados semi estruturados
 - Json
 - Xml
- Bancos de dados NoSql
 - MongoDB

Introdução

“O que sabemos é uma gota; o que ignoramos é um oceano.” — Isaac Newton

Introdução



A cada dois anos, o universo digital dobra.

Em 2013, eram 4,4 trilhões de gigabytes no planeta crescendo para 44 trilhões de gigabytes até 2020;

No Brasil, o volume de dados deve ir de 212 bilhões de gigabytes em 2013 para 1.600 bilhões de gigabytes em 2020. Isso representa crescimento de 7,5 vezes.

Estima-se que, do total de dados no mundo, apenas 22% contêm informação útil.

Apenas 5% foram analisados e utilizados de alguma forma.

Introdução



O crescimento da chamada internet das coisas, a comunicação de máquina a máquina, sem interferência humana corresponde a apenas 2% dos dados produzidos no mundo. Mas deve atingir 10% deles em 2020.

Introdução

No Brasil, o governo disponibiliza muitas informações através do [portal da transparência](#). Neste site, pode-se baixar inúmeras informações sobre programas sociais, orçamentos, gastos entre outros dados.

The screenshot displays the official website of the Portal da Transparência. At the top, there is a navigation bar with links for 'Ir para o conteúdo', 'Ir para o menu', 'Ir para a busca', and 'Ir para a notificação'. To the right of these links are options for 'Acessibilidade', 'Alto Contraste', and 'Mapa do Site'. The main header features the title 'Portal da Transparência' in large white letters, with the subtitle 'MINISTÉRIO DA TRANSPARÊNCIA e CONTROLADORIA GERAL DA UNIÃO' below it. A secondary navigation bar contains links such as 'Sobre o Portal', 'Painéis', 'Consultas Detalhadas', 'Controle social', 'Recibo de Transparência', 'Receba Notificações', and 'Aprenda mais'. Below this is a search section titled 'Busque no Portal da Transparência' with a dropdown menu set to 'Todos' and a search input field with the placeholder text 'Busque por órgão, cidade, CNPJ, servidor...'. The main content area is divided into three horizontal panels. The left panel, titled 'ORÇAMENTO DA DESPESA EM 2018', shows a value of 'R\$ 3,46 TRILHÕES'. The middle panel, titled 'TOTAL DE PAGAMENTOS REALIZADOS EM 2018', shows a value of 'R\$ 2,49 TRILHÕES'. The right panel is titled 'CONHEÇA O PANORAMA DO GOVERNO FEDERAL'. Below these panels are three vertical sections. The first, 'Acesso rápido', lists links for 'Documentos básicos de execução da despesa pública', 'Execução mensal da despesa pública por: Órgão, Área de atuação, Ação/programa, orçamento/estado, Período', and 'Consulta de Pessoa Física'. The second, 'Localidade', includes a search box for 'Estado e Município' and a map of Brazil with state abbreviations. The third, 'Receitas e despesas', lists links for 'Orçamento anual da despesa', 'Orçamento anual da receita', 'Receitas públicas', 'Despesas públicas', 'Recursos transferidos', and 'Gastos com cartão de crédito'. The final section, 'Políticas públicas', lists links for 'Áreas de atuação (Funções)', 'Programas de governo', 'Benefícios sociais', 'Programas e ações orçamentárias', and 'Emendas parlamentares'. The Senac logo is visible in the bottom right corner.

Armazenamento

“A melhor maneira de prever o futuro é inventá-lo.”
Graham Bell

Armazenamento

- Dados estruturados



- Dados estruturados contém uma organização para serem recuperados.
- É o modelo de armazenamento de dados mais usado nos últimos 40 anos .
- Os dados são armazenados de acordo com
- Antes de armazenar alguma informação, é necessário definir a estrutura, a sequência, o tamanho e os tipos de dados em questão.

Armazenamento

- Dados estruturados

Uma característica desse modelo é o suporte à propriedade **ACID**, que garante a integridade dos dados por meio dos seguintes recursos:

Atomicidade: garante que todas as alterações realizadas por uma transação serão efetivadas no banco de dados, ou nenhuma delas, caso ocorra algum problema. Ou seja, não há atualização parcial da transação.

Consistência: nesse caso é garantido que novas transações somente serão completadas se elas não ferirem nenhuma regra do banco de dados que possa torná-lo inconsistente.

Isolamento: propriedade que permite que os eventos em uma transação não interfiram nos eventos de outra transação concorrente.

Durabilidade: garante que o resultado de toda transação executada com sucesso deverá ser mantido no banco de dados, mesmo na ocorrência de falhas.



Armazenamento

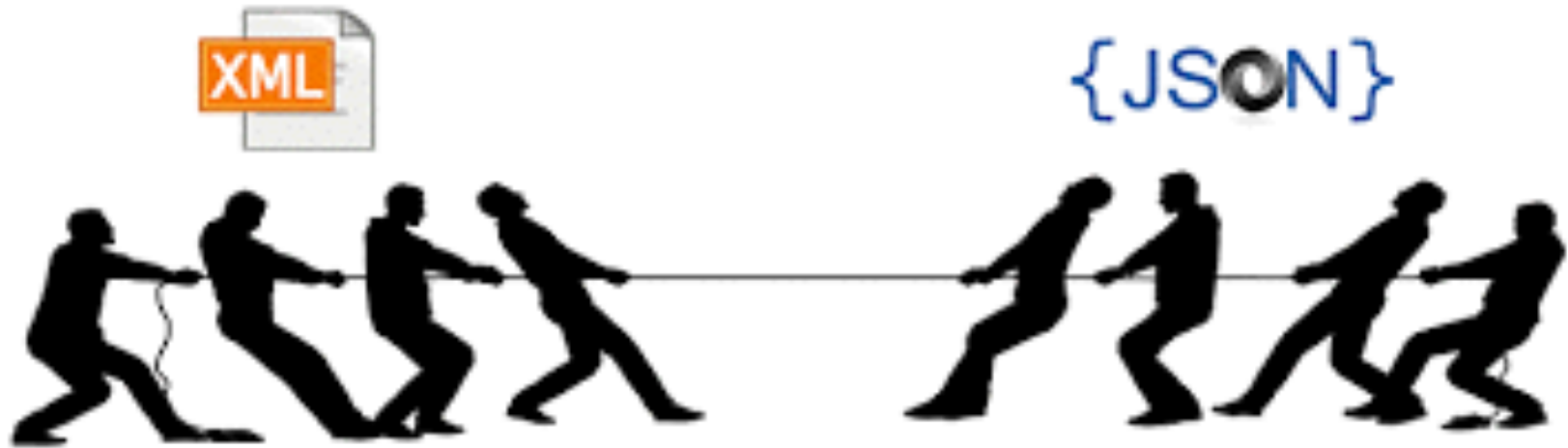
- Dados estruturados

Exercício 1 - Entendendo o modelo de banco de dados relacional

No Brasil, o governo disponibiliza muitas informações através do [portal da transparência](#). Neste site, pode-se baixar inúmeras informações sobre programas sociais, orçamentos, gastos entre outros dados. Iremos utilizar os dados do Bolsa Família para a criação de um banco de dados relacional. Neste [link](#), podemos obter o dicionário de dados das informações do bolsa família. Através dele, criaremos o nosso modelo entidade relacional. Os dados do bolsa família mês a mês podem ser baixados neste [link](#). [Link para um arquivo simplificado](#).

Armazenamento

- Dados semi estruturados



Dados semi estruturados são aqueles que possuem uma estrutura pré-definida. Essas estruturas são usadas normalmente apenas como um meio de marcação dos dados, como é o caso dos arquivos no formato **JSON** (JavaScript Object Notation) e **XML** (eXtensible Markup Language).

Armazenamento

- Dados semi estruturados - Json

{JSON}

```
{  
  "species": "Dog",  
  "breed": "Labrador Retriever",  
  "color": "Yellow",  
  "age": 6  
}
```



JSON *JavaScript Object Notation* é um formato de dados usado para armazenar informações, semelhante a um banco de dados. Ele consiste em pares nome - valor na forma de strings. Os pares nome - valor são separados por dois pontos e cada par é separado por uma vírgula. Muitas linguagens de programação podem gerar e ler o formato JSON. É muito popular para armazenar, analisar, ler e compartilhar informações.

Armazenamento

- Dados semi estruturados - Xml



```
<?xml version="1.0" encoding="UTF-8"?>
<dog>
  <species>Dog</species>
  <breed>Labrador Retriever</breed>
  <color>Yellow</color>
  <age>6</age>
</dog>
```



XML, *Extensible Markup Language* é um formato de texto simples e flexível. Originalmente projetada para enfrentar os desafios da publicação eletrônica em grande escala, a XML desempenha um papel importante na troca de uma ampla variedade de dados.

Armazenamento

- Dados semi estruturados

Exercício 2 - Entendendo o modelo de dados semi estruturado usando Json e Xml

Use os mesmos dados do bolsa família e crie uma lista de beneficiários. Utilize o [JsonLint](#) para validar os dados em Json e o [Xml Formatter](#) para validar o arquivo em XML.

Enumere alguns prós e contras sobre as tecnologias Json e XML

Armazenamento

- Banco de dados NoSql



“Bancos de dados NoSQL são criados para modelos de dados específicos e têm esquemas flexíveis para a criação de aplicativos modernos. Os bancos de dados NoSQL são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala. Eles usam vários modelos de dados, incluindo documento, gráfico, chave-valor, memória e pesquisa. Esta página inclui recursos para ajudar você a compreender melhor os bancos de dados NoSQL e a começar a usá-lo.”

<https://aws.amazon.com/pt/nosql/>

Armazenamento

- Banco de dados NoSql

Document 1	Document 2	Document 3
<pre>{ "id": "1", "name": "John Smith", "isActive": true, "dob": "1964-30-08" }</pre>	<pre>{ "id": "2", "fullName": "Sarah Jones", "isActive": false, "dob": "2002-02-18" }</pre>	<pre>{ "id": "3", "fullName": { "first": "Adam", "last": "Stark" }, "isActive": true, "dob": "2015-04-19" }</pre>

Os bancos de dados NoSQL usam diversos modelos para acessar e gerenciar dados, como documento, gráfico, chave-valor, em memória e pesquisa. Esses tipos de banco de dados são otimizados especificamente para aplicativos que exigem modelos de grande volume de dados, baixa latência e flexibilidade. Esses requisitos são atendidos mediante o relaxamento de algumas restrições de consistência de dados dos outros bancos.

Armazenamento

- Banco de dados NoSql

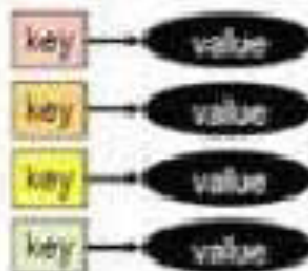
- **Flexibilidade:** os bancos de dados NoSQL geralmente fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e interativo. O modelo de dados flexível torna os bancos de dados NoSQL ideais para dados semi estruturados e não estruturados.
- **Escalabilidade:** os bancos de dados NoSQL 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 (como documento, chave-valor e gráfico) 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.

Armazenamento

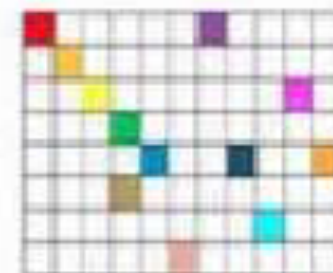
- Banco de dados NoSql

- Chave-valor
 - Documento.
 - Gráfico
 - Em memória
- Pesquisar

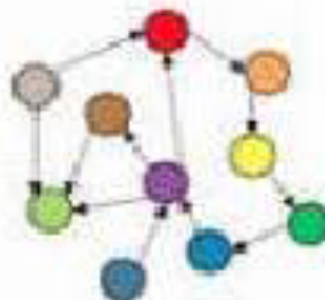
Key-Value



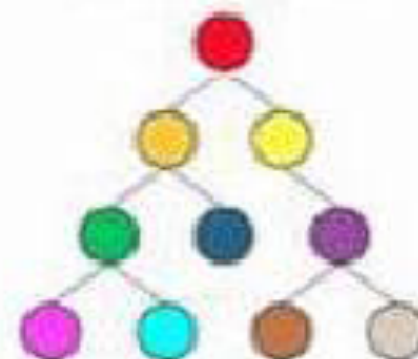
Column-Family



Graph



Document

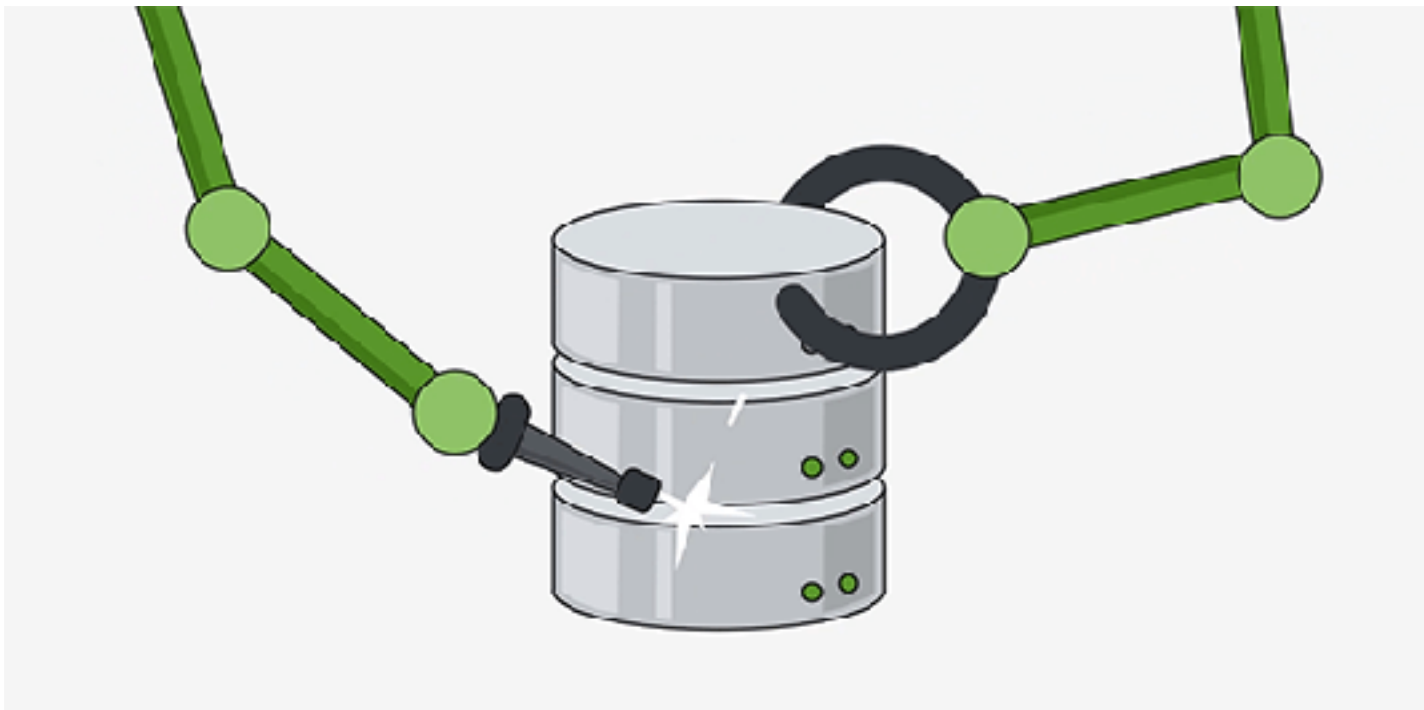


MongoDb



mongoDB

MongoDB



MongoDB é um banco de dados NoSql orientado a documentos que armazena as informações em um único documento sem esquemas relacionais.

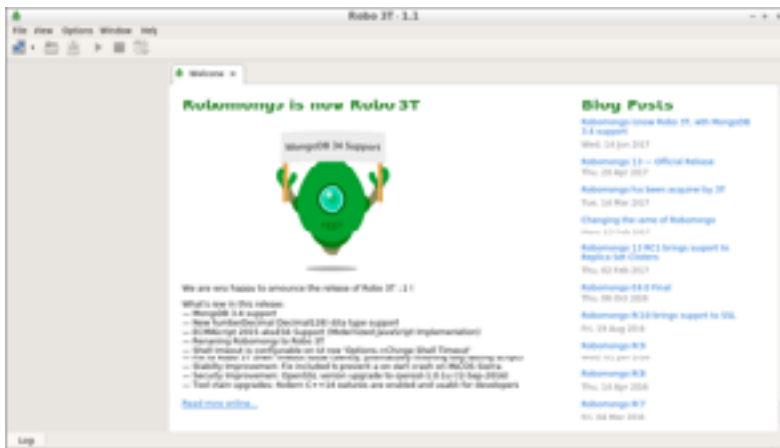
MongoDB - configuração

The screenshot displays the MongoDB Atlas website interface. At the top, there is a navigation bar with links for 'DOCS', 'LEARN', 'WHAT'S MONGODB?', 'BLOG', and 'LOGIN'. A green 'Get MongoDB' button is located in the top right corner. Below the navigation bar, the MongoDB logo and the tagline 'FOR GIANT IDEAS' are visible on the left, while 'SOLUTIONS', 'CLOUD', 'CUSTOMERS', 'RESOURCES', and 'ABOUT US' are listed on the right. The main content area features the heading 'MongoDB Atlas' and a descriptive paragraph: 'Move faster with an automated cloud MongoDB service built for agile teams who'd rather spend their time building apps than managing databases. Available on AWS, Azure, and GCP.' Below this text is a green 'Start free' button and a link for existing users: 'Already have an account? Log in here →'. A large, semi-transparent overlay window titled 'Cloud Provider & Region' is positioned on the right side of the page. This window guides the user through selecting a cloud provider (AWS, Google Cloud, or Azure) and then choosing a specific region. It lists various global regions such as N. Virginia, London, and Tokyo, with some marked as 'recommended' or 'not available'. At the bottom of the overlay, there is a section for 'Configure cross-region replication' with a 'Done' button.

MongoDB - client

Robo 3T

MongoDb Compass



MongoDB - comandos básicos

MongoDB - comandos básicos

Inserindo documentos

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,             ← field: value
  status: "pending"   ← field: value
}                      } document
)
```

[insertMany\(\)](#) retorna um documento que inclui os valores dos campos `_id` dos documentos recém-inseridos. Use [db.collection.insertOne\(\)](#) para inserir um único documento.

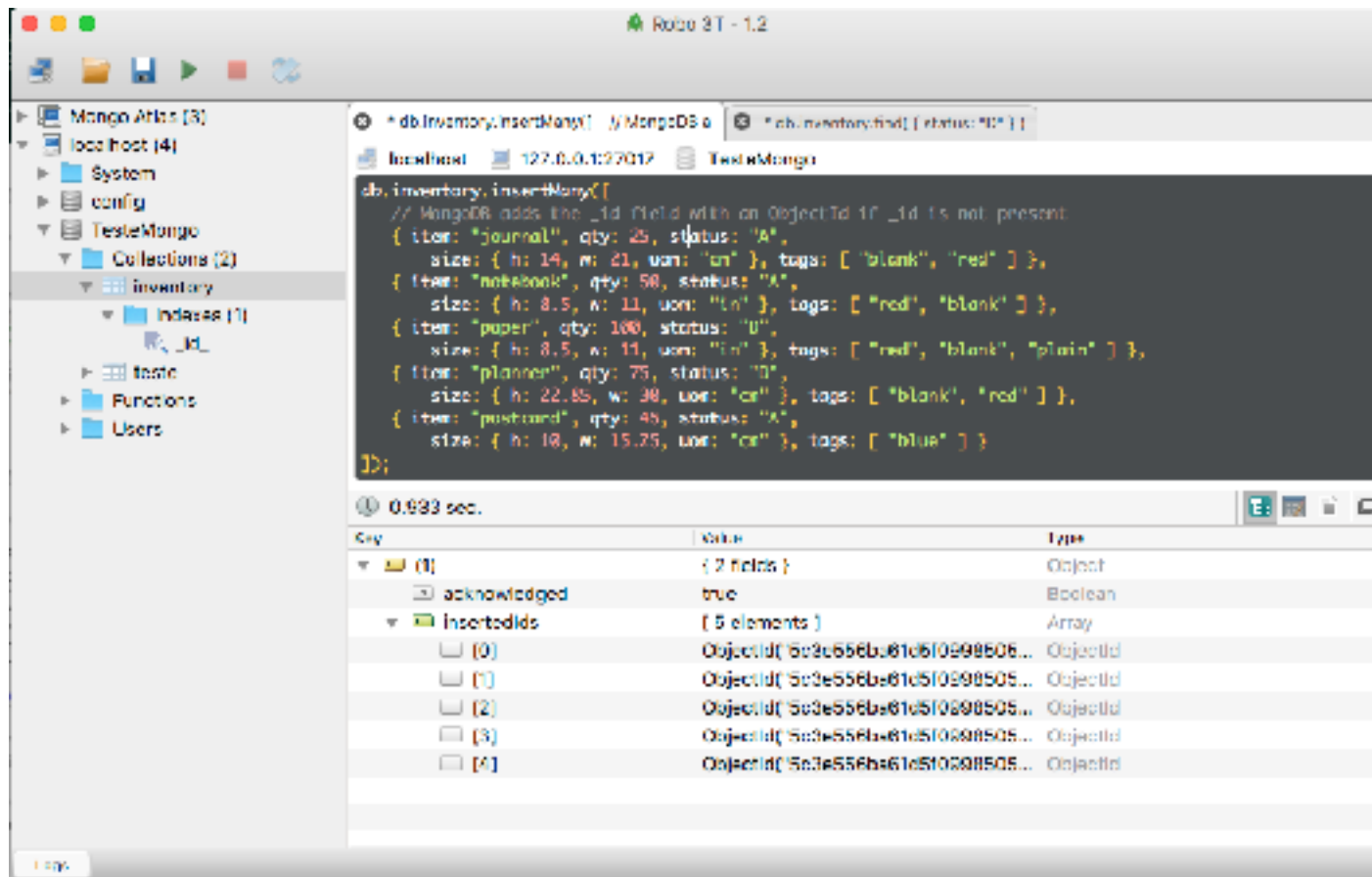
MongoDB - comandos básicos

Inserindo documentos

```
db.inventory.insertMany([
  // MongoDB adds the _id field with an ObjectId if _id is not present
  { item: "journal", qty: 25, status: "A",
    size: { h: 14, w: 21, uom: "cm" }, tags: [ "blank", "red" ] },
  { item: "notebook", qty: 50, status: "A",
    size: { h: 8.5, w: 11, uom: "in" }, tags: [ "red", "blank" ] },
  { item: "paper", qty: 100, status: "D",
    size: { h: 8.5, w: 11, uom: "in" }, tags: [ "red", "blank", "plain" ] },
  { item: "planner", qty: 75, status: "D",
    size: { h: 22.85, w: 30, uom: "cm" }, tags: [ "blank", "red" ] },
  { item: "postcard", qty: 45, status: "A",
    size: { h: 10, w: 15.25, uom: "cm" }, tags: [ "blue" ] }
]);
```

MongoDB - comandos básicos

No MongoDB as informações são inseridas dentro de collections. Caso ela não exista o MongoDB cria automaticamente.



The screenshot shows the Robo 3T - 1.2 interface. On the left, the 'Mongo Atlas (3)' tree shows the 'inventory' collection under 'TesteMongo'. The main window displays the command `db.inventory.insertMany()` and its execution details. The command is executed on the 'localhost' connection at '127.0.0.1:27017'. The execution time is 0.033 sec. The result is a table with columns 'Key', 'Value', and 'Type'.

Key	Value	Type
acknowledged	true	Boolean
insertedIds	[5 elements]	Array
[0]	ObjectId('5c3e556b61d510990505...')	ObjectId
[1]	ObjectId('5c3e556b61d510990505...')	ObjectId
[2]	ObjectId('5c3e556b61d510990505...')	ObjectId
[3]	ObjectId('5c3e556b61d510990505...')	ObjectId
[4]	ObjectId('5c3e556b61d510990505...')	ObjectId

MongoDB - comandos básicos

Pesquisando documentos

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

MongoDB - comandos básicos

Pesquisando documentos

```
db.inventory.find( {} )
```

```
db.inventory.find( { status: "D" } )
```

```
db.inventory.find( { size: { h: 14, w: 21, uom: "cm" } } )
```

```
db.inventory.find( { "size.uom": "in" } )
```

Procurando elementos na matriz

```
db.inventory.find( { tags: "red" } )
```

```
db.inventory.find( { tags: ["red", "blank"] } )
```

MongoDB - comandos básicos

Atualizando documentos

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

← collection
← update filter
← update action

MongoDB - comandos básicos

Atualizando documentos

```
db.inventory.update(  
  { item: "paper" },  
  { $set: { qty: 11} }  
)
```

```
db.inventory.update(  
  { tags: "red" },  
  { $set: { qty: 200} }  
)
```



MongoDB - comandos básicos

Apagando documentos

```
db.users.deleteMany(  
  { status: "reject" }  
)
```



collection



delete filter

MongoDB - comandos básicos

Apagando documentos

```
db.inventory.deleteMany({})  
db.inventory.deleteMany({ status : "A" })
```

Deleta apenas um documento que combine com a pesquisa

```
db.inventory.deleteOne( { status: "D" } )
```


MongoDB - pesquisa avançada

MongoDB - pesquisa avançada

Operações de comparação

\$gt : maior do que

\$gte : igual ou maior do que

\$lt : menor do que

\$lte : igual ou menor do que

```
db.inventory.find(  
  { "qty" : {$gt: 50} }  
)
```

Operador in que busca valores absolutos

```
db.inventory.find(  
  { "qty" : {$in: [100, 50]} }  
)
```

MongoDB - pesquisa avançada

Operador not equal (**ne**)

Traz o resultado oposto ao critério especificado

```
db.inventory.find(  
  { "qty" : {$ne: 50} }  
)
```

Operador **Distinct** que elimina as repetições do resultado de uma consulta

```
db.inventory.distinct( "qty" )
```

MongoDB - pesquisa avançada

Operadores lógicos - **and**, **nor**, **not**

Traz o resultado oposto ao critério especificado

```
db.inventory.find( {$or:  
  [  
    {"qty" : {$eq: 50}},  
    {"qty" : {$eq: 100}}  
  ]  
})
```

Operadores estilo Like - O operador like é muito comum nas bases relacionais e permite fazer buscas por trechos de texto nas tabelas.

```
db.inventory.find( {"item": /te/} )
```

MongoDB - pesquisa avançada

Para a busca sem considerar maiúsculas ou minúsculas, é preciso colocar i (de case Insensitive):

```
db.inventory.find( {"item": /TE/i} )
```

Para buscar por palavras que terminam com um trecho de caracteres, é preciso colocar \$

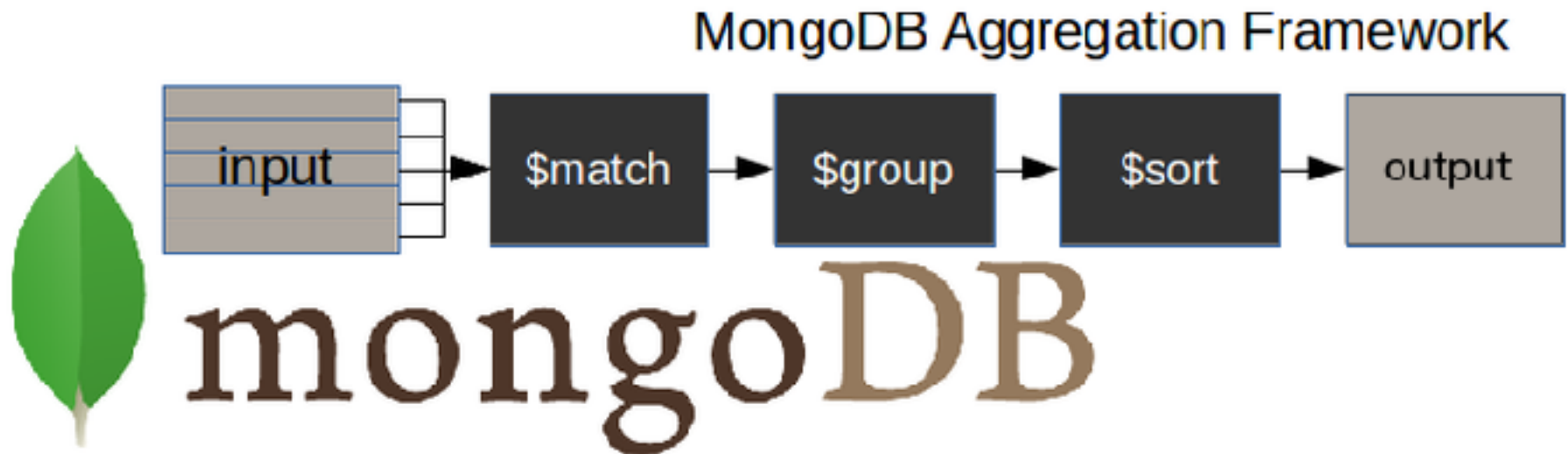
```
db.inventory.find( {"item": /ok$/} )
```

Para buscar por palavras que se iniciam com um trecho de caracteres, é preciso colocar ^

```
db.inventory.find( {"item": /^no/} )
```

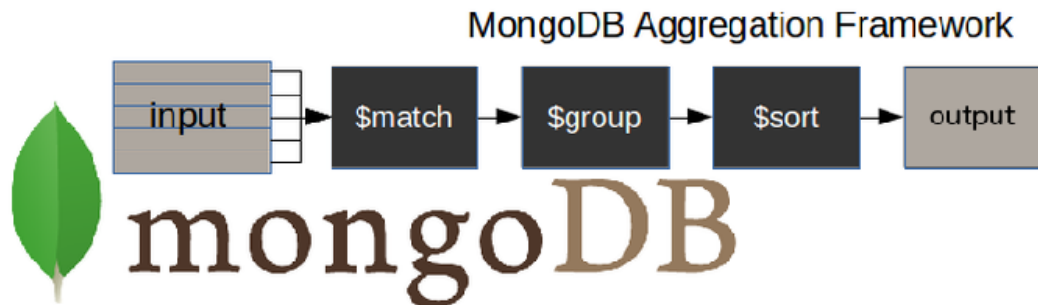
MongoDB - aggregation framework

MongoDB - aggregation framework



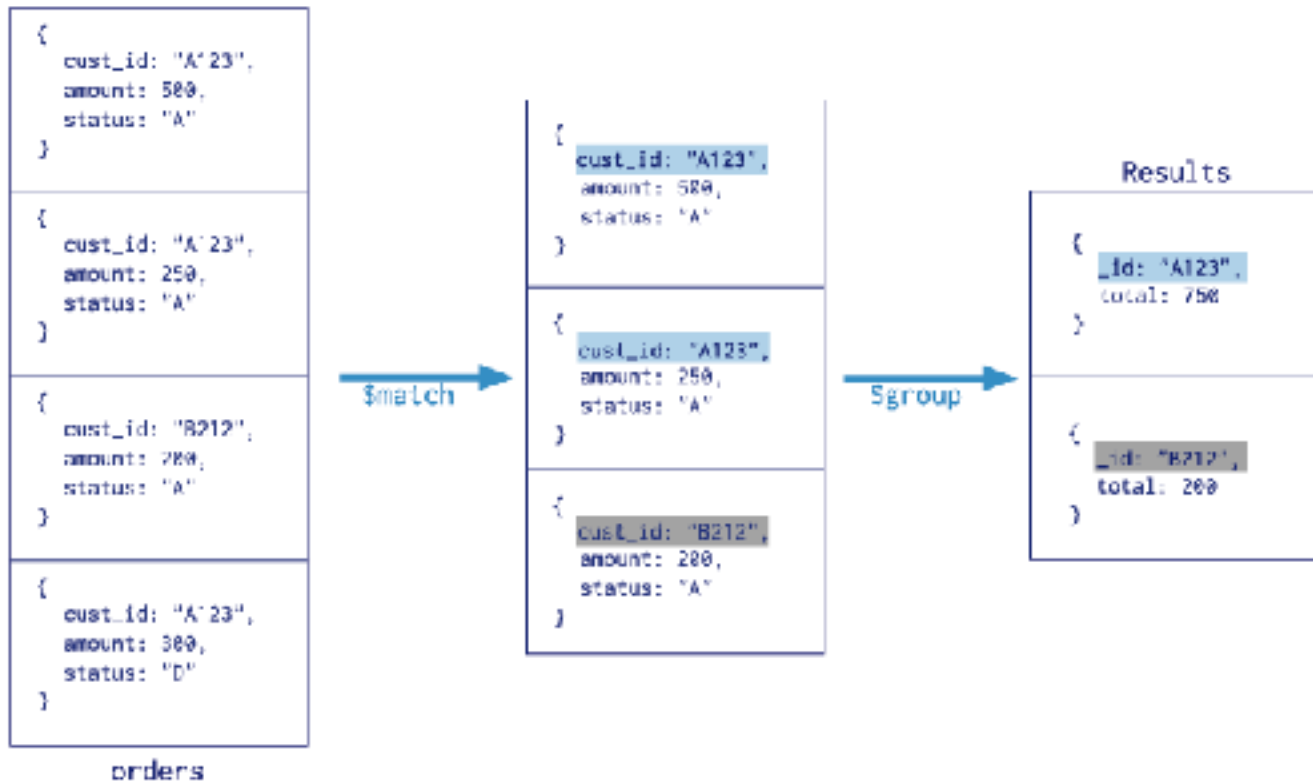
MongoDB - aggregation framework

As operações de agregação processam registros de dados e retornam resultados agrupados. Agregação de valores agrupa operações de vários documentos juntos e pode executar uma variedade de operações retornar um único resultado. A grosso modo, é como se fosse um **GroupBy** de um banco relacional mas muito mais performático



MongoDB - aggregation framework

Collection
↓
`db.orders.aggregate([`
 \$match stage → `{ $match: { status: "A" } },`
 \$group stage → `{ $group: { _id: "$cust_id", total: { $sum: "$amount" } } }`
 `]`)



MongoDB - aggregation framework

Soma os valores totais do campo valor

```
db.bolsafamilia.aggregate(  
  [  
    { $group:  
      {  
        id: null,  
        total: { $sum: "$VALOR" },  
        count: { $sum: 1 }  
      }  
    ]  
  )
```

Soma os valores totais do campo valor pelo município
de Paulo Afonso

```
db.bolsafamilia.aggregate(  
  [  
    { $match: { MUNICIPIO: "PAULO AFONSO" } },  
    { $group:  
      {  
        id: "$MUNICIPIO",  
        total: { $sum: "$VALOR" },  
        count: { $sum: 1 }  
      }  
    ]  
  )
```

MongoDB - aggregation framework

Soma os valores totais do campo valor agrupados por município

```
db.bolsafamilia.aggregate(  
  [  
    { $group:  
      {  
        id: "$MUNICIPIO",  
        total: { $sum: "$VALOR" },  
        count: { $sum: 1 }  
      }  
    ]  
  )
```

Soma os valores totais do campo valor agrupados por município e ordenado pelo total

```
db.bolsafamilia.aggregate(  
  [  
    { $group:  
      {  
        id: "$MUNICIPIO",  
        total: { $sum: "$VALOR" },  
        count: { $sum: 1 }  
      }  
    }, { $sort: { total: 1 } }  
  ]  
)
```

MongoDB - aggregation framework

Executar operação de classificação utilizando o disco como cache

```
db.bolsafamilia.aggregate(  
  [  
    { $group:  
      {  
        _id: "$MUNICIPIO",  
        total: { $sum: "$VALOR" },  
        count: { $sum: 1 }  
      }  
    }, { $sort: { total: 1 } }  
  ],  
  {  
    allowDiskUse: true  
  }  
)
```

MongoDB - performance

MongoDB - performance



O **MongoDB** como qualquer outro banco de dados, pode, com o aumento dos dados, sofrer com a performance.

Para monitorar e resolver este problema, ele possui algumas ferramentas úteis que serão vistas neste tópico.

MongoDB - performance

O **MongoDB** oferece o método **stats** que mostra informações sobre as **collects**.

```
db.bolsafamilia.stats()
```

```
{
  "ns" : "governo.bolsafamilia",
  "size" : 24008997,
  "count" : 121018,
  "avgObjSize" : 198,
  "storageSize" : 7315456,
  "capped" : false,
  "nindexes" : 1,
  "totalIndexSize" : 1753088,
  "indexSizes" : {
    "_id_" : 1130496
  },
  "ok" : 1.0,
  "operationTime" : Timestamp(1547680911, 1)
}
```

→ Total registro da collect

→ Quantidade de índices criados

→ Nome e tamanho dos índices

MongoDB - performance

Usando o comando **explain**, pode-se extrair informações importantes de consultas

```
db.bolsafamilia.find(  
  {"MUNICIPIO" : "BAURU"}  
) .explain( 'executionStats' )  
{
```

O parâmetro **executionStats**
ajuda a extrair tópicos interessantes

```
  "winningPlan" : {  
    "stage" : "COLLSCAN"  
  }
```

A consulta percorreu
toda a tabela

```
},
```

```
"executionStats" : {
```

```
  "nReturned" : 12,
```

Quantidade de dados retornados

```
  "executionTimeMillis" : 67,
```

Tempo da pesquisa

```
  "totalKeysExamined" : 0,
```

```
  "totalDocsExamined" : 121018,
```

Total documentos
analisados para trazer
a pesquisa

MongoDB - performance

```
{
  "winningPlan" : {
    "stage" : "COLLSCAN"
  },
  "executionStats" : {
    "nReturned" : 12,
    "executionTimeMillis" : 67,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 121018,
```

→ A consulta percorreu toda a tabela

→ Quantidade de dados retornados

→ Tempo da pesquisa

→ Total documentos analisados para trazer a pesquisa

Examinado as informações, percebe-se que foi percorrida toda a colect e foram lidos todos os registros para a pesquisa, selecionando apenas 12 deles

MongoDB - performance

Para resolver este problema, criaremos índices nos campos a serem pesquisados

```
db.bolsafamilia.ensureIndex( { "MUNICIPIO" : 1 } )
```

```
{
  "winningPlan" : {
    "stage" : "IXSCAN"
  },
  "executionStats" : {
    "nReturned" : 12,
    "executionTimeMillis" : 0,
    "totalKeysExamined" : 12,
    "totalDocsExamined" : 12,
```

→ Indica que o índice foi utilizado

→ Quantidade de dados retornados

→ Tempo da pesquisa

→ Total de documentos analisados para trazer a pesquisa

MongoDB - performance

Índices textual

A busca textual no MongoDB é utilizada para fazer pesquisa em trechos de texto, bem como, auxilia na aproximação dos resultados através da aproximação de resultados relevantes para determinada chave de procura.

```
db.bolsafamilia.ensureIndex(  
  {"NOME" : "text"},  
  {default_language: "portuguese"}  
)
```

Nome do campo a ser indexado

Idioma que o índice foi criado.
Se omitido, é usada o idioma inglês

MongoDB - performance

Índices textual

Fazendo uma pesquisa textual

```
db.bolsafamilia.find( { $text: { $search: "maria zen" } } )
```

O índice textual permite excluir resultado da pesquisa, por exemplo

```
db.textos.ensureIndex( {texto: "text"},  
    {default_language: "portuguese"} )
```

```
db.textos.insert({texto: "Eu gosto de São Paulo"})
```

```
db.textos.insert({texto: "Eu gosto de São Paulo e Rio Claro"})
```

```
db.textos.find( { $text: { $search: "gostar -claro" } } )
```

MongoDB - performance

Listar índices

```
db.bolsafamilia.getIndexes()
```

Remover índices

```
db.bolsafamilia.dropIndex("MUNICIPIO_1")
```

MongoDB - importando dados

```
mongoimport --host=127.0.0.1 -d <nome banco> -c <nome collect> --  
type csv --file <caminho arquivo> --headerline
```

Armazenamento

Exercício 3 - Análise de Base de Dados

Escolha uma base de dados do governo e faça uma análise dos seus dados utilizando as ferramentas vistas em MongoDB

<https://www.kaggle.com/datasets>

<http://www.portaltransparencia.gov.br/>



mongoDB

Obrigado

