

TRABALHANDO COM NOSQL – MONGODB

Unidade 02



Sumário

03	Introdução
04	Objetivos de Aprendizagem
05	2.1 Bancos de dados NoSQL
06	2.1.1 Ausência de esquema
07	2.1.2 Agregação dos dados
07	2.1.3 Armazenamento por documentos
11	2.2 MongoDB
11	2.2.1 Instalando o banco de dados
16	2.2.2 Fazendo o primeiro acesso ao MongoDB
17	2.3 Explorando o MongoDB com o MongoDB Compass
20	2.4 Alguns exemplos de aplicações para o MongoDB
21	2.4.2 Análise em tempo real
23	Síntese
24	Fullture Insight

Olá, Fullturist,

Seja bem-vindo(a)!

BANCOS DE DADOS: DEFINIÇÕES

Iremos acompanhá-lo(a) no desenvolvimento desta trilha de aprendizagem, que trará informações a respeito da tecnologia e do ambiente de negócios em bancos de dados.

Pode ser surpreendente notar que este é um tema muito importante em nossa vida atualmente. Praticamente todos os eventos de nossa vida passada, e até mesmo de nossas tendências futuras, estão registrados em um banco de dados. Desde informações de nossa vida civil, como documentos pessoais e registros de propriedades, nossa vida pessoal, por meio das redes sociais, e nosso trajeto ao trabalho são registrados em um banco de dados.

Como você pode perceber, assim como esse tema pode ser vasto em nossa vida cotidiana, a tecnologia oferecida para atender essas diversas situações também é ampla. Entretanto, embora as opções tecnológicas sejam numerosas, esta trilha fornecerá a você todos os conhecimentos para entender o ambiente tecnológico e de negócios disponíveis para a solução da maioria das necessidades. Além disso, conheceremos o que há por trás de toda essa tecnologia e como todas essas informações são armazenadas e recuperadas! Para isso, utilizaremos um dos bancos de dados mais promissores do mercado atualmente: o MongoDB. Vamos aprender a instalar, inserir, recuperar e remover dados, bem como organizar e executar um projeto completo neste banco de dados.

Bom estudo!

Unidade 2

Olá, Fullturist!

Esta é a unidade 2 da trilha de aprendizagem Banco de Dados. Aqui, você terá a oportunidade de conhecer e/ou se aprofundar em bancos de dados NoSQL e as questões que estes bancos de dados procuram solucionar. Durante o aprendizado, além de conhecer melhor suas principais características, você também entenderá como os dados são trabalhados por estes bancos de dados. Para a fixação dos conceitos, utilizaremos o software MongoDB, executando, de forma prática e seguindo um passo a passo, sua instalação e a inserção e visualização dos dados inseridos por você neste banco de dados.

Vamos lá?

Objetivos de aprendizagem

Ao final do estudo desta unidade, você será capaz de:

- conhecer o contexto em que os bancos de dados NoSQL surgiram e as demandas que estes bancos de dados procuram solucionar;
- entender a razão pela qual as características mais marcantes dos bancos de dados NoSQL são a ausência de esquema e a alta escalabilidade;
- conhecer como a agregação prévia de dados, que futuramente serão manipulados juntos, contribui para a simplicidade no desenvolvimento e para o desempenho no processamento de altos volumes de dados;
- construir um documento padrão JSON, compreendendo que se trata da unidade de armazenamento em bancos de dados NoSQL que utilizam este modelo de armazenamento;
- instalar o banco de dados NoSQL MongoDB e a ferramenta MongoDB Compass, a qual facilita a interação com este banco de dados;
- utilizar a ferramenta MongoDB Compass para a inserção e a visualização de um documento no banco de dados;
- conhecer alguns exemplos de aplicações que se beneficiam da tecnologia de armazenamento em documentos.

2.1 Bancos de dados NoSQL

Um ponto fundamental para entender bancos de dados NoSQL é conhecer o contexto em que estes bancos de dados surgiram. Para isso, vamos fazer uma comparação bastante simples para compreender como o processamento de dados evoluiu ao longo do tempo.

“Antes”, tínhamos sistemas corporativos, como os sistemas bancários, comerciais e industriais. Estes sistemas são fontes de dados bastante consolidadas e trabalham com um tipo de dado muito padronizado. As fontes de dados para estes sistemas são previamente conhecidas e seu processamento, tanto em estrutura como em volume, pode ser previsto com muita confiabilidade.

“Agora”, além destes dados já tradicionais, a “nuvem” também é uma fonte importante de dados e informações. Sabemos que o ritmo das inovações é alucinante e que a partir da nuvem podem chegar, além dos tipos tradicionais já conhecidos, também outros tipos de dados, e ainda outros, que possam surgir em uma inovação repentina.

Então, “agora” temos uma relativa imprevisibilidade no processamento dos dados: em relação à fonte – uma fonte de dados adicional pode surgir a qualquer momento; em relação a como estes dados estarão estruturados – uma nova fonte pode conter novos dados; e, principalmente, em relação ao volume que nossa aplicação deverá manipular – os volumes de dados podem crescer repentinamente.

Foi neste contexto do “agora” – de nosso momento atual em processamento de dados – que os bancos de dados NoSQL surgiram. Estes bancos de dados foram projetados para proporcionarem maior flexibilidade para que as aplicações possam lidar com esta relativa elasticidade das fontes e dos volumes.

Já vimos as características dos bancos de dados NoSQL na unidade 1, entretanto, com este cenário em mente, vamos refletir novamente sobre como os bancos de dados NoSQL endereçam particularmente estas questões:

alta escalabilidade: esta é a característica chave para lidar com crescentes volumes de dados. É a habilidade nativa que bancos de dados NoSQL têm para executarem em clusters de máquinas, de forma elástica, conforme a necessidade da aplicação. Nestes bancos de dados, adicionar outras máquinas, ou seja, escalar de forma horizontal o cluster, não implica em maior complexidade. Além disso, estes bancos de dados também são flexíveis em relação à diversidade hardware, diminuindo em muito os custos de processamento;

ausência de esquema: flexibilidade para a adição de várias fontes ou novos dados à aplicação;

código aberto (open source): não exige a aquisição de licenças. Quando necessitamos de escalabilidade, isso reduz, senão elimina, os custos com aquisição de licenças. Outra vantagem é que, sendo open source, o desenvolvedor ainda pode fazer modificações no software conforme desejar ou necessitar. Também vale observar que alguns fornecedores de bancos de dados NoSQL também oferecem versões pagas de seus produtos. Neste caso, os fornecedores oferecem um suporte mais especializado ao produto.

Como você pode perceber, as características mais marcantes em bancos de dados NoSQL são a alta escalabilidade e a ausência de esquema. No próximo tópico, veremos como o processamento dos dados pode ocorrer na ausência de um esquema de dados previamente definido. Vamos lá?

2.1.1 Ausência de esquema

É importante termos em mente que os dados por si só não resultam em informação. A informação emerge quando estes dados são associados e processados para responder a um determinado questionamento. Por exemplo, nome, idade, profissão, formação acadêmica e histórico profissional só representam verdadeiramente uma informação quando estes dados são associados à uma “pessoa”. Ou seja, estes dados somente representarão uma informação quando armazenados de uma forma que, ao processá-los, seja possível associá-los à uma pessoa. Assim, por exemplo, quando o questionamento for “qual é o curriculum de uma determinada pessoa”, estes dados serão recuperados conforme sua associação com cada pessoa cadastrada em um banco de dados de Curriculum. Mas, então, como bancos de dados NoSql fazem esta associação?

Como já vimos na unidade 1, a maioria dos bancos de dados NoSQL são conceitualmente divididos em 4 categorias, conforme o modelo de armazenamento do banco. Vamos relembrar estas categorias.

Documento: os dados são armazenados por meio de documentos autoexplicativos. O documento, em si, já é uma associação de dados, e autocontém a estrutura e o significado destes dados.

Chave-valor: a associação entre os dados é feita por meio de pares chave-valor. Por meio da chave, todos os valores (dados) são relacionados.

Família de colunas: a chave é projetada para associar atributos (dados isolados) e colunas múltiplas. Estas colunas múltiplas podem representar uma nova estrutura de dados, aninhada e associada à mesma chave.

Grafos: os dados são armazenados conceitualmente como grafos com nós relacionados, e a leitura percorre suas arestas por meio de expressões que determinam o caminho a seguir.

Refletindo sobre estas formas de armazenamento, percebemos que as três primeiras formas (documento, chave-valor e família de colunas) trabalham com o conceito de uma prévia agregação dos dados que são frequentemente acessados juntos. No próximo tópico, veremos o que isso significa e como a forma com que os dados são agregados interfere no desempenho da aplicação.

2.1.2 Agregação dos dados

O conceito de agregação é extremamente ligado à alta escalabilidade e ao processamento em cluster. Cada unidade de agregação (documento, chave-valor, família de colunas) estará intimamente ligada à replicação e à fragmentação dos dados em cada nó do cluster. Dados que serão frequentemente recuperados ou manipulados juntos, em prol do bom desempenho do sistema, deverão ser armazenados no mesmo cluster. Por isso, é de extrema importância que o modelo de armazenamento a ser escolhido (documento, chave-valor ou famílias de colunas) leve em consideração a forma como os dados serão futuramente recuperados.

Nesta trilha, e ainda nesta unidade, para a demonstração prática destes conceitos, vamos utilizar o banco de dados MongoDB. Este banco de dados utiliza o modelo de armazenamento por documentos. Por esta razão, no próximo tópico vamos explorar melhor o conceito de agregação utilizando este modelo de armazenamento de dados.

2.1.3 Armazenamento por documentos

Neste modelo, os dados são armazenados na forma de documentos autoexplicativos, isto é, o documento já define a estrutura e o significado dos dados que ele contém. O MongoDB utiliza o padrão de documento JSON (Java Script Object Notation). Este formato é livre de esquema e aceita quase qualquer coisa que você quiser armazenar. O documento pode conter vários dados e cada um deles deve ser identificado por um nome, seguido de seu conteúdo. A Figura 01, já apresentada na unidade 1, exemplifica este padrão.

```
{
  "id": 21,
  "Pais": "Brasil",
  "Estado": "São Paulo",
  "Cidade": "Campinas",
  "Nome": "Rosangela Bittencourt",
  "Documentos": [
    {
      "RG": 14584942,
      "CPF": "58491785476",
    }
  ]
}
```

Fonte: Elaborada pelo autor (2021).

Observando esta estrutura de armazenamento, percebemos que este padrão atende perfeitamente à característica de ausência de esquema dos bancos NoSQL. A construção do documento é livre e de forma muito simples atende a necessidade de flexibilidade para a utilização de várias fontes de dados pela aplicação.

Para que você visualize o quanto este modelo de armazenamento é simples, vamos fazer uma comparação com o modelo de armazenamento em bancos de dados relacionais. A Figura 02, também já apresentada na unidade 1, demonstra graficamente o modo de armazenamento relacional e as associações necessárias para a produção da informação de um pedido de venda.



A informação do Pedido é produzida através da associação com o Cliente que efetuou a compra e com o produto comprado. Agora, vejamos como este mesmo modelo de armazenamento é feito a partir de documentos JSON.

```
{ "Pedido_id": 16879,
  "Cliente":
  {
    "Cliente_id": 45679
    "Cliente_nome": "Pedro Soares",
    "Cliente_rg": 18673920,
    "Cliente_cpf": 47892307643,
    "Cliente_endereço": "Av Primavera, 98 CEP 17590432 São Paulo"
  },
  "Produtos":
  [
    {
      "Produto_ID": 6689
      "Produto_descricao": "Adidas Advantage Base",
      "Produto_cor": "Preto",
      "Produto_tamanho": "43"
      "Produto_preco": 299
    }
    {
      "Produto_id": 7239
      "Produto_descricao": "Nike SB Check Solar",
      "Produto_cor": "Chumbo",
      "Produto_tamanho": "39",
      "Produto_preco": 399
    }
  ]
}
```

Fonte: Elaborada pelo autor (2021).

Percebemos que o modelo por documentos é realmente mais simples. No modelo relacional, os dados são armazenados em três tabelas – Pedido, Cliente e Produto. Para obter a informação completa a respeito de um Pedido, o desenvolvedor terá que codificar a associação entre as três tabelas. No modelo por documentos, todos os dados referentes à entidade Pedido são armazenados juntos, em um único documento.

Entretanto, isso não significa que um modelo de organização dos dados seja melhor ou pior que o outro. Em aplicações intensivas em I/O e com alta concorrência pelos dados, o modelo relacional controla com muita eficiência a redundância e a consistência das informações. Porém, nem todas as aplicações necessitam destas características. Aplicações analíticas ou de baixa latência, com altos volumes de dados são mais bem atendidas por bancos de dados NoSQL. O sucesso na construção de uma aplicação depende primordialmente da escolha da ferramenta mais adequada.

Em relação à construção do documento, a sintaxe é bastante simples. Observe a Figura 04. Em cada linha, há um par “nome do campo”: valor. Agora, observe atentamente a posição das vírgulas, dos parênteses e das chaves. Se ainda houver outros campos a definir, a linha deve ser encerrada com uma vírgula. Além disso, os parênteses e as chaves são símbolos para agregação de dados. Os parênteses são usados para um agrupamento lógico de um objeto, como dados de uma entidade (Cliente, Produto). As chaves são usadas para a definição de um array de dados ou de objetos. No exemplo, o primeiro conjunto de dados agrupados são os dados do objeto Cliente. Como há a possibilidade de venda de vários produtos em um mesmo pedido, os dados de cada produto são agrupados em um array, e cada elemento do array contém os dados de um produto vendido.

```

{
  "Pedido_id": 16879,
  "Cliente": {
    "Cliente_id": 45679,
    "Cliente_nome": "Pedro Soares",
    "Cliente_rg": 18673920,
    "Cliente_cpf": 47892307643,
    "Cliente_endereço": "Av Primavera, 98 CEP 17590432 São Paulo"
  },
  "Produtos": [
    {
      "Produto_ID": 6689,
      "Produto_descrição": "Adidas Advantage Base",
      "Produto_cor": "Preto",
      "Produto_tamanho": "43",
      "Produto_preço": 299
    },
    {
      "Produto_ID": 7239,
      "Produto_descrição": "Nike SB Check Solar",
      "Produto_cor": "Chumbo",
      "Produto_tamanho": "39",
      "Produto_preço": 399
    }
  ]
}
    
```

Objetos ou dados

Array de objetos ou dados

Fonte: Elaborada pelo autor (2021).

Neste ponto da unidade, você já possui os conhecimentos necessários para colocar todos os conceitos apresentados em prática. Então, que tal iniciar instalando e explorando o MongoDB?

2.2 MongoDB

O MongoDB é um banco de dados open source com distribuição gratuita para Windows, Linux e Mac. Possui alta performance e diversas linguagens possuem drivers para este banco de dados, entre elas destacam-se: C, C#, C++, Java, JavaScript, Perl, PHP e Python



Fonte: MONGODB, on-line, 2021.

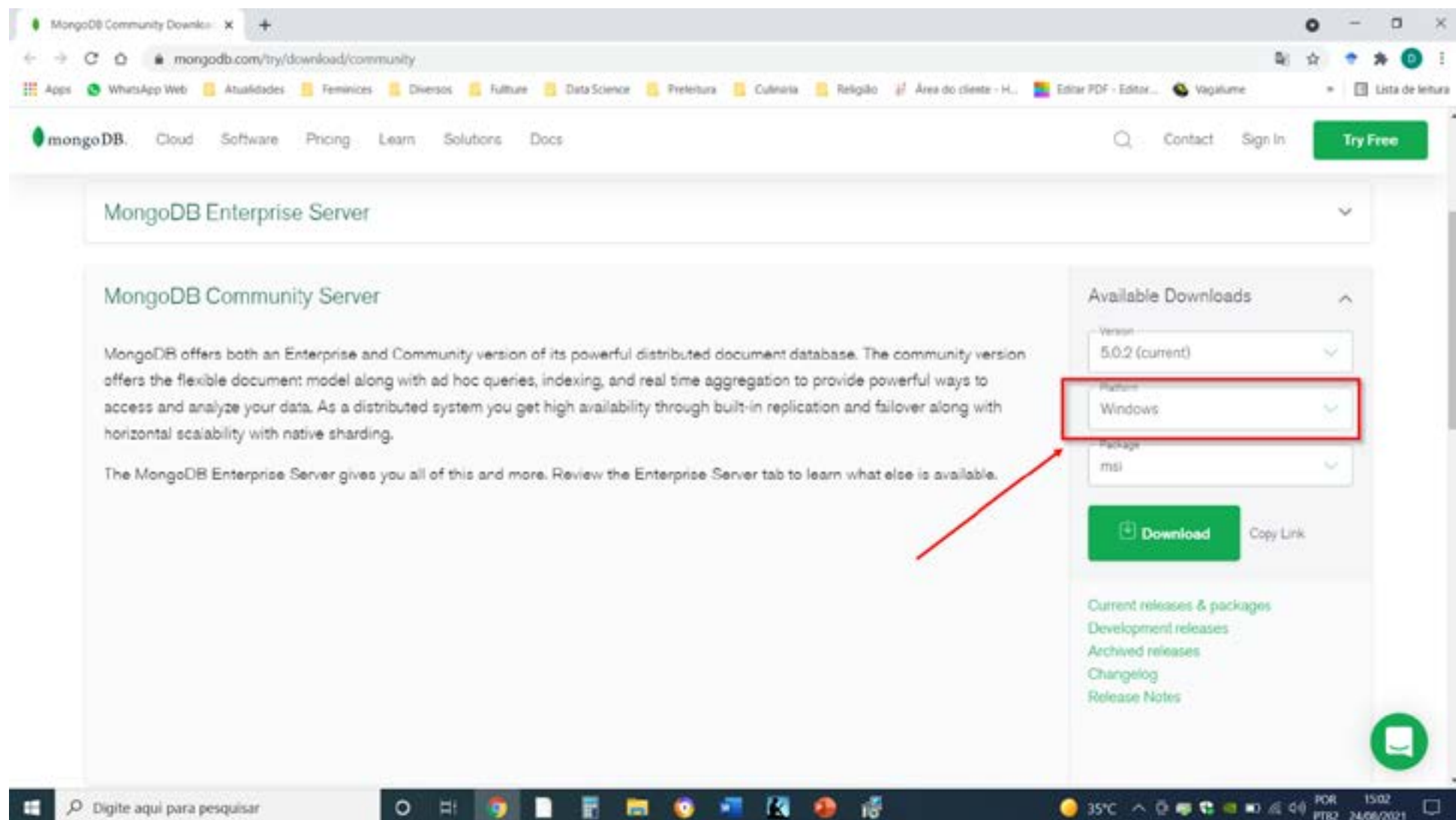
O MongoDB também possui um conjunto de ferramentas oficiais especialmente projetadas para o desenvolvedor e, nesta trilha, daremos foco ao MongoDB Compass, que é uma ferramenta interativa para visualização e manipulação dos dados. As quatro principais características desta ferramenta são:

- possibilidade de analisar os documentos e exibir as estruturas chaves;
- informações sobre o status do servidor;
- informações para análise do desempenho de consultas;
- informações para tomada de decisão sobre indexação e validação de documentos.

2.2.1 Instalando o banco de dados

O primeiro passo para a instalação do MongoDB é fazer o download do arquivo de instalação, acessando este link: <https://www.mongodb.com/try/download/community>

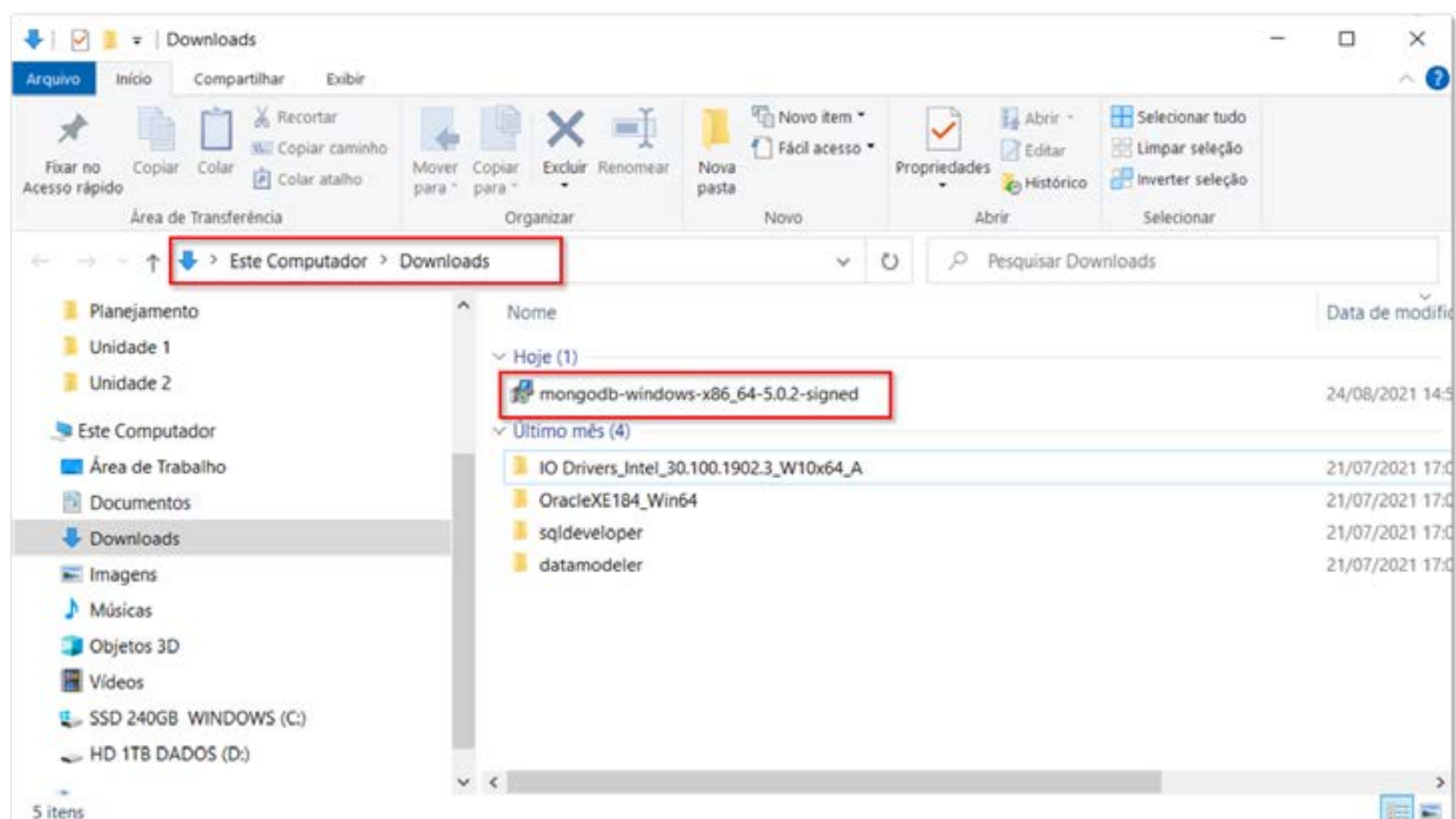
A Figura 06 demonstra a tela apresentada quando o link é acessado.



Fonte: Captura de tela do site MongoDB (2021).

A Figura 06 também destaca, em vermelho, a caixa de diálogo onde é possível selecionar para qual sistema operacional se deseja a instalação. Em nosso exemplo, utilizaremos o sistema operacional Windows, entretanto, os procedimentos serão praticamente os mesmos para os demais sistemas operacionais.

O arquivo de instalação será gravado no diretório de downloads do sistema. A Figura 07 destaca, ao final do download, o arquivo depositado neste diretório.



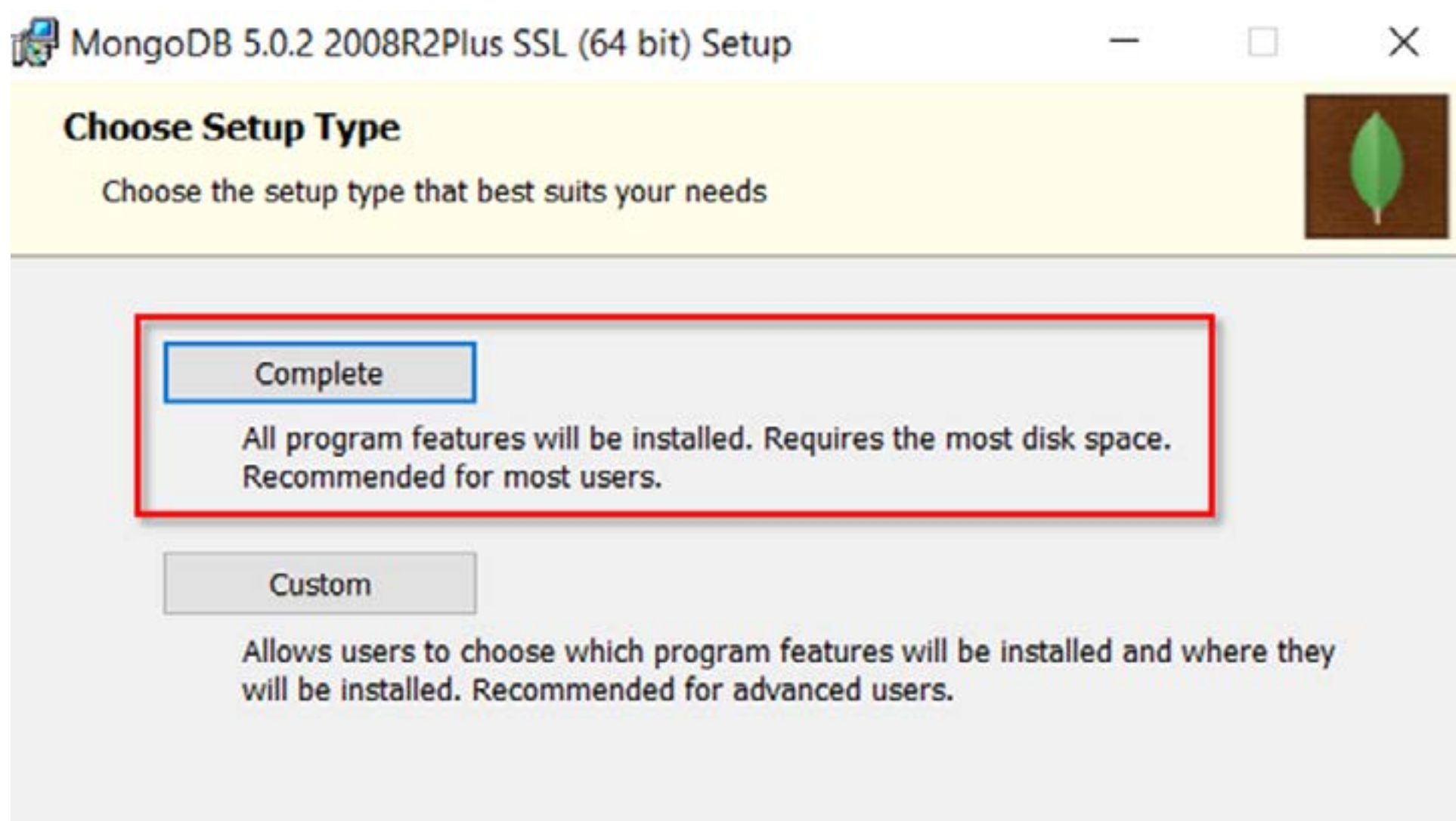
Fonte: Elaborada pelo autor (2021).

O próximo passo será executar o arquivo de instalação. Ao ser executado, o programa instalador apresentará a tela da Figura 08. Nesta tela, você deve escolher a opção “Next”.



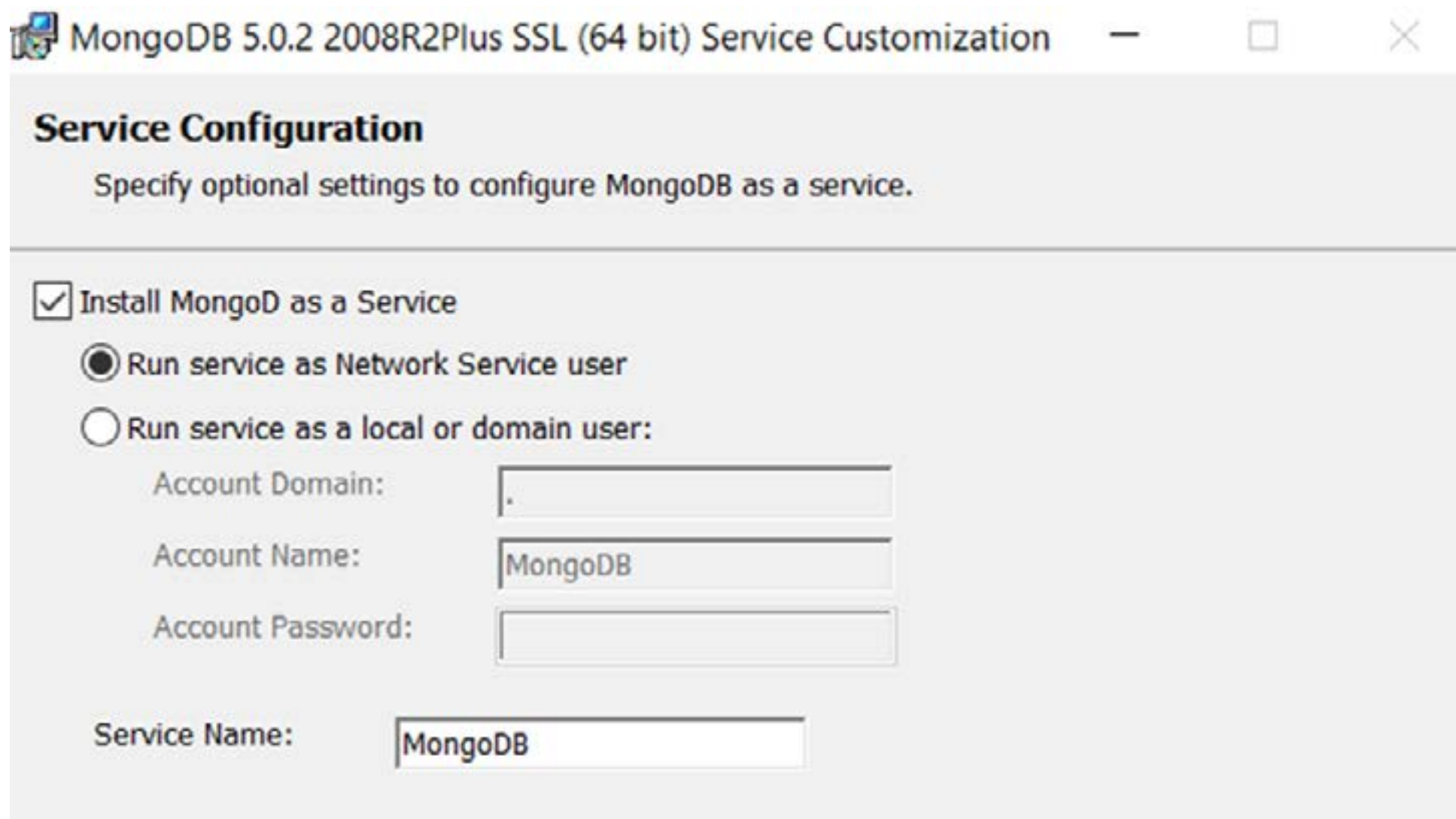
Fonte: Captura de tela do arquivo de instalação MongoDB (2021).

Após iniciar a execução do programa de instalação, você deverá aceitar os termos de uso e, a seguir, a tela da Figura 09 será apresentada. Selecione instalação completa.



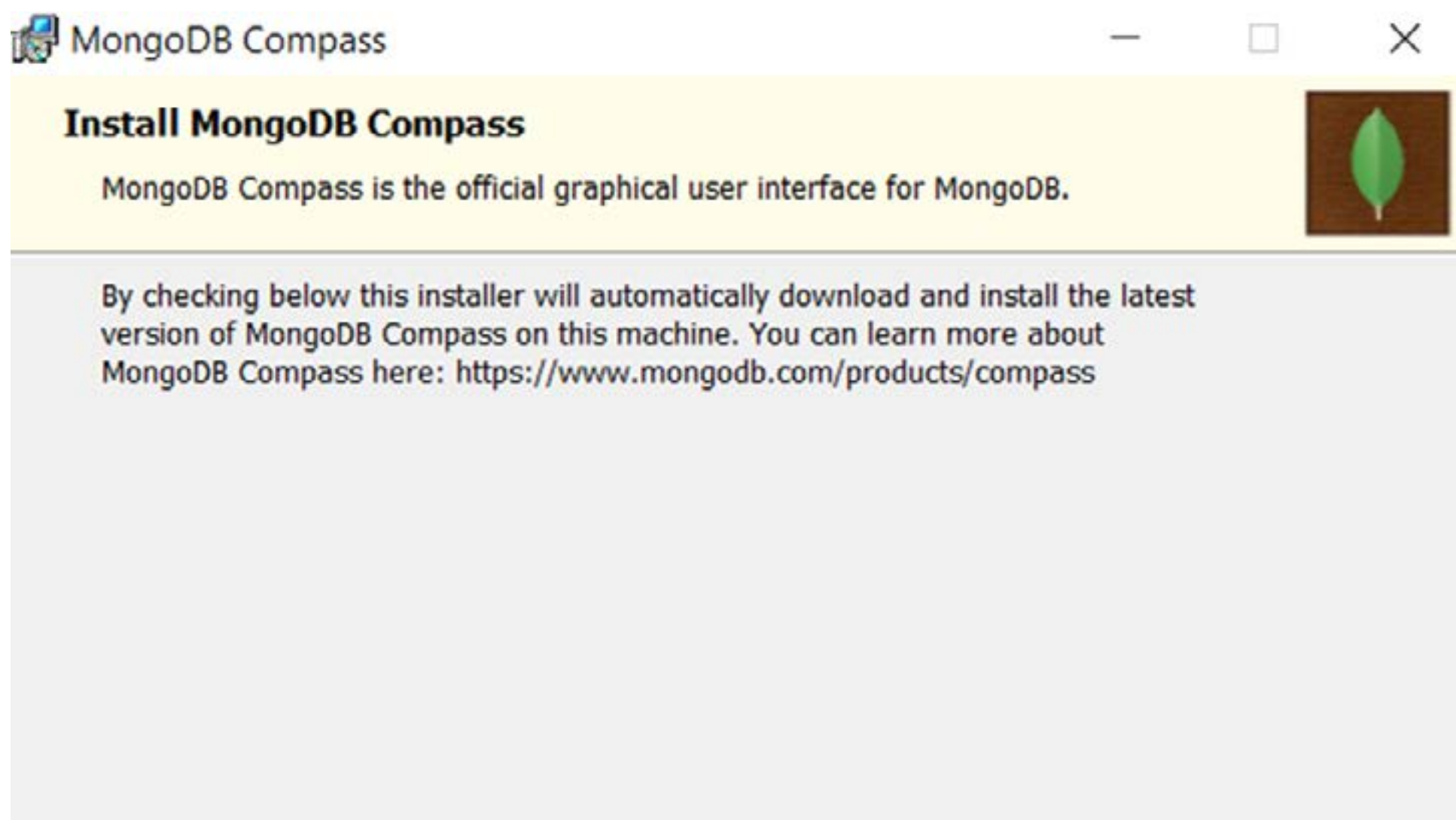
Fonte: Captura de tela do tipo de instalação do MongoDB (2021).

A tela da Figura 10 apresenta a possibilidade de alterar algumas opções de instalação do produto. Para nosso primeiro contato com o banco de dados, não há necessidade de modificação em qualquer destas opções. Você deve aceitar as opções sugeridas.



Fonte: Captura de tela das configurações de instalações do MongoDB (2021).

O programa ainda oferece a opção de instalar a ferramenta MongoDB Compass, demonstrada na Figura 11. O MongoDB Compass é uma excelente ferramenta e, ainda nesta unidade, vamos utilizá-la. Portanto, você também deve aceitar essa instalação.



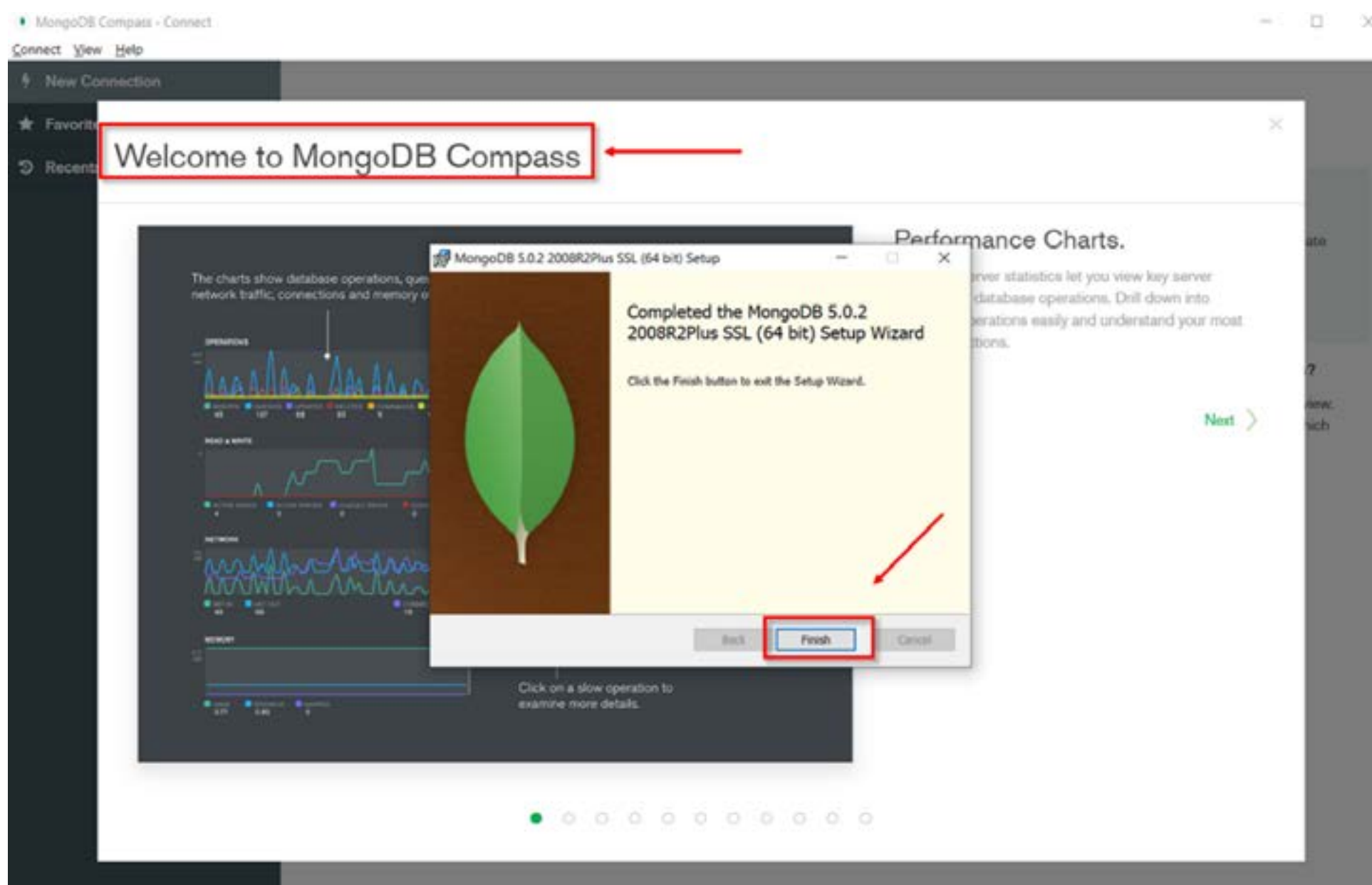
Fonte: Captura de tela da instalação da ferramenta MongoDB Compass (2021).

Neste ponto, todas informações necessárias à instalação já foram fornecidas. A tela da Figura 12 será apresentada e você deve confirmar a instalação do MongoDB.



Fonte: Captura de tela da instalação do MongoDB (2021).

Se a instalação tiver sido bem-sucedida, você verá a tela apresentada pela Figura 13 e basta apenas encerrar a execução do programa de instalação clicando em “Finish”.

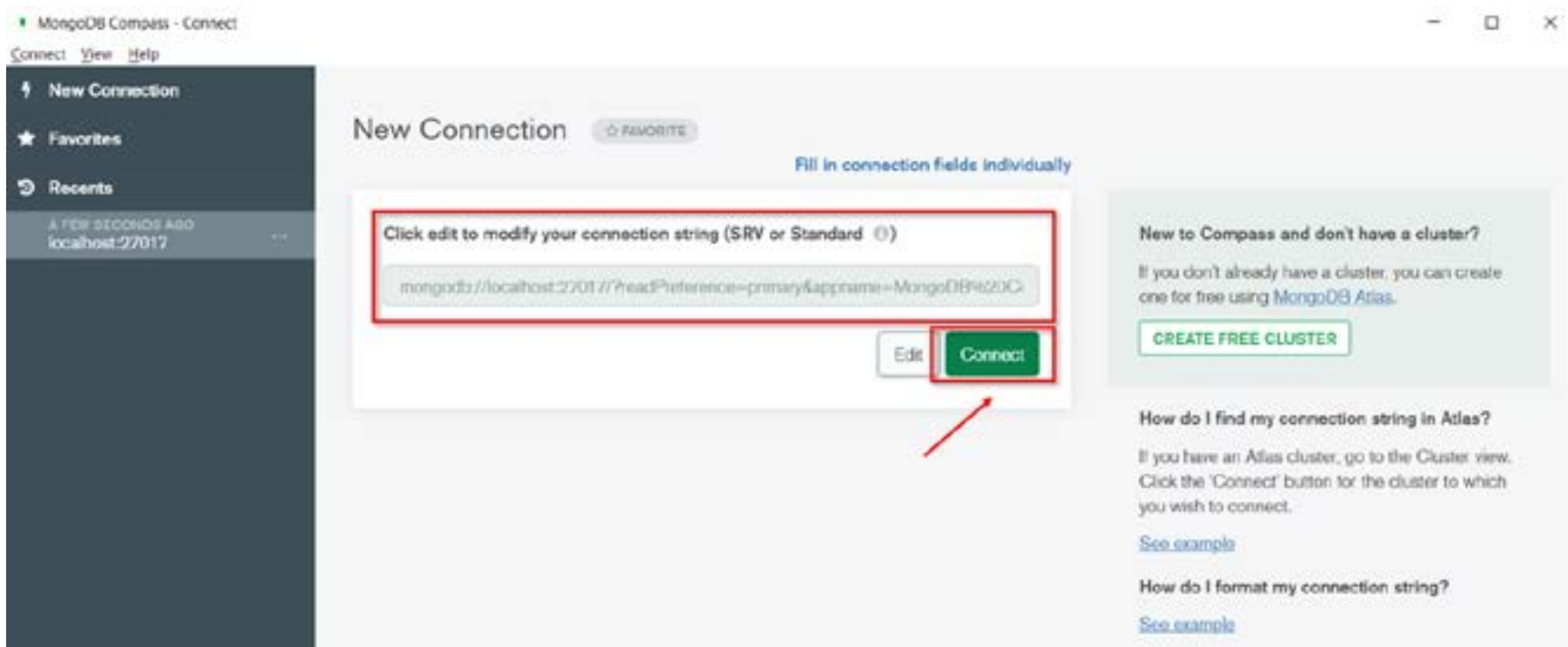


Fonte: Captura de tela da instalação bem-sucedida do MongoDB (2021).

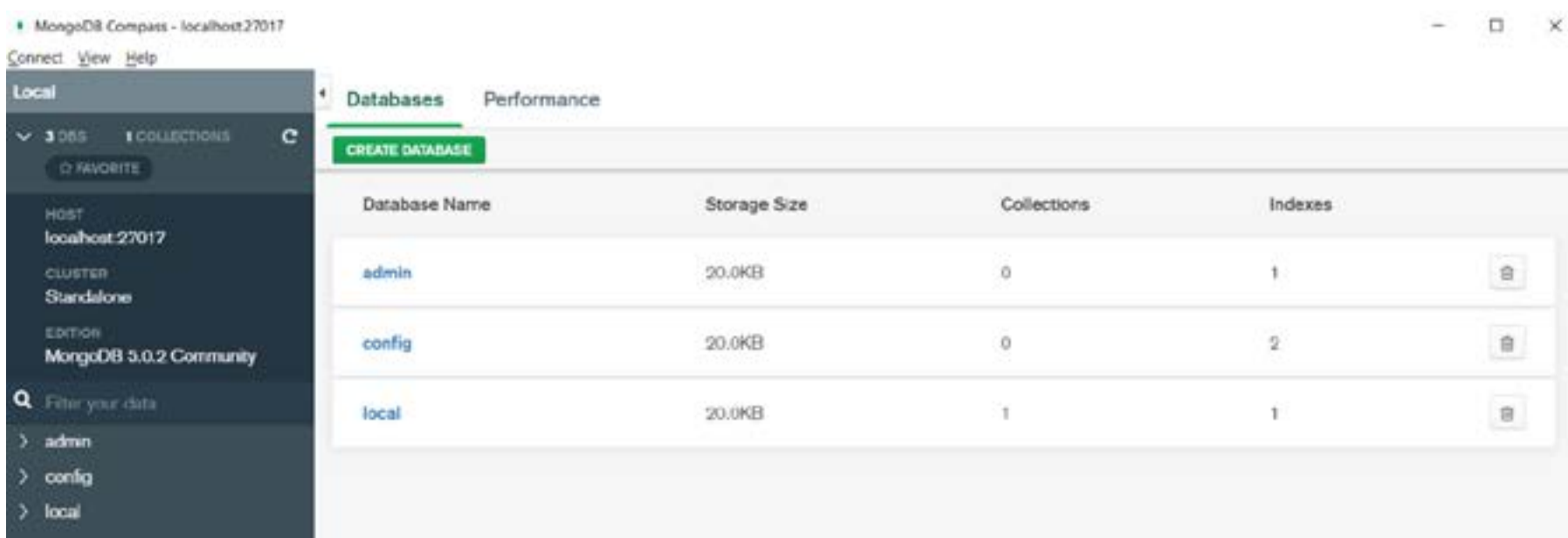
Com a instalação completa, no próximo tópico iremos explorar o MongoDB, por meio do MongoDB Compass.

2.2.2 Fazendo o primeiro acesso ao MongoDB

Após a instalação, um ícone do MongoDB terá sido disponibilizado na área de trabalho do computador. Então, faremos o primeiro acesso ao banco de dados através deste ícone. Ao acioná-lo, a tela da Figura 14 será apresentada, solicitando as informações para a conexão com o banco de dados. Como sugestão, o MongoDB Compass sugere o endereço da máquina que está executando a instalação, tecnicamente referenciamos esta máquina como localhost. Em uma execução em cluster você deveria informar o endereço da máquina que contém o banco de dados. Como usaremos o MongoDB na máquina local, não se preocupe, você pode aceitar os valores sugeridos pelo MongoDB e a conexão será feita.



Com a conexão feita, a tela da Figura 15 será apresentada, e, assim, o banco de dados MongoDB estará disponível para uso.



Com o MongoDB disponível para uso, no próximo tópico, você executará a inclusão de um documento no banco de dados e poderá verificar na prática os conceitos apresentados até o momento. Vamos lá!

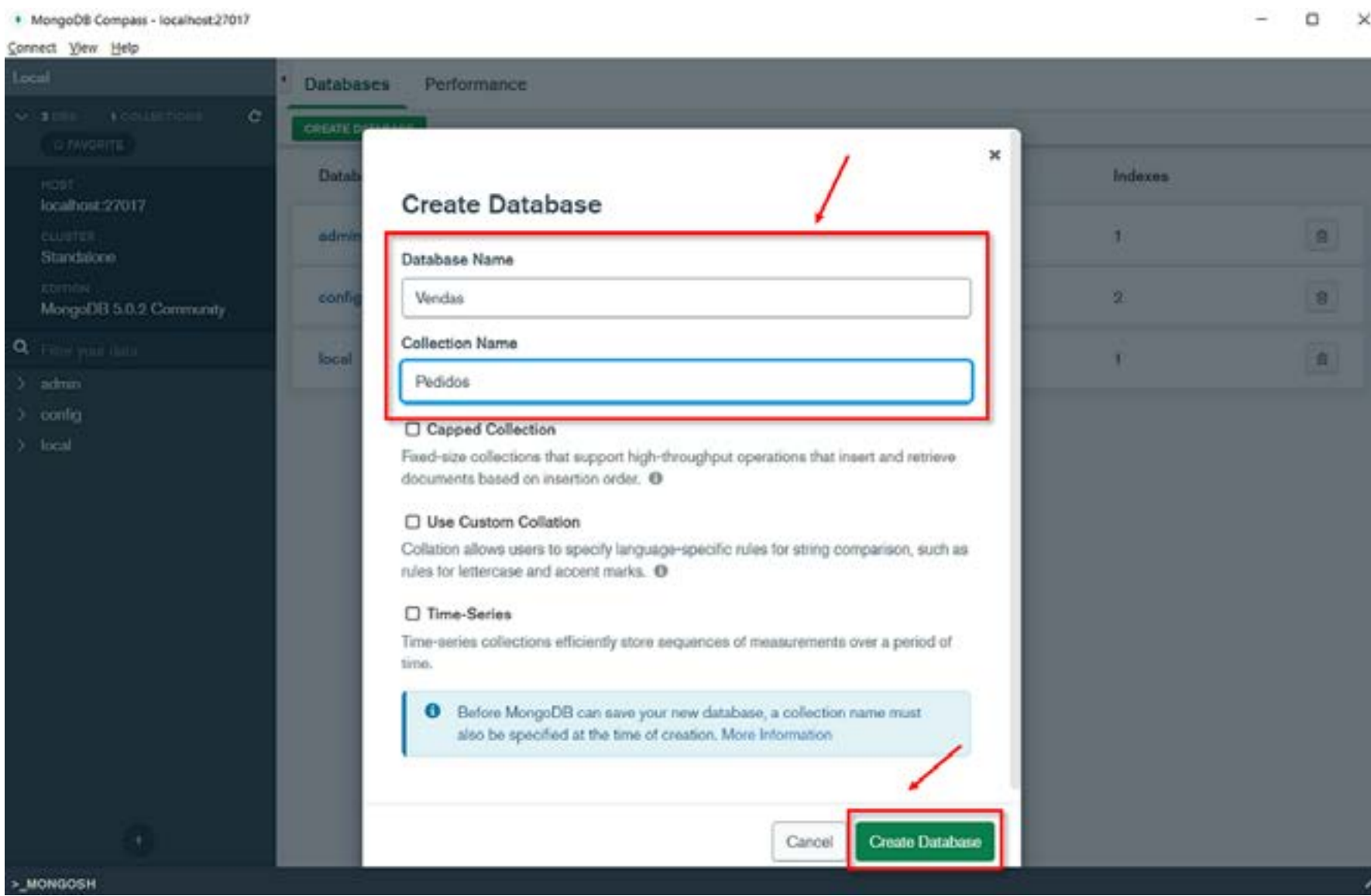
2.3 Explorando o MongoDB com o MongoDB Compass

Para o primeiro contato com o MongoDB Compass, vamos utilizar o mesmo documento apresentado na Figura 03, que mostra um documento padrão JSON para o armazenamento da informação de um pedido de venda para um cliente. Entretanto, antes de realmente iniciarmos a prática, ainda é necessário que você conheça alguns conceitos que serão utilizados. Como qualquer outro banco de dados, o MongoDB utiliza alguns objetos para a organização conceitual e lógica do armazenamento. É referenciando estes objetos que o desenvolvedor obtém acesso aos dados e às informações desejadas. O Quadro 01 apresenta alguns destes objetos.

Objeto	Descrição
Campo	É cada dado isolado, por exemplo, o número do pedido do cliente ou o código do produto comprado.
Documento	Unidade de agregação dos dados, por exemplo, o documento que agrega todos os dados do pedido do cliente, como número do pedido, código do cliente, produto etc.
Coleção	É um conjunto de documentos do mesmo tipo. Por exemplo, podemos ter a coleção “Pedidos do Cliente”, onde são armazenados todos os pedidos de vendas feitos pelos clientes.
Database	Divisão lógica, onde todas as coleções de uma aplicação são armazenadas. Por exemplo o Database de “Vendas”, pode conter as coleções de pedidos, notas fiscais etc.
Chave Primária	É um campo que identifica uma agregação de dados a respeito de uma entidade, por exemplo, código do cliente, número do pedido.
Índice	Um índice pode ser formado por um único campo ou um conjunto de campos. É usado pelo banco de dados para melhorar o desempenho das consultas.

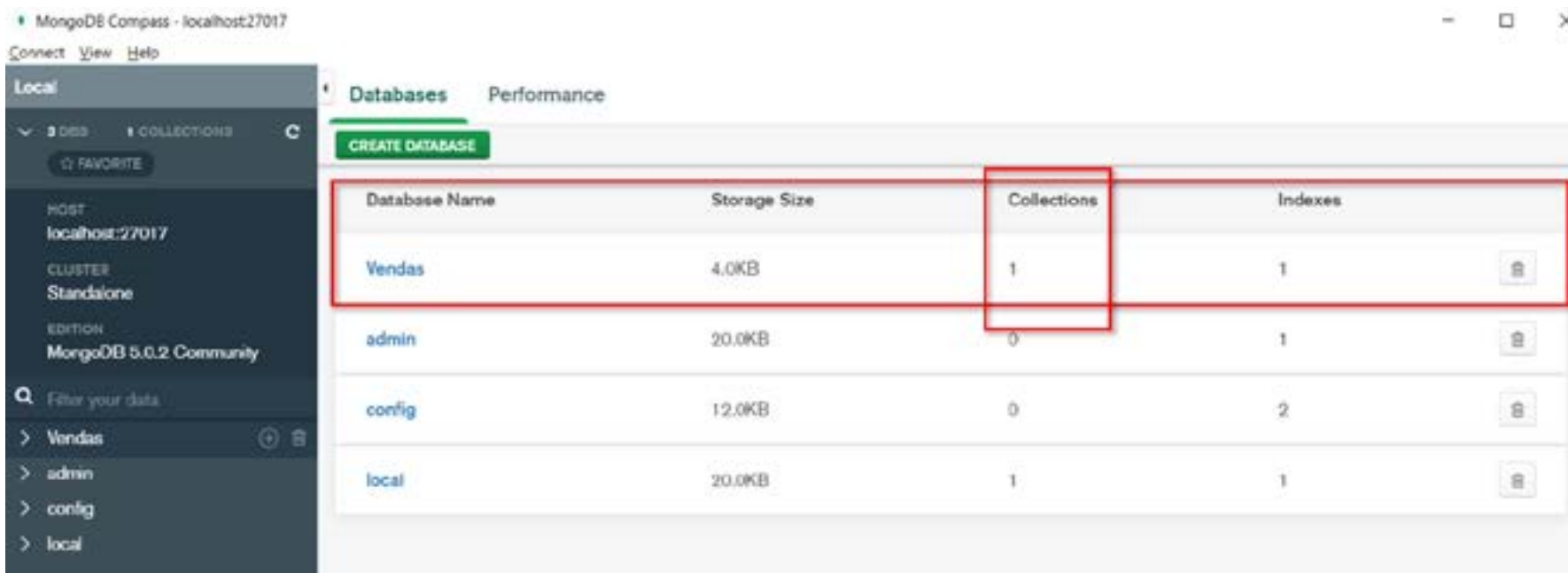
Fonte: elaborado pelo autor (2021).

Já conhecendo estes objetos, vamos iniciar a prática. A primeira coisa a fazer é criar o database para o armazenamento das coleções. Observe que a tela da Figura 15 apresenta o botão “CREATE DATABASE”. Ao acioná-lo, o MongoDB Compass apresentará a Figura 16.



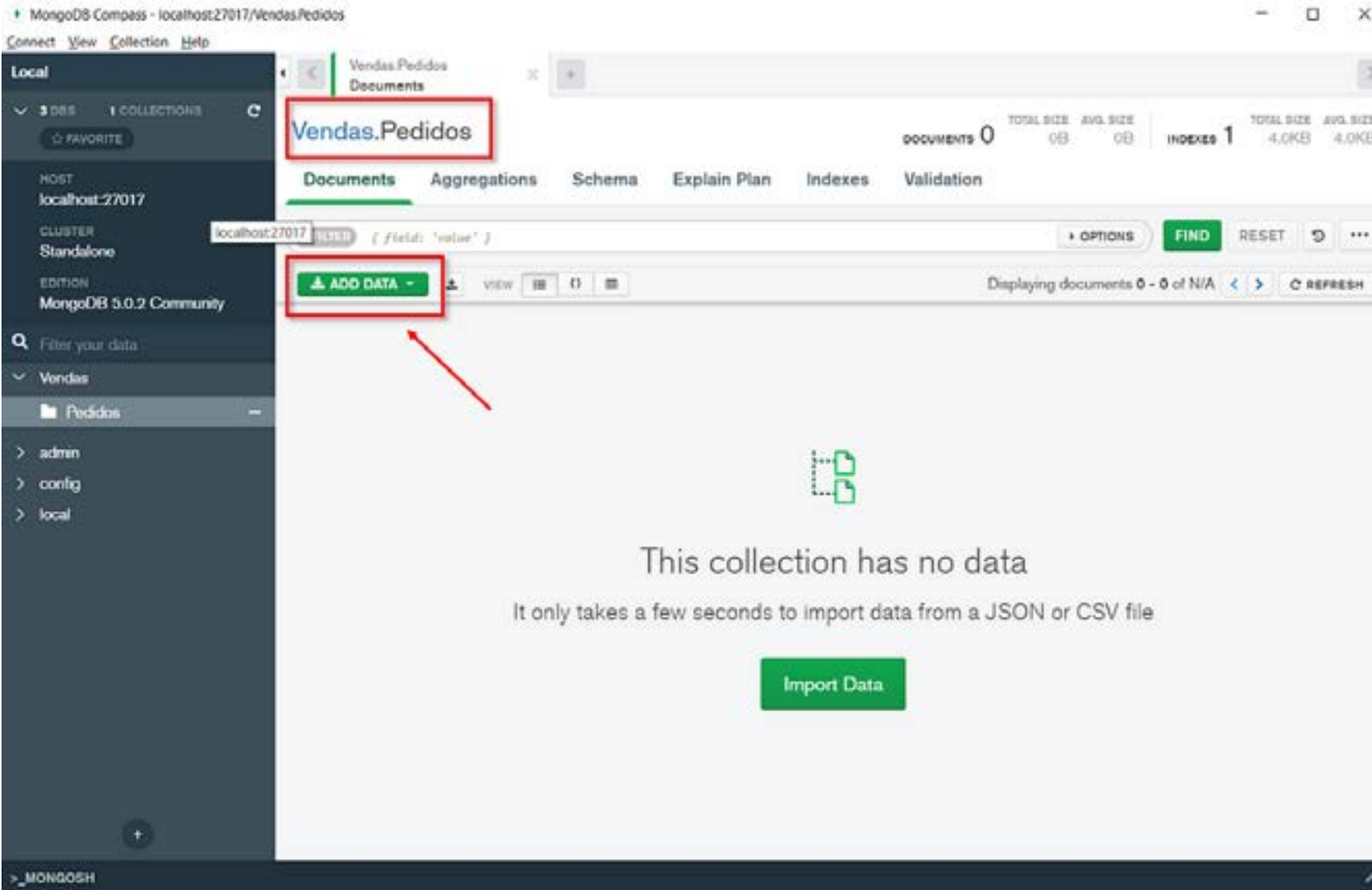
Fonte: Captura de tela do MongoDB (2021)

Conforme indicado na Figura 16, nomearemos o database como “Vendas” e a primeira coleção a ser criada será a coleção “Pedidos”. A Figura 17 apresenta o database e a coleção criados.



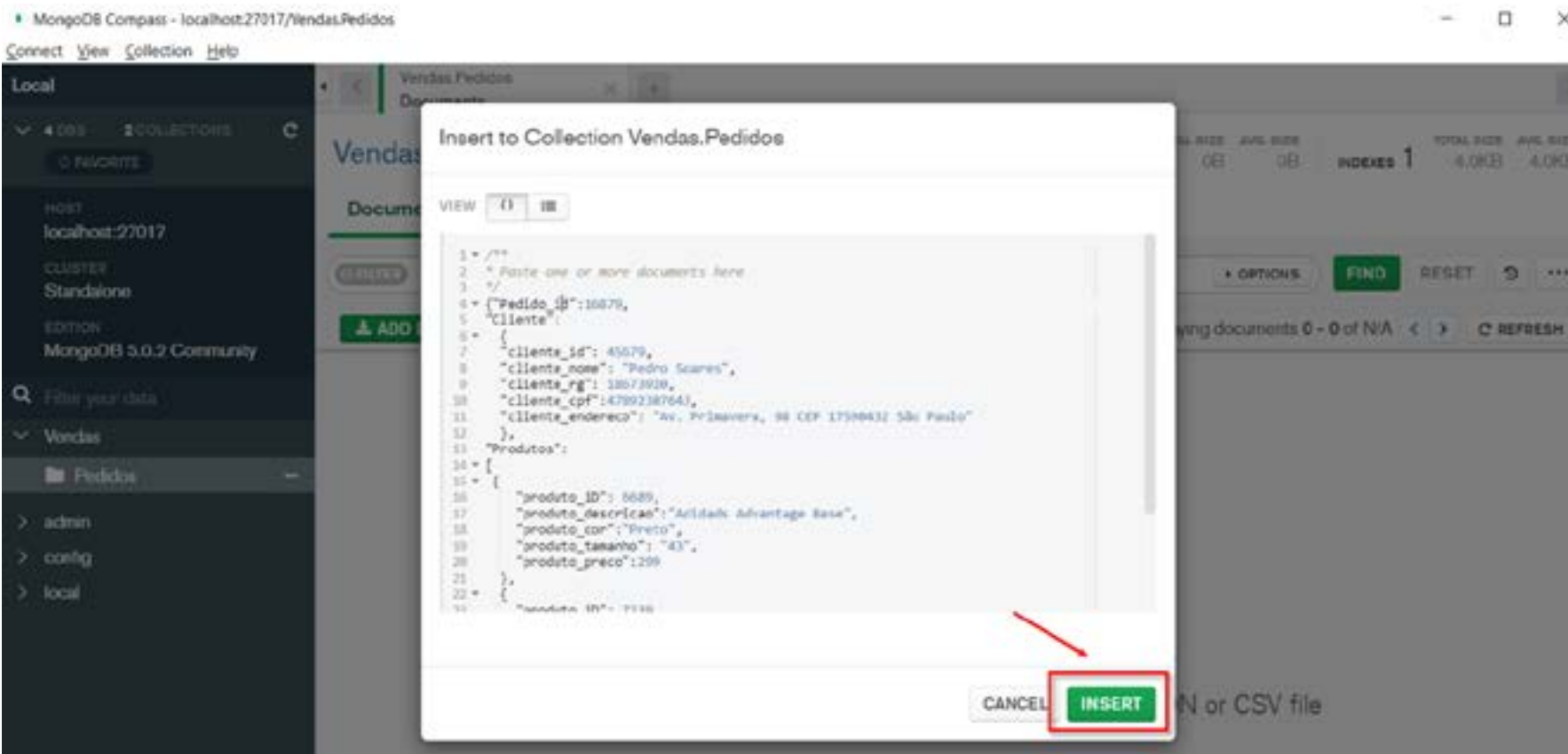
Fonte: Captura de tela do MongoDB (2021)

Agora, realize a inclusão do primeiro documento na coleção “Pedidos”. Ainda na tela apresentada na Figura 17, você deve selecionar o database em que iremos trabalhar, no caso, “Vendas”, e, em seguida, o MongoDB apresentará as coleções criadas neste database. Até o momento a única coleção criada é a de “Pedidos”. Você deverá selecionar esta coleção, e então o MongoDB Compass apresentará a tela da Figura 18.



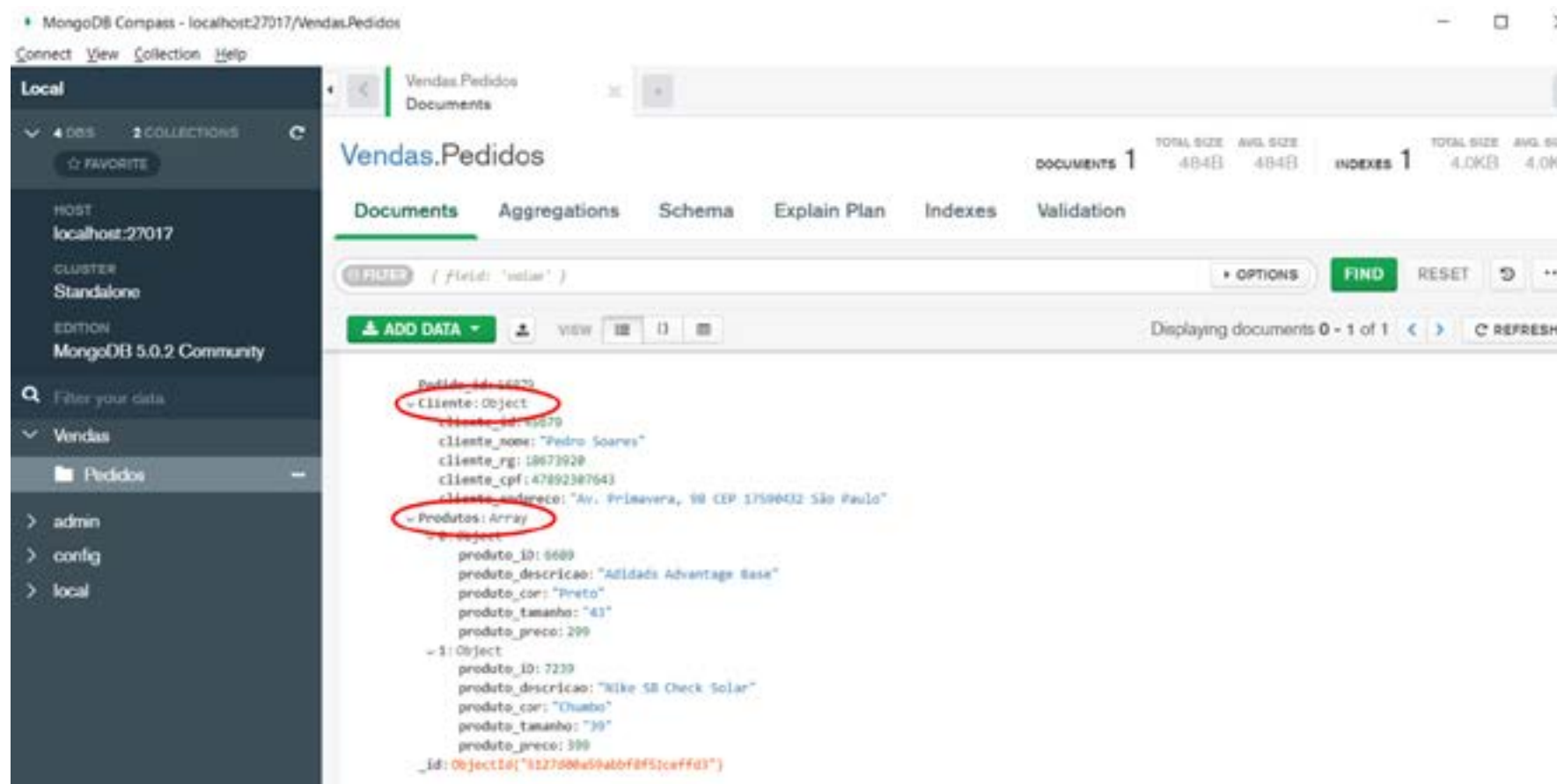
Fonte: Captura de tela do MongoDB (2021)

Ao acionar o botão “ADD DATA”, selecione a opção “Insert Document” e o MongoDB Compass apresentará uma janela com uma pré-formatação para a inclusão do documento. Nesta tela, você deve construir o documento conforme o conteúdo da Figura 03. A Figura 19 apresenta a tela com o documento exemplo já construído.



Fonte: Captura de tela do MongoDB (2021)

Ao acionar o botão INSERT, o MongoDB Compass apresentará o documento inserido. Você poderá vê-lo em sua totalidade, expandindo todos os níveis de agregação do documento, clicando no símbolo “>” ao lado de cada nível de agregação. A Figura 20 demonstra todos os níveis já expandidos.



Fonte: Captura de tela do MongoDB (2021)

ocê conseguiu perceber que o armazenamento em MongoDB e a utilização da ferramenta MongoDB Compass são bastante simples? Esta simplicidade favorece muito a produtividade na atividade de desenvolvimento. Saber como e quando usar esta simplicidade é saber extrair o melhor deste produto bastante poderoso. Vejamos, então, algumas aplicações, onde um banco de dados NoSQL com o modelo por documentos, se torna bastante adequado no desenvolvimento.

2.4 Alguns exemplos de aplicações para o MongoDB

Hoje em dia, praticamente tudo pode ser gerenciado por uma aplicação. Ao acessarmos uma loja de aplicativos, a Google Play por exemplo, podemos observar a variedade de opções e necessidades atendidas pelos mais variados aplicativos. Seria ingenuidade pensar que toda esta variedade de aplicações é atendida por uma única opção tecnológica.

Conforme apresentamos no tópico Armazenamento por documentos, saber escolher a ferramenta que atende melhor a necessidade é fundamental para o sucesso da aplicação. Tendo isso em mente, este tópico apresentará alguns tipos de aplicação que se beneficiam muito do modelo de armazenamento por documento. O primeiro tipo do qual vamos falar são as plataformas de BLOG e Comércio Eletrônicos.

2.4.1 Plataformas de BLOGS e Comércio

Estes são os exemplos mais intuitivos na utilização do modelo de armazenamento por documentos.

Sistemas de Blog são sistemas que publicam e gerenciam conteúdos na WEB. Geralmente, estes sistemas gerenciam conteúdos como publicação de informações, comentários, sons e imagens. Para este tipo de aplicação, podemos citar várias vantagens no armazenamento por documentos, entretanto, vamos citar o tratamento nativo na variedade dos tipos e estruturas de dados.

No comércio eletrônico, temos, como exemplo, os catálogos de produto. Nestas aplicações, cada produto pode ser armazenado por um documento. Uma vez que há uma grande variedade de atributos entre os produtos, isso pode ser tratado muito mais facilmente pelo armazenamento por documentos.

Em bancos de dados relacionais, cada tipo de atributo do produto deve ser previamente definido no esquema do banco de dados, e quando houver necessidade de inclusão de um novo atributo, há também a necessidade de se alterar, além de todas as funcionalidades que manipularão o novo atributo, o esquema do banco de dados.

No armazenamento por documentos, basta inserir o novo atributo somente no documento que trata as informações do produto em questão e alterar a aplicação que deverá considerar este atributo.

2.4.2 Análise em tempo real

Precisamos considerar que sistemas de análise em tempo real atendem demandas de análise em dados muito voláteis. O exemplo mais intuitivo, neste caso, é a análise de tweets, como a reação das pessoas em relação à uma notícia ou tendência.

Em relação aos dados, podemos citar a flexibilidade para construir rapidamente um documento com os dados de interesse. Em relação aos volumes, a aplicação deverá manipular volumes gigantescos de informação por um período curto, quase momentâneo. Isto exemplifica a característica bastante marcante dos bancos NoSQL e do armazenamento por documentos em relação ao desempenho e à capacidade elástica para trabalhar com cluster de máquinas.

Na próxima unidade, você terá a oportunidade de conhecer como manipular os dados

armazenados no MongoDB. A unidade 3 terá foco nos métodos de inserção, leitura, alteração e remoção de dados do banco – métodos conhecidos pela sigla CRUD (Create, Retrieve, Update e Delete). Até!

Síntese

Nesta unidade, você estudou mais profundamente as características mais marcantes dos bancos de dados NoSQL, a ausência de esquema e a alta escalabilidade. Também se aprofundou no modelo de armazenamento por documento e pôde verificar o quanto este modelo de armazenamento é simples e fácil de implementar. Para exercitar os conceitos apresentados, você realizou passo a passo a instalação do MongoDB e, ainda, utilizando a ferramenta MongoDB Compass, inseriu e visualizou um documento completo. Confira a seguir os itens mais importantes:

- Bancos de dados NoSQL surgiram com o objetivo de atender à crescente demanda por maior flexibilidade e simplicidade na manipulação de dados. Por esta razão, estes bancos de dados foram especialmente projetados para proporcionarem maior flexibilidade na manipulação de várias fontes de dados e maior desempenho na manipulação de altos volumes de informação.
- As características mais marcantes dos bancos de dados NoSQL são a ausência de esquema e a escalabilidade elástica no tratamento de altos volumes de dados.
- A ausência de esquema possibilita que uma nova fonte ou um novo dado seja facilmente inserido no contexto da aplicação.
- A habilidade de escalar, de forma elástica, altos volumes de informação traz vantagens, tanto na alta disponibilidade quanto na relação custo-benefício do processamento e do armazenamento dos dados.
- A modelagem através da agregação prévia de dados que futuramente serão frequentemente acessados em conjunto traz desempenho no processamento e simplicidade para a atividade de desenvolvimento.
- O armazenamento por documentos é uma boa ferramenta para a agregação prévia dos dados. A construção do documento é simples e fornece todos os recursos para a agregação lógica e física dos dados.
- O MongoDB é um banco de dados open source, que utiliza armazenamento por documentos e tem distribuição gratuita para Windows, Linux e Mac. Possui alta performance e diversas linguagens possuem drivers para este banco de dados.
- O MongoDB Compass é uma ferramenta interativa para visualização e manipulação dos dados. Através desta ferramenta pode-se analisar os documentos, exibir as estruturas chaves, exibir o status do servidor, obter informações sobre a indexação e o desempenho de consultas e a validação de documentos.

Fulture Insights

MONGODB. Community Server. 2021. Disponível em: <https://www.mongodb.com/try/download/community>. Acesso em: 20 set. 2021.

MONGODB. Guides. 2021. Disponível em: <https://docs.mongodb.com/guides/>. Acesso em: 20 set. 2021.

MONGODB. O banco de dados para aplicativos modernos. 2021. Disponível em: <https://www.mongodb.com/pt-br>. Acesso em: 13 set. 2021.

FU
LL
TU
RE

www.fulture.com