INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

RAISA PRISCILA DA SILVA

PROJETO DE BANCO DE DADOS NOSQL

CAMPOS DO JORDÃO

2024

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

RAISA PRISCILA DA SILVA

Entrega de pesquisa apresentado ao Instituto Federal de São Paulo (IFSP), em cumprimento a exigência da disciplina de Banco de Dados 2, do curso de Analise e Desenvolvimento de Sistemas.

PROFESSOR: Paulo Giovani de Faria Zeferine.

2024

RESUMO

Este trabalho apresenta uma pesquisa abrangente sobre sistemas de gerenciamento de banco de dados NoSQL, destacando suas características, vantagens, limitações e aplicações práticas. Inicialmente, a pesquisa contextualiza o surgimento dos bancos de dados NoSQL como uma resposta às limitações dos sistemas relacionais tradicionais frente ao crescimento exponencial de dados e à necessidade de alta performance e escalabilidade em aplicações modernas. O estudo explora os quatro principais tipos de bancos de dados NoSQL: chave-valor, orientado a documentos, colunar, grafos e objetos, detalhando suas estruturas, casos de uso e exemplos de implementações populares como MongoDB. Adicionalmente, são discutidos os principais desafios na adoção de NoSQL, incluindo a falta de padronização e a complexidade de migração de sistemas legados. A pesquisa também usará o sistema gerenciador de banco de dados MongoDB para fins comparativos com um sistema em SQL. Por fim, o trabalho conclui com uma reflexão sobre as tendências futuras e as áreas de pesquisa emergentes em NoSQL, sugerindo direções para estudos adicionais e melhorias tecnológicas.

Palavras-Chave: banco de dados; sistema; nosql; pesquisa.

ABSTRACT

This work presents a comprehensive survey of NoSQL database management systems, highlighting their characteristics, advantages, limitations and practical applications. Initially, the research contextualizes the emergence of NoSQL databases as a response to the limitations of traditional relational systems in the face of exponential data growth and the need for high performance and scalability in modern applications. The study explores the four main types of NoSQL databases: key-value, document-oriented, columnar and graph, detailing their structures, use cases and examples of popular implementations such as MongoDB. Additionally, the main challenges in adopting NoSQL are discussed, including the lack of standardization and the complexity of migrating from legacy systems. The research will also use the MongoDB database management system for comparative purposes with an SQL system. Finally, the work concludes with a reflection on future trends and emerging research areas in NoSQL, suggesting directions for additional studies and technological improvements.

Keywords: database; system; nosql; search.

LISTA DE ILUSTRAÇÕES

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Os bancos de dados NoSQL (Not Only SQL) surgiram como uma resposta às limitações dos bancos de dados relacionais tradicionais (SQL), especialmente em relação ao armazenamento e processamento de grandes volumes de dados não estruturados ou semi-estruturados. Esses sistemas oferecem flexibilidade, escalabilidade horizontal e alta performance, características essenciais para aplicações modernas, como redes sociais, sistemas de recomendação e big data..

O presente trabalho concentra-se na comparação entre os dois tipos de banco de dados e então na concepção de um sistema básico feito em NoSQL.

1.1 Objetivos

O objetivo desta pesquisa é explorar, analisar e avaliar os sistemas de gerenciamento de banco de dados NoSQL, investigando suas características distintivas, vantagens e limitações em comparação com os bancos de dados relacionais tradicionais. Esta pesquisa visa proporcionar uma compreensão profunda dos diferentes tipos de bancos de dados NoSQL, incluindo chave-valor, orientado a documentos, colunar e grafos, e suas respectivas arquiteturas, modelos de dados e casos de uso. A pesquisa também pretende oferecer uma análise comparativa detalhada entre os sistemas NoSQL e os bancos de dados relacionais, fornecendo insights sobre quando e por que uma organização pode optar por adotar soluções NoSQL.

1.2 Justificativa

Há uma demanda crescente por soluções de gerenciamento de dados que possam lidar eficientemente com dados não estruturados e semi-estruturados. Os bancos de dados NoSQL foram desenvolvidos precisamente para abordar essas necessidades, oferecendo modelos de dados mais flexíveis que permitem o armazenamento e a recuperação de grandes volumes de informações diversificadas.

Além disso, a diversidade dos sistemas NoSQL – que inclui bancos de dados chave-valor, orientados a documentos, colunar e grafos – apresenta uma ampla ga-

ma de opções para diferentes cenários de aplicação. Cada tipo de banco de dados NoSQL possui características específicas que o tornam mais adequado para determinados tipos de problemas.

Este trabalho justifica-se na importância de fornecer um entendimento claro e abrangente dos sistemas NoSQL, suas aplicações práticas e desafios, com o objetivo de apoiar organizações e profissionais na implementação de soluções de banco de dados que atendam às exigências contemporâneas de gerenciamento de dados.

1.3 Aspectos Metodológicos

O presente trabalho fez uso de pesquisas bibliográficas, utilizando de alguns autores para auxiliar no desenvolvimento dessa pesquisa comparativa.

1.4 Aporte Teórico

Foi utilizado arquivos e documentos da Microsoft e outros autores que enriquecem o aporte teórico, contribuindo no desenvolvimento dessa pesquisa, na comparação entre SQL e NoSQL e na criação de um sistemas sistema em NoSQL usando MongoDB para o presente trabalho.

2 PROJETO PROPOSTO

Nesta seção será apresentada a metodologia que utilizada neste trabalho, porque esta foi a escolhida, como eles foram elaborados e demais artefatos referentes a este projeto.

2.1 Considerações Iniciais

Os sistemas de gerenciamento de banco de dados NoSQL apresentaram nos últimos anos o crescente volume, variedade e velocidade dos dados gerados pelas modernas aplicações digitais. Esse cenário desafia os modelos tradicionais de banco de dados relacionais, que foram projetados para ambientes com requisitos diferentes dos atuais. Com a ascensão de big data, redes sociais, Internet das Coisas (IoT) e outras tecnologias emergentes, torna-se crucial explorar alternativas que possam oferecer melhor desempenho e escalabilidade. Os bancos de dados NoSQL surgem como uma resposta a essas necessidades, proporcionando maior flexibilidade e capacidade de lidar com dados não estruturados e semi-estruturados.

NoSQL não é uma tecnologia monolítica, mas uma coleção diversificada de sistemas de banco de dados que incluem modelos chave-valor, orientado a documentos, colunar e de grafos. Cada um desses tipos atende a diferentes necessidades e cenários de uso.

2.2 Relembrando SQL

SQL (Structured Query Language) é uma linguagem de programação padrão utilizada para gerenciar e manipular bancos de dados relacionais. Ela permite executar diversas operações como a criação de bancos de dados e tabelas, inserção, atualização, exclusão e consulta de dados. SQL é essencial para interagir com sistemas de gerenciamento de bancos de dados (SGBDs) relacionais, como MySQL, PostgreSQL, Oracle e SQL Server. Com comandos específicos, como SELECT, INSERT, UPDATE e DELETE, os usuários podem definir e alterar a estrutura dos dados, além

de realizar consultas complexas para extrair informações relevantes de grandes conjuntos de dados.

2.3 O que é NoSQL?

Os bancos de dados NoSQL (Not Only SQL) surgiram como uma resposta às limitações dos bancos de dados relacionais tradicionais (SQL), especialmente em relação ao armazenamento e processamento de grandes volumes de dados não estruturados ou semi-estruturados. Esses sistemas oferecem flexibilidade, escalabilidade horizontal e alta performance, características essenciais para aplicações modernas, como redes sociais, sistemas de recomendação e big data.

Os bancos de dados NoSQL se distinguem por não seguirem o modelo relacional de tabelas e esquemas fixos. Eles permitem a armazenagem de dados de forma mais flexível, adaptando-se melhor a diferentes tipos de aplicações e padrões de acesso. A escalabilidade horizontal é uma de suas principais vantagens, permitindo a distribuição de dados e operações através de múltiplos servidores, facilitando a expansão de infraestrutura conforme a necessidade.

Entre as vantagens dos bancos de dados NoSQL estão a flexibilidade no modelo de dados, a escalabilidade horizontal e a capacidade de lidar com grandes volumes de dados. Contudo, eles também apresentam desvantagens, como a falta de suporte a transações ACID completas em alguns casos e a necessidade de os desenvolvedores adaptarem suas abordagens ao modelo específico do banco de dados NoSQL escolhido.

2.4 Modelos de Dados NoSQL

Os bancos de dados NoSQL são classificados em diferentes modelos de dados, cada um adequado a necessidades específicas:

2.4.1 Documentos

Um armazenamento de dados de documento gerencia campos nomeados e seus valores em uma estrutura conhecida como documento. Tipicamente, esses

bancos de dados armazenam informações no formato JSON, onde cada campo pode conter dados simples, como números, ou estruturas mais complexas, como listas ou objetos pai-filho. Os dados podem ser codificados em vários formatos, como XML, YAML, JSON, BSON (JSON binário) ou texto sem formatação. Os campos dos documentos são acessíveis ao sistema de gerenciamento de armazenamento, permitindo que aplicativos consultem e filtrem dados com base nesses valores.

O aplicativo pode recuperar documentos utilizando a chave de documento. A chave é um identificador exclusivo do documento, que geralmente tem um hash, para ajudar a distribuir os dados de maneira uniforme. Alguns bancos de dados de documentos criam a chave do documento automaticamente. Outros permitem que você especifique um atributo do documento a ser usado como chave.

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

Figura 1 – Exemplo do Modelo de Documentos

2.4.2 Colunas

Um armazenamento de dados de colunas ou de família de colunas organiza dados em estruturas de colunas e linhas, semelhantes a um banco de dados relacional em conceito básico. No entanto, sua abordagem desnormalizada permite lidar efici-

entemente com estruturas de dados esparsas. Cada família de colunas contém um conjunto de colunas logicamente relacionadas, tratadas como unidades indivisíveis para recuperação e manipulação de dados. Essa flexibilidade permite adicionar novas colunas dinamicamente e suporta linhas com valores variáveis, tornando-o ideal para esquemas de dados variados e dinâmicos.

CustomerID	Column Family: Identity
001	**First name:** Mu Bae **Last name:** Min
002	**First name:** Francisco **Last name:** Vila Nova **Suffix:** Jr.
003	**First name:** Lena **Last name:** Adamcyz **Title:** Dr.

CustomerID	Column Family: Contact Info
001	**Phone number:** 555-0100 **Email:** someone@example.com
002	**Email:** vilanova@contoso.com
003	**Phone number:** 555-0120

Figura 2 – Exemplo do Modelo de Colunas

2.4.3 Chave-Valor

Um armazenamento de valor/chave é como uma tabela de hash ampliada, onde cada valor de dados é associado a uma chave exclusiva utilizando uma função de hash apropriada para distribuir uniformemente as chaves pelo armazenamento. Este tipo de armazenamento suporta operações básicas como inserção, exclusão e consulta simples. Para modificar um valor, é necessário substituir o valor completo existente, pois a maioria das implementações considera a leitura ou gravação de um valor como operação atômica. Alguns armazenamentos podem impor limites ao tamanho máximo dos valores armazenados, que são opacos para o sistema de armazenamento, exigindo que o aplicativo gerencie quaisquer detalhes de esquema associados aos dados.

Em essência, o armazenamento de valor/chave opera tratando os valores como blobs e acessando-os através de suas chaves.

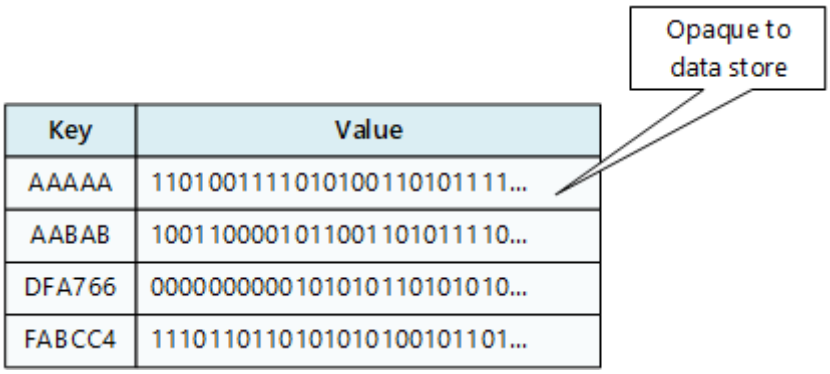

Figura 3 – Exemplo do Modelo de Chave-Valor

2.4.4 Grafos

Projetado para representar e manipular dados que possuem estruturas complexas de relacionamentos entre entidades, o modelo de grafos utiliza a teoria dos grafos para armazenar informações na forma de nós, arestas e propriedades. Esse modelo é particularmente eficaz para aplicações que necessitam de representações detalhadas de redes complexas de interconexões.

Oferecendo uma abordagem poderosa para representar e manipular dados complexos com relacionamentos ricos, é ideal para aplicações que dependem fortemente da análise e exploração de redes e interconexões entre entidades.

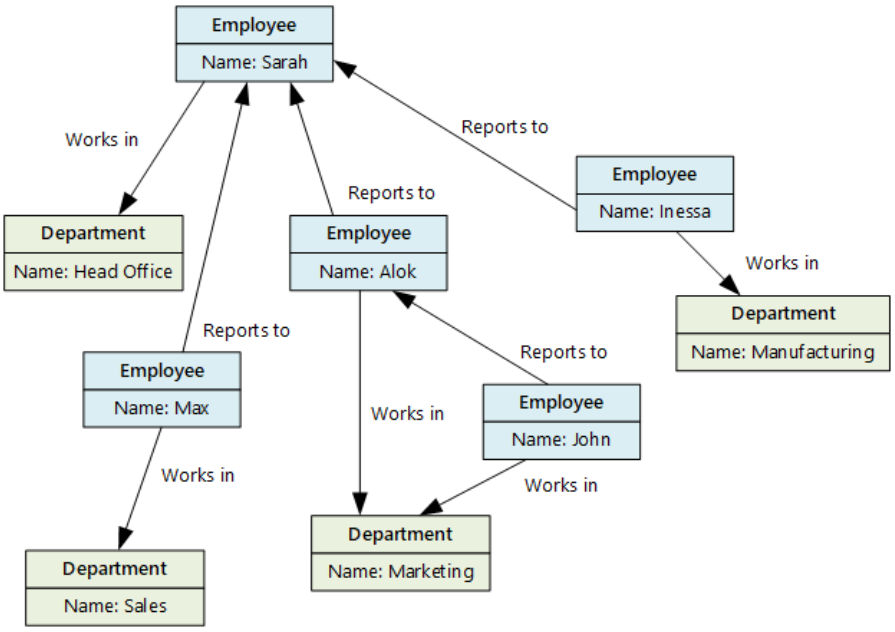

Figura 4 – Exemplo do Modelo de Grafos

2.4.5 Objetos

Os armazenamentos de dados de objetos são otimizados para armazenar e recuperar grandes objetos binários (blobs), como imagens, arquivos de texto, áudio e vídeo, documentos extensos e dados de aplicativos, incluindo imagens de discos de máquinas virtuais. Cada objeto contém os dados em si, metadados relevantes e uma ID única para acesso.

Esses sistemas são projetados para lidar com arquivos individualmente volumosos e oferecem grande capacidade de armazenamento total para gerenciar toda a gama de arquivos. Alguns armazenamentos replicam blobs específicos em múltiplos nós de servidor, possibilitando leituras paralelas rápidas. Esse método facilita consultas distribuídas em dados extensos, permitindo que vários processos em diferentes servidores consultem simultaneamente os mesmos arquivos grandes.

path	blob	metadata
/delays/2017/06/01/flights.cscv	0XAABBCCDDEEF...	{created: 2017-06-02}
/delays/2017/06/02/flights.cscv	0XAADDCCDDEEF...	{created: 2017-06-03}
/delays/2017/06/03/flights.cscv	0XAEBBDEDDEEF...	{created: 2017-06-03}

Figura 5 – Exemplo do Modelo de Objetos

2.5 SQL versus NoSQL

SQL (Structured Query Language) e NoSQL (Not Only SQL) representam duas abordagens distintas para o gerenciamento de bancos de dados, cada uma com suas próprias vantagens e desvantagens, adequadas a diferentes tipos de aplicações e requisitos de dados.

Modelos de Dados: SQL é baseado no modelo relacional, onde os dados são organizados em tabelas com linhas e colunas. Esse modelo é altamente estruturado, o que facilita a integridade referencial e a normalização dos dados. Por outro lado, NoSQL engloba uma variedade de modelos de dados, incluindo chave-valor, orien-

tado a documentos, colunar e grafos. Esses modelos são mais flexíveis, permitindo o armazenamento de dados não estruturados e semi-estruturados, o que é ideal para aplicações que lidam com grandes volumes de dados diversificados.

Escalabilidade: Os bancos de dados SQL tradicionalmente escalam verticalmente, o que significa que a capacidade do servidor precisa ser aumentada (mais CPU, memória, etc.) para lidar com a crescente carga de trabalho. Em contraste, NoSQL é projetado para escalar horizontalmente, distribuindo a carga de trabalho através de múltiplos servidores. Essa escalabilidade horizontal é crucial para aplicativos que precisam processar grandes quantidades de dados e tráfego, como redes sociais e plataformas de e-commerce.

Consistência e Disponibilidade: SQL segue o modelo ACID (Atomicidade, Consistência, Isolamento e Durabilidade), garantindo transações seguras e consistentes. Isso é essencial para aplicações onde a integridade dos dados é crítica, como sistemas bancários e de pagamento. NoSQL, por outro lado, muitas vezes adota o modelo BASE (Basicamente Disponível, Estado Soft, Consistência Eventual), que prioriza a disponibilidade e a escalabilidade em detrimento da consistência imediata. Isso pode ser vantajoso para aplicações onde a disponibilidade contínua é mais importante do que a consistência em tempo real, como redes sociais e análise de grandes volumes de dados.

Flexibilidade de Esquema: Bancos de dados SQL requerem um esquema predefinido, o que significa que a estrutura dos dados deve ser definida antes que os dados possam ser inseridos. Isso pode ser uma limitação em ambientes onde os requisitos de dados mudam frequentemente. NoSQL, por outro lado, oferece flexibilidade de esquema, permitindo que a estrutura dos dados evolua ao longo do tempo sem a necessidade de redefinição do esquema. Isso torna NoSQL ideal para desenvolvimento ágil e iterações rápidas.

Consultas e Linguagem: SQL utiliza a linguagem de consulta SQL, que é poderosa e expressiva, permitindo consultas complexas e operações de junção entre múltiplas tabelas. NoSQL, devido à sua diversidade de modelos de dados, não possui uma linguagem de consulta padrão. As consultas em NoSQL são frequentemente mais simples e dependem do tipo específico de banco de dados (por exemplo, MongoDB usa consultas JSON, enquanto Cassandra utiliza CQL, uma variação de SQL).

Casos de Uso: SQL é ideal para aplicações que requerem transações comple-

xas e consistência de dados, como sistemas financeiros, CRM e ERP. NoSQL é mais adequado para aplicações que necessitam de alta escalabilidade e flexibilidade, como big data, IoT, redes sociais e serviços em tempo real.

A escolha entre SQL e NoSQL depende das necessidades específicas da aplicação. SQL oferece estrutura, consistência e suporte transacional robusto, sendo ideal para ambientes onde a integridade dos dados é crucial. NoSQL, com sua flexibilidade, escalabilidade e capacidade de lidar com grandes volumes de dados não estruturados, é mais adequado para aplicações modernas que demandam agilidade e desempenho em escala. Ambas as abordagens têm seu lugar no ecossistema de dados, e a decisão de qual usar deve ser guiada pelos requisitos específicos do projeto e pela natureza dos dados a serem gerenciados.

2.6 MongoDB

O SGDB escolhido para esse trabalho é o MongoDB, que é um sistema de banco de dados NoSQL orientado a documentos, desenvolvido para lidar com grandes volumes de dados não estruturados ou semi-estruturados de maneira eficiente. Ele adota um modelo de dados orientado a documentos, onde os dados são armazenados em documentos no formato BSON (Binary JSON), uma extensão do JSON que suporta tipos de dados adicionais como Date, Binary, e outros.

Em MongoDB, um documento é uma estrutura de dados flexível que pode conter pares de campos e valores, sendo similar a um objeto JSON. Cada documento é atribuído a uma chave única dentro de uma coleção, que pode ser pensada como equivalente a uma tabela em bancos de dados relacionais, mas sem a rigidez de um esquema fixo. Isso significa que cada documento em uma coleção pode ter estruturas diferentes, permitindo uma modelagem de dados dinâmica e adaptável às necessidades da aplicação.

3 RESULTADOS OBTIDOS

Nessa seção será apresentado o resultado de um sistema básico de uma biblioteca criado no SGDB MongoDB.

3.1 Regras de Negócio

Considerando um sistema de gerenciamento de biblioteca como exemplo, será criado as seguintes tabelas: Livros, Autores, Usuários e Empréstimos. A seguir, as regras de negócio dessas tabelas:

Para o cadastro de livros, cada um deve ter um título único e estar associado a pelo menos um autor. Para o cadastro de autores, é importante saber que um autor pode ter vários livros associados a ele.

Sobre o cadastro de usuários, cada um deve ter um número de identificação e o sistema deve registrar a data de cadastro do usuário. Para os empréstimos de livros, um usuário pode emprestar até 3 livros por vez, e cada empréstimo deve ter uma data de inicio e uma data de devolução prevista, e também não é possível emprestar o mesmo livro para o mesmo usuário, ou para outro, ao mesmo tempo. Após a devolução, o sistema deve registrar a data real de devolução e calcular se houve atraso. O sistema deve ser capaz de manter um registro de todos os livros emprestados no momento.

Estas são regras básicas para um sistema de gerenciamento de biblioteca. Elas garantem a integridade dos dados e ajudam a manter o sistema organizado e funcional para todos os usuários.

3.2 Estrutura do Banco de Dados no MongoDB

Para implementar o sistema de gerenciamento de biblioteca em MongoDB, vamos estruturar o banco de dados utilizando coleções para representar as entidades Livros, Autores, Usuários e Empréstimos. MongoDB utiliza um modelo de dados baseado em documentos BSON (JSON-like) dentro de coleções. Abaixo está um exemplo de como poderíamos modelar esse sistema:

```
{
  "livros": [
    {
      "_id": ObjectId("5d27d84f15d25e6431a1893a"),
      "titulo": "Dom Casmurro",
      "idAutor": ObjectId("5d27d84f15d25e6431a1893b"),
      "disponivel": true
    }
  ],

  "autores": [
    {
      "_id": ObjectId("5d27d84f15d25e6431a1893b"),
      "nome": "Machado de Assis"
    }
  ],

  "usuarios": [
    {
      "_id": ObjectId("5d27d84f15d25e6431a1893c"),
      "nome": "João Silva",
      "dataCadastro": ISODate("2023-06-01T00:00:00Z")
    }
  ],

  "emprestimos": [
    {
      "_id": ObjectId("5d27d84f15d25e6431a1893d"),
      "idLivro": ObjectId("5d27d84f15d25e6431a1893a"),
      "idUsuario": ObjectId("5d27d84f15d25e6431a1893c"),
      "dataInicio": ISODate("2023-06-10T00:00:00Z"),
      "dataDevolucao": ISODate("2023-06-24T00:00:00Z"),
      "dataDevolucaoReal": ISODate("2023-06-23T15:30:00Z")
    }
  ]
}
```

Algoritmo 1 – Sistema em JSON

4 CONCLUSÃO

Os bancos de dados NoSQL representam uma evolução significativa no gerenciamento de dados, proporcionando soluções flexíveis e escaláveis para as demandas crescentes de armazenamento e processamento de informações das aplicações modernas. Com uma variedade de modelos de dados, eles oferecem ferramentas adaptáveis para diferentes cenários, tornando-se uma escolha indispensável no ecossistema tecnológico atual.

REFERÊNCIAS

MICROSOFT. Azure Microsoft, c2024. Banco de Dados NoSQL – O que é NoSQL?. Disponível em: <*https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-nosql-database#:~:text=Os%20bancos%20de%20dados%20NoSQL%20podem%20ser%20chamados%20de%20"não,SQL)%20com%20linhas%20e%20tabelas*>*. Acesso em: 17 de* junho *de 2024.*

MICROSOFT*.* Learn Microsoft, c2024. Dados não relacionais e NoSQL. *Disponível em:* <*https://learn.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data*>. Acesso em 17 de junho de 2024.

AMAZON*.* Amazon, c2024. O que é NoSQL?. Disponível em: < *https://aws.amazon.com/pt/nosql/*>. Acesso em 18 de junho de 2024.