

THIAGO ADRIANO

POSTECH

MACHINE LEARNING ENGINEERING

BANCOS DE DADOS

AULA 02

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	12
REFERÊNCIAS.....	13

EMANSP

O QUE VEM POR AÍ?

Nesta aula conheceremos o NoSQL, um modelo de banco de dados que está cada vez mais difundido no mercado.

Vamos praticar um pouco com o MongoDB, SGBD orientado a documentos. Criaremos exemplos de coleções e documentos, além de praticar operações de inserção, exclusão, alteração e consulta de dados.

Na aula em vídeo, nós abordaremos DynamoDB, um serviço de banco de dados NoSQL totalmente gerenciado pela Amazon Web Services (AWS).

Para acessar o GitHub desta aula, clique [aqui](#).

HANDS ON

Nesta aula prática, vamos abordar algumas das principais características dos bancos de dados NoSQL. Para isso, exploraremos um serviço de banco de dados NoSQL gerenciado na nuvem: o DynamoDB, da Amazon Web Services (AWS).



SAIBA MAIS

Banco de dados NoSQL (Not Only SQL) são projetados para armazenar e recuperar informações em formato de documentos, geralmente no formato JSON (JavaScript Object Notation) ou XML (eXtensible Markup Language).

Diferentemente dos bancos de dados relacionais, em que os dados são armazenados em tabelas com esquemas predefinidos, os bancos de dados documentais permitem uma estrutura flexível, na qual os documentos podem ter diferentes campos e estruturas, sem a necessidade de um esquema rígido.

Essa flexibilidade é especialmente útil em cenários onde os dados são semiestruturados ou variam em sua estrutura ao longo do tempo.

Esses bancos de dados são amplamente utilizados em diversas aplicações, como gerenciamento de conteúdo, sistemas de gerenciamento de documentos, aplicações web e mobile, análise de big data e personalização de conteúdo.

Eles oferecem vantagens como escalabilidade horizontal, alta disponibilidade e desempenho, permitindo o armazenamento e recuperação eficiente de grandes volumes de dados.

Alguns exemplos populares de bancos de dados documentais incluem o MongoDB, o Couchbase e o DocumentDB. Essas tecnologias têm sido adotadas por empresas de todos os tamanhos, de startups a grandes corporações, para lidar com a complexidade e o volume crescente de dados semiestruturados em suas aplicações.

O MongoDB, um dos primeiros e mais conhecidos bancos de dados documentais, foi lançado em 2009. Ele foi pioneiro no modelo de armazenamento de documentos flexíveis e escalabilidade horizontal.

O sucesso do MongoDB incentivou o surgimento de outros bancos de dados documentais, como o Couchbase e o DocumentDB. O MongoDB é um banco de dados de código aberto, orientado a documentos e não relacional, projetado para armazenar, consultar e gerenciar dados de forma eficiente e flexível.

O nome "MongoDB" é uma junção das palavras "humongous" (enorme) e "database" (banco de dados), enfatizando sua capacidade de lidar com grandes volumes de dados.

Diferentemente dos bancos de dados relacionais tradicionais, que utilizam tabelas com linhas e colunas para armazenar dados, o MongoDB adota um modelo de dados flexível baseado em documentos. Os dados são representados em formato de documentos BSON (Binary JSON), que é uma extensão do formato JSON. Esses documentos podem ter uma estrutura dinâmica, permitindo a inclusão de campos diferentes para cada documento e aninhamento de subdocumentos.

O MongoDB oferece diversas funcionalidades e recursos, tais como:

- **Consultas e indexação eficientes:** o MongoDB oferece uma poderosa linguagem de consulta e suporta índices para facilitar a recuperação eficiente de dados. Ele possui recursos avançados de consulta, incluindo consultas ad hoc, consultas por campos específicos, consultas geoespaciais e agregações.
- **Alta disponibilidade e tolerância a falhas:** o MongoDB inclui recursos para garantir a alta disponibilidade dos dados. Ele suporta replicação, permitindo que os dados sejam automaticamente copiados em vários servidores para garantir redundância e tolerância a falhas. Além disso, ele oferece o recurso de sharding, que permite a distribuição dos dados em clusters para melhor desempenho e escalabilidade.
- **Transações:** a partir da versão 4.0, o MongoDB oferece suporte a transações multidocumento, permitindo que você execute operações atômicas em vários documentos. Isso é especialmente útil quando você precisa manter a consistência dos dados em situações complexas.
- **Integração com várias linguagens de programação:** o MongoDB é compatível com uma ampla variedade de linguagens de programação, como JavaScript, Python, Java, C#, Ruby e muitas outras. Isso facilita a integração do banco de dados com diferentes aplicações e ecossistemas de desenvolvimento.
- **Comunidade ativa e suporte:** o MongoDB possui uma comunidade ativa de desenvolvedores e usuários, o que significa que há uma vasta quantidade de recursos, documentação e suporte disponíveis. Além disso, a empresa por trás do MongoDB oferece serviços comerciais e suporte profissional para atender às necessidades empresariais.

A seguir você tem um exemplo de um registro de dados no banco mongoDB:

```
{
  "_id": ObjectId("615a9a677a18a43a59c6c261"),
  "cliente": "Jane Smith",
  "data_vencimento": "2023-06-10",
  "data_pagamento": "2023-06-09",
  "valor_pago": 320.50,
  "status": "Pago",
  "metodo_pagamento": "Transferência Bancária",
  "numero_fatura": "INV-789012",
  "descricao": "Taxa anual de associação",
  "observacoes": "Pagamento antecipado para a assinatura anual",
  "itens": [
    {
      "produto": "Associação",
      "quantidade": 1,
      "preco_unitario": 320.50
    },
    {
      "produto": "Workshop Adicional",
      "quantidade": 1,
      "preco_unitario": 50.00
    }
  ]
}
```

Código-fonte 1 – Exemplo de estrutura de dado JSON
Fonte: Elaborado pelo autor (2024)

Custos

No que se refere a custos, o MongoDB é distribuído sob uma licença de código aberto e oferece uma versão gratuita chamada "Community Server". Essa versão é adequada para a maioria dos casos de uso e fornece a maior parte dos recursos essenciais do MongoDB.

Além da versão gratuita, o MongoDB também oferece uma edição paga chamada "MongoDB Enterprise Server". Essa edição inclui recursos adicionais voltados para necessidades empresariais, como suporte avançado, segurança aprimorada e ferramentas de gerenciamento e monitoramento. O custo da edição Enterprise varia de acordo com o número de instâncias e o tipo de suporte necessário.

É importante observar que, embora o MongoDB ofereça uma edição paga, muitas organizações utilizam com sucesso a versão gratuita em projetos de produção. A escolha entre a versão gratuita e paga dependerá das necessidades específicas do seu projeto, dos recursos adicionais desejados e dos requisitos de suporte.

Além disso, o MongoDB possui uma plataforma chamada "MongoDB Atlas", que é um serviço de banco de dados como serviço (DBaaS) baseado na nuvem. O MongoDB Atlas permite que você implante e gerencie facilmente clusters do MongoDB em provedores de nuvem populares, como AWS, Azure e Google Cloud. Os preços do MongoDB Atlas variam de acordo com a configuração do cluster, a capacidade de armazenamento e outros fatores.

Modelagem de dados e Consultas

A modelagem de dados no MongoDB difere um pouco da modelagem em bancos de dados relacionais tradicionais devido à natureza flexível do esquema de documentos. No MongoDB, você trabalha com documentos BSON (Binary JSON) que podem ter estruturas hierárquicas e campos variáveis.

Os dados são armazenados em coleções, que são análogas às tabelas em um banco de dados relacional. Cada documento em uma coleção é um registro independente e contém todos os dados relacionados a um determinado objeto ou entidade. Para realizar consultas no MongoDB, você pode usar uma linguagem de consulta chamada MongoDB Query Language (MQL), que é semelhante à sintaxe de consulta do JavaScript.

A modelagem de dados no MongoDB envolve a identificação das entidades principais em um sistema e a definição dos atributos e relacionamentos entre essas entidades. Em vez de normalizar os dados em várias tabelas, como em um banco de dados relacional, é comum incorporar os relacionamentos diretamente nos documentos. Além disso, o MongoDB suporta índices para melhorar o desempenho das consultas e permite consultas flexíveis usando sua linguagem de consulta baseada em JSON.

O MongoDB permite que os documentos em uma coleção tenham estruturas diferentes, o que oferece flexibilidade na modelagem de dados. No entanto, isso não significa que o MongoDB não possua um esquema. Cada documento possui uma estrutura interna que pode ser considerada um "esquema dinâmico". Embora não haja restrições rígidas em relação aos campos e tipos de dados, é importante pensar em um design de esquema que melhor atenda às necessidades da aplicação.

Boas práticas

Ao realizar consultas no MongoDB, existem algumas boas práticas que podem ajudar a otimizar o desempenho e melhorar a eficiência das operações. Aqui estão algumas dicas:

- **Índices adequados:** certifique-se de criar índices apropriados para os campos usados em suas consultas mais frequentes. Os índices ajudam a acelerar a recuperação de dados, especialmente em coleções grandes. Analise suas consultas, identifique os campos usados com mais frequência e crie índices para esses campos.
- **Limitar a quantidade de dados recuperados:** use projeções para limitar os campos retornados em suas consultas. Isso reduz a quantidade de dados transferidos entre o servidor e o cliente, melhorando o desempenho geral da consulta.
- **Evite consultas excessivamente complexas:** consultas complexas podem exigir mais tempo e recursos para serem executadas. Se possível, divida consultas complexas em várias consultas menores e mais simples.
- **Utilize a agregação de forma eficiente:** o framework de agregação do MongoDB permite que você execute várias operações em um pipeline. Utilize as etapas de agregação adequadas, como \$match, \$group, \$sort e \$project para filtrar e manipular os dados conforme necessário.
- **Considere o modelo de dados e o esquema de consulta:** ao projetar o esquema de dados, leve em consideração as consultas que serão executadas com mais frequência. Modelar os dados de acordo com as consultas frequentes pode melhorar significativamente o desempenho.
- **Ajuste de configurações:** ajuste as configurações do MongoDB, como o tamanho do cache do servidor (cacheSizeGB) e o tamanho do arquivo de página (wiredTiger.engineConfig.cacheSizeGB), para melhorar o desempenho com base nas necessidades específicas do seu ambiente.
- **Monitore o desempenho:** utilize ferramentas de monitoramento, como o MongoDB Compass ou o MongoDB Cloud Manager, para analisar o desempenho das consultas e identificar possíveis gargalos ou consultas lentas.

Lembrando que as melhores práticas podem variar de acordo com a aplicação e o contexto específicos. É recomendado analisar as necessidades do seu projeto e consultar a documentação oficial do MongoDB para obter orientações mais detalhadas sobre a otimização de consultas.

Agora que já conhecemos o banco de dados NoSQL MongoDB, vamos avançar nos estudos e aprender sobre o DynamoDB.

DynamoDB é um banco de dados não relacional (NoSQL) oferecido pela Amazon Web Services (AWS) que opera totalmente na nuvem. Essa solução é reconhecida por sua escalabilidade, alta disponibilidade, gerenciamento simplificado, abordagem serverless (sem servidor) e robustas práticas de segurança. Amplamente empregado em diversos projetos, o DynamoDB se destaca especialmente em cenários que demandam processamento eficiente de grandes volumes de dados.

Principais Características do DynamoDB

- **Escalabilidade automática:** o DynamoDB é projetado para escalar automaticamente conforme a demanda, adaptando-se dinamicamente a aumentos ou reduções no tráfego de dados. Essa característica é essencial para aplicações que experimentam variações significativas na carga de trabalho.
- **Alta disponibilidade:** oferece alta disponibilidade, garantindo que as operações de leitura e gravação possam ser executadas sem interrupções, mesmo em ambientes distribuídos e sob cargas intensas.
- **Gerenciamento simplificado:** o DynamoDB é um serviço totalmente gerenciado, o que significa que a AWS cuida de tarefas operacionais, como provisionamento de hardware, manutenção e backup, permitindo que especialistas em desenvolvimento se concentrem mais na lógica de aplicação.
- **Serverless (sem servidor):** como serviço serverless, o DynamoDB elimina a necessidade de gerenciar a infraestrutura subjacente. Isso resulta em maior flexibilidade, eficiência de custos e facilidade de

implementação, pois os usuários não precisam se preocupar com a configuração ou manutenção de servidores.

- **Segurança robusta:** o DynamoDB fornece recursos avançados de segurança, incluindo controle de acesso granular por meio do AWS Identity and Access Management (IAM), criptografia em repouso e em trânsito, além de outras medidas para proteger os dados do usuário.

EXEMPLO

O QUE VOCÊ VIU NESTA AULA?

Nessa aula, aprendemos sobre os bancos de dados NoSQL. Isso nos permitiu explorar algumas das boas práticas para esses bancos de dados, além de demonstrar os principais benefícios.

Além disso, nos aprofundamos no MongoDB, o banco de dados de documentos mais popular da atualidade. Exploramos suas principais características, questões relacionadas a custos e modelagem de dados e, por fim, praticamos com exemplos de criação de coleções, inclusão, alteração, exclusão de dados e, é claro, consultas de dados nos documentos.

REFERÊNCIAS

BOAGLIO, F. **MongoDB**: construa novas aplicações com novas tecnologias. São Paulo: Casa do Código, 2020.

PANIZ, D. **NoSQL: Como armazenar os dados de uma aplicação moderna**. São Paulo: Casa do Código, 2016.

SADALAGE, PJ; FOWLER, M. **NoSQL essencial**: um guia conciso para o mundo emergente da persistência poliglota. São Paulo: Novatec Editora, 2019.

SCYLLADB. **Introduction to DynamoDB**. [s.d.]. Disponível em: <https://www.scylladb.com/learn/dynamodb/introduction-to-dynamodb/>. Acesso em: 19 mar 2024.

ADRIANO, T. S. **[Dica rápida]Conhecendo o Relational Migrator**. 2024. Disponível em: <https://programadriano.medium.com/dica-r%C3%A1pida-conhecendo-o-relational-migrator-956dddc7e5e>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comandos básicos Docker**. 2017. Disponível em: <https://programadriano.medium.com/principais-comandos-docker-f9b02e6944cd>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comparando os termos utilizados no NoSQL com SQL**. 2024. Disponível em: <https://medium.com/xp-inc/comparando-os-termos-utilizados-no-nosql-com-sql-e862788e2374>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Trabalhando com AWS em um ambiente localhost**. 2024. Disponível em: <https://dev.to/programadriano/trabalhando-com-aws-em-um-ambiente-localhost-20i8>. Acesso em: 14 jun. 2024.

AMAZON. **Download NoSQL Workbench for DynamoDB**. 2024. Disponível em: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html>. Acesso em: 14 jun. 2024.

AMAZON.COM. **Nível gratuito do Amazon RDS**. 2024. Disponível em: <https://aws.amazon.com/pt/rds/free/>. Acesso em: 14 jun. 2024.

DIAS, L. A. Grafos, teoria e aplicações. 2019. Disponível em: <https://medium.com/xp-inc/grafos-teoria-e-aplica%C3%A7%C3%B5es-2a87444df855>. Acesso em: 14 jun. 2024.

GITHUB. **Criação do container postgres**. 2024. Disponível em: <https://gist.github.com/programadriano/2171fe63e6a2c1b08985bccf263fe980>. Acesso em: 14 jun. 2024.

GITHUB. **Docker + Neo4J**. 2024. Disponível em: <https://gist.github.com/programadriano/0224c1a1d389af25177eb397d912dbe7>. Acesso em: 14 jun. 2024.

GITHUB. **Estrutura**. 2024. Disponível em: https://app.diagrams.net/#G1_-jdpuomwUwU3KD01rSRGepNwA3kYA9X%7B%22pageld%22%3A%22apt8INFUySjbP2altylb%22%7D. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (1)**. 2024. Disponível em: <https://gist.github.com/programadriano/01508490e0f53745f2778a045e1cc070>. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (2)**. 2024. Disponível em: <https://gist.github.com/programadriano/e22931ff2bc032dcc68498bc3597d1a7>. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (3)**. 2024. Disponível em: <https://gist.github.com/programadriano/8c8e7ea613acc4e854a7f2ffe9f44111>. Acesso em: 14 jun. 2024.

GITHUB. **Query X Scan**. 2024. Disponível em: <https://gist.github.com/programadriano/8169d065da7e0cf8494435ca8caca70>. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (1)**. 2024. Disponível em: <https://gist.github.com/programadriano/e7c314539571c811a48940deca8c8491>. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (2)**. 2024. Disponível em: <https://gist.github.com/programadriano/c70281293a5720f1ab8719d65fe5e410>. Acesso em: 14 jun. 2024.

IBM. **O que é o Docker?** 2024. Disponível em: <https://www.ibm.com/br-pt/topics/docker>. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **Abrir e mostrar o ambiente rodando**. 2024. Disponível em: <https://app.localstack.cloud/dashboard>. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **LocalStack Desktop**. 2024. Disponível em: <https://app.localstack.cloud/download>. Acesso em: 14 jun. 2024.

MIRO. **Fluxo Miro**. 2024. Disponível em: https://miro.com/welcomeonboard/Mzh1bEhNczFWeWtFeFpUS0tXeIJIQ1RjZnJjZHhmbmhFU0kyd3RjenpjZVhTdVZqajJPN3NxdFpuR3lkMWc5NHwzNDU4NzY0NTY0Ng5NTc4MjlfDI=?share_link_id=113865211181. Acesso em: 14 jun. 2024.

MONGODB. **MongoDB Relational Migrator Download**. 2024. Disponível em: <https://www.mongodb.com/try/download/relational-migrator>. Acesso em: 14 jun. 2024.

NEO4J. **Exploring LinkedIn in Neo4j**. 2013. Disponível em: <https://neo4j.com/blog/exploring-linkedin-in-neo4j/>. Acesso em: 14 jun. 2024.

NEO4J. **Use Cases: Social Media and Social Network Graphs**. 2024. Disponível em: <https://neo4j.com/use-cases/social-network/>. Acesso em: 14 jun. 2024.

PGADMIN.ORG. **PgAdmin4.** 2024. Disponível em:
<https://www.pgadmin.org/download/pgadmin-4-windows/>. Acesso em: 14 jun. 2024.

EMANIP

PALAVRAS-CHAVE

Palavras-chave: Bancos de dados de documentos. NoSQL. Documentos JSON. MongoDB. DynamoDB.

EXEMPLO



POSTECH