

Grupo: Edson Lima, Gustavo Nunes, Gabriel Rocha e Layla Couto

TRABALHO BANCO DE DADOS II

Banco de dados não convencionais

CONCEITO

O banco de dados não relacional surgiu como uma alternativa para situações em que há uma quantidade exorbitante de dados para armazenar, onde a estruturação em tabelas se torna mais difícil.

Um banco de dados não relacional pode ser definido como um banco de dados que não usa o sistema de tabelas, linhas e colunas encontrado na maioria dos sistemas de bancos de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado.

DIFERENÇA ENTRE TIPO DE BANCOS

O **banco de dados relacional** sempre irá fornecer dados íntegros e imutáveis, garantindo um controle transacional consistente. Além disso, seu esquema é rígido, sendo possível atribuir campos e estabelecer se o dado de uma coluna é nulo ou não nulo.

Já o **banco de dados não relacional**, que representa diversos tipos de bancos de dados, não exige a rigidez de esquemas para armazenar os dados ou seja, ele não limita os campos, diferente das colunas do SQL. Além disso, é possível adicionar novas propriedades, sem a preocupação com o impacto nas demais informações já armazenadas.

Existem também os **bancos de dados orientados a objetos**, onde as informações são organizadas em blocos de informações com identificadores.

Os **bancos de dados autônomos** são gerenciados por machine learning, na nuvem. Desse modo, os backups, segurança, consultas e demais tarefas rotineiras ficam sob responsabilidade da inteligência artificial, sendo ideal para situações em que há um alto fluxo de dados e as pessoas envolvidas podem facilmente se perder ao realizar as tarefas manualmente.

Bancos de dados OLTP realizam o processamento de dados em transações online, sendo ideais para lidar com informações de organizações, como pagamentos e gerenciamento de pedidos.

Os **bancos de dados distribuídos** são compostos por uma rede de nós, em que cada nó é um computador em um local diferente. Para se comunicarem, as máquinas conectadas ao sistema utilizam a rede, sendo seu principal diferencial a distribuição de informações.

Existem também os **bancos de dados JSON**, que se trata basicamente de uma base de dados que armazena e realiza consultas a documentos JSON. O JSON é um elemento textual, mas também pode receber outros valores, como um BLOB, usado para armazenar imagens. Para gerenciar e manipular esse banco podem ser usadas tabelas.

Já os **data warehouses** permitem a integração de dados de diferentes fontes. Além de manter os dados atuais, também é possível consultar um histórico. Geralmente, o Data warehouse é indicado para criação de relatórios de análise, sendo possível compilar grande quantidade de dados.

Banco de dados gráficos armazenam estruturas de dados complexas, que seriam incompatíveis em uma base tradicional. Esse tipo de sistema é geralmente usado em redes sociais, pois é ideal para lidar com dados altamente interconectados.

Os **bancos de dados de código aberto** tratam de qualquer base de dados que possa ser distribuída e reutilizada, fornecendo às pessoas desenvolvedoras as condições necessárias para a criação de novas tecnologias, que utilizam o código aberto do sistema como molde.

Os **banco de dados na nuvem** são uma base de dados que podem ser acessadas por meio de uma plataforma na internet, significando que as informações não são armazenadas em um servidor local, mas sim em servidores remotos, com todo o sistema funcionando como um serviço.

Por último, existe o **banco de dados multi modelo**, que reúne uma combinação de banco de dados em um único back-end, o tornando extremamente flexível em relação aos dados que podem ser armazenados e as estruturas que podem ser montadas.

BANCO NoSQL

O termo foi utilizado pela primeira vez em 1998 por Carlo Strozzi, ao falar sobre um banco de dados não relacionais de código aberto. Com o crescimento do uso da internet e das soluções digitais, tornou-se cada vez mais necessário encontrar maneiras inteligentes de gerenciar os bancos de dados das empresas.

O termo foi novamente utilizado em 2006, quando a empresa Google publicou um artigo sobre o armazenamento de dados. Já em 2009, um colaborador do Rackspace organizou um evento para tratar de bancos de dados open source distribuídos e novamente citou o termo NoSQL.

Os bancos de dados NoSQL (não-relacionais) surgiram a partir da necessidade de utilizar soluções mais fáceis de ampliar ou reduzir a infraestrutura em que há uma quantidade exorbitante de dados para armazenar, onde a estruturação em tabelas se torna mais difícil conforme as empresas crescem. O termo 'NoSQL' se refere a tipos não relacionais de bancos de dados que não usam o SQL para consultas. Esses bancos de dados armazenam dados em um formato diferente das tabelas relacionais.

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.

No banco de dados não relacional (NoSQL), os esquemas não são necessários. Eles representam qualquer banco ligado à Big Data e não seguem o modelo relacional fornecido pelos sistemas tradicionais de gerenciamento de dados.

Os bancos de dados NoSQL são ideais para muitos aplicativos modernos, como dispositivos móveis, Web e jogos, que exigem bancos de dados flexíveis, escaláveis, de alta performance e altamente funcionais para proporcionar ótimas experiências aos usuários. Por ter essas características, são a escolha preferida dos desenvolvedores, pois aceitam um paradigma de desenvolvimento ágil, adaptando-se rapidamente aos requisitos em constante mudança, ajudando a gerenciar os novos desafios da diversidade crescente de tipos de dados e modelos e de serem altamente eficazes ao processar dados imprevisíveis. Permitem que os dados sejam armazenados de maneiras mais intuitivas e fáceis de entender, ou

mais próximas da maneira como os dados são usados pelos aplicativos - com menos transformações necessárias ao armazenar ou recuperar usando APIs no estilo NoSQL. Além disso, os bancos de dados NoSQL podem aproveitar ao máximo a nuvem para oferecer tempo de inatividade zero.

Tipos de bancos de dados NoSQL

- **Chave-valor:** Os armazenamentos de chave-valor fazem o pareamento de chaves e valores usando uma tabela de hash. Os tipos de chave-valor são melhores quando uma chave é conhecida e o valor associado dela é desconhecido.
- **Documento:** Os bancos de dados de documentos ampliam o conceito do banco de dados chave-valor organizando documentos inteiros em grupos chamados coleções. Eles são compatíveis com os pares chave-valor aninhados e permitem consultas em qualquer atributo em um documento.
- **Gráfico:** Os bancos de dados de grafo usam um modelo baseado em nós e bordas para representar dados interconectados – como relações entre pessoas em uma rede social – e oferecem armazenamento simplificado e navegação por meio de relações complexas.
- **Colunar:** Bancos de dados colunares, de coluna larga ou de famílias de colunas armazenam dados de modo eficiente, consultam linhas de dados esparsos e são vantajosos ao consultar em colunas específicas no banco de dados.

Principais bancos de dados não relacionais utilizados no mercado

- **Redis:** O Redis é o banco de dados NoSQL de chave-valor mais utilizado em todo o mundo. Ele vincula um valor a uma chave na sua estrutura, o que facilita o armazenamento e a busca desses dados. Por isso, é muito utilizado pelos desenvolvedores.
- **Memcached:** também faz o armazenamento com chave-valor e usa um cache de memória distribuída. Geralmente, é utilizado para criar

sites dinâmicos, pois acelera a abertura das páginas e diminui as buscas de dados de fontes externas.

- **Cassandra:** Este banco de dados NoSQL foi desenvolvido no Facebook. Ele usa um banco de dados descentralizado, em que os dados são armazenados em vários datacenters. Ele é otimizado para cluster e fornece baixa latência em suas atualizações.
- **Hbase:** O Hbase é um banco de dados que utiliza conjunto de linhas e colunas para armazenar as informações. Ele é utilizado em diferentes plataformas como o LinkedIn, Facebook e Spotify.
- **Amazon DynamoDB:** Este é um banco de dados NoSQL em nuvem, disponibilizado pela Amazon Web Service. Ele tem baixa latência, é rápido e flexível, sendo o modelo ideal para aplicações móveis, jogos na web e soluções com internet das coisas. Ele ainda apresenta alto desempenho e escalabilidade automática, características imprescindíveis para negócios que precisam crescer com eficiência.
- **Neo4j:** O Neo4j é um banco de dados não-relacional que se baseia em grafos (arestas que se relacionam aos nodes). Ele é uma implementação de código aberto e pode ser útil para casos de mineração de dados e reconhecimento de padrões.
- **MongoDB:** Este também é um banco de dados de código aberto com alta performance. Ele é aceito em diferentes sistemas operacionais e tem como característica ser orientado a documentos. Sendo assim, ele armazena todas as informações relevantes em um documento e utiliza sistemas avançados de agrupamento e filtragem. Diferentes plataformas e linguagens possuem suporte ao MongoDB, entre elas estão o Java, JavaScript, PHP, Python e Ruby. Os principais exemplos de empresas que usam o MongoDB são: o site Globo.com, MailBox, MTV e Pearson Education.

Banco MongoDB

É um banco de dados opensource, de alta performance e flexível, sendo considerado o principal banco de dados NoSQL. Os banco de dados NoSQL apresentam algumas vantagens sobre os outros tipos, principalmente quando precisamos de escalabilidade, flexibilidade, bom desempenho e facilidade para consultas.

O MongoDB é orientado a documentos, ou seja, os dados são armazenados como documentos, ao contrário de bancos de dados de modelo relacional, onde trabalhamos com registros em linhas e colunas. Os documentos podem ser descritos como dados no formato de chave-valor, no caso, utilizando o formato JSON (JavaScript Object Notation).

Organizações de todos os tamanhos estão usando o MongoDB pois ele permite que as empresas sejam mais ágeis e escaláveis, permitindo que os esquemas mudem rapidamente à medida que os aplicativos evoluem, sempre fornecendo as funcionalidades que os desenvolvedores esperam dos bancos de dados tradicionais.

Foi desenvolvido para oferecer escalabilidade, desempenho e alta disponibilidade, desde a implantação de um único servidor até grandes arquiteturas complexas de vários centros de dados. A replicação nativa do MongoDB e a tolerância automática a falhas oferecem confiabilidade e flexibilidade operacional em toda a empresa.

Características

O MongoDB possui algumas características que o tornam uma das melhores opções para incorporar como banco de dados no desenvolvimento de aplicações.

- **Sintaxe para consultas:** permite que você faça consultas das mais simples às mais complexas, podendo obter todos os tipos de informações;
- **Indexação:** é um conceito similar ao dos banco de dados relacionais, podendo criar índices, tendo um grande impacto no desempenho de suas consultas à medida que o banco de dados aumenta de tamanho;

- **Escalabilidade horizontal:** a escalabilidade se refere sobre a necessidade do seu banco de dados ser adaptável, aumentando seu tamanho em função das informações que são armazenadas pelos usuários. A escalabilidade horizontal envolve a divisão do conjunto de dados do sistema e a carga em vários servidores, adicionando servidores adicionais para aumentar a capacidade, aumentando a disponibilidade da sua aplicação, de acordo com o volume dos dados ou o número de acessos ao banco de dados;
- **Permite executar consultas executando código JavaScript:** se você é um desenvolvedor JavaScript ou conhece essa linguagem pode tirar proveito de todo o conhecimento de certas funções que essa linguagem oferece para utilizá-la no MongoDB.

Vantagens

- Ideal para ambientes com poucos recursos, onde qualquer servidor pode atender você;
- Custo baixo: É uma ferramenta com custo baixo, pois por ser de código aberto, você não precisa pagar uma licença;
- Complemento perfeito para o JavaScript.

Desvantagens

- Não se adequa para aplicações com transações complexas. Neste caso os banco de dados relacionais talvez sejam mais ideais;
- Ainda é uma tecnologia nova, apesar de ser amplamente utilizada;
- Não utiliza 'joins' para consultas, ou seja, aquelas consultas nas quais combinamos tabelas diferentes. No MongoDB temos que executar esse tipo de consulta de maneira diferente.

Por que utilizar o MongoDB?

- **Flexibilidade:** oferece uma grande flexibilidade devido a seu banco de dados que não segue nenhum esquema, o que acaba o deixando muito flexível,

embora possa parecer uma desvantagem, é um recurso que o diferencia de outros tipos de banco de dados.

- **Alta disponibilidade:** se refere ao alto volume de acesso. Como esses acessos poderão aumentar consideravelmente, isso pode afetar o desempenho do seu banco de dados e nesse sentido, o MongoDB tem capacidade para se adaptar a essas necessidades, permitindo uma grande ou alta disponibilidade.
- **Projetos que envolvem Big Data:** MongoDB se encaixa muito bem neste tipo de projeto.
- **Documentação oficial:** a documentação oficial é muito boa, portanto, durante o desenvolvimento do seu projeto, você pode utilizá-la como um ótimo recurso para melhorar suas operações.

Situações vantajosas de usar um banco NoSQL

Por conter certa flexibilidade, a utilização de um banco NoSQL pode vir a ser por diversas razões, mas assim como todas as situações podem apresentar prós e contras, tem que analisar caso a caso.

No caso do NoSQL, pode se tornar útil e atrativo aos olhos de quem utiliza desenvolvimento ágil, pois dessa forma elimina a complexidade de mudança dos bancos de dados, assim tendo um bom suporte e adaptações rápidas.

Outro fator que pode trazer vantagem, é a grande eficiência de busca por conta de sua arquitetura, assim podendo fazer buscas ainda mais rápidas para certa quantidade de dados.

É um fator, que pode se tornar eficiente ainda mais para uma empresa, é os custos a se manter com um banco de dados relacional para um não relacional. Com os bancos relacionais, é preciso manter mega computadores, mainframes e suportes muito caros, e dependendo do banco ainda o pagamento do mesmo.

Com o NoSQL, isso é reduzido de forma drástica, visto que como grande maioria é open source, o seu custo é muito abaixo, pois não necessita especificamente de um mainframe, pois pode ser desenvolvido para trabalhar em ambientes distribuídos.

Referências

<https://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>

<https://www.digitalhouse.com/br/blog/banco-de-dados-relacional-e-nao-relacional/>

<https://www.oracle.com/br/database/nosql/what-is-nosql/>

<https://azure.microsoft.com/pt-br/overview/nosql-database/>

<https://blog.saphir.com.br/conheca-os-principais-bancos-de-dados-nosql-nao-relacionais/>

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

<https://blog.betrybe.com/tecnologia/bancos-de-dados/>

<https://www.treinaweb.com.br/blog/o-que-e-mongodb>

<https://www.devmedia.com.br/introducao-ao-mongodb/30792>

<https://dicasdeprogramacao.com.br/6-motivos-para-usar-bancos-de-dados-nosql/>