

Pedro Sergio Gomes Quinderé

## **Questões para Pesquisa e Resposta:**

### **1. Definição e Características:**

a) O que caracteriza um banco de dados não-relacional?

Um banco de dados não-relacional (NoSQL) é caracterizado pela falta de um esquema fixo de tabelas e pela flexibilidade na forma como os dados são armazenados, usando modelos como pares chave/valor, documentos (JSON), colunas ou grafos. Ele prioriza a escalabilidade horizontal, agilidade e velocidade, o que o torna ideal para lidar com grandes volumes de dados não estruturados ou semiestruturados, como os encontrados em Big Data e aplicações web em tempo real.

b) Quais são as vantagens desse tipo de banco em relação aos bancos relacionais?

As vantagens de um banco de dados relacional incluem forte consistência de dados através das propriedades ACID, facilidade de uso e consultas com a linguagem SQL padronizada, redução de redundância através da normalização, e maior maturidade e suporte da comunidade.

c) Cite ao menos 3 situações práticas em que um banco de dados não-relacional é mais indicado.

Aplicações de redes sociais; Internet das Coisas (IoT); e, Catálogos de produtos e e-commerce.

## 2. Tipos de Bancos de Dados Não-Relacionais:

a) Explique as diferenças entre os quatro principais tipos:

- Orientados a Documentos.
- Chave-Valor.
- Colunas Amplas.
- Grafos.

Bancos de dados orientados a documentos organizam dados em documentos semiestruturados (como JSON) e são flexíveis para dados com esquemas variáveis. Já os que trabalham com estrutura chave-valor armazenam dados simples como um par chave-valor, ótimos para sessões e caches. Por sua vez, os bancos colunares armazenam dados em colunas em vez de linhas, ideais para consultas em grandes conjuntos de dados e alta velocidade de escrita. Por fim, os de grafo usam nós e arestas para representar dados e suas relações complexas, sendo perfeitos para redes sociais e análise de conexões.

b) Dê exemplos de ferramentas populares para cada tipo.

**Orientado a documentos:** um exemplo de banco de dados não-relacional orientado a documentos é o MongoDB, que armazena dados em formato de documentos, como no estilo JSON. Outros exemplos incluem o ArangoDB e o CouchDB.

**Chave-Valor:** exemplos de bancos de dados não-relacionais chave-valor incluem Redis, Amazon DynamoDB e Memcached. Eles armazenam dados em pares de chave e valor, sendo ideais para cenários que exigem acesso rápido e escalável a dados, como cache e carrinhos de compras online.

**Colunas Amplas:** exemplos de bancos de dados não-relacionais com colunas amplas incluem Apache Cassandra, Apache HBase, ScyllaDB e Google Bigtable. Esses sistemas são projetados para

armazenar dados em famílias de colunas, o que permite que cada linha tenha um conjunto de colunas diferente, ideal para lidar com grandes volumes de dados com esquemas variados e flexíveis.

**Grafos:** exemplos de bancos de dados não-relacionais do tipo grafo incluem Neo4j, Amazon Neptune e ArangoDB. Eles são projetados para armazenar e consultar dados com relações complexas, organizando-os em nós (vértices) e conexões (arestas).

### 3. Comparação Prática:

- a) Qual a principal diferença entre o modelo relacional e o modelo de grafos?

A principal diferença é que o modelo relacional prioriza entidades de dados, armazenando-os em tabelas, enquanto o modelo de grafos prioriza os relacionamentos, representando os dados como uma rede de nós e arestas. O modelo relacional é ideal para dados estruturados e consultas que utilizam principalmente junções, já o modelo de grafos se destaca em lidar com relacionamentos complexos e dinâmicos.

- b) Em um cenário de redes sociais, qual tipo de banco de dados seria mais adequado? Justifique.

Em um cenário de redes sociais, o banco de dados NoSQL é o mais adequado, especificamente o banco de dados de grafos, devido à natureza relacional e à necessidade de escalabilidade, flexibilidade e desempenho para lidar com um grande volume de dados diversos e interconexões complexas. Um sistema moderno de rede social, no entanto, utilizaria uma abordagem híbrida, combinando diferentes tipos de bancos de dados para tarefas específicas. Isso se justifica em função da escalabilidade, flexibilidade e desempenho.

#### 4. Escalabilidade e Flexibilidade:

- a) Explique como a escalabilidade horizontal funciona em bancos não-relacionais.

A escalabilidade horizontal em bancos não-relacionais (NoSQL) funciona distribuindo a carga de trabalho e os dados por vários servidores (nós) em um cluster, em vez de aumentar a potência de um único servidor. Isso é possível devido à arquitetura distribuída, esquemas flexíveis e técnicas como fragmentação (sharding) e replicação. Essa abordagem permite que o sistema cresça adicionando mais máquinas, tornando-o mais escalável, tolerante a falhas e econômico para lidar com grandes volumes de dados.

- b) Por que a flexibilidade de schema é importante em bancos orientados a documentos?

Porque permite que cada registro (documento) tenha uma estrutura diferente, facilitando a evolução rápida de aplicações e a adaptação a dados que mudam frequentemente sem a necessidade de um esquema rígido e pré-definido. Isso agiliza o desenvolvimento, permite armazenar dados que não se encaixam facilmente em um modelo relacional e facilita a incorporação de novos campos ou a remoção de campos existentes sem complexas migrações de esquema.

## 5. Estudo de Caso:

- a) Pesquise um caso de uso real de uma empresa que utiliza bancos de dados não-relacionais.

O LinkedIn é um caso de uso real de banco de dados não-relacional, utilizando o Apache Cassandra (um banco de dados colunar) para gerenciar os enormes volumes de dados de seus usuários de forma escalável e com alta disponibilidade. Isso permite que a plataforma processe e armazene eficientemente informações como conexões, postagens e perfis, lidando com dados de diversos formatos e em grande escala.

- b) Descreva como esse banco é utilizado e quais foram os benefícios obtidos.

Essa abordagem permite o armazenamento de dados flexíveis, como descrições de produtos com diferentes atributos, e o uso de modelos de dados como documentos e chave-valor para armazenar perfis de clientes e histórico de navegação de forma rápida e escalável. Os benefícios incluem melhor desempenho em operações de leitura/escrita e a capacidade de escalar a infraestrutura para atender a um volume massivo de dados e usuários.