

Questões para Pesquisa e Resposta:

1. Definição e Características:

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

O banco flexível, que armazena as informações sem uma estrutura fixa ou semi organizados os dados.

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

Ele é escalável e robusto, capaz de guardar grandes volumes de dados e ágil para processar os dados.

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

Bancos de dados não-relacionais são a melhor escolha para lidar com um volume gigante de dados que chegam rápido, como de sensores em IoT. Eles também são perfeitos para catálogos de produtos ou conteúdos variados, já que cada item pode ter informações diferentes sem quebrar a estrutura. Além disso, são essenciais para aplicações que precisam de velocidade máxima, como gerenciar sessões de usuários ou carrinhos de compra em tempo real.

2. Tipos de Bancos de Dados Não-Relacionais:

a) Explique as diferenças entre os quatro principais tipos:

- Orientados a Documentos. Armazena dados no formato Json ou Bson. Ex de ferramenta MongoDB
- Chave-Valor. Armazena cache e sessões, Ex de ferramenta Redis.

- Colunas Amplas. Organiza as informações por colunas, ideal para análise de grandes volumes de dados. , Ex de ferramenta Cassandra.
 - Grafos. Armazenados como nós e arestas, ideal para modelar relacionamentos complexos, Ex de ferramenta Neo4j.
- b) Dê exemplos de ferramentas populares para cada tipo.

3. Comparação Prática:

- a) Qual a principal diferença entre o modelo relacional e o modelo de grafos? Modelo relacional segue uma estrutura linear, com tabelas multidimensionais e o modelo de grafo armazena as informações em forma de nós, facilitando as conexões complexas.
- b) Em um cenário de redes sociais, qual tipo de banco de dados seria mais adequado? Justifique. Modelo de grafo, pelo modo de armazenamento vai gerando informações neste formato gerando subdivisões linkadas a uma interação principal

4. Escalabilidade e Flexibilidade:

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

Ela traz escalabilidade na armazenação dos dados de forma em múltiplos servidores , trazendo um caminho flexível e econômico para o crescimento.

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

No lugar de forçar os dados a se encaixarem em uma estrutura pré-definida, os dados são definidos pela própria aplicação no momento da escrita dos dados. Tem capacidade de lidar com dados polimórficos e semi-estruturados.

5. Estudo de Caso:

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

Caixa Econômica Federal

A Caixa Econômica Federal, como o principal banco público e agente de políticas sociais do Governo Federal, foi designada para a tarefa monumental de operacionalizar o pagamento do Auxílio Emergencial em 2020. A missão era cadastrar e pagar dezenas de milhões de brasileiros (muitos deles sem conta em banco) em um prazo curíssimo, em meio a uma pandemia global. O desafio era um dos maiores da história da tecnologia bancária mundial.

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

- Cadastro Massivo de Cidadãos:** A primeira etapa era cadastrar milhões de pessoas. Os dados dos cidadãos vinham de diversas fontes e com qualidades distintas. Um banco de dados orientado a documentos (como MongoDB) é ideal para esse cenário, pois seu schema flexível permite armazenar perfis de usuários com informações variadas sem a necessidade de uma estrutura rígida e pré-definida. Um novo cadastro podia ser salvo como um "documento" completo do cidadão.
- Gerenciamento de Filas Virtuais e Sessões de Usuário:** Um dos maiores desafios foi lidar com o acesso simultâneo de milhões de pessoas ao aplicativo Caixa Tem. Para evitar o colapso do sistema, foram implementadas as famosas "filas virtuais". Esse tipo de funcionalidade, que precisa gerenciar o estado de milhões de sessões em tempo real, é um caso de uso perfeito para bancos de dados NoSQL de chave-valor e em memória (como o Redis). Eles conseguem lidar com um volume altíssimo de leituras e escritas de dados pequenos e temporários com latência extremamente baixa.
- Processamento e Análise de Elegibilidade:** Os dados dos cadastrados precisavam ser cruzados com múltiplas bases do governo

(como Receita Federal, INSS, etc.) para determinar a elegibilidade.

Plataformas de Big Data, que frequentemente utilizam ecossistemas

NoSQL (como o Hadoop/HBase), são projetadas para processar e analisar volumes gigantescos de dados de forma distribuída e eficiente.

Benefícios Obtidos com a Adoção do NoSQL:

- **Velocidade de Desenvolvimento e Implementação:** Este foi o benefício mais crítico. A Caixa e seus parceiros precisavam construir e lançar uma solução de escala nacional em **semanas**. A flexibilidade dos bancos de dados NoSQL permitiu que as equipes de desenvolvimento construissem e adaptassem a aplicação rapidamente, sem as amarras de migrações lentas e complexas de um banco de dados relacional. Era possível adicionar novos campos e adaptar o sistema conforme as regras do auxílio eram definidas e ajustadas.
- **Escalabilidade Sob Demanda Extrema:** A demanda foi astronômica e imprevisível. Em um único dia, milhões de pessoas tentavam acessar o sistema ao mesmo tempo. A arquitetura precisava escalar horizontalmente de forma elástica. Os bancos de dados NoSQL são nativamente projetados para isso: quando a carga aumentava, mais servidores eram adicionados ao cluster na nuvem para distribuir o trabalho e suportar o tráfego, garantindo que o serviço não saísse do ar.
- **Alta Disponibilidade:** O sistema não podia parar. Milhões de famílias dependiam do auxílio para sobreviver. Os bancos de dados NoSQL modernos possuem mecanismos de replicação e tolerância a falhas. Se um servidor ou um data center ficasse indisponível, o sistema continuaria operando a partir das réplicas, garantindo a continuidade do serviço essencial.
- **Custo-Eficiência em Escala:** Construir um sistema dessa magnitude com tecnologia tradicional seria proibitivamente caro. A escalabilidade horizontal em nuvem, utilizando servidores mais comuns (commodities), permitiu à Caixa gerenciar os custos de forma mais eficiente, pagando pela capacidade computacional conforme a necessidade.

