

FECAP



Questionário de Revisão - Big Data

Técnico em Inteligência Artificial | Turma: 2IAB

Professor: Luis Fernando dos Santos Pires

Aluno: Carlos Eduardo de Lima

02/11/25



1. Em termos de modelagem de dados, como a prioridade na representação de relações complexas em um banco de Grafos difere da prioridade no agrupamento de todos os dados de uma entidade em um banco Documental?

Em um banco de dados de Grafos, a representação visual e intuitiva dos relacionamentos entre os dados tem uma função principal: tornar essa visualização muito mais direta, intuitiva e completa.

Já o caso de banco de dados Documentais, a representação dos relacionamentos é feita em um mesmo documento, por meio de estruturas alinhadas ou referências simples a outros dados, sendo menos eficientes quando o assunto é relacionamentos complexos, dado que as prioridades do banco de dados de Documentos não é representar o relacionamento dos dados em si.

2. Descreva a principal otimização de leitura que um banco Colunar oferece em consultas analíticas (OLAP), e compare-a com o método de recuperação de dados de um banco Documental.

A principal otimização de leitura de consultas analíticas (OLAP) está em armazenar e ler apenas colunas necessárias para a consulta, ao invés de todas as linhas dos registros presentes. Os dados são gravados coluna por coluna, ao invés de linha por linha, sendo armazenado de forma contígua dentro do disco, reduzindo o volume dos dados lidos no armazenamento, além de aumentar a compreensão e velocidade.

Já no caso do modelo de recuperação dos dados de um banco Documental, a leitura acaba sendo feita de modo a ler todo o documento, já que todas as entidades estão agrupadas em um único registro (JSON, por exemplo). Obviamente, sua comparação acaba sendo clara, já que o diferencial de um banco Colunar (no quesito consultas) está em armazenar informações de forma independente (coluna por coluna), tornando o processo analítico mais eficaz, rápido e compreensível.



3. A ausência de "joins" é uma limitação de vários modelos NoSQL. Qual modelo supera isso de forma nativa e intuitiva, usando elementos como nós (vértices) e arestas (bordas) para representar as conexões?

É o caso de bancos de dados NoSQL baseado a grafos, cuja sua representação está diretamente ligada a utilização de vértices para representar os dados do banco de dados e arestas que interligam esses vértices, representando suas conexões.

4. Em que o conceito de Família de Colunas em um banco Colunar se assemelha e difere de uma Coleção em um banco Documental?

A semelhança entre eles é que, tanto a Família de Colunas (no Banco Colunar) quanto a Família de Coleção (do banco de Documentos) serve para agrupar os dados de forma lógica, sendo equivalentes a tabelas de bancos relacionais (de certo modo), além de representar uma unidade organizacional que contém registros (linhas ou documentos). Exemplificando, ambos agrupam entidades do mesmo tipo, como “usuários”, “produto”, “id”, “pedidos”, “cpf”, entre outros.

Já o que destoa ambos é que, enquanto a família de colunas organiza os dados a partir de colunas (priorizando eficiência de consultas), a coleção está preocupada em organizar os dados em documentos completos, visando a praticidade no armazenamento e recuperação de entidades inteiras.



5. Explique por que a Escalabilidade Horizontal é o principal foco para bancos Colunares e Chave-Valor, e porque a Escalabilidade Vertical tende a ser mais eficaz para bancos de Grafos.

A escalabilidade horizontal está na capacidade de transpor as informações de um determinado banco de dados para outros servidores, mas mantendo a relação entre esses dados, mantendo suas conexões, padrões e variáveis. Pensando dessa forma, esse método de aumento de servidores é ideal para bancos de dados colunares e de chave-valor pelo fato de que a estrutura desses dados armazenados é consistente, além de conter informações fundamentais como relacionamento e semelhanças bem estruturados e definidos.

Isso muda de cenário quando pensamos em bancos de dados a base de Grafos, já que a representação visual e intuitiva dos dados se conectando limite sua capacidade de se expandir para outros servidores e mantendo suas conexões, já que isso exige uma comunicação muito maior entre os servidores, tornando o processo de consulta e análise muito mais lento e frágil. Devido a isso, é preferível o uso da Escalabilidade Vertical, que basicamente se refere a prática de aumentar o armazenamento do servidor em si, ao invés de compartilhar seus dados com outros servidores externos.

6. Para um sistema de e-commerce que precisa de alta escalabilidade para gerenciar catálogos com diferentes atributos de produtos, qual modelo (Documental ou Chave-Valor) é mais adequado e por quê?

O modelo Documental costuma ser o mais utilizado. Sua capacidade de armazenar informações distintas de acordo com as necessidades específicas de certos produtos, além da capacidade de elaborar consultas a partir de colunas ou valores específicos torna esse banco de dados muito mais utilizado para e-commerce.



7. Qual modelo NoSQL é mais eficiente para um sistema de monitoramento de IoT que exige alta taxa de escrita e leitura de séries temporais de dados, e qual característica estrutural apoia essa eficiência?

Para IoT, bancos de dados colunares são mais eficientes para o monitoramento de informações praticamente em tempo real. Essa preferência ocorre devido a sua característica estrutural, que opera a partir da “Família de Colunas”, que busca armazenar as informações em colunas, facilitando a compreensão dos dados e velocidade de consulta, já que é possível consultar informações específicas sem ler todos os atributos dos dados.

8. O que é a Consistência Eventual (adotada por muitos bancos NoSQL de alta disponibilidade, como DynamoDB ou Riak) e qual modelo NoSQL tipicamente a prioriza para garantir resiliência a falhas?

A Consistência Eventual basicamente diz que os dados não precisam estar completamente conectados entre si ao mesmo tempo, mas sim que todos os nós do sistema terão os mesmos dados em algum momento (não necessariamente ao mesmo tempo). Logo, isso significa que os dados podem estar temporariamente desatualizados em algum nó, mas que, em algum modelo, eles se atualizarão.

Pensando nisso, existe alguns bancos NoSQL que resistem a essa Consistência Eventual, que é o caso de bancos de dados orientados a Grafos, que exige uma consistência forte, já que a integridade das conexões entre os nós depende de atualizações imediatas.



9. Qual modelo (Documental ou Chave-Valor) é mais propenso a falhar em consultas que dependem de filtros por campos internos do valor, e por que a busca primária em Chave-Valor é limitada?

O modelo mais propenso a falhar no que diz respeito a filtros por campos internos do valor é o de Chave-Valor, já que sua busca depende necessariamente de atributos específicos como ID, tornando a consulta mais limitada e restrita. A limitação no que diz respeito a busca primária em Chave-Valor é justamente devido a causa descrita acima, devido a necessidade de depender de atributos específicos para fazer sua consulta.

10. Qual é o comando de linguagem do Cassandra (Colunar) que permite consultar por uma coluna que não faz parte da chave primária, e qual é o custo de desempenho associado a essa ação?

O Cassandra não permite consultas diretas que não fazem parte da chave primária sem criar um índice adicional. Para isso, devemos criar um “Secondary Index” ou usar “materialized views”, veja:

- `CREATE INDEX idx_coluna ON nome_tabela(nome_coluna);`

Depois disso, podemos fazer a consulta usando a coluna indexada que nós criamos a partir desse comando:

- `SELECT * FROM nome_tabela WHERE nome_coluna = 'valor';`

O custo disso reflete na consulta lenta para grandes volumes de dados, já que precisa varrer múltiplos nós para encontrar os valores correspondentes, o que acaba se tornando extremamente custoso. Além disso, não é escalável para tabelas extremamente grandes, já que funcionam bem apenas para baixas cardinalidades (poucos valores distintos na coluna), sendo que, para altas cardinalidades ou tabelas grandes, a operação pode causar alta latência e sobrecarga da rede.



11. Qual modelo NoSQL utiliza a sintaxe de consulta N1QL (SQL for JSON), permitindo consultas SQL sobre a estrutura de dados JSON?

É o banco de dados Documental, sendo mais específico o Couchbase. Com ele você consegue utilizar a linguagem de consulta N1QL, que se baseia em consultas SQL adaptadas para arquivos em formato JSON, permitindo consultas complexas, filtros, joins e agregações.

12. O que é um Tombstone no contexto de um banco Colunar como o Cassandra, e qual é o objetivo desse mecanismo na operação de DELETE?

Se refere a um marcador especial que indica que uma linha ou coluna foi excluída, sem remover imediatamente esse dado do disco. Ele acaba sendo fundamental principalmente para o Cassandra pois ele é um banco de dados distribuído por replicação, o que torna a exclusão dos dados de forma instantânea algo ruim, podendo gerar inconsistência nos outros nós.

O objetivo na operação DELETE é que, quando você executa esse comando, o Cassandra cria um Tombstone, sendo que esse marcador é replicado para todos os nós que possuem aquela linha ou coluna, garantindo que os outros nós “saibam” que o dado foi removido, evitando conflitos durante a replicação, além de remoção física posterior.



13. Em um banco de Grafos, como os conceitos de Nós (vértices) e Bordas (arestas) com Propriedades se relacionam para permitir a representação de relacionamentos complexos, diferente dos Documentos?

A estrutura visual de bancos de Grafos é pensada na busca por representar relacionamentos complexos entre os dados. A construção de nós (vértices) e bordas (arestas) com propriedades se relacionam de modo a construir todo um ecossistema que consegue relacionar os dados de uma forma intuitiva e simples.

Essa estrutura difere em diversos pontos de bancos de Documentos, já que os relacionamentos que são feitos nesses tipos de bancos não possuem uma preocupação visual nítida, mas sim “mínima”, já que trata a questão do relacionamento entre os dados a partir de pequenos atributos dentro dos próprios dados que vinculam um dado a outro, mas sem construir um ecossistema que facilita essa visualização, o deixando atrás quando o assunto é “representação e entendimento de relacionamentos complexos entre os dados.”

14. Justifique o uso do Apache Cassandra (Colunar) por empresas como Uber e Netflix, focando na necessidade de alta disponibilidade e tolerância a falhas sem um ponto único de falha.

É extremamente utilizado por empresas como Uber e Netflix devido a sua arquitetura distribuída, que atende necessidades fundamentais de alta disponibilidade e tolerância a falhas.

O Cassandra é projetado para operar em clusters de múltiplos nós, sendo que cada nó pode receber esses dados e replicá-los, tirando a ideia de “nó central”, que elimina o risco de interrupção entre os dados caso um nó falhe.

Além disso, temos a alta disponibilidade e escalabilidade horizontal, sendo que cada nó do cluster pode atender a leituras e gravações, permitindo que o servidor



permaneça ativado (mesmo que alguns caiam). Vinculado a isso, temos a escalabilidade horizontal, permitindo que seja possível armazenar novos nós e aumentar a capacidade dos dados sem que haja problemas de armazenamento.

Outro fator relevante é a replicação configurável e consistência eventual, já que os dados são replicados de forma flexível, garantindo que as informações nunca se percam se um nó ficar indisponível temporariamente.

Ademais, temos um extenso suporte a workloads de alta taxa de escrita e leitura, já que empresas como Uber e Netflix lidam com grandes volumes de dados e eventos em tempo real, sendo fundamental a capacidade de processar milhões de operações por minuto sem comprometer a disponibilidade.

15. Qual é a linguagem de consulta do banco Documental mais popular (MongoDB) e qual tipo de linguagem de programação ela se assemelha?

A linguagem de consulta do MongoDB é o MongoDB Query Language (MQL), uma linguagem nativa baseada em documentos JSON. Sobre sua linguagem de consulta, ela se assemelha ao JSON (JavaScript Object Notation), se assemelhando a construção de arquivos JSON dentro do JavaScript.

16. Por que o Modelo Colunar é classificado como ideal para análises agregadas, enquanto o Modelo Chave-Valor não é recomendado para consultas analíticas complexas?

O Modelo Colunar acaba se destacando quando o assunto é análises agregadas devido ao fato de que sua estrutura de armazenamento “Família de Colunas” é interessante para comparar diversos tipos de atributos específicos, sem a necessidade de processar todo um conjunto de dados.

No que diz respeito a utilização de modelos Chave-Valor e o fato de não ser recomendado para consultas analíticas complexas, isso ocorre devido ao fato de que esse modelo de banco de dados depende de especificações criteriosas sobre os atributos que desejam ser observados, além de processar todos os atributos do conjunto de dados, tornando inviável em aplicações complexas.



17. Como o Sharding é aplicado nos bancos Chave-Valor para lidar com o volume massivo de dados e picos de acesso?

Se trata de uma técnica fundamental para ser usada em bancos de dados Chave-Valor para lidar com grandes volumes de dados e picos de acesso, garantindo desempenho e escalabilidade.

O Sharding começa com a divisão dos dados em fragmentos (chamados de Shards). Sobre essas divisões, o banco distribui os pares chave-valor entre múltiplas divisões do cluster, sendo que cada shard armazena apenas uma parte do conjunto total de dados, evitando que um único nó fique sobrecarregado.

Após isso, ele é mapeado por chaves (Hashing ou Range) que é usado para determinar em qual shard o dado em específico será armazenado, sendo que os métodos comuns para isso consistem o “Consistent Hashing”, que basicamente busca distribuir uniformemente os dados entre os shards e facilita a adição ou remoção dos nós, ou então o “Range-based Sharding”, que basicamente define que cada shard recebe um intervalo específico de chaves.

Após esses processos, deve-se garantir o balanceamento de cargas, que basicamente permite que o sharding faça leitura e escrita de forma paralela em múltiplos nós, suportando picos de acesso sem limitar o desempenho, além da possibilidade de adicionar novos nós aos clusters, redistribuindo esses shards de forma dinâmica para acomodar todo o crescimento dos dados (escalabilidade horizontal, resumidamente).

18. Qual modelo NoSQL é o mais versátil por ser Multi-Modelo (chave-valor, documento JSON, colunar e grafo), e qual serviço da Microsoft oferece essa flexibilidade?

O modelo NoSQL mais versátil por conter diversos tipos de bancos de dados NoSQL é o Multi-Modelo, sendo que a Microsoft oferece um serviço específico que trata desse modelo de banco de dados, sendo o caso do Azure Cosmos DB.



19. Qual banco Documental é o mais adequado para o ecossistema .NET/C#, sendo conhecido por sua facilidade de uso e garantias ACID?

Essencialmente o RavenDB, embora bancos de dados NoSQL priorizem escalabilidade e flexibilidade.

20. Qual a diferença entre a função MERGE() e as funções CREATE() / DELETE() em Cypher, a linguagem de consulta de Grafos?

A diferença é que os comandos CREATE(), DELETE() e MERGE() está no seguinte:

CREATE(): É a função utilizada para criar (acrescentar) novos vértices para o banco de dados, sendo aqui que você define os primeiros parâmetros, rótulos e relacionamentos;

DELETE(): É a função responsável por remover os vértices do banco de dados. Entretanto, essa remoção nunca é feita sem que você exclua primeiramente os relacionamentos que aquele dado em específico faz;

MERGE(): Seu objetivo é encontrar relacionamentos entre os vértices, criando o relacionamento caso não exista. Se existir, ele não duplica o relacionamento. Além disso, podemos criar o que chamamos de “pesos”, que pode representar valores numéricos que carregam quantas vezes aquele determinado relacionamento existe sem recriar esses relacionamentos visualmente, evitando confusões ou poluições visuais.



21. Por que é mais difícil realizar a Escalabilidade Horizontal em bancos de Grafos do que em bancos Chave-Valor?

Pois a Escalabilidade Horizontal se trata da busca por repassar dados para outros servidores, ainda assim, mantendo a conexão entre os dados e suas relações - mesmo em servidores diferentes, aumentando sua capacidade de armazenamento.

Entretanto, bancos de dados de Grafos possuem um critério fundamental: tornar a visualização dos relacionamentos e conexões intuitivos, o que acaba exigindo uma intensa conexão entre os diversos servidores, tornando o processo muito mais custoso e lento.

Isso não ocorre quando pensamos em bancos de dados baseados a Chave-Valor, pois seus dados são armazenados de forma completa, sendo que a visualização dos relacionamentos e a conexão entre esses dados não depende de uma intensa comunicação entre os servidores, tornando a escalabilidade horizontal muito mais simples.

22. Descreva as três principais fases de armazenamento em um banco Colunar (como Cassandra) após a inserção de um novo dado, envolvendo Memória, Log e Disco.

Quando um novo dado é inserido, ele não é gravado diretamente em Disco, mas sim, é armazenado na “MemTable”, que é uma estrutura de dados em memória RAM ordenada e otimizada para escrita rápida. Sobre essa etapa, ela permite que se tenha inserções de modo muito mais eficiente, evitando o custo imediato de escrita na RAM. Depois disso, temos o registro no Log de Commit, que chamamos de Commit Log. Ao mesmo tempo que os dados são gravados na MemTable, o Cassandra também registra a operação no Commit Log, que é gravado em disco de forma sequencial, servindo como mecanismo de recuperação em caso de falha do sistema – garantindo que nenhuma escrita seja perdida.

Por final, temos a escrita em Disco (chamada de SSTable), que é quando a MemTable atinge o seu limite de tamanho e acaba sendo descarregado para o disco, gerando um novo arquivo SSTable, sendo que esses arquivos são imutáveis e o Cassandra gerencia diversos desses SSTables através de processos periodicos de compactação, que une e otimiza os dados que estão armazenados.



23. Qual banco de dados Chave-Valor foi desenvolvido pelo LinkedIn para substituir bancos relacionais que não suportavam o volume de acessos?

O banco de dados NoSQL desenvolvido pelo LinkedIn para substituir bancos relacionais que não suportavam grandes volumes de acesso foi o “Project Voldemort”, sendo que esse banco de dados NoSQL é baseado em Chave-Valor.

24. Qual modelo de banco de dados se concentra em evitar pontos únicos de falha e garantir a disponibilidade e tolerância a falhas através da replicação entre nós?

Bancos de dados que se preocupam em evitar pontos únicos de falha para garantir disponibilidade e tolerância a falhas através da replicação de nós é característico de bancos de dados NoSQL baseados a Chave-Valor.

Entretanto, apesar do modelo Chave-Valor ser o mais clássico e evidente, outros modelos também se destacam por isso, sendo o Colunar e o Documental, já que eles também utilizam conjuntos para garantir tolerância e evitar falhas de comunicação - já que todos se baseiam em uma mesma filosofia: Consistência Eventual.

25. Qual banco Documental é mais adequado para aplicações de tempo real, jogos e e-commerce, por ter arquitetura de armazenamento e computação separada (Couchbase)?

O banco de dados Documental que é particularmente adequado para essas aplicações de alta demanda como jogos, e-commerce e aplicações em tempo real é o CouchBase, pois, conforme o próprio enunciado especifica, possui uma arquitetura de armazenamento e computação separado, tornando o processo de consulta e construção de relacionamento muito mais específico, eficiente e menos custoso.



26. Qual a principal limitação dos bancos Colunares que impede a execução de consultas SQL complexas, como joins?

A principal limitação de bancos Colunares que impede a execução de consultar SQL complexas é o fato de que esse modelo de banco de dados foi projetado para consultar colunas específicas de forma eficiente e menos custosa, invés de trabalhar com registros completos ou relacionamentos entre tabelas. Como resultado disso, a operação depende da combinação de várias tabelas ou atributos completos, tornando os joins muito difíceis, ineficientes e custosos.

27. Como o modelo Colunar organiza os dados em um nível fundamental (chave primária → famílias de colunas → colunas → valores) para otimizar a leitura e o processamento de grandes volumes de dados?

O modelo colunar organiza os dados a partir de 4 processos fundamentais que foram destacados no enunciado: chave-primária, que especifica os dados, particularizando as informações, famílias de colunas, já que esse modelo de banco de dados opera com a partição dos dados, separando os atributos em conjuntos específicos de dados, facilitando o processo de consulta. Depois disso, temos as colunas em si que são separadas, e os valores contidos nessas colunas.

Todo esse processo otimiza a leitura e o processamento de grandes volumes de dados devido ao fato de que particulariza as informações em “clusters” menores e específicos, evitando a necessidade de processar todos os atributos de um determinado dado durante a consulta, tornando o processo muito mais rápido e direcionado.



28. Qual banco de dados em Grafos (TigerGraph ou Neo4j) é considerado mais rápido e analítico para Big Data, utilizando sua própria linguagem GSQL?

O banco de dados em Grafos mais rápido e analítico para Big Data é o TigerGraph, usando sua própria linguagem GSQL. Esse banco de dados baseado a Grafos utiliza uma arquitetura de processamento paralelo e distribuído, tornando altamente eficiente para análises profundas. Sobre sua linguagem, ela é projetada para realizar análises complexas e algorítmicas de grafos de forma eficiente e expressiva.

29. Por que a representação de perfis e preferências de usuário é um caso de uso principal para os bancos Chave-Valor (como DynamoDB)?

Pois bancos Chave-Valor como o DynamoDB são projetados para armazenar atributos por usuário de forma flexível, tornando o processo de armazenamento de preferências de usuário muito mais simples, já que cada usuário pode conter uma estrutura de atributos diferente.

30. Cite a principal desvantagem do modelo Chave-Valor em termos de complexidade de transação em comparação com bancos relacionais.

A principal desvantagem está na falta de Atomicidade de Múltiplas chaves e a adoção da Consistência Eventual como filosofia de bancos Chave-Valor. Esses dois pontos tornam o banco de dados baseado a Chave-Valor menos eficiente quando se trata de transações, já que é menos confiável e suscetível a falhas. Isso não ocorre quando pensamos em bancos de dados relacionais, já que eles priorizam ACID – Atomicidade dos dados, consistência imediata, isolamento dos dados e durabilidade.



31. Qual o comando CQL (Cassandra Query Language) usado em bancos Colunares para criar uma tabela e qual a importância da cláusula PRIMARY KEY nesse contexto?

O comando usado para criar uma tabela dentro do CQL (Cassandra Query Language) é o CREATE TABLE(), que é bem semelhante ao SQL. Sobre a criação da tabela, a cláusula PRIMARY KEY é exclusiva para especificar os dados, sendo um identificador único para cada dado dentro do meu banco de dados, sendo fundamental para especificar consultas, criar relacionamentos, fazer edições ou exclusões entre esses dados de forma direcionada.

Além disso, ele é que define a “arquitetura de armazenamento e distribuição dos dados”, pois ela que determina em qual servidor o dado será armazenado, sendo que o Cassandra usa o algoritmo de hash de Chave de Partição para saber onde colocar e onde procurar os dados. De modo geral, o PRIMARY KEY opera como esquema de particionamento e ordenação, sendo o fator mais crítico para o design de dados e a performance de consulta em bancos de dados colunares (como o Cassandra).

32. Por que a Modelagem Intuitiva é uma das maiores vantagens dos bancos de Grafos?

Pois ela é capaz de representar conexões a partir de poucos elementos visuais, se tornando uma vantagem exclusiva dos bancos de dados a grafos, já que é possível observar conexões e relação entre os dados a partir de pouca programação, sendo simples de visualizá-los.



33. Em termos de licenciamento, quais são as diferenças entre o Apache Cassandra e o Google Bigtable/Amazon Redshift?

As diferenças de licenciamento entre o Apache Cassandra e serviços como o Google Bigtable e Amazon Redshift está principalmente na forma como o software é distribuído e utilizado. O Apache Cassandra é livre para uso, modificação e redistribuição, inclusive para ambientes comercial, podendo ser hospedado em qualquer infraestrutura ou em nuvem sem custos de licenciamento. Nesse caso, você apenas paga pelo hardware ou os serviços de infraestrutura necessários.

Já o Google Bigtable e Amazon Redshift tem uma licença proprietária (SaaS), sendo que suas características incluem o gerenciamento dos serviços - logo, o usuário não controla o software diretamente. Além disso, tem a cobrança baseada em uso de recursos como armazenamento e instâncias de computação, não sendo possível modificar o código-fonte do sistema, sendo que sua escalabilidade, manutenção e atualização são gerenciadas pelo provedor (Google ou Amazon).

34. Qual modelo NoSQL é o mais adequado para a telemetria e IoT que exige armazenamento rápido do estado atual de sensores ou dados agregados?

O modelo Chave-Valor se destaca para aplicações como telemetria (tecnologia de coleta, transmissão e análise remota de dados) e IoT, onde é necessário armazenar rapidamente o estado atual de sensores ou dados agregados.



35. Por que o Redis é classificado como um banco Chave-Valor com desempenho ultrarrápido, sendo ideal para cache e filas?

É classificado como desempenho ultrarrápido porque armazena os dados na memória RAM, permitindo leituras e gravações de forma extremamente rápida e segura - já que os dados são salvos diretamente.

Ele acaba se destacando para cache e filas devido ao fato de que não armazena apenas strings como a maioria dos bancos Chave-Valor, mas também suporta estruturas complexas nativamente, sendo todas otimizadas para operações na memória.

36. Qual a principal diferença na leitura de dados entre a Escalabilidade Horizontal e a Escalabilidade Vertical nos bancos de dados?

A principal diferença está na maneira como esses dados são distribuídos e acessados. Na escalabilidade vertical nós temos o aumento na capacidade de um único servidor (com mais CPU, RAM, armazenamento), sendo que as leituras de dados ocorrem em um único nó, o que mantém a simplicidade – mas tem limite físico de performance.

Já a escalabilidade horizontal é a adição de mais servidores ou nós ao cluster, distribuindo os dados entre eles. Sobre sua leitura, as consultas podem precisar acessar vários nós, exigindo coordenação e comunicação entre esses servidores, sendo justamente uma desvantagem - já que pode haver latência adicional e maior complexidade para garantir consistência, mas permite lidar com volumes muito maiores de dados.



37. Qual o modelo NoSQL (Colunar ou Documental) é mais adequado para a gestão de conteúdo com textos e imagens dinâmicos?

O modelo NoSQL Documental é o mais adequado por consegue armazenar os dados em documentos JSON ou BSON que possuem estruturas flexíveis, sendo que essa flexibilidade que torna ideal para conteúdos como textos e imagens dinâmicas.

38. Qual modelo NoSQL é o mais eficiente para a análise de conexões e rotas (logística, transporte), permitindo simular falhas e calcular rotas otimizadas?

Bancos de dados baseados a Grafos são mais eficientes para análises de conexões e rotas devido a sua representação intuitiva e visual, tendo capacidade de se conectar com outros dados e calcular rotas e simular falhas de forma otimizada, rápida, eficiente e em grande escala.

39. Qual banco de dados Colunar é focado em análise de dados e oferece o melhor desempenho em consultas e relatórios?

O banco de dados mais focado para análise de dados e oferece maior desempenho em consultas e relatórios é o Amazon Redshift.

40. Qual banco de dados em Grafos utiliza Cassandra ou HBase como backend de armazenamento e é conhecido por ser Open Source e altamente distribuído?

O banco de dados baseado a grafos que utiliza Cassandra ou HBase como backend de armazenamento, sendo open source e altamente distribuído é o JanusGraph. Ele é ideal quando se precisa de grafos distribuídos e em larga escala, sendo que essa tecnologia entregue pelo JanusGraph só acontece devido a utilização do Cassandra ou



HBase "por baixo dos panos", tornando esse banco de dados baseado a grafos extremamente complexo e eficiente.