

Desafio Parte 1

Grupo: Solutions Database

Alisson Rodrigues - RM 550239

Luis Augusto de Lara Cavalcanti - RM 98873

Luis Henrique Vargas Santos - RM 99309

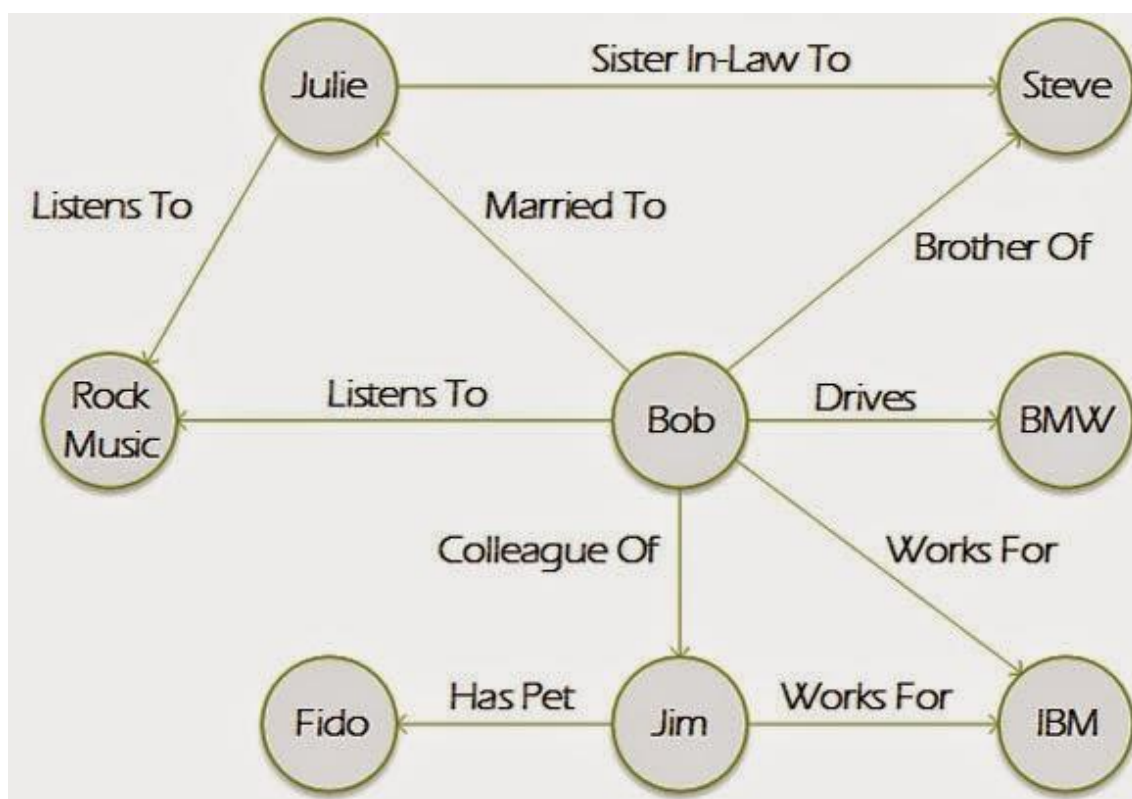
Natan Fogaca Tenorio - RM 551347

Cenário 1: Recomendações de Produtos

Escolha Proposta: Banco de Dados NoSQL do Tipo Grafo.

Acreditamos que o modelo tipo Grafo seja o mais eficiente para essa tarefa. Por ser um formato de Dados Flexível, os bancos de dados de grafo são ideais para representar dados com relações complexas.

Eles usam uma estrutura de grafo composta por nós (entidades) e arestas (relacionamentos). Isso se alinha perfeitamente com a necessidade de calcular recomendações com base nas compras de clientes, onde os relacionamentos entre produtos e clientes são cruciais.



Os bancos de dados de grafos são uma boa opção para aplicativos de recomendação, você pode armazenar em um grafo os relacionamentos entre as categorias de informação, como os interesses, os amigos e o histórico de compras dos clientes.

Por isso o formato de grafo fornece uma plataforma mais flexível para encontrar conexões distantes ou analisar dados com base em aspectos como força ou qualidade do relacionamento. Os grafos permitem que você explore e descubra conexões e padrões em redes sociais, IoT, big data, data warehouses e também dados de transações complexas para vários casos de uso de negócios, incluindo detecção de fraude em bancos, descoberta de conexões em redes sociais.

Como os bancos de dados de grafos armazenam os relacionamentos de forma explícita, as consultas e os algoritmos que utilizam a conectividade entre vértices podem ser executados em subsegundos em vez de horas ou dias. Os usuários não precisam executar inúmeras junções e os dados podem ser usados com mais facilidade para análise e aprendizado de máquina para descobrir mais sobre o mundo ao nosso redor.

Sendo assim, para esses casos de relacionamentos complexos, podem ser determinados para insights mais profundos com muito menos esforço. Os bancos de dados de grafos geralmente executam consultas em linguagens como Property Graph Query Language (PGQL). O exemplo abaixo mostra a mesma consulta em PGQL e SQL.

PGQL:

```
PATH shares_movie_with AS (from) <- (acted_in) -> (to)
SELECT y.name
MATCH (x:Actor) -/:shares_movie*/->(y:Actor)
WHERE x.name = 'Iron Man'
AND x <> y
```

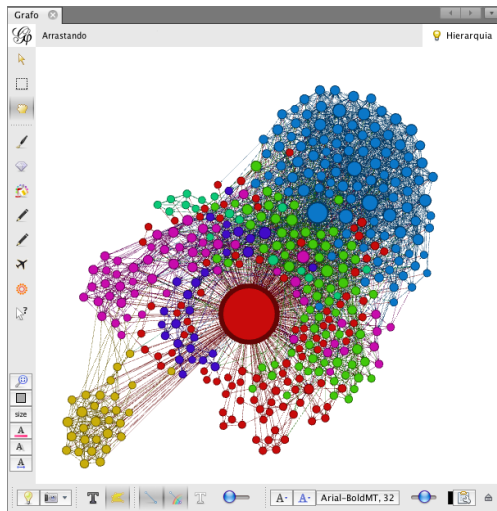
SQL Equivalent:

```
WITH temp(actor_id, actor_name) AS (
  —Anchor member:
  SELECT actor_id, name
  FROM Devices
  WHERE name = 'Iron Man'
  UNION ALL
  —Recursive member:
  SELECT Actors.actor_id, Actors.name
  FROM temp, Actors, Connections conn1,
       Connections conn2, Movies
  WHERE temp.actor_id = conn1.to_actor_id
        AND conn1.from_acted_in_id = Connectors.movies_id
        AND Connectors.movie_id = conn2.from_movie_id
        AND conn2.to_actor_id = Devices.actor_id
        AND temp.actor_id != Actors.actor_id)
CYCLE actor_id SET cycle TO 1 DEFAULT 0
SELECT DISTINCT actor_name
FROM temp
WHERE cycle = 0
AND actor_name <> 'Iron Man'
```

Outro motivo é a escalabilidade para Grandes Conjuntos de Dados, bancos de dados de grafo, como o NEO4J, são escaláveis e podem lidar com grandes volumes de dados interconectados. Além disso, a flexibilidade oferecida por esses bancos de dados permite que a plataforma de e-commerce se adapte rapidamente às mudanças nos padrões de compra e preferências dos clientes. Com a capacidade de adicionar novos tipos de nós e relacionamentos conforme necessário, a plataforma pode evoluir de forma ágil para atender às demandas em constante evolução do mercado.

Em resumo, os bancos de dados de grafo são uma escolha poderosa para empresas de comércio eletrônico que desejam oferecer recomendações personalizadas, otimizar a eficiência das consultas de dados e escalonar para acomodar grandes volumes de informações interconectadas.

Exemplo Prático: O LinkedIn é uma plataforma de rede social profissional que faz uso extensivo de bancos de dados de grafo para modelar conexões entre usuários, grupos e empresas, bem como para fornecer recursos de busca avançada. Essa abordagem de banco de dados de grafo é fundamental para a experiência do LinkedIn, permitindo que os usuários construam perfis detalhados, conectem-se com colegas de trabalho, sigam empresas de interesse e explorem oportunidades profissionais de maneira eficaz.



A estrutura de grafo do LinkedIn possibilita a representação precisa das relações complexas entre os usuários. Cada perfil de usuário é um nó no grafo, e as conexões entre eles são representadas por arestas. Isso permite que os usuários visualizem de forma clara e interajam com suas redes profissionais, identifiquem conexões mútuas e explorem oportunidades de carreira.

Além disso, os bancos de dados de grafo também são cruciais para a funcionalidade de busca avançada do LinkedIn. Eles permitem que os usuários pesquisem de maneira eficiente por critérios específicos, como habilidades, localização geográfica e conexões em comum. Essa capacidade de busca avançada ajuda os usuários a encontrar perfis relevantes e oportunidades de emprego de forma mais precisa e eficaz.

Cenário 2: Controle de Estoque para Entrega Rápida

Para esse desafio, escolhemos o Banco de Dados NoSQL do Tipo Colunar, focando principalmente na eficiência em Armazenamento e Recuperação.

Row-oriented			
ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

Column-oriented					
Name	ID	Grade	ID	GPA	ID
John	001	Senior	001	4.00	001
Karen	002	Freshman	002	3.67	002
Bill	003	Junior	003	3.33	003

Bancos de dados colunares são otimizados para o armazenamento eficiente de grandes conjuntos de dados com muitas colunas. Isso é crucial para o controle de estoque, pois

envolve armazenar informações sobre muitos produtos diferentes, suas quantidades e status de disponibilidade.

Vantagens das bases de dados em colunas:

As bases de dados em colunas são excelentes em termos de compressão, pelo que são eficientes em termos de armazenamento. Isto significa que é possível reduzir os recursos do disco, mantendo grandes quantidades de informação numa única coluna. Uma vez que a maioria das informações é armazenada numa coluna, as consultas de agregação são bastante rápidas, o que é importante para projectos que requerem grandes quantidades de consultas num curto espaço de tempo. A Alta Velocidade de Leitura também é um dos principais benefícios dos bancos de dados colunares é sua capacidade de realizar leituras rápidas. Isso é vital para verificar rapidamente a disponibilidade de produtos em estoque e atender às solicitações dos clientes por entregas rápidas.

A escalabilidade é excelente nas bases de dados de armazenamento em coluna. Podem ser expandidas quase infinitamente e estão frequentemente distribuídas por grandes grupos de máquinas, que chegam a atingir os milhares. Isto também significa que são ótimas para o processamento paralelo maciço, os tempos de carregamento são igualmente excelentes, uma vez que é possível carregar facilmente uma tabela de mil milhões de linhas em poucos segundos. Isto significa que pode carregar e consultar quase instantaneamente.

Grande flexibilidade, uma vez que as colunas não têm necessariamente de se parecer umas com as outras. Isto significa que pode adicionar colunas novas e diferentes sem perturbar toda a base de dados. Dito isto, a introdução de consultas de registos completamente novos requer uma alteração em todas as tabelas.

De um modo geral, as bases de dados com armazenamento de colunas são ótimas para análises e relatórios: as velocidades de consulta rápidas e a capacidade de armazenar grandes quantidades de dados sem acrescentar muitas despesas gerais tornam-nas ideais.

Por fim, temos o suporte para Atualizações de Alta Frequência: Nos cenários de controle de estoque, é comum que as atualizações de estoque ocorram com alta frequência à medida que os produtos são reservados e entregues. Bancos de dados colunares podem lidar bem com essas atualizações frequentes.

Exemplo Prático: A Netflix é um exemplo do uso de um banco de dados NoSQL do tipo colunar, como o Cassandra, para lidar com volumes maciços de dados relacionados a filmes e séries, incluindo o controle de disponibilidade de conteúdo para streaming.

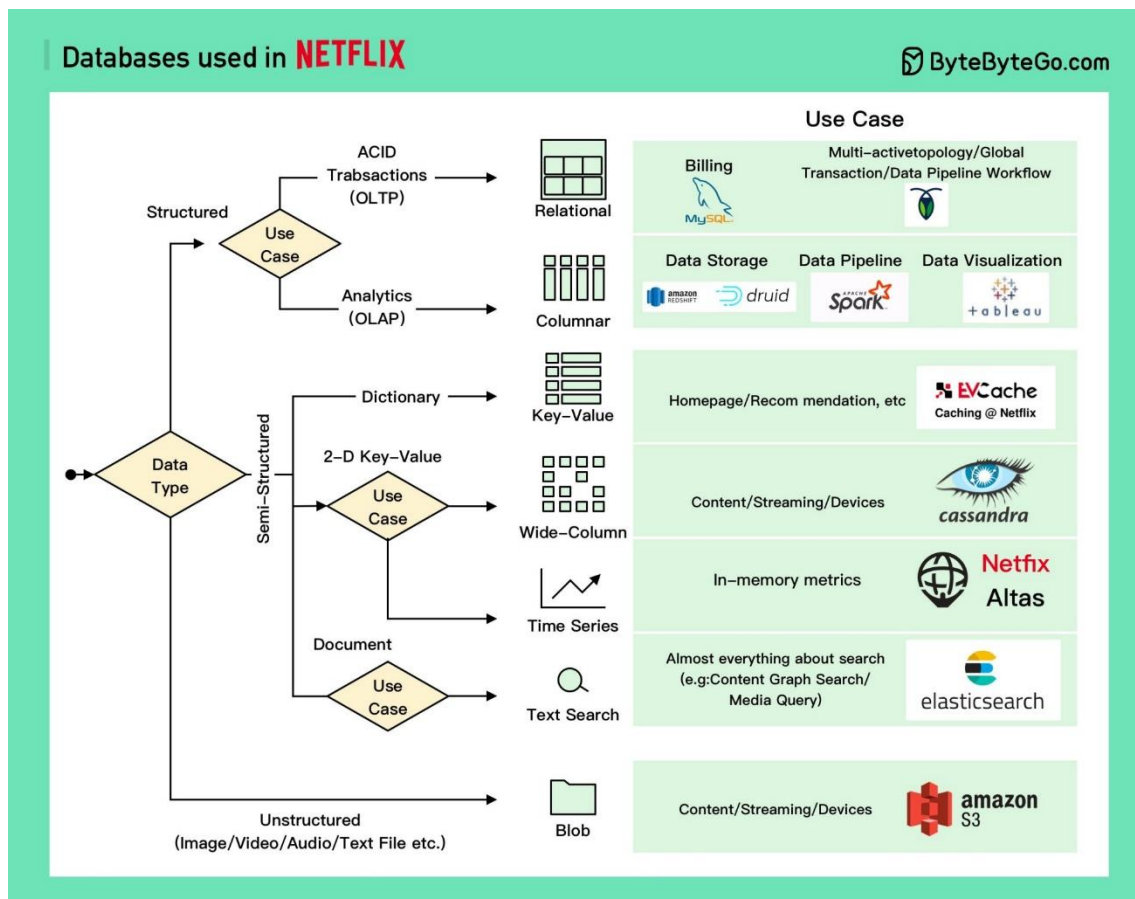
Cassandra satisfaz os requisitos de uma empresa que cresceu e necessitava de performance junto com alta escalabilidade. Hoje, Cassandra é o banco de dados preferido para quase todas as necessidades de banco de dados na Netflix. A empresa armazena 95% de todos os seus dados em Cassandra, incluindo informações de contas de clientes, classificações de filmes, metadados de filmes, marcadores de filmes e registos.

Podemos citar assim os principais motivos para a escolha do Cassandra como solução de armazenamento:

- Extremamente escalável;
- Solução comprovada;
- Forte desempenho;
- Tecnologia descentralizada;
- Alta flexibilidade;
- Ótimo para múltiplos lugares;

- Suporte de grandes empresas.

De acordo com os últimos dados divulgados, a Netflix opera mais de 50 clusters Cassandra com mais de 25000 nós. No pico, eles processam mais de 50.000 leituras por segundo e 100.000 gravações por segundo em todos os seus clusters. Em média, eles processam mais de 1 trilhão de requisições por dia.

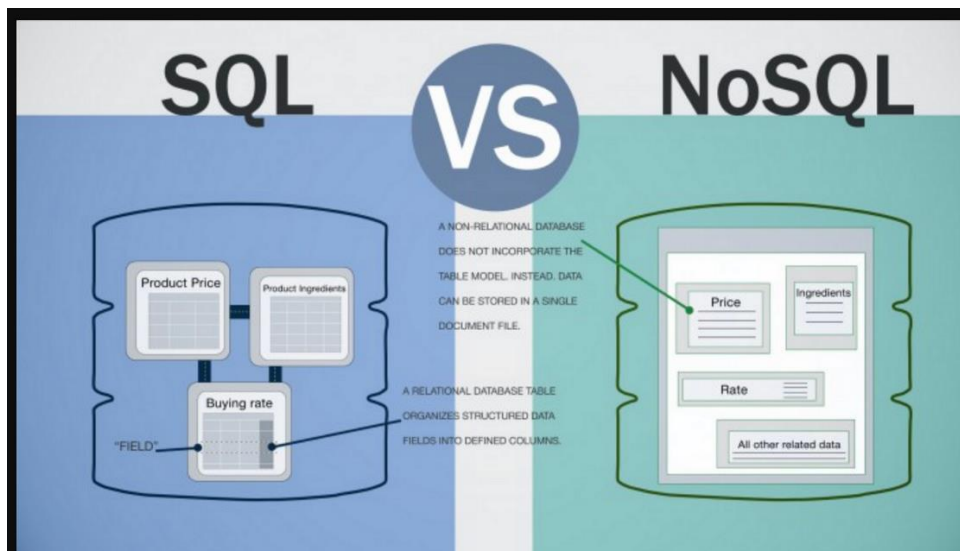


A Netflix utiliza o Cassandra pela sua escalabilidade e ausência de pontos únicos de falha e para implementações inter-regionais. "De facto, um único cluster global do Cassandra pode servir simultaneamente aplicações e replicar dados de forma assíncrona em várias localizações geográficas.", escreve Izrailevsky

Ao contrário de uma solução de banco de dados distribuído usando, por exemplo, MySQL ou até mesmo SimpleDB, o Cassandra (como o HBase) pode escalar horizontalmente e dinamicamente adicionando mais servidores, sem a necessidade de re-shard - ou reiniciar, por falar nisso. De facto, o Cassandra procura evitar limites de escalabilidade vertical e estrangulamentos de qualquer tipo: não existem nós de nomes dedicados (todos os nós do cluster podem servir como tal), não existem limitações arquitectónicas práticas em termos de dimensão dos dados, contagem de linhas/colunas, etc.

Também faz uma menção especial ao modelo de dados do Cassandra, que oferece representações de modelos flexíveis para além do típico modelo de pesquisa de valores-chave.

Concluimos como o Cassandra forneceu uma nova base mais sólida para a Netflix. Como Adrian Cockcroft diz: "Torna possível para nós entregar o serviço em que nossos membros dependem."



Cenário 3: Armazenamento de Dados de Produtos Detalhados

Nossa escolha para o armazenamento desses dados foi o Banco de Dados NoSQL do Tipo Documento. Um documento é um registro numa base de dados de documentos, que armazena informações sobre um objeto e qualquer um dos seus metadados relacionados, armazenados em pares campo-valor. Ao contrário de outras bases de dados que armazenam dados em estruturas como tabelas ou gráficos), os documentos são mapeados para objectos nas linguagens de programação mais populares, o que permite aos programadores desenvolver rapidamente as suas aplicações.

Os valores podem ser de uma variedade de tipos e estruturas, incluindo cadeias de caracteres, números, datas, matrizes ou objectos. Os documentos podem ser armazenados em formatos como JSON, BSON e XML.

Visto que Bancos de dados de documentos permitem que os dados sejam armazenados com esquemas flexíveis, o que é fundamental quando se trata de informações detalhadas de produtos. Esses dados podem variar significativamente, e a capacidade de adicionar campos conforme necessário é uma vantagem.

Gerenciamento de Dados Semiestruturados: No cenário de armazenamento de informações de produtos, os dados podem incluir informações semiestruturadas, como avaliações de produtos, imagens, recomendações e dicas. Os bancos de dados de documentos podem acomodar facilmente esses tipos de dados.

Facilidade de Consulta: Os bancos de dados de documentos geralmente oferecem consultas eficientes, permitindo que os detalhes do produto sejam recuperados facilmente para exibição na plataforma de e-commerce.

As bases de dados de documentos têm normalmente uma API ou linguagem de consulta que permite aos programadores executar as operações CRUD (criar, ler, atualizar e eliminar).

Criar: Os documentos podem ser criados na base de dados. Cada documento tem um identificador único.

Ler: Os documentos podem ser lidos a partir da base de dados. A API ou linguagem de consulta permite que os programadores consultem os documentos utilizando os seus identificadores únicos ou valores de campo. Podem ser adicionados índices à base de dados para aumentar o desempenho da leitura.

Atualizar: Os documentos existentes podem ser actualizados - no todo ou em parte.

Eliminar: Os documentos podem ser eliminados da base de dados.

Exemplo de uma base de documentos:

Document 1

```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

Exemplo Prático: O MongoDB é um exemplo prático de um banco de dados NoSQL do tipo documento amplamente utilizado. Empresas como a Airbnb usam o MongoDB para armazenar informações detalhadas sobre propriedades e avaliações de hóspedes. A escolha do MongoDB se baseia em sua flexibilidade na modelagem de dados e sua capacidade de lidar com volumes significativos de informações.

MY ORGANIZATION > MY PROJECT > CLUSTERS

Cluster0

Overview Real Time Metrics Collections Search Profiler

DATABASES: 8 COLLECTIONS: 21

+ Create Database

Q NAMESPACES

sample_airbnb

listingsAndReviews

sample_analytics

sample_geospatial

sample_mflix

sample_restaurants

sample_supplies

sample_training

sample_weatherdata

sample_airbnb.listingsAndReviews

COLLECTION SIZE: 89.99MB TOTAL DOCUMENTS: 5555

Find Indexes Schema Anti-Patterns

FILTER {"filter": "example"}

QUERY RESULTS 1-20 OF MANY

>

_id: "10006546"

listing_url: "https://www.airbnb.com/rooms/10006546"

name: "Ribeira Charming Duplex"

summary: "Fantastic duplex apartment with three spaces, privileged views of the Douro River and..."

description: "Fantastic duplex apartment with three spaces, privileged views of the Douro River and..."

neighborhood_overview: "In the neighborhood of the Douro River, the apartment is located in a quiet street with a view of the river and the city of Porto." notes: "Lose yourself in the narrow streets and squares of the historic center of Porto." transit: "Transport: • Metro station and S. Bento station, 5 minutes walk."

O MongoDB é particularmente adequado para cenários em que a estrutura dos dados pode variar de documento para documento. No contexto da Airbnb, onde as propriedades podem variar consideravelmente em termos de detalhes e características, a flexibilidade oferecida

pelo MongoDB é crucial. Os documentos JSON ou BSON podem conter campos diferentes, permitindo que a empresa acomode facilmente informações diversas, desde fotos e descrições até avaliações e preços, em um único banco de dados.

Outra vantagem do MongoDB é sua escalabilidade horizontal, que é essencial para lidar com a grande quantidade de dados gerados pela plataforma Airbnb. À medida que a empresa cresce e adiciona mais propriedades e hóspedes, o MongoDB pode ser dimensionado para acomodar essa expansão sem interrupções significativas nos serviços.

Além disso, a capacidade de consulta eficiente do MongoDB permite que a Airbnb forneça aos seus usuários resultados de pesquisa rápidos e personalizados, ajudando os hóspedes a encontrar facilmente as propriedades que atendam às suas necessidades e aos anfitriões a gerenciar suas listagens de forma eficaz.