**Bancos de dados colunares**

Banco de dados colunares possuem o armazenamento orientado a colunas, o que influencia significativamente na sua performance, já que diminuem a quantidade de dados que você precisará carregar no disco.

Esses bancos de dados utilizam-se de tabelas para representar entidades e seus dados são gravados em discos e agrupados por colunas separadas.

# **Vantagens:**

* **Alto desempenho em consultas analíticas**: O armazenamento colunar é extremamente eficiente para executar **consultas analíticas** complexas, especialmente em grandes datasets. Isso se deve à capacidade de ler apenas as colunas necessárias para uma consulta, reduzindo significativamente o tempo de acesso aos dados.
* **Otimização de armazenamento**: Dados armazenados colunarmente, podem ser comprimidos de forma mais eficaz, reduzindo os requisitos de armazenamento e melhorando o desempenho das consultas.

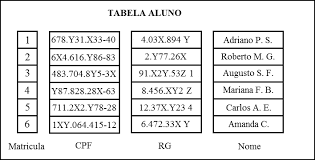
# Desvantagens:

* **Ineficiência em transações**: Dado que cada coluna é armazenada separadamente, as operações de inserção, atualização e deleção podem ser mais lentas em comparação com o armazenamento em linha, tornando esta abordagem menos ideal para aplicações que exigem alta taxa de transações.

# Casos de Uso

* **Armazenamento em linha**: É mais adequado para aplicações que exigem uma alta taxa de transações e operações CRUD (Criar, Ler, Atualizar, Deletar), como sistemas de gerenciamento de pedidos, aplicações bancárias, e sistemas de gerenciamento de conteúdo.
* **Armazenamento colunar**: Excelente para aplicações de análise de dados e business intelligence (BI), onde consultas complexas e a análise de grandes volumes de dados são comuns. É a escolha preferida para soluções de análise de dados em grande escala.

Exemplo :



## Bancos de dados orientado a documentos

Os bancos de dados orientados a documentos são um dos tipos mais populares de bancos de dados não relacionais. Nesse modelo, os dados são armazenados em documentos, geralmente no formato JSON ou XML.

Cada documento pode ter uma estrutura diferente, o que oferece flexibilidade para lidar com dados não estruturados ou semiestruturados. Além disso, esse tipo de banco de dados permite consultas ricas e ágeis, bem como a capacidade de indexação dos campos dos documentos.

Vantagens

* Permitem esquemas flexíveis, adaptando-se facilmente a mudanças nos requisitos de dados sem a necessidade de alterar estruturas fixas.
* Contém todas as informações importantes em um único documento.
* Oferecem alto desempenho em operações de leitura e gravação de documentos individuais ou pequenos conjuntos de documentos.
* Consultar os documentos através de métodos avançados de agrupamento e filtragem.
* Codifica os documentos com XML, JSON, BSON e os formatos binários mais conhecidos como PDF,DOC,XLS e etc.

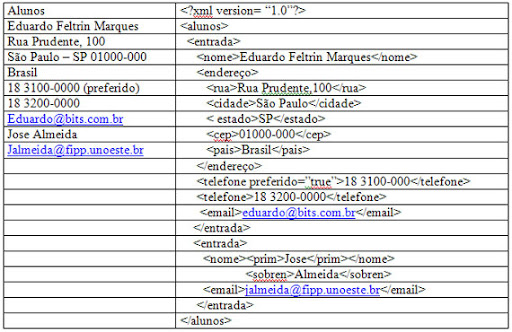
Desvantagens

* A flexibilidade de esquema pode levar a uma modelagem complexa de dados, especialmente em cenários com relacionamentos complexos entre documentos.
* Muitos sistemas NoSQL sacrificam consistência forte em favor de disponibilidade e tolerância a falhas, o que pode resultar em inconsistências temporárias nos dados.
* Podem enfrentar dificuldades em consultas que envolvem operações complexas, como joins extensivos ou agregações complexas entre diferentes documentos.
* O crescimento significativo de documentos pode afetar o desempenho de operações de leitura e gravação, especialmente em ambientes distribuídos.

**Casos de uso**

* **Gerenciamento de conteúdo e CMS (Content Management Systems)**: Sistemas que lidam com conteúdo variável e estruturado de maneira flexível, como blogs, sites de notícias e plataformas de e-commerce, podem se beneficiar da flexibilidade de esquema dos bancos de dados orientados a documentos
* **Aplicações de jogos**: Jogos frequentemente têm uma variedade de entidades e objetos com propriedades variáveis e mudanças frequentes na estrutura de dados. Bancos de dados orientados a documentos são úteis para armazenar e consultar dados de jogadores, inventários, progresso e outras informações relacionadas ao jogo.
* **Personalização e perfis de usuários**: Plataformas que precisam armazenar perfis de usuários com campos opcionais ou que variam com o tempo podem se beneficiar da flexibilidade de esquema oferecida por esses bancos de dados.
* **Logs e registros de eventos**: Armazenar logs e registros de eventos, como logs de servidor, registros de aplicativos, ou registros de transações, onde a estrutura de dados pode variar ao longo do tempo, é outro exemplo de uso adequado para bancos de dados orientados a documentos.

Exemplo em XML



# **Bancos de dados Chave e Valor**

Um banco de dados chave valor é uma estrutura para armazenamento de dados que funciona de forma muito similar a uma estrutura de dados do tipo mapa ou um dicionário, onde nós temos uma chave. Esta chave é um identificador para um registro, é como se ela guardasse o endereço para o valor que procuramos.

Utilizando uma analogia simples, pode se dizer que uma estrutura deste tipo é como a recepção de uma pousada. Ao chegar na pousada e informar a hospedagem a recepção nos entrega uma chave e esta chave contém o número do nosso quarto e nos permite acessá-lo. De posse dessa chave sempre conseguimos ter um acesso direto ao nosso quarto.

**Vantagens**

* A capacidade de distribuir dados entre vários servidores e o acesso direto a dados por meio de chaves únicas resultam em altas taxas de desempenho e escalabilidade.
* A simplicidade do modelo de chave-valor facilita a implementação e a manutenção do banco de dados.
* Não requer um esquema rígido, permitindo que os desenvolvedores ajustem os valores conforme necessário sem necessidade de alterações complexas na estrutura do banco de dados.

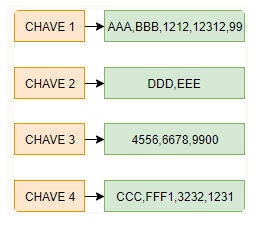
Desvantagens

* A simplicidade do modelo de chave-valor pode ser uma limitação para consultas complexas ou relacionamentos entre dados, tornando necessário o uso de técnicas adicionais para unir ou agregar dados.
* Alguns bancos de dados chave-valor podem não oferecer suporte robusto para transações complexas, o que pode ser uma limitação para certas aplicações.

**Casos de uso**

* Aplicações web para informações de sessão, como carrinhos de compra.
* Cache distribuído para melhorar a performance de aplicações (e.g., Redis, Memcached).
* Plataformas de streaming para armazenar preferências e histórico de usuários.
* Armazenamento de configurações de aplicação e metadados.
* Dados de jogo, como estados, pontuações e inventários.
* Dados de sensores e dispositivos IoT.
* Informações sobre produtos, inventário e histórico de compras.
* Perfis de usuários, conexões e interações.
* Filas de mensagens e sistemas de comunicação entre serviços.
* Informações sobre sessões de jogos multiplayer.

Exemplo chave e valor:

****

**Banco de dados em grafo**

Um banco de dados em grafos utiliza a teoria dos grafos para armazenar, mapear e consultar relações complexas entre dados. Em vez de usar tabelas ou documentos, utilizam nós (vértices) e arestas para representar e armazenar dados. Cada nó representa uma entidade (como uma pessoa ou um produto), e cada aresta representa uma relação ou conexão entre duas entidades.

Existem distintos tipos de bancos de dados de grafos, sendo que aqueles que mais se destacam:

**Grafos de propriedades:** essenciais para modelar relacionamentos entre dados, os grafos de propriedades proporcionam a capacidade de consultar e analisar informações com base em conexões específicas. Seu foco primário está nas análises detalhadas e consultas especializadas.

**RDF (Resource Description Framework):** configurado como um modelo de dados que enfatiza a integração entre conjuntos de dados, o RDF é empregado para representar informações de maneira processável por máquinas. Isso viabiliza a interconexão eficiente de dados, promovendo uma abordagem mais inteligente para a manipulação e interpretação das informações.

**Vantagens**

* Permite modelar dados e suas relações de maneira muito intuitiva e próxima à realidade.
* Excelente desempenho para consultas que navegam entre nós e arestas, mesmo em grandes conjuntos de dados.
* Fácil de adaptar e expandir o modelo de dados sem reestruturação complexa.

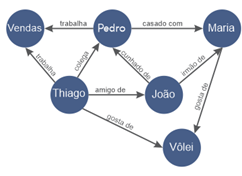
**Desvantagens**

* A administração e otimização de bancos de dados em grafos podem ser complexas, especialmente em grafos muito grandes.
* Pode haver uma sobrecarga desnecessária para consultas simples que não exploram a estrutura de grafos.
* O armazenamento de dados pode ser menos eficiente devido à necessidade de armazenar tanto nós quanto arestas.
* Em ambientes distribuídos, a resolução de conflitos pode ser complexa devido à natureza interconectada dos dados.

**Casos de uso**

* Serviços financeiros;
* Redes sociais;
* Manufatura;
* Varejo;
* Detecção de fraudes fiscais;
* Redes de computadores e operações de TI;
* Provedores de identidade e gerência de acessos;
* Telecomunicações;

Exemplo grafo:

****

**Bibliografia (Links)**

* <https://www.treinaweb.com.br/blog/tipos-de-banco-de-dados-nosql>
* <https://turing.pro.br/anais/ERBD-2013/artigos/pesquisa/111410.pdf>
* [https://medium.com/@fabiojmf/armazenamento-em-linha-vs-colunar-fd739b0de5e0#:~:text=desempenho%20das%20consultas.-,Desvantagens%3A,exigem%20alta%20taxa%20de%20transa%C3%A7%C3%B5es.](https://medium.com/@fabiojmf/armazenamento-em-linha-vs-colunar-fd739b0de5e0%23:~:text=desempenho%20das%20consultas.-,Desvantagens%3A,exigem%20alta%20taxa%20de%20transa%C3%A7%C3%B5es.)
* <https://www.cherrypickintegration.com/bancos-nao-relacionais/>
* <https://ilegra.com/blog/bancos-de-dados-chave-e-valor/>
* <https://micreiros.com/banco-de-dados-baseado-em-grafos-e-suas-principais-caracteristicas/>