Inatel
Instituto Nacional de Telecomunicações

BANCOS DE DADOS II

Prof. MSc. Renzo P. Mesquita
renzo@inatel.br

Bem-vindos ao curso de Bancos de Dados II

Critérios de Avaliação

NPA = NPT * 0.60 + NPL * 0.40;

NPT = (NP1 + NP2)/2;

NP1 = PT1*0.70 + SE1*0.30;

NP2 = PT2*0.70 + SE2*0.30;

Obs: NP1 (Caps. 1 e 2) e NP2 (Caps. 3 e 4).

NPL = (RP * 0.50 + PF * 0,50);

Resumo:

2 (duas) Provas da Teoria (PT);

Séries de Exercícios (SE);

Relatórios Práticos (RP);

Projeto Final (PF);

Bem-vindos ao curso de Bancos de Dados II

Critérios de Aprovação

```
...
if(NPT >= 60 && NPL >= 60)
{
    escreve("APROVADO!"); NFA=NPA;
}
else if(NPA < 30)
{
    escreve("REPROVADO!"); NFA=NPA;
}
else
{
    calculaNP3(NPA);
}
...
```

```
calculaNP3(int npa)
{
    NFA = (npa + NP3)/2;
    if(NFA >= 50)
    {
        escreve("APROVADO!");
    }
    else
    {
        escreve("REPROVADO!");
    }
}
```

Inatel
Instituto Nacional de Telecomunicações

BANCOS DE DADOS II

Cap.1 - Introdução ao NoSQL

Prof. MSc. Renzo P. Mesquita
renzo@inatel.br

Objetivos

- Apresentar de forma introdutória novos tipos de Bancos de Dados (BDs) que vêm ganhando espaço para armazenamento de grandes quantidades de dados;
- Conhecer características gerais e as novas filosofias dos BDs NoSQL;
- Compreender suas principais diferenças em relação aos BDs Relacionais (SQL);

Capítulo 1
Introdução ao NoSQL

1.1. O que é NoSQL?
1.2. Principais Objetivos do NoSQL;
1.3. O Modelo BASE;
1.4. Tipos de Bancos de Dados NoSQL;
1.5. O Teorema CAP;
1.6. O que geralmente falta nos Bancos NoSQL?
1.7. O que geralmente todos os Bancos NoSQL oferecem?
1.8. Quando usar Bancos Relacionais ou NoSQL?
1.9. Bancos NoSQL Populares.

1.1. O que é NoSQL?

- Acrônimo para "Not Only SQL";
- Bancos de Dados ou ferramentas de armazenamento que não seguem a popular e bem conhecida filosofia dos Bancos de Dados Relacionais (que utilizam de tabelas como estruturas de armazenamento e da linguagem SQL para definição/manipulação/consulta de dados);
- Bancos de dados feitos para trabalhar com "Big Data";
- Big Data = grande quantidades de dados (volume) que são gerados a todo momento (velocity) e com alta variedade (variety);
- São tipos de SGBD's (Sistemas de Gerenciamento de Bancos de Dados) que vêm ganhando cada vez mais reconhecimento no mercado;

1.2. Principais Objetivos do NoSQL

Apesar dos BDs SQL serem altamente úteis e populares, eles foram criados em uma época em que ainda não se existia o conceito de Big Data.

Por isso, os BDs NoSQL não vieram para substituí-los, mas para COMPLEMENTÁ-LOS em pontos específicos, como por exemplo:

1. Esquema de armazenamento de dados maleável (Schemaless);
   Em suma, não é necessário criar uma estrutura de um BD inicialmente (schema) para começar a usá-lo e novos tipos de dados podem ser adicionados com facilidade a qualquer momento;

2. Fácil escalabilidade (Escalabilidade Horizontal);
   Adicione e reorganize nós de armazenamento de dados com facilidade, sem a necessidade de parar o serviço quando isso acontecer;

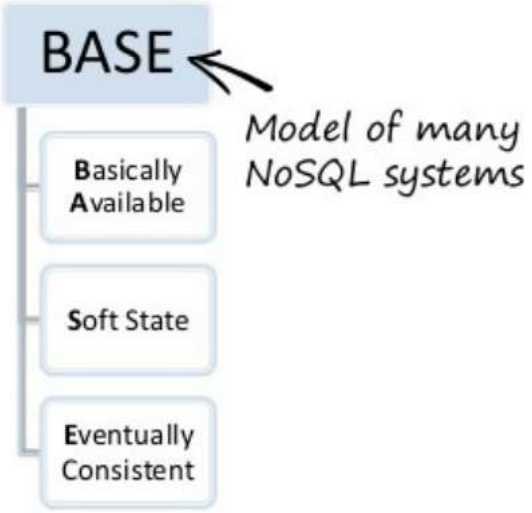
3. Seguem o modelo BASE de operação ao invés do ACID (seguido pelos BD's Relacionais).

Mas o que necessariamente seria o BASE?
Vamos compreendê-lo um pouco melhor ;)

1.3. O Modelo BASE

Modelo BASE

- BA - algumas partes do sistema ficam disponíveis mesmo após uma falha, mas não disponível por completo;
- S - ao acontecer uma operação de escrita, exclusão ou atualização em um dado do BD, o sistema não avisa todas as réplicas deste nó imediatamente, mas assim que for possível e de forma assíncrona;
- E - em um mesmo momento no tempo, pode se ter 2 valores diferentes para um mesmo dado que está em diferentes nós, até que o banco os sincronize.



Mas o que isso chega a diferenciar do modelo ACID dos Bancos Relacionais?

1.3. O Modelo BASE

ACID

- Alta consistência dos dados (garante que todas as réplicas de um dado tenham o mesmo valor em um momento);
- Isolamento das operações (operações mais demoradas, porém mais precisas);
- Banco responsável pelo gerenciamento das transações;
- Menor disponibilidade;
- Sistema mais conservador;
- Evolução dos tipos de dados a serem armazenados mais lenta (pois exige Esquemas);

BASE

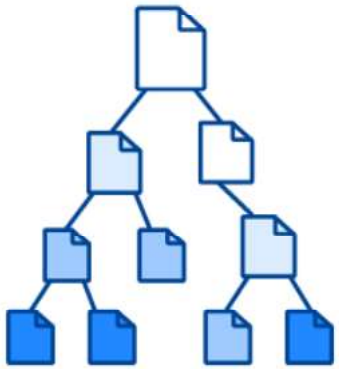
- Consistência mais fraca (como é o caso do Eventually Consistent que vimos);
- Disponibilidade de acesso em primeiro lugar;
- Transações devem ser implementadas pelo programador;
- Maior disponibilidade;
- Sistema mais simplificado;
- Evolução rápida dos tipos de dados a serem armazenados (Schemaless);

1.4. Tipos de Bancos de Dados NoSQL

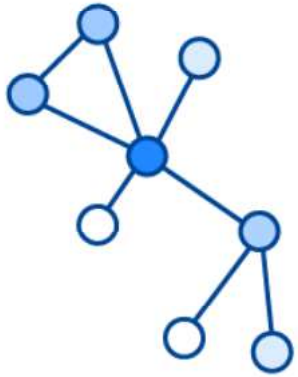
Diferente dos BDs Relacionais que utilizam de tabelas para armazenamento de dados e da linguagem SQL como padrão, BDs NoSQL incluem diferentes modelos de acesso e armazenamento de dados, cada um adequado para um uso específico.

Os principais modelos NoSQL são os BD's orientados a:

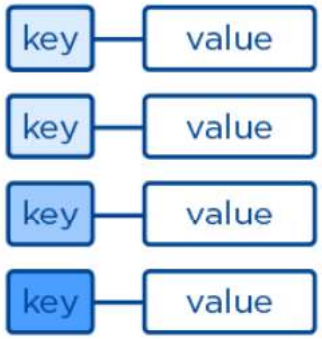
Documentos



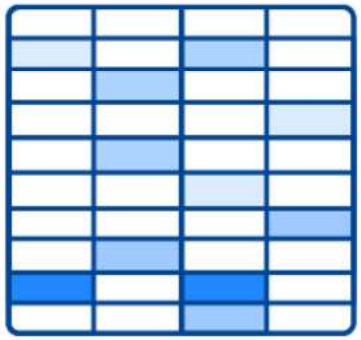
Grafos



Chave-Valor



Colunas



Vamos compreender estes modelos um pouco mais em detalhes?

1.4. Tipos de Bancos de Dados NoSQL

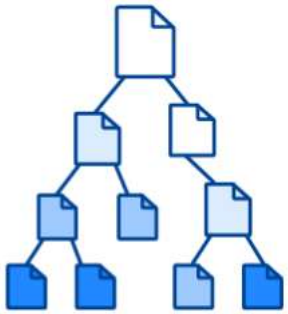
Documentos

- Em vez dos dados serem armazenados em tabelas, são armazenados em documentos (JSON documents) vagamente definidos;
- Permite a criação rápida de novos atributos em documentos individuais;
- Não possuem Esquema, mas uma Coleção de documentos;
- Não existe naturalmente relacionamentos entre documentos, um só documento pode possuir todas as informações necessárias (Evitando uma consulta adicional a outro documento);

Exemplo de uso:

Como armazenar de forma mais simples comentários e likes de posts em uma rede social?

Os comentários e likes se aplicam somente a um único post, de modo que não faz sentido separá-los dele. Por isso, todas estes dados poderiam gerar documentos que seriam guardados dentro de outros documentos, deixando a busca destes dados bem mais objetiva.



1.4. Tipos de Bancos de Dados NoSQL

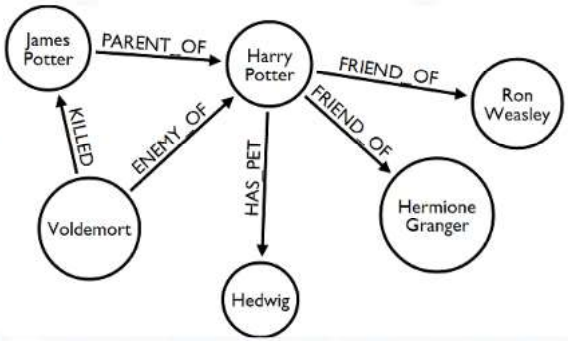
Grafos

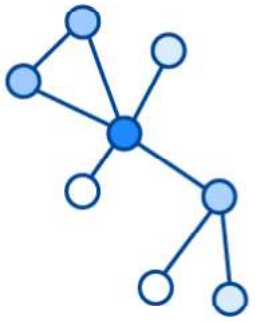
- Representa os dados ou esquema dos dados como grafos dirigidos;
- Interessante em casos quando a interconectividade dos dados são muito importantes;
- Formado por três componentes básicos: Nós (Vértices), Relacionamentos (Arestas) e propriedades (atributos dos nós e relacionamentos);

Exemplo de uso:

Como responder facilmente a relação entre alguns dos personagens de um filme?

Utilizando de outro BD, esta consulta poderia ser muito complexa devido a necessidade de muitas operações. Por meio de bancos orientados a grafos, esta pergunta seria respondida de forma muito simples.





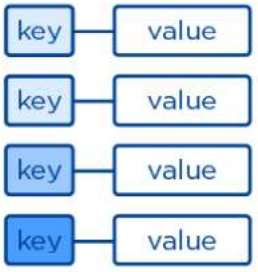
1.4. Tipos de Bancos de Dados NoSQL

Chave-Valor

- Armazena dados como um conjunto de pares de chave-valor, em que uma chave funciona como um identificador exclusivo (como uma tabela Hash, Map do Java ou Dictionary do Python);
- A chave e os valores podem ser qualquer coisa, desde objetos simples até objetos compostos complexos;
- Um dos modelos de BDs NoSQL de maior performance por conta da sua estrutura simples;
- Geralmente usados para criação de "cache" de dados que precisam ter um desempenho diferenciado.

Exemplo de uso:

Como guardar e acessar com rapidez dados que estão sendo utilizados por um usuário em uma sessão de acesso a uma plataforma web?



Um BD pro Chave-valor poderia ser um grande aliado neste processo, proporcionando uma navegação mais fluída (com menos atrasos) ao usuário.

1.4. Tipos de Bancos de Dados NoSQL

Colunas

- Enquanto BDs Relacionais são otimizados para operar sobre linhas, BDs Colunares são otimizados para recuperação rápida de colunas;
- Reduz expressivamente custo de I/O para operações no BD;
- Permite que dados sejam armazenados efetivamente, evitando consumir espaço quando existem colunas com valores nulos;
- Colunas são capazes de armazenar qualquer tipo de dado, desde que o dado possa ser transformado em um Array de bytes;
- Dados armazenados baseado em "Famílias de Colunas".

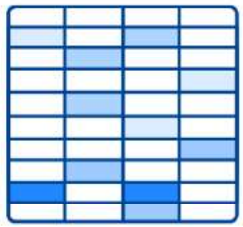
Exemplo de uso:

Como guardar e buscar com eficiência apenas informações específicas de uma pessoa?

Observe o conceito de Família de Colunas deixando os dados de uma pessoa de forma mais modular.

DADOS VISUALIZADOS NO FORMATO FAMÍLIAS DE COLUNAS

NAME	LOCATION	PROFILE
For row-key: 1 first_name: John last_name: Doe For row-key: 2 first_name: Jane	For row-key: 1 zip_code: 10001 For row-key: 2 zip_code: 94303	For row-key: 1 gender: male



1.5. O Teorema CAP

Teorema CAP

Não existe o BD dos sonhos! Tudo é uma questão de tradeoff. As forças e fraquezas de um BD distribuído, por exemplo, podem ser objetivamente descritas pelo Teorema CAP.

C - Consistency (Consistência);
A - Availability (Disponibilidade);
P - Partioning Tolerance (Tolerância ao Particionamento);

- O Teorema CAP é usado para definir o perfil de armazenamento de dados distribuídos de um sistema de Bancos de Dados;
- Destas 3 caracteristicas, um BD pode oferecer 2 ao mesmo tempo (CA, CP ou AP);
- Por exemplo, o MongoDB (NoSQL Orientado a Documentos) é um Banco CP, o Cassandra (NoSQL Orientado a Colunas) é um AP e geralmente os BDs Relacionais são CA;

1.6. O que geralmente falta nos Bancos NoSQL?

1. Suporte a Junções (Joins);

Operações de junções entre tabelas realizadas pelos BDs Relacionais. Essas operações em certas situações são muito úteis, porém, são operações lentas;

2. Suporte a Transações (Transactions);

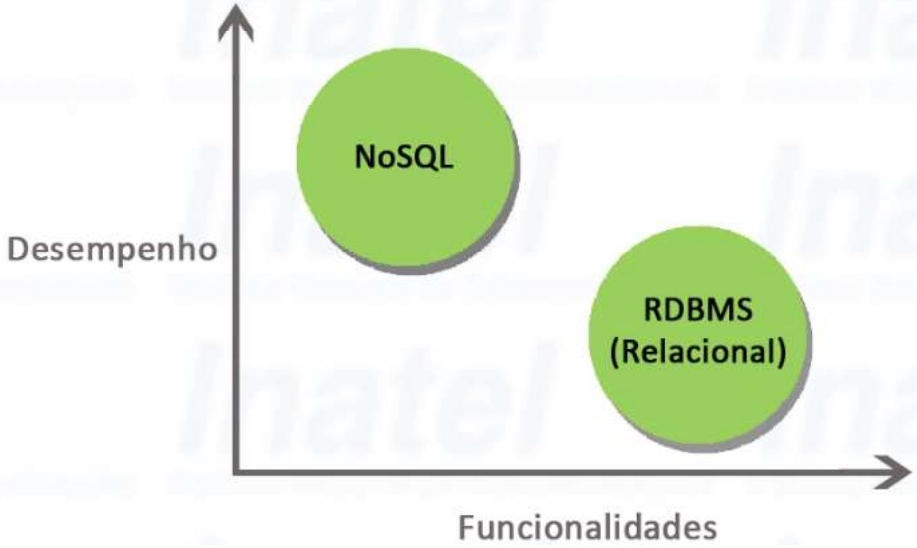
Operações que exigem atomicidade geralmente devem ser desenvolvidas a nível de aplicação. Falta de suporte para realização de operações commit e rollback;

3. Suporte a Restrições (Constraints);

Como vimos antes, restrições permitem que o banco de dados tome ações automáticas visando manter a sua integridade;

1.7. O que geralmente todos os Bancos NoSQL oferecem?

1. Linguagem Própria para consulta e manipulação dos dados (Query Language);
2. No geral possuem melhor desempenho comparado aos Bancos de Dados Relacionais, principalmente em bases de dados distribuídas;
3. Escalabilidade Horizontal;



1.8. Quando usar Bancos Relacionais ou NoSQL?

Use RDBMS quando...	Use NoSQL quando...
Suas aplicações forem centralizadas (ERP, CRM)	Suas aplicações forem descentralizadas (Web, Mobile, Big Data, IoT)
Alta disponibilidade moderada for necessária	Quando a disponibilidade tiver que ser contínua, sem interrupção
Dados gerados em velocidade moderada	Dados gerados em alta velocidade (sensores)
Dados forem gerados a partir de poucas fontes	Dados forem gerados a partir de múltiplas fontes
Dados forem estruturados	Dados forem semi ou não-estruturados
Transações complexas	Transações simples
For necessário manter moderado volume de dados	For necessário manter alto volume de dados

SGBDs Relacionais já conhecemos vários.
Quais seriam alguns BDs populares NoSQL?

1.9. Bancos NoSQL Populares

Documentos

Quem usa? eBay, Adobe, Google...

Colunas

Apple, Facebook, Netflix...

Grafos

Quem usa? Linkedin, Microsoft, IBM...

Chave-Valor

Twitter, Github, Flickr...

FIM
DO
CAPÍTULO 1

EXERCÍCIOS