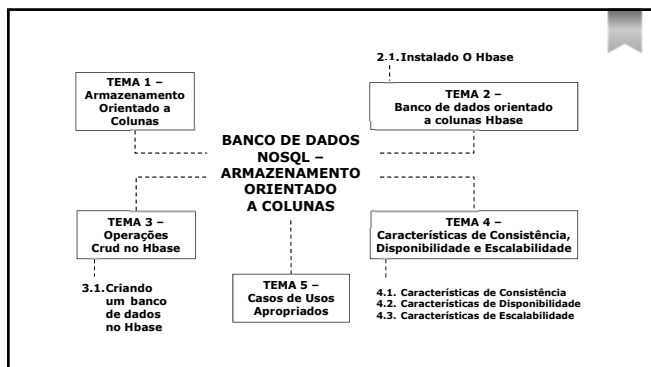


## Aula 4

### Banco de Dados NoSQL

Prof. Alex Mateus Porn

### Conversa Inicial



### Armazenamento orientado a colunas

#### Armazenamento orientado a colunas

- Apresenta conceitos muito similares aos bancos de dados relacionais
- Formado basicamente pela composição de linhas e colunas
- Pode ser considerada como a mais complexa entre as quatro estruturas NoSQL que estudamos

#### Armazenamento orientado a colunas

- Exemplo: comparação com um BD Relacional

CLIENTES	
id_cliente	int
nome	varchar (200)
nascimento	date
telefone	varchar (30)
renda	decimal (5,2)

Fonte: Elaborado com base em Marquesone (2017, p. 50)

### Armazenamento orientado a colunas

- Exemplo: comparação com um BD Relacional
  - Todos os registros devem contemplar as cinco colunas, mesmo que preenchidas com Null
  - Alto processamento de consultas à medida que o banco de dados cresce
  - Alto impacto em todos os registros ao adicionar um novo campo na tabela

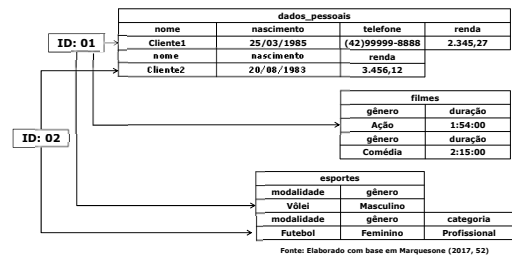
### Armazenamento orientado a colunas

- Exemplo: banco de dados orientado a colunas
  - Busca resolver principalmente o problema de escalabilidade e flexibilidade no armazenamento de dados
  - Primeiramente, define-se as famílias de colunas e, em seguida, as colunas
  - Cada família permite a flexibilidade de inserir quantas colunas forem necessárias para cada registro armazenado, sem precisar alterar a estrutura dos dados já armazenados

### Exemplificando

- Imaginemos um cenário para armazenamento das preferências do usuário em um aplicativo de compras on-line, utilizando um banco de dados orientado a colunas:
  - dados\_pessoais
  - filmes
  - esportes

### Exemplificando



### Características

- Fornecer flexibilidade e grande escalabilidade
- O registro de um item pode ter informações gravadas em diversas famílias de colunas
- As colunas podem estar armazenadas em diferentes servidores, pois os dados são armazenados fisicamente em uma sequência orientada a colunas e não linhas

### Banco de dados orientado a colunas HBase

## Banco de dados HBase

- SGBD distribuído orientado a colunas
- Utilizado principalmente com grande quantidade de dados e muitos atributos
- Desenvolvido para funcionar sobre o Hadoop:
  - Plataforma para projetos distribuídos
- Disponibiliza uma versão para gerenciamento do banco de dados local (standalone)

## Instalando o HBase



## Instalando o HBase

- Criar um diretório específico para o Hbase:
  - Ex: /home/user/HBase
- Devemos configurar dois arquivos:
  1. /home/usuário/hbase/conf/hbase-env.sh
  2. /home/usuário/hbase/conf/hbase-site.xml

## Configurando o arquivo hbase-env.sh

- Configurar o endereço da variável de ambiente JAVA\_HOME

```
# This script sets variables multiple times over the course
# of starting an hbase process,
# so try to keep things idempotent unless you want to take
# an even deeper look
# into the startup scripts (bin/hbase, etc.)

# The java implementation to use. Java 1.8+ required.
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

# Extra Java CLASSPATH elements. Optional.
# export HBASE_CLASSPATH=

# The maximum amount of heap to use. Default is left to JIMR
# or -Xmx1024m
```

## Configurando o arquivo hbase-site.xml

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>file:/home/usuario/hbase</value>
  </property>
</configuration>
```

## Inicializando o HBase

```
alex@AlexDell: ~/hbase/bin
Arquivo Editar Ver Pesquisar Terminal Ajuda
(base) alex@AlexDell:~/hbase/bin$ cd /home/alex/hbase/bin/
(base) alex@AlexDell:~/hbase/bin$

alex@AlexDell: ~/hbase/bin
Arquivo Editar Ver Pesquisar Terminal Ajuda
(base) alex@AlexDell:~/hbase/bin$ ./start-hbase.sh
running master, logging to /home/alex/hbase/bin/../logs/hbase-alex-master-AlexDell.out
(base) alex@AlexDell:~/hbase/bin$
```

## Visualizando o HBase

```
alex@AlexDell: ~/hbase/bin
Arquivo Editar Ver Pesquisar Terminal Ajuda
(base) alex@AlexDell:~/hbase/bin$ ./start-hbase.sh
running master, logging to /home/alex/hbase/bin/../logs/hbase-alex-master-AlexDell.out
(base) alex@AlexDell:~/hbase/bin$ jps
31760 HMaster
873 Jps
(base) alex@AlexDell:~/hbase/bin$
```

- http://localhost:16010



## Operações CRUD no HBase

### Interface Shell no HBase

- Interface de linha de comando (CLI) para criação e gerenciamento dos bancos de dados:
- ./hbase shell

```
alex@AlexDell: ~/hbase/bin
Arquivo Editar Ver Pesquisar Terminal Ajuda
(base) alex@AlexDell:~/hbase/bin$ ./hbase shell
08/11/25 20:27:42,643 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.2.5, r776d801278d34267055c0c0a12474596283f64c, 2020年 05月 21日 星期四 18:13:40 CST
Took 0.0005 seconds
hbase(main):001:0>
```

### Criação do banco de dados

- O primeiro passo consiste em criarmos uma tabela e a família de colunas desejada
- Criar a tabela Clientes e as famílias de colunas "dados\_pessoais" e "endereco"
- create 'clientes', 'dados\_pessoais', 'endereco'



### Criação do banco de dados

- Cada família de colunas pode ter quantos atributos forem necessários. Em nosso exemplo, cada família de colunas ficará da seguinte forma:
- dados\_pessoais – nome, idade
- endereco – rua, cidade

### Inserindo dados na tabela Clientes

- put 'clientes', '1', 'dados\_pessoais:nome', 'Cliente1'
- put 'clientes', '1', 'dados\_pessoais:idade', '35'
- put 'clientes', '1', 'endereco:rua', 'Rua 18'
- put 'clientes', '1', 'endereco:cidade', 'Curitiba'

### Consultando os dados

- Visualizar os dados inseridos em uma tabela:
  - scan 'nome\_da\_tabela'
- Localizar registros específicos em uma tabela:
  - get 'nome\_da\_tabela', 'id\_do\_registro'

### Atualizando os dados

- put 'clientes', '1', 'dados\_pessoais:idade', '50'

### Excluindo registros

- Excluir uma célula de um registro:
  - delete 'clientes', '1', 'dados\_pessoais:idade'
- Excluir um registro completo:
  - deleteall 'clientes', '1'

### Características de consistência, transações e disponibilidade

### Características de consistência

- Todas as leituras e gravações são roteadas por meio de um único servidor de região
- Isso garante que todas as gravações ocorram em ordem e todas as leituras acessem os dados confirmados mais recentemente
- Se o servidor ficar indisponível, as regiões das tabelas hospedadas no servidor de região ficarão indisponíveis por algum tempo, até que sejam recuperadas

(Claudera, 2012)

### Características de consistência

- Consistência na linha do tempo:
  - Garante a consistência por servidores de região no modo secundário
  - Os dados são visualizados na mesma ordem, mas podem estar um pouco desatualizados. Apenas o servidor de região primário tem a garantia de ter os dados mais recentes

(Claudera, 2012)

### Características de consistência

- **Consistência forte:**
  - Os dados mais recentes são sempre veiculados
  - No entanto, pode aumentar muito a latência no caso de uma falha do servidor de região, porque apenas o servidor de região primário tem garantia de ter os dados mais recentes

(Claudera, 2012)

- Garante a disponibilidade dos dados de vários modos:
  - Distribuição de dados em vários nós
    - a perda de um nó afeta somente os dados armazenados nesse nó
  - Garante que a perda de um único nó não resulte na perda de disponibilidade de dados
  - O formato HFile armazena dados diretamente no HDFS. O HFile pode ser lido ou escrito por diversas tecnologias Apache, permitindo análises profundas no HBase sem movimento de dados (Filipa, 2020)

- Fornece um recurso chamado *replicação de região*:
  - Para cada região de uma tabela, pode haver várias réplicas que são abertas em servidores de região diferentes
  - Por padrão, a replicação da região é definida como 1, portanto, apenas uma única réplica da região é implantada e não há mudanças no modelo original
  - Se a replicação de região for definida como 2 ou mais, o mestre atribuirá réplicas das regiões da tabela

(Claudera, 2012)

### Características de escalabilidade

- Escala linearmente quando lida com grandes conjuntos de dados formados por bilhões de linhas e milhões de colunas
- Possui uma recomendação mínima de 5 nós por cluster Hadoop
- Permite escalonar com facilidade para centenas de nós de acordo com a demanda

### Casos de usos apropriados

### Casos de usos apropriados

- Conforme Shon (2014):
  - Análise em lote de dados de logs, devido a sua otimização para leituras e varreduras sequenciais
  - Captura de métricas em tempo real de aplicativos, servidores, preferências do usuário, entre outros

### Casos de usos apropriados

- Conforme Shon (2014):
- Facebook – gerenciamento de mensagens de usuários
- Pinterest – fornecer feeds personalizados, capturar dados e potencializar seu processo de recomendações
- Explorys – capturar bilhões de pontos anônimos de dados clínicos, operacionais e financeiros. A Explorys usa essa plataforma para ajudar seus clientes a obter atendimento de qualidade, minimizar custos e mitigar riscos

### Casos de usos apropriados

- Para Filipa (2020):
- Cenários que exigem análise em tempo real e dados tabulares para aplicativos de usuários finais
- Serviços de segurança na web para manter bilhões de traços de eventos e registros de atividades dos desktops dos seus clientes todos os dias

### Referências

### Referências

- MARQUESONE, R. Big Data: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código, 2017.
- SHON, P. Apache HBase Explained in 5 Minutes or Less. Cloudera, 2014. Disponível em: <<https://www.cloudera.com/insights/apache-hbase-explained-5-minutes-less/>>. Acesso em: 7 maio 2021

- FILIPA, S. Apache HBase: O que é, Conceitos e Definições. Cetax, 2020. Disponível em: <<https://www.cetax.com.br/blog/o-que-e-o-apache-hbase/>>. Acesso em: 7 maio 2021
- Introduction to HBase High Availability, Cloudera, c2012-2020. Disponível em: <[https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.6.0/bk\\_hadoop-high-availability/content/ha-hbase-intro.html#:~:text=Hbase%2C%20architecturally%2C%20has%20had%20a,the%20most%20recently%20committed%20data](https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.6.0/bk_hadoop-high-availability/content/ha-hbase-intro.html#:~:text=Hbase%2C%20architecturally%2C%20has%20had%20a,the%20most%20recently%20committed%20data)>. Acesso em: 7 maio 2021