

Claretiano Centro Universitário

Hannah dos Santos Horta
RA 8098412

Banco de dados NoSQL – Portfólio 2º Ciclo

Curso de Análise de Dados

Belo Horizonte
Outubro/2020

Hannah dos Santos Horta
RA 8098412

Banco de Dados NoSQL – Portfólio 2º Ciclo

Curso de Análise de Dados

Pesquisa e representação gráfica dos
principais tipos de bancos de dados
NoSQL e modelos de armazenamento
de dados.

Prof. Orientador: Jane Aparecida Meneguelli

Belo Horizonte
Outubro/2020

Sumário

Introdução	4
Bancos de Dados NoSQL	5
Modelos de Distribuição de Dados	7
Conclusão	11

Introdução

Este trabalho apresenta uma breve pesquisa a respeito dos tipos de bancos de dados NoSQL com representações gráficas explicativas e modelos de armazenamento de dados também com suas respectivas representações gráficas.

Um pouco sobre banco de dados NoSQL

Os bancos de dados NoSQL fornecem uma variedade de modelos de dados, como chave-valor, documento e gráfico, que são otimizados para performance e escala. Sendo projetados para vários padrões de acesso aos dados que incluem aplicativos de baixa latência. Os bancos de dados de pesquisa NoSQL são projetados para análise de dados semiestruturados.

1 – Representação Gráfica dos Principais Modelos de Banco de Dados NoSQL

NoSQL Chave-valor

Um banco de dados de chave-valor é um tipo de banco de dados não relacional que usa um método de chave-valor simples para armazenar dados. Um banco de dados de chave-valor armazena dados como um conjunto de pares de chave-valor em que uma chave funciona como um identificador exclusivo. A chave e os valores podem ser qualquer coisa, desde objetos simples até objetos compostos complexos. Bancos de dados de chave-valor são altamente particionáveis e permitem escalabilidade horizontal que outros tipos de bancos de dados não conseguem alcançar.

Exemplo: Amazon DynamoDB

Veja a representação gráfica desse tipo de BD abaixo:

Chave	Valor
carro_3345_cor	preto
carro_3345_pneu	17
carro_3365_cor	branco
carro_3365_pneu	15
carro_4560_peso	1215
carro_4715_ano	2016

NoSQL Orientado a Documentos

Um banco de dados orientado a documentos, ou armazenamento por documentos, é um modelo de banco de dados projetado para armazenar, recuperar e gerenciar informações orientadas a documentos, também conhecidas como dados semi-estruturados.

Exemplo: MongoDB

Veja abaixo uma imagem de um arquivo de armazenamento de dados do MongoDB:

```

{
  "id": 49,
  "País": "Alemanha",
  "Região": "Europa",
  "População":
  "PrincipaisCidades": [
    {
      "NomeCidade": "Berlin",
      "População": 3610156,
    },
    {
      "NomeCidade": "Hamburg",
      "População": 1746342,
    }
  ]
}

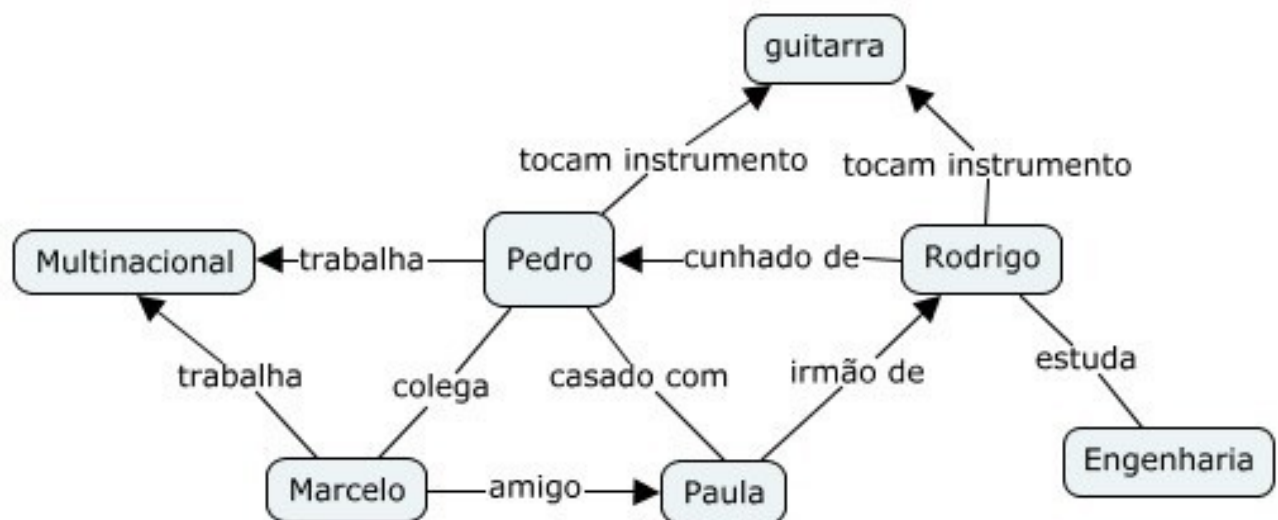
```

NoSQL Orientado por Grafos

Um banco de dados baseado em grafos (graph database), também chamado de banco de dados orientado por grafos, é um tipo de banco de dados NoSQL que usa a Teoria de Grafos para armazenar, mapear e consultar relacionamentos. Um banco de dados baseado em grafos é essencialmente uma coleção de nós e arestas.

Exemplo: Neo4j

Veja na imagem abaixo um exemplo, de como funciona o relacionamento e armazenamento de dados, em um banco desse modelo:



NoSQL Orientado a Colunas

Bancos de dados orientados a colunas armazenam os dados em “column Families” como linhas que possuem muitas colunas associadas a uma chave desta linha, “column Families” são grupos de dados relacionados que normalmente são acessados em conjunto, por exemplo, para a “família de colunas” clientes normalmente teríamos as informações do perfil deste cliente armazenadas, porém não teríamos os pedidos realizados por ele.

Exemplo: Cassandra

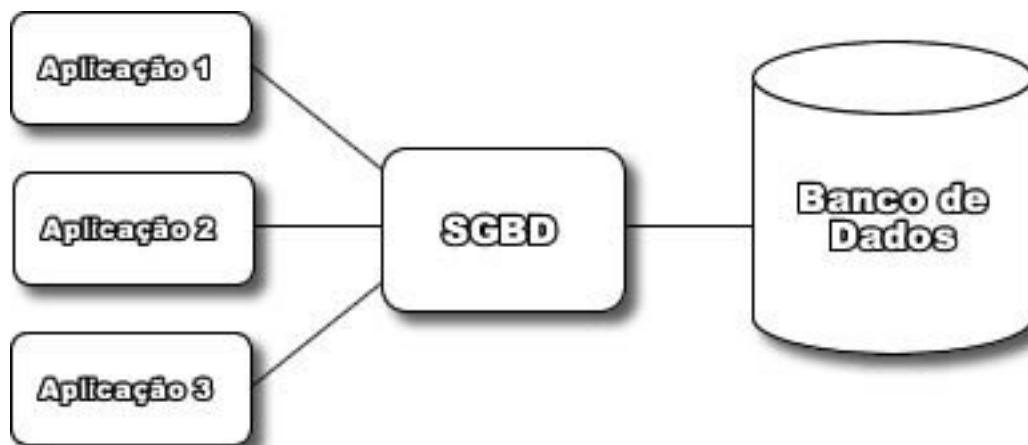
Veja na figura abaixo a representação de como funciona o armazenamento nos bancos de dados de famílias de colunas:

Tabela			
Família de coluna 1		Família de coluna 2	Família de coluna 3
Coluna 1	Coluna 2	Coluna 3	Coluna 4
#1 {Chave: Valor, Chave: Valor}	#1 {Chave: Valor, Chave: Valor}	#1 {Chave: Valor, Chave: Valor}	#1 {Chave: Valor, Chave: Valor}
#2 {Chave: Valor, Chave: Valor}	#2 {Chave: Valor, Chave: Valor}	#2 {Chave: Valor, Chave: Valor}	#2 {Chave: Valor, Chave: Valor}

2 – Representação Gráfica dos Modelos de Distribuição de Dados

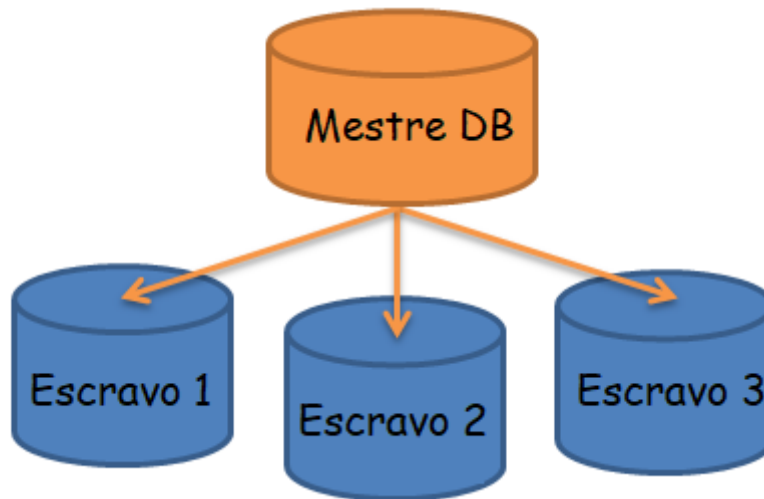
Único Servidor

Estrutura cliente/servidor, onde várias estações (computadores) fazem acesso a uma única base de dados.



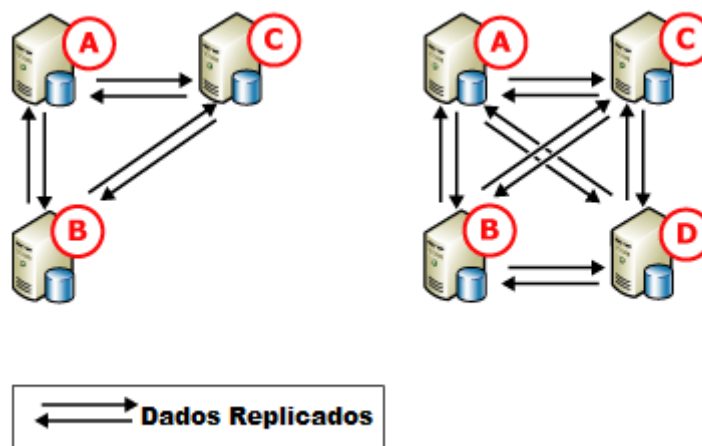
Replicação Mestre Escravo

A replicação Mestre a Escravo fornece replicação automática dos tipos de dados customizados selecionados a partir de um ou mais pontos principais designados para um ou mais pontos escravos pré-determinados.



Replicação Ponto a Ponto (P2P)

A replicação Ponto-a-Ponto é uma solução orientada para *Escalabilidade Horizontal em alta disponibilidade*, permitindo gerenciar muitas cópias de dados em mais instâncias do servidor, chamadas "nós". Com base na replicação transacional, a replicação Ponto-a-Ponto propaga mudanças de dados para todos os nós quase em tempo real. Graças a essa redundância, os aplicativos que solicitam a escalabilidade horizontal das operações de leitura podem dividir os processos de leitura dos clientes em múltiplos nós, e acessar dados de forma muito mais rápida. Este tipo de replicação é absolutamente um dos procedimentos de replicação mais executados e "à prova de falhas", pois a falta de disponibilidade de um ou mais nós não afeta o funcionamento e a disponibilidade de todo o sistema em si.

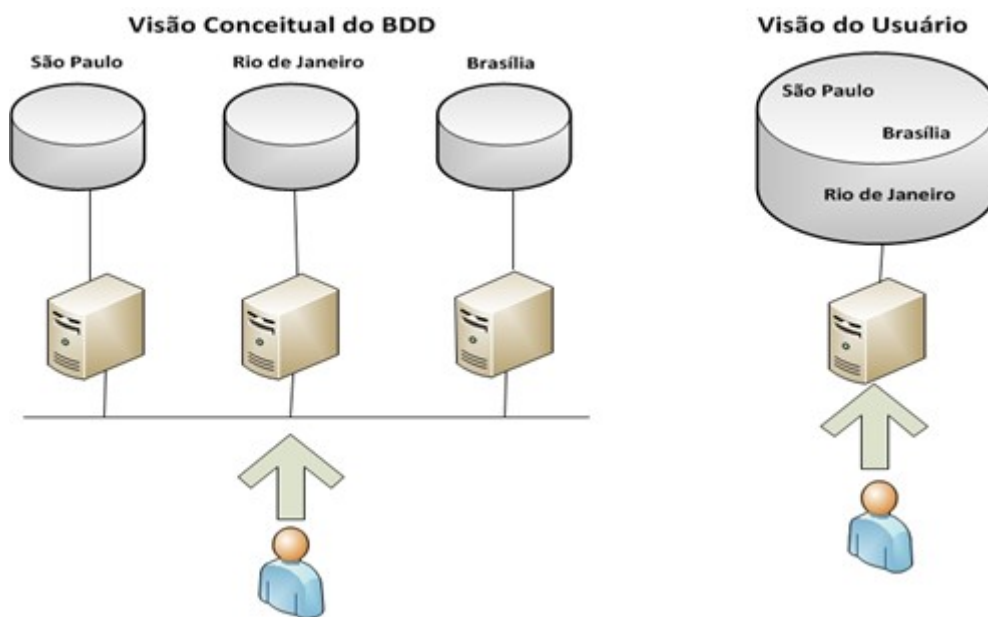


Fragmentação de Dados

Uma relação é dividida em fragmentos, onde cada fragmento contém informação suficiente para permitir a reconstrução da relação original.

Existem duas formas de fazer a fragmentação:

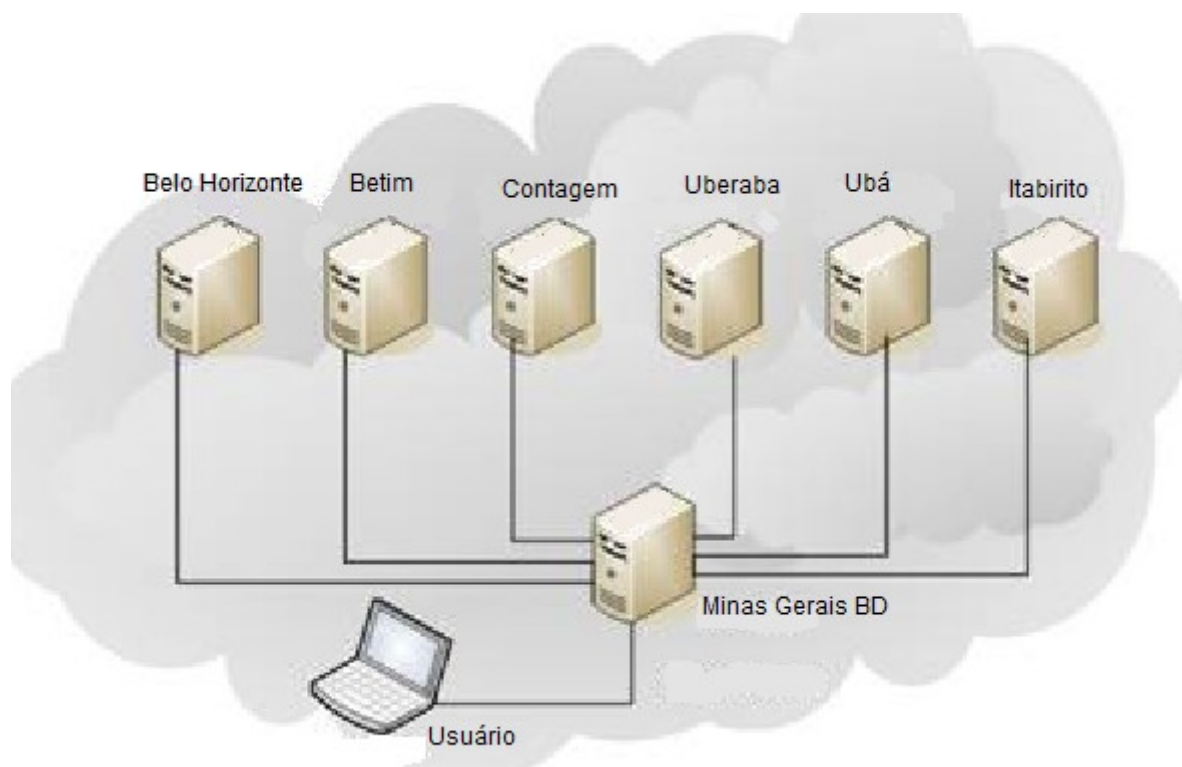
- Fragmentação Horizontal: divide a relação separando as tuplas de r em dois ou mais fragmentos.
- Fragmentação Vertical: divide a relação pela decomposição do esquema R da relação r .



Fragmentação com Replicação de Dados

As técnicas de fragmentação e replicação podem ser aplicadas sucessivamente a uma mesma relação. Um fragmento pode ser replicado, e as réplicas podem ser fragmentadas novamente e assim por diante.

Veja na imagem abaixo um exemplo onde os dados replicados são fragmentados, e os dados mestre (originais) armazenados em um único local.



Conclusão

Com o desenvolvimento deste portfólio eu pude absorver melhor os conceitos de armazenagem de dados dos bancos NoSQL e compreender as suas diferenças usuais. Pude também aprender mais sobre os modelos de distribuição de dados e entender as suas aplicações reais como a replicação e a fragmentação, que não necessidades atuais para todos que trabalham com dados diariamente.