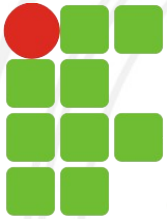


INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

NoSQL

Prof. Msc Denival A. dos Santos

Introdução

- Com um aumento tão drástico, quer os motores de busca e mais tarde as redes sociais, tiveram de lidar com quantidades enormes de dados a processar, alcançando a ordem dos Petabytes (o Google lida com 24 Petabytes de informação por dia e uma rede social como o Twitter lida com 4 Petabytes de informação por ano (Boyce, 2010)).
- Uma base de dados por si só não consegue lidar com tanta informação, isto é, as bases de dados relacionais não foram feitas para lidar com um enorme volume de informação num curto espaço de tempo, nem preveem o aumento das suas capacidades de processamento, de um dia para o outro, sem ser necessário uma reconfiguração.

Limitações do modelo Relacional

- As limitações do modelo relacional começaram a ser evidentes à medida que o volume de dados começou a aumentar.
- De forma a aumentar o desempenho de uma base de dados relacional, as únicas opções são:
  - **Escalar verticalmente** a base de dados, que consiste em substituir o hardware existente por um mais rápido; e
  - **Escalar horizontalmente** que consiste em distribuir a base de dados por múltiplas Instâncias.
- Das duas soluções anteriores a mais adotada era o escalonamento vertical, mas, hoje em dia, esta solução tende a não ser a mais viável pois para processar informação na ordem dos Petabytes e a crescer, é necessário hardware extremamente caro e mesmo assim não é suficiente.

Introdução

Problema

- A medida que o número de linhas numa base de dados relacional aumenta drasticamente, o desempenho da base de dados diminui, (a escalabilidade de uma base de dados relacional tem um limite) pois, à medida que a base de dados é distribuída, a complexidade dos joins com tabelas que não estejam na mesma máquina aumenta.
- Do mesmo modo, à medida que o número de instâncias da base de dados aumenta, a garantia das propriedades ACID em transações distribuídas fica também muito difícil de gerir. Para agravar, pode ainda ser necessária a contratação de mais DBA's (Administradores de Base Dados) para gerirem toda esta complexidade.

NoSQL

- As bases de dados NoSQL foram desenvolvidas para serem fáceis de distribuir. Isso implica um diferente modelo de consistência em contrapartida ao comum ACID. Como tal, as bases de dados NoSQL fazem uso da consistência eventual que nem sempre a consistência dos dados é mantida. Esta característica tem embasamento no teorema CAP (Consistency, Availability e Partition tolerance) que afirma que em um dado momento só é possível garantir duas destas três propriedades, que seriam Consistência, Disponibilidade e tolerância à partição .
- NoSQL engloba todas as bases de dados que não seguem os princípios das bases de dados relacionais e que estão relacionadas com grandes volumes de dados.
- A junção de “No” e “SQL” deveria significar algo como “Não SQL” mas na prática NoSQL significa uma coleção de produtos e conceitos sobre a manipulação de grandes volumes de dados sem usar unicamente SQL, ao invés de um produto único que contraria de alguma forma o SQL.

NoSQL - características

Schema-Free

- Schema por definição é um conjunto de fórmulas que descrevem a base de dados, relações e constrangimentos codificados na sintaxe dessa base de dados. Esta estrutura é fruto de um processo chamado normalização para garantir que nas tabelas possam ser inseridos dados sem anomalias (inconsistência de dados, redundância, etc.)..
- Schema free (livre de esquema) nada mais é do que ter uma base de dados que não tem uma estrutura fixa isto é, ter uma base de dados que não obriga a que os dados obedeçam à rigidez de um schema definido A Priori.

NoSQL - características

Schema-Free - Desvantagem

- Não ter um schema é possível ter dados redundantes ou até incoerentes.
- Os dados passam a estar menos organizados.
- A garantia das propriedades ACID deixa de ser válida devido à não existência de um schema e disso deriva um modelo de consistência diferente, consistência eventual ao invés de uma consistência imediata como nas bases de dados relacionais.
- No entanto, todos estes “males”, por assim dizer, são necessários porque obriga a abrir mão, neste caso, da Consistência dos dados para ganhar em Tolerância a falhas, alta disponibilidade e distribuição dos dados - requisitos inerentes a grandes volumes de dados.

NoSQL - Características

API Simples

- O acrônimo API significa Interface de Programação Simples que na verdade são um conjunto de funções que um software disponibiliza para ser usado por outro.

Produto
Hadoop/HBase
Cassandra
HyperTable
Mongo
CouchDB
RavenDB
Azure Table Storage
RIAK
Chrodless
Neo4J

NoSQL - Modelos

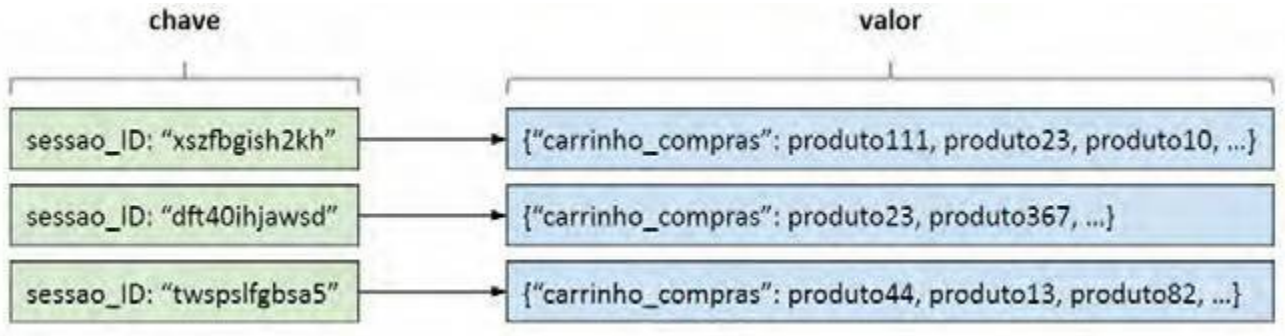
Os modelos NoSQL são divididos em 4 grandes tipos:

- Key-value (chave valor)
- Document (documentos)
- Column-family (famílias de colunas)
- Graph (grafos)

NoSQL - Modelos

Chave valor

- Todos os registros fazem parte da mesma coleção de elementos, e a única coisa que todos eles tem em comum é uma chave única.
- Os demais modelos derivam dele.
- Este modelo é otimizado para ambientes em que seja necessário um bom mecanismo de cache tal como: categorias de produtos, listas de compras, produtos mais vendidos, etc.



NoSQL - Modelos

Chave valor - Exemplos

- DynamoDB
- Redis
- Riak
- Memcached

NoSQL - Modelos

Documentos

- Considerado uma extensão do banco de dados orientado a chave-valor.
- É provavelmente o mais popular.
- Nele, cada entrada na base corresponde a um documento.
- Podemos definir documentos como sendo estruturas flexíveis que podem ser obtidas por meio de dados semiestruturados, como XML e Json.

Documento XYZ

```
{
   "estado": "PA",
   "municipio": "Altamira",
   "localizacao": "Lat 03°13'00'' Long 052°13'00'' W Alt 74,04 m",
   "atributo": "Pluviosidade",
   "periodo": "01/01/2010 a 31/12/2015",
   "coleta": [
         {
               "valor": "40",
               "data": "12/05/2010 12:31:09"
         }
   ]
}
```

NoSQL - Modelos

Documento - Exemplos

- CouchBase
- CouchDB
- MarkLogic
- MongoDB
- Firebase

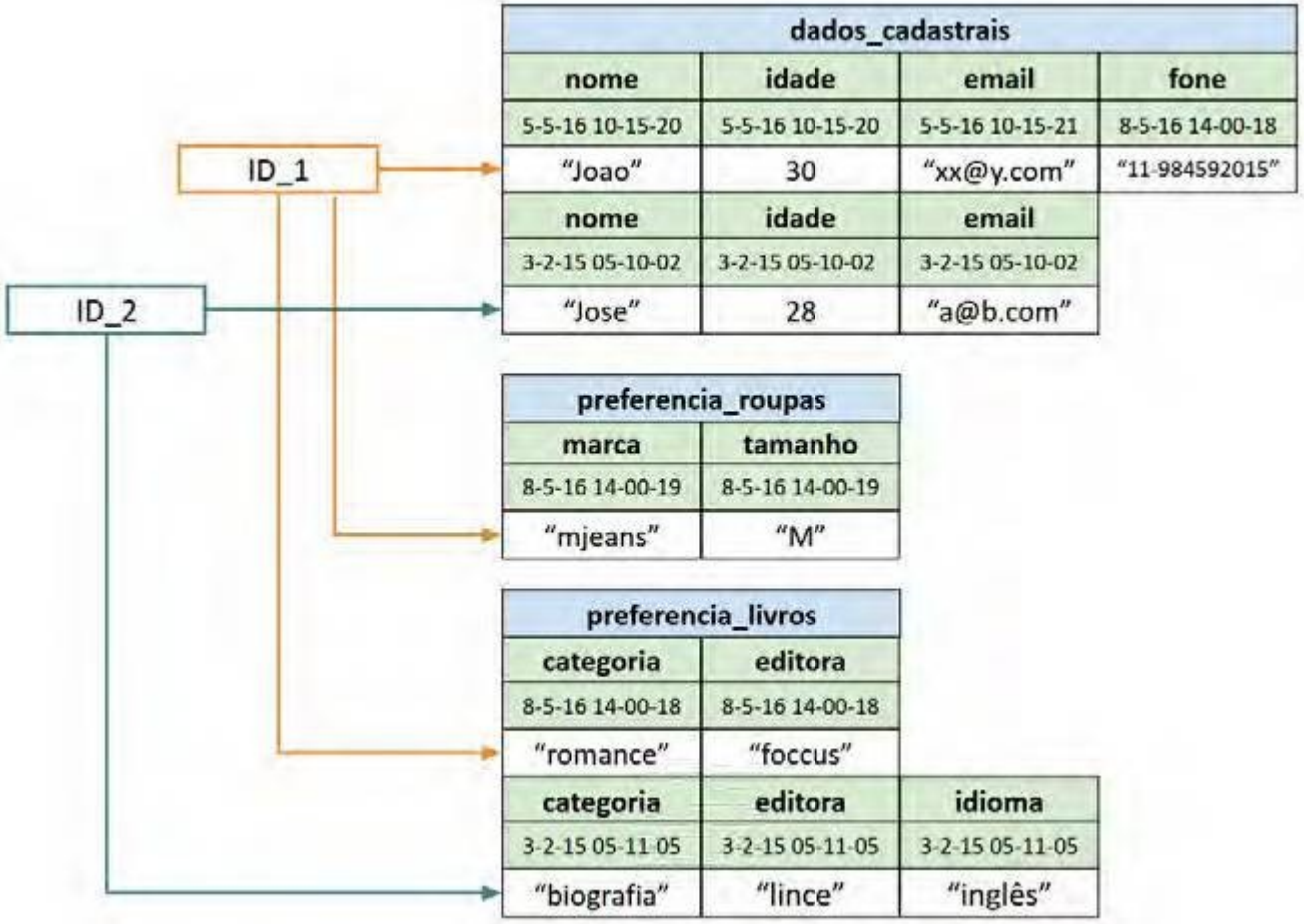
NoSQL - Modelos

Colunas

- De todos é o mais complexo. Sendo uma evolução do chave-valor
- O responsável pela modelagem de dados define o que é chamado de "famílias de colunas".
- As famílias de colunas são organizadas em grupos de itens de dados que são frequentemente usados em conjunto em uma aplicação.

NoSQL - Modelos

- Coluna



NoSQL - Modelos

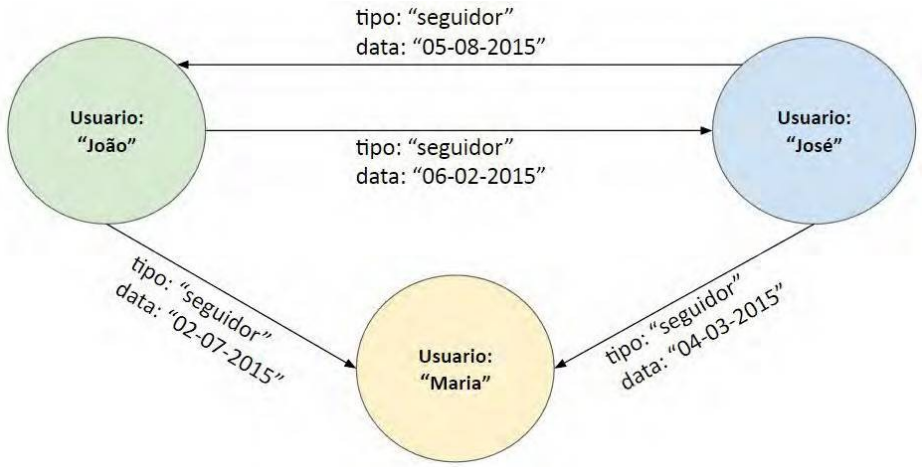
Coluna - Exemplos

- Accumulo
- Cassandra
- Hbase
- Hypertable

NoSQL - Modelos

Grafo

- É um modelo que se baseia na teoria dos grafos.
- Utiliza conceitos de vértices e arestas para armazenar os dados dos itens coletados (como pessoas, cidades, produtos e dispositivos) e os relacionamentos entre esses dados, respectivamente.
- Oferece maior desempenho nas aplicações que precisam traçar os caminhos existentes nos relacionamentos entre os dados, como por exemplo identificar como um conjunto de amigos está conectado em uma rede.



NoSQL - Modelos

Grafo - Exemplos

- AllegroGraph
- ArangoDb
- InfoGrid
- Neo4J
- Titan