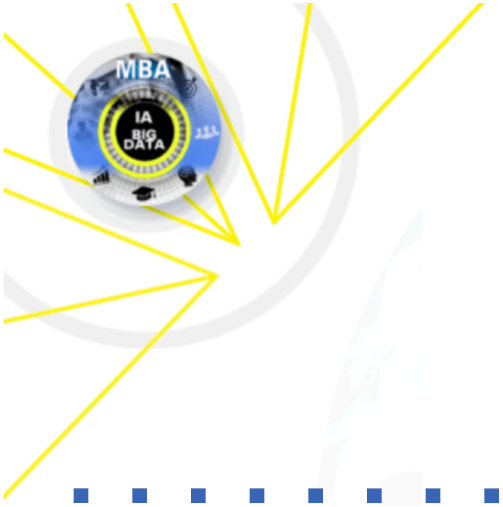



# NoSQL e o processamento de dados em larga escala (Parte 1)

Prof. Dr. Robson L. F. Cordeiro
robson@icmc.usp.br

USP-ICMC
GBDI - Grupo de Bases de Dados e de Imagens

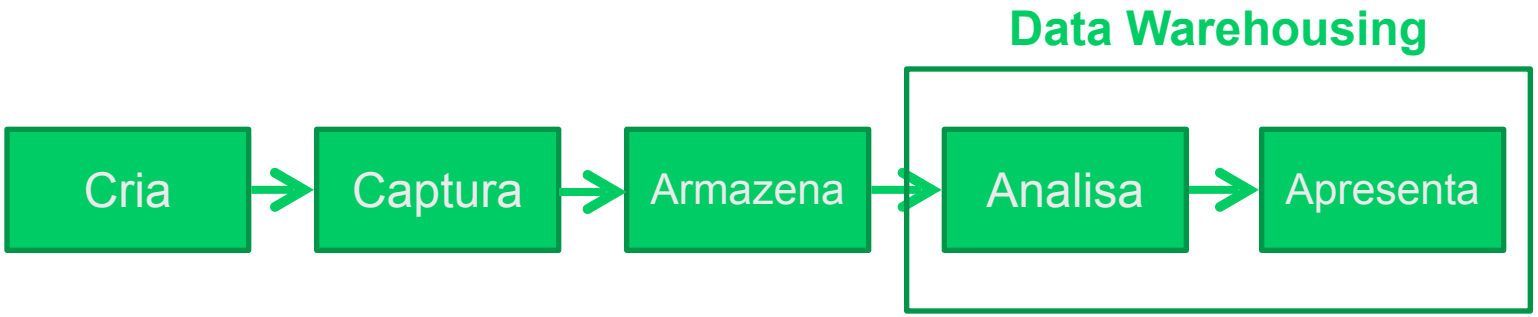
# Introdução

Dados: ciclo de vida


Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# Introdução

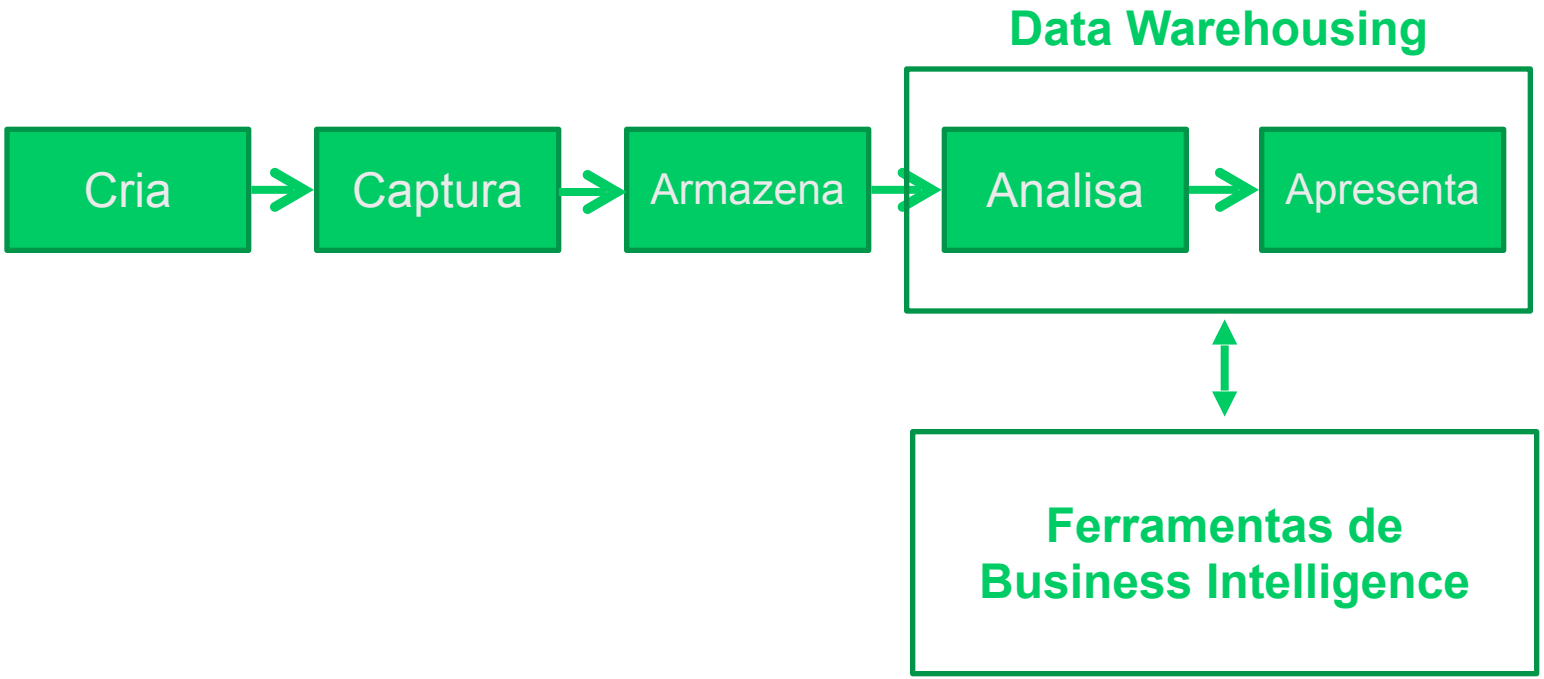
Dados: ciclo de vida




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

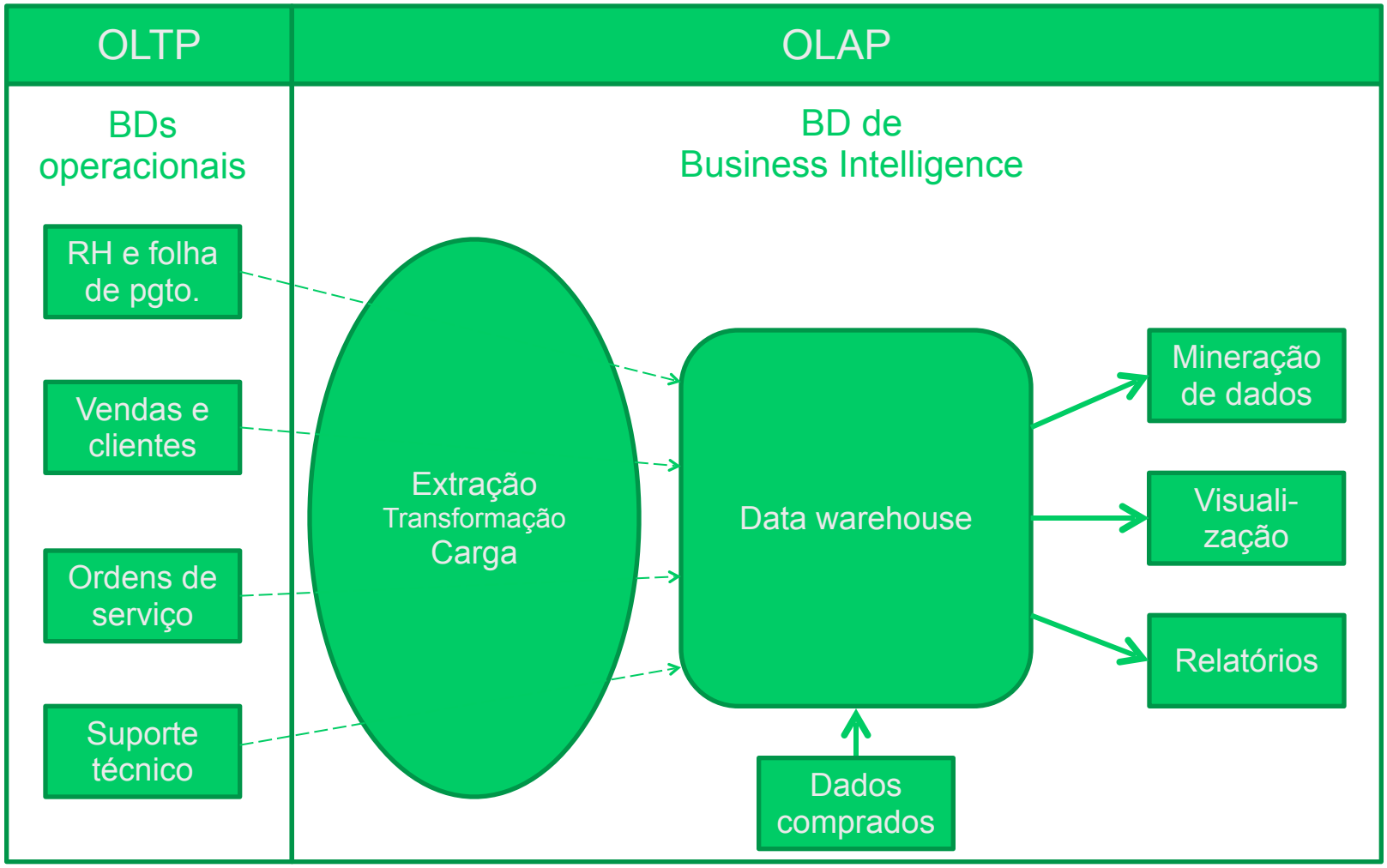
# Introdução

Dados: ciclo de vida




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# Introdução




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# O que é NoSQL?

- Qualquer sistema gerenciador de dados em larga escala, relacional ou não-relacional, é dito um sistema **NoSQL**
- "**N**ot **o**nly **SQL**" → **NoSQL**
- **Não** é anti-SQL ou anti-Relacional


Fonte: www.improgrammer.net

# O que é NoSQL (cont.)?

# O que é NoSQL (cont.)?


- Não são apenas tabelas
  - Sistemas NoSQL armazenam e recuperam dados em vários formatos, e.g., texto, csv, xml, graphml

# O que é NoSQL (cont.)?


- Não são apenas tabelas
  - Sistemas NoSQL armazenam e recuperam dados em vários formatos, e.g., texto, csv, xml, graphml

- Não são apenas junções
  - Sistemas NoSQL permitem que você extraia dados utilizando interfaces simples, ao invés de sempre precisar de junções

# O que é NoSQL (cont.)?


- Não são apenas tabelas
  - Sistemas NoSQL armazenam e recuperam dados em vários formatos, e.g., texto, csv, xml, graphml
- Não são apenas junções
  - Sistemas NoSQL permitem que você extraia dados utilizando interfaces simples, ao invés de sempre precisar de junções
- Não são apenas esquemas
  - Sistemas NoSQL permitem que você copie e cole dados para um diretório, sem ter que organizá-los e consultá-los com base em entidades, atributos, relacionamentos, etc.

# O que é NoSQL (cont.)?

# O que é NoSQL (cont.)?


- Não são apenas executados em um único processador
  - Sistemas NoSQL permitem que você armazene e processe em paralelo dados em clusters de diversas máquinas com alta performance

# O que é NoSQL (cont.)?

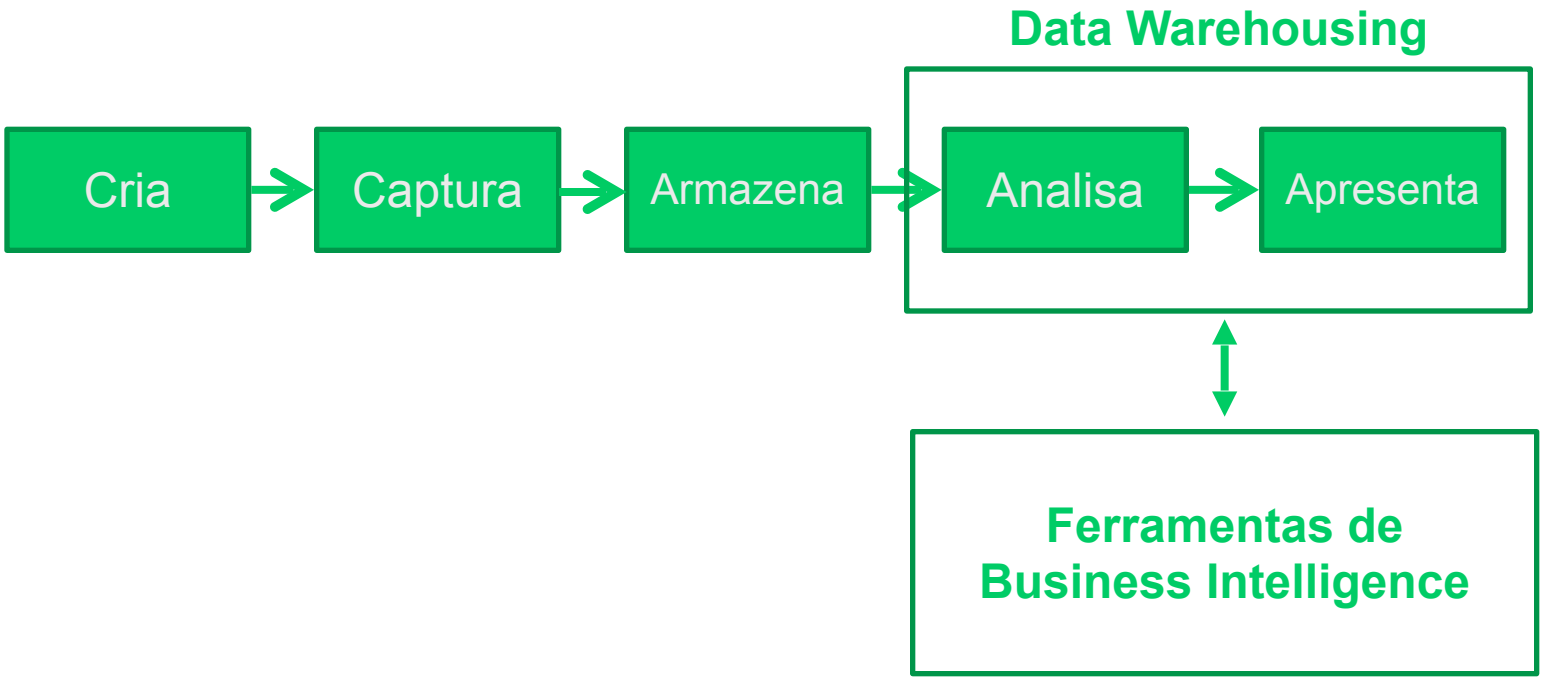

- Não são apenas executados em um único processador
  - Sistemas NoSQL permitem que você armazene e processe em paralelo dados em clusters de diversas máquinas com alta performance
- Não são apenas para super computadores
  - Sistemas NoSQL permitem que você utilize máquinas comuns de baixo custo com processadores, memória RAM e discos independentes

# O que é NoSQL (cont.)?


- Não são apenas executados em um único processador
  - Sistemas NoSQL permitem que você armazene e processe em paralelo dados em clusters de diversas máquinas com alta performance
- Não são apenas para super computadores
  - Sistemas NoSQL permitem que você utilize máquinas comuns de baixo custo com processadores, memória RAM e discos independentes
- Não são apenas **nada**, na realidade…
  - Sistemas NoSQL enfatizam a inovação e inclusão, contemplando estratégias diversas para o armazenamento, a busca e a manipulação de dados em geral, incluindo "soluções padrão" baseadas em SQL

# NoSQL

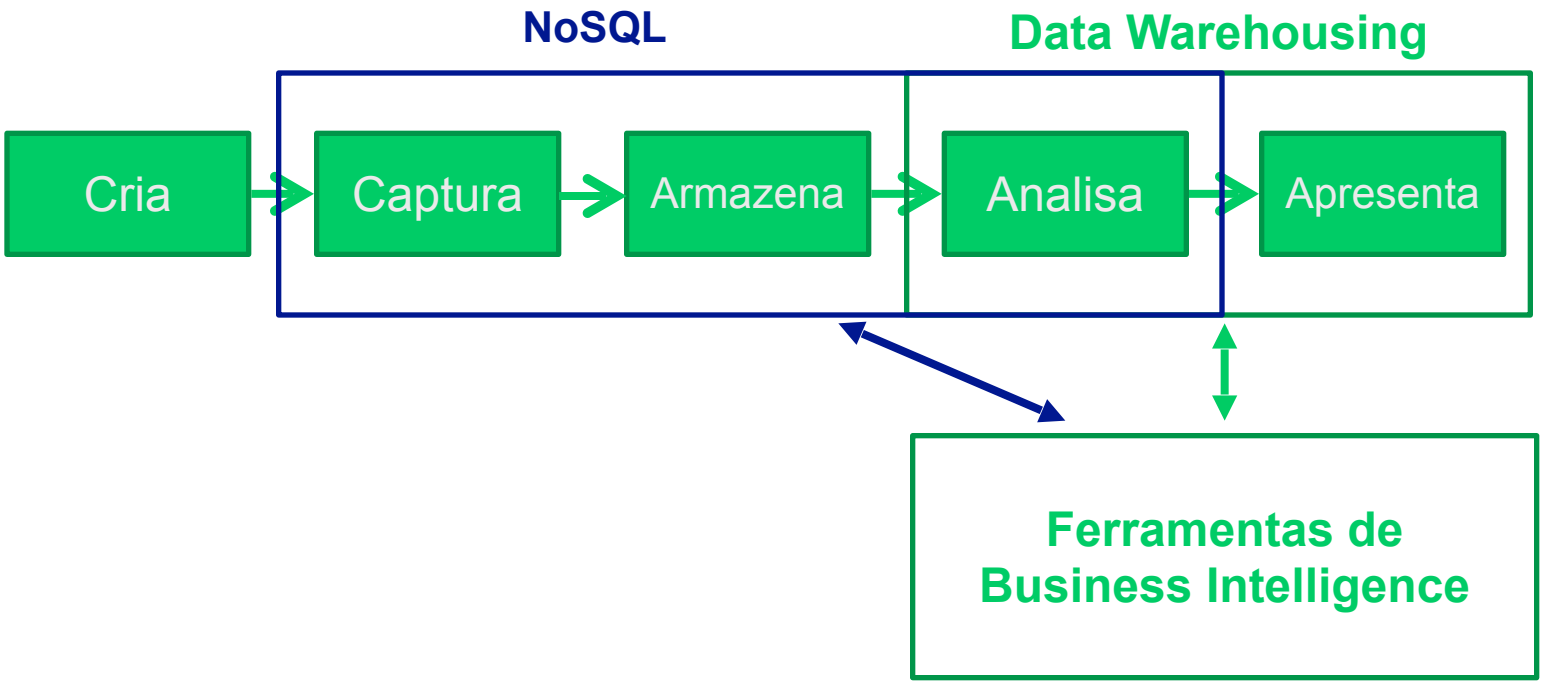
Dados: ciclo de vida




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# NoSQL

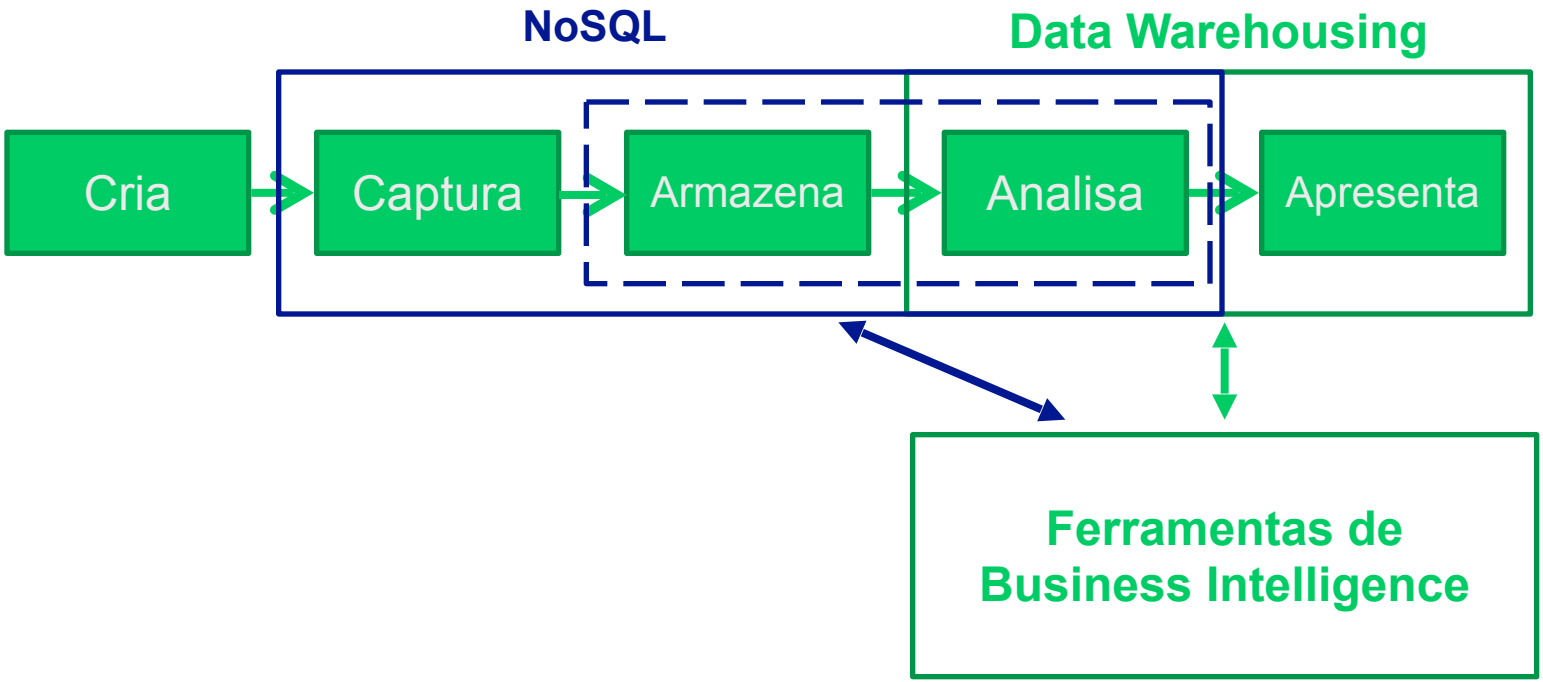
Dados: ciclo de vida




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# NoSQL

Dados: ciclo de vida




Fonte: https://www.youtube.com/watch?v=IjpU0dLIRDI

# Quem usa NoSQL?


Google

facebook

YAHOO!

Linked in

Mc Graw Hill Education

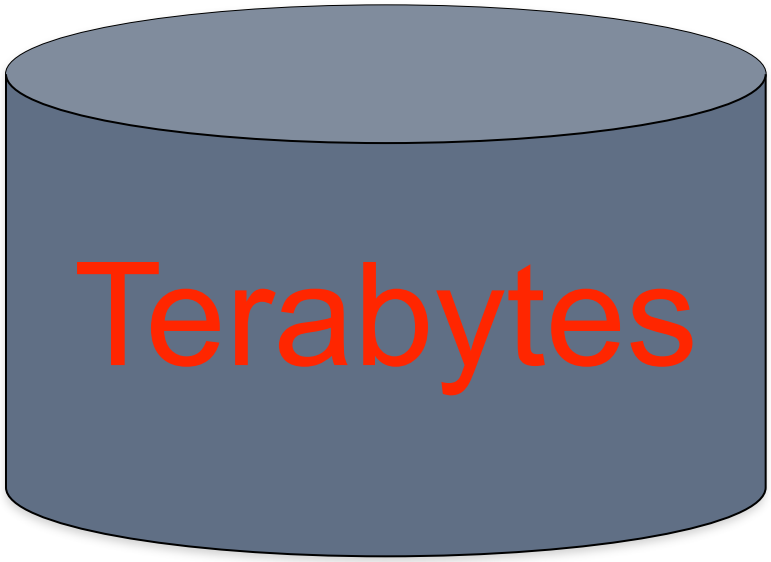
...

# Por que NoSQL?

**Empresa**

(biologia, medicina, astronomia,etc.)

**Acumulando dados**

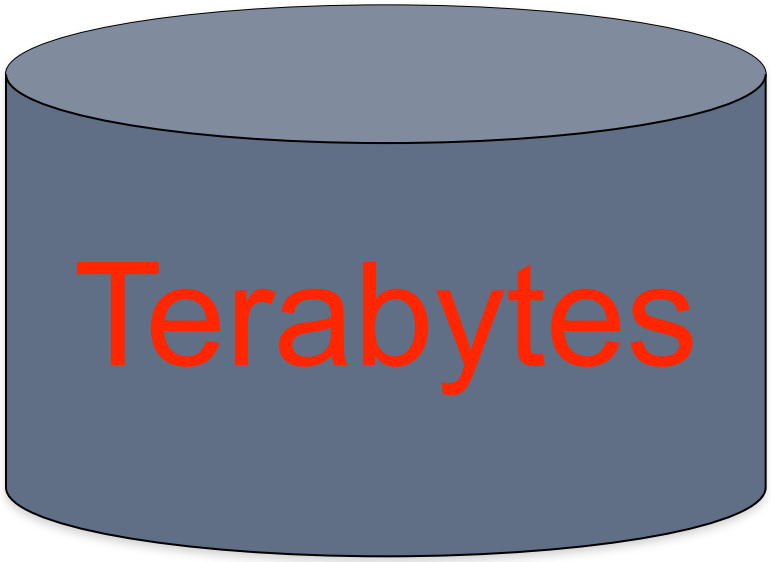


# Por que NoSQL?

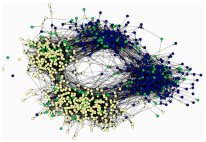
**Empresa**

(biologia, medicina, astronomia,etc.)

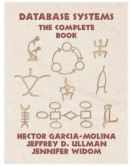
**Acumulando dados**




**Complexidade dos dados**



Redes sociais, etc.




# Por que NoSQL?

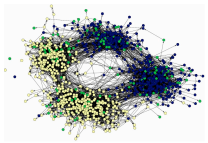
Big Data

bytes

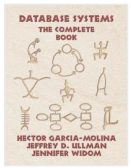
**Empresa**

(biologia, medicina, astronomia,etc.)

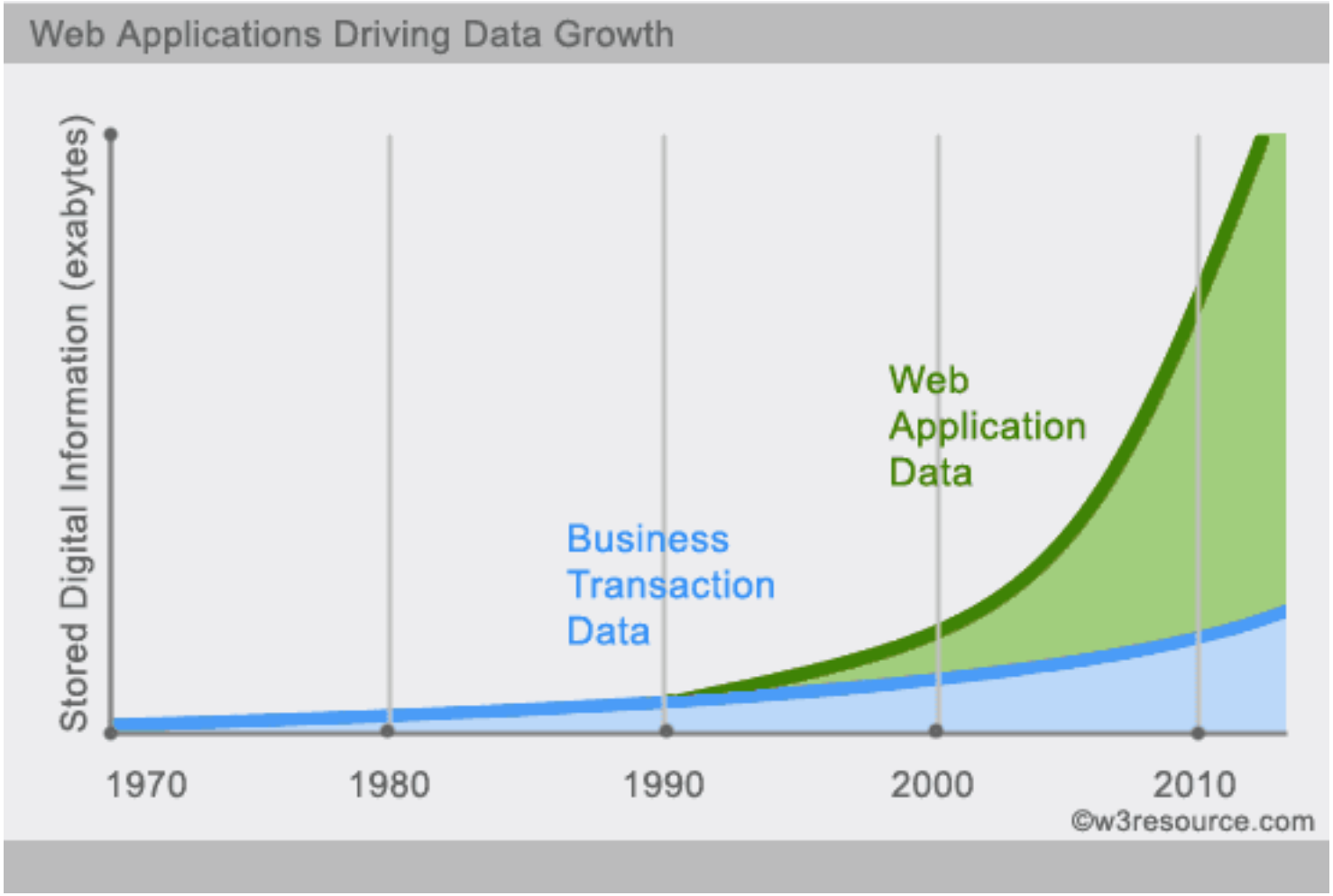
**Complexidade dos dados**



Redes sociais, etc.




# Por que NoSQL (cont.)?



# Por que NoSQL (cont.)?


- Parte relevante desses dados é manipulada por **Sistemas Gerenciadores de Bases de Dados Relacionais – SGBDR**

# Por que NoSQL (cont.)?

- Parte relevante desses dados é manipulada por **Sistemas Gerenciadores de Bases de Dados Relacionais – SGBDR**
  - E.F.Codd. **A relational model of data for large shared data banks**. Communications of the ACM; Volume 13 Issue 6, June 1970; Pages 377-387

# Por que NoSQL (cont.)?


- Parte relevante desses dados é manipulada por **Sistemas Gerenciadores de Bases de Dados Relacionais – SGBDR**
  - E.F.Codd. **A relational model of data for large shared data banks**. Communications of the ACM; Volume 13 Issue 6, June 1970; Pages 377-387
  - Facilita a modelagem e o desenvolvimento de aplicações

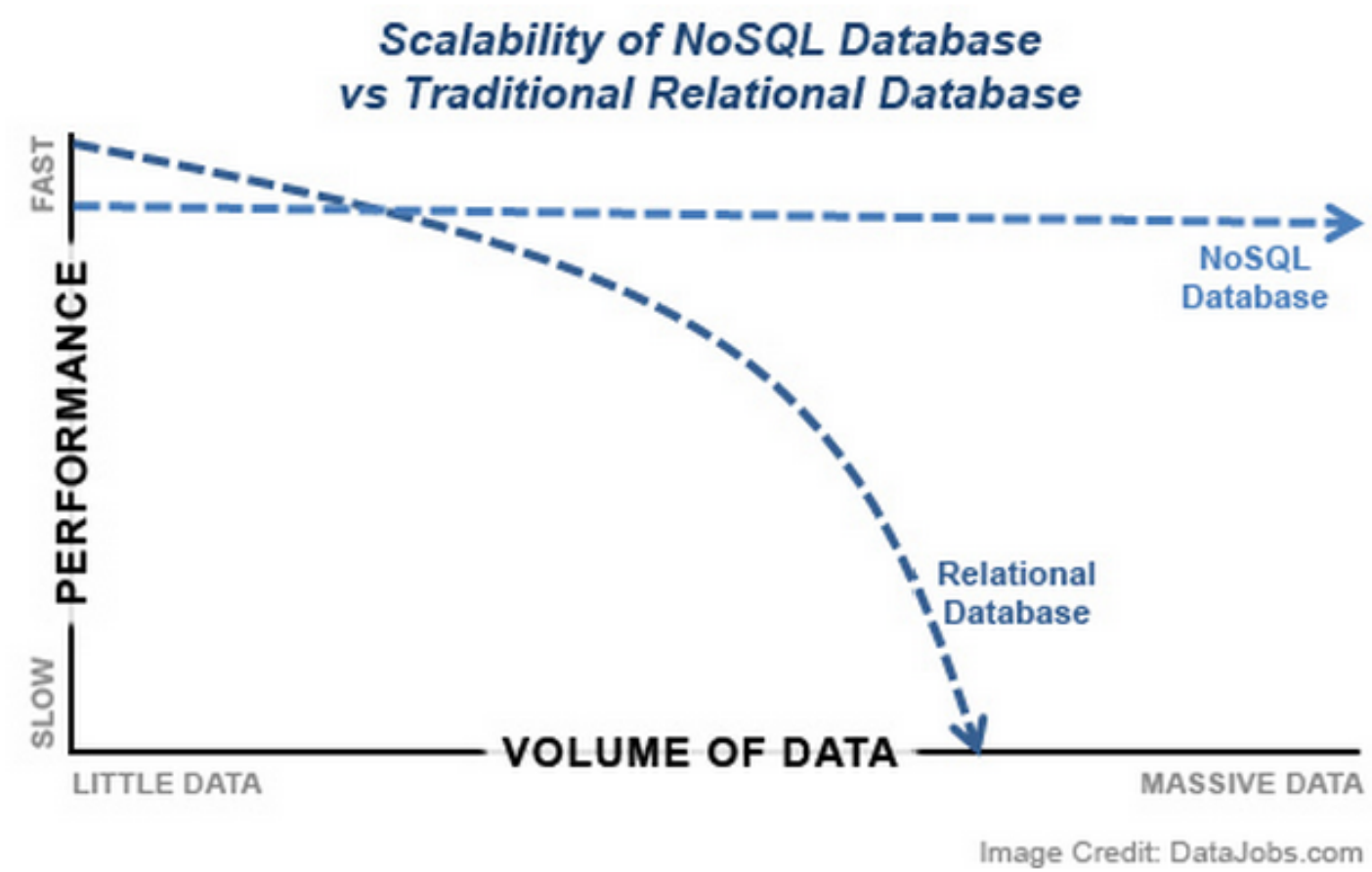
# Por que NoSQL (cont.)?


- Parte relevante desses dados é manipulada por **Sistemas Gerenciadores de Bases de Dados Relacionais – SGBDR**
  - E.F.Codd. **A relational model of data for large shared data banks**. Communications of the ACM; Volume 13 Issue 6, June 1970; Pages 377-387
  - Facilita a modelagem e o desenvolvimento de aplicações
  - Bem adequado à programação cliente/servidor

# Por que NoSQL (cont.)?


- Parte relevante desses dados é manipulada por **Sistemas Gerenciadores de Bases de Dados Relacionais – SGBDR**
  - E.F.Codd. **A relational model of data for large shared data banks**. Communications of the ACM; Volume 13 Issue 6, June 1970; Pages 377-387
  - Facilita a modelagem e o desenvolvimento de aplicações
  - Bem adequado à programação cliente/servidor
  - Tecnologia predominante para o armazenamento de dados estruturados, tanto em aplicações comerciais quanto em aplicações para a Web

# Por que NoSQL (cont.)?



# SGBDR → propriedades ACID

Transações em BDs Relacionais são:

# SGBDR → propriedades ACID

Transações em BDs Relacionais são:

- **Atômicas:** todas as operações de uma transação devem ser efetivadas. Ou, na ocorrência de uma falha, nada deve ser efetivado.
  - "tudo ou nada"

# SGBDR → propriedades ACID

Transações em BDs Relacionais são:

- **Atômicas:** todas as operações de uma transação devem ser efetivadas. Ou, na ocorrência de uma falha, nada deve ser efetivado.
  - "tudo ou nada"
- **Consistentes:** transações preservam a consistência/integridade dos dados
  - Consistência: se há redundância, todas as cópias são iguais
  - Integridade: dados seguem regras do mundo real, e.g., (nota1+nota2)/2 = média final
  - Estado inicial consistente/íntegro → Estado final consistente/íntegro

# SGBDR → propriedades ACID


Transações em BDs Relacionais são:

- **Atômicas:** todas as operações de uma transação devem ser efetivadas. Ou, na ocorrência de uma falha, nada deve ser efetivado.
  - "tudo ou nada"
- **Consistentes:** transações preservam a consistência/integridade dos dados
  - Consistência: se há redundância, todas as cópias são iguais
  - Integridade: dados seguem regras do mundo real, e.g., (nota1+nota2)/2 = média final
  - Estado inicial consistente/íntegro → Estado final consistente/íntegro
- **Isoladas:** uma transação A não vê o efeito de uma transação B até que B termine

# SGBDR → propriedades ACID

Transações em BDs Relacionais são:

- **Atômicas:** todas as operações de uma transação devem ser efetivadas. Ou, na ocorrência de uma falha, nada deve ser efetivado.
  - "tudo ou nada"
- **Consistentes:** transações preservam a consistência/integridade dos dados
  - Consistência: se há redundância, todas as cópias são iguais
  - Integridade: dados seguem regras do mundo real, e.g., (nota1+nota2)/2 = média final
  - Estado inicial consistente/íntegro → Estado final consistente/íntegro
- **Isoladas:** uma transação A não vê o efeito de uma transação B até que B termine
- **Duráveis:** uma vez terminada a transação, as alterações realizadas permanecem no banco até que outras alterações sejam explicitamente realizadas

# Quando preciso de SGBDR / ACID?


- BDs operacionais em geral
  - RH e folha de pagamento
  - Vendas e clientes
  - Suporte técnico
  - Ordens de serviço
  - Transações bancárias
  - ...

# Quando preciso de SGBDR / ACID?


- BDs operacionais em geral
  - RH e folha de pagamento
  - Vendas e clientes
  - Suporte técnico
  - Ordens de serviço
  - Transações bancárias
  - ...

**Consistência e integridade são obrigatórias**

# Quando preciso de **NoSQL** (exceto SGBDR)?


- Redes sociais:

**Usuários**: ID, nome, sobrenome, idade, sexo,...

**Amizades**: UID1, UID2

**Tarefa**: Encontre todos os amigos de amigos de amigos de ... amigos de um dado usuário.

# Quando preciso de **NoSQL** (exceto SGBDR)?


- Redes sociais:

**Usuários**: ID, nome, sobrenome, idade, sexo,...

**Amizades**: UID1, UID2

**Tarefa**: Encontre todos os amigos de amigos de amigos de ... amigos de um dado usuário.

**Recursividade**

# Quando preciso de **NoSQL** (exceto SGBDR)?


- Redes sociais:

**Usuários**: ID, nome, sobrenome, idade, sexo,...

**Amizades**: UID1, UID2

**Tarefa**: Encontre todos os amigos de amigos de amigos de ... amigos de um dado usuário.

**Recursividade**

- Páginas da Wikipédia:

Grande coleção de documentos

**Tarefa**: Encontre todas as páginas referentes a atletas participantes das Olimpíadas até 1950.

# Quando preciso de **NoSQL** (exceto SGBDR)?


- Redes sociais:

**Usuários**: ID, nome, sobrenome, idade, sexo,...
**Amizades**: UID1, UID2
**Tarefa**: Encontre todos os amigos de amigos de amigos de ... amigos de um dado usuário.

**Recursividade**

- Páginas da Wikipédia:

Grande coleção de documentos

**Combinação de dados estruturados e não estruturados**

**Tarefa**: Encontre todas as páginas referentes a atletas participantes das Olimpíadas até 1950.

# Quando preciso de **NoSQL** (exceto SGBDR)?


- Redes sociais:

**Usuários**: ID, nome, sobrenome, idade, sexo,...

**Amizades**: UID1, UID2

**Tarefa**: Encontre todos os amigos de amigos de amigos de ... amigos de um dado usuário.

**Recursividade**

- Páginas da Wikipédia:

Grande coleção de documentos

**Combinação de dados estruturados e não estruturados**

**Tarefa**: Encontre todas as páginas referentes a atletas participantes das Olimpíadas até 1950.

**Consistência e integridade são desejáveis, mas não obrigatórias**

# SGBDR versus NoSQL

- SGBDR
  - Dados estruturados e organizados

# SGBDR versus NoSQL

- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

- NoSQL (exceto SGBDR)
  - Dados semi-/não estruturados e imprevisíveis

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

- NoSQL (exceto SGBDR)
  - Dados semi-/não estruturados e imprevisíveis
  - **Consistência eventual:** transações BASE, **não** ACID

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

- NoSQL (exceto SGBDR)
  - Dados semi-/não estruturados e imprevisíveis
  - **Consistência eventual:** transações BASE, **não** ACID
  - Não possui linguagem de consulta declarativa, e nem esquema de dados pré-definido

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

- NoSQL (exceto SGBDR)
  - Dados semi-/não estruturados e imprevisíveis
  - **Consistência eventual:** transações BASE, **não** ACID
  - Não possui linguagem de consulta declarativa, e nem esquema de dados pré-definido
  - Quatro categorias principais, com propósitos distintos

# SGBDR versus NoSQL


- SGBDR
  - Dados estruturados e organizados
  - Linguagem estruturada de consulta SQL, com DDL e DML
  - Dados e relacionamentos comumente armazenados em arquivos distintos
  - Controle rígido de **consistência e integridade de dados**

- NoSQL (exceto SGBDR)
  - Dados semi-/não estruturados e imprevisíveis
  - **Consistência eventual:** transações BASE, **não** ACID
  - Não possui linguagem de consulta declarativa, e nem esquema de dados pré-definido
  - Quatro categorias principais, com propósitos distintos
  - Prioriza alta performance, escalabilidade e disponibilidade

# Teorema CAP

## (Teorema de Brewer)

- Três requisitos básicos:
  - **Consistency** – dados consistentes/íntegros após a execução de cada operação. Por exemplo, após uma atualização de dados, todos os usuários “verão” os mesmos dados

# Teorema CAP

## (Teorema de Brewer)

- Três requisitos básicos:
  - **Consistency** – dados consistentes/íntegros após a execução de cada operação. Por exemplo, após uma atualização de dados, todos os usuários “verão” os mesmos dados
  - **Availability** – todos os dados estão sempre disponíveis, 24x7

# Teorema CAP

## (Teorema de Brewer)

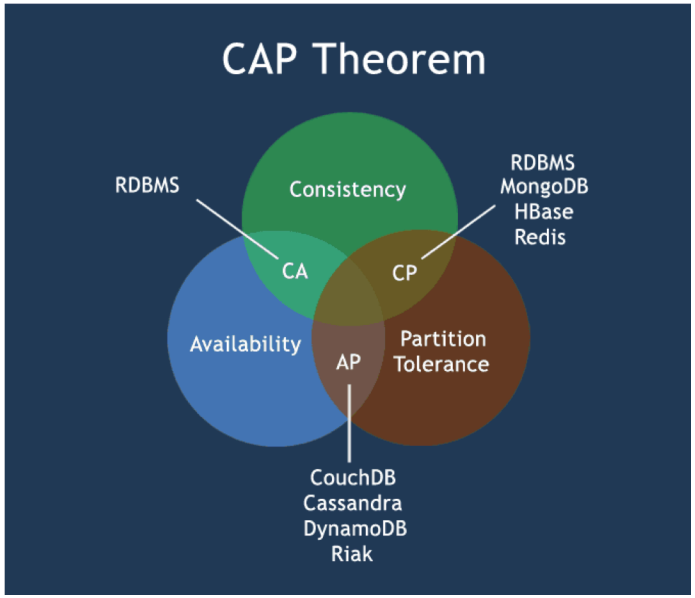
- Três requisitos básicos:
  - **Consistency** – dados consistentes/íntegros após a execução de cada operação. Por exemplo, após uma atualização de dados, todos os usuários “verão” os mesmos dados
  - **Availability** – todos os dados estão sempre disponíveis, 24x7
  - **Partition tolerance** – sistema funcional mesmo quando a comunicação entre os servidores é deficiente, i.e., mesmo com servidores particionados em múltiplos grupos isolados entre si

# Teorema CAP

## (Teorema de Brewer)

- Três requisitos básicos:
  - **Consistency** – dados consistentes/íntegros após a execução de cada operação. Por exemplo, após uma atualização de dados, todos os usuários "verão" os mesmos dados
  - **Availability** – todos os dados estão sempre disponíveis, 24x7
  - **Partition tolerance** – sistema funcional mesmo quando a comunicação entre os servidores é deficiente, i.e., mesmo com servidores particionados em múltiplos grupos isolados entre si

- É **impossível** obter os três requisitos ao mesmo tempo. Têm-se então:
  - **CA** – servidor único
  - **CP** – parte dos dados poder estar inacessível temporariamente, porém os demais dados são consistentes/ntegros
  - **AP** – todos os dados são acessíveis, mesmo com particionamento, porém podem existir dados inconsistentes/não íntegros (**consistência eventual**)



# NoSQL: vantagens e desvantagens

# NoSQL: vantagens e desvantagens


- Vantagens
  - Alta escalabilidade
  - Processamento distribuído
  - Baixo custo
  - Flexibilidade de esquema; dados semi-estruturados ou não estruturados

# NoSQL: vantagens e desvantagens


- Vantagens
  - Alta escalabilidade
  - Processamento distribuído
  - Baixo custo
  - Flexibilidade de esquema; dados semi-estruturados ou não estruturados

- Desvantagens
  - Consistência eventual
  - Falta de padronização
  - Capacidade limitada de consulta
  - É pouco intuitivo programar quando se tem consistência eventual

# NoSQL AP → Propriedades BASE

# NoSQL AP → Propriedades BASE


- Transações em sistemas NoSQL que adotam consistência eventual seguem propriedades BASE, não ACID.
  - **Basically Available:** todos os dados acessíveis a qualquer momento
  - **Soft state:** o sistema/dados podem mudar ao longo do tempo, mesmo sem nenhuma requisição de usuário
  - **Eventual consistency:** o sistema se tornará consistente em algum momento, desde que não ocorram novas requisições de usuários

# NoSQL AP → Propriedades BASE


- Transações em sistemas NoSQL que adotam consistência eventual seguem propriedades BASE, não ACID.
  - **Basically Available:** todos os dados acessíveis a qualquer momento
  - **Soft state:** o sistema/dados podem mudar ao longo do tempo, mesmo sem nenhuma requisição de usuário
  - **Eventual consistency:** o sistema se tornará consistente em algum momento, desde que não ocorram novas requisições de usuários

| ACID | BASE |
|---|---|
| Atomicidade | Basically Available |
| Consistência | Soft state |
| Isolamento | Eventual consistency |
| Durabilidade | |

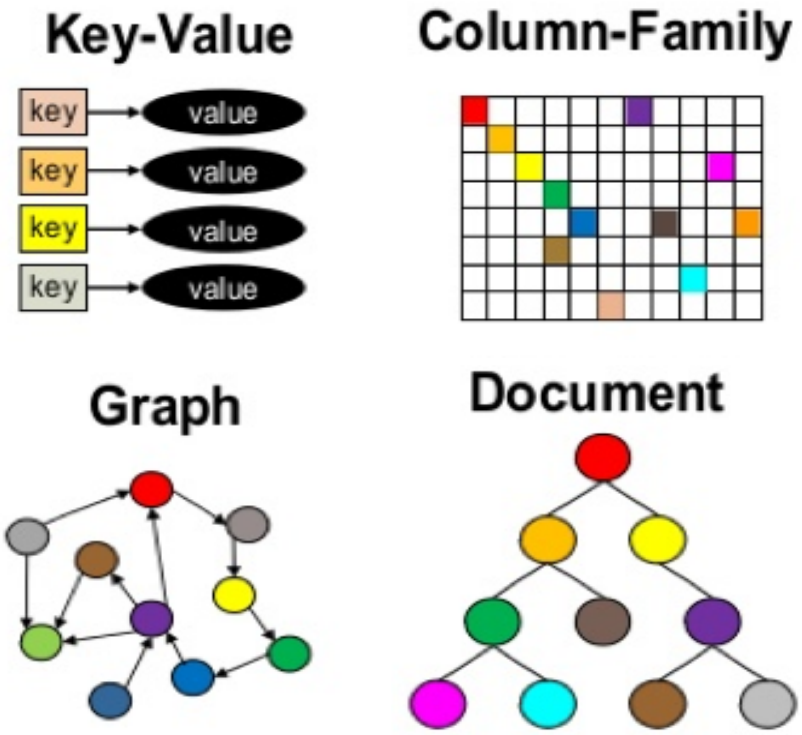
# NoSQL AP → Propriedades BASE


- Transações em sistemas NoSQL que adotam consistência eventual seguem propriedades BASE, não ACID.
  - **Basically Available:** todos os dados acessíveis a qualquer momento
  - **Soft state:** o sistema/dados podem mudar ao longo do tempo, mesmo sem nenhuma requisição de usuário
  - **Eventual consistency:** o sistema se tornará consistente em algum momento, desde que não ocorram novas requisições de usuários

| ACID | BASE |
|---|---|
| Atomicidade | Basically Available |
| Consistência | Soft state |
| Isolamento | Eventual consistency |
| Durabilidade | |

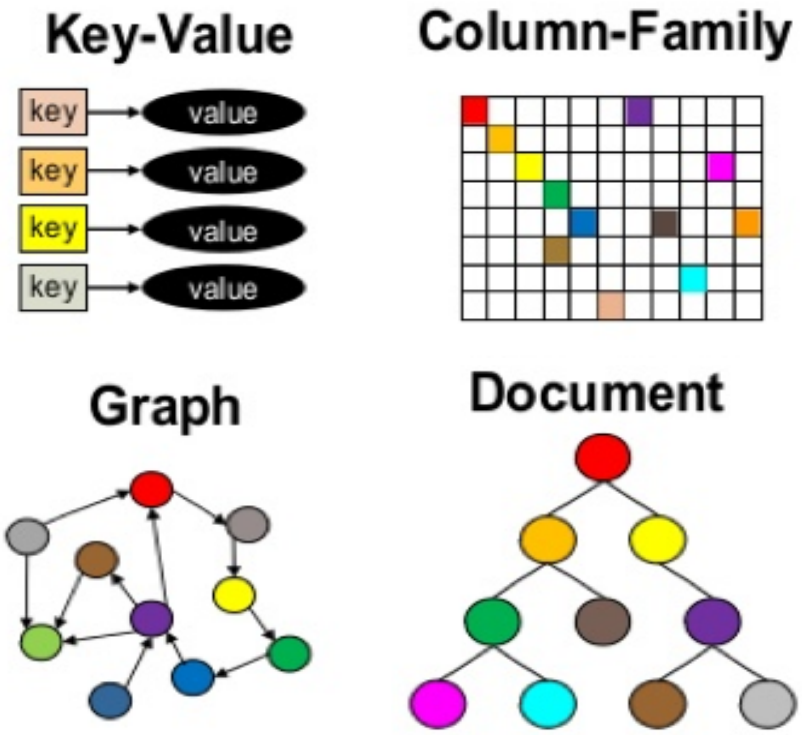
- Exemplos: BigTable, Cassandra e SimpleDB

# Categorias NoSQL



# Categorias NoSQL

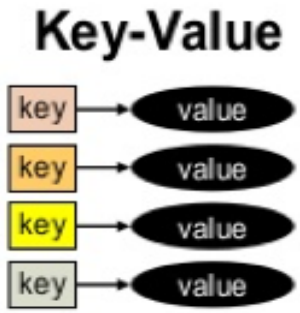
- Existem quatro categorias principais de SGBDs NoSQL:
  - Pares de chave-valor
  - Orientado a colunas
  - Grafos
  - Orientado a documentos

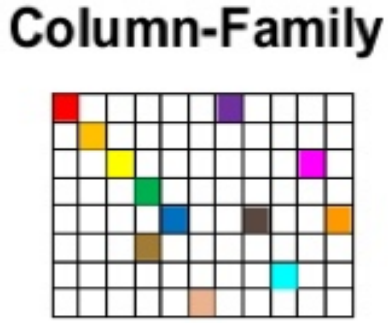


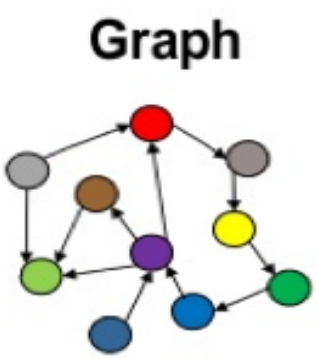
# Categorias NoSQL

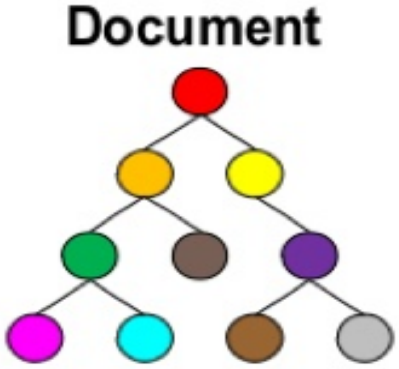
- Existem quatro categorias principais de SGBDs NoSQL:
  - Pares de chave-valor
  - Orientado a colunas
  - Grafos
  - Orientado a documentos

- Características e limitações específicas



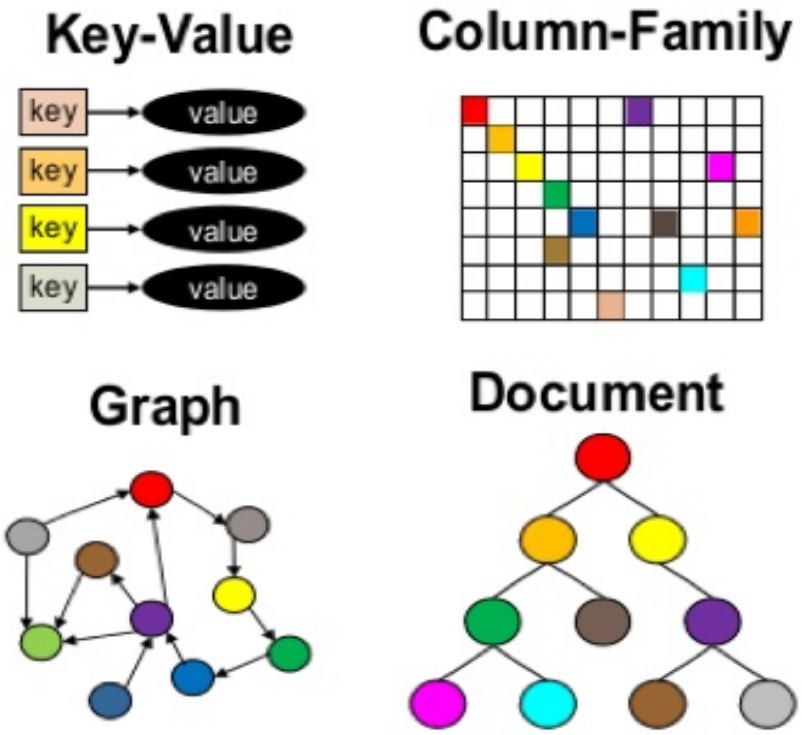



# Categorias NoSQL

- Existem quatro categorias principais de SGBDs NoSQL:
  - Pares de chave-valor
  - Orientado a colunas
  - Grafos
  - Orientado a documentos

- Características e limitações específicas
- Melhor opção? Depende do problema em mãos




Fonte: http://www.slideshare.net/KrishnakumarSukumaran/

# Conclusões

# Conclusões

- **BDs Relacionais**
  - Esquema pré-definido e fixo
  - Interface padrão aplicação/usuário ↔ BD
    - SQL
  - Rígida **consistência e integridade**
  - Semântica de dados bem definida

# Conclusões

- **BDs Relacionais**
  - Esquema pré-definido e fixo
  - Interface padrão aplicação/usuário ↔ BD
    - SQL
  - Rígida **consistência e integridade**
  - Semântica de dados bem definida
- **BDs NoSQL**
  - Esquema parcialmente definido ou até inexistente
  - Definições e interfaces aplicação/usuário ↔ BD distintas por produto
  - **Obter respostas rápidas é mais importante do que obter a resposta correta**

# Conclusões

- **BDs Relacionais (NewSQL???)**
  - Esquema pré-definido e fixo
  - Interface padrão aplicação/usuário ↔ BD
    - SQL
  - Rígida **consistência e integridade**
  - Semântica de dados bem definida
- **BDs NoSQL**
  - Esquema parcialmente definido ou até inexistente
  - Definições e interfaces aplicação/usuário ↔ BD distintas por produto
  - **Obter respostas rápidas é mais importante do que obter a resposta correta**

# Conclusões (cont.)

# Conclusões (cont.)

- **BDs NoSQL** evitam
  - Overhead de transações ACID
  - Limitações de consultas SQL
    - Expressões declarativas de consulta
  - Trabalho com modelagem e normalização
  - Uso de tecnologias “antigas”???

# Conclusões (cont.)

- **BDs NoSQL** evitam
  - Overhead de transações ACID
  - Limitações de consultas SQL
    - Expressões declarativas de consulta
  - Trabalho com modelagem e normalização
  - Uso de tecnologias "antigas"???
- Responsabilidade do **programador**
  - Escrever códigos procedurais (passo-a-passo)
  - Navegar por caminhos/endereços

# Referências


- Material de aulas do Prof. Dr. Rob Gleasure, University College Cork, Ireland. http://corvus2vm.ucc.ie/phd/rgleasure/rgleasure/index.html

- NoSQL, w3resource. http://www.w3resource.com/mongodb/nosql.php

- Material de aulas do Prof. Dr. Xuanhua Shi, Huazhong University of Science and Technology, China. grid.hust.edu.cn/xhshi/

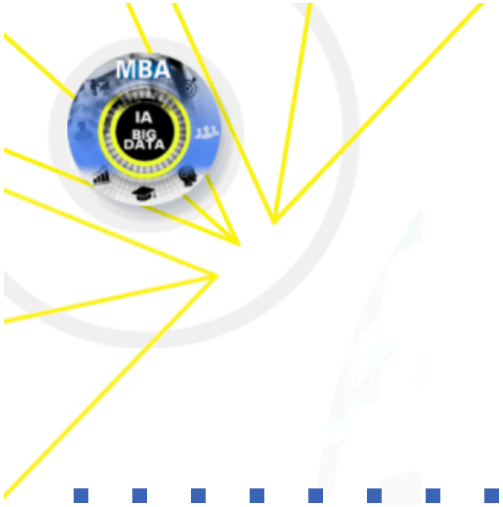
# Referências (cont.)


- Material de aulas do Prof. Dr. Ray R. Larson, UC Berkeley School of Information, USA. http://courses.ischool.berkeley.edu/i257/f15/

- Jeffrey Dean and Sanjay Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”, OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, 2004.

- Apache Hadoop. http://lucene.apache.org/hadoop/

- http://code.google.com/edu/parallel/mapreduce-tutorial.html




# NoSQL e o processamento de dados em larga escala (Parte 1)

Prof. Dr. Robson L. F. Cordeiro
robson@icmc.usp.br

USP-ICMC
GBDI - Grupo de Bases de Dados e de Imagens