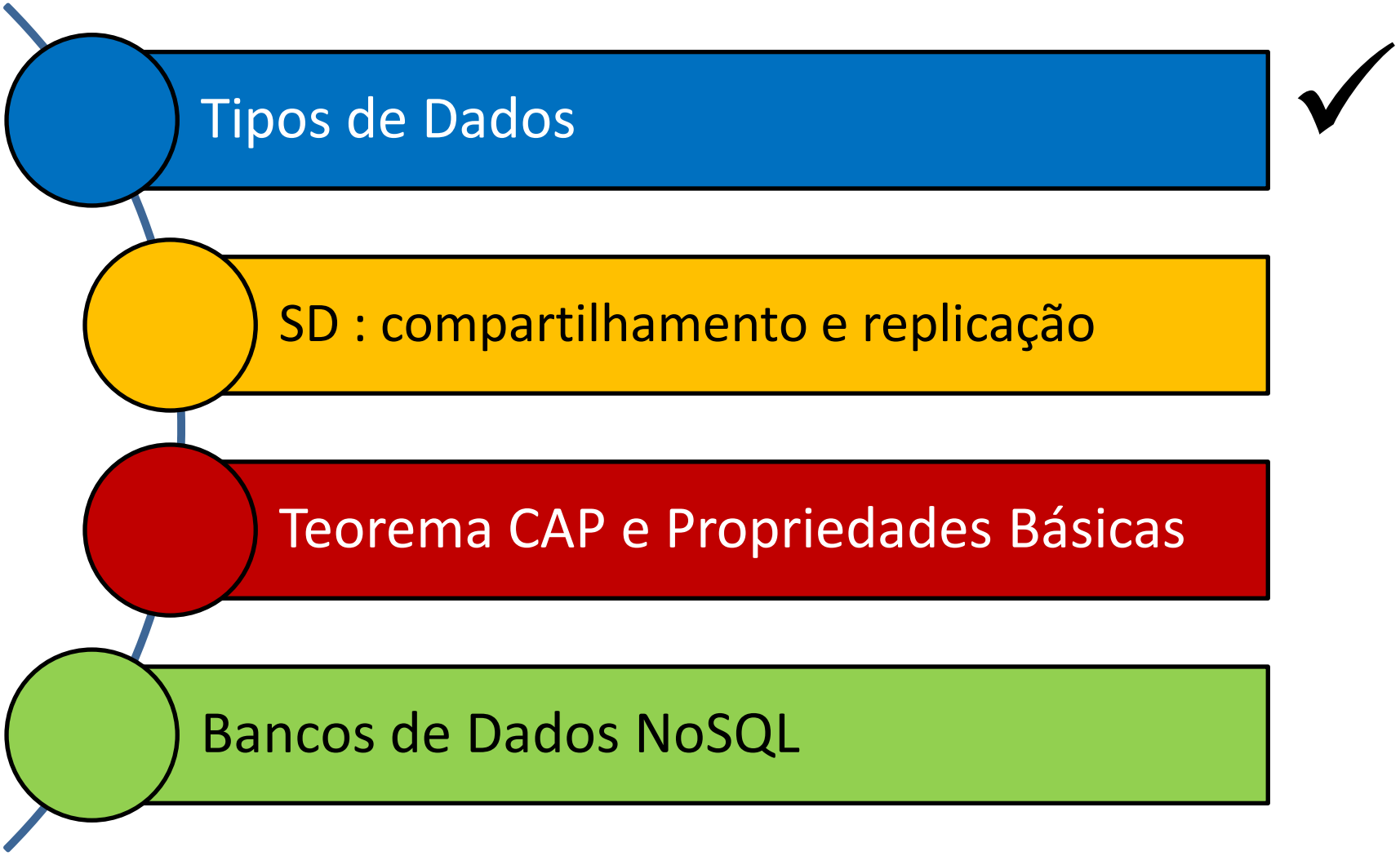
Bancos de Dados NoSQL

Renata Galante

Sumário



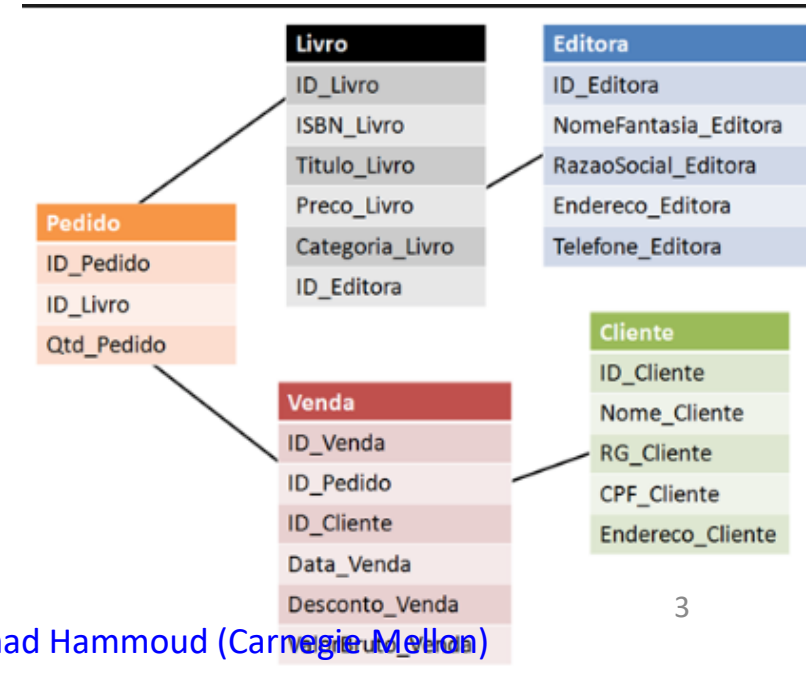
Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Tipos de Dados

1. Dados Estruturados

- Modelo pré-definido
- Dados organizados em formato que facilita
  - Armazenamento
  - Gerencia
  - Recuperação
  - Processamento

BD Relacionais



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Tipos de Dados

2. Dados Não Estruturados

- Não possuem formato padrão
- Difícil gerenciar, recuperar e processar

TXT, áudio, vídeo, imagem

- Atenção: podem possuir metadados associados (não totalmente sem estrutura)

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Tipos de Dados

3. Dados Dinâmicos

- Dados mudam com frequência

Documentos, dados financeiros, dados transacionais

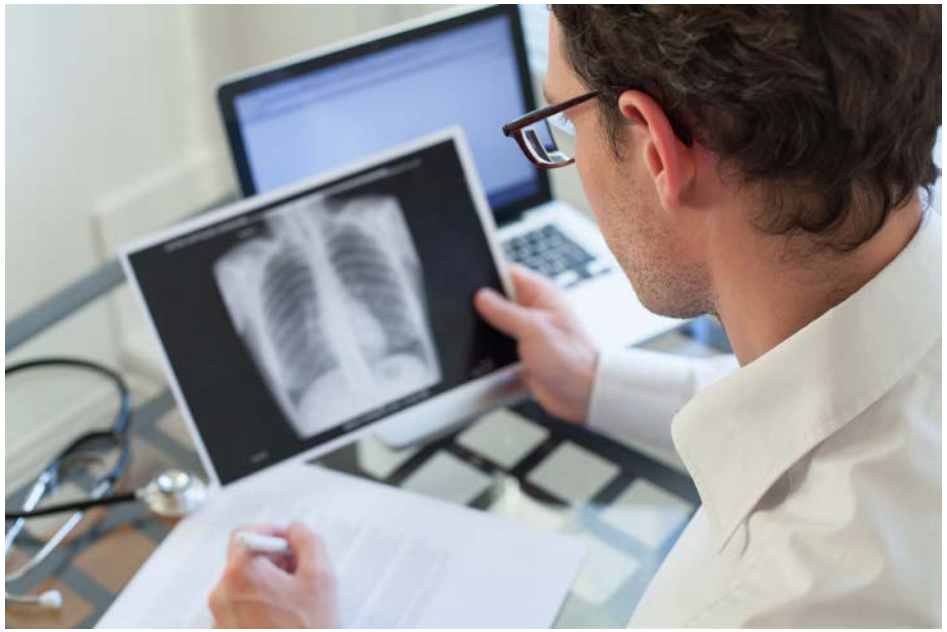
Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Tipos de Dados

4. Dados Estáticos

- Dados não mudam

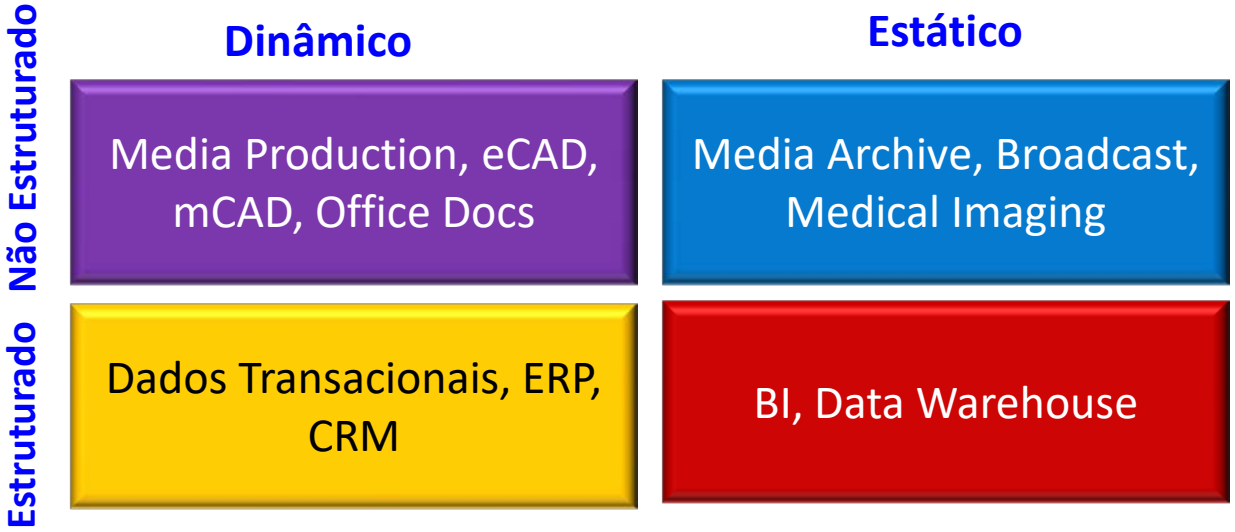
Imagens médicas



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Classificação dos Dados

- Segmentação em quadrantes pode nos ajudar a projetar e desenvolver soluções de armazenamento

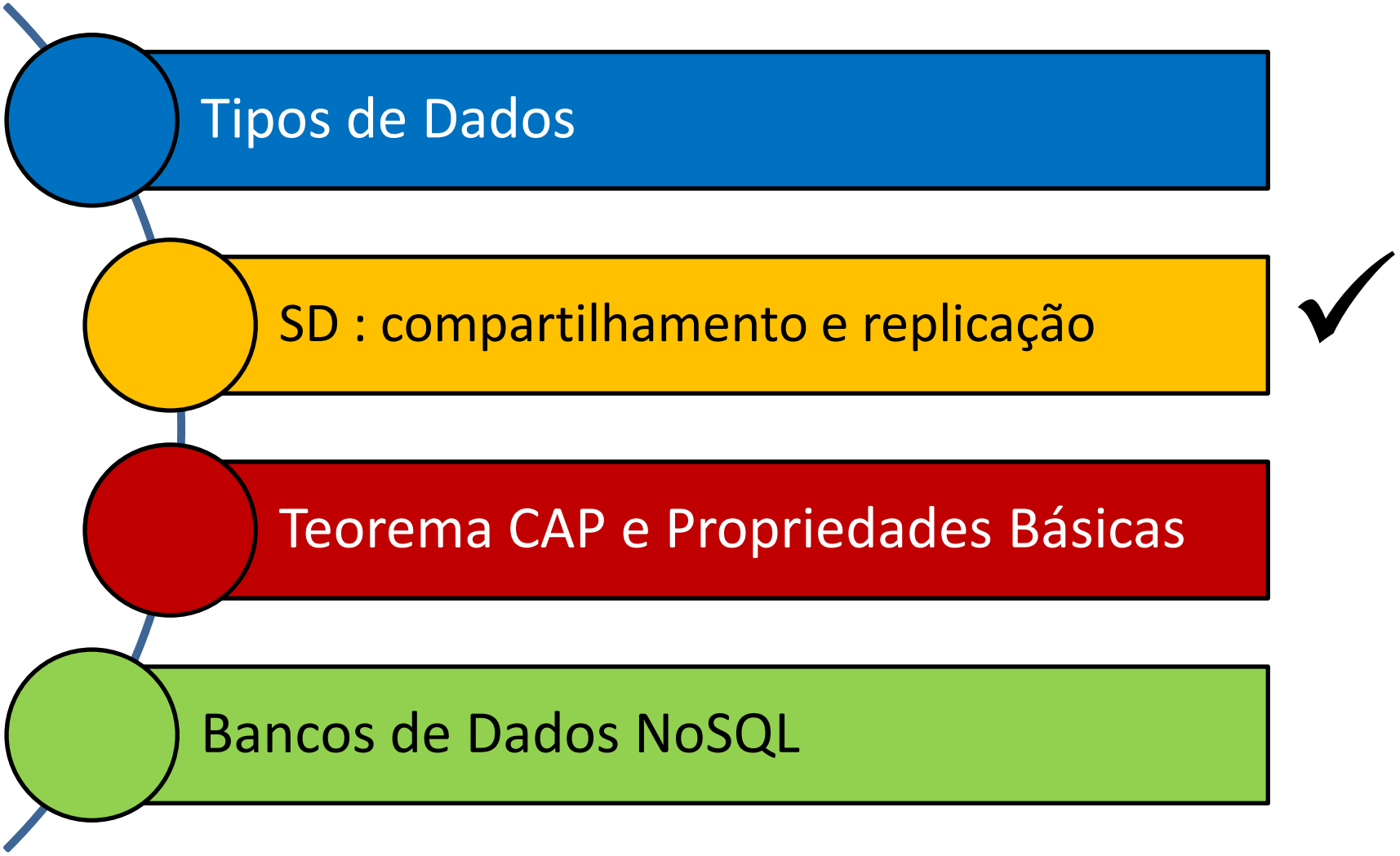


- BD Relacionais são usados para dados estruturados

- Sistemas de Arquivo ou *NoSQL* podem ser usados para (estático), não estruturados (veremos mais detalhes depois)

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Sumário



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

BD Tradicionais Distribuídos

BD relacionais podem ser distribuídos:

- Verticalmente
  - Atualizações de hardward (e.g., CPU, mais rápida, mais memória, mais disco)
  - Limitado pela quantidade de CPU, RAM e disco que podem ser configurados em uma única máquina

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

BD Tradicionais Distribuídos

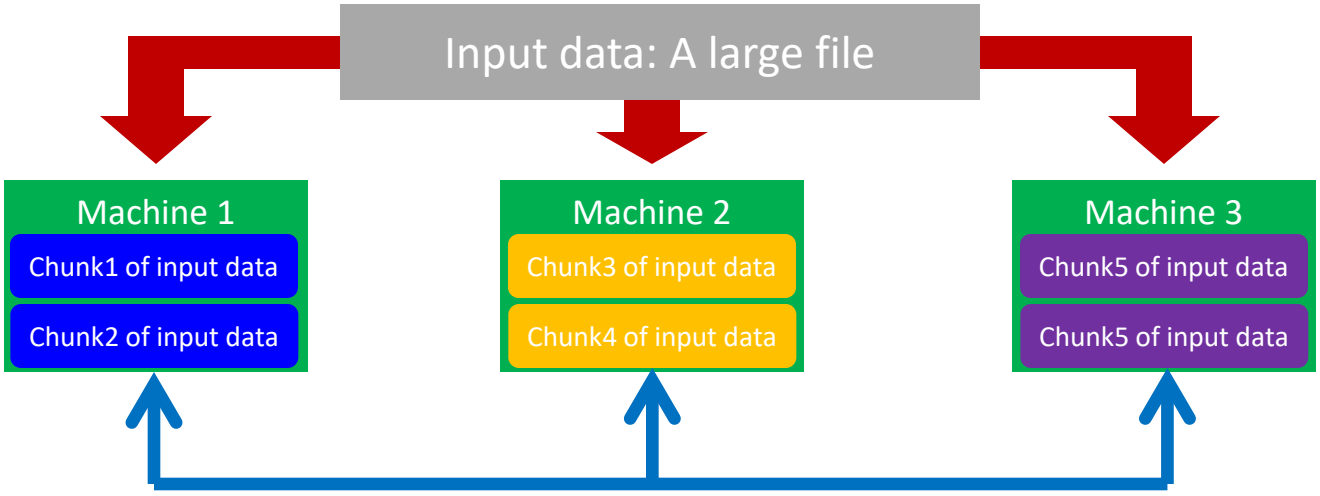
BD relacionais podem ser distribuídos:

- Verticalmente
  - Atualizações de hardward (e.g., CPU, mais rápida, mais memória, mais disco)
  - Limitado pela quantidade de CPU, RAM e disco que podem ser configurados em uma única máquina
- Horizontalmente
  - Adição de mais máquinas
  - Requer compartilhamento e *replicação*
  - Limitado pela taxa de leitura e gravação e pela sobrecarga de comunicação

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Por que compartilhar dados?

Permitem concorrência e paralelismo



Exemplo: 1, 3 e 5 podem ser acessados em paralelo

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Benefícios do Paralelismo

1. Aumentar a fração de programas que podem ser paralelizaados

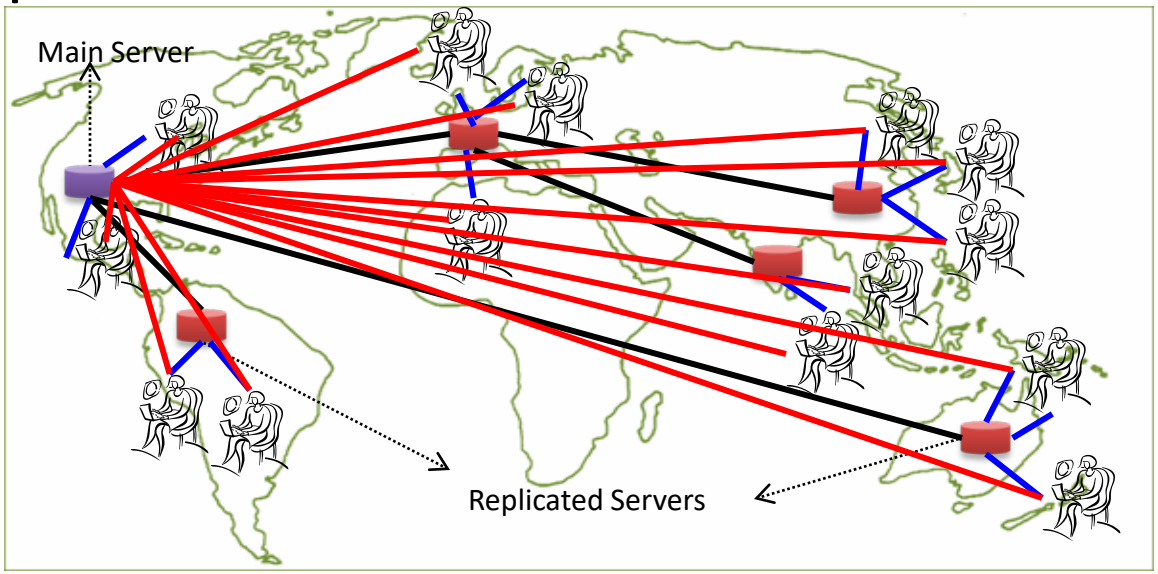
2. Equilibrar a carga de trabalho na execução de processos paralelos

3. Diminuir o tempo gasto na comunicação

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Por que replicar dados?

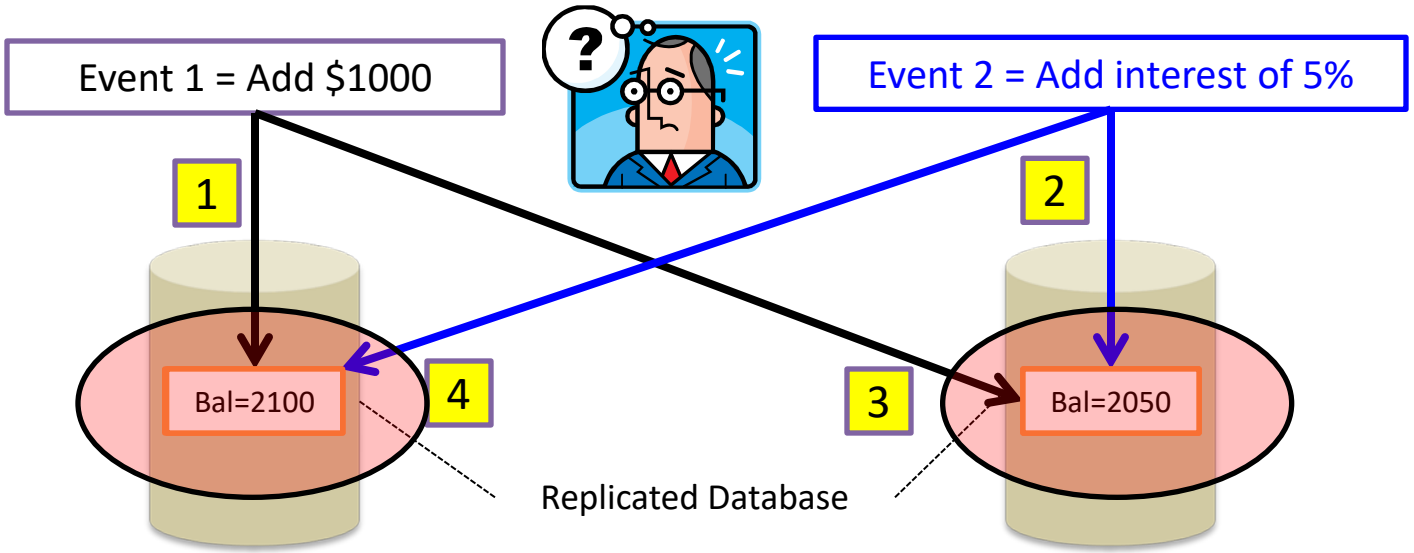
- Evitar gargalos de desempenho
- Evitar o ponto único de falhas
- E, portanto, aprimorando a escalabilidade e a disponibilidade



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

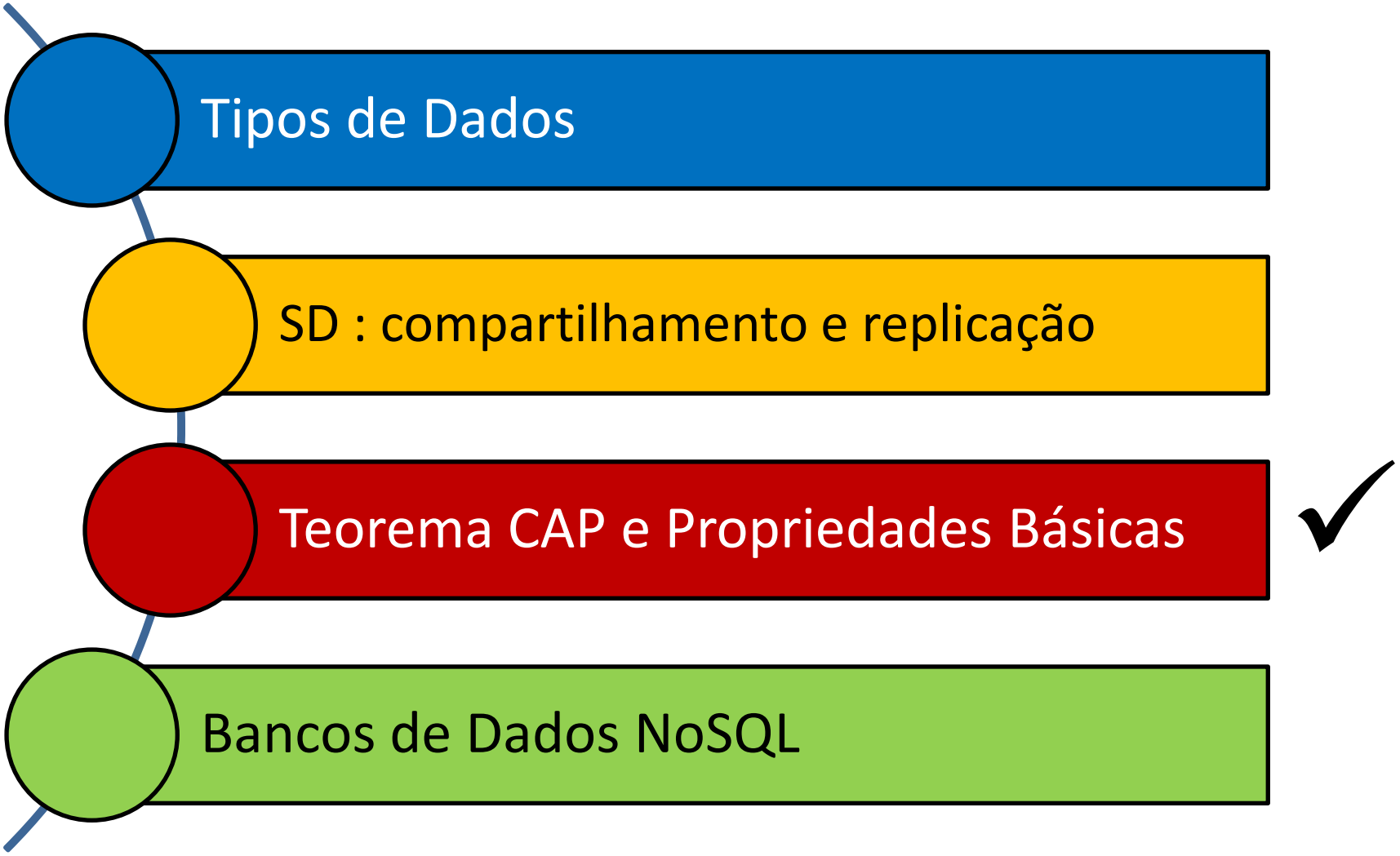
Consistência torna-se um desafio

- Banco replica os dados em dois servidores
- Desafio: manter a consitência das replicas de dados



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Sumário



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Teorema CAP

Limitação dos Sistemas Distribuídos descrita pelo Teorema CAP

- ***C****onsistency*: cada nó sempre vê os mesmos dados em qualquer instância

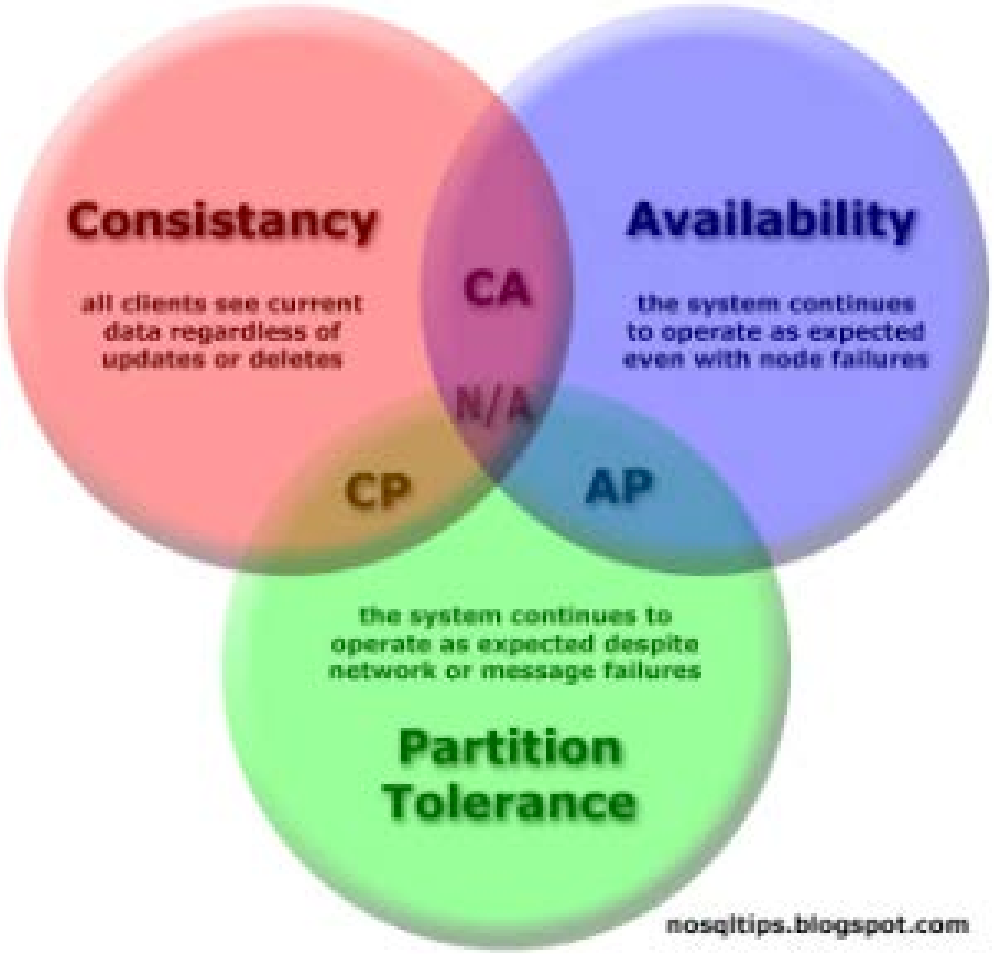
- ***A****vailability*: o sistema permanece operando mesmo na ocorrência de falha de algum nó

- ***P****artition Tolerance*: o sistema continua operando na presença de falhas na rede

Teorema CAP: qualquer sistema distribuído com dados compartilhados, pode ter, no máximo, ***duas*** das três ***propriedades*** C, A or P

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

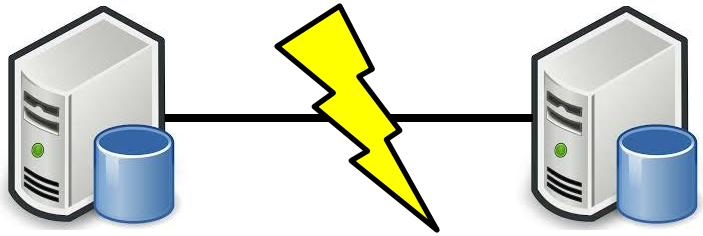
Teorema CAP



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Teorema CAP

Considere dois nodos em lados opostos da rede



- *Availability* + *Partition Tolerance* compromete a consistência

- *Consistency* + *Partition Tolerance* um lado da partição deve agir como se não estivesse disponível, perdendo assim a disponibilidade

- *Consistency* + *Availability* só é possível se não houver particionamento da rede (caso contrário, compromete a *Partition Tolerance*)

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

BD Laga Escala

- Quando empresas como Google e Amazon projetavam BD em grande escala, a disponibilidade 24 horas por dia era um fator essencial

  minutos de inatividade significam perda de receita

- Quando dimensionar horizontalmente BD para milhares de máquinas, a probabilidade de um nó ou uma falha de rede aumenta tremendamente
- Portanto, para ter fortes garantias de Disponibilidade e Tolerância à Partição, eles tiveram que sacrificar a consistência "estrita" (implícita no teorema da PAC)

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Gerenciamento da Consistência

Equilíbrio consistência X disponibilidade/escalabilidade

- Consistência depende da aplicação

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Gerenciamento da Consistência

Equilíbrio consistência X disponibilidade/escalabilidade

- Consistência depende da aplicação

Fácil de implementar e eficiente (tempo de execução)

Geralmente é difícil de implementar e ineficiente

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Propriedades BASE

- O teorema CAP prova que é impossível garantir consistência e disponibilidade rígidas ao mesmo tempo em que se tolera as partições de rede

- **ACID Relaxadas**

  ACID (**A**tomicity, **C**onsistency, **I**solation, **D**urability)

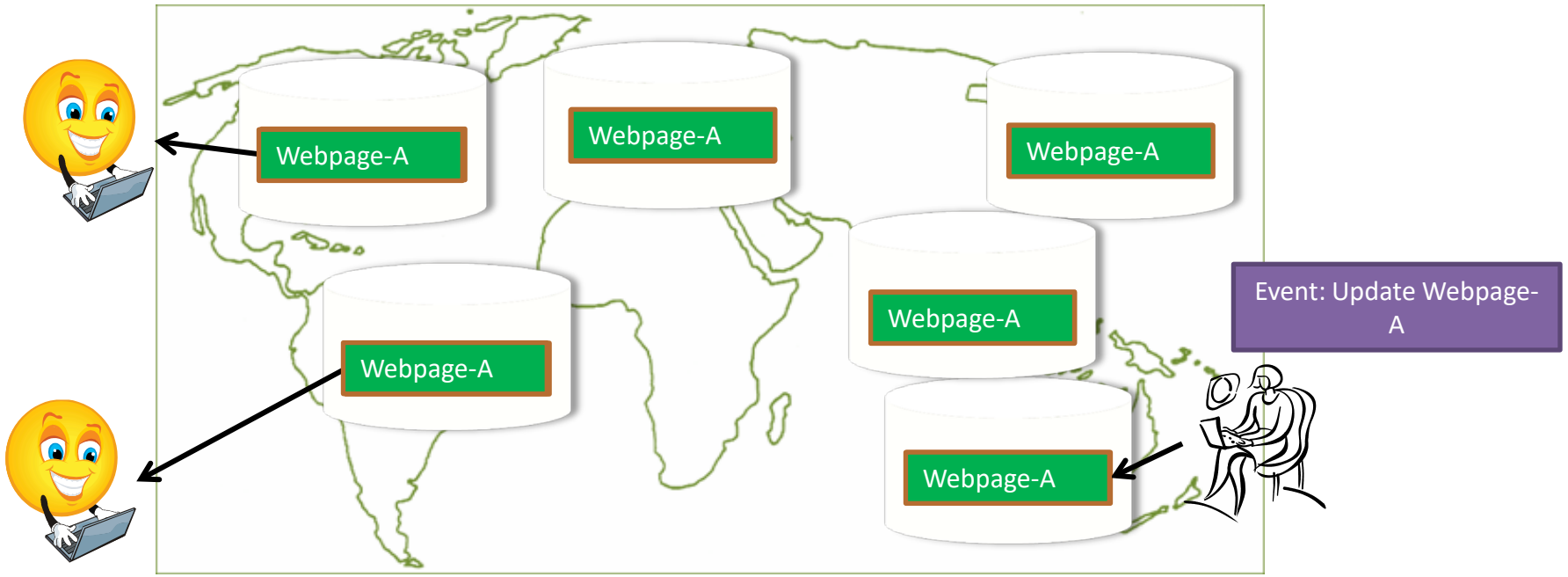
  Consistência e disponibilidade

- NoSQL usam Propriedades BASE:
  - ***B**asically **A**vailable*: sistema garante disponibilidade
  - ***S**oft-State*: o estado do sistema pode mudar ao longo do tempo
  - ***E**ventual Consistency*: sistema eventualmente se torna consistente

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Consistência Eventual

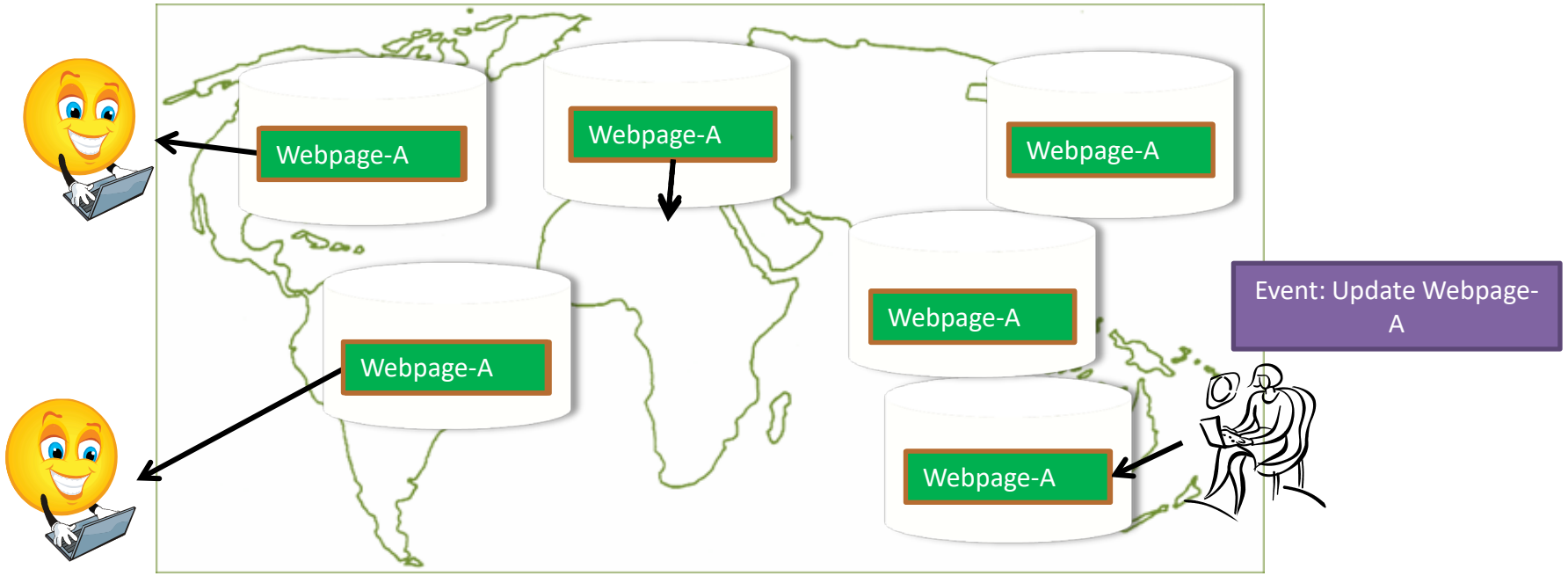
Todas réplicas se tornam consistentes na ausência de atualizações



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Consistência Eventual: *Principal Desafio*

Se o cliente acessar dados de diferentes réplicas?



Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

ACID X BASE

- **ACID** (**A**tomicity, **C**onsistency, **I**solation, **D**urability)

  Consistência e disponibilidade

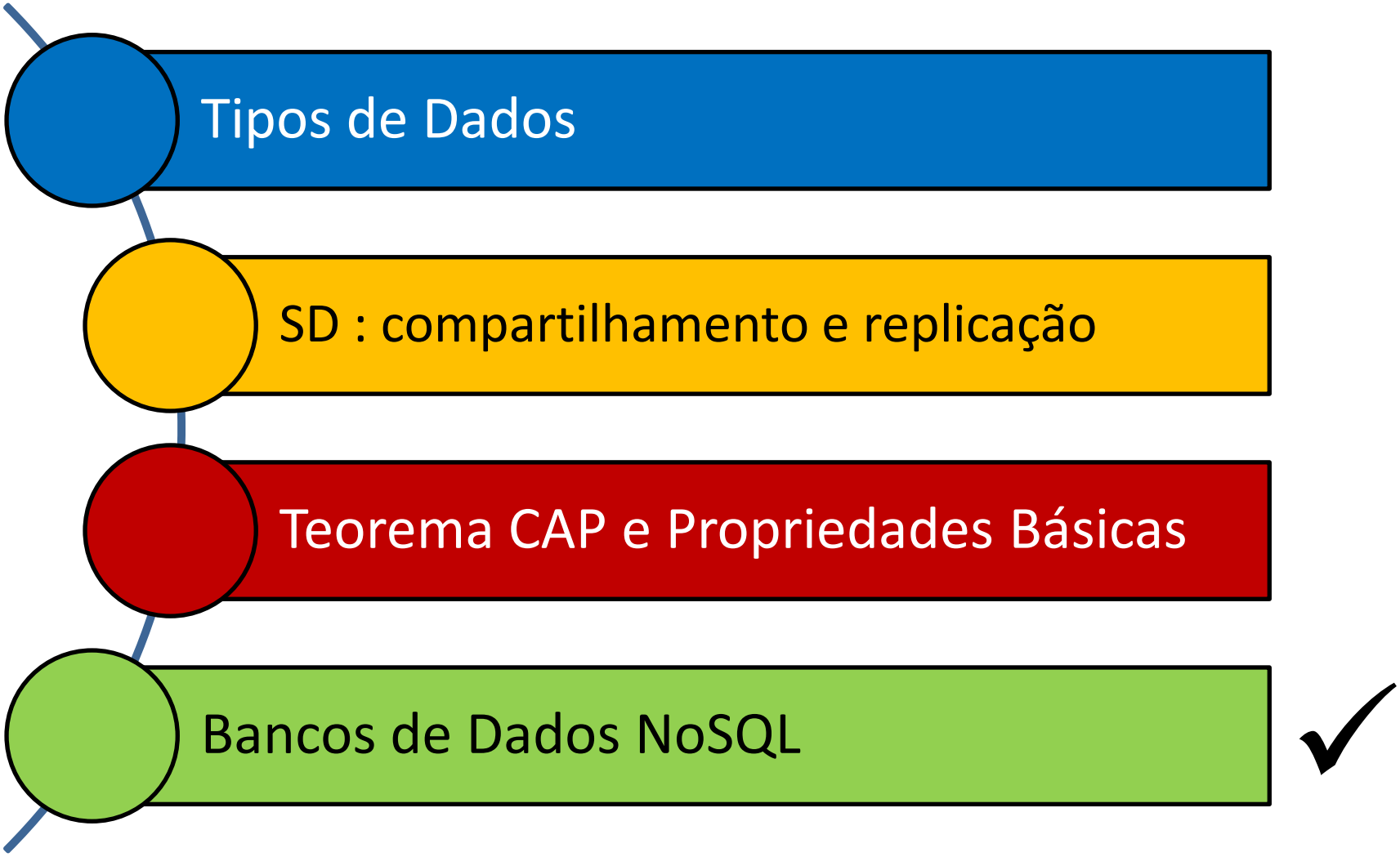
- **BASE** (**B**asically **A**vailable, **S**oft state, **E**ventually consistent)

  Consistência e Tolerância a particionamento

  Disponibilidade e Tolerância a particionamento

Essa parte da aula é uma tradução do material do prof. Mohammad Hammoud (Carnegie Mellon)

Sumário


Tipos de Dados
SD : compartilhamento e replicação
Teorema CAP e Propriedades Básicas
Bancos de Dados NoSQL

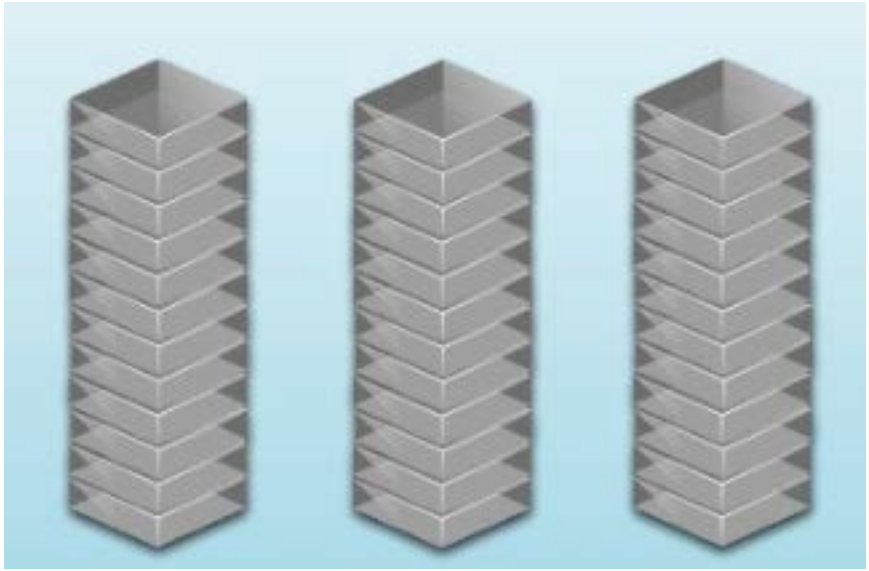
NoSQL Databases

- Comunidade de BD migra para as propriedades BASE
  - BD NoSQL
  - E.g., Amazon's Dynamo and Google's Bigtable

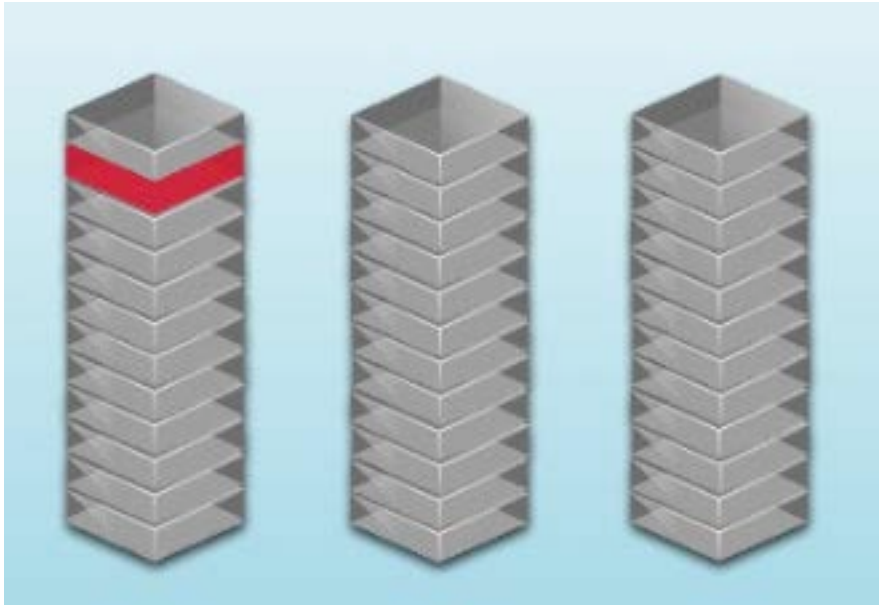
- Principais características dos BD NoSQL
  - Não há requisitos rígidos de esquema
  - Não há uma aderência rígida às propriedades ACID
  - Consistência é negociada em favor da disponibilidade

Antes de BD NoSQL
Vamos revisar RELACIONAL

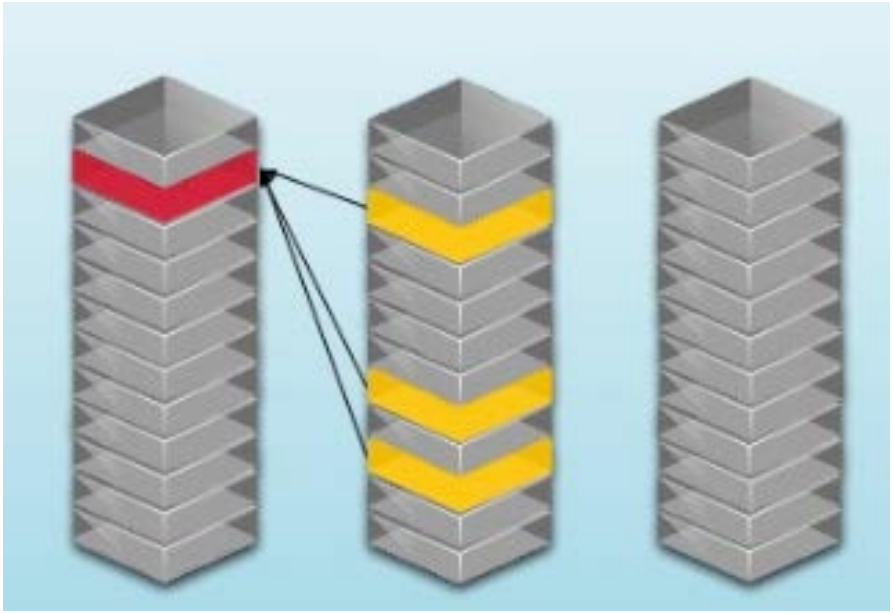
Relacional (revisão)



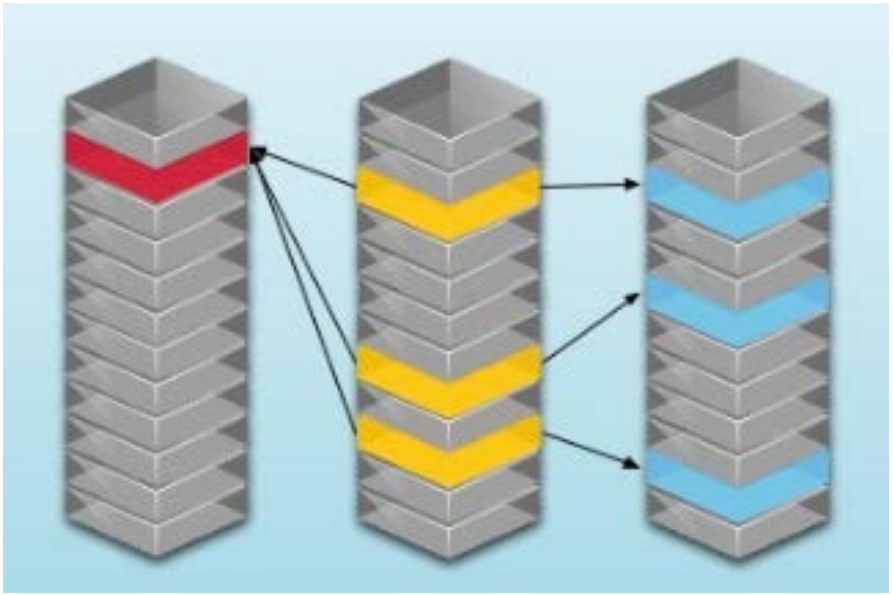
Relacional (revisão)



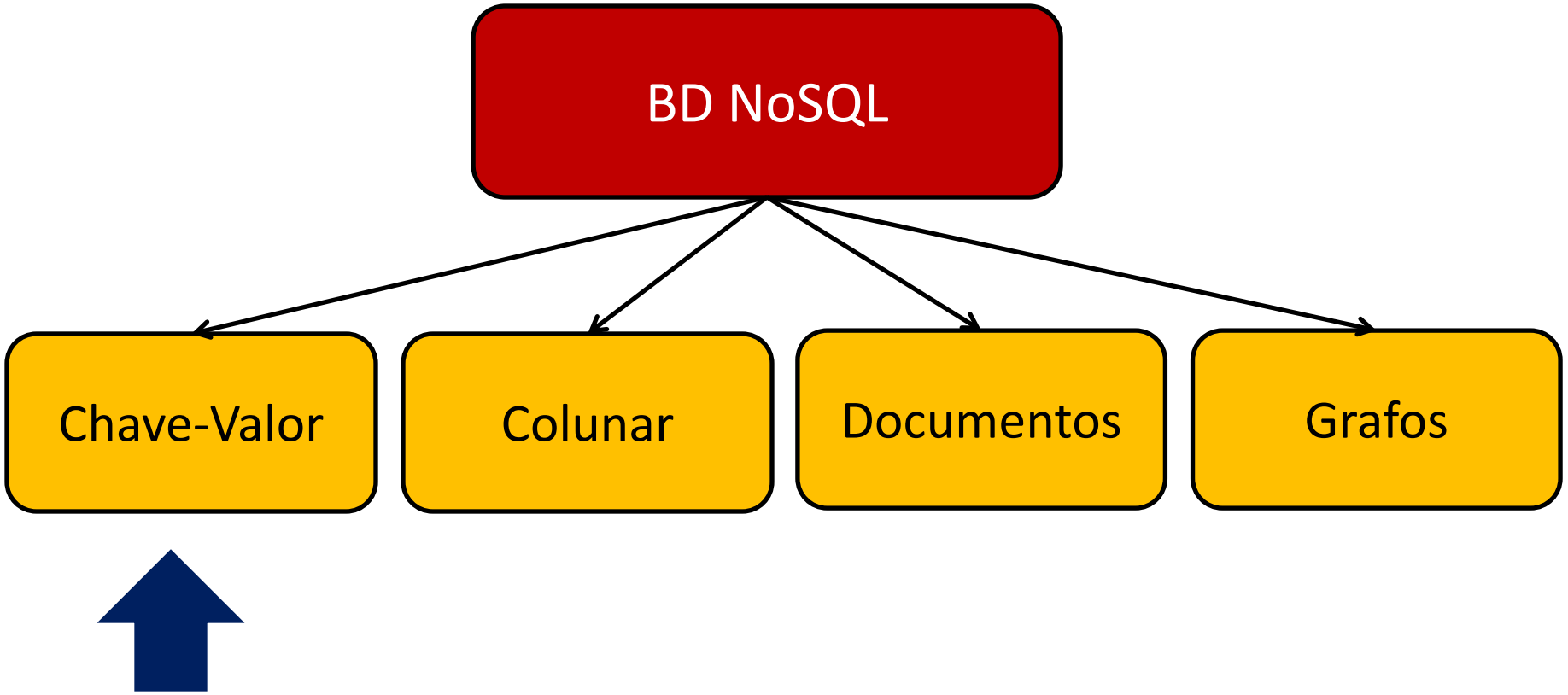
Relacional (revisão)



Relacional (revisão)



Tipos de BD NoSQL



Chave-Valor

Chave	Valor
CPF1	88456707830
Nome1	Webber
Telefone1	999999999
NVI1	9BWHE21JX24060960
Modelo1	Gol 2013
Preço1	28.000,00
CPF2	76052657278
Nome2	Sidartha
Telefone2	[999572110, 988081046]

Chave-Valor

- Forma mais simples de um sistema de armazenamento de dados
- Armazena os dados no formato de pares **<chave, valor>**
- Permite consulta indexada a valores dada uma determinada chave
- Chaves podem ser armazenadas em uma tabela *hash* e facilmente distribuidas
- Permitem as operacões de BD CRUD (*create*, *read*, *update*, and *delete*)

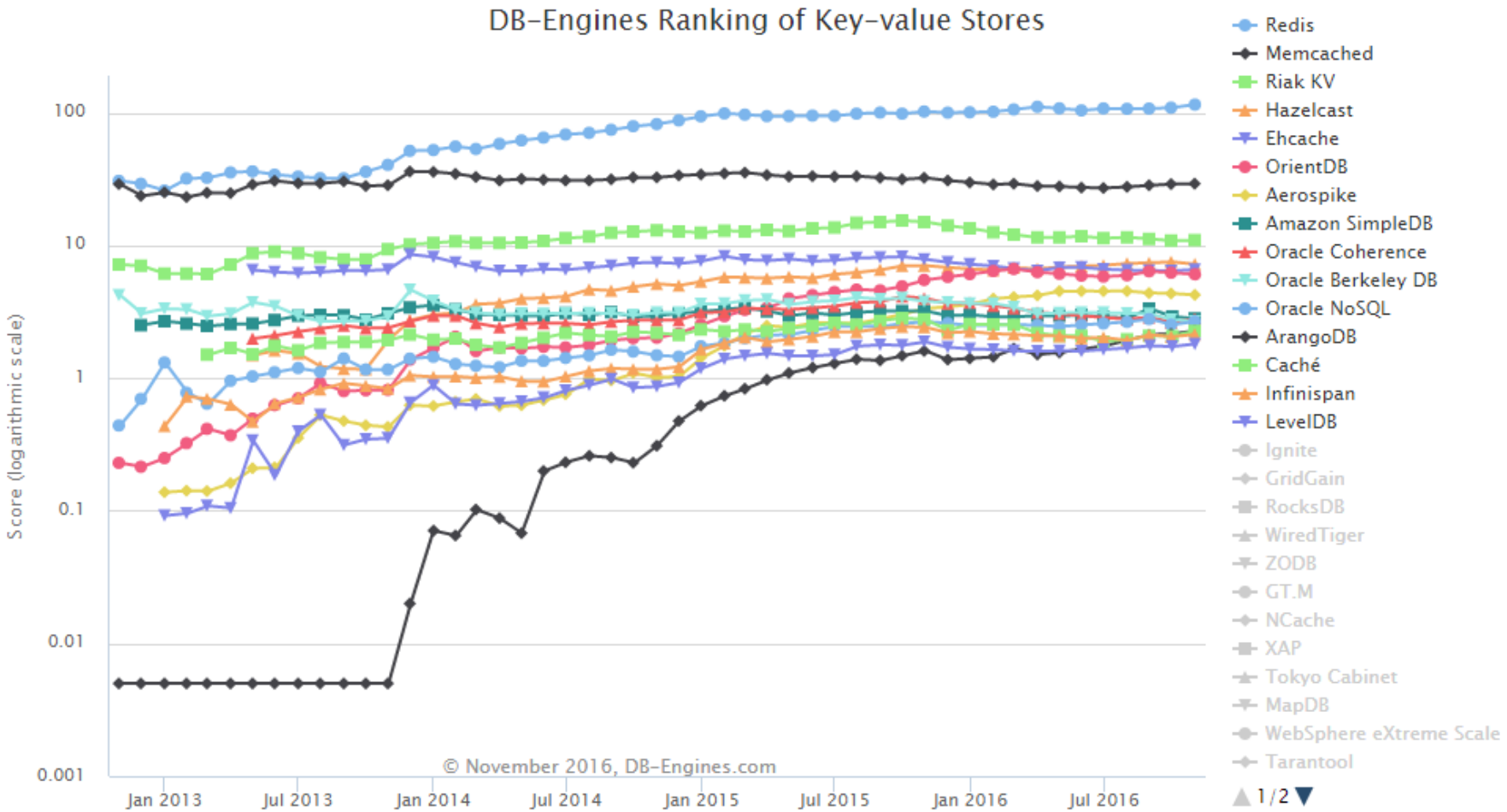
  Não permitem junções e agregação

- Exemplos: Amazon DynamoDB and Apache Cassandra

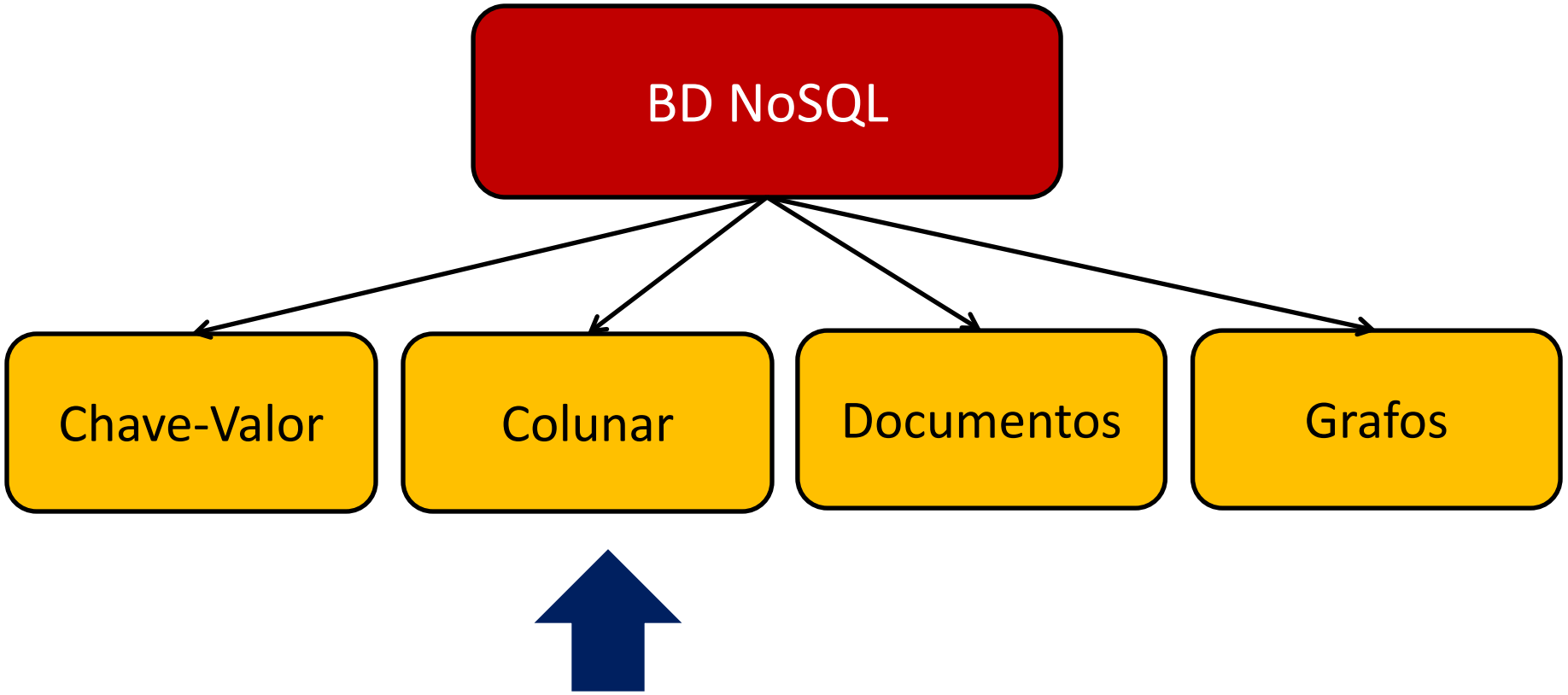
Por que usar Chave-Valor?

Menor tempo de resposta permitindo que a capacidade de armazenamento de suas bases de dados seja uma das maiores dos sistemas enquadrados no conceito NoSQL

Chave-Valor


DB-Engines Ranking of Key-value Stores
Score (logarithmic scale)
100
10
1
0.1
0.01
0.001
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
© November 2016, DB-Engines.com
Redis
Memcached
Riak KV
Hazelcast
Ehcache
OrientDB
Aerospike
Amazon SimpleDB
Oracle Coherence
Oracle Berkeley DB
Oracle NoSQL
ArangoDB
Caché
Infinispan
LevelDB
Ignite
GridGain
RocksDB
WiredTiger
ZODB
GT.M
NCache
XAP
Tokyo Cabinet
MapDB
WebSphere eXtreme Scale
Tarantool
1/2

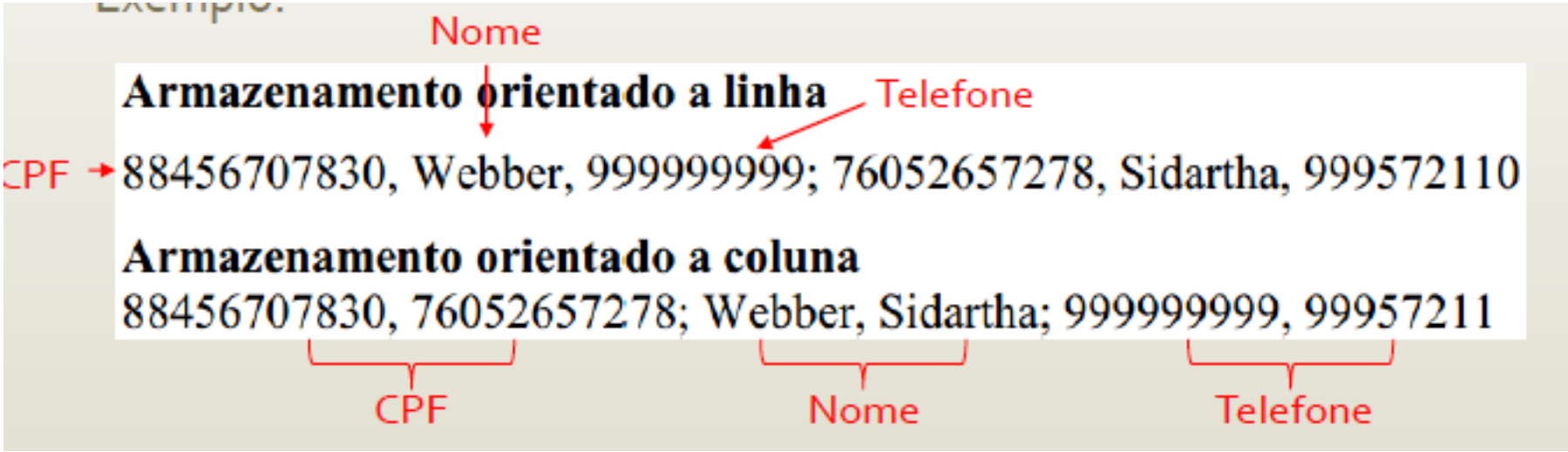
Tipos de BD NoSQL



Colunar

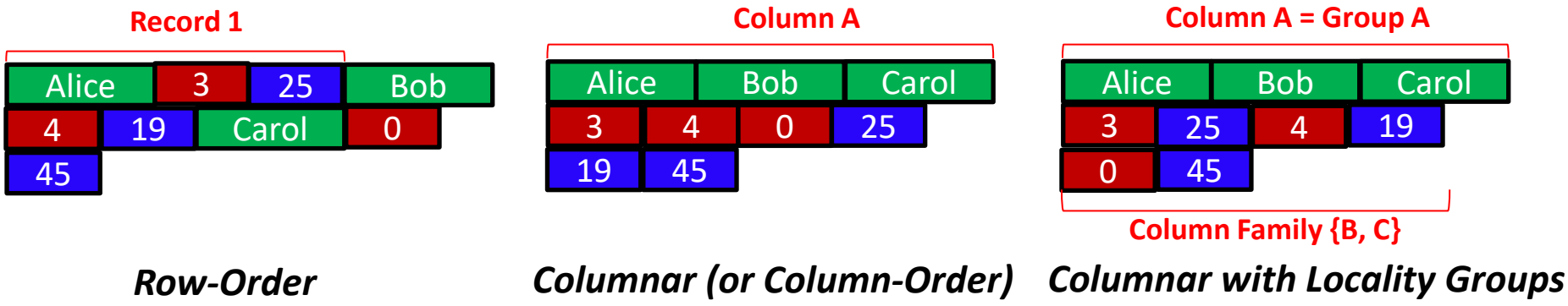
- Armazena os dados em registros e colunas, assim como os bancos de dados relacionais
- Baseados no *BigTable* da Google
- Nomes e formato das colunas podem variar em cada linha do conjunto de dados

Colunar


Nome
Armazenamento orientado a linha
Telefone
CPF
88456707830, Webber, 999999999; 76052657278, Sidartha, 999572110
Armazenamento orientado a coluna
88456707830, 76052657278; Webber, Sidartha; 999999999, 99957211
CPF
Nome
Telefone

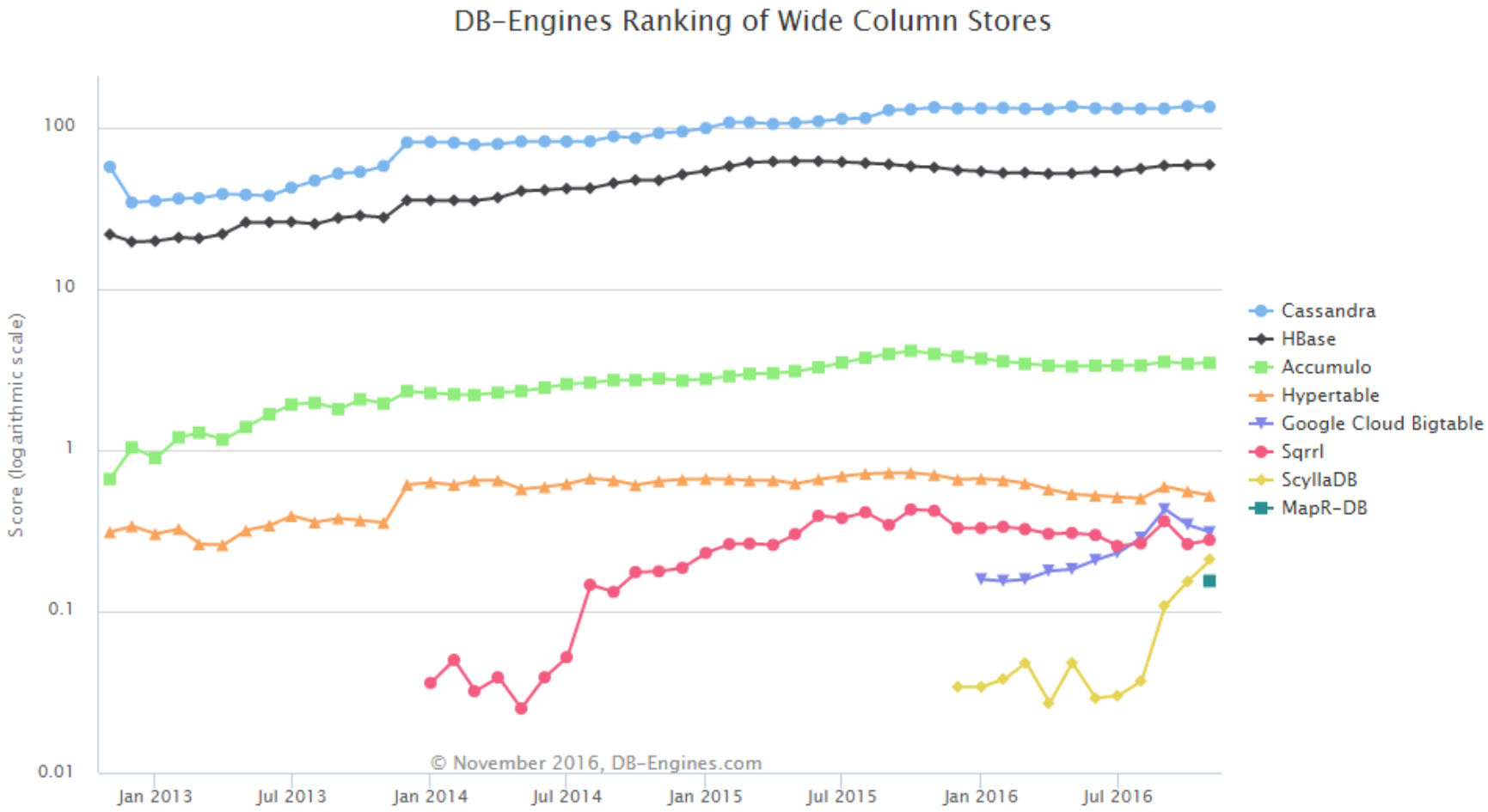
Colunar

- Híbrido : relacional e chave-valor
  - Valores são armazenados em grupos de zero ou mais colunas, mas em ordem

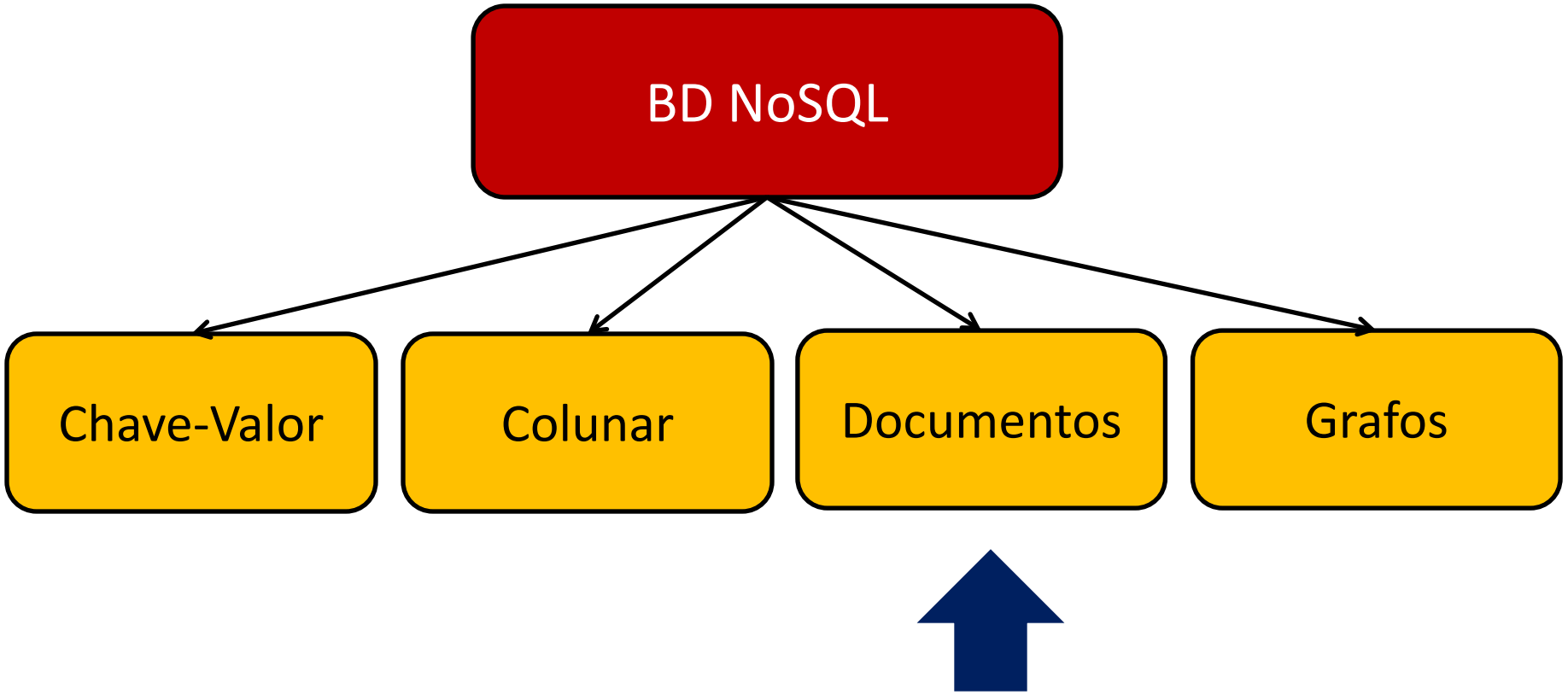


  - Valores são consultados por casamento (*match*) de chaves
- Exemplos: HBase and Vertica

Colunar


DB-Engines Ranking of Wide Column Stores
Score (logarithmic scale)
100
10
1
0.1
0.01
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
© November 2016, DB-Engines.com
Cassandra
HBase
Accumulo
Hypertable
Google Cloud Bigtable
Sqrrl
ScyllaDB
MapR-DB

Tipos de BD NoSQL



Documentos

```
{
  _id: “88456707830”,
  Nome: “Webber”,
  Telefone: “999999999”
}
{
  _id: “76052657278”,
  Nome: “Sidartha”,
  Telefones: [
    {telefone: “999957211”},
    {telefone: “988081046”}
  ]
}
{
}
```

Documentos

- Documentos são armazenados em um formato padrão (ex., XML, JSON, PDF or Office Documents)
  - Chamados de *Binary Large Objects* (BLOBs)
- Diferentes registros no mesmo conjunto de dados podem ter estruturas diferentes
- Colunas multivaloradas (*arrays*) e documentos aninhados
- Documentos podem ser indexados
  - Permite armazenamento
  - Superam os tradicionais sistemas de arquivo

- Exemplos: MongoDB and CouchDB (ambos podem usar MapReduce)

Documentos

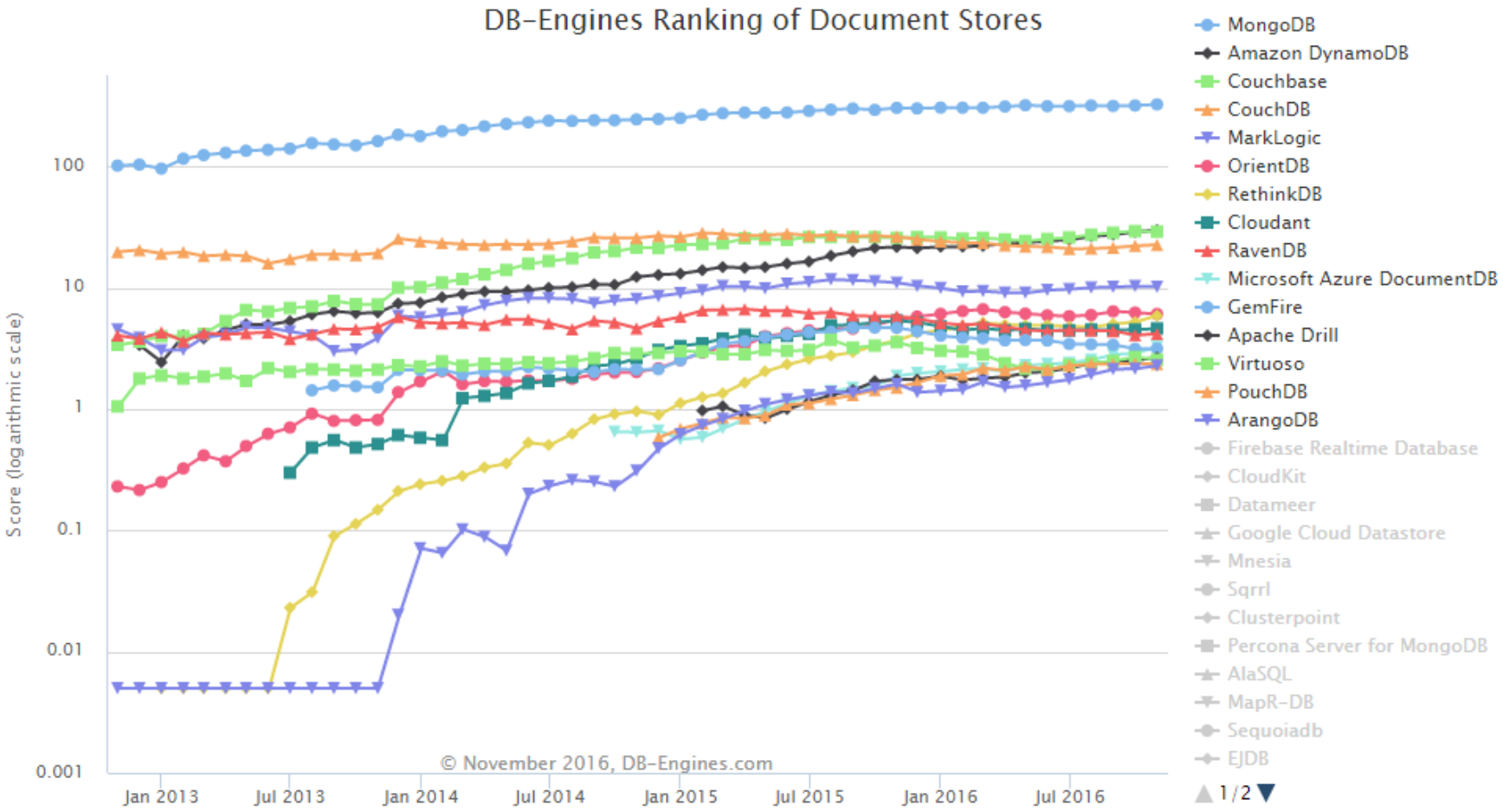
Implementação	Documentos
Aninhamento (*embedded*)	{ _id: “76052657278”, Nome: “Sidartha”, Telefones: [{telefone: “999957211”}, {telefone: “988081046”}], Carro: { NVI: “C021E21JX240LP925”, Modelo: “fiesta 2012”, Preço: “15.000,00” } }

Documentos

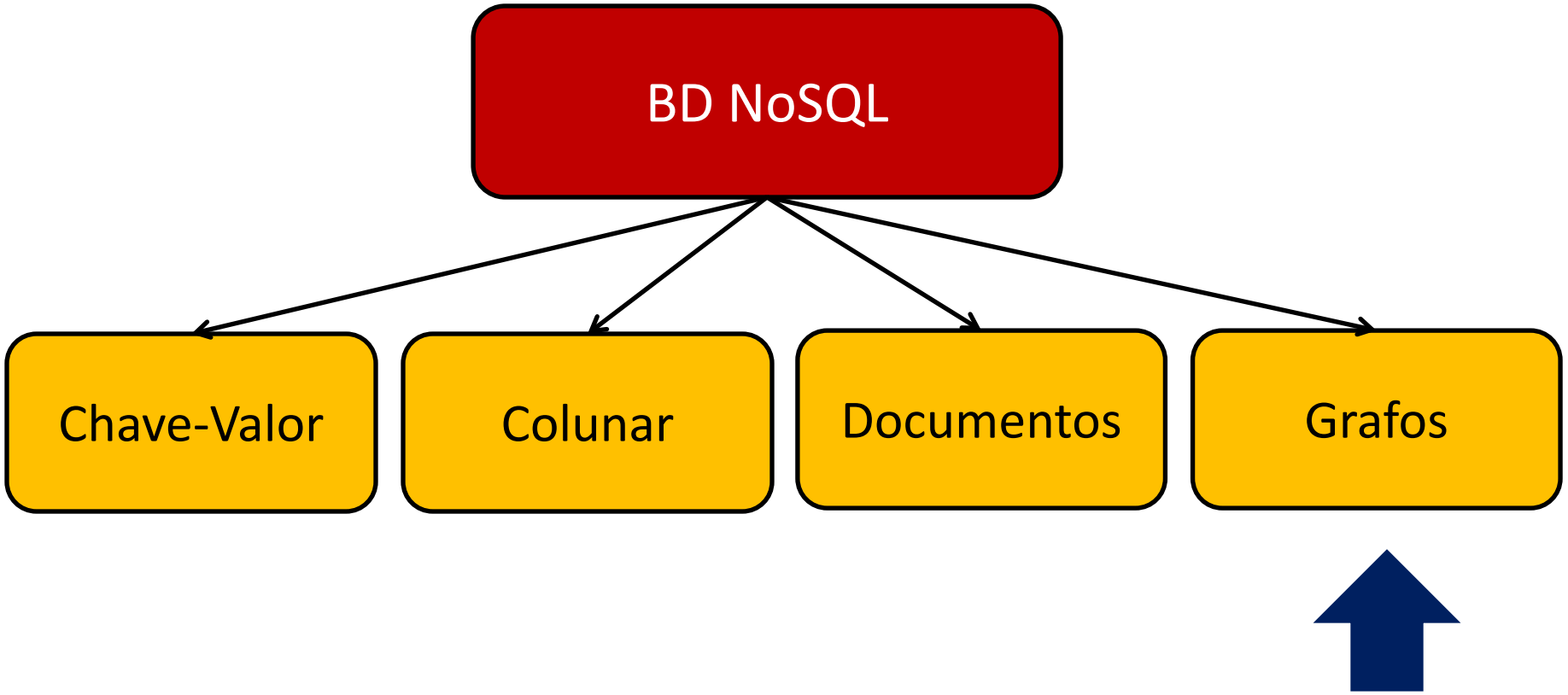
Implementação	Documentos

DBRef	{
	_id: “88456707830”,
	Nome: “Webber”,
	Telefone: “999999999”,
	Carro: objectId(“9BWHE21JX24060960”)
	}
	---------- // ----------
	{
	_id: “9BWHE21JX24060960”,
	Modelo: “gol 2015”,
	Preço: 28.000,00
	}

Documentos

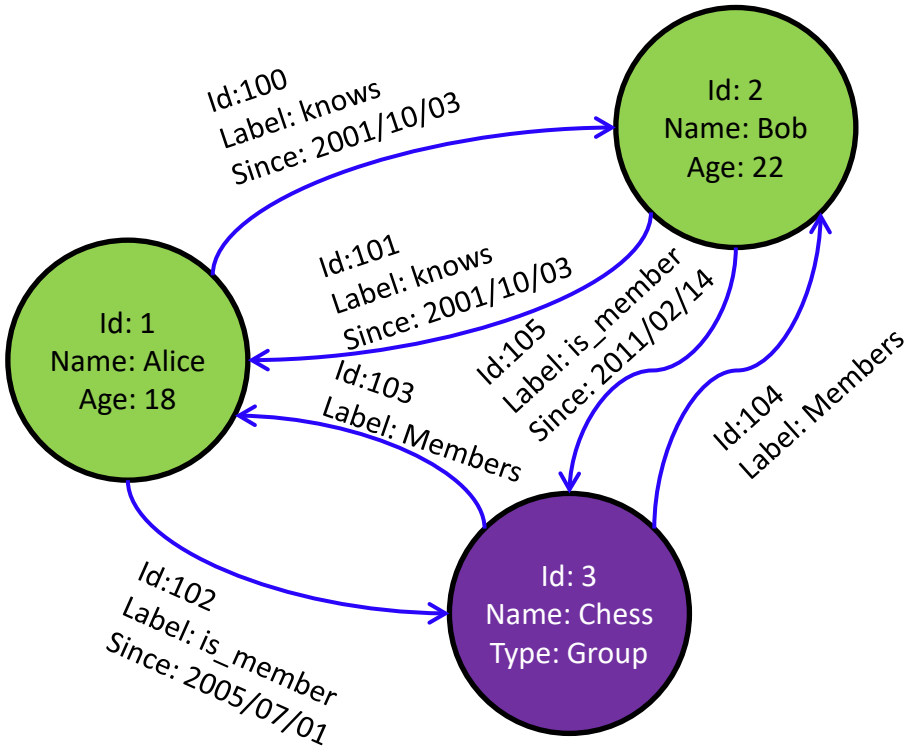

DB-Engines Ranking of Document Stores
Score (logarithmic scale)
100
10
1
0.1
0.01
0.001
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
© November 2016, DB-Engines.com
MongoDB
Amazon DynamoDB
Couchbase
CouchDB
MarkLogic
OrientDB
RethinkDB
Cloudant
RavenDB
Microsoft Azure DocumentDB
GemFire
Apache Drill
Virtuoso
PouchDB
ArangoDB
Firebase Realtime Database
CloudKit
Datameer
Google Cloud Datastore
Mnesia
Sqrrl
Clusterpoint
Percona Server for MongoDB
AlaSQL
MapR-DB
Sequoiadb
EJDB
1/2

Tipos de BD NoSQL



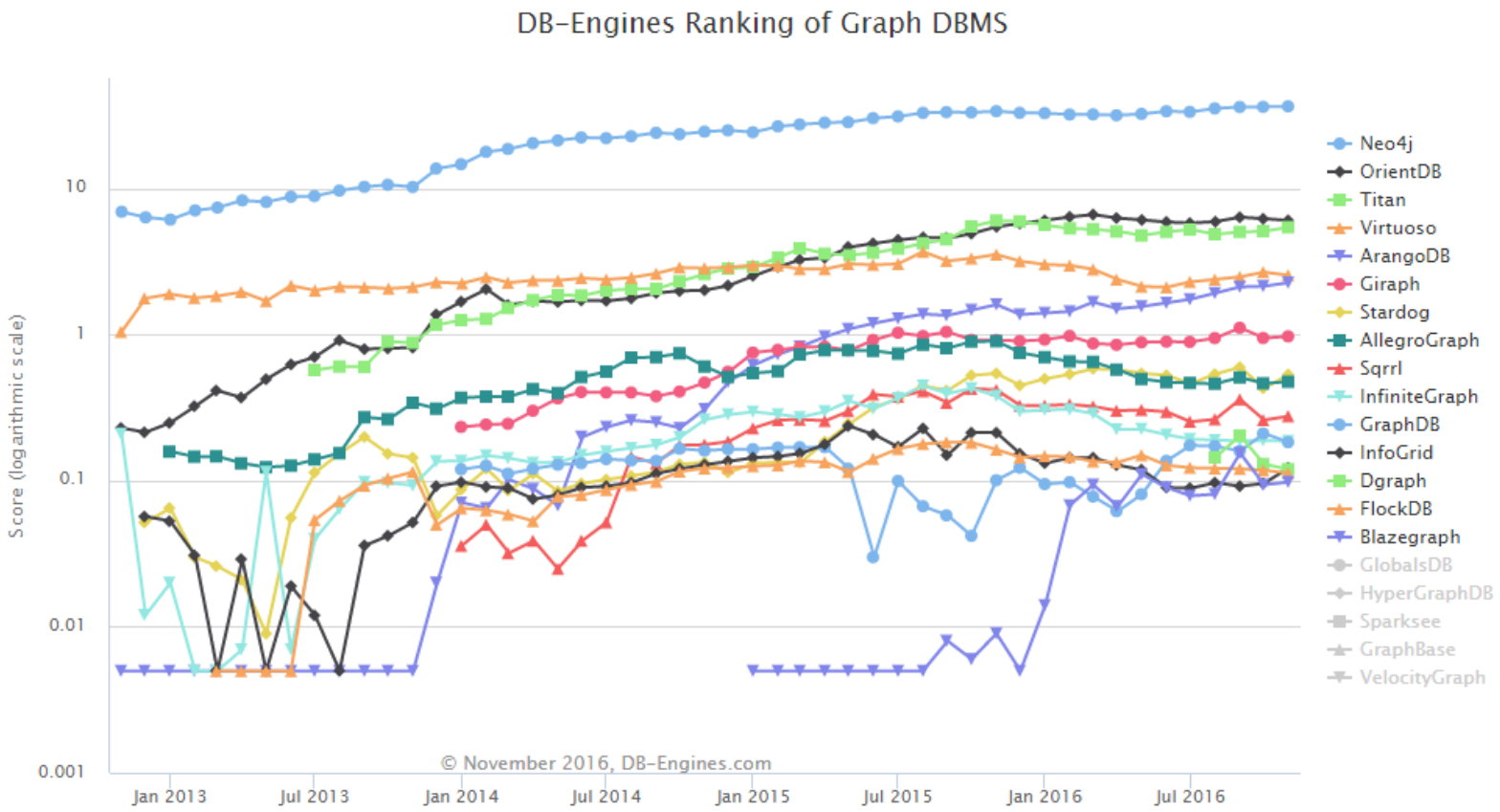
Grafos

- Dados são representados por vértices e arestas

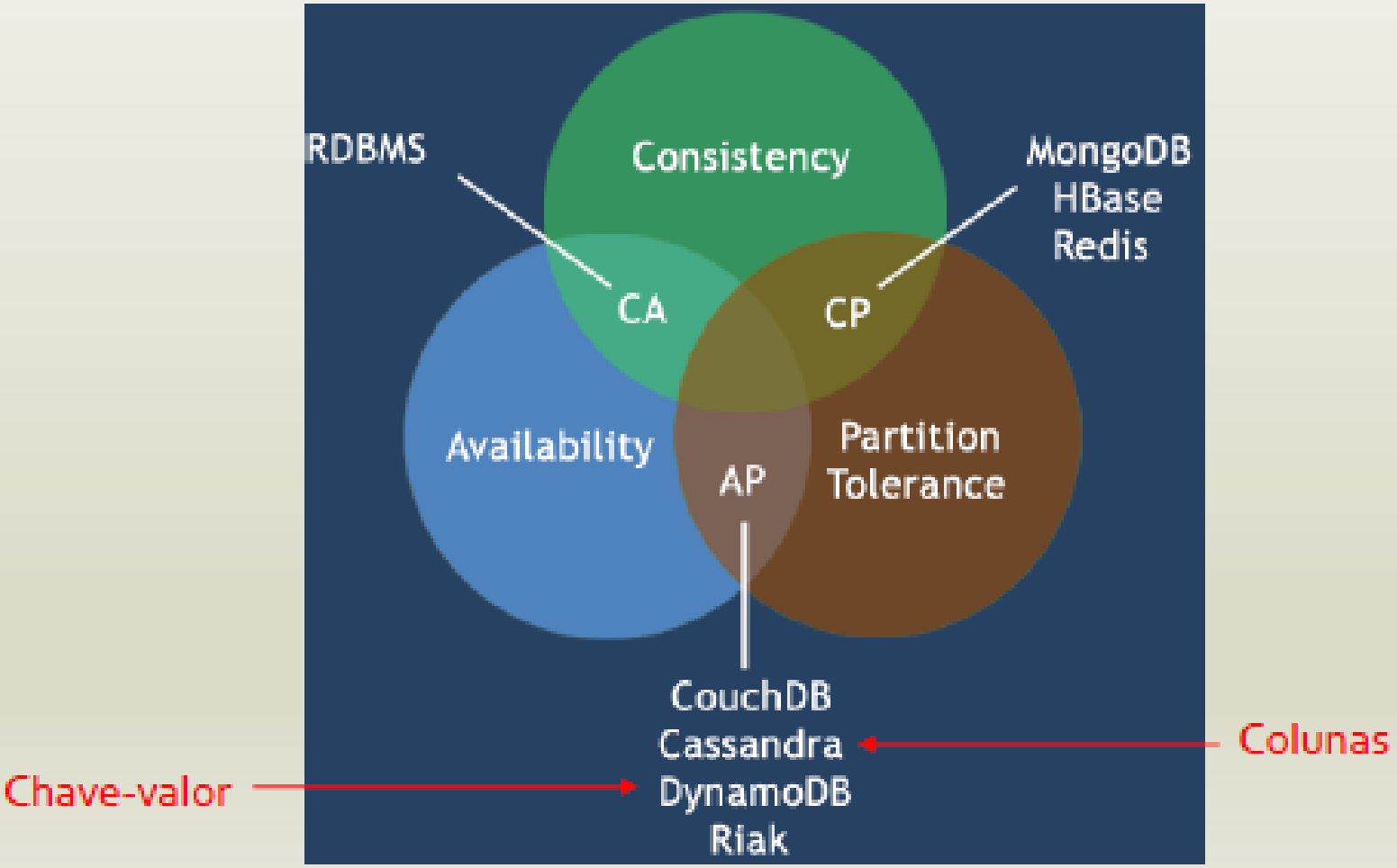


- Grande poder de consulta :: navegação entre vértices e arestas
- Envolvem uma grande quantidade de junções complexas
  - Google PageRank
  - Reachability
- Exemplos: Neo4j and VertexDB

Grafos


DB-Engines Ranking of Graph DBMS
Score (logarithmic scale)
10
1
0.1
0.01
0.001
© November 2016, DB-Engines.com
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Jan 2016
Jul 2016
Neo4j
OrientDB
Titan
Virtuoso
ArangoDB
Giraph
Stardog
AllegroGraph
Sqrrl
InfiniteGraph
GraphDB
InfoGrid
Dgraph
FlockDB
Blazegraph
GlobalsDB
HyperGraphDB
Sparksee
GraphBase
VelocityGraph

CAP *versus* Modelos



Fabricantes

- Chave-Valor
  - Dynamo
  - Azure Table Storage
  - Couchbase Server
  - Riak
  - Redis
  - LevelDB
  - Chordless
  - GenieDB

- Scalaris
- Tokyo
- Cabinet/Tyrant
- Berkeley DB
- Voldemort
- Dynomite
- MemcacheDB
- Faircom C-Tree
- HamsterDB
- RaptorDB
- allegro-C
- BangDB

Fabricantes

- Colunar
  - Hbase
  - Cassandra
  - Hypertable
  - Accumulo
  - Amazon SimpleDB
  - Cloudata
  - Cloudera
  - SciDB
  - HPCC
  - Stratosphere;

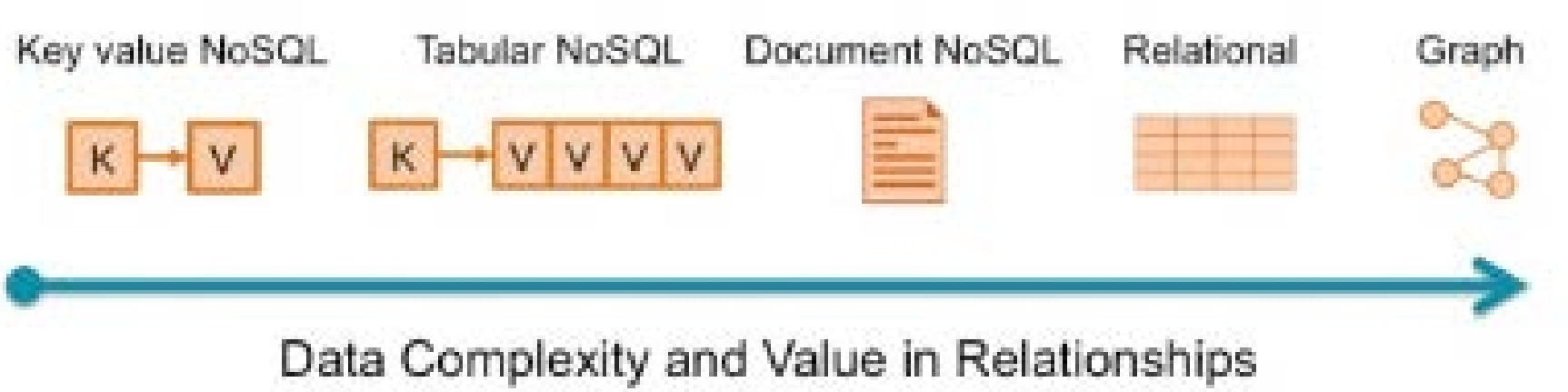
Fabricantes

- Documentos
  - MongoDB
  - CouchDB
  - BigCouch
  - RavenDB
  - Clusterpoint Server
  - ThruDB
  - TerraStore
  - RaptorDB
  - JasDB
  - SisoDB
  - SchemaFreeDB
  - djondb

Fabricantes

- Grafos
  - Neo4J
  - Infinite Graph
  - Sones
  - InfoGrid
  - HyperGraphDB
  - DEX
  - Trinity
  - AllegroGraph
  - BrightStarDB
  - OpenLink Virtuoso
  - VertexDB
  - FlockDB

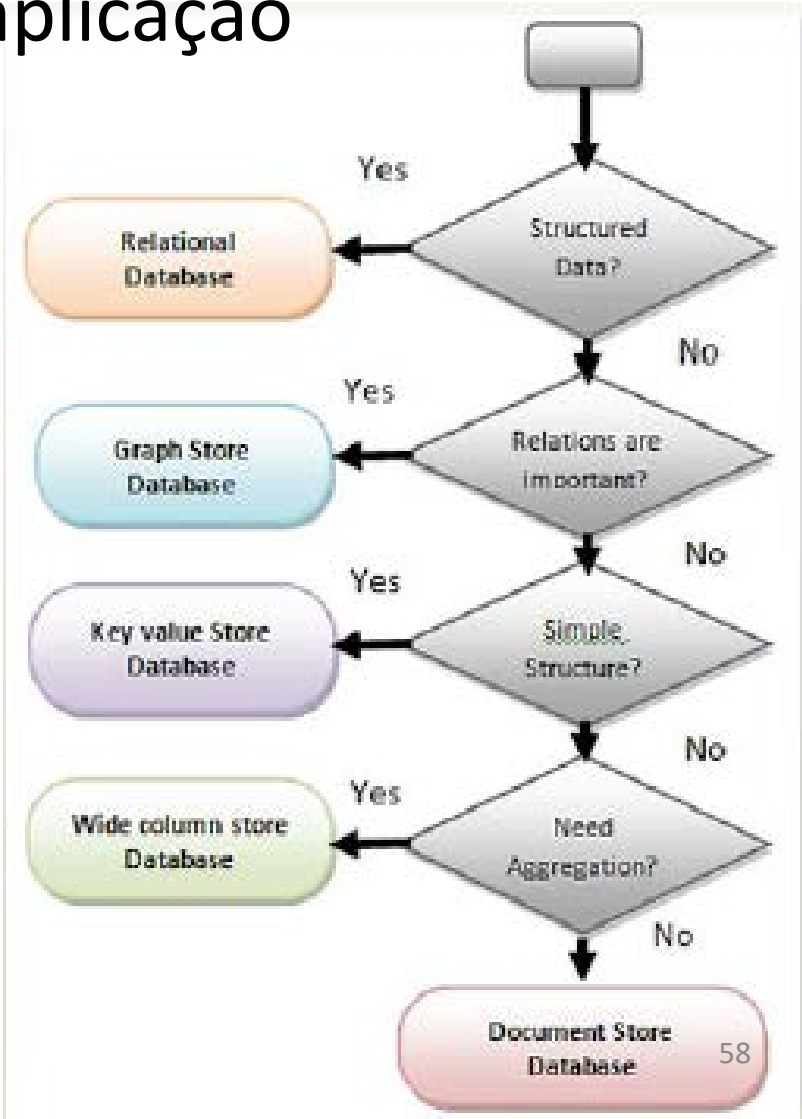
Complexidade do Modelos de BD



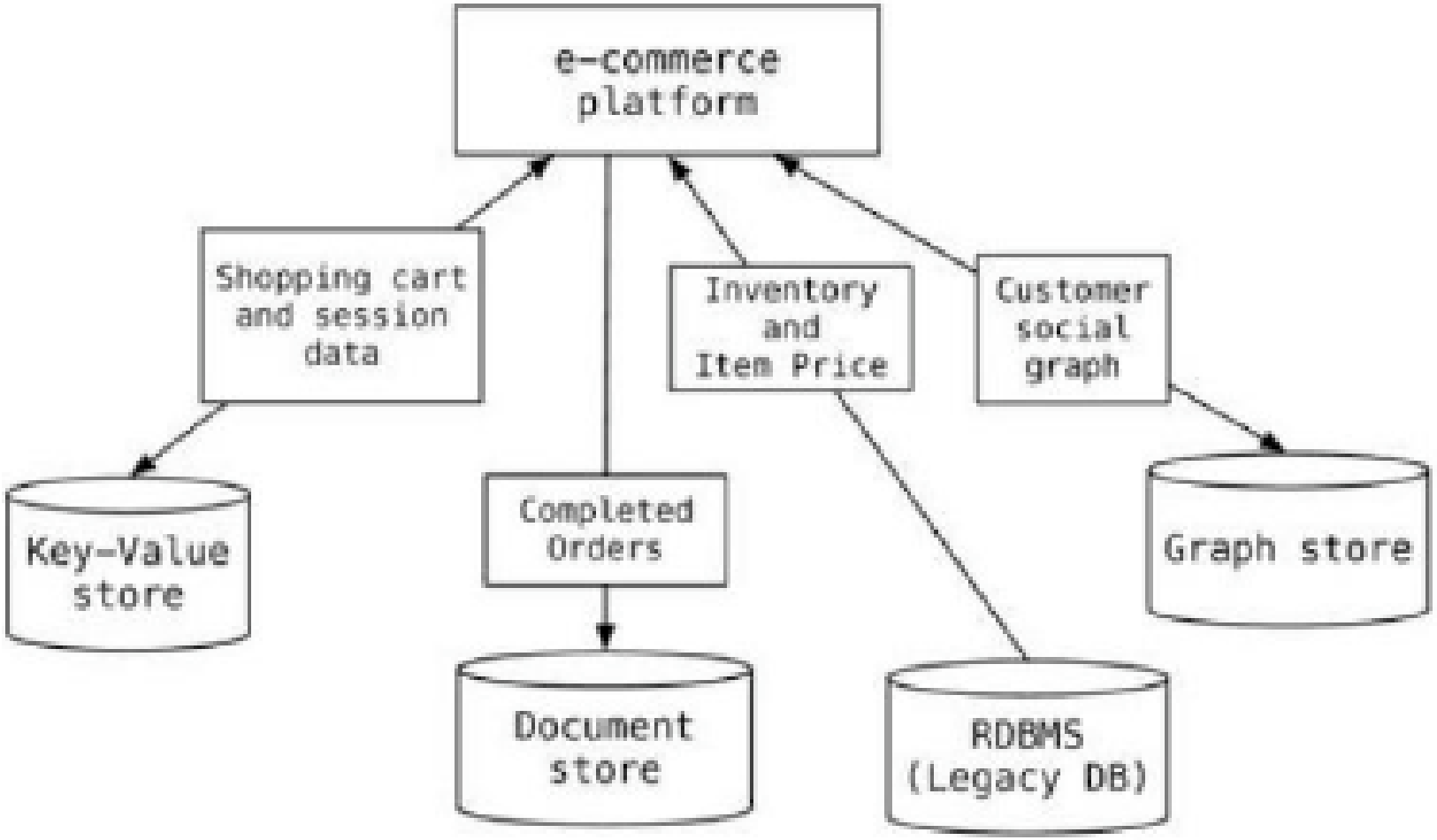
Como escolher o melhor modelo

Depende dos requisitos da aplicação

- Tamanho dos Dados
- Complexidade
- Teorema CAP
- Formato de Dados



Persistência Poliglota



Resumo da Aula

- Dados classificados em 4 tipos: *estruturados*, *não estruturados*, *dinâmicos* and *estáticos*
- Diferentes tipos de dados levam a escolher diferentes BD NoSQL
- Bancos de dados escalam: distribuição ou replicação
- Consistência pode limitar a escalabilidade

Resumo da Aula

- Teorema *CAP* BD distribuídos que compartilham dados podem ter 2 de 3 propriedades
  - ***C****onsistency*
  - ***A****vailability*
  - ***P****artition Tolerance*
- O teorema CAP levou à produção de vários BD com as propriedades ACID relaxadas

Resumo da Aula

- *NoSQL* seguem as propriedades *BASE*
  - ***B**asically **A**vailable*
  - ***S**oft-State*
  - ***E**ventual Consistency*
- Tipos de BD NoSQL
  - Chave-Valor (*Key-Value Stores*)
  - Documentos (*Document Stores*)
  - Grafo (*Graph Databases*)
  - Colunar (Colu*mnar Databases*)