

# **UNIDADE: TÓPICOS AVANÇADOS**

***Bancos não-relacionais***

***Prof. Roberson Alves***

1

# AGENDA

- ✓ **Bancos não-relacionais**
- ✓ **NoSQL**
- ✓ **Modelos de Banco NoSQL**

# **Baseado no material de Andréa Bordin – NoSQL: uma breve introdução**

# O QUE SIGNIFICA?

- NoSQL é um termo genérico que define bancos de dados *não-relacionais*.
- A tecnologia NoSQL foi iniciada por companhias líderes da Internet - incluindo Google, Facebook, Amazon e LinkedIn - para superar as limitações (45 anos de uso da tecnologia) de banco de dados relacional para aplicações web modernas(2008).

# BANCO NÃO-RELACIONAIS

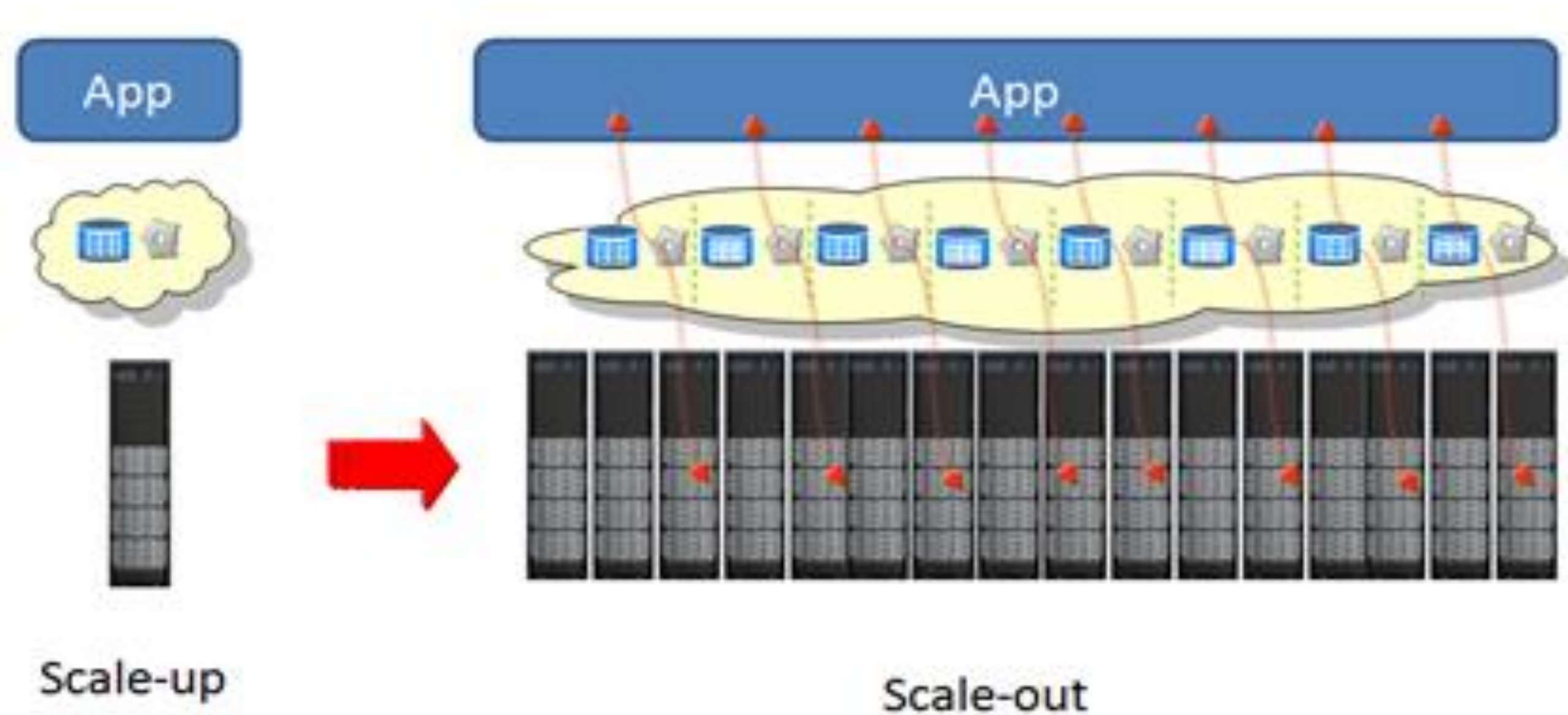
## **Características em comum:**

- ✓ **tais como serem livres de esquema;**
- ✓ **promoverem alta disponibilidade; e**
- ✓ **maior escalabilidade**

## **Características em singulares:**

- **certos sistemas promovem**
- **o particionamento e a replicação dos dados**
- **sistemas baseados em armazenamento chave-valor**
- **sistemas orientados a documentos**
- **sistemas orientados a coluna**
- **sistemas baseados em grafos**

# ESCALABILIDADE VERTICAL VS HORIZONTAL



# POR QUE NOSQL?

- Hoje as empresas estão adotando NoSQL para um número crescente de aplicações/casos de uso.
- A escolha que é impulsionada por quatro megatendências inter-relacionadas:
  - Big Users
  - Big Data
  - Internet das coisas
  - Cloud Computing

# BIG USERS



**3**

**Billion  
Global Online  
Population**



**35**

**Billion Hrs./Mo.  
Spent Online**



**1.75**

**Billion  
Smartphone  
Users**

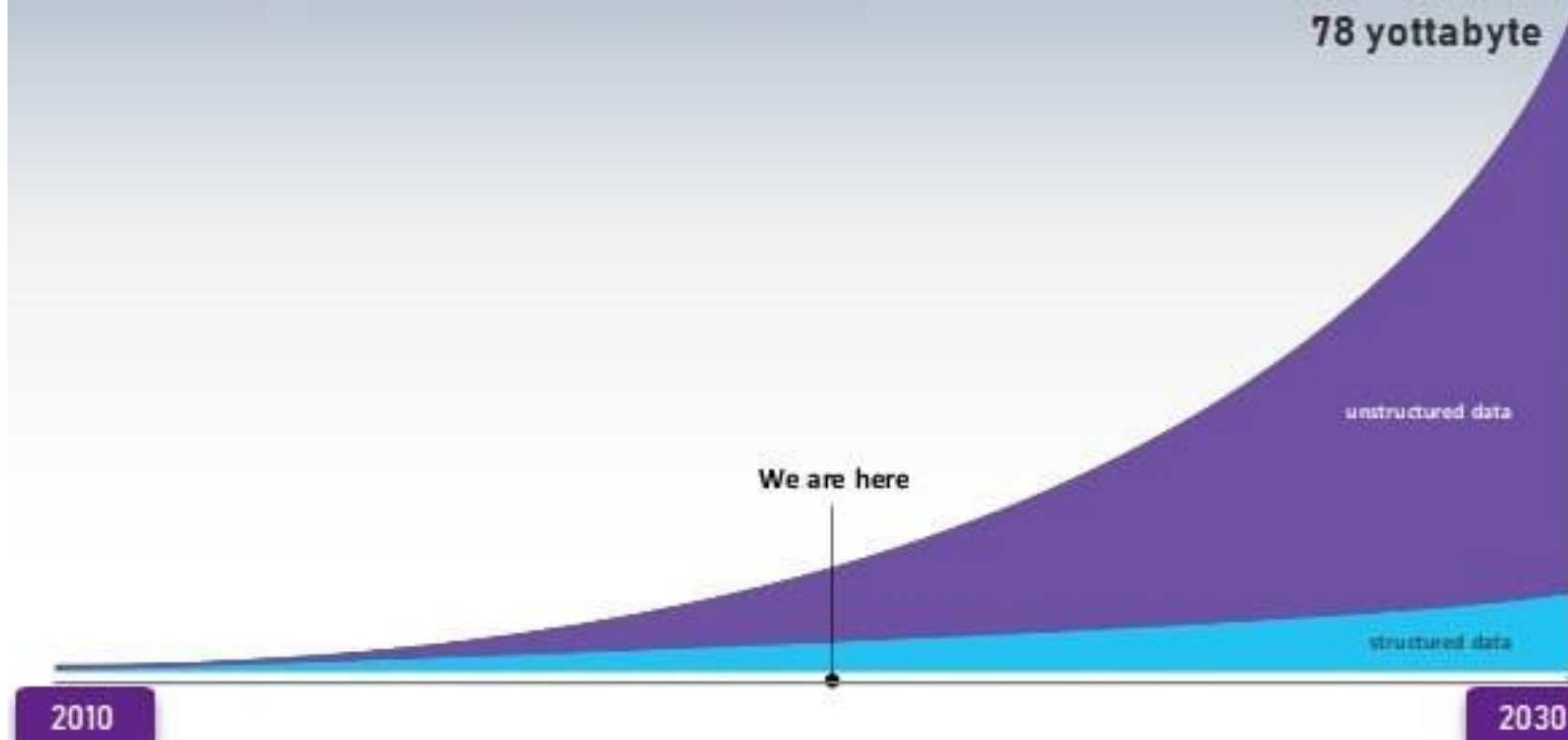
O crescente uso de aplicativos online resultou em um número crescente de operações de banco de dados e uma necessidade de uma maneira mais fácil de **escalar** bancos de dados para atender a essas demandas.

Um grande número de usuários, combinados com a natureza dinâmica dos padrões de uso está demandando uma tecnologia de banco de dados mais facilmente **escalável**.

**NoSQL é a solução.**



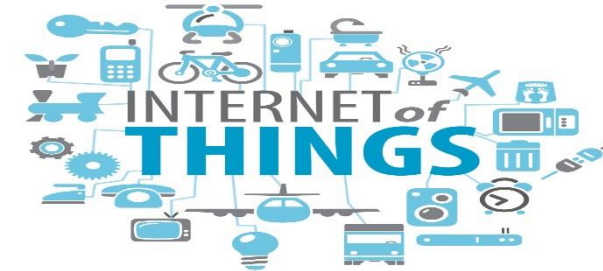
# BIG DATA



É necessário uma solução altamente flexível, que acomode facilmente qualquer novo tipo de dado (não-estruturado e semi-estruturado) e que não seja corrompida por mudanças na estrutura de conteúdo.

NoSQL fornece um modelo de dados **sem esquema** muito mais flexível que mapeia melhor a organização de dados de uma aplicação e simplifica a interação entre a aplicação e o banco de dados, resultando em menos código para escrever, depurar e manter.

# A INTERNET DAS COISAS



32 bilhões de coisas vão estar conectadas a internet

10% de todos os dados serão gerados por sistemas embarcados

21% dos mais valiosos dados serão gerados por sistemas embarcados

Dados de telemetria - semi-estruturados e contínuos - representam um desafio para bancos de dados relacionais, que exigem um esquema fixo e dados estruturados.

Empresas inovadoras estão utilizando tecnologia NoSQL para dimensionar o acesso simultâneo de dados para milhões de dispositivos e sistemas conectados, armazenar bilhões de pontos de dados e atender aos requisitos de infra-estrutura e operações de missão crítica de performance.

# CLOUD COMPUTING

Atualmente a maioria das novas aplicações são executadas em um sistema em nuvem privado, público ou híbrido, suportam um grande número de usuários e usam uma arquitetura de internet de três camadas.

Na camada de banco de dados, bancos de dados relacionais são originalmente a escolha popular.

Seu uso é cada vez mais problemático porque eles são uma tecnologia centralizada, cuja escalabilidade é vertical ao invés de horizontal.

Isso não os torna adequados para aplicações que requerem escalabilidade fácil e dinâmica.

Bancos de dados NoSQL são construídos a partir do zero para serem distribuídos, escaláveis dinamicamente e são, portanto, mais adequados à natureza altamente distribuída da arquitetura de três camadas da internet.

# MODELO DE DADOS MAIS FLEXÍVEL

```
{  
  "ID":1,  
  "FIRST":"Frank",  
  "LAST":"Weigel",  
  "ZIP": "94040",  
  "CITY": "MV",  
  "STATE": "CA"  
}
```

JSON

=

KEY	First	Last	ZIP_id
1	Frank	Weigel	2
2	Ali	Bob	2
3	Mark	Azad	2
4	Steve	Yen	3

+

ADDRESS INFO

ZIP_id	City	State	ZIP
1	DEN	CO	30303
2	MV	CA	94040
3	CHI	IL	60609
4	NY	NY	10010

# MODOS DE ARMAZENAMENTO DE DADOS

□ Temos os sistemas que...

- mantêm suas informações em memória realizando persistências ocasionais:

*Scalaris, Redis*

- mantêm suas informações em disco:

*CouchDB, MongoDB, Riak, Voldemort*

- são configuráveis:

*BigTable, Cassandra, Hbase, HyperTable*

# MODELO DE DADOS

- Existem quatro categorias principais:
  - Sistemas baseados em armazenamento chave-valor
  - Sistemas orientados a documentos
  - Sistemas orientados à coluna
  - Sistemas baseados em grafos

# MODELO DE DADOS

KEY VALUE

COLUMN

GRAPH

DOCUMENT

- ❑ Coleção de chaves únicas associada a um valor, que pode ser de qualquer tipo (binário,string)

*Exemplo:*

*Key: 1234 Value: “Fernando”*

*Key: 2343 Value: “Name=Fernando,age=29”*

Key	Value
123435	Joao da Silva
334545	Name=Fernando, age=29

# MODELO DE DADOS

COLUMN

GRAPH

DOCUMENT

- Famílias de colunas (um repositório para colunas, análogo a uma tabela do Modelo Relacional) e super-colunas (compostas por arrays de colunas)
- o benefício de armazenar dados em colunas, é a busca/acesso rápido e a agregação de dados.

*Exemplo:*

*Column family, Key, Column name e value*  
*Pessoas ; 4564 ; nome : Ana ; idade : 30;*

example-db (Keyspace)

users (Column Family)

johndoe (key)

name (column)	email (column)	phone (column)
John Doe	<a href="mailto:john@example.com">john@example.com</a>	5558675309

janedoe (key)

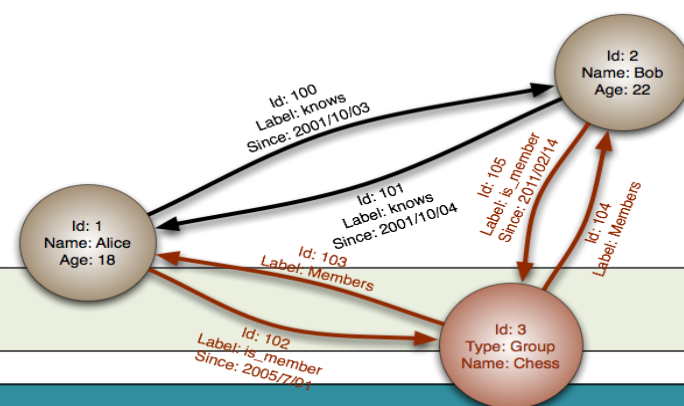
name (column)	email (column)
Jane Doe	<a href="mailto:jane@example.com">jane@example.com</a>



# MODELO DE DADOS

GRAPH

DOCUMENT



- Banco de dados baseado em grafos, nele temos as entidades chamadas de vértices (ou node) que são ligadas entre elas pelas arestas (ou relationships) cada um podendo guardar dados entre os relacionamentos e cada relacionamento pode ter uma direção.

Exemplo:

- *Vértice: Chave->Valor representa entidade. Nome: Alice*
- *Aresta: relacionamentos*  
*Ex: Vértice “Alice” conhece o vértice “Bob” desde 2001*

# MODELO DE DADOS

## DOCUMENT

- Os documentos são as unidades básicas de armazenamento e estes não utilizam necessariamente qualquer tipo de estruturação pré-definida. São baseados em JSON (JavaScript Object Notation);

- Exemplo:

```
{"user":{  
  "id": "123",  
  "name": "Emmanuel",  
  "addresses":[  
    {"city":"Paris"},  
    {"city":"Sao Paulo"}}]}
```

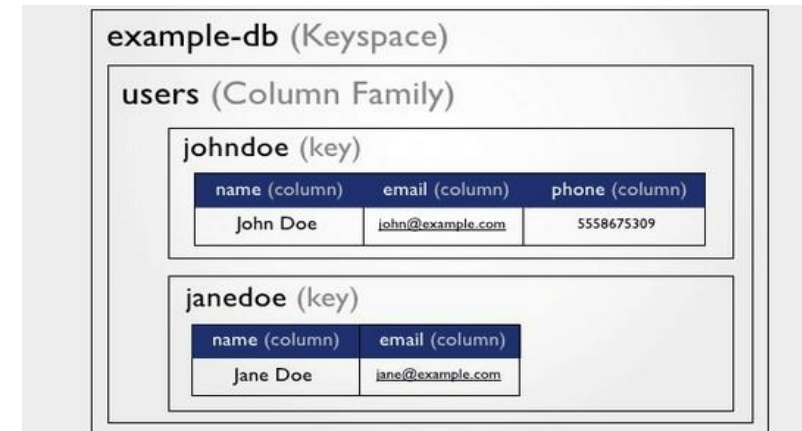


# CLASSIFICAÇÃO NOSQL

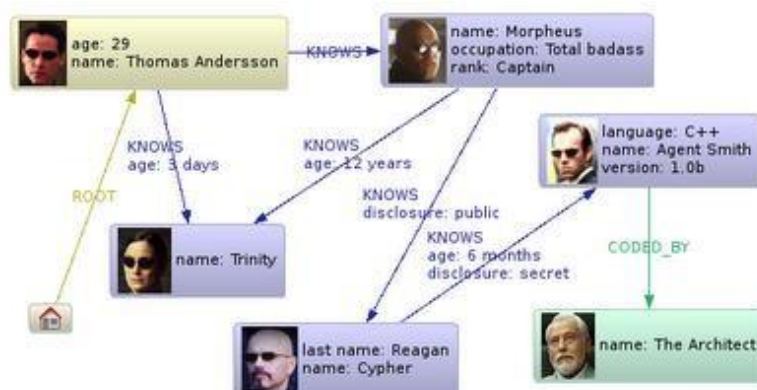
## Key - Value

Key	Value
123435	Joao da Silva
334545	Name=Fernando, age=29

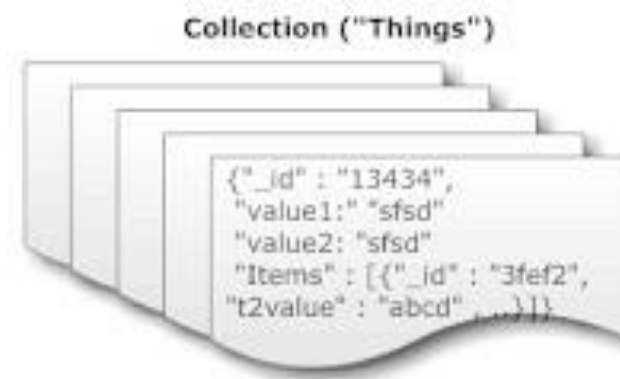
## Column



## Graph



## Document



# DB RANKING

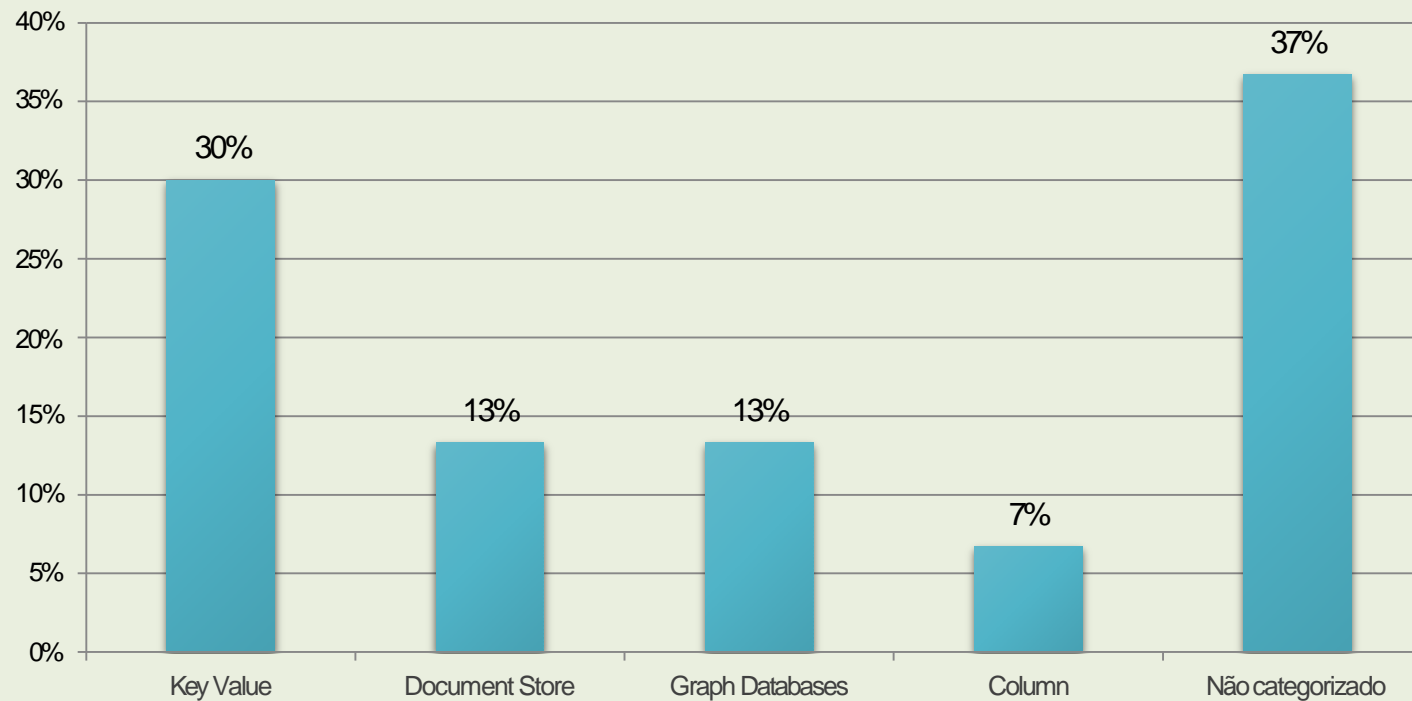
416 systems in ranking, November 2023

Rank			DBMS	Database Model	Score		
Nov 2023	Oct 2023	Nov 2022			Nov 2023	Oct 2023	Nov 2022
1.	1.	1.	Oracle +	Relational, Multi-model i	1277.03	+15.61	+35.34
2.	2.	2.	MySQL +	Relational, Multi-model i	1115.24	-18.07	-90.30
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	911.42	+14.54	-1.09
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	636.86	-1.96	+13.70
5.	5.	5.	MongoDB +	Document, Multi-model i	428.55	-2.87	-49.35
6.	6.	6.	Redis +	Key-value, Multi-model i	160.02	-2.95	-22.03
7.	7.	7.	Elasticsearch	Search engine, Multi-model i	139.62	+2.48	-10.70
8.	8.	8.	IBM Db2	Relational, Multi-model i	136.00	+1.13	-13.56
9.	9.	↑ 10.	SQLite +	Relational	124.58	-0.56	-10.05
10.	10.	↓ 9.	Microsoft Access	Relational	124.49	+0.18	-10.53

<http://db-engines.com/en/ranking>







# CLASSIFICAÇÃO DE SGBDS NOSQL



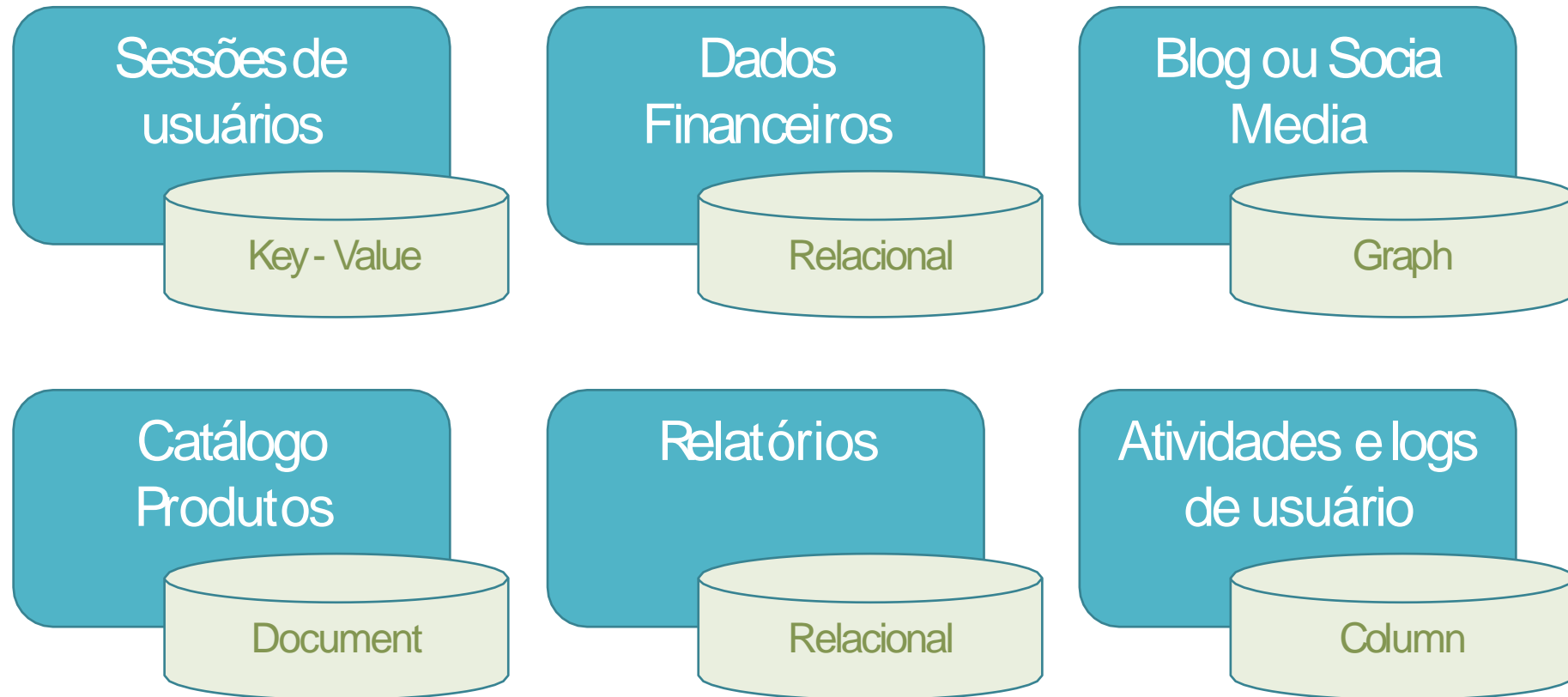
150 tipos de banco de dados NOSQL

- Fonte: <http://nosql-database.org/>
- Dados compilados manualmente

# CLASSIFICAÇÃO DE SGBD NOS SEUS PRODUTOS

KEY VALUE	COLUMN	GRAPH	DOCUMENT
			
<p>Amazon DynamoDB (Key-value) Desenvolvido em: Java Quem Usa: -Washingtonpost.com -Elsevier (Editora)</p>	<p>BigTable(column) Google Desenvolvido em: C++ Quem Usa: Gmail Google Maps, YouTube</p> <p>Cassandra (column) Desenvolvido em: Java Quem Usa? Twitter NetFlix Facebook</p>	<p>Neo4j (graph) Desenvolvido em: Java Quem Usa: -WalMart -National Geographic -Ebay</p>	<p>MongoDB (Document) Desenvolvido em: C Quem Usa: -Globo.com -Apontador -Forbes -New York Times</p>

# QUANDO E QUAL UTILIZAR?



Fonte: <http://www.martinfowler.com/bliki/PolyglotPersistence.html>

**DÁ PRA FAZER QUERY?**

**I**  **SQL**



# DÁ PRA FAZER QUERY?

## Amazon DynamoDB—Key-Value

### SQLQuery

```
SELECT _id, name, address  
FROM users  
WHERE age > 18  
LIMIT 5
```

← projection  
← table  
← select criteria  
← cursor modifier

### AWSQuery

Amazon DynamoDB Explore Table: Reply

List Tables Browse Items

☐ Scan ☒ Query Go New Item Edit Item Copy to New Delete Item

Index Name: [Table] Reply: Id, ReplyDateTime

Hash Key (\*): Id equal to Amazon DynamoDB#DynamoDB Thre

Range Key: ReplyDateTime greater than 2014-01-12

Sort: ☒ Ascending ☐ Descending

# DÁ PRA FAZER QUERY?

## MongoDB - Document

### SQLQuery

```
SELECT _id, name, address ← projection
FROM   users              ← table
WHERE  age > 18            ← select criteria
LIMIT  5                  ← cursor modifier
```

### Operation Find

```
db.users.find(              ← collection
  { age: { $gt: 18 } },     ← query criteria
  { name: 1, address: 1 }   ← projection
).limit(5)                  ← cursor modifier
```

# DÁ PRA FAZER QUERY?

## Neo4j- Graph

### SQLQuery

```
SELECT _id, name, address  
FROM users  
WHERE age > 18  
LIMIT 5
```

← projection  
← table  
← select criteria  
← cursor modifier

### Cyber query

```
MATCH a  
WHERE a.age>18  
RETURN a.id, a.name, a.address  
LIMIT 5
```

# DÁ PRA FAZER QUERY?

## Cassandra - Column

### SQL Query

```
SELECT _id, name, address ← projection
FROM   users               ← table
WHERE  age > 18             ← select criteria
LIMIT  5                   ← cursor modifier
```

### CQL – Cassandra Query Language

```
SELECT _id, name, address ← projection
FROM   users               ← table
WHERE  age > 18             ← select criteria
LIMIT  5                   ← cursor modifier
```

Comandos CRUD  
(Create, Read, Update, Delete)  
são iguais

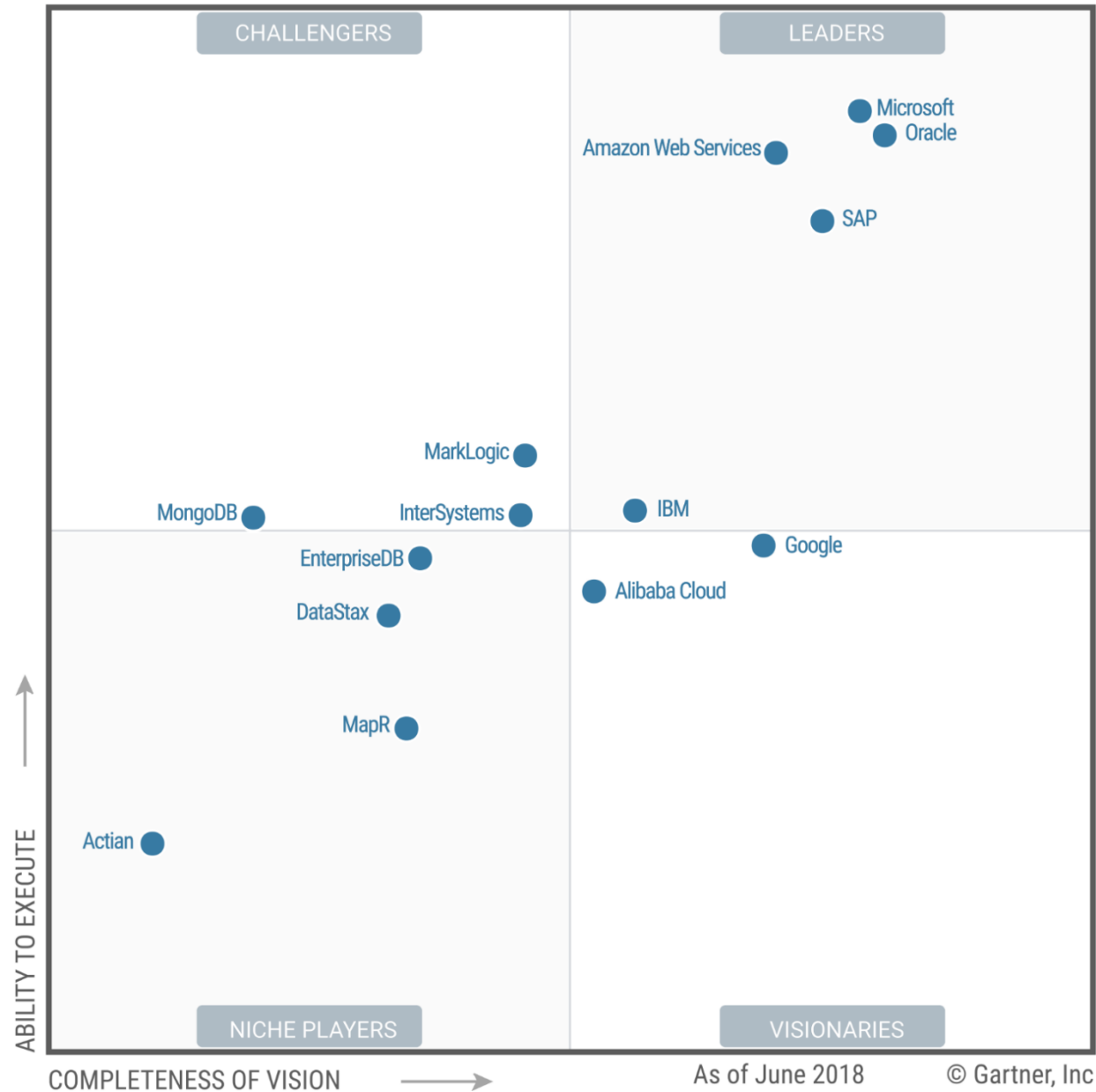
# QUAIS LINGUAGENS SUPORTAM NOSQL?

	Amazon Dynamo	Neo4j	Cassandra	MongoDB
C				X
C#				X
C++			X	X
Go			X	X
Java	X	X	X	X
Javascript	X			X
Node.js	X	X	X	X
Perl			X	X
PHP	X	X	X	X
Python		X	X	X
Ruby	X	X	X	X
Scala		X	X	X

# OS BANCO DE DADOS RELACIONAIS IRÃO MORRER?



# POSIÇÃO NOSQL – GARTNER MAGIC QUADRANT



# CASES

The Netflix logo, consisting of the word "NETFLIX" in white, bold, sans-serif capital letters with a 3D effect, set against a solid red rectangular background.

## **SGBD:**

sistema de processamento de faturas mensais

## **NOSQL:**

Sistema focado em recomendações de melhores filmes.





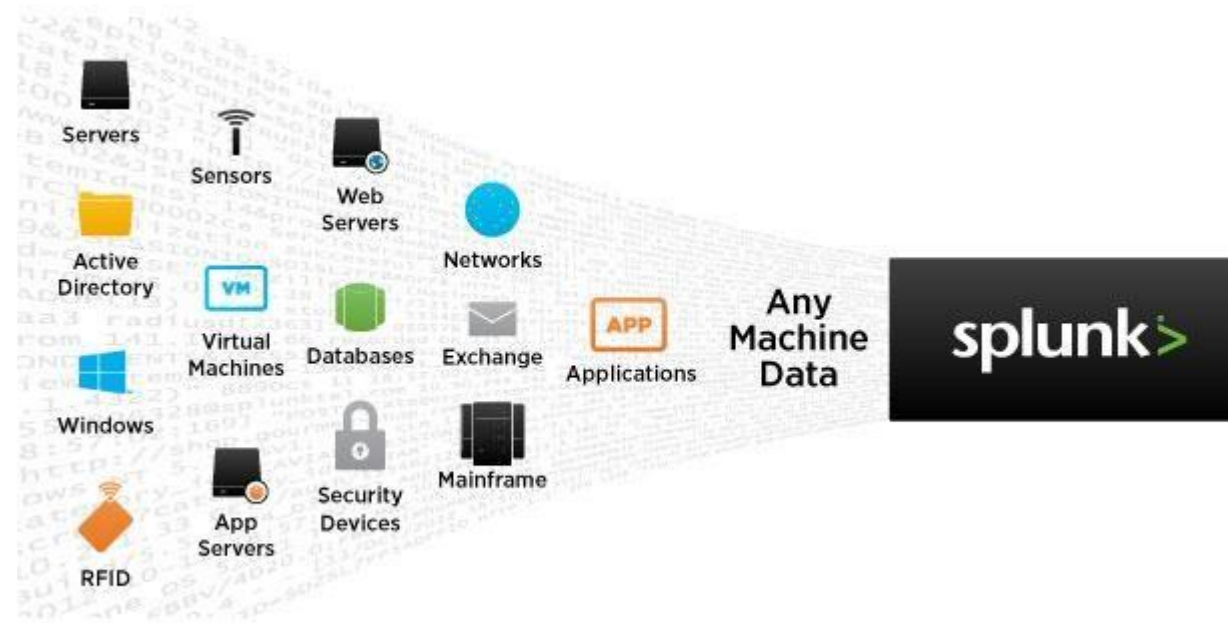
## **SGBD:**

**Sistemas de processamento de ordem de venda**

## **NOSQL:**

**Sistema de pesquisa, recomendações e adaptações de preços em tempo real**

# CASES



*Plataforma para inteligência operacional*

**SGBD:**

Dados de clientes, produtos e RH

**NOSQL:**

Explorar, analisar e virtualização de dados

# **ATIVIDADE PRÁTICA: EQUIPES**

- **Selecionar um banco de dados da categoria NoSQL;**
- **2 minutos para apresentar;**
- **Para este banco de dados, pesquisar:**
  - **Características do banco;**
  - **Compatibilidade com linguagens de programação, ambientes de execução;**
  - **Quem utiliza?**