

Bancos de Dados NO-SQL



PROF. ANTONIO GUARDADO

Agenda

- 1 – Conceitos e definições
- 2 – Características
- 3 – Modelos de dados
- 4 – Cases
- 5 – Oportunidades no Mercado

1- O que significa?

- É uma denominação para bancos de dados **não-relacionais**.
- Isso não quer dizer que seus modelos não possuem relacionamentos.
- E sim, que não são orientados a tabelas.
- Not Only SQL. (**Não Apenas SQL**)
- A tecnologia NoSQL foi iniciada por companhias líderes da Internet : Google, Facebook , Amazon e LinkedIn - para superar as limitações (45 anos de uso da tecnologia) de banco de dados relacional para aplicações web modernas (2009)

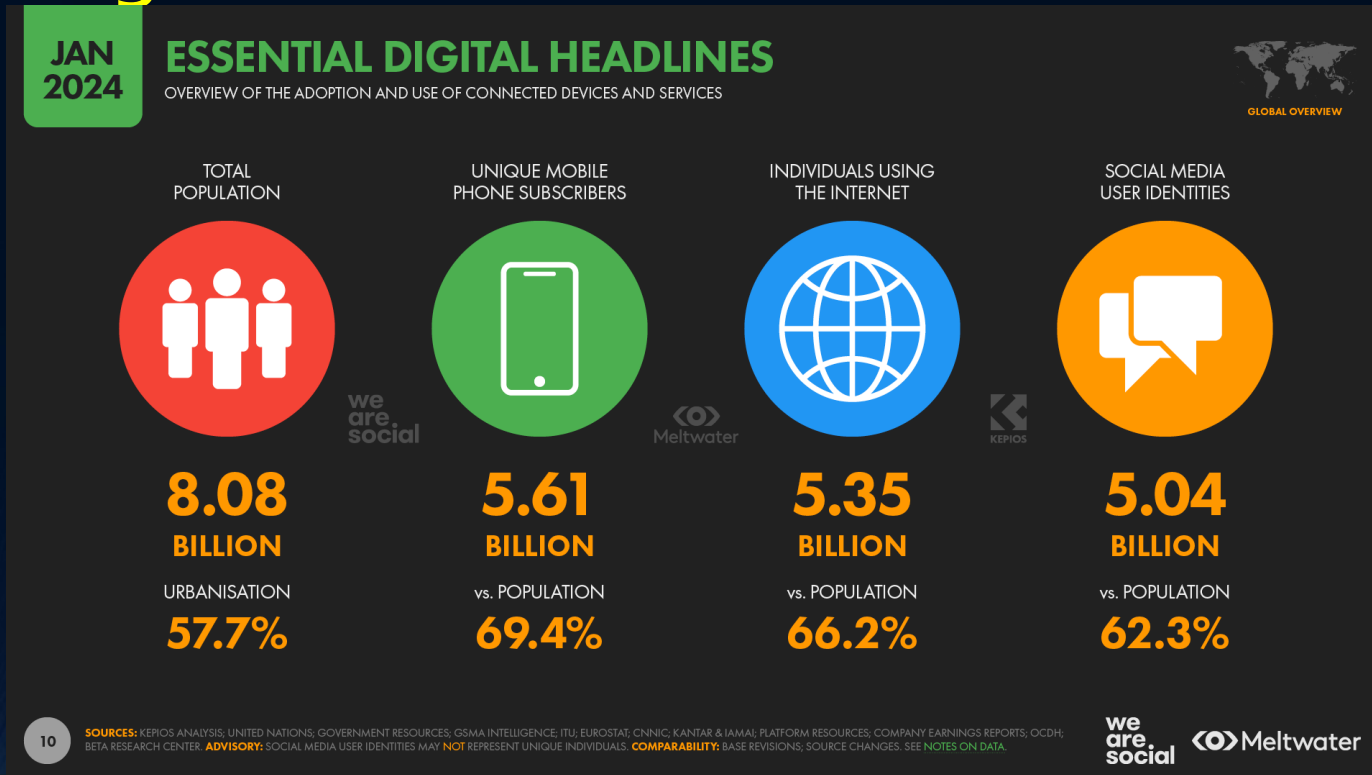
1.1 - Banco de Dados Relacional

- ❑ Dados são estruturados de acordo com o modelo relacional
- ❑ Padrão para a grande maioria dos SGBDs
SQL Server, Oracle, PostgreSQL, MySQL, DB2, etc.
- ❑ Elementos básicos
Relações (tabelas) e registros (tuplas)
- ❑ Características fundamentais
Restrições de integridade (PK, FK, UK, CK, NN)
Normalização
Linguagem SQL (*Structured Query Language*)

1.2 - Por que NoSQL ?

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

1.2.1 - Big Users



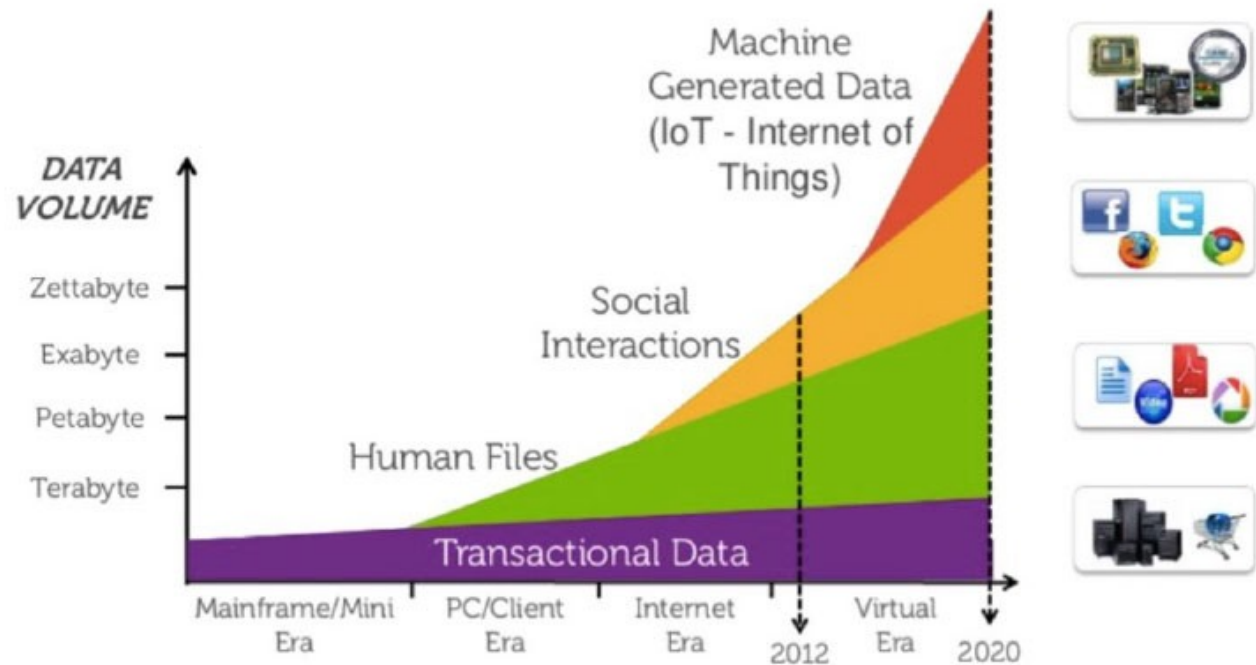
O crescente uso de aplicativos online resultou em um número crescente de operações de banco de dados e uma necessidade de uma maneira mais fácil de **escalar** bancos de dados para atender a essas demandas.

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.

1.2.2 - 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.

The Explosion of Data

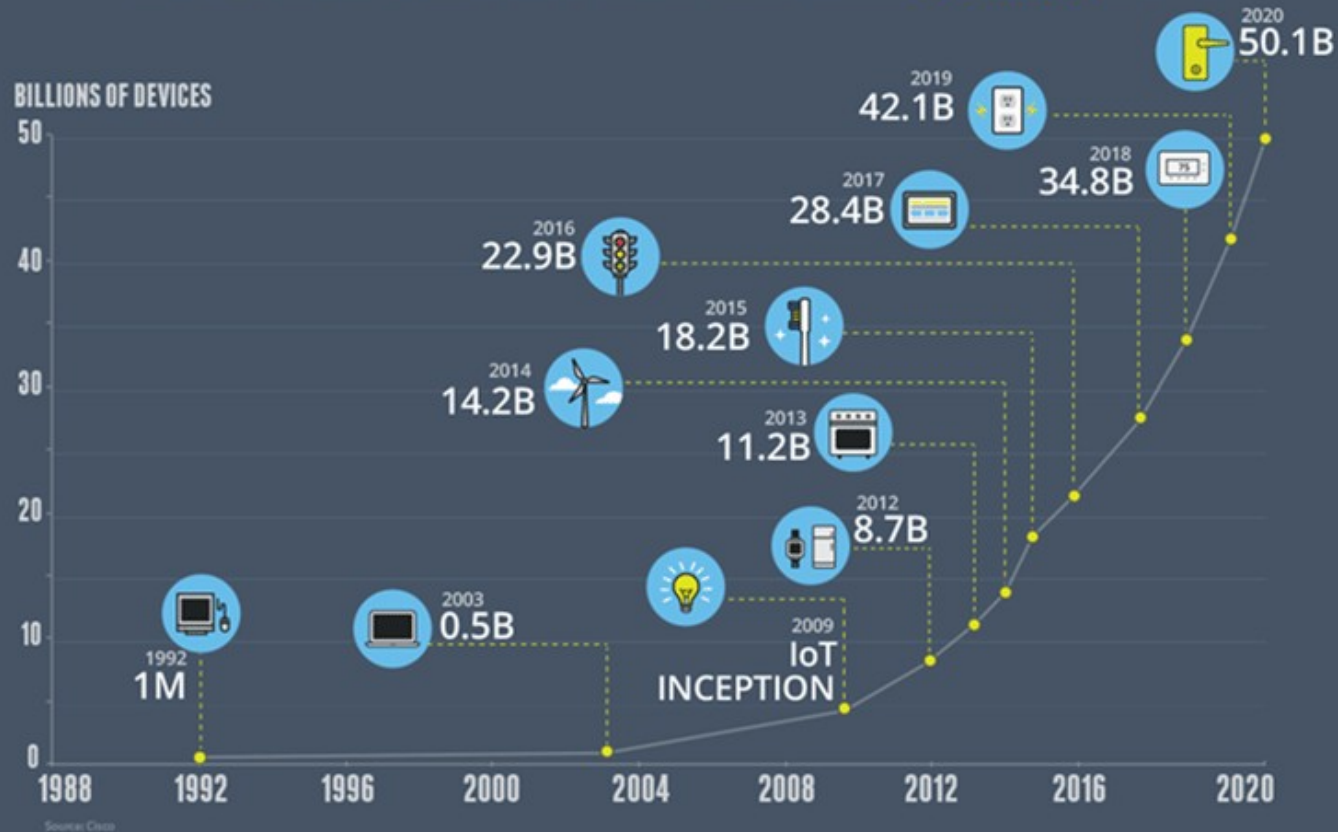


NoSQL fornece um modelo de dados **sem esquema** muito mais flexível que mapeia melhor a organização de dados de uma aplicação e simplifica a interação entre a aplicação e o banco de dados, resultando em menos código para escrever, depurar e manter.

1.2.3 - A Internet das Coisas

GROWTH IN THE INTERNET OF THINGS

THE NUMBER OF CONNECTED DEVICES WILL EXCEED **50 BILLION** BY 2020



1.2.3 - A Internet das Coisas

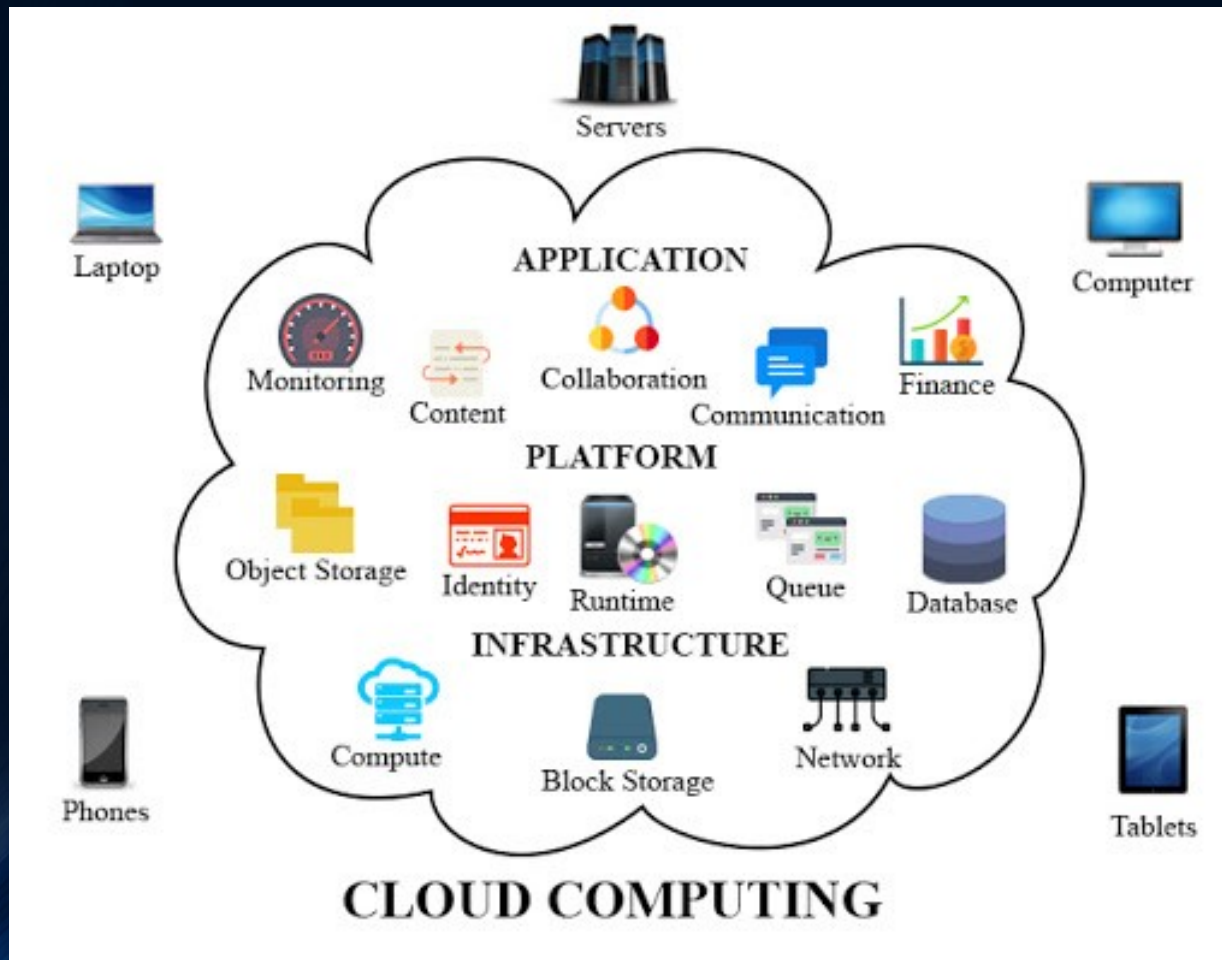


Dados de telemetria - semi- estruturados e contínuos - representam um desafio para bancos de dados relacionais, que exigem um esquema fixo e dados estruturados.

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.

1.2.4 - 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 ou mais camadas.



1.2.4 - Cloud Computing

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 ou invés de horizontal.

Isso não os torna adequado 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 a natureza altamente distribuída da arquitetura três camadas da internet.

2 - Principais Características

- Escalabilidade Horizontal.
- Ausência de Esquema ou Esquema Flexível.
- Suporte a Replicação.
- API Simples
- Nem Sempre é Consistente.
- Alta disponibilidade (Confiabilidade, recuperabilidade, detecção rápida de erros e operações contínuas)
- Modos de armazenamento

2.1 - Escalabilidade Horizontal

- A escalabilidade Horizontal consiste em aumentar o número de máquinas disponíveis.
- A escalabilidade Horizontal em modelos relacionais seria inviável devido a concorrência.
- Como nos modelos NoSQL não existe bloqueios, esse tipo de escalabilidade é a mais viável.

2.1 – Escalabilidade Vertical x Horizontal

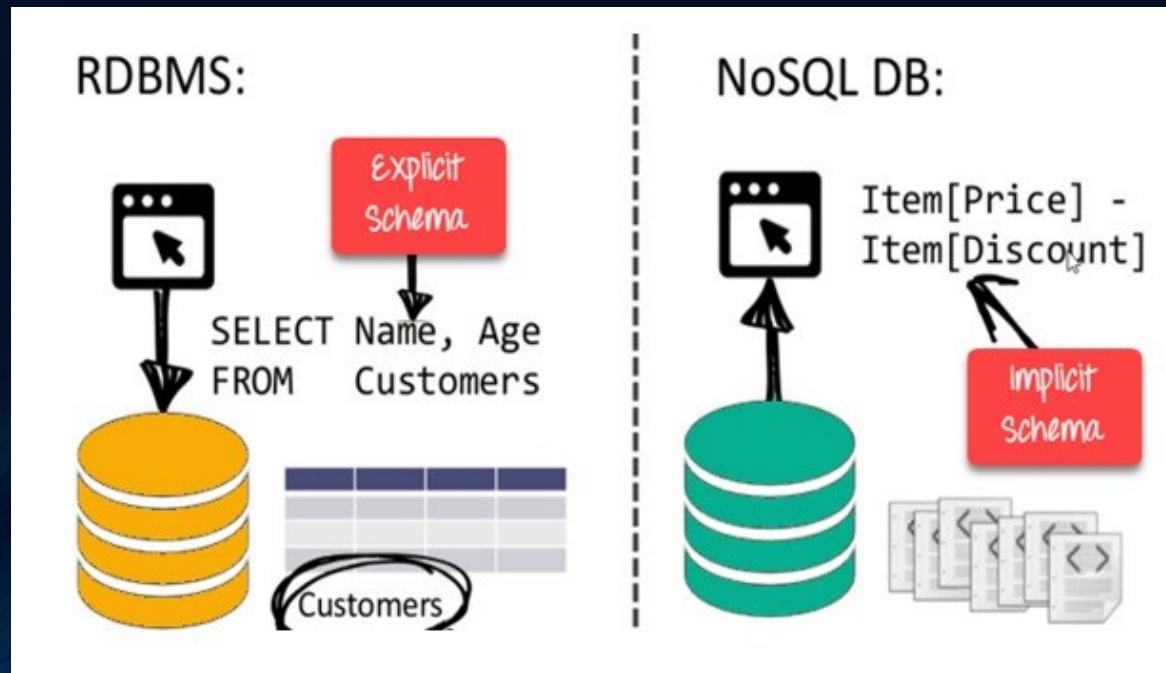


- Vertical : aumentar a capacidade do hardware -> + memória / + HD / + processador

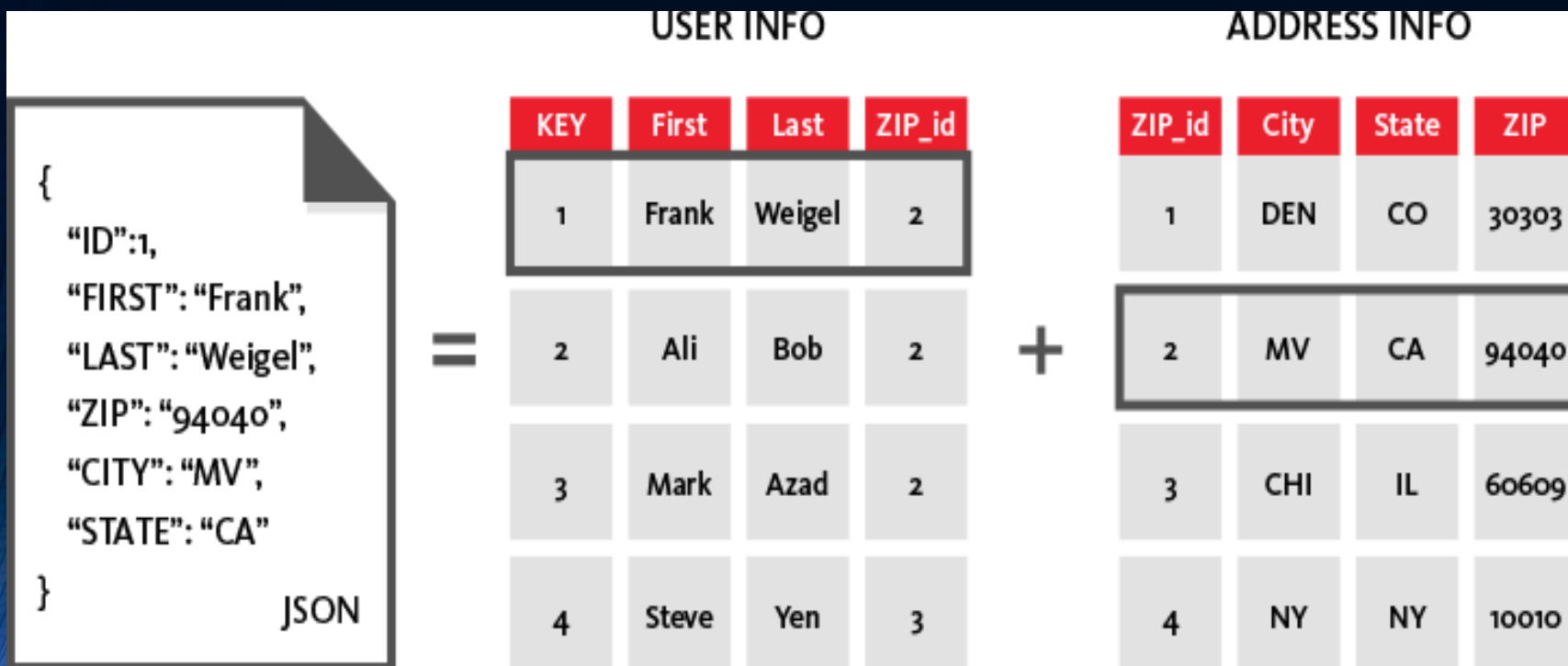
Horizontal : mais máquinas com processamento distribuído

2.2 - Ausência de Esquema (Schema Free)

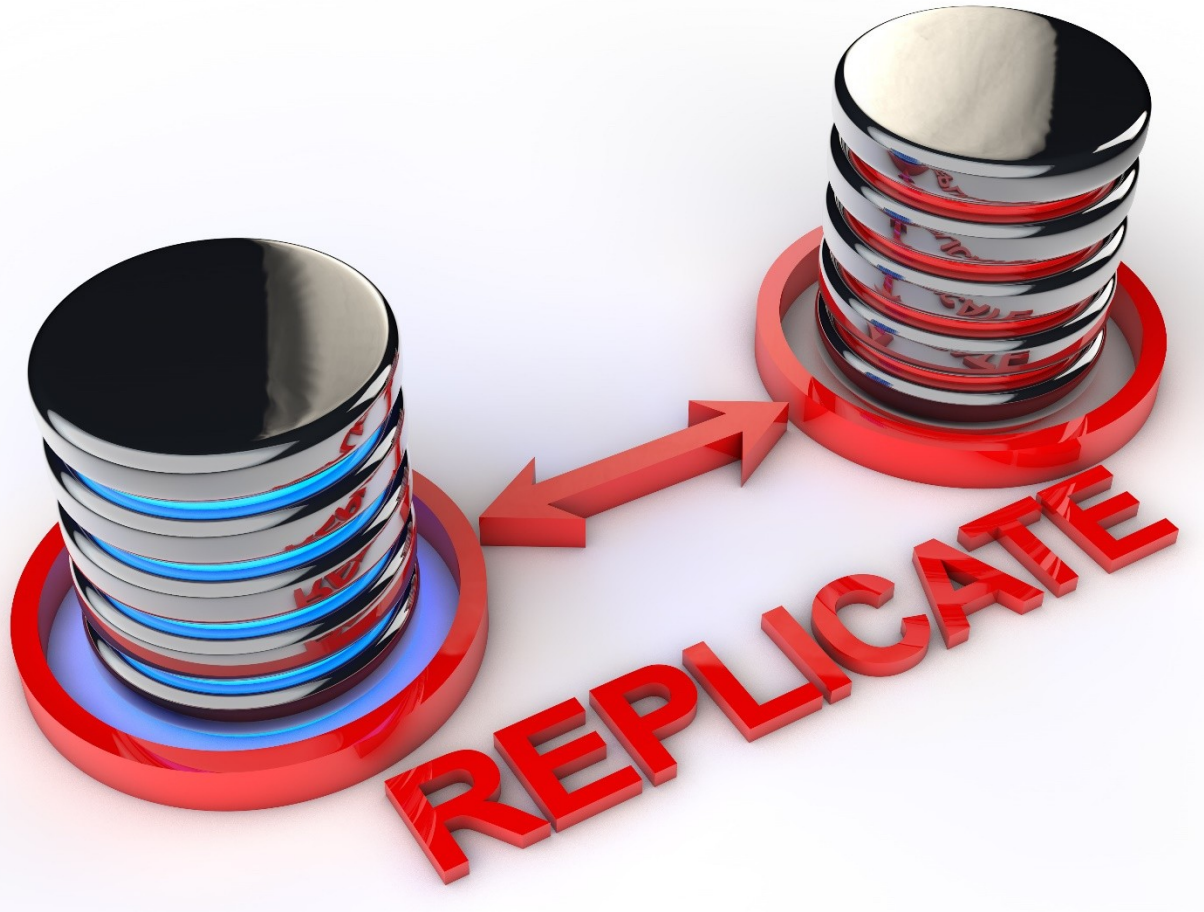
- Apresentam ausência de Esquema ou esquema flexível, isso permite uma fácil aplicação da escalabilidade e também um aumento na disponibilidade dos dados
- Mas também devido a essa ausência, não há garantia da integridade dos dados.



2.2 - Modelo de dados mais flexível



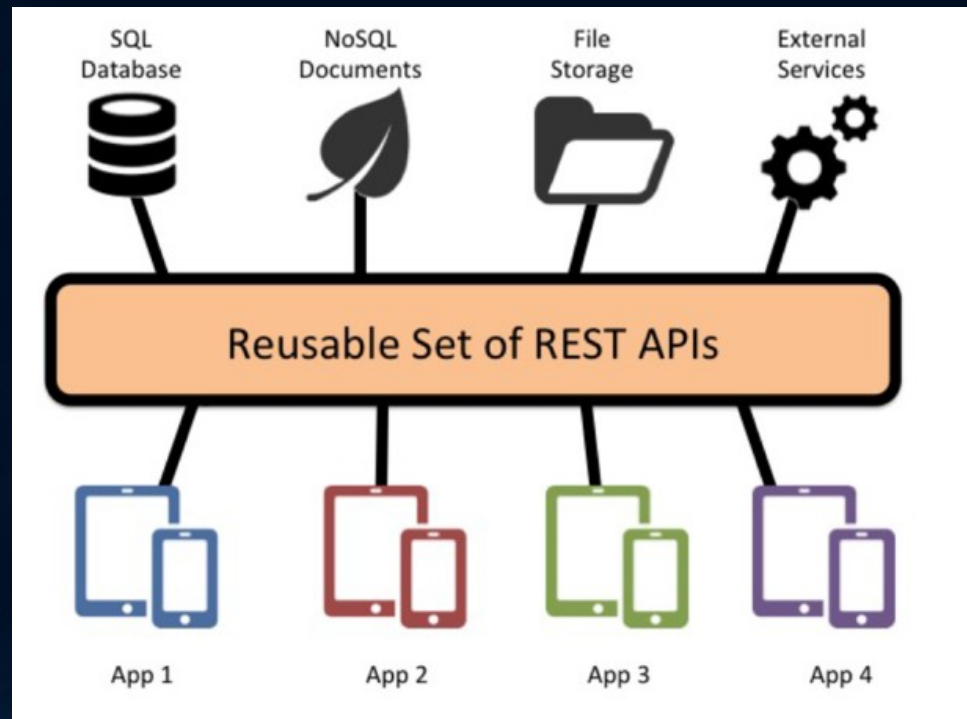
2.3 - Suporte a replicação



- Permitem a replicação de uma forma nativa o que provém uma escalabilidade maior e também uma diminuição do tempo gasto para a recuperação de informações

2.4 – APIs simples

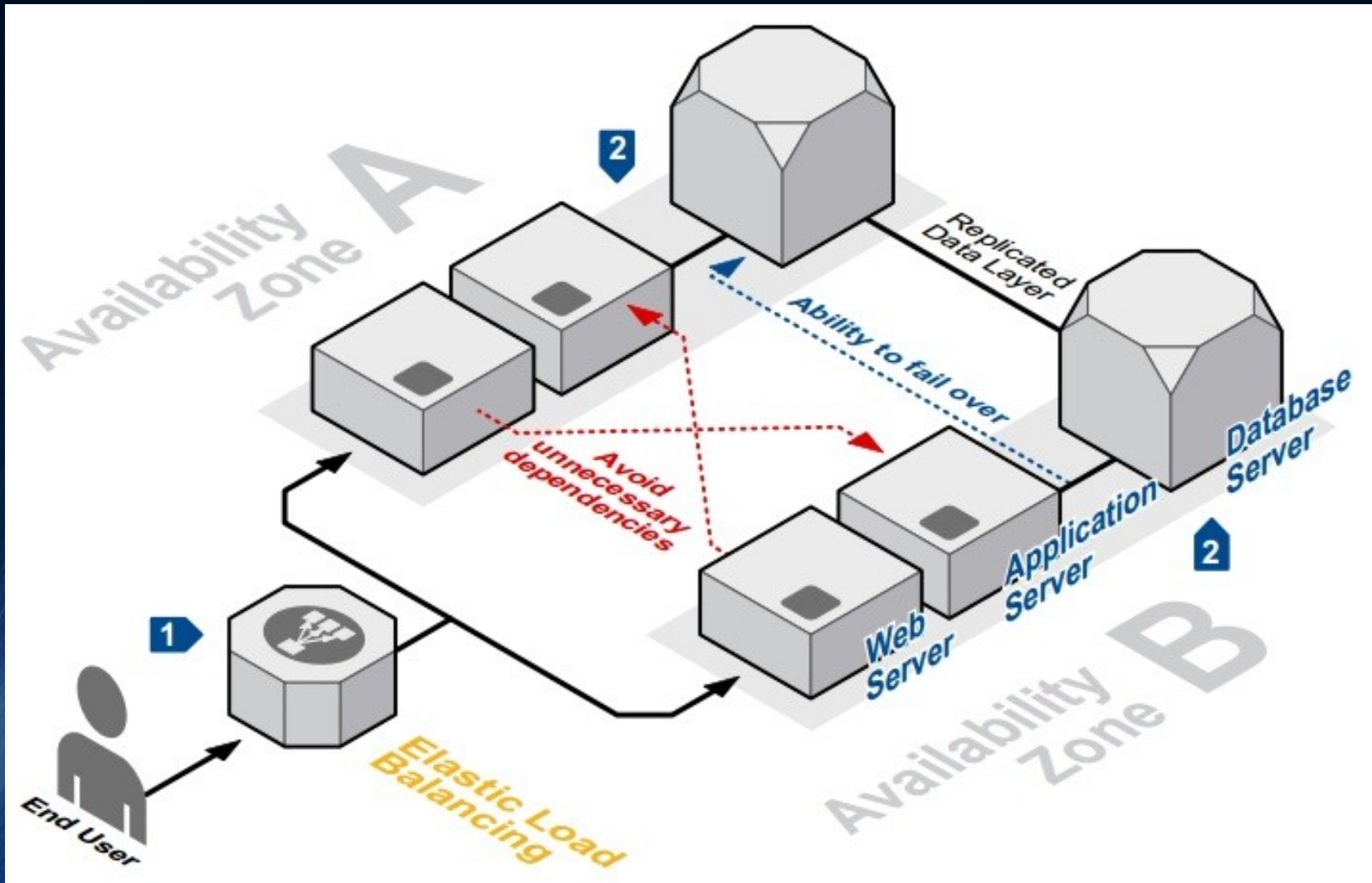
- Para que o acesso às informações seja feito da forma mais rápida possível APIs são desenvolvidas para que qualquer aplicação possa ter acesso aos dados do SGBD.



2.5 - Nem sempre consistente

- Os bancos de dados NoSQL nem sempre conseguem se manter consistentes, ou seja, o valor do dado lido nem sempre é o mais atual.

2.6 – Alta disponibilidade



2.7 - Modo 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

3 - Modelo de Dados

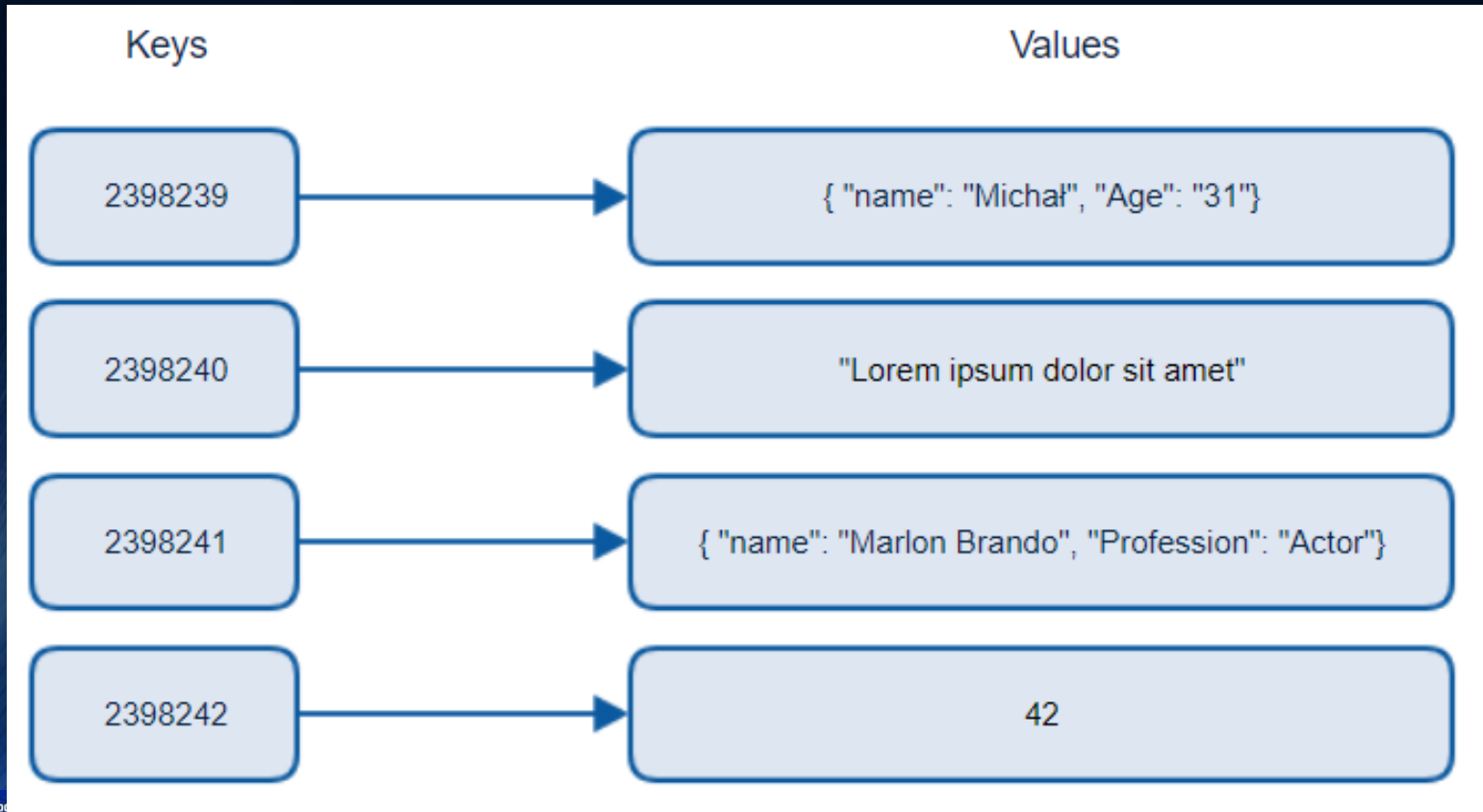
❑ Existem quatro categorias:

- Sistemas baseados em armazenamento chave-valor
- Sistemas orientados a documentos
- Sistemas orientados à coluna
- Sistemas baseados em grafos

3.1 - Modelo de Dados – Chave-Valor

KEY VALUE COLUMN GRAPH DOCUMENT

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



3.2 - Modelo de Dados – Orientado a Colunas

KEY VALUE 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.

3.2 - Modelo de Dados – Orientado a Colunas

COLUMN FAMILIES

Row Key	personal		professional	
employeeid	name	city	designation	salary
1000	Raju Karappan	St.Auguistine	Sr. Technical Architect	125000
2000	Ravi Shankar	Jacksonville	Chief Technology Officer	180000
3000	Srikiran Gurugubeli	Jacksonville	Sr. Developer	100000

3.3 - Modelo de Dados - Grafos

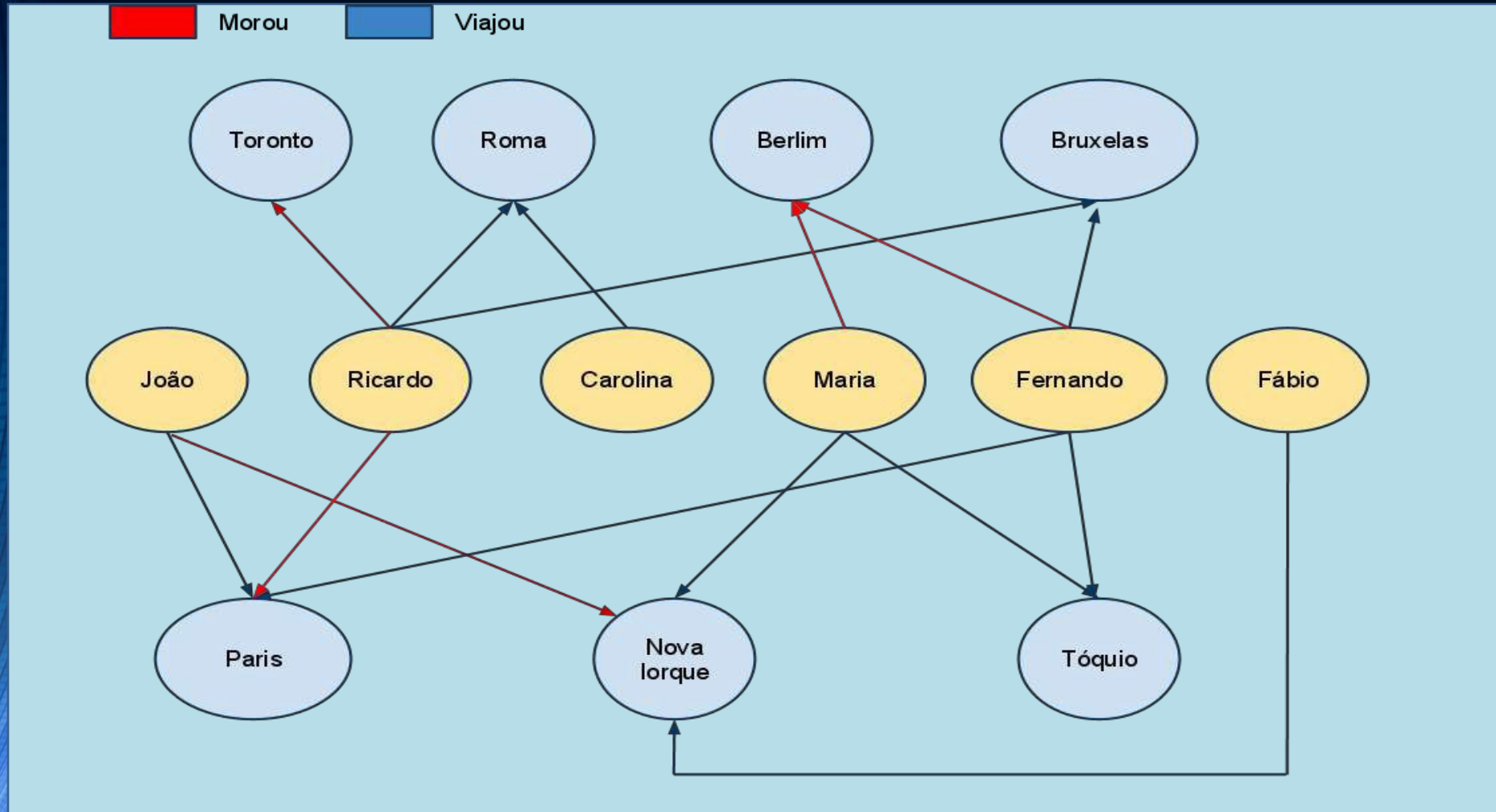
KEY VALUE COLUMN

GRAPH

DOCUMENT

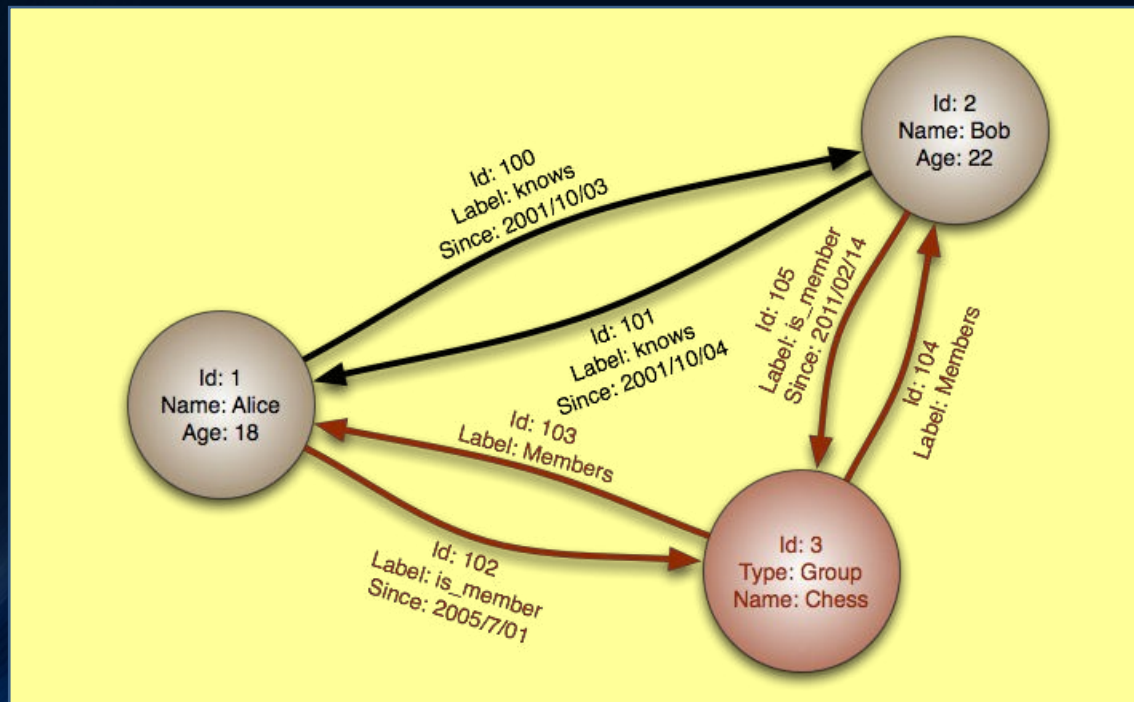
- ❑ Banco de dados baseado em grafos, nele temos as entidades chamadas de vértices (ou node) que são ligadas entre elas pelas arestas (ou relationships) cada um podendo guardar dados entre os relacionamentos e cada relacionamento pode ter uma direção.

3.3 - Modelo de Dados - Grafos



3.3 - Modelo de Dados - Grafos

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



3.4 - Modelo de Dados - Documentos

KEY VALUE COLUMN GRAPH **DOCUMENT**

- ❑ Os documentos são as unidades básicas de armazenamento e estes não utilizam 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"}}]}
```

3.4 - Modelo de Dados - Documentos

KEY VALUE COLUMN GRAPH **DOCUMENT**

Document 1

```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

3.5 - Classificação NoSQL

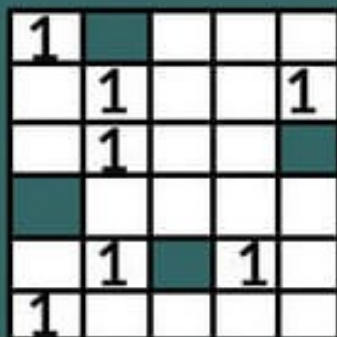
Key Value



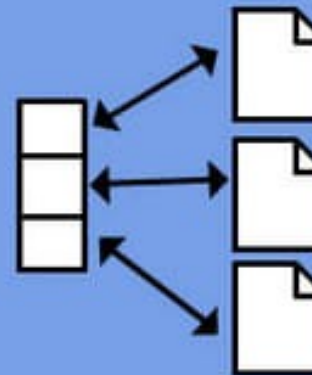
Graph DB



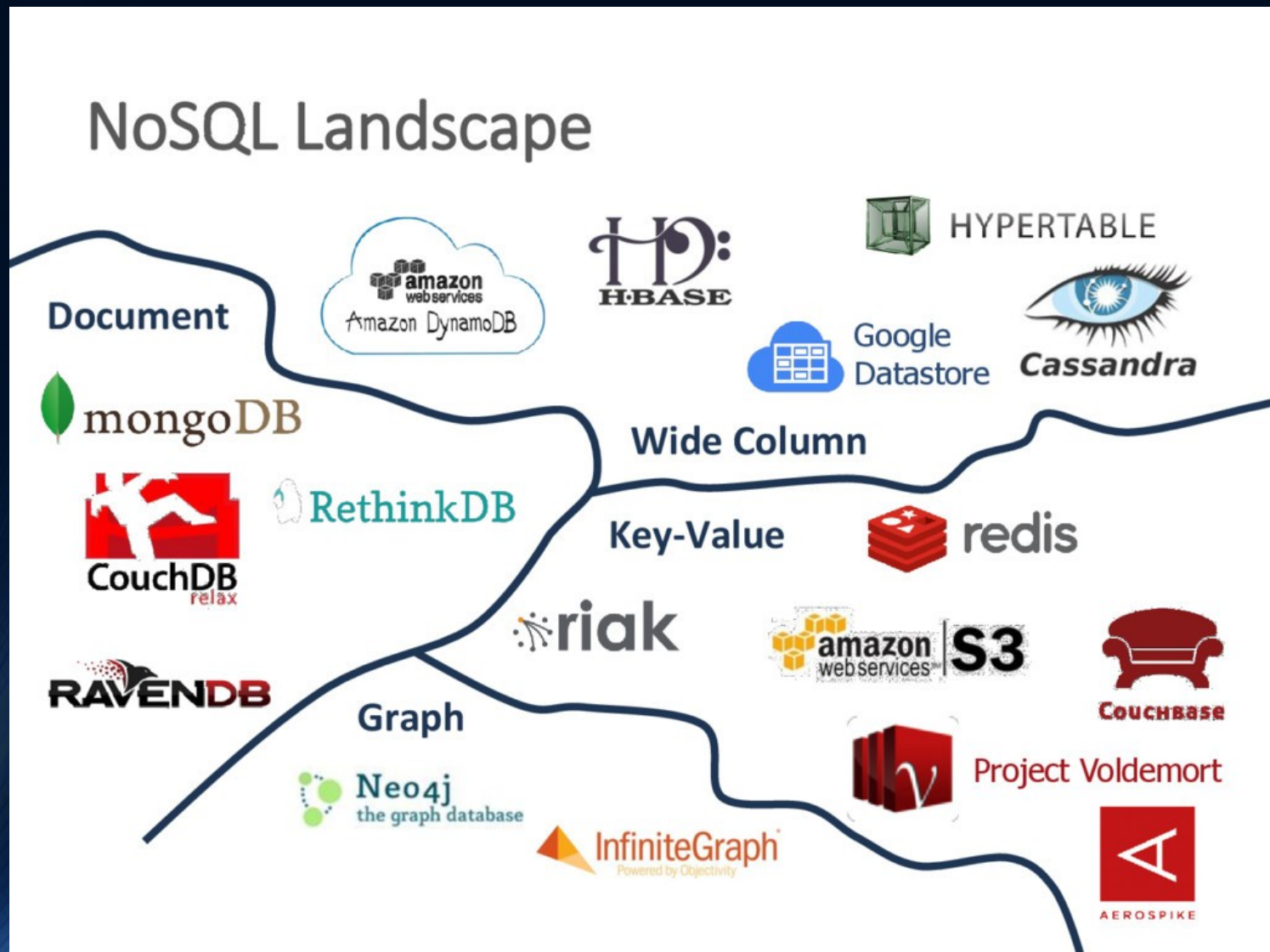
Column Family



Document



3.5 - Classificação NoSQL



3.6–Consultas em BDs NoSQL

Amazon DynamoDB– Key-Value

SQL Query

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

← projection
← table
← select criteria
← cursor modifier

AWS
Query

Amazon DynamoDB Explore Table: Reply

☐ Scan ☒ Query

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

3.6 – Consultas em BDs NoSQL

MongoDB - Document

SQL Query

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

← projection
← table
← select criteria
← cursor modifier

Operation
Find

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

3.6–Consultas em BDs NoSQL

Neo4j- Graph

SQL Query

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

Cypher query

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

3.6 – Consultas em BDs NoSQL

Cassandra - Column

SQL Query

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

← projection
← table
← select criteria
← cursor modifier

Comandos CRUD

(Create, Read, Update,
Delete)

CQL – Cassandra Query
Language

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


























← projection
← table
← select criteria
← cursor modifier

3.7 - Quais linguagens suportam

	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

3.8 - DB Ranking

423 systems in ranking, August 2024

Rank			DBMS	Database Model	Score		
Aug 2024	Jul 2024	Aug 2023			Aug 2024	Jul 2024	Aug 2023
1.	1.	1.	Oracle 	Relational, Multi-model 	1258.48	+18.12	+16.39
2.	2.	2.	MySQL 	Relational, Multi-model 	1026.86	-12.60	-103.59
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	815.18	+7.52	-105.64
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	637.39	-1.52	+17.01
5.	5.	5.	MongoDB 	Document, Multi-model 	420.98	-8.85	-13.51
6.	6.	6.	Redis 	Key-value, Multi-model 	152.71	-4.06	-10.26
7.	7.	 11.	Snowflake 	Relational	135.97	-0.56	+15.34
8.	8.	 7.	Elasticsearch	Search engine, Multi-model 	129.83	-0.99	-10.09
9.	9.	 8.	IBM Db2	Relational, Multi-model 	123.00	-1.40	-16.23
10.	10.	10.	SQLite 	Relational	104.79	-5.16	-25.13
11.	 12.	 12.	Apache Cassandra 	Wide column, Multi-model 	97.00	-2.12	-10.38
12.	 11.	 9.	Microsoft Access	Relational	96.37	-4.26	-33.97

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

3.9 - Classificação de SGBDs NoSQL



150 tipos de banco de dados NOSQL

❑ Fonte: <http://nosql-database.org/>

❑ Dados compilados manualmente

3.9.1- Classificação NoSQL e Produtos

KEY VALUE

COLUMN

GRAPH

Amazon
DynamoDB (Beta)

ORACLE 11g
BERKELEY DB

redis



Cassandra

Google Bigtable

Neo4j
the graph database

InfiniteGraph

sones



CouchDB
relax

mongoDB

Amazon
DynamoDB (Key-value)
Desenvolvido em: Java
Quem Usa?
-Washingtonpost.com
-Elsevier (Editora)

BigTable(column)
Google Desenvolvido
em: C++ Quem Usa:
Gmail
Google Maps, YouTube

Cassandra (column)
Desenvolvido em:
Java Quem Usa?
Twitter Netflix
Facebook

Neo4j (graph)
Desenvolvido em: Java
Quem Usa?
-WalMart
-National Geographic
-Ebay

MongoDB (Document)
Desenvolvido em: C Quem
Usa:
-Globo.com
_Apontador
-Forbes
-New York Times

3.10 - Quando e qual utilizar?



Fonte: Martin Fowler

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

4 - 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.

4 - Cases



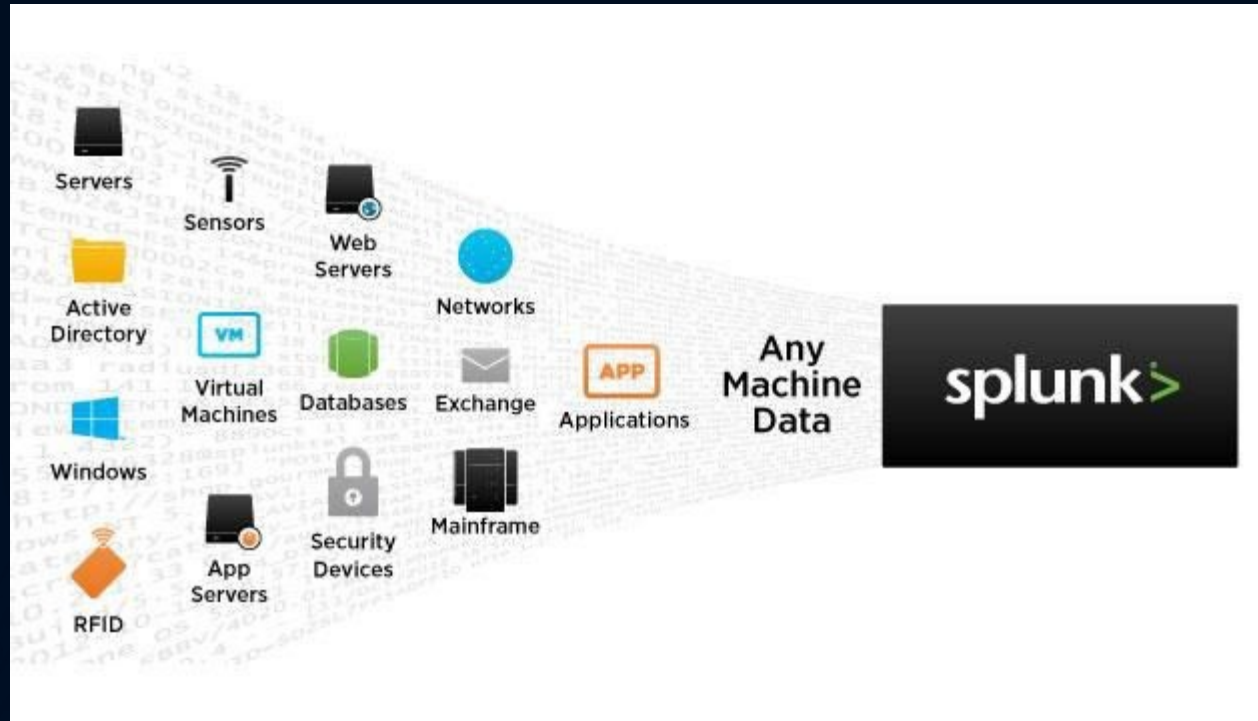
SGBD:

Sistemas de processamento de ordem de venda

NOSQL:

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

4 - Cases



Plataforma para inteligência operacional

SGBD:

Dados de clientes, produtos e RH

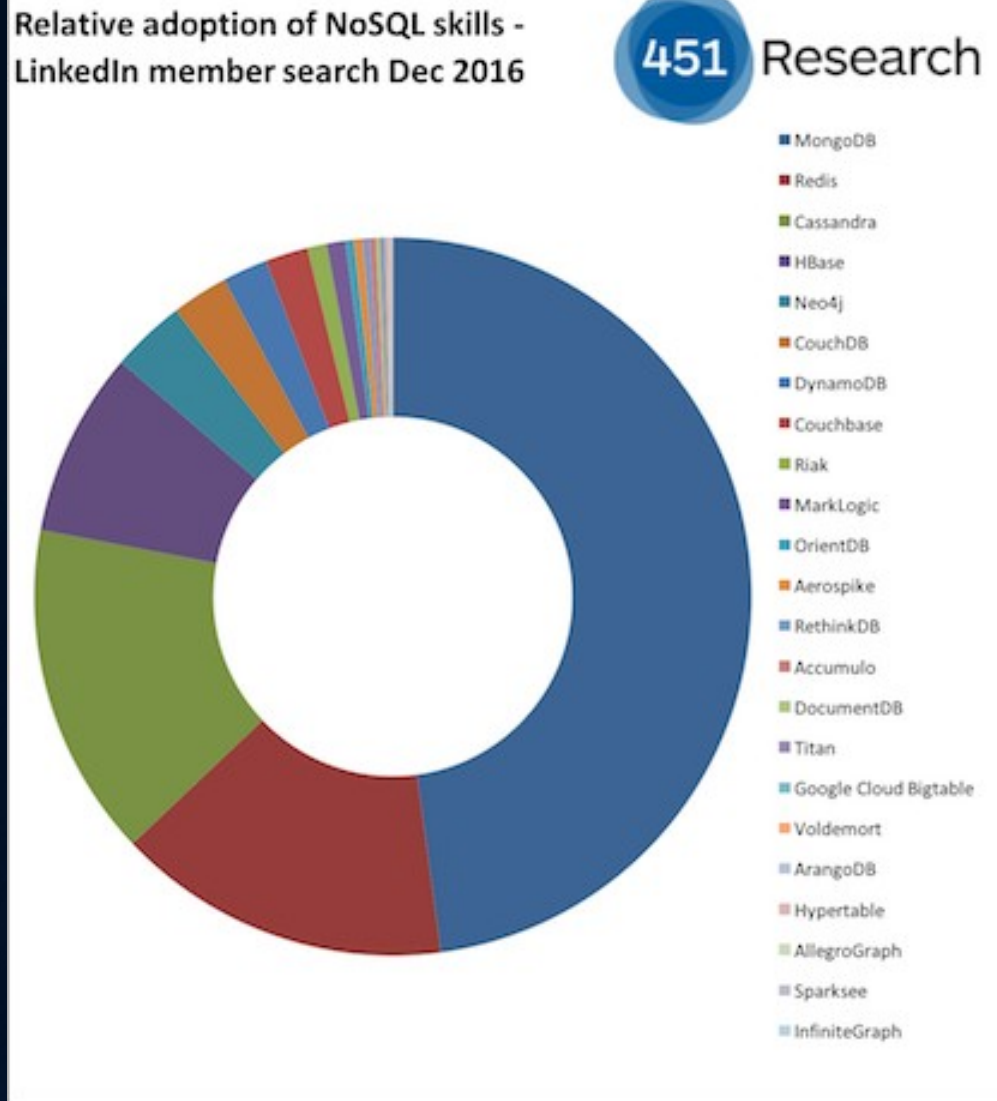
NOSQL:

Explorar, analisar e virtualização de dados

5 - Oportunidades no mercado



5.1 - Profissionais no mercado



https://blogs.the451group.com/information_management/?s=NoSQL+LinkedIn+Skills