

Not
Only **SQL**

NO **SQL** *Uma Breve Introdução*

Andréa Bordin

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.
 - 2009

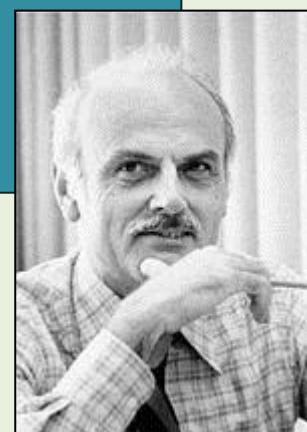
Banco de Dados Relacional

- 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*)

Edgar F. Codd

*August 23, 1923 +April 18, 2003

Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377–387. doi:10.1145/362384.362685.



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

Big Users



3

Billion
Global Online
Population



35

Billion Hrs./Mo.
Spent Online



1.75

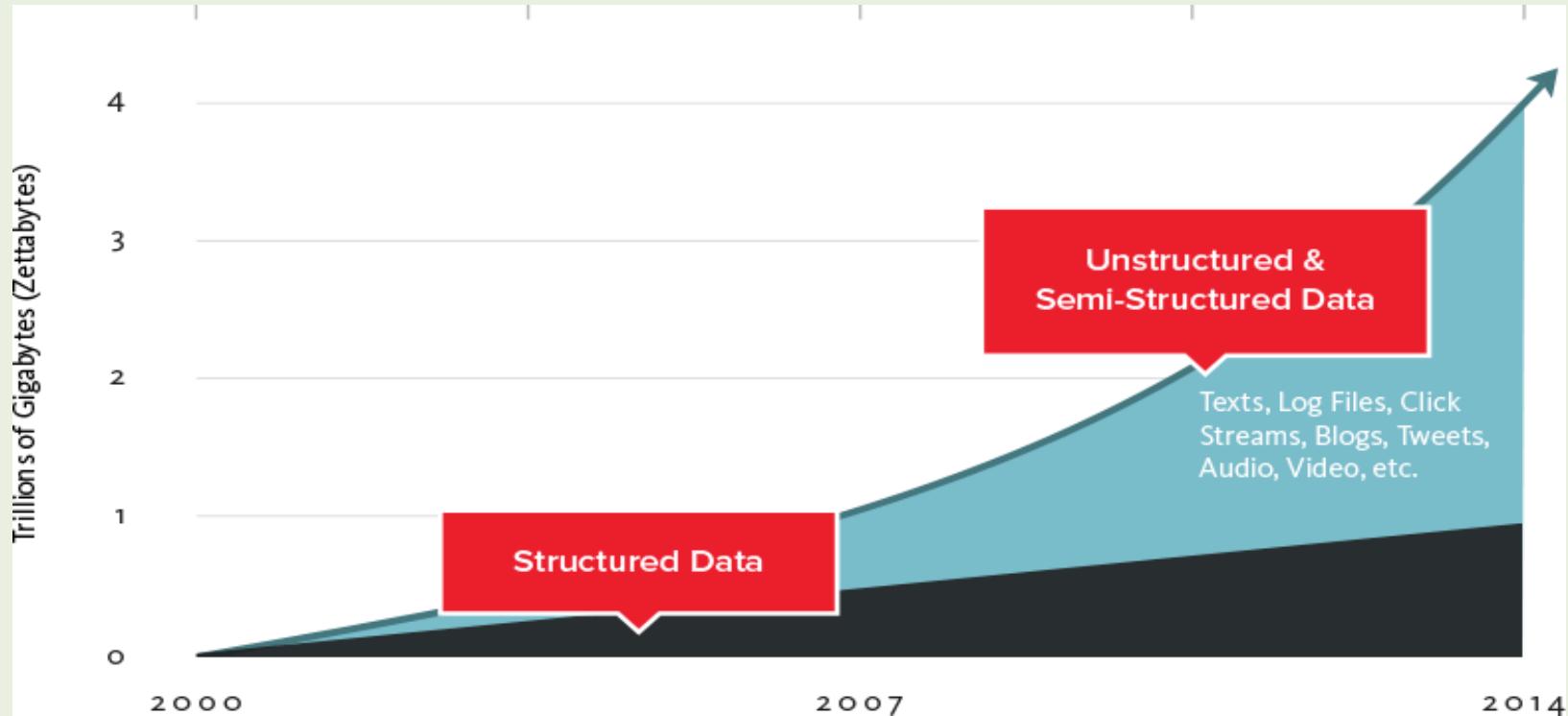
Billion
Smartphone
Users

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**.

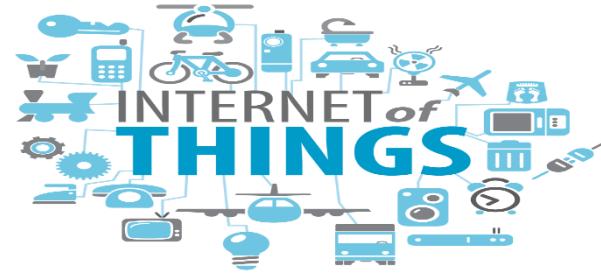
NoSQL é a solução.

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 (vs 2% hoje)

21% dos mais valiosos dados serão gerados por sistemas embarcados (vs 8% hoje)

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 escabilidade é vertical ou invés de horizontal.

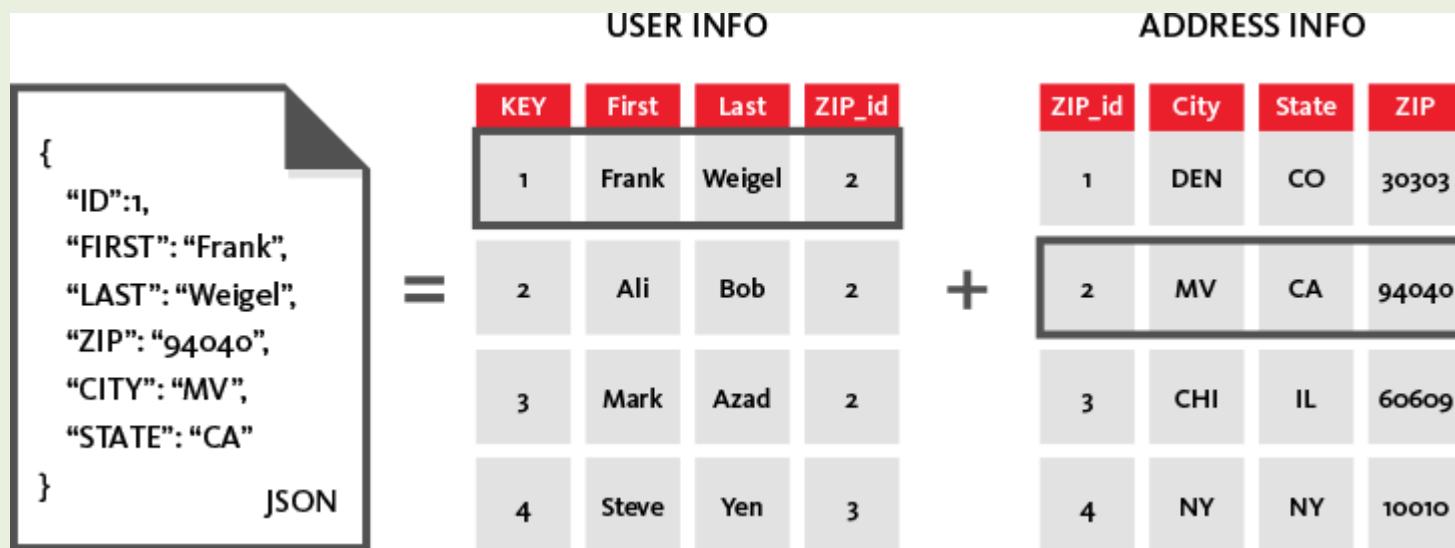
Isso não os torna adequado para aplicações que requerem escalabilidade fácil e dinâmica.

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.

Características NoSQL

- Sistemas NoSQL possuem várias características em comum
 - Livres de esquema
 - Alta disponibilidade (Confiabilidade, recuperabilidade, detecção rápida de erros e operações contínuas)
 - Escalabilidade
- Mesmo assim, possuem diversas características únicas quanto ao
 - Modo de armazenamento dos dados
 - Modelo de dados

Modelo de dados mais flexível



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

Modelo de Dados

□ Existem quatro categorias:

- 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

- 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:

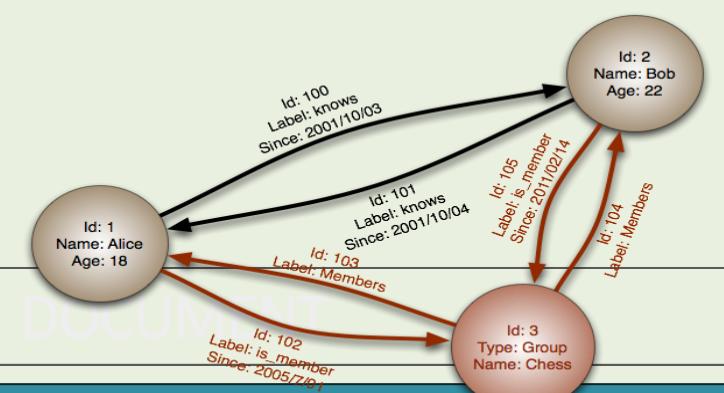
Column family, Key, Column name e value

Pessoas ; 4564 ; nome : Ana ; idade : 30

example-db (Keyspace)	
users (Column Family)	
john Doe	john@example.com
jane Doe	jane@example.com

Modelo de Dados

KEY VALUE COLUMN GRAPH



- 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: Vertice “Alice” conhece o vertice “Bob” desde 2001

Modelo de Dados

KEY VALUE

COLUMN

GRAPH

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"}]}}
```

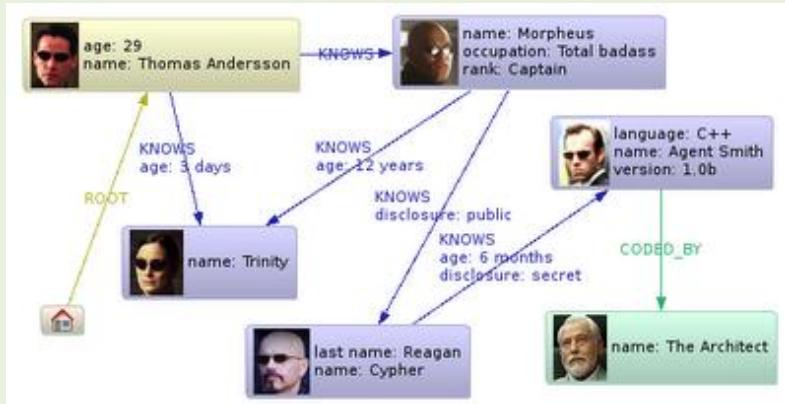


Classificação NOSQL

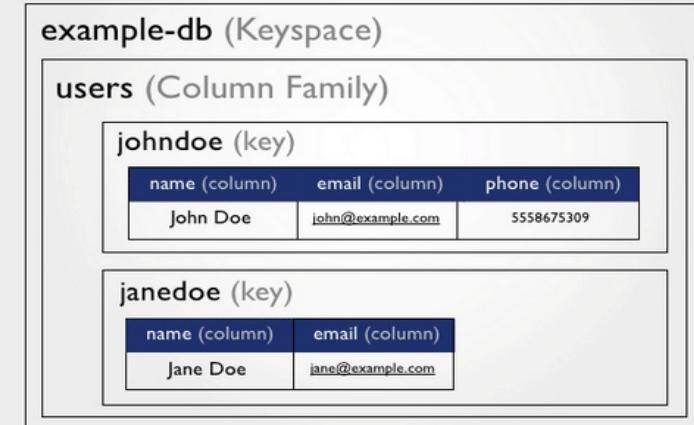
Key - Value

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

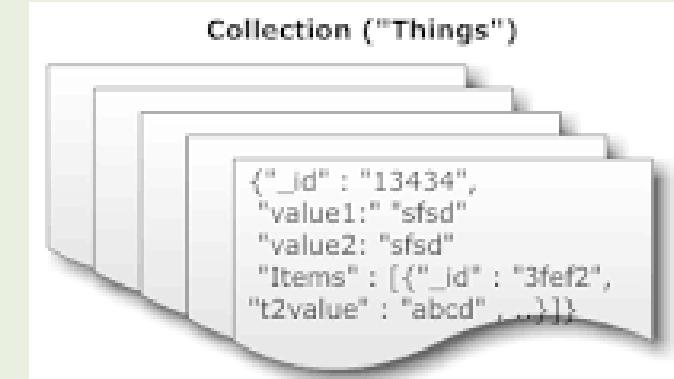
Graph



Column



Document

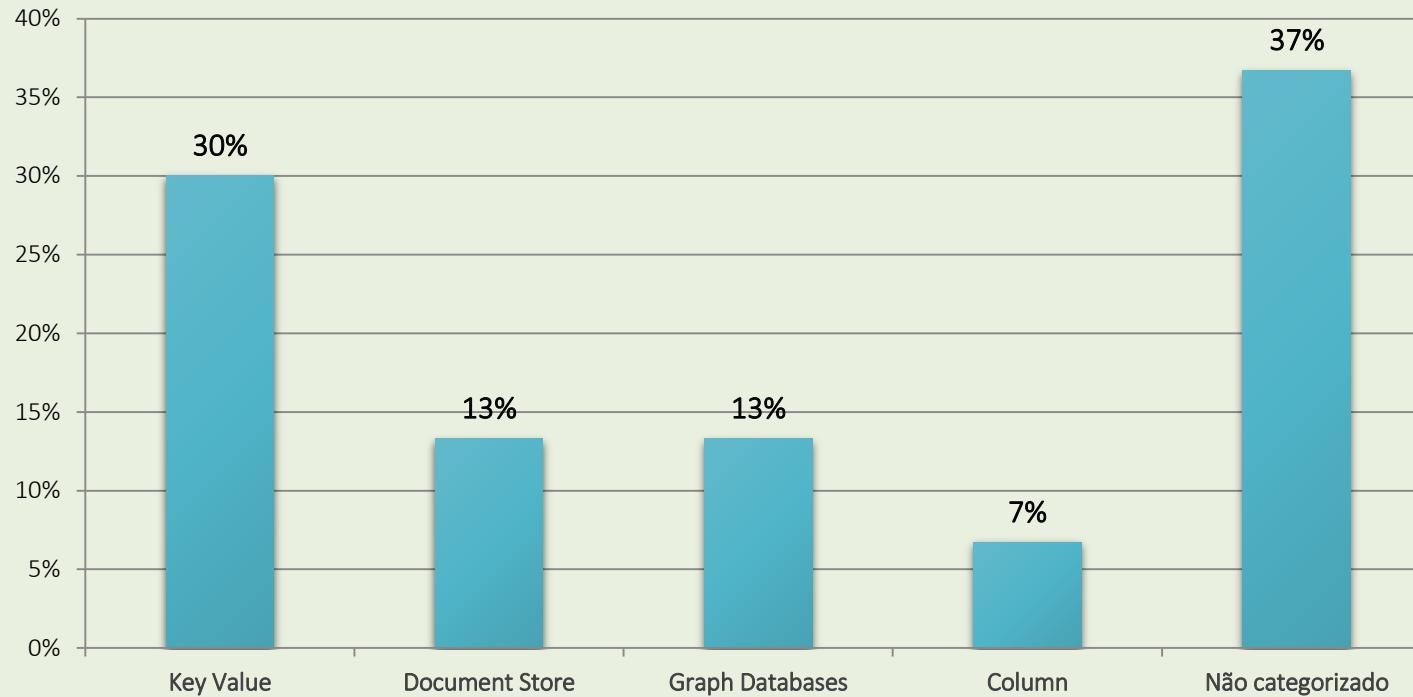


DB Ranking

283 systems in ranking, October 2015

Rank			DBMS	Database Model	Score		
Oct 2015	Sep 2015	Oct 2014			Oct 2015	Sep 2015	Oct 2014
1.	1.	1.	Oracle	Relational DBMS	1466.95	+3.58	-4.95
2.	2.	2.	MySQL	Relational DBMS	1278.96	+1.21	+15.99
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1123.23	+25.40	-96.37
4.	4.	↑ 5.	MongoDB 	Document store	293.27	-7.30	+52.86
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	282.13	-4.05	+24.41
6.	6.	6.	DB2	Relational DBMS	206.81	-2.33	-0.86
7.	7.	7.	Microsoft Access	Relational DBMS	141.83	-4.17	+0.19
8.	8.	↑ 10.	Cassandra 	Wide column store	129.01	+1.41	+43.30
9.	9.	↓ 8.	SQLite	Relational DBMS	102.67	-4.99	+7.71
10.	10.	↑ 12.	Redis 	Key-value store	98.80	-1.86	+19.42

Classificação de SGBDs NoSQL



150 tipos de banco de dados NOSQL

❑ Fonte: <http://nosql-database.org/>

❑ Dados compilados manualmente

Classificação de SGBD NOSQL e Produtos

KEY VALUE	COLUMN	GRAPH	DOCUMENT
Amazon DynamoDB (Beta)  	 Cassandra 	 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

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

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

Quando e qual utilizar?

Sessões de usuários

Dados Financeiros

Blog ou Social Media

Key - Value

Relacional

Graph

Catálogo Produtos

Relatórios

Atividades e logs de usuário

Document

Relacional

Column

Fonte: Martin Fowler

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

Dá pra fazer query?

I ❤️ SQL

Dá pra fazer query?

Amazon DynamoDB – Key-Value

SQL Query

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

AWS Query

The screenshot shows the 'Amazon DynamoDB Explore Table: Reply' interface. At the top, there are tabs for 'List Tables' and 'Browse Items', with 'Browse Items' being active. Below the tabs are buttons for 'Scan' (unchecked), 'Query' (checked), 'Go', 'New Item', 'Edit Item', 'Copy to New', and 'Delete Item'. To the right of these buttons are icons for 'Edit', 'Back', 'Forward', and '1 to 1'. Underneath these controls, there are two sets of filter fields. The first set, under 'Index Name:', shows 'Reply: Id, ReplyDateTime'. The second set, under 'Hash Key (*):', shows 'Id' with the condition 'equal to'. The third set, under 'Range Key:', shows 'ReplyDateTime' with the condition 'greater than' and the value '2014-01-12'. At the bottom, there is a 'Sort:' section with radio buttons for 'Ascending' (selected) and 'Descending'.

Dá pra fazer query?

MongoDB - Document

SQL Query

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

Operation Find

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

The diagram illustrates the mapping of SQL query components to MongoDB operation components. It shows a SQL query on the left and its corresponding MongoDB operation on the right, with arrows pointing from the SQL components to the MongoDB components they map to.

- projection:** Points to the columns selected in the SQL query (_id, name, address).
- table:** Points to the table name in the SQL query (users).
- select criteria:** Points to the WHERE clause in the SQL query (age > 18).
- cursor modifier:** Points to the LIMIT clause in the SQL query (LIMIT 5).
- collection:** Points to the collection name in the MongoDB operation (db.users).
- query criteria:** Points to the query condition in the MongoDB operation ({ age: { \$gt: 18 } }).
- projection:** Points to the projection fields in the MongoDB operation ({ name: 1, address: 1 }).
- cursor modifier:** Points to the cursor modifier in the MongoDB operation (.limit(5)).

Dá pra fazer query?

Neo4j- Graph

SQL Query

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

Comandos CRUD

(Create, Read, Update, Delete)

são iguais

← select criteria
← cursor modifier

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
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



Fonte:

Cases



SGBD:

sistema de processamento de faturas mensais

NOSQL:

Sistema focado em recomendações de melhores filmes.

Cases



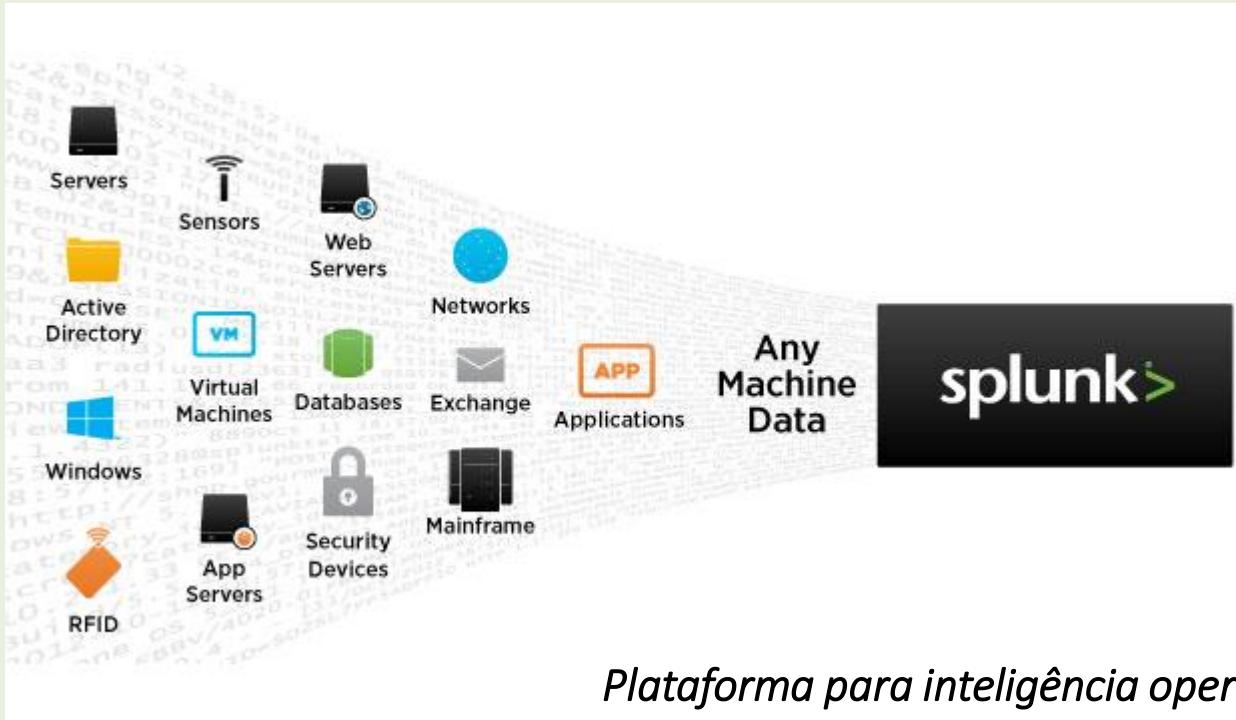
SGBD:

Sistemas de processamento de ordem de venda

NOSQL:

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

Cases



SGBD:

Dados de clientes, produtos e RH

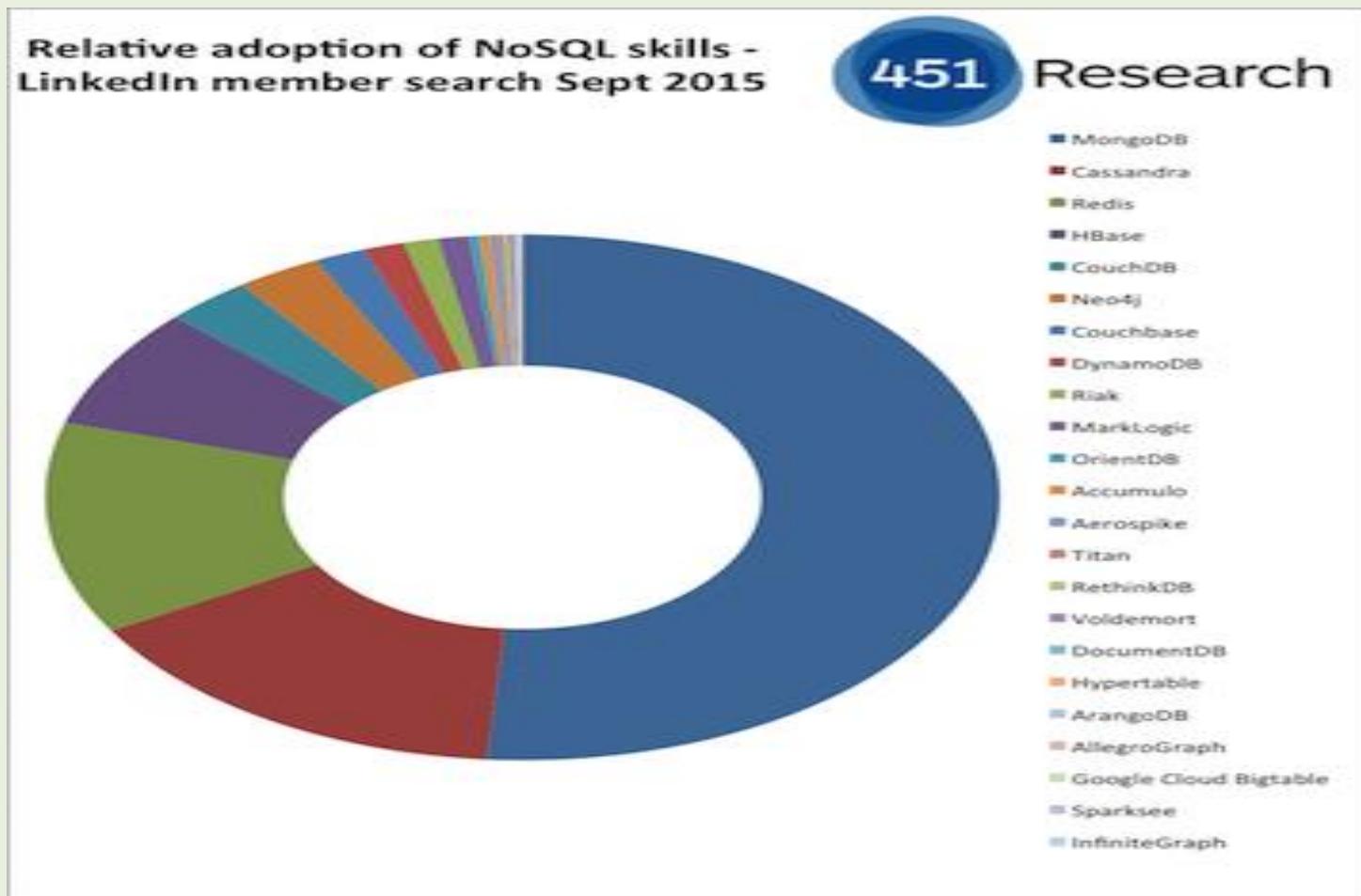
NOSQL:

Explorar, analisar e virtualização de dados

Oportunidades no mercado



Profissionais no mercado



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

Perguntas



Referências

<http://nosql-database.org/>

<http://www.couchbase.com/nosql-resources/what-is-no-sql>

<http://neo4j.com/customers/>

<http://aws.amazon.com/dynamodb/>

<http://www.nosqlfordummies.com/>

http://www.strozzi.it/cgi-bin/CSA/tw7/l/en_US/NoSQL/Home%20Page

<http://blog.parityresearch.com/21-nosql-innovators-to-look-for-in-2020/>

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

<http://www.gartner.com/technology/reprints.do?id=1-1M9YEHW&ct=131028&st=sb>

<http://www.tomsitpro.com/articles/rdbms-sql-cassandra-dba-developer,2-547-2.html>

<http://www.slideshare.net/kevinweil/nosql-at-twitter-nosql-eu-2010>

<http://www.slideshare.net/thobe/nosql-for-dummies>

Referências

<http://www.mongodb.com/events/>

<http://docs.mongodb.org/manual/core/read-operations-introduction/>

http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf

<https://cassandra.apache.org/doc/cql/CQL.html>

<http://www.indeed.com/jobtrends?q=nosql+developer&l=>

<http://blog.nahurst.com/visual-guide-to-nosql-systems>

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

<http://www.infoq.com/br/news/2014/06/oracle-nosql-database-3.0>

<http://www.infoq.com/br/news/2014/06/splunk-hunk-6.1>

<http://docs.neo4j.org/chunked/stable/query-predicates.html>

<http://http://www.splunk.com/>

<http://pt.slideshare.net/FernandoCunha15/nosql-uma-breve-introduo-44513664>