



Banco de dados NoSQL - Introdução

Prof. Gustavo Leitão

AGENDA

- Motivação
- O Problema
- O que é Big Data?
- Bancos SQL
- Bancos NoSQL



- 2,74 bilhões de usuários ativos por mês
- 1,82 bilhão de usuários logam por dia
- 4,5 bilhões de likes por dia
- 300 milhões de upload de fotos por dia
- 510 mil comentários por minuto
- 1 em cada 5 pageviews no EUA é do facebook



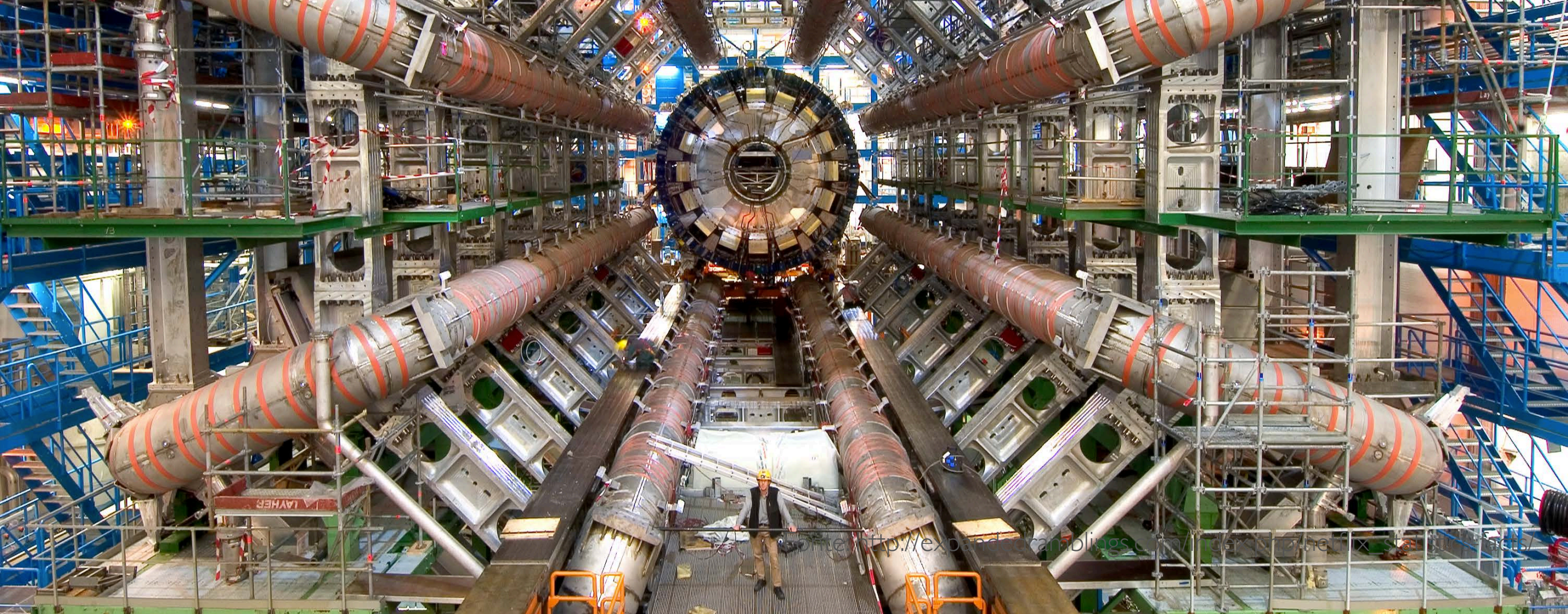
- 183 milhões de assinantes
- 19% da banda do EUA é ocupada pelo Netflix
- Usuários assistem 1 bilhão de horas por semana

Fonte: http://expandedramblings.com/index.php/netflix_statistics-facts/



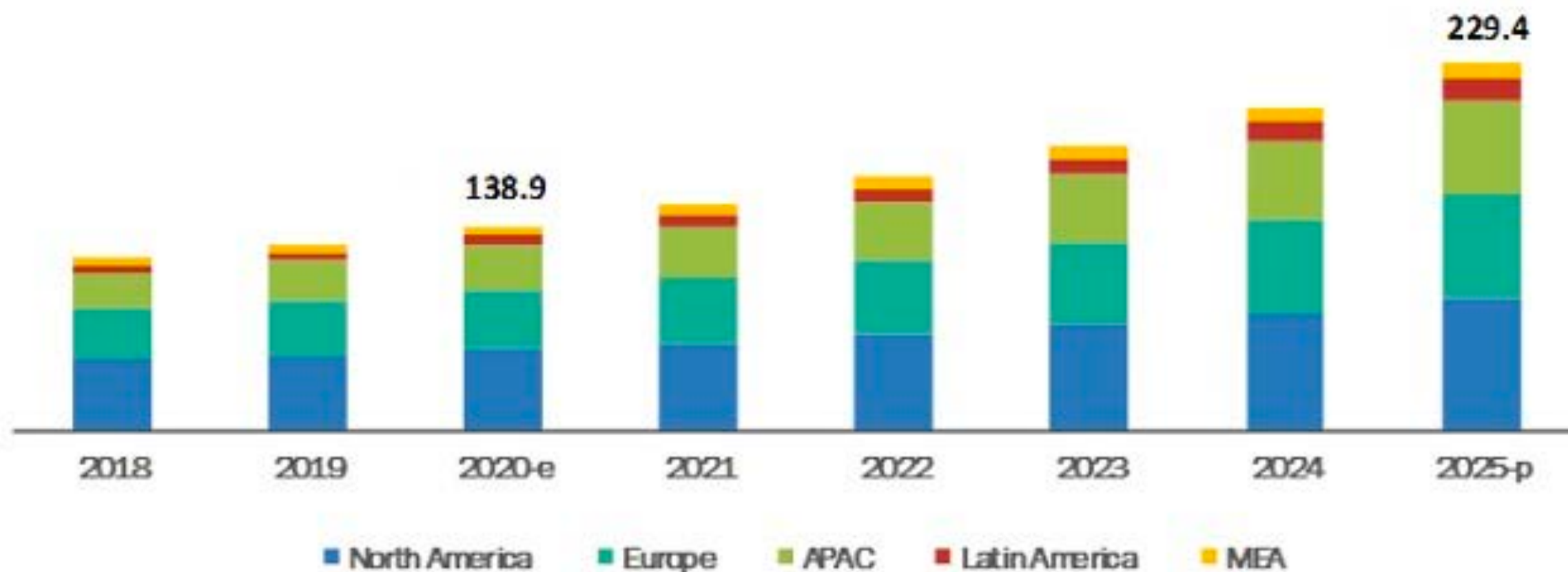
- 1,3 bilhão de usuários
- 330 milhões de usuário ativos por mês
- 500 milhões de tweets por dia (6 mil por segundo)

Fonte: <https://www.brandwatch.com/blog/44-twitter-stats-2016/>



- LHC produz 1 petabyte de dados por segundo (durante experimento)
- Apenas 1% são processados
- 25 petabytes por ano

Big Data Market, By Region (USD Billion)



Source: MarketsandMarkets Analysis

PROBLEMA

Aumento do volume de dados



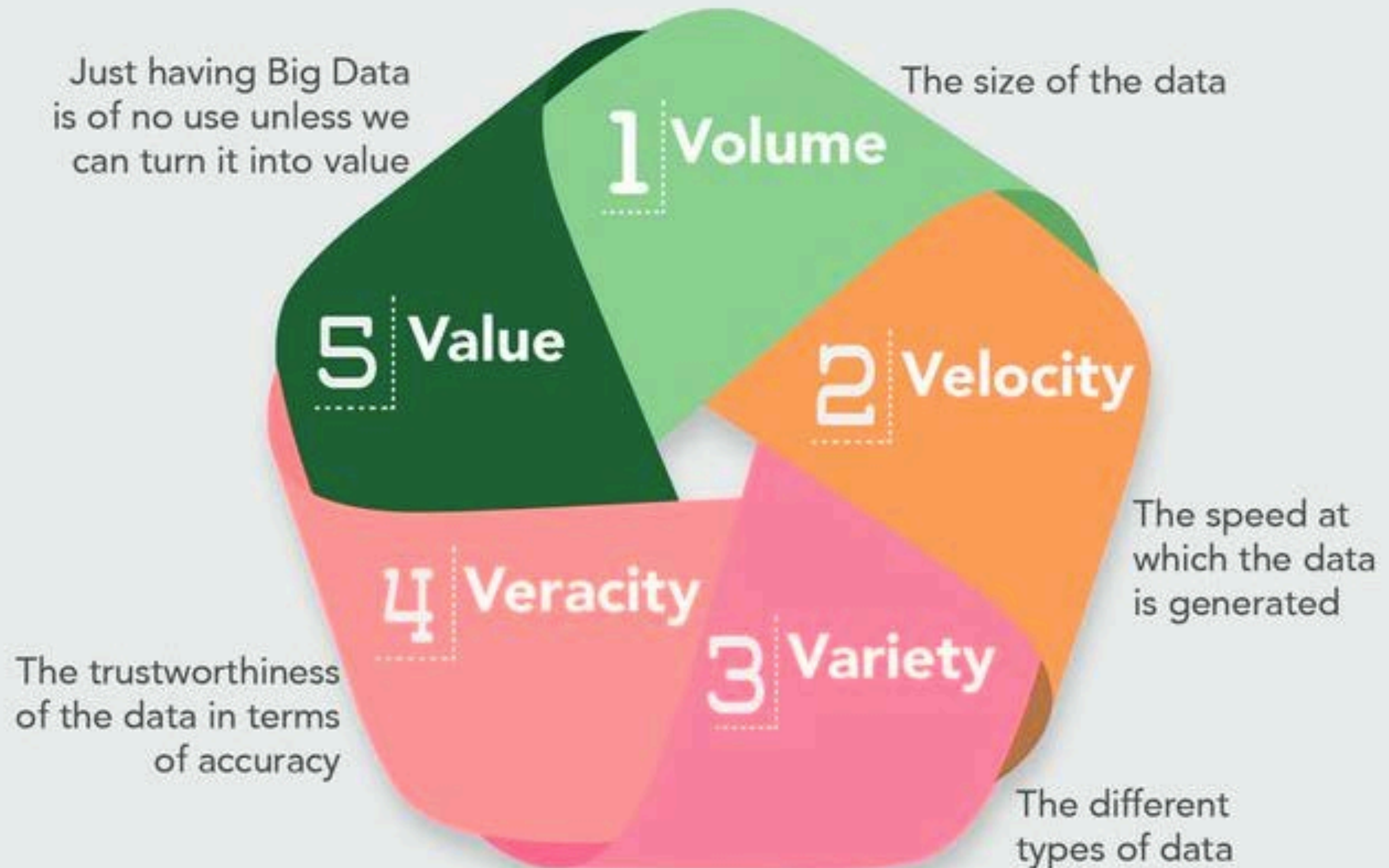
Limitação das soluções clássicas de armazenamento



Oportunidade de utilizar computação paralela com hardware de baixo custo

“Big data é uma nova geração de tecnologias e arquiteturas, desenhadas de maneira econômica para extrair **valor** de grandes **volumes** de dados, provenientes de uma **variedade** de fonte, permitindo alta **velocidade** na captura, exploração e análise dos dados.” (IDC, 2011)

THE 5 Vs OF BIG DATA



Estruturados

- Possui esquema
- Formato bem definido
- Conhecimento prévio da estrutura de dados
- Simplicidade para relacionar informações
- Dificuldade para alterar o modelo

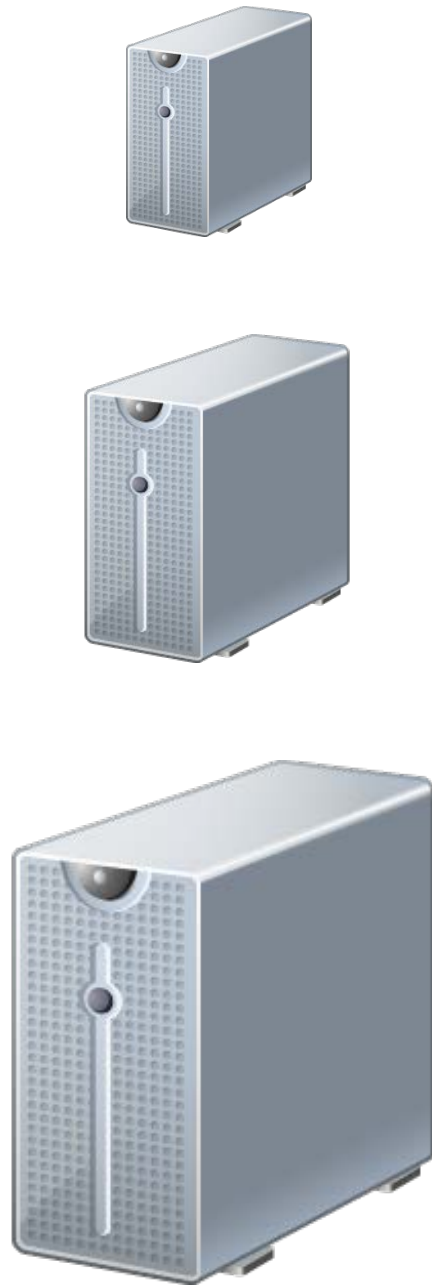


Não estruturados

- Sem tipo pré-definido
- Não possui estrutura regular
- Pouco ou nenhum controle sobre a forma
- Manipulação mais simplificada
- Facilidade de Alteração

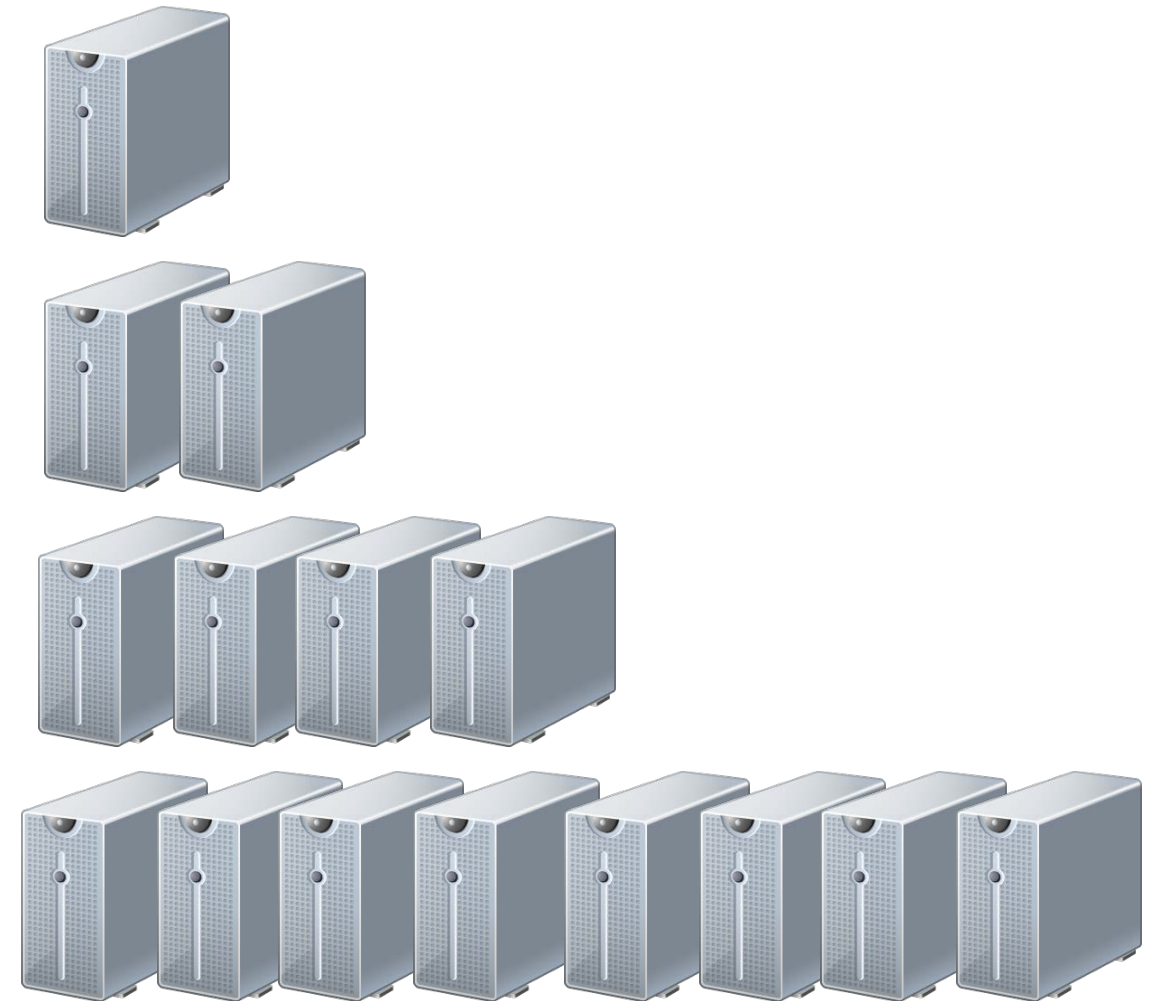


Escalabilidade Vertical



Scale-in

Escalabilidade Horizontal



Scale-out

BANCOS SQL

BANCOS DE DADOS SQL

- Dados são organizados em tabelas
- A tabela é composta por um conjunto de colunas pré-definidas
- Cada item da tabela é chamado de **registro**

Emp No	Name	Age	Department	Salary
001	Alex S	26	Store	5000
002	Golith K	32	Marketing	5600
003	Rabin R	31	Marketing	5600
004	Jons	26	Security	5100

BANCOS DE DADOS SQL

As tabelas podem possuir
relacionamentos
permitindo evitar
replicação dos dados

Cada registro deve conter
um identificador único
chamado **chave primária**

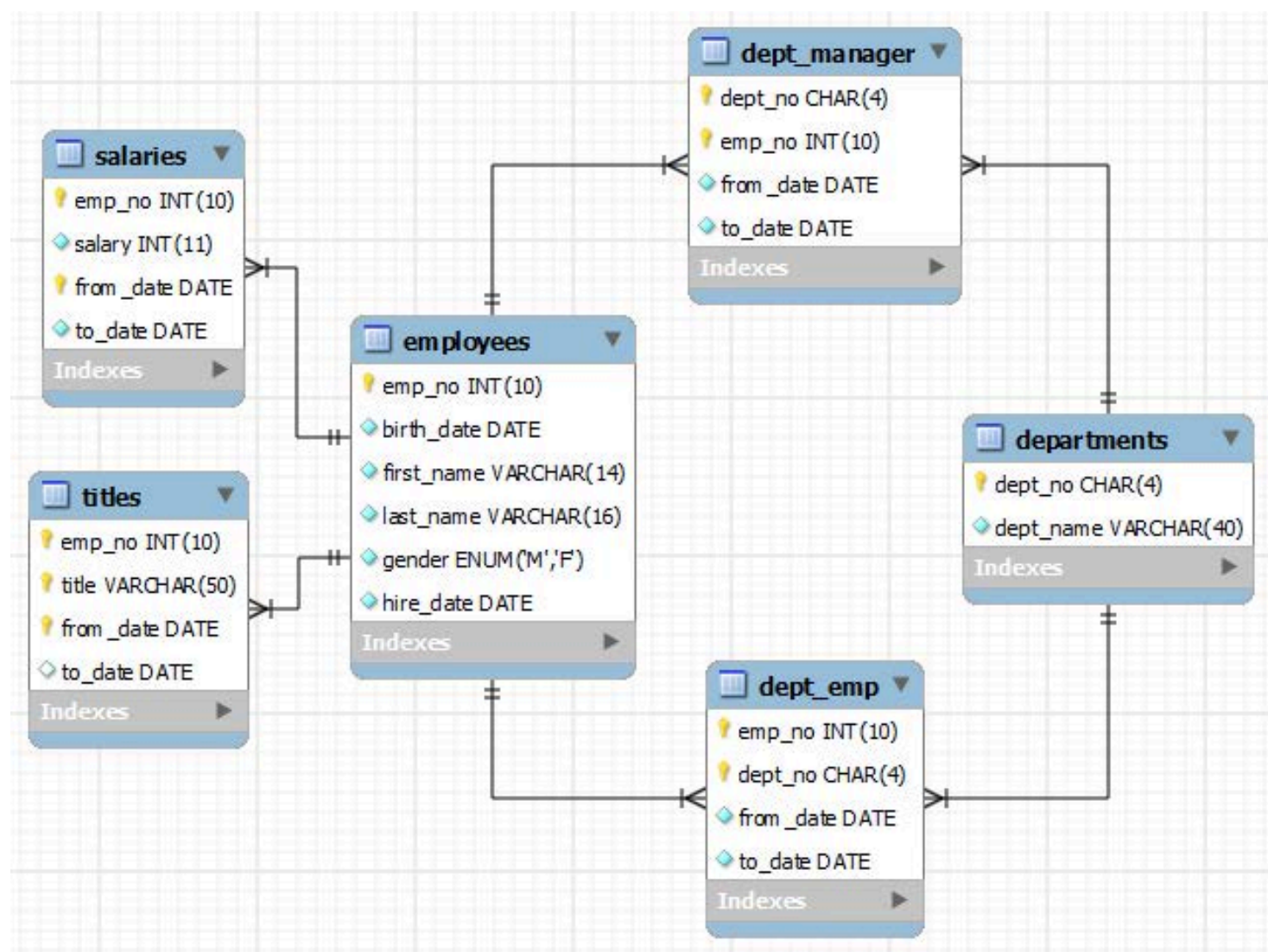
Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	\$36	Swimming	\$17
100	Squash	\$40	Swimming	\$17
182	Tennis	\$36		
219	Swimming	\$15	Golf	\$47

BANCOS DE DADOS SQL

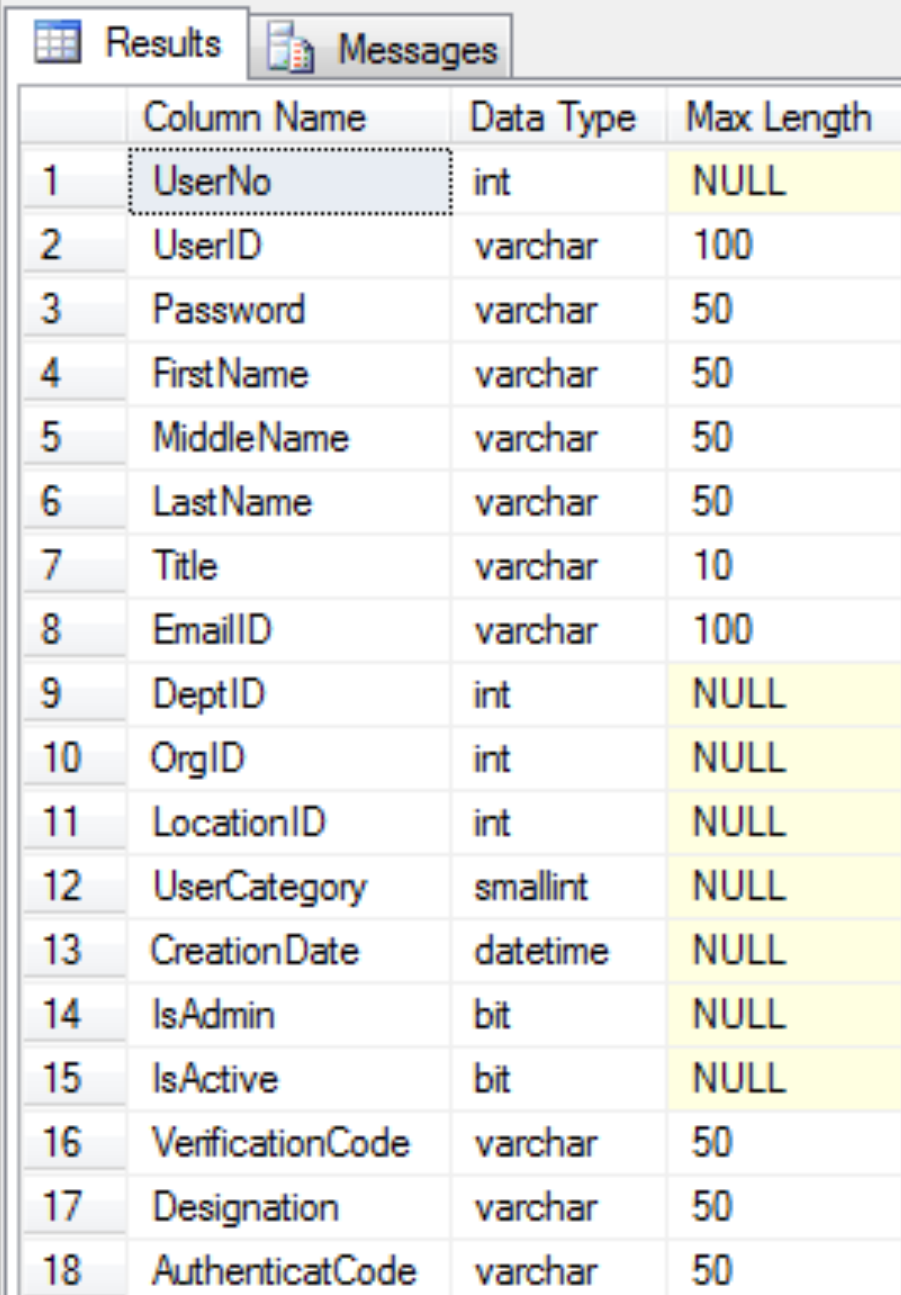


Para se fazer referência a outra tabela utiliza-se o identificador do registro na outra tabela (chave primária). O identificador de uma outra tabela é chamado de **chave estrangeira**

BANCOS DE DADOS SQL

Cada coluna possui um tipo pré-definido.

Campos textuais é possível inclusive definir um tamanho máximo de caracteres

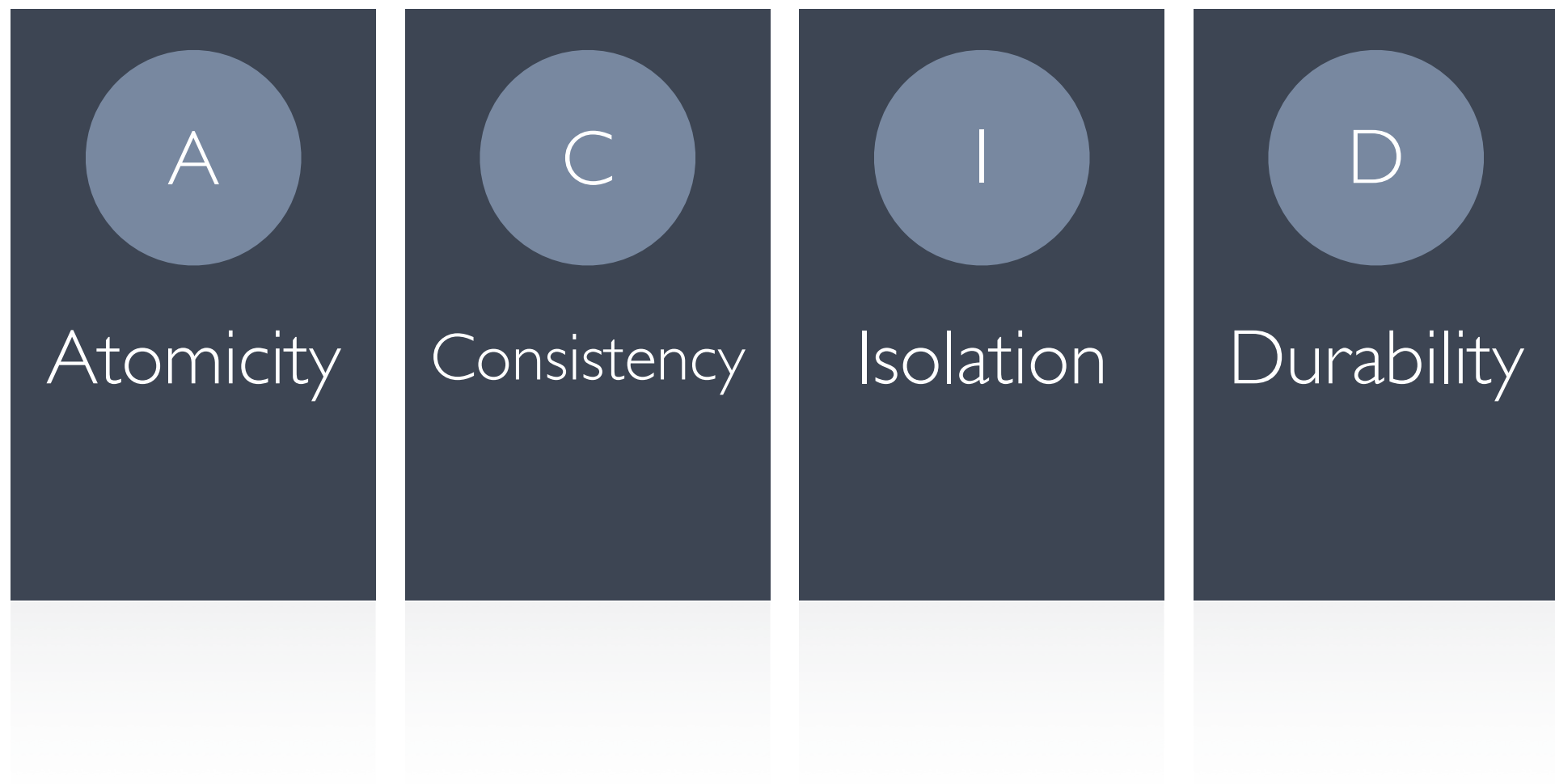


The screenshot shows a SQL Server interface with two tabs: 'Results' and 'Messages'. Below the tabs is a table with 18 columns. The columns are: Column Name, Data Type, and Max Length. The table is as follows:

	Column Name	Data Type	Max Length
1	UserNo	int	NULL
2	UserID	varchar	100
3	Password	varchar	50
4	FirstName	varchar	50
5	MiddleName	varchar	50
6	LastName	varchar	50
7	Title	varchar	10
8	EmailID	varchar	100
9	DeptID	int	NULL
10	OrgID	int	NULL
11	LocationID	int	NULL
12	UserCategory	smallint	NULL
13	CreationDate	datetime	NULL
14	IsAdmin	bit	NULL
15	IsActive	bit	NULL
16	VerificationCode	varchar	50
17	Designation	varchar	50
18	AuthenticatCode	varchar	50

BANCOS DE DADOS SQL

Implementam principio ACID



SQL

- SQL - Structured Query Language
- Utilizada para manipular dados em um banco de dados
- Desenvolvida na década de 70
- Suportada pelos bancos de dados relacionais

SQL

- Principais comandos
 - **INSERT** - Insere dados em uma tabela
 - **UPDATE** - Edita dados de uma tabela
 - **DELETE** - Remove registros de uma tabela
 - **SELECT** - Seleciona um conjunto de dados de uma ou mais tabelas

BANCOS NOSQL



- Termo utilizado para definir qualquer base de dados que não relacional
- Normalmente não utilizam SQL como linguagem de manipulação de dados (apesar de haver em alguns casos alguma semelhança)
- Muitas vezes possui "Schema free"
- Criada para resolver problemas que os bancos de dados não foram projetados.
- Diversos tipos diferentes
 - Bancos de chave-valor
 - Orientado a família de colunas
 - Orientados a documentos
 - Orientado a Grafos

NoSQL

Not Only SQL

Teorema CAP: Não é possível que um sistema de armazenamento distribuído satisfaça os três requisitos simultaneamente

Consistency



Availability



Partition Tolerance





Teorema CAP: Não é possível que um sistema de armazenamento distribuído satisfaça os três requisitos simultaneamente

[C] Consistência - Todos os nós veem os mesmos dados ao mesmo tempo

Uma operação de leitura retornará o valor da operação de gravação mais recente, fazendo com que todos os nós retornem os mesmos dados.



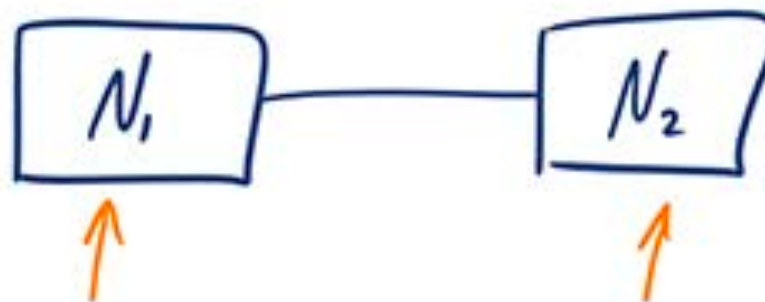


Tereoma CAP: Não é possível que um sistema de armazenamento distribuído satisfaça os três requisitos simultaneamente

[A] Disponibilidade - Todas as requisições são respondidas (sucesso ou erro)

Disponibilidade significa que qualquer cliente que fizer uma solicitação de dados obtém uma resposta, mesmo se um ou mais nós estiverem inativos.

Outra maneira de afirmar isso - todos os nós de trabalho no sistema distribuído retornam uma resposta válida para qualquer solicitação, sem exceção.





Tereoma CAP: Não é possível que um sistema de armazenamento distribuído satisfaça os três requisitos simultaneamente

[P] Tolerância a partição - O sistema continua a funcionar apesar da perda de mensagem ou falha parcial.

Uma partição é uma quebra de comunicação dentro de um sistema distribuído - uma conexão perdida ou temporariamente atrasada entre dois nós. A tolerância de partição significa que o cluster deve continuar a funcionar apesar de qualquer número de falhas de comunicação entre os nós no sistema.



NoSQL

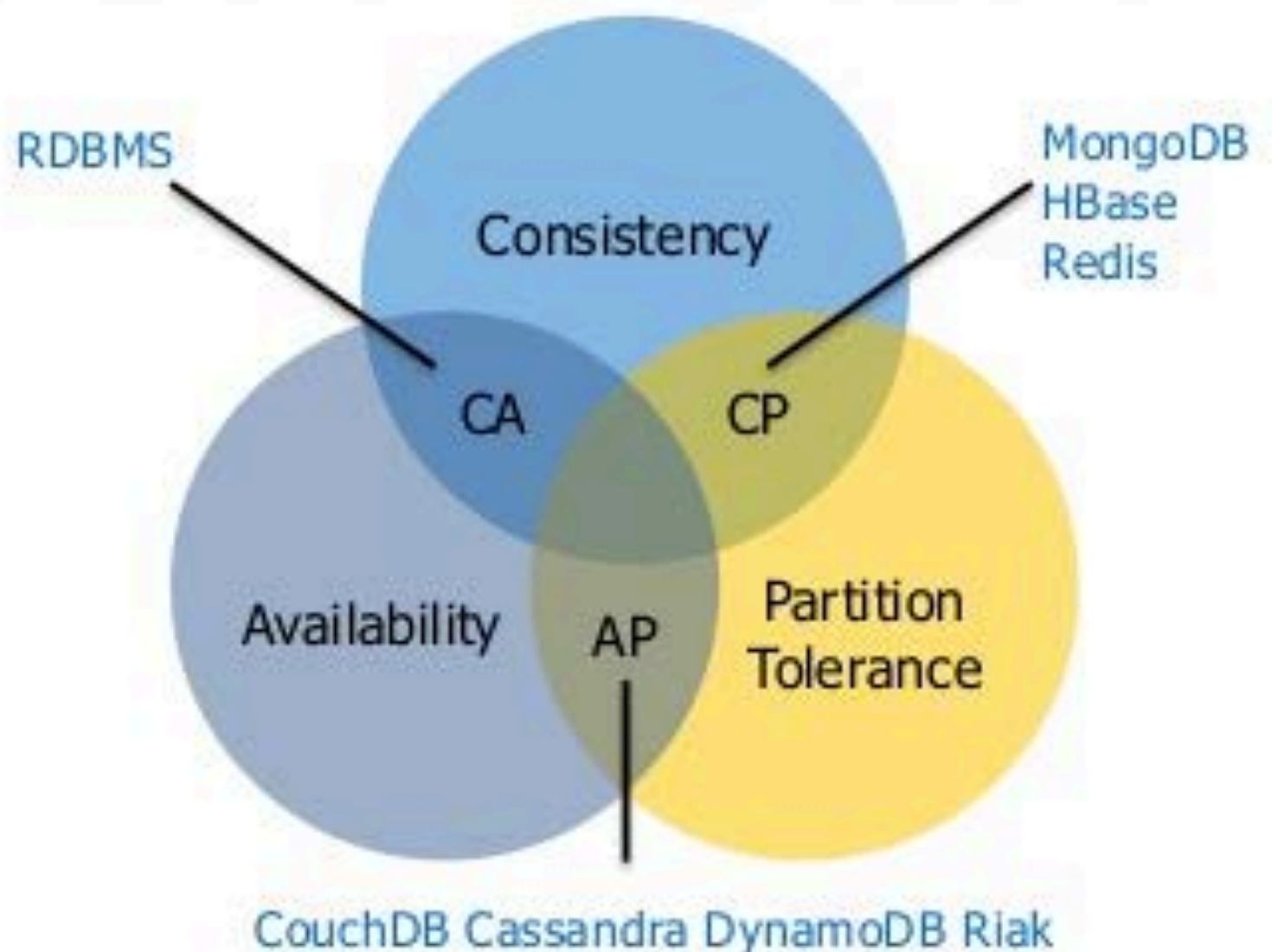
Not Only SQL

Tereoma CAP: Não é possível que um sistema de armazenamento satisfaça os três requisitos simultaneamente

C - Consistency

A - Availability

P - Partition Tolerance





- Diferentemente dos bancos relacionais que seguem o padrão ACID, os bancos NoSQL seguem o BASE
 - **B**asically **A**vailable
 - **S**oft State
 - **E**ventually Consistent



- ACID vs BASE

ACID	BASE
Consistência Forte	Consistência fraca
Isolamento	Último a escrever “ganha”
Transacional	Gerenciado pela aplicação*
Foco na consistência	Alta disponibilidade/Tolerante a falha
Banco robusto/codificação simples*	Banco simples/codificação complexa*

CHAVE-VALOR

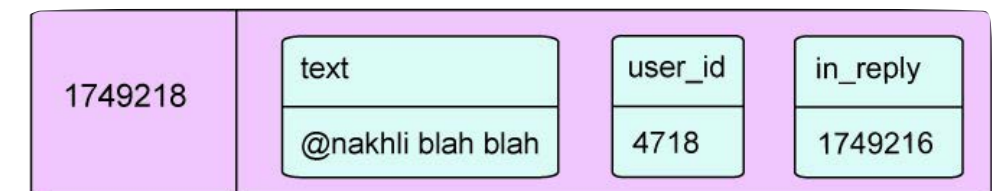
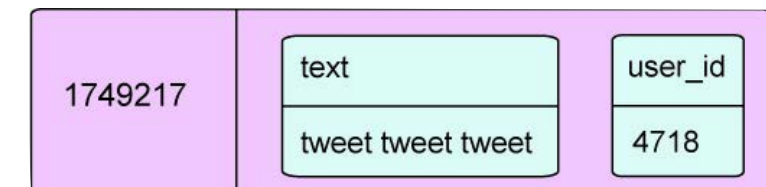
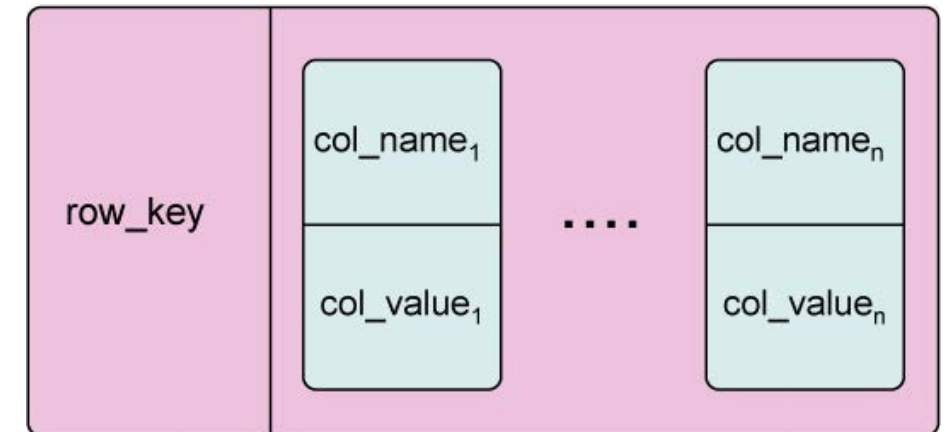
- Possui estrutura simples baseada em chave/valor
- Para cada dado a ser armazenado é definida uma chave (que deve ser única) e um valor
- Os dados são recuperados a partir de sua chave
- Projetados para lidar com altíssimo volume de dados

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623



FAMÍLIA DE COLUNAS

- Os dados são organizados por coluna
- Cada coluna é armazenada em uma estrutura separada
- Apenas as colunas de interesse são acessadas gerando um melhor desempenho em relação aos bancos baseado em tabelas
- Consultas analíticas se tornam mais eficientes, pois acessam apenas os dados necessários
- Melhora compressão dos dados



FAMÍLIA DE COLUNAS

id	Nome	Ano nascimento
1	Isaac Newton	1643
2	Albert Einstein	1879
3	Nikola Tesla	1856

Armazenamento por linha

1	Isaac Newton	1643
2	Albert Einstein	1879
2	Nikola Tesla	1856

Armazenamento por coluna

1	Isaac Newton	1643
2	Albert Einstein	1879
3	Nikola Tesla	1856

ORIENTADO A DOCUMENTO

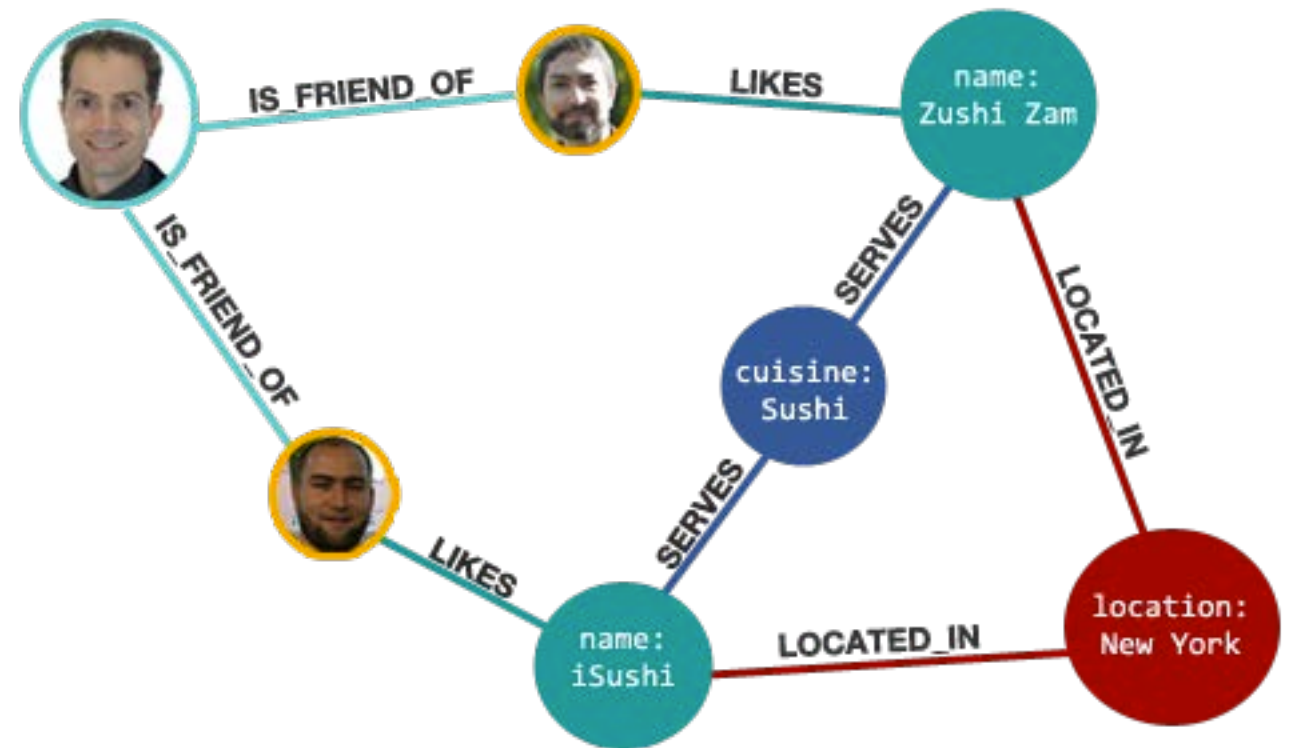
- Livre de schema
- Dados são armazenados como documentos
- Coleção de um conjunto de chave/valor
- Documentos são serializadas com JSON, XML ou BSON
- Contém todas informações importantes em um único documento
- Os documentos podem ter diferentes atributos

```
{
  "id": "1234"
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  }
}
```



ORIENTADO A GRAFOS

- Utiliza teoria de grafos para armazenamento de dados (nós e arestas)
- Os dados são acessados através do percurso do grafo
- Torna mais simples e eficientes realizar consultas de dados relacionados
 - Todos os amigos de fulano
 - Quem tem fulano como amigo



SQL



NoSQL

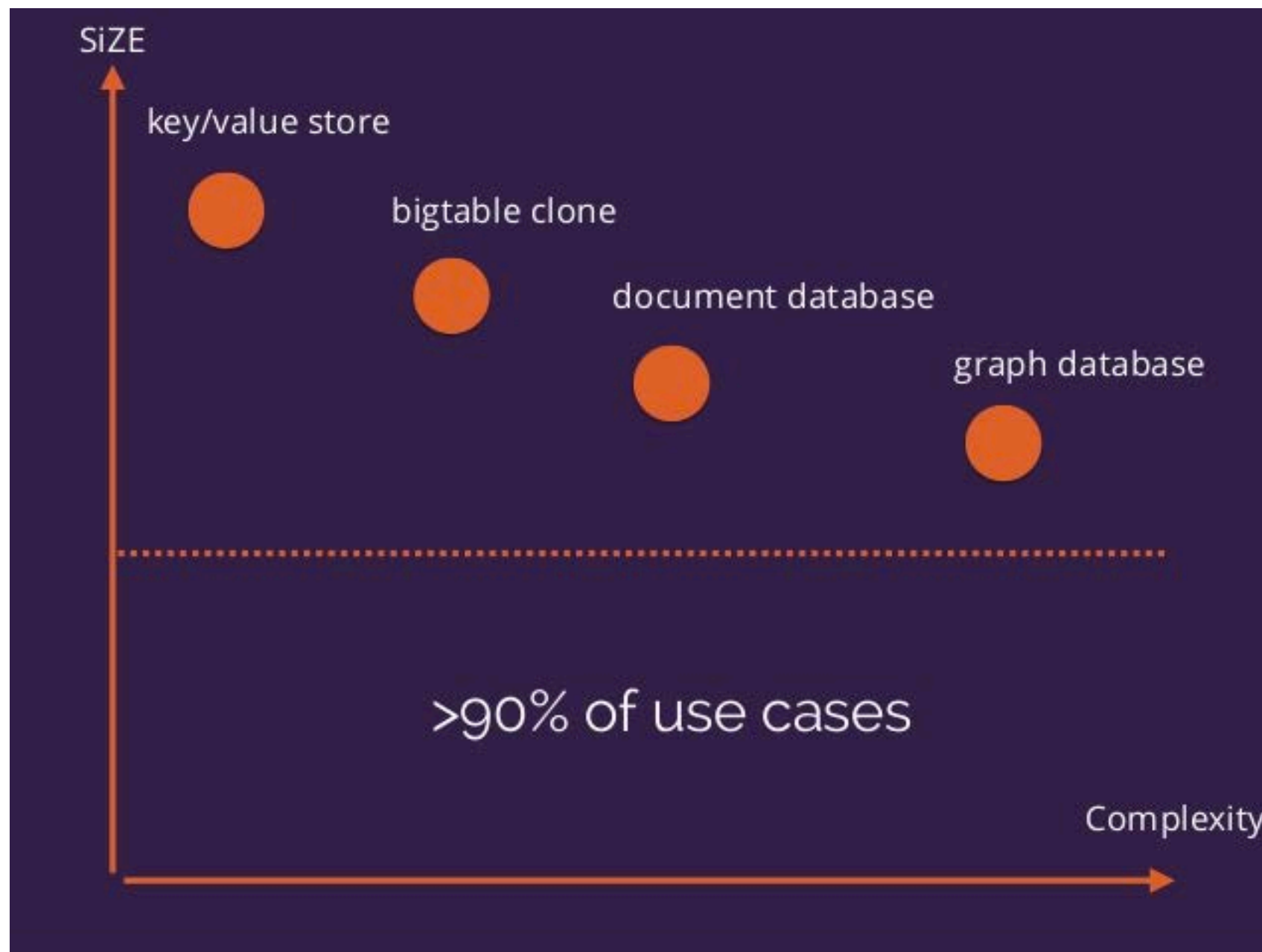


HYPERTABLE^{INC}



	NoSQL	SQL
Model	Non-relational	Relational
	Stores data in JSON documents, key/value pairs, wide column stores, or graphs	Stores data in a table
Data	Offers flexibility as not every record needs to store the same properties	Great for solutions where every record has the same properties
	New properties can be added on the fly	Adding a new property may require altering schemas or backfilling data
	Relationships are often captured by denormalizing data and presenting it in a single record	Relationships are often captured in a using joins to resolve references across tables
	Good for semi-structured data	Good for structured data
Schema	Dynamic or flexible schemas	Strict schema
	Database is schema-agnostic and the schema is dictated by the application. This allows for agility and highly iterative development	Schema must be maintained and kept in sync between application and database
Transactions	ACID transaction support varies per solution	Supports ACID transactions
Consistency	Consistency varies per solution, some solutions have tunable consistency	Strong consistency supported
Scale	Scales well horizontally	Scales well vertically

QUANDO USAR?



RESUMINDO

- SQL
 - Consistência for mais importante que disponibilidade
 - Volume de dados não for extremamente grande
- NoSQL
 - Disponibilidade e Desempenhos forem mais importantes que consistência
 - Volume de dados for extremamente grande (>500GB)



Banco de dados NoSQL - Introdução

Prof. Gustavo Leitão