

BANCO DE DADOS

**Introdução a Bancos de Dados Não
Relacionais - NoSQL**

BANCO DE DADOS NoSQL

- NoSQL quer dizer exatamente isso, **não SQL** ou não Relacional.
- Atualmente o termo NoSQL foi estendido para “**Not Only SQL**”
– Não Somente SQL.
- Os Bancos NoSQL tem tido como principal insumo o Big Data e tem sido pautado muitas vezes como elemento importante no desenvolvimento de Data Science.

BANCOS RELACIONAIS

- Os bancos mais utilizados durante muitos anos, padrão para a grande maioria das necessidades de armazenar dados desde 1970.
- Formato de Tabelas (**linhas x colunas**)
- Uso de **PK (Chave Primaria)** e **FK (Chave Estrangeira)** para conectar dados entre tabelas.
- Alguns **limites de uso** passaram a ser notados:
 - Escala (petabytes)
 - Perda de performance e dificuldade de processar relacionamentos em grandes conjunto de dados.
 - Dificuldade com esquemas complexos.

BANCOS NoSQL - HISTÓRIA

- Criado por Carlo Strozzi em 1998, com a estrutura de um banco relacional, sendo aberto e sem o SQL.
- Em 2006, o artigo: “BigTable: A Distributed Storage System for Structured Data”, publicado pelo Google, reacende o tema sobre os bancos NoSQL.
- Em 2009, a discussão sobre NoSQL é retomada por Eric Evans (Rackspace), na reunião organizada por Johan Oskarsson, da Last.fm, em 11 de junho de 2009 em São Francisco nos EUA.
- Nos últimos anos, o uso de bancos de dados NoSQL tem aumentado de forma significativa.

BANCOS NoSQL – CARACTERÍSTICAS

- Não usam SQL ;
- Geralmente são open-source;
- Formas diferentes de organização (não usam tabelas): usam grafos, documentos, colunas, entre outros, o que garante melhor performance.
- Clusterização
- Usam esquemas flexíveis (formatos diferentes para o mesmo registro).
- Persistência Poliglota;
- Escala Horizontal;
- Replicação Nativa, entre outras...

BANCOS NoSQL – CARACTERÍSTICAS

- Escalonamento horizontal (Scaling out) é a capacidade de aumentar a quantidade de computadores.
- Escalonamento vertical (Scaling up) é a capacidade de alocar mais recursos ao computador (memória e processamento).
- Escalonamento horizontal é sempre mais complexo.

MODELOS NoSQL: CHAVE-VALOR

- Em chave-valor, o banco tem sua formação a partir de estruturas de chaves, que são associadas a um único valor.
- Estrutura flexível e própria para armazenamento de big data.
- Favorece a escalabilidade ao invés da consistência.
- Exemplos: Redis, DynamoDB, Ryak, Azure Table Storage, entre outros.

Nome_1010: José

Endereco_1010: Rua Y 9

Cidade_1010: São Paulo, Ribeirão Preto

Funcao_1010: DBA

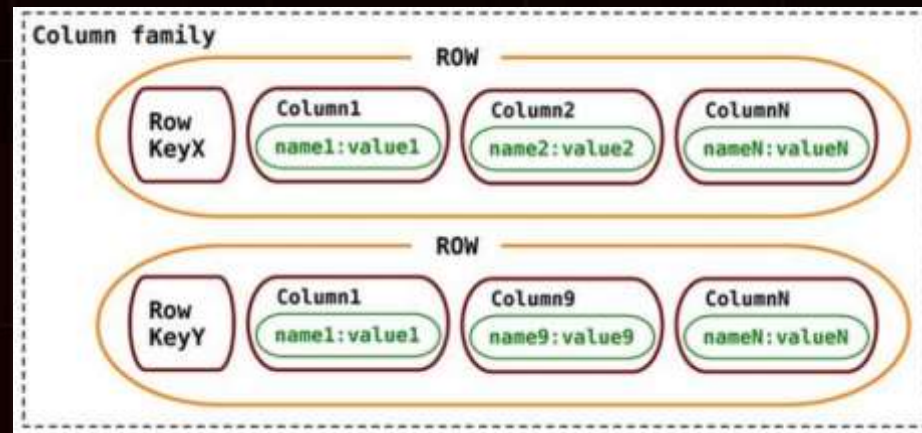
MODELOS NoSQL: CHAVE-VALOR

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	Redis	Key-value, Multi-model	165.25	+3.08	+19.61
2.	2.	2.	Amazon DynamoDB	Multi-model	73.76	+3.69	+8.90
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	36.47	+1.76	+5.67
4.	4.	4.	Memcached	Key-value	25.18	+0.68	+0.37
5.	5.	6.	etcd	Key-value	10.22	+0.80	+2.17
6.	6.	5.	Hazelcast	Key-value, Multi-model	9.37	+0.19	+0.96
7.	7.	8.	Ehcache	Key-value	7.49	+0.26	+1.21
8.	8.	7.	Aerospike	Key-value, Multi-model	5.77	+0.86	-0.89
9.	9.	10.	Riak KV	Key-value	5.40	+0.82	+0.40
10.	10.	11.	Ignite	Multi-model	4.93	+0.54	+0.06
11.	11.	9.	ArangoDB	Multi-model	4.92	+0.53	-0.47
12.	12.	12.	OrientDB	Multi-model	4.45	+0.26	-0.37
13.	13.	13.	Oracle NoSQL	Multi-model	4.31	+0.61	+0.09
14.	14.	17.	RocksDB	Key-value	3.58	+0.49	+0.72
15.	15.	14.	InterSystems Caché	Multi-model	3.24	+0.34	-0.22














Fonte: <https://db-engines.com/en/ranking/key-value+store>

MODELOS NoSQL: COLUNAS

- No modelo colunas, o banco de dados faz armazenamento em linhas particulares de tabela.
- Esse esquema é o oposto dos bancos relacionais.
- O armazenamento em colunas é similar a uma tabela, onde cada linha pode ter uma ou mais colunas. O número de colunas por linha pode variar.
- Exemplos: Hbase, Cassandra, Hypertable, Apache Flink, etc.



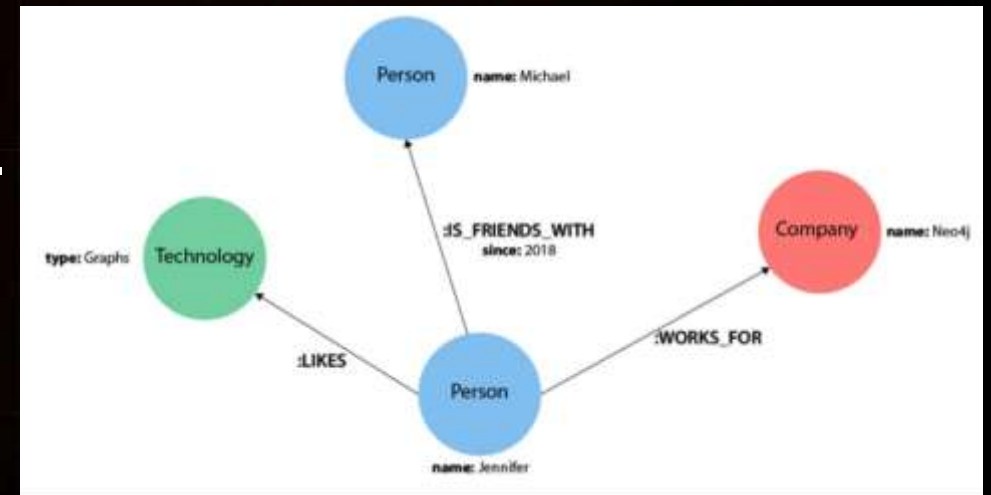
MODELOS NoSQL: COLUNAS

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	Cassandra 	Wide column	114.11	+3.18	-4.90
2.	2.	2.	HBase 	Wide column	43.52	+0.27	-5.22
3.	3.	3.	Microsoft Azure Cosmos DB 	Multi-model 	36.47	+1.76	+5.67
4.	4.	4.	Datastax Enterprise 	Wide column, Multi-model 	7.76	+0.21	-0.60
5.	5.	5.	Microsoft Azure Table Storage	Wide column	4.75	+0.46	-0.52
6.	6.	6.	Accumulo	Wide column	3.94	+0.05	+0.08
7.	7.	7.	Google Cloud Bigtable	Wide column	3.58	+0.47	+0.65
8.	8.	8.	ScyllaDB 	Multi-model 	2.71	+0.57	0.00
9.	9.	9.	HPE Ezmeral Data Fabric	Multi-model 	0.90	+0.04	+0.27
10.	10.	 11.	Elassandra	Wide column, Multi-model 	0.59	+0.04	+0.32
11.	11.		Amazon Keyspaces	Wide column	0.44	0.00	
12.	12.	 10.	Alibaba Cloud Table Store	Wide column	0.38	+0.02	+0.03
13.	13.		SWC-DB	Wide column, Multi-model 	0.00	±0.00	

Fonte: <https://db-engines.com/en/ranking/wide+column+store>

MODELOS NoSQL: GRAFOS

- Um grafo é formado por vértices (pontos) e arestas (ligações entre os pontos).
- Semelhante ao modelo relacional, com estrutura de relacionamento por meio de atributos.
- Muito usado em redes sociais, bioinformática e estrutura de bancos semânticos.
- Exemplos: Neo4J, Virtuoso, ArangoDB.



MODELOS NoSQL: GRAFOS

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	Neo4j	Graph	55.75	+3.52	+7.48
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	36.47	+1.76	+5.67
3.	3.	3.	ArangoDB	Multi-model	4.92	+0.53	-0.47
4.	4.	4.	OrientDB	Multi-model	4.45	+0.26	-0.37
5.	5.	5.	Virtuoso	Multi-model	3.69	+0.25	+1.41
6.	7.	7.	JanusGraph	Graph	2.49	+0.32	+0.48
7.	6.	9.	GraphDB	Multi-model	2.31	+0.06	+1.06
8.	8.	6.	Amazon Neptune	Multi-model	2.20	+0.26	+0.03
9.	9.	13.	TigerGraph	Graph	1.88	+0.14	+0.99
10.	10.	11.	Stardog	Multi-model	1.73	+0.02	+0.57
11.	12.	8.	Dgraph	Graph	1.71	+0.35	+0.31
12.	11.	10.	Fauna	Multi-model	1.67	+0.18	+0.48
13.	13.	14.	AllegroGraph	Multi-model	1.34	+0.01	+0.45
14.	14.	12.	Giraph	Graph	1.20	+0.07	+0.23
15.	15.	21.	Nebula Graph	Graph	1.04	+0.01	+0.79

Fonte: <https://db-engines.com/en/ranking/graph+dbms>

MODELOS NoSQL: DOCUMENTOS

- Dados são compreendidos como documentos.
- Estrutura totalmente flexível e não precisa ter colunas pré-montadas, como é o caso dos bancos de dados colunares.
- Especialmente eficiente para tratar dados não estruturados, já que cada documento pode ter um formato diferente.
- Pense no documento como um arquivo Json.
- Exemplos: MongoDB, Elastic, IBM Domino, CouchBase.



MODELOS NoSQL: DOCUMENTOS

- Json – É um formato de intercâmbio de dados. Fácil de entender por humanos e também para análise, geração e leitura por máquinas.

```
{
  "book": [
    {
      "id": "444",
      "language": "C",
      "edition": "First",
      "author": "Dennis Ritchie "
    },
    {
      "id": "555",
      "language": "C++",
      "edition": "second",
      "author": " Bjarne Stroustrup "
    }
  ]
}
```





























Json

```
_id: ObjectId("5eb3d668b31de5d588f4292a")
> address: Object
  borough: "Brooklyn"
  cuisine: "American"
~ grades: Array
  ~ 0: Object
    date: 2014-06-10T00:00:00.000+00:00
    grade: "A"
    score: 5
  > 1: Object
  > 2: Object
  > 3: Object
name: "Riviera Caterer"
restaurant_id: "40356018"
```

```
_id: ObjectId("5eb3d668b31de5d588f4292b")
> address: Object
  borough: "Brooklyn"
  cuisine: "Delicatessen"
~ grades: Array
  > 0: Object
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
  > 5: Object
name: "Wilken'S Fine Food"
restaurant_id: "40356483"
```

BD - Documento

MODELOS NoSQL: DOCUMENTOS

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	MongoDB 	Document, Multi-model 	488.22	+7.20	+51.14
2.	2.	2.	Amazon DynamoDB 	Multi-model 	73.76	+3.69	+8.90
3.	3.	3.	Microsoft Azure Cosmos DB 	Multi-model 	36.47	+1.76	+5.67
4.	4.	4.	Couchbase 	Document, Multi-model 	29.07	-1.16	-0.07
5.	5.	 6.	Firebase Realtime Database	Document	17.13	+0.22	+3.14
6.	6.	 5.	CouchDB	Document, Multi-model 	16.12	+0.15	+0.05
7.	7.	 8.	Realm 	Document	9.68	-0.11	+1.33
8.	8.	 7.	MarkLogic 	Multi-model 	9.38	-0.15	-1.87
9.	9.	9.	Google Cloud Firestore	Document	7.42	+0.46	-0.10
10.	10.	10.	ArangoDB 	Multi-model 	4.92	+0.53	-0.47
11.	 12.	11.	Google Cloud Datastore	Document	4.68	+0.67	-0.47
12.	 11.	12.	OrientDB	Multi-model 	4.45	+0.26	-0.37
13.	13.	13.	Oracle NoSQL	Multi-model 	4.31	+0.61	+0.09
14.	14.	 20.	Virtuoso 	Multi-model 	3.69	+0.25	+1.41
15.	 17.	 14.	RethinkDB	Document, Multi-model 	3.51	+0.42	-0.35

Fonte: <https://db-engines.com/en/ranking/document+store>

POR QUE USAR BANCOS NoSQL?

- Dificuldade em modelar bancos de dados complexos, com inúmeros joins por exemplo.
- Necessidade de implementar bancos de dados com estruturas flexíveis.
- Considerar escalonamento:
 - Escalonamento em um banco de dados relacional é muito complexo por sua natureza relacional.
 - O modelo flexível favorece muito o escalonamento em bancos NoSQL.

POR QUE USAR BANCOS NoSQL?

- Considerar consistência:
 - Os bancos de dados relacionais são muito consistentes. A consistência está no DNA dos bancos relacionais.
 - Um banco de dados NoSQL não tem como premissa garantir a consistência, isso acontece de forma eventual.

POR QUE USAR BANCOS NoSQL?

- Considerar disponibilidade:
 - O banco relacional não trabalha de forma natural com distribuição de dados, portanto tem dificuldade com a questão de disponibilidade para grandes cargas de dados.
 - Bancos NoSQL possuem alto nível de distribuição, com possibilidade de atender grandes fluxos de solicitações de dados.

BANCOS NoSQL

- Considerar o uso sempre que o **volume de dados for muito grande**.
- Bancos de dados relacionais continuam funcionando muito bem!!
Os bancos NoSQL não vieram substituí-los.
- Avalie sempre a necessidade e as características necessárias para escolher entre Relacional ou NoSQL, e fazendo a opção, deverá escolher qual o produto a usar de acordo com as necessidades do projeto.

BANCO DE DADOS

**Introdução a Bancos de Dados Não
Relacionais - NoSQL**