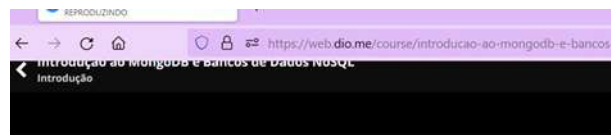


INTRODUÇÃO AO MONGODB E BANCO DE DADOS NOSQL



Introdução

Diferenças BD Relacional x BD NoSQL

Escalabilidade: Banco de dados relacional Vertical



- Aumento da capacidade para um unico recurso
- Processador, memoria e disco rígido.



Introdução

Diferenças BD Relacional x BD NoSQL

Escalabilidade: Banco de dados relacional Horizontal



- Replicas de dados APENAS PARA LEITURA



Introdução

Diferenças BD Relacional x BD NoSQL

Escalabilidade

Banco de dados **NoSQL** Horizontal



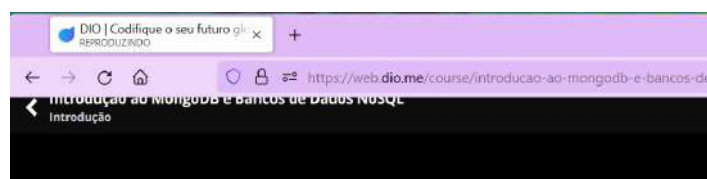
Particionando os dados (sharding) entre os nós é o mais conhecido.



Introdução

Diferenças BD Relacional x BD NoSQL

Schema: BD Relacional



Introdução

Diferenças BD Relacional x BD NoSQL

Schema: BD NoSQL



Schema-free/schemaless



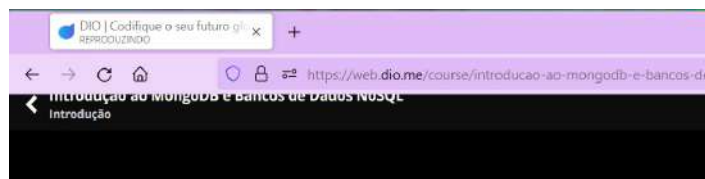
]



Introdução

Diferenças BD Relacional x BD NoSQL

Performance: BD Relacional

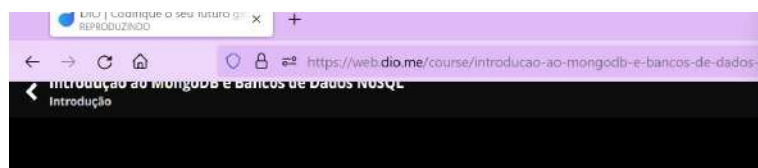


Introdução

Diferenças BD Relacional x BD NoSQL

Performance: BD NoSQL

Cluster

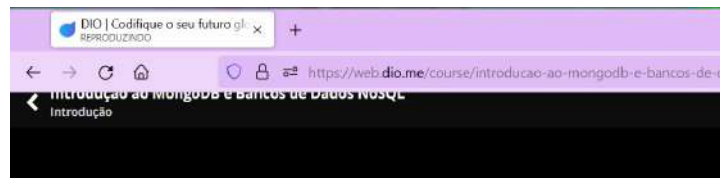


Introdução

Diferenças BD Relacional x BD NoSQL

Transações

BD Relacional	BD NoSQL
Atomicidade	Basically Available
Consistência	Soft-State
Isolamento	Eventually Consistent
Durabilidade	



Introdução

Características e vantagens

- Flexibilidade
- Escalabilidade
- Alta performance



Tipos de banco NoSQL

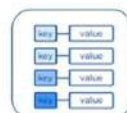
Rank	DBMS	Database Model	Score
Apr 2021	Mar 2021	Apr 2021	Mar 2021
1.	Oracle	Relational, Multi-model	1274.92
2.	MySQL	Relational, Multi-model	1220.69
3.	Microsoft SQL Server	Relational, Multi-model	1007.97
4.	PostgreSQL	Relational, Multi-model	553.52
5.	MongoDB	Document, Multi-model	460.97
6.	IBM DB/2	Relational, Multi-model	157.78
7.	Redshift	Relational, Multi-model	155.89
8.	Elasticsearch	Search engine, Multi-model	132.18
9.	SQL-DB	Relational	125.05
10.	Microsoft Access	Relational	116.72
11.	Cassandra	Wide column	114.85
12.	MarkDB	Relational, Multi-model	96.57
13.	Splunk	Search engine	89.49
14.	Hive	Relational	76.50
15.	Microsoft Azure SQL Database	Relational, Multi-model	71.04
16.	Amazon DynamoDB	Multi-model	70.77
17.	Teradata	Relational, Multi-model	70.35
18.	SAP HANA	Relational, Multi-model	53.44
19.	SAP Adaptive Server	Relational, Multi-model	51.67
20.	Nasdaq	Graph	51.04
21.	Big5	Search engine, Multi-model	50.60
22.	Pinecone	Relational	46.40
23.	Hbase	Wide column	44.16
24.	Google BigQuery	Relational	35.58
25.	Microsoft Azure Cosmos DB	Multi-model	35.52
26.	Couchbase	Document, Multi-model	30.76
27.	Presto	Search engine, Multi-model	29.45
28.	InfuxDB	Time Series, Multi-model	28.35
29.	Google Cloud	Relational	27.00



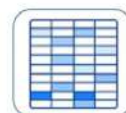
Tipos de banco NoSQL



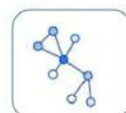
Document Store



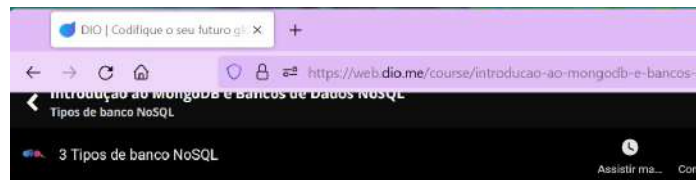
Key-Value Store



Wide-Column Store



Graph Store



Grafos

Comum em detecção de fraudes, mecanismos de recomendação, redes sociais, sistemas de arquivo, games ...



Grafos

Rank			DBMS	Database Model	Score		
May 2021	Apr 2021	May 2020			May 2021	Apr 2021	May 2020
1.	1.	1.	Neo4j	Graph	52.23	+1.19	+2.47
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	34.71	+1.19	+4.05
3.	3.	3.	ArangoDB	Multi-model	4.38	-0.39	-0.30
4.	4.	4.	OrientDB	Multi-model	4.19	-0.04	+0.05
5.	5.	5.	Virtuoso	Multi-model	3.44	+0.27	+1.09
6.	6.	8.	GraphDB	Multi-model	2.25	+0.01	+1.07
7.	7.	7.	JanusGraph	Graph	2.17	+0.07	+0.52
8.	8.	6.	Amazon Neptune	Multi-model	1.94	+0.04	+0.17
9.	10.	13.	TigerGraph	Graph			
10.	9.	10.	Stardog	Multi-model			

neo4j MATCH (bob_esponja)

neo4j CREATE (:Client {name: "Bob Esponja", age: 28, hobbies: ["Caça água-viva, Comer hamburg..."]})

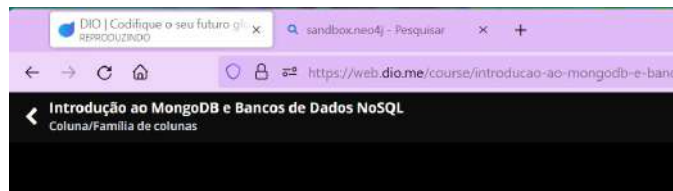
Added 1 label, created 1 node, set 3 properties, completed after 42 ms.

neo4j CREATE (:Client {name: "Bob E...})

Added 1 label, created 1 node, set 3 properties, completed after 42 ms.

Added 1 label, created 1 node, set 3 properties, completed after 42 ms.

Connect You are connected as user neo4j



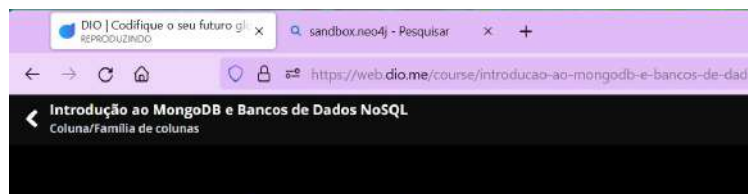
Coluna/Família de colunas:

Table with single-row partitions

partition key	performer	born	country	died	founded	style	type
John Lennon	1940	England	1980		Rock	artist	
Paul McCartney	1942	England			Rock	artist	
The Beatles		England		1927	Rock	band	

Column family view

partition key	born	country	died	founded	style	type
John Lennon	1940	England	1980		Rock	artist
Paul McCartney	1942	England			Rock	artist
The Beatles		England		1927	Rock	band



Coluna/Família de colunas:

Rank			DBMS	Database Model	Score		
May 2021	Apr 2021	May 2020			May 2021	Apr 2021	May 2020
1.	1.	1.	Cassandra	Wide column	110.93	-3.92	-8.2
2.	2.	2.	HBase	Wide column	43.24	-0.92	-6.4
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	34.71	+1.19	+4.2
4.	4.	4.	Datastax Enterprise	Wide column, Multi-model	7.55	-0.44	-0.07
5.	5.	5.	Microsoft Azure Table Storage	Wide column	4.29	-0.41	-0.39
6.	6.	6.	Accumulo	Wide column	3.88	-0.23	+0.21
7.	7.	7.	Google Cloud Bigtable	Wide column	3.11	-0.28	+0.93
8.	8.	8.	ScyllaDB	Multi-model	2.14	+0.08	+0.24
9.	9.	9.	HPE Ezmeral Data Fabric	Multi-model	0.86	+0.01	+0.26
10.	10.	11.	Elassandra	Wide column, Multi-model	0.56	-0.03	+0.32



Coluna/Família de colunas

Keyspace: agrupamento de famílias de colunas => database

Column Family/table: agrupamento de colunas => table

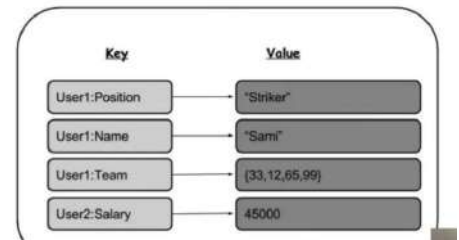
Row key: chave que representa uma linha de coluna => Primary Key

Column: representa um valor contendo: Name, Value Timestamp

Coluna/Família de colunas:

Registro de transações: compras, resultados de testes, filmes assistidos e localização mais recente do filme.
Rastreando praticamente qualquer coisa, incluindo status do pedido, pacotes etc.

Chave-valor



Chave-valor

Armazena um conjunto de dados, seja ele simples ou complexo, identificados por um identificador exclusivo.

- + Bom desempenho em aplicações na nuvem.
- Menor capacidade de busca.

Chave-valor

Uso: cache, sessão de usuário, carrinhos de compra

Chave-valor

Rank	DBMS			Database Model	Score		
	May 2021	Apr 2021	May 2020		May 2021	Apr 2021	May 2020
1.	1.	1.	1.	Redis	Key-value, Multi-model	162.17	+6.26 +18.69
2.	2.	2.	2.	Amazon DynamoDB	Multi-model	70.07	-0.66 +5.35
3.	3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	34.71	+1.19 +4.03
4.	4.	4.	4.	Memcached	Key-value	24.50	-0.52 +0.57
5.	6.	6.	6.	etcd	Key-value	9.42	+0.20 +2.14
6.	5.	5.	5.	Hazelcast	Key-value, Multi-model	9.18	-0.09 +0.51
7.	7.	7.	7.	Ehcache	Key-value	7.23	-0.18 +1.00
8.	8.	8.	8.	Aerospike	Key-value, Multi-model	4.92	-0.41 -1.14
9.	9.	9.	9.	Riak KV	Key-value	4.58	-0.50 -0.18
10.	11.	11.	11.	Ignite	Multi-model	4.39	+0.02 -0.17

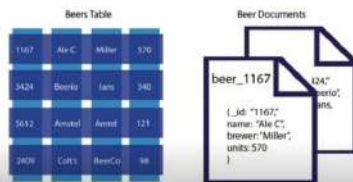
Na prática

Iremos utilizar o Redis um banco de dados, cache, messenger e fila.

- Alto desempenho
- Estrutura de dados na memória.
- Versatilidade de uso
- Replicação e persistência.

Quem usa: twitter, github, stackoverflow entre outros.

Documento



Documento

Dados e documentos autocontidos e auto descritivos.
Permite redundância e inconsistência.
Livre de esquemas podendo utilizar JSON, XML entre outros.

Documento

Rank	DBMS			Database Model	Score		
	May 2021	Apr 2021	May 2020		May 2021	Apr 2021	May 2020
1.	1.	1.	1.	MongoDB	Document, Multi-model	481.01	+11.84 +42.02
2.	2.	2.	2.	Amazon DynamoDB	Multi-model	70.07	-0.66 +5.35
3.	3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	34.71	+1.19 +4.03
4.	4.	4.	4.	Couchbase	Document, Multi-model	30.23	-0.53 +1.54
5.	5.	5.	5.	Firebase Realtime Database	Document	16.91	-0.02 +3.79
6.	6.	6.	6.	CouchDB	Document, Multi-model	15.97	-0.04 -0.95
7.	7.	7.	7.	Realm	Document	9.79	+0.66 +1.41
8.	8.	8.	8.	MarkLogic	Multi-model	9.52	-0.40 -1.44
9.	9.	9.	9.	Google Cloud Firestore	Document	6.96	+0.26 +0.53
10.	10.	10.	10.	ArangoDB	Multi-model	4.38	+0.02 -0.17



DIGITAL
INNOVATION
ONE

Introdução ao MongoDB

Quando não usar:

- Necessidade de relacionamentos/joins.
- Propriedades ACID e transações são importantes.
- Curiosidade: Diversas entidades de pagamento não homologam sistemas cujos dados financeiros dos clientes não estejam em bancos de dados relacionais tradicionais.



DIGITAL
INNOVATION
ONE

Schema Design Embedding vs Referência



Documentos autocontido

Documentos com
depência de outros
documentos ou
collections



DIGITAL
INNOVATION
ONE

Schema Design



```
{
  id: ObjectId('AAA'),
  name: 'Kale Monster',
  ssn: '123-456-7890',
  addresses: [
    {
      street: '123 Sesame St',
      city: 'Anytown', cc: 'USA'
    },
    {
      street: '123 Avenue Q',
      city: 'New York',
      cc: 'USA'
    }
  ]
}
```

```
{
  name: 'left-handed smoke shifter',
  manufacturer: 'Acme Corp',
  catalog_number: 1234,
  parts: [
    ObjectId('AAAA'),
    ObjectId('BBBB')
  ]
}
```

```
{
  id: ObjectId('AAAA'),
  partno: '123-456-789',
  name: '44 sprocket',
  qty: 54,
  cost: 0.94,
  price: 3.99
}
```

```
{
  id: ObjectId('BBBB'),
  partno: '425-EFF-123',
  name: '16 Frontal',
  qty: 12,
  cost: 0.34,
  price: 7.99
}
```



DIGITAL
INNOVATION
ONE

Schema Design



Embedding:

Pros:

- Consulta informações em uma única query
- Atualiza o registro em uma única operação

Contras:

- Limite de 16 MB por documento

Schema Design

Referência:

Pros:

- Documentos pequenos
- Não duplica informações
- (Usado quando os dados não são acessados em todas as consultas)

Contras:

- Duas ou mais queries ou utilização do \$lookup

Schema Design

Recomendações de acordo com os relacionamentos:

One-to-one: prefira atributos chave-valor no documento

```
{
  "_id" : ObjectId("609aee3aa1a68ca982b47d7"),
  "name" : "Patrick Z",
  "street" : "Av das conchas",
  "number" : 102
}
```

Schema Design

Recomendações de acordo com os relacionamentos:

One-to-few: prefira embedding

```
{
  "_id" : ObjectId("609aee3aa1a68ca982b47d7"),
  "name" : "Patrick Z",
  "addresses": [
    { "street": "Av das conchas", "number": 102 },
    { "street": "Rua das perolas", "number": 100 }
  ]
}
```

Schema Design

Recomendações de acordo com os relacionamentos:

One-to-many e
Many-to-many: prefira
referência

```
{
  "_id" : ObjectId("609aee3aa1a68ca982b47d7"),
  "name" : "Patrick Z",
  "addresses": [
    ObjectId("123"), ObjectId("1234")
  ]
}

/** Address */
{
  "_id" : ObjectId("123"),
  "street" : "Av das conchas",
  "number" : 102
}

{
  "_id" : ObjectId("1234"),
  "street" : "Av das conchas",
  "number" : 102
}
```

Schema Design



<https://www.mongodb.com/blog/post/building-with-patterns-a-summary>

Boas Práticas

Evite documentos muito grandes

Use nome campos objetivos e curtos

Analise as suas queries utilizando explain()

Atualize apenas os campos alterados

Evite negações em queries

Boas Práticas

Listas/Arrays dentro dos documentos não podem crescer sem limite

JSON vs BSON

BSON:

- É uma serialização codificada em binário de documentos semelhantes a JSON.
- Contém extensões que permitem a representação de tipos de dados que não fazem parte da especificação JSON. Por exemplo, BSON tem um tipo Date, ObjectID

Agregações

- Agregação é o procedimento de processar dados em uma ou mais etapas, onde o resultado de cada etapa é utilizado na etapa seguinte, de modo a retornar um resultado combinado.

Agregações

■ Operadores de comparação:

- > - Maior que. O equivalente no Mongo é o operador \$gt
- < - Menor que. O equivalente no Mongo é o operador \$lt
- <> - Diferente de. O equivalente no Mongo é o operador \$ne
- = - Igual. No Mongo, é o operador \$eq
- <= - Menor ou igual. No Mongo é o operador \$lte
- >= Maior ou igual. No Mongo é o operador \$gte