

Disciplina:

BANCOS DE DADOS NoSQL

Professor: Augusto Zadra



2.4 MongoDB Operações

TRABALHANDO COM O MONGODB

- O ***documento*** é a unidade básica de dados para MongoDB e é equivalente a uma linha em um sistema de gerenciamento de banco de dados relacional.
- Uma ***coleção*** pode ser considerada uma tabela com um esquema dinâmico, já que não tem uma estrutura fixa.
- Uma única ***instância*** do MongoDB pode hospedar vários bancos de dados independentes, cada um contendo suas próprias coleções.

TRABALHANDO COM O MONGODB

- O MongoDB é distribuído com uma ferramenta simples, mas poderosa, chamada shell mongo.
- O shell MongoDB fornece suporte integrado para administrar instâncias e manipular os dados usando a linguagem de consulta própria.
- É também um interpretador **JavaScript** totalmente funcional que permite aos usuários criar e carregar seus próprios scripts para uma variedade de propósitos.

Bancos de dados NoSQL

mongoDB

Documents

- *Documents* são o coração do MongoDB.
- Representam um conjunto ordenado de chaves com valores associados e varia de acordo com a linguagem de programação, mas a maioria das linguagens tem uma estrutura de dados que se ajusta naturalmente como um mapa, *hash* ou dicionário.
- Documentos são representados no formato JSON e geralmente seguem o padrão de codificação UTF-8.

Bancos de dados NoSQL

mongoDB

Documents

- As chaves em um documento são strings.
- Qualquer caractere UTF-8 é permitido em uma chave, com algumas exceções:
- As chaves não devem conter o caractere |0 (o caractere nulo) sendo que este caractere é usado para significar o fim de uma chave.
- . e \$ são considerados caracteres especiais utilizados somente em algumas situações.

Bancos de dados NoSQL

mongoDB

Documents

- O MongoDB faz distinção entre tipos e maiúsculas e minúsculas ou seja, ele é *case sensitive* para todos os apontamentos no documento.
- Por exemplo, estas definições são distintas:
 - ✓ {"contador": 5}
 - ✓ {"contador": "5"}
- Assim como estas definições:
 - ✓ {"contar": 5}
 - ✓ {"Contagem": 5}

Bancos de dados NoSQL

mongoDB

Documents

- Uma última coisa importante a observar é que os documentos no MongoDB não podem conter chaves duplicadas como por exemplo:

```
{  
  "turma": "Banco de dados SQL",  
  "saudação": "Bancos de dados NoSQL"  
}
```


Bancos de dados NoSQL

mongoDB

Collections

- A *collection* é um grupo de documentos.
- Se um documento for o análogo do MongoDB a uma linha em um banco de dados relacional, uma coleção pode ser considerada o análogo de uma tabela.

Bancos de dados NoSQL

mongoDB

Collections

- A *collection* é um grupo de documentos.
- Se um documento for o análogo do MongoDB a uma linha em um banco de dados relacional, uma coleção pode ser considerada o análogo de uma tabela.

Bancos de dados NoSQL

mongoDB

Schemas **Dinâmicos**

- ***Schemas* dinâmicos** armazenam as ***collections***.
- Esta propriedade habilita a possibilidade de que os documentos de uma única coleção possuam formatos (ou estruturas) diferentes.
 - ✓ {"saudação": "O galo ganhou!",
"visualizações": 3}
 - ✓ {"signoff": "Boa noite e boa sorte!"}

Bancos de dados NoSQL

mongoDB

Schemas
Dinâmicos

- Os documentos anteriores têm diferentes chaves, diferentes números de chaves e valores de diferentes tipos.
- Como qualquer documento pode ser colocado em qualquer coleção, muitas vezes surge a pergunta: “Por que precisamos de coleções separadas?”
- Sem a necessidade de esquemas separados para diferentes tipos de documentos, por que devemos usar mais de uma coleção?

Bancos de dados NoSQL

mongoDB

Schemas
Dinâmicos

**Cuidados com
o projeto**

- Manter diferentes tipos de documentos na mesma coleção pode ser um pesadelo para desenvolvedores e administradores.
- Os desenvolvedores precisam ter certeza de que cada consulta está retornando apenas documentos que aderem a um esquema específico ou que o código do aplicativo que executa uma consulta pode manipular documentos de formas diferentes.
- Se estamos procurando por postagens de *blog*, é um mais complicado eliminar documentos que contenham dados do autor.

Bancos de dados NoSQL

mongoDB

Schemas
Dinâmicos

**Cuidados com
o projeto**

- Manter diferentes tipos de documentos na mesma coleção pode ser um pesadelo para desenvolvedores e administradores.
- Os desenvolvedores precisam ter certeza de que cada consulta está retornando apenas documentos que aderem a um esquema específico ou que o código do aplicativo que executa uma consulta pode manipular documentos de formas diferentes.
- Se estamos procurando por postagens de *blog*, é um mais complicado eliminar documentos que contenham dados do autor.

Bancos de dados NoSQL

mongoDB

Schemas
Dinâmicos

**Cuidados com
o projeto**

- É muito mais rápido obter uma lista de coleções do que extrair uma lista dos tipos de documentos em uma coleção.
- Por exemplo, se tivéssemos um campo "tipo" em cada documento que especificava se o documento era um "nota fiscal", "anuncio" ou "página web", seria muito mais lento encontrar esses três valores em uma única coleção do que ter três coleções separadas e consultar a coleção correta.

Bancos de dados NoSQL

mongoDB

Schemas
Dinâmicos

**Cuidados com
o projeto**

- O agrupamento de documentos do mesmo tipo na mesma coleção permite a localização dos dados.
- Obter várias páginas de uma coleção contendo apenas páginas provavelmente exigirá menos buscas em disco do que obter as mesmas postagens de uma coleção contendo campeonato e anúncios.
- Colocando apenas documentos de um único tipo na mesma coleção, possibilitamos a criação de índices nas nossas coleções com mais eficiência.



Esta organização não é obrigatória mas é uma boa prática e existem bibliotecas para mapa objeto-documento em algumas linguagens.

Bancos de dados NoSQL

mongoDB

Documents

Suporte a dados

- **nulo:** o tipo de dados nulo é usado para representar o valor nulo, bem como um valor que não existe.

```
{"numeroConcurso" : null}
```

- **booleano:** o tipo booleano é usado para representar valores verdadeiros e falsos.

```
{"houveGanhador" : true}
```

- **number:** o padrão do shell oferece suporte a números de ponto flutuante de 64 bits NumberLong e NumberInt, que representam 4 bytes e 8 bytes, respectivamente.

```
{"Bola1" : 5}
```

Bancos de dados NoSQL

mongoDB

Documents

Suporte a dados

- **string:** o tipo de dados string representa a coleção de caracteres e o padrão do MongoDB oferece suporte à codificação de caracteres UTF- *.

```
{"nomeConcurso" : "LotoFácil"}
```

- **date:** o MongoDB armazena datas em milissegundos porém as informações de fuso horário não são salvas.

```
{"dataSorteio" : new Date(20211202201015003)}
```

- **Array:** Um conjunto ou lista de valores representando uma matriz.

```
{"Cidades" : ["São Paulo", "Rio de Janeiro", "Belo Horizonte"]}
```

Bancos de dados NoSQL

mongoDB

Documents

Suporte a dados

- **Documento:** estrutura semelhante a um documento do MongoDB, que também pode ser usada como uma chave ao invés de criar uma coleção separada.

```
{  
  "Primeiro Nome" : "Augusto",  
  "Ultimo Nome"   : "Zadra",  
  "Idade"         : 35,  
  "email"         : "az@email.com.br",  
  "endereço"      : [  
    {  
      "cidade": "Brasília",  
      "estado": "Brasília",  
      "país"  : "Brasil"  
    }  
  ]  
}
```

Bancos de dados NoSQL

mongoDB

Databases

- Os dados são armazenados em um banco de dados na forma de coleções.
- O database portanto é um contêiner para coleção, assim como nos bancos de dados SQL, onde o banco de dados é um contêiner para tabelas.

- Utilizamos o comando **use** para criar o banco de dados.

```
> use Loterias;
```

- Cada banco de dados possui seu próprio conjunto de arquivos no sistema de arquivos e podemos lista-los com o comando **show databases**.



Se o banco não possuir uma **collection** associada não será listado.



```
> show databases;
Loterias      0.000GB
SocialMidias  0.000GB
admin         0.000GB
config        0.000GB
escola        0.000GB
local         0.000GB
```

Bancos de dados NoSQL

mongoDB

Databases

- Uma boa prática é armazenar todos os dados de um único aplicativo no mesmo banco de dados.
- Os nomes dos bancos de dados podem ser qualquer string UTF-8, com as seguintes restrições:
- A string vazia ("") não é um nome de banco de dados válido que não pode conter nenhum destes caracteres: /, \, ., ", *, <, >, :, |, ?, \$, (Um único espaço) ou |0 (o caractere *null*).
- Os nomes dos bancos de dados não diferenciam maiúsculas de minúsculas e limitados a no máximo 64 bytes.
- Antes do uso do mecanismo de armazenamento [WiredTiger](#), os nomes dos bancos de dados se tornaram arquivos em seu sistema de arquivos mas não é mais o caso.

Bancos de dados NoSQL

mongoDB

Databases

- Há nomes de **databases** reservados que precisam de semântica especial para acesso:
- **admin:** desempenha um papel na autenticação e autorização e o acesso a este banco de dados é necessário para algumas operações administrativas.
- **local:** armazena dados específicos para um único servidor. Em conjuntos de réplicas, o **local** armazena dados usados no processo de replicação sendo que ele nunca é replicado.

Bancos de dados NoSQL

mongoDB

Collections

- A **collection** é um contêiner para documentos MongoDB e são equivalentes a tabelas SQL, que armazenam os dados em linhas.
- Recomenda-se que a **collection** deve armazenar apenas documentos relacionados.
- Para criar uma nova **collection** utilizamos o comando `db.createCollection`:

```
> db.createCollection("MegaSena");
{ "ok" : 1 }
```
- Caso já exista uma collection com o mesmo nome já criada retornará o erro:

```
> db.createCollection("LotoFacil");
{
  "ok" : 0,
  "errmsg" : "Collection already exists. NS: Loterias.LotoFacil",
  "code" : 48,
  "codeName" : "NamespaceExists"
}
```


Bancos de dados NoSQL

mongoDB

Collections

- Uma coleção é identificada por seu nome. Os nomes das coleções podem ser qualquer string UTF-8.
- A string vazia ("") não é um nome de coleção válido e os nomes de coleção não podem conter o caractere|0 (caractere *null*), porque isso delinea o final de um nome de coleção.
- Você não deve criar nenhuma coleção com nomes que começam com sistema exemplo: system.LotoMania pois o *database system* possui as informações de sistema.
- As *collections* criadas pelo usuário não devem conter o caractere reservado \$ em seus nomes.

Bancos de dados NoSQL

mongoDB

Collections

Subcollections

- Para ter uma melhor legibilidade do banco é preciso organizar os dados.
- Uma proposta para organizar coleções é usar subcoleções com espaço de nomes separadas por “.” como:

```
> db.createCollection("MegaSena.NumerosSorteados");  
{ "ok" : 1 }
```

```
> db.createCollection("MegaSena.Ganhadores");  
{ "ok" : 1 }
```

```
> db.createCollection("MegaSena.Soretios");  
{ "ok" : 1 }
```

- Não há vínculo entre as subcoleções e a organização é apenas para fins de organização. No entanto, se você for utilizar o **GRIDFS** para armazenar arquivos maiores ele utiliza esta estrutura para armazenar os metadados.

Bancos de dados NoSQL

mongoDB

Documents

- Documents possuem um campo especial denominado **_id** que é um número hexadecimal de 12 bytes que garante a exclusividade do documento.
- Ele é gerado pelo MongoDB se não for fornecido pelo desenvolvedor.
- **_id** também representa a chave primária para documentos MongoDB.

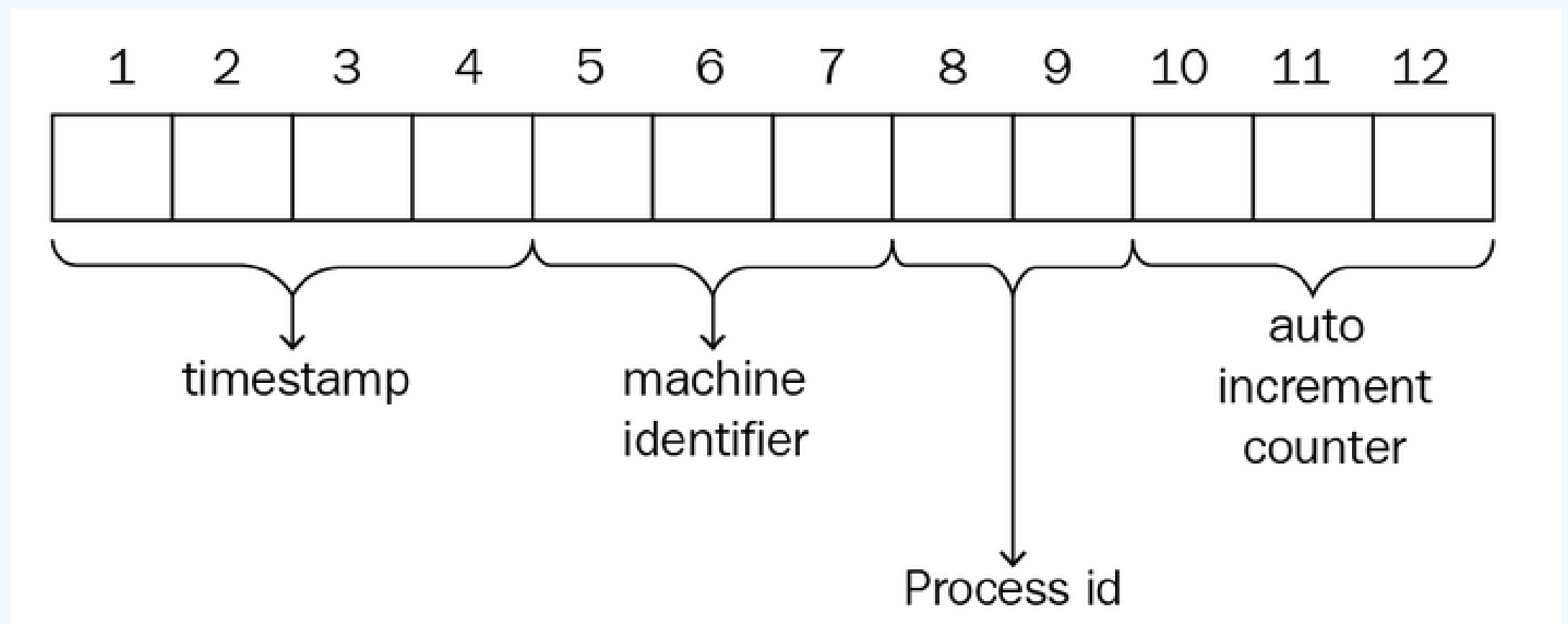
```
{
  "_id" : ObjectId("61aec07c5e2662d74e002e1e"),
  "NumeroSorteio" : 1,
  "DataSorteio" : "Mon Dec 06 2021 23:01:32 GMT-0300 (Hora oficial do Brasil)",
  "Acumulou" : "N",
  "QtdeGanhadores_15" : 5,
  "PremioPrincipal_R" : 49766.82
}
```

Bancos de dados NoSQL

mongoDB

Documents

- Desses 12 bytes, os primeiros 4 bytes representam o carimbo de data / hora atual, os próximos 3 bytes representam a ID da máquina, os próximos 2 bytes representam a ID do processo no servidor MongoDB e os 3 bytes restantes são valores de incremento automático simples.



Bancos de dados NoSQL

mongoDB

Limitações

- O tamanho máximo do documento suportado pelo é 16MB e o nível máximo de aninhamento de documentos compatível com o MongoDB é 100.
- O **namespace** máximo (banco de dados + nome da coleção) suportado pelo é de **123 caracteres** sendo que o nome do banco de dados é limitado a **64 caracteres**.
- Ao criar índice em qualquer campo, o valor desse campo não pode conter mais de 1MB e somente são permitidos 64 índices por coleção com no máximo 34 campos em índices compostos.

Bancos de dados NoSQL

mongoDB

Limitações

- Um índice com hash não pode ser exclusivo e somente 12 nós são permitidos em um conjunto de réplicas.
- São permitidos no máximo 512 bytes para chaves de shard.
- Você não pode fazer rollback automaticamente se os dados tiverem mais de 300 MB sendo que a intervenção manual é necessária em tais casos.

Bancos de dados NoSQL

mongoDB

shell

- O **shell** é um interpretador JavaScript completo, capaz de executar programas JavaScript.
- Pode-se fazer diversos testes desde cálculos e aplicação de funções Javascript.

```
> base = 25;
25
> Math.sqrt(base);
5
```

```
> angulo=360
360
> Math.sin(angulo);
0.9589157234143065
```

```
> x=200; y=100; z=300;
300
> x+y+z;
600
```

```
> function fatorial (n){
...     if (n<=1)
...         return 1
...     return n * fatorial (n-1);
... };
> fatorial (1)
1
> fatorial (5);
120
```

***Now, Let's go to pratice!
On MongoDB!***

REFERÊNCIAS BIBLIOGRÁFICAS

SANTOS, Hudson Leandro, 2020. Vamos falar um pouco do WiredTiger?. Disponível em: <<https://hudsondosantos.com/2020/08/18/vamos-falar-um-pouco-do-wiredtiger/>>. Acesso em: 25 Nov. 2021.

LI, Y.; **MANOHARAN**, S. A performance comparison of SQL and NoSQL databases. In: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing - PACRIM, 14., 2013, Victoria, B.C., Canadá. Proceedings... IEEE, 2013.

LÓSCIO, B. F.; **OLIVEIRA**, H. R. de;; **PONTES**, J. C. de S. NoSQL no desenvolvimento de aplicações web colaborativas. In: Simpósio Brasileiro de Sistemas Colaborativos – SBSC, 8., 2011, Paraty (RJ). Anais... Paraty: SBC, 2011.

VIEIRA, M. R. et al. Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data. In: Simpósio Brasileiro de Bancos de Dados, 27., 2012, São Paulo. Anais... São Paulo: SBC, 2012.