

INSTITUTO FEDERAL  
PIAUÍ  
Campus Parnaíba

# MongoDB

Prof. Msc Denival A. dos Santos

# Introdução

- MongoDB é um banco de dados de documentos, de código aberto.
- Um registro em MongoDB é um documento (estrutura de dados composta de pares de campos e valores).
- Os documentos do MongoDB são semelhantes aos objetos JSON.
- Os valores dos campos podem incluir outros documentos, matrizes e documentos.

```
{  
  field1 : valor1 ,  
  campo2 : valor2 ,  
  campo3 : valor3 ,  
  ...  
  campoN : valorN  
}
```

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value  
← field: value  
← field: value  
← field: value

# Introdução

- Vantagens de usar documentos:
  - Documentos (ou seja, objetos) correspondem a tipos de dado nativos em muitas linguagens de programação.
  - Documentos e matrizes incorporados reduzem a necessidade de junções caras.
  - Esquema dinâmico suporta polimorfismo fluente.

# Introdução

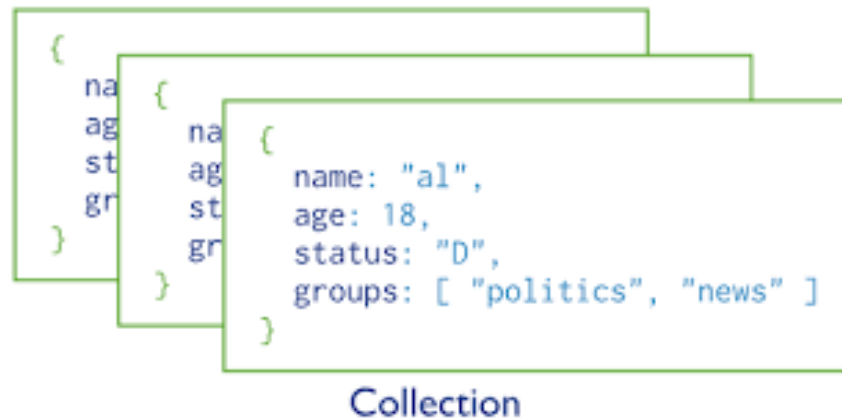
- Recursos principais:
  - Alto desempenho
  - Alta disponibilidade
  - Escalabilidade Horizontal
  - Linguagem de consulta avançada
  - Suporte para vários mecanismo de armazenamento

# MongoDB x SQL

SQL Terms/Concepts	MongoDB Terms/Concepts
database	<a href="#">database</a>
table	<a href="#">collection</a>
row	<a href="#">document</a> or <a href="#">BSON document</a>
column	<a href="#">field</a>
index	<a href="#">index</a>
table joins	<a href="#">\$lookup</a> , embedded documents
primary key	<a href="#">primary key</a>
Specify any unique column or column combination as primary key.	In MongoDB, the primary key is automatically set to the <a href="#">_id</a> field.
aggregation (e.g. group by)	<a href="#">aggregation pipeline</a>  <a href="#">See the SQL to Aggregation Mapping Chart.</a>
transactions	<a href="#">transactions</a>

# Banco de Dados e Coleções

- O MongoDB armazena registros de dados como documentos BSON.
- BSON é uma representação binária de documentos JSON , embora contenha mais tipos de dados do que JSON.
- Tabelas em MongoDB são chamadas de coleções.



# Mongo - Restrições

- O nome de um banco de dados não pode:
  - Conter os seguintes caracteres no UNIX: / \ . "\$
  - Conter os seguintes caracteres no WINDOWS: / \ . "\$ \* < > : | ?
  - Conter caracteres nulos ou ser vazio
  - Conter mais de 64 caracteres
- O nome de uma coleção não pode:
  - Conter o caracter \$ ou ser uma string vazia
  - Começar com o prefixo system. (ele é reservado para o sistema)
- O nome de um campo (chave) não pode:
  - Conter pontos (.) ou ser uma string vazia
  - Não pode começar com \$
- Cada documento JSON/BSON pode ter no máximo 16MB.

# Banco de Dados e Coleções

## ▪ Nome dos campo:

- O nome do campo `_id` é reservado para uso como chave primária, e o valor deve ser único na coleção, é imutável e pode ser de qualquer tipo diferente de uma matriz.
- O `_id` é sempre o primeiro campo no documento e o MongoDB gerará automaticamente um ObjectId para o campo `_id`.
- Utiliza a notação de ponto para acessar os elementos de uma matriz e acessar os campos de um documento incorporado.
- Para especificar ou acessar um campo de um documento incorporado com notação de ponto, concatene o nome do documento incorporado com o ponto (.) e o nome do campo e coloque entre aspas.

`“contato.telefone.tipo”:”celular”`

- Para especificar ou acessar um elemento de uma matriz pela posição de índice com base em zero, concatene o nome da matriz com a posição de índice com pontos e zero e coloque entre aspas.

`“vetor.2”:”valor”`



# Principais componentes do MongoDB

- O MongoDB é composto (basicamente) por:
  - **mongo**: cliente do MongoDB (Mongo Shell);
  - **mongod**: servidor do MongoDB;

## ▪ Conexão

- Caso o banco de dados não seja especificado na conexão, o mongo sempre se conectará ao banco de dados test.
- Além disso, por padrão, a autenticação do MongoDB não vem habilitada.

## ▪ Comando básicos

- **help** - exibe os principais comandos do Mongo Shell.
- **show dbs** - lista todos os BDs do servidor mongod.
- **db** - exibe o banco de dados em uso.
- **show collections** - lista todas as coleções de um BD.
- **use <db\_name>** - conecta ao BD especificado.

# API do MongoDB

- A API do MongoDB é baseada em JavaScript. Por isso, podemos executar scripts js no Mongo Shell.

```
for(i=0; i < 3; i++){  
  print ("hello, " + i);  
}
```

- Manipular dicionários (documentos JSON)

```
dict = {"a":1, "b":2};  
dict  
dict.a  
dict["a"]  
w = "a"  
dict[w]
```

# Mongo Shell

- Criando um banco e dados chamado “minha\_nova\_colecao”.

```
> use minha_colecao  
switched to db minha_colecao
```

- Para criar coleções, basta inserir um documento à nova coleção.

```
> db.minha.colecao.insert({nome:"Denival", idade:"40"})  
WriteResult({ "nInserted" : 1 })
```

- Ou criar apenas uma coleção.

```
> db.createCollection("log")  
{ "ok" : 1 }
```

- Para visualizar os documentos inseridos utilizamos o comando find().

```
> db.minha_colecao.find({})  
{ "_id" : ObjectId("5b9abd70fe9a1c26e1c6564b"), "nome" : "Denival", "idade" : "40" }
```

# Mongo Shell

- Para excluir uma coleção.

```
//exclui my_collection  
db.my_collection.drop();
```

- Para excluir um banco de dados.

```
use my_db;  
//exclui my_db  
db.dropDatabase();
```