



INSTITUTO FEDERAL  
PIAUÍ  
Campus Parnaíba

# MongoDB - CRUD

Prof. Msc Denival A. dos Santos

# Introdução

- Não existe uma linguagem separada para descrever as operações de CRUD no MongoDB.
- As operações existem como métodos/funções dentro da API.

CRUD	
Create	
Read	
Update	
Delete	

# Identificador

- Cada documento deve conter um **\_id** único associado, especificado pelo campo `_id`.
- Caso o usuário não especifique um valor para esse campo, ele é gerado automaticamente pelo MongoDB, definido como um `ObjectId()`.
- O MongoDB cria também um índice para o campo `_id`, a fim de tornar as consultas mais eficientes.
- Podemos especificar um `_id` (que será único na coleção)

```
_id: ObjectId("5c0d8149a8591e7b9a44f37d")  
field1 : "value1 "  
field2 : "value2 "
```

```
ObjectId  
String  
String
```

# Inserindo um documento

Existem duas operações de inserção no MongoDB.

- Inserção de um único documento:
  - **insertOne({}).**
- Inserção de múltiplos documentos de uma só vez:
  - **insertMany({}).**

# Inserindo um documento

**InsertOne({}).**

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,             ← field: value
  status: "pending"   ← field: value } document
}
```

)

```
INSERT INTO users      ← table
      ( name, age, status ) ← columns
VALUES      ( "sue", 26, "pending" ) ← values/row
```

# Inserindo muitos documentos

**InsertMany({}).**

```
db.users.insertMany(  
  [  
    {  
      name: "sue",  
      age: 26,  
      status: "pending"  
    },  
    {  
      name: "bob",  
      age: 25,  
      status: "enrolled"  
    },  
    {  
      name: "ann",  
      age: 28,  
      status: "enrolled"  
    }  
  ]  
)
```

← collection

← field: value  
← field: value  
← field: value } document

} document

} document

# Localizando documentos

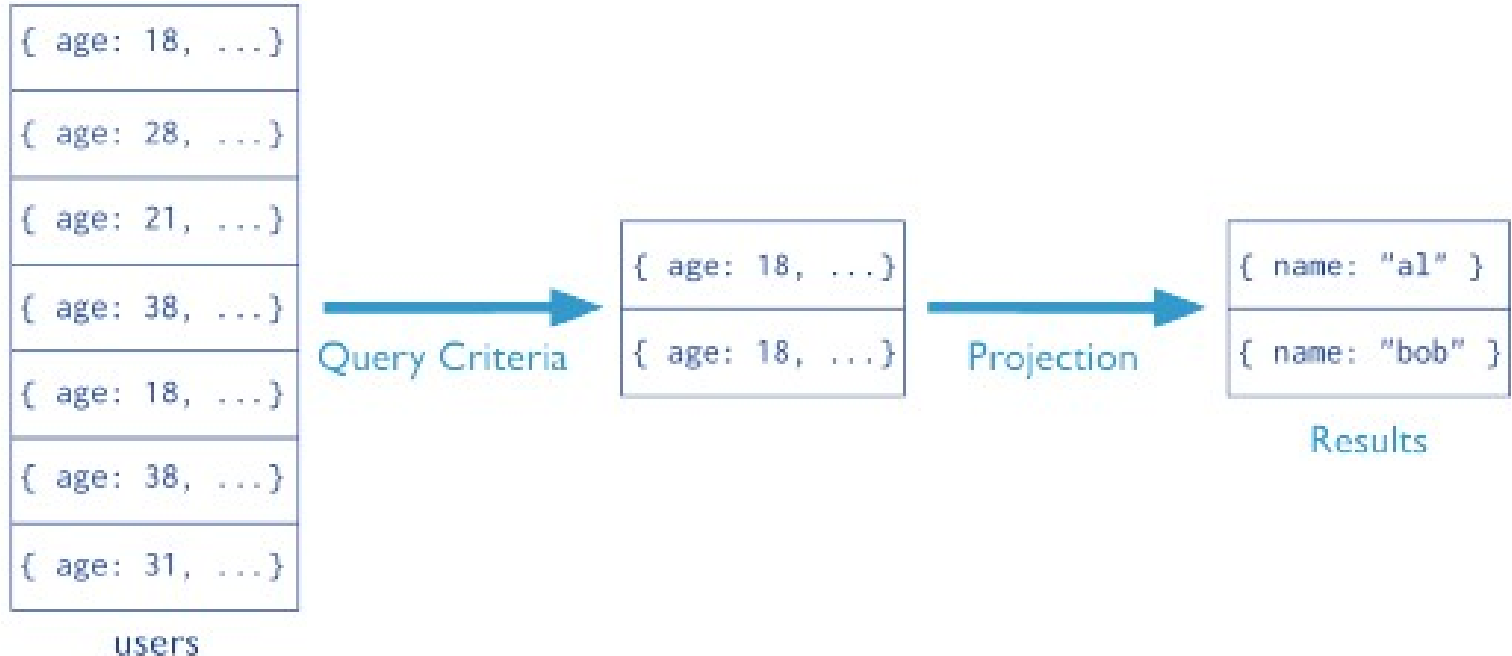
## Principais métodos para retornar informações de documentos.

- **Find()** - retorna todos os documentos que atendem aos critérios especificados.
- **FindOne()** - retorna um único documento que atende aos critérios especificados.
  - Observação: caso exista mais de um documento atendendo aos critérios, o método findOne() retorna apenas o primeiro.

# Localizando documentos

Collection      Query Criteria      Projection  
`db.users.find( { age: 18 }, { name: 1, _id: 0 } )`

**.find({}).**





# Localizando documentos

**.findOne({}).**

```
db.bios.findOne(  
  { },  
  { name: 1, contribs: 1 }  
)
```

# Modificar documentos

Os principais métodos para modificação de documentos são:

- **UpdateOne()** - atualiza um único documento que atende aos critérios especificados.
- **UpdateMany()** - atualiza todos os documentos que atende aos critérios especificados.

# Modificar documentos

`.updateOne({}).`

```
db.carro.updateOne(  
  {modelo: "Siena"},  
  {$set: {modelo: "Grand Siena"}}  
)
```

# Modificar documentos

## **.updateMany({}).**


```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "Reject" } }  
)
```



← collection  
← update filter  
← update action

The following diagram shows the same query in SQL:

```
UPDATE users  
SET    status = 'Enrolled'  
WHERE  age < 18
```



← table  
← update action  
← update filter

# Modificar documentos

```
db.pessoa.insertOne({
  _id : 1,
  nome : "José",
  idade : 30,
  altura : 1.70,
  time : "Vasco"
})
```

**\$unset - utilizado para excluir um campo específico.**

```
db.pessoa.updateOne(
  {_id:1},
  {$unset : {time:1}})
```

# Deletar documentos

Os principais métodos para modificação de documentos são:

- **deleteOne()** - remove um único documento que atende aos critérios especificados.
- **deleteMany()** - remove todos os documento que atende aos critérios especificados.

# Remover um documento

**.deleteOne({}).**

```
db.users.deleteOne(  
  { status: "Rejected" }  
)
```



← collection  
← delete filter

```
DELETE FROM users  
WHERE      status = 'reject'  
LIMIT     1
```



← table  
← delete filter  
← delete limit

# Remover muitos documentos

**.deleteMany({}).**

```
db.users.deleteMany(  
  { status: "Rejected" }  
)
```



← collection

← delete filter

```
DELETE FROM users  
WHERE      status = 'reject'
```



← table

← delete filter