

# Filter Operators 📅 (03/11/2022)



## Objetivo da aula

Expandir nosso conhecimento sobre queries do MongoDB e principais operadores de consulta

---

🍿 Nosso Contexto...

A TrybeFlix 🎬



Imagine o seguinte cenário:

A *TrybeFlix* é uma grande empresa do setor de entretenimento. Atuando com uma plataforma de streaming de filmes e fornece um catálogo vasto para seus assinantes.

Nós somos as pessoas desenvolvedoras responsáveis por elaborar consultas que entreguem os resultados esperados às pessoas utilizadoras da plataforma.

## Iniciando nosso Projeto com Docker

Como na aula anterior, usaremos uma imagem do MongoDB em um container Docker e nos conectaremos com a respectiva porta utilizando a extensão **MongoDB for VSCode** para utilizarmos o **Playground**



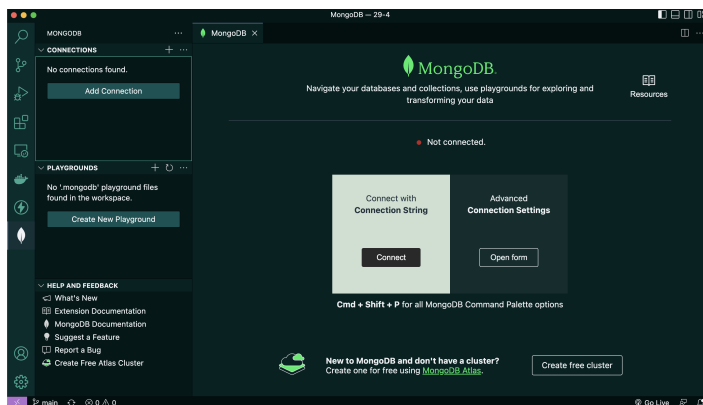
```
docker run -d -p 27017:27017 --name trybeFlix mongo:latest
```

## Carregando nossa base

Vamos realizar uma cópia da base de dados disponível no Gist abaixo:

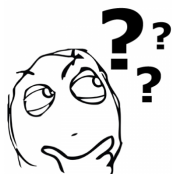
```
[url](https://gist.githubusercontent.com/isaacbatst/09db7be562c87a9528e8665174e6f5e1/raw/23f4c456d627bb66a7d0a00a18e326fe02855f20/aula_29_2
```

1. Crie um arquivo chamado `aula_11_2.mongodb` e cole os dados
2. Realize a conexão com o MongoDB utilizando a extensão MongoDB for VSCode
3. Execute o arquivo para carregar os dados



## O Problema de Negócio

Agora que temos nosso setup pronto, algumas problemas nos foram informados pela pessoa de negócios. Precisamos realizar consultas que façam comparações com os valores. Por exemplo: Quais filmes ocorreram após o ano de 1993?



## Operadores de Comparação



### Conhecendo os operadores de Comparação!

Operadores `$lt` e `$lte`

#### Sintaxe Base:

```
{ field: { $operator: value }}
```

#### Atividade:

- ☐ Exibir filmes com ano menor que 2000
- ☐ Exibir filmes com ano menor ou igual a 2000

Operadores `$gt` e `$gte`

#### Sintaxe Base:

```
{ field: { $operator: value }}
```

#### Atividade:

- ☐ Exibir filmes com ano maior que 1993
- ☐ Exibir filmes com ano maior ou igual a 1993

---

## Combinando Operadores

### Sintaxe Base:

```
{ field: { $operator: value, $operator: value }}
```

- ☐ Exibir filmes com ano entre 1993 e 2000 (inclusive)
- 

Operadores `$eq` e `$ne`

```
{ field: { $operator: value }}
```

### Atividade:

- ☐ Exibir filmes com ano 1993
- ☐ Exibir filmes com avaliação bom igual a 7
- ☐ Exibir filmes que não tenham avaliação ruim igual a 8
- 

## Operando em Arrays

Operadores `$in` e `$nin`

### Sintaxe Base:

```
{ field: { $operator: [value1, value2, value3...] }}
```

### Atividade:

- ☐ Exibir filmes com os anos 2001, 1968 e 1995
- ☐ Exibir filmes que não estejam nos anos 2013 e 2008
- 

## Um Novo Problema de Negócios

Agora a pessoa de negócios precisa de algumas informações que só conseguiremos obter utilizando queries mais complexas. Precisamos combinar expressões e os critérios precisam ser mais elaborados.

# Operadores Lógicos



Operador **\$and**

**Sintaxe Base:**

```
{ $and: [{ field: {$operator: value}}, {field: {$operator: value}}] }  
{ $and: [{ field: value}, {field: value }] }
```

**Atividade:**

- ☐ Exiba os documentos em que o campo **ano** seja maior que 2000 e o campo **bom** de avaliação seja maior ou igual a 8.

Operador **\$or**

**Sintaxe Base:**

```
{ $or: [{ field: {$operator: value}}, {field: {$operator: value}}] }  
{ $or: [{ field: value}, {field: value }] }
```

**Atividade:**

- ☐ Exiba os documentos em que o campo **ano** seja maior que 2000 ou o campo **bom** de avaliação seja maior ou igual a 8.

Operador **\$not**

**Sintaxe Base:**

```
{ field: { $not: { $operator: value } } }
```

**Atividade:**

- ☐ Exiba os documentos sem o diretor Michael Bay

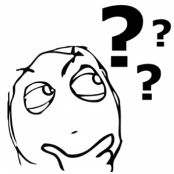
Operador **\$nor**

#### Sintaxe Base:

```
{ $nor: [ { field: value }, { field: value } ] }  
  
{ $nor: [ { field: { $operator: value } }, { field: { $operator: value } } ] }
```

#### Atividade:

- ☐ Exiba os documentos sem o diretor Michael Bay, nem avaliação “ruim” igual a 10
- ☐ Exiba os documentos sem o diretor Michael Bay, nem avaliação “bom” menor que 7
- ☐ Exiba os documentos sem ano maior que 2000, nem avaliação “ruim” maior que 6



Surgiu um problema... precisamos verificar se um dado campo existe ou não nos documentos. hmmm Schemaless...

## Operador de Elementos

O operador **\$exists**

Vamos carregar uma nova coleção para teste

```
db.records.insertMany(  
  [  
    { primeiro: 5, segundo: 5, terceiro: null },  
    { primeiro: 3, segundo: null, terceiro: 8 },  
    { primeiro: null, segundo: 3, terceiro: 9 },  
    { primeiro: 1, segundo: 2, terceiro: 3 },  
    { primeiro: 2, terceiro: 5 },  
    { primeiro: 3, segundo: 2 },  
    { primeiro: 4 },  
    { segundo: 2, terceiro: 4 },  
    { segundo: 2 },  
    { terceiro: 6 }  
  ]  
);
```

#### Atividade:

- ☐ Exiba todos os documentos que contenham o campo “primeiro”, mesmo que tenha o valor null

- ☐ Exiba todos os documentos que não contenham o campo "segundo"
- 

## Desafio

Recarregue a base de dados de filmes e realize os seguintes exercícios:

- ☐ Trazer os filmes onde o diretor é `Quentin Tarantino` e `avaliacao.bom` esteja entre 5 e 10.
- ☐ Trazer os filmes onde o diretor é `Quentin Tarantino` ou `avaliacao.bom` esteja entre 5 e 10.
- ☐ Trazer os filmes onde do ano 2014 e `avaliacao.ruim` **\*\*não\*\*** esteja entre 5 e 10.
- ☐ Trazer os filmes onde o ano seja 2001, 1968 e 1995
- ☐ Trazer os filmes onde o ano não seja 2013 e 2008
- ☐ Filtre os filmes do `Steven Spielberg` que nem o nome seja `Indiana Jones e a Última Cruzada` nem o ano seja 1989. (Usando o operador \$nor)

## O que aprendemos?

Aprendemos sobre algumas das principais queries e operadores para atualização de dados no MongoDB como:

- Operadores de Comparação
- Operadores Lógicos
- Operadores de Elementos