

DB NoSQL – Atividade 01

Aluna: Barbara Ramos Alves

1 - Faça uma pesquisa e explique as diferenças dos comandos(Pesquise também os operadores que podem ser utilizados para buscas):

- **db.users.insertOne e db.users.insertMany:**

Diferenças	insertOne	insertMany
Número de documentos	Insere um documento por vez.	Insere vários documentos de uma só vez.
Execução	Realiza a inserção de um único documento.	Realiza a inserção de vários documentos em uma única operação.
Ordenação	Não aplicável.	Pode ser ordenada (ordered: true) ou desordenada (ordered: false). Com ordered: false, tenta inserir todos os documentos, mesmo em caso de erro.
Retorno	Retorna o insertedId do documento inserido.	Retorna um array com os insertedIds dos documentos inseridos.

- **db.users.updateOne, db.users.updateMany e db.users.replaceOne:**

Comando	Descrição	Quando usar
db.users.updateOne()	Atualiza um único documento que corresponde ao filtro	Quando você deseja modificar apenas um documento em uma coleção, mesmo que vários correspondam ao filtro.
db.users.updateMany()	Atualiza todos os documentos que correspondem ao filtro	Quando você deseja modificar múltiplos documentos em uma coleção de uma só vez.
db.users.replaceOne()	Substitui completamente um documento	Quando você precisa substituir um documento inteiro, ao invés de atualizar apenas campos específicos.

O comando replaceOne() é usado para **substituir completamente** um documento existente por outro. Ao contrário de updateOne e updateMany, que alteram apenas campos específicos, o replaceOne substitui o documento inteiro.

- **db.users.deleteOne e db.users.deleteMany:**

Comando	Descrição	Quando usar
db.users.deleteOne()	Exclui apenas o primeiro documento que corresponde ao filtro	Quando você deseja remover apenas um documento específico, mesmo que outros correspondam ao filtro.
db.users.deleteMany()	Exclui todos os documentos que correspondem ao filtro	Quando você precisa remover vários documentos de uma só vez, com base em um critério de filtro.

Operadores de consulta: Neste [Link](#) estão alguns dos operadores segundo a documentação do Mongo BD.

2 - Vamos usar a coleção chamada filmes, que contém documentos representando filmes com seus títulos, diretores, anos de lançamento, e notas de avaliação.

Atividades:

1 - Busque todos os filmes dirigidos por "Christopher Nolan".

```
db.filmes.find({ diretor: "Christopher Nolan" }).pretty()
```

2 - Busque filmes lançados antes do ano 2000.

```
db.filmes.find({ ano: { $lt: 2000 } }).pretty()
```

Utilizando o operador \$lt (menor que) para filtrar os filmes cujo campo ano é anterior a 2000.

3 - Busque os filmes com nota maior ou igual a 8.6, mostrando apenas o título e a nota dos filmes.

```
db.filmes.find(  
  
  { nota: { $gte: 8.6 } },  
  
  { _id: 0, titulo: 1, nota: 1 }  
  
)
```

O operador \$gte (maior ou igual a) é para buscar filmes com nota maior ou igual a 8.6, e o segundo parâmetro é utilizado para projetar apenas o título e a nota, excluindo o _id.

4 - O filme "Parasite" recebeu uma nova avaliação. Atualize a nota de 8.6 para 9.0.

```
db.filmes.updateOne( { titulo: "Parasite" }, { $set: { nota: 9.0 } })
```

O operador \$set para atualizar o campo nota do filme com o título "Parasite".

5 - Atualize todos os filmes dirigidos por "Christopher Nolan" para incluir uma nova chave categoria com o valor "Ficção Científica".

```
db.filmes.updateMany(  
  { diretor: "Christopher Nolan" },  
  { $set: { categoria: "Ficção Científica" } }  
)
```

O comando updateMany foi utilizado para atualizar todos os documentos onde o campo diretor é "Christopher Nolan", adicionando o campo categoria com o valor "Ficção Científica".

6 - Remova todos os filmes com nota menor que 8.7.

```
db.filmes.deleteMany({ nota: { $lt: 8.7 } })
```

O operador \$lt para encontrar todos os filmes cuja nota é menor que 8.7 e remove todos esses documentos com deleteMany.