Consulta com Operadores

Profa. Dra. Kelly Lais Wiggers

Operadores de Comparação

- $eq (igual): { nome: { $eq: "João" } }

- $ne (diferente): { nome: { $ne: "João" } }

- $gt (maior que): { idade: { $gt: 25 } }

- $lt (menor que): { idade: { $lt: 25 } }

- $gte (maior ou igual): { idade: { $gte: 25 } }

- $lte (menor ou igual): { idade: { $lte: 25 } }

Alguns exemplos

- Encontrar todos os documentos com nome "João"

db.pessoas.find({ nome: "Joao"})

```
meubanco> db.pessoas.find({ "nome": "Joao"})
[
  {
    _id: ObjectId('672a9cefe12f303b05fe6911'),
    nome: 'Joao',
    idade: '27'
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com nome "João" e idade maior que 25:

db.pessoas.find({ “nome”: "Joao", “idade”: { $gt: “25” } })

```
meubanco> db.pessoas.find({ "nome": "Joao", "idade": { $gt: "25" } })
[
  {
    _id: ObjectId('672a9cefe12f303b05fe6911'),
    nome: 'Joao',
    idade: '27'
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com idade maior que 25 e menor que 40:

db.pessoas.find({ “idade”: { $gt: “25”, $lt: “40” } })

Inserindo uma pessoa com data

```
meubanco> db.pessoas.insertOne({
... "nome": "Aluno",
... "idade": "29",
... "dataNascimento": ISODate("1995-11-10T00:00:00.000Z")})
{
  acknowledged: true,
  insertedId: ObjectId('672a9f1ae12f303b05fe6912')
}
```

```
db.pessoas.insertOne({ "nome": "Aluno de tal", "idade": "39", "dataNascimento": new Date("1985-12-10") })
```

Alguns exemplos

- Encontrar todos os documentos com data de nascimento maior que 01/01/1990:

db.pessoas.find({ dataNascimento: { $gt: ISODate("1990-01-01T00:00:00.000Z") } })

```
meubanco> db.pessoas.find({ dataNascimento: { $gt: ISODate("1990-01-01T00:00:00.
000Z") } })
[
  {
    _id: ObjectId('672a9f1ae12f303b05fe6912'),
    nome: 'Aluno',
    idade: '29',
    dataNascimento: ISODate('1995-11-10T00:00:00.000Z')
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com valor de conta maior ou igual a 1000 e menor ou igual a 5000:

db.contas.find({ “valor”: { $gte: “1000”, $lte: “5000” } })

Operadores lógicos

- $and (e): { $and: [{ idade: { $gt: 25 } }, { nome: { $eq: "João" } }] }

- $or (ou): { $or: [{ idade: { $gt: 25 } }, { nome: { $eq: "João" } }] }

- $not (não): { idade: { $not: { $gt: 25 } } }

Alguns exemplos

-Encontrar todos os documentos com idade maior que 25 **ou** nome "João":

db.pessoas.find({ $or: [{ idade: { $gt: 25 } }, { nome: "João" }] })

- Encontrar todos os documentos com idade maior que 25 **e** nome "João":

db.pessoas.find({ $and: [{ idade: { $gt: 25 } }, { nome: "João" }] })

Alguns exemplos

-Encontrar todos os documentos com idade maior que 25 **ou** nome "João":

db.pessoas.find({ $or: [{ idade: { $gt: 25 } }, { nome: "João" }] })

```
meubanco> db.pessoas.find({ $or: [{ "idade": { $gt:"25" } }, { "nome": "Joao" }]
 })
[
  {
    _id: ObjectId('672a9cefe12f303b05fe6911'),
    nome: 'Joao',
    idade: '27'
  },
  {
    _id: ObjectId('672a9f1ae12f303b05fe6912'),
    nome: 'Aluno',
    idade: '29',
    dataNascimento: ISODate('1995-11-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672a9f9ae12f303b05fe6913'),
    nome: 'Aluno de tal',
    idade: '39',
    dataNascimento: ISODate('1985-12-10T00:00:00.000Z')
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com idade maior que 25 e (nome "João" ou cidade "São Paulo"):

db.pessoas.find({ “idade”: { $gt: “25” }, $or: [{“nome”: "Joao" }, { “cidade”: "São Paulo" }] })

Operadores de Array

- $in (contém): { interesses: { $in: ["leitura", "cinema"] } }

- $nin (não contém): { interesses: { $nin: ["leitura", "cinema"] } }

- $all (contém todos): { interesses: { $all: ["leitura", "cinema", "música"] } }

- $elemMatch (contém elemento): verifica se um elemento está presente em um array

- $size (tamanho): verifica o tamanho de um array

Alguns exemplos

Fiz uma inserção para simular:

meubanco> db.pessoas.insertOne({ "nome": "Kelly", "idade": "33", "interesses":["esporte", "musica"]})

- Encontrar todos os documentos que possuem interesse em esporte e musica

```
meubanco> db.pessoas.find({ interesses: { $in: ["esporte", "musica"] } })
[
  {
    _id: ObjectId('672aa330e12f303b05fe6914'),
    nome: 'Kelly',
    idade: '33',
    interesses: [ 'esporte', 'musica' ]
  }
]
```

Alguns exemplos

- Encontrar todos os documentos sem interesse "esportes":

db.pessoas.find({ “interesses”: { $nin: ["esportes"] } })

- Encontrar todos os documentos com todos os interesses "leitura", "cinema" e "música":

db.pessoas.find({ “interesses”: { $all: ["leitura", "cinema", "música"] } })

Alguns exemplos

- Encontrar todos os documentos com um interesse que contenha "leitura":

db.pessoas.find({ “interesses”: { $elemMatch: { $eq: "musica" } } })

- Encontrar todos os documentos que notas contenha 94

```
meubanco> db.pessoas.find({ "notas": { $elemMatch: { $eq: 94 } } })
[
  {
    _id: ObjectId('6718d0ca3857bf08b2fe6911'),
    nome: 'Kelly Lais Wiggers',
    email: 'kwiggers@utfpr.edu.br',
    notas: [ 94 ],
    interesses: [ 'esportes' ]
  },
  {
    _id: ObjectId('672369eb2de66b0a2cfe6912'),
    nome: 'Aluno',
    email: 'email@email.com',
    notas: [ 44, 78, 38, 80, 94 ]
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com 3 interesses:

db.pessoas.find({ “interesses”: { $size: 3 } })

Operadores de String

- $regex (expressão regular): { nome: { $regex: /^J/ } }

- $options (opções de expressão regular): { nome: { $regex: /^J/, $options: "i" } }

- $text (busca textual): busca strings em campos de texto

Alguns exemplos

- Encontrar todos os documentos com nome que começa com "Jo":

db.pessoas.find({ nome: { $regex: /^Jo/ } })

```
meubanco> db.pessoas.find({ nome: { $regex: /^Jo/ } })
[
  {
    _id: ObjectId('672a9cefe12f303b05fe6911'),
    nome: 'Joao',
    idade: '27'
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com nome que contém "ano" (ignorando maiúsculas e minúsculas):

db.pessoas.find({ nome: { $regex: /ano/, $options: 'i' } })

Alguns exemplos

- Encontra
maiúscula

db.pesso

```
meubanco> db.pessoas.find({ nome: { $regex: /ano/, $options:'i'} })
[
  {
    _id: ObjectId('6718d1843857bf08b2fe6913'),
    nome: 'Fulano de Tal',
    email: 'fulano@email.com'
  },
  {
    _id: ObjectId('6718d2523857bf08b2fe6914'),
    nome: 'Beltrano',
    email: 'beltrano@email.com'
  },
  {
    _id: ObjectId('6718d2523857bf08b2fe6915'),
    nome: 'Ciclano',
    email: 'ciclano@gmail.com'
  }
]
```

Alguns exemplos

- Encontrar todos os documentos com descrição que contém a palavra "MongoDB":

```
meubanco> db.pessoas.createIndex({"descricao":"text"})
descricao_text
meubanco> db.pessoas.find({$text:{$search: "MongoDB"}})
[
  {
    _id: ObjectId('672aa5ace12f303b05fe6915'),
    nome: 'ABC',
    descricao: 'Nesse texto tem MongoDB'
  }
]
```