

Realizando Operações em um Banco de Dados NoSQL



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Agenda 13/10



- Recapitulação dos Conteúdos Anteriores
- Realizando Consultas no Banco de Dados
- Realizando Atualizações no Banco de Dados
- Realizando Remoções no Banco de Dados
- Indo Além

Kahoot! (Aula Passada)



Kahoot!

Recapitulação de Conteúdos



SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document
column	field
primary key	primary key

In MongoDB, the primary key is automatically set to the `_id`

Documentação Mongo DB



Selecionando **Todos** os Documentos e **Todos** Campos



```
SELECT *  
FROM people
```

```
db.people.find()
```

```
db.alunos.find(  
)
```

Selecionando **Alguns** Campos em **Todos** os Documentos



```
SELECT id,  
       user_id,  
       status  
FROM people
```

```
db.people.find(  
  { },  
  { user_id: 1, status: 1 }  
)
```

```
FROM beobje
```

```
db.alunos.find({}, {"nome":1}  
)
```

Selecionando Documentos com Campos Iguais (=)

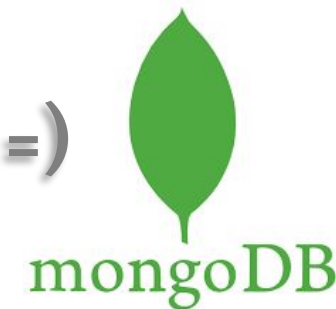


```
SELECT *  
FROM people  
WHERE status = "A"
```

```
db.people.find(  
  { status: "A" }  
)
```

```
db.alunos.find({"sexo":"F"  
})
```


Selecionando Documentos com **Campos Diferentes (!=)**



```
SELECT *  
FROM people  
WHERE status != "A"
```

```
db.people.find(  
  { status: { $ne: "A" } }  
)
```

```
db.alunos.find({"sexo":{"$ne":"F"}})
```

Selecionando Documentos com Campos Maiores Que (>)

```
SELECT *  
FROM people  
WHERE age > 25
```

```
db.people.find(  
  { age: { $gt: 25 } }  
)
```

```
db.alunos.find({"idade":{$gt:35}}  
)
```

Selecionando Documentos com Campos Menores Que (<)

```
SELECT *  
FROM people  
WHERE age < 25
```

```
db.people.find(  
  { age: { $lt: 25 } }  
)
```

```
db.alunos.find({"idade":{$lt:25}})
```

Selecionando Documentos com **Campos Maiores ou Iguais Que (>=)**

```
SELECT *  
FROM people  
WHERE age >= 25
```

```
db.people.find(  
  { age: { $gte : 25 } }  
)
```

```
db.alunos.find({"idade":{$gte:35}})
```

Selecionando Documentos com **Campos Menores ou Iguais Que (<=)**

```
SELECT *  
FROM people  
WHERE age <= 25
```

```
db.people.find(  
  { age: { $lte: 25 } }  
)
```

```
MONGODB 986 <= 32
```

```
db.alunos.find({"idade":{$lte:25}})
```

Atividade



Selecione Todos os Alunos por Nome e Idade
(Que Possuam Idade \geq 30 Anos)



Atividade



Selecione Todos os Alunos por Nome e Idade
(Que Sejam Homens)



?

Perguntas

Selecionando Documentos de forma **combinada** com **E**

```
SELECT *  
FROM people  
WHERE status = "A"  
AND age = 50
```

```
db.people.find(  
  { status: "A",  
    age: 50 }  
)
```

```
AND age = 20
```

```
db.alunos.find({"sexo":"F",trabalha:true})
```

Selecionando Documentos de forma **combinada** com E

```
SELECT *  
FROM people  
WHERE age > 25  
AND age <= 50
```

```
db.people.find(  
  { age: { $gt: 25, $lte: 50 } }  
)
```

```
db.alunos.find({"idade":{$gt:20,  
$lte:25}})
```

Selecionando Documentos de forma **combinada** com **OU**

```
SELECT *  
FROM people  
WHERE status = "A"  
OR age = 50
```

```
db.people.find(  
  { $or: [ { status: "A" } , { age: 50 } ]  
}
```

```
db.alunos.find({$or:[{"sexo":"M"}, {"trabalha":false}  
]})
```

Atividade



Selecione Todos os Alunos por Nome e Idade
(Que Sejam Homens com Idade \geq 30 Anos)



?

Perguntas

Selecionando Documentos com Ordenação Crescente

```
SELECT *  
FROM people  
WHERE status = "A"  
ORDER BY user_id ASC
```

```
db.people.find( { status: "A" } ).sort( { user_id: 1 } )
```

```
ORDER BY user_id ASC
```

```
db.alunos.find({"sexo":"M"}).sort({"idade":+1  
})
```

Selecionando Documentos com **Ordenação Decrescente**

```
SELECT *  
FROM people  
WHERE status = "A"  
ORDER BY user_id DESC
```

```
db.people.find( { status: "A" } ).sort( { user_id: -1 } )
```

```
ORDER BY user_id DESC
```

```
db.alunos.find({"sexo":"M"}).sort({"idade":-1})
```

Atividade



Selecione Todos os Alunos por Nome e Idade
(Onde os Nomes Devem Ser Ordenados de forma Crescente)



?

Perguntas

Selecionando Documentos com **Limite** de **Resultados**

```
SELECT *  
FROM people  
LIMIT 1
```

```
db.people.find().limit(1)
```

```
db.alunos.find().limit(  
2)
```

Selecionando o **Menor Valor Encontrado**

```
SELECT MAX(age)  
FROM people
```

```
db.people.find().sort({age :+1}).limit(1)
```

```
db.alunos.find().sort({"idade":+1}).limit(  
1)
```

Selecionando o **Maior Valor Encontrado**

```
SELECT MIN(age)  
FROM people
```

```
db.people.find().sort({age:-1}).limit(1)
```

```
db.alunos.find().sort({"idade":-1}).limit(  
1)
```

Atividade



Selecione Todos Os Alunos por Nome e Idade
(Retorne o Aluno que Possuir a Menor Idade)



?

Perguntas

Contando a Quantidade de Documentos Encontrados (**COUNT**)



```
SELECT COUNT(*)  
FROM people
```

```
db.people.find().count()
```

```
db.alunos.find().count(  
)
```

Contando a Quantidade de Documentos Encontrados (COUNT)



```
SELECT COUNT(user_id)
FROM people
```

```
db.people.find( { user_id: { $exists: true } } ).count()
```

```
db.alunos.find({"trabalha:true":{"$exists:true"}}).count()
```


Contando a Quantidade de Documentos Encontrados (COUNT)



```
SELECT COUNT(*)  
FROM people  
WHERE age > 30
```

```
db.people.find( { age: { $gt: 30 } } ).count()
```

```
db.alunos.find({"sexo":"F"}).count  
( )
```

Calculando a Média de um Campo dos Documentos (AVG)



```
SELECT AVG (salary)  
FROM people
```

```
db.people.aggregate([{$group: {_id: null, media_salario: {$avg: "$salary"}}}])
```

```
db.alunos.aggregate([{$group: {_id: null, soma: {$avg: "$idade"}}}  
])
```

Realizando a **Soma** de um **Campo** dos Documentos (**SUM**)



```
SELECT SUM (salary)
FROM people
```

```
db.people.aggregate([{$group: {_id: null, soma_salario: {$sum: "$salary"}}}])
```

```
db.alunos.aggregate([{$group: {_id: null, soma: {$sum: "$idade"}}}])
```

Atividade



Selecione Todos os Alunos
(Calcule a Quantidade de Alunos do Sexo Masculino)



?

Perguntas

Selecionando Documentos com BETWEEN

```
SELECT *  
FROM people  
WHERE salary BETWEEN 3000 AND 6000
```

```
db.pessoa.find({salary: { $gte : 3000, $lte : 6000}}); )
```

```
db.alunos.find({"idade":  
{$gte:35,$lte:40}})
```

Selecionando **Todos** os Documentos que **Iniciam** por **Algo (Like)**

```
SELECT *  
FROM people  
WHERE user_id like "bc%"
```

```
db.people.find( { user_id: { $regex: /bc/ } }
```

```
MHEKE 026L~Jq fJK6 „PC&„
```

```
db.alunos.find({"nome":{"regex:/Ju/}})
```

Selecionando Documentos com IN

```
SELECT*  
FROM people  
WHERE job IN ("Teacher", Developer)
```

```
db.pessoa.find({ genero: {$in:["Teacher", "Developer"]}})
```

```
db.alunos.find({"cidade":{$in:["Curitiba","Salvador"]  
}})
```


Atividade



Selecione Todos os Alunos por Nome e Idade
(Traga Apenas Aqueles Residentes em Florianópolis)



?

Perguntas

Atualizando Documentos



```
db.collection.update(query, update)
```

Atualizando Documentos



```
UPDATE alunos  
SET nome = "Raffael Bottoli Schemmer"  
WHERE id = 11
```

```
db.people.updateOne({"id":11},{ $set: {"nome":"Rafael Bottoli Schemmer"}})
```

```
db.alunos.updateOne({"id":11},{ $set: {"nome":"Raffael Bottoli Schemmer"}})
```

Atualizando Documentos



Atualize um ID de um Aluno
(Faça a Complementação do Nome e Troque o Status de Trabalho)



?

Perguntas

Removendo Documentos



```
DELETE FROM pessoa
```

```
db.pessoa.remove({})
```

```
db.alunos.remove({})
```

Removendo Documentos



```
DELETE FROM pessoa  
WHERE sexo = "M"
```

```
db.pessoa.remove({"sexo":"M"})
```

```
db.alunos.remove({"sexo":"M"  
})
```


Removendo Documentos



Utilize Todos os Operadores para Consulta
(A Sintaxe é a Mesma para a Remoção)

Removendo Documentos



Remova Todos os Alunos
(Que Forem do Sexo Masculino)



?

Perguntas

Indo Além



Aggregation e Performance (Query/Index) (Módulo 8)



Mongo DB + NodeJS



Mongo DB + NodeJS (Configuração)



```
var express = require('express');
var router = express.Router();
var mongoose = require('mongoose');
mongoose.connect('mongodb://user:senha@host/base');
var Schema = mongoose.Schema;

var userDataSchema = new Schema({
  nome: {type: String, required: true},
  email: String,
  telefone: String
}, {collection: 'contatos'});

var Contatos = mongoose.model('UserData', userDataSchema);
module.exports = router;
```

Mongo DB + NodeJS (Criando)



```
router.post('/new', function(req, res, next) {  
  var item = {  
    nome: req.body.nome,  
    email: req.body.email,  
    telefone: req.body.telefone  
  };  
  
  var data = new Contatos(item);  
  data.save();  
  
  res.redirect('/');  
});
```

Mongo DB + NodeJS (Editando)



```
router.post('/edit', function(req, res, next) {
  var id = req.body.id;

  Contatos.findById(id, function(err, doc) {
    if (err) {
      console.error('error, no entry found');
    }
    doc.nome = req.body.nome;
    doc.email = req.body.email;
    doc.telefone = req.body.telefone;
    doc.save();
  })
  res.redirect('/');
});
```


Mongo DB + NodeJS (Removendo)



```
router.post('/delete', function(req, res, next) {  
  var id = req.body.id;  
  Contatos.findByIdAndRemove(id).exec();  
  res.redirect('/');  
});
```

Mongo DB + NodeJS (Listando)



```
router.get('/', function(req, res, next) {  
  Contatos.find()  
    .then(function(doc) {  
      res.render('index', {items: doc});  
    });  
});
```

Atividade



Desenvolver Find/Update/Delete NoSQL
(Considerando as Operações da Aplicação)



?

Perguntas