SÃO PAULO
GOVERNO DO ESTADO
SÃO PAULO SÃO TODOS

DESENVOLVIMENTO DE SOFTWARE MULTIPLATAFORMA

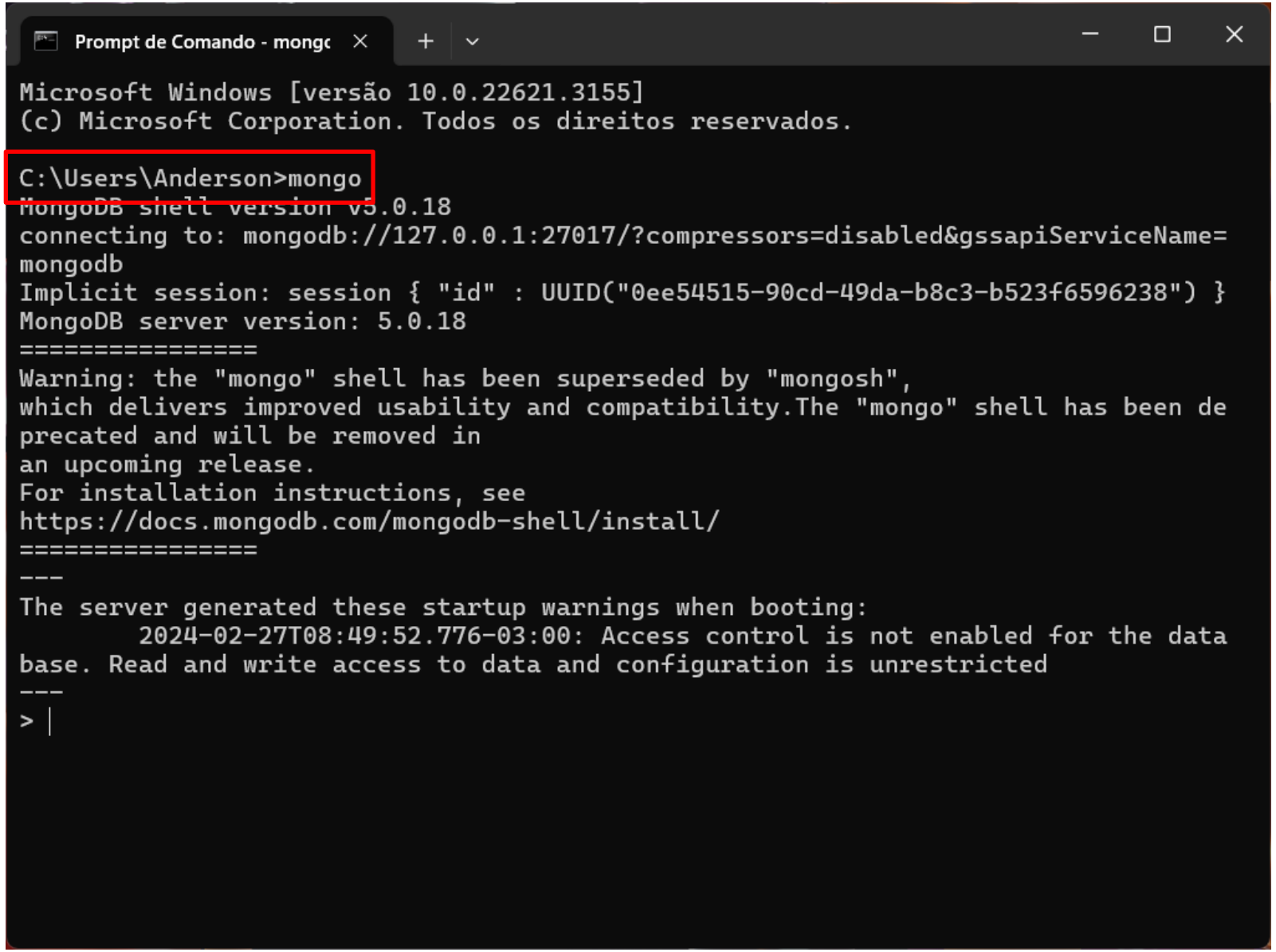
Disciplina: IBD-016 – BANCO DE DADOS - NÃO RELACIONAL

Aula 03: Tipos de Bancos de Dados NoSQL

Data 14/03/2024

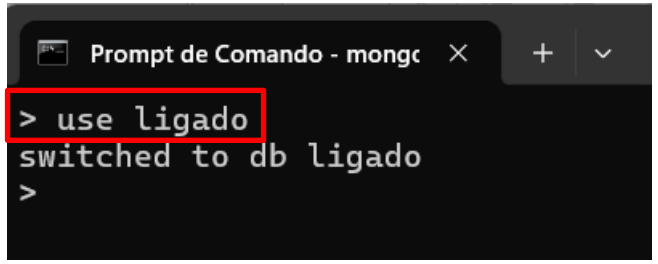
Prof. Me. Anderson Silva Vanin

BD LIGADO


Prompt de Comando - mongc
Microsoft Windows [versão 10.0.22621.3155]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\Anderson>mongo
MongoDB shell version v5.0.18
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0ee54515-90cd-49da-b8c3-b523f6596238") }
MongoDB server version: 5.0.18
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================

The server generated these startup warnings when booting:
2024-02-27T08:49:52.776-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
>

BD LIGADO




```
db.albuns.insert({
    "nome":"Master of Puppets",
    "dataLancamento": new Date(1986,2,3),
    "duracao": 3286
})
```

```
> db.albuns.insert({
...     "nome":"Master of Puppets",
...     "dataLancamento": new Date(1986,2,3),
...     "duracao": 3286
... })
WriteResult({ "nInserted" : 1 })
>
```

```
> db.albuns.find().pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
>
```

BD LIGADO

```
db.albuns.insertMany([
    {
        "nome":"...And Justice for All",
        "dataLancamento": new Date(1988,7,25),
        "duracao": 3929
    },
    {
        "nome":"Peace Sells... but who's Buying?",
        "dataLancamento": new Date(1986,8,19),
        "duracao": 2172,
        "estudioGravacao":"Music Grinder Studios"
    },
    {
        "nome":"Reign in Blood",
        "dataLancamento": new Date(1986,9,7),
        "duracao": 1738,
        "artistaCapa":"Larry Carroll"
    },
    {
        "nome":"Among the Living",
        "produtor": "Eddie Kramer"
    }
])
```

```
> db.albuns.insertMany([
...     {
...         "nome":"...And Justice for All",
...         "dataLancamento": new Date(1988,7,25),
...         "duracao": 3929
...     },
...     {
...         "nome":"Peace Sells... but who's Buying?",
...         "dataLancamento": new Date(1986,8,19),
...         "duracao": 2172,
...         "estudioGravacao":"Music Grinder Studios"
...     },
...     {
...         "nome":"Reign in Blood",
...         "dataLancamento": new Date(1986,9,7),
...         "duracao": 1738,
...         "artistaCapa":"Larry Carroll"
...     },
...     {
...         "nome":"Among the Living",
...         "produtor": "Eddie Kramer"
...     }
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("65e8cafa10d1f5d7e4cb56db"),
                ObjectId("65e8cafa10d1f5d7e4cb56dc"),
                ObjectId("65e8cafa10d1f5d7e4cb56dd"),
                ObjectId("65e8cafa10d1f5d7e4cb56de")
        ]
}
>
```

BD LIGADO

```
> db.albuns.find().pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56db"),
        "nome" : "...And Justice for All",
        "dataLancamento" : ISODate("1988-08-25T03:00:00Z"),
        "duracao" : 3929
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dc"),
        "nome" : "Peace Sells... but who's Buying?",
        "dataLancamento" : ISODate("1986-09-19T03:00:00Z"),
        "duracao" : 2172,
        "estudioGravacao" : "Music Grinder Studios"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dd"),
        "nome" : "Reign in Blood",
        "dataLancamento" : ISODate("1986-10-07T03:00:00Z"),
        "duracao" : 1738,
        "artistaCapa" : "Larry Carroll"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56de"),
        "nome" : "Among the Living",
        "produtor" : "Eddie Kramer"
}
>
```

BD LIGADO – USO DO FIND

Definição

```
db.collection.find(query, projection, options)
```

Operadores de consulta

Projeção

Opções

Parâmetro	Tipo	Descrição
consulta	documento	Opcional. Especifica o filtro de seleção usando operadores de consulta . Para retornar todos os documentos de uma coleção, omita este parâmetro ou passe um documento vazio ({}).
projeção	documento	Opcional. Especifica os campos a serem retornados nos documentos que correspondem ao filtro de consulta. Para retornar todos os campos nos documentos correspondentes, omita este parâmetro. Para obter detalhes, consulteProjeção .
opções	documento	Opcional. Especifica opções adicionais para a consulta. Estas opções modificam o comportamento da consulta e como os resultados são retornados. Para ver as opções disponíveis, consulteEncontre opções .

```
db.collection.find({},{},{})
```

BD LIGADO – SELECIONANDO UM DOCUMENTO PELO VALOR DE UM CAMPO

```
db.albuns.find({
    "nome":"Master of Puppets"
})
```

```
> db.albuns.find({
...     "nome":"Master of Puppets"
... })
{ "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"), "nome" : "Master of Puppets", "da
taLancamento" : ISODate("1986-03-03T03:00:00Z"), "duracao" : 3286 }
> db.albuns.find({
...     "nome":"Master of Puppets"
... }).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
>
```

BD LIGADO – OPERADORES DE COMPARAÇÃO

Comparação

Nome	Descrição
`$eq`	Corresponde a valores iguais a um valor especificado.
`$gt`	Corresponde a valores maiores que um valor especificado.
`$gte`	Corresponde a valores maiores ou iguais a um valor especificado.
`$in`	Corresponde a qualquer um dos valores especificados em uma matriz.
`$lt`	Corresponde a valores menores que um valor especificado.
`$lte`	Corresponde a valores menores ou iguais a um valor especificado.
`$ne`	Corresponde a todos os valores que não são iguais a um valor especificado.
`$nin`	Não corresponde a nenhum dos valores especificados em uma matriz.

BD LIGADO – SELECIONANDO OS DOCUMENTO CUJA DURAÇÃO SEJA MAIOR QUE 3000

Sintaxe

O `$gt` operador tem o seguinte formato:

```
{ field: { $gt: value } }
```

```
db.albuns.find({
    "duracao":{
        $gt: 3000
    }
}).pretty()
```

```
> db.albuns.find({
...     "duracao":{
...         $gt: 3000
...     }
... }).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56db"),
        "nome" : "...And Justice for All",
        "dataLancamento" : ISODate("1988-08-25T03:00:00Z"),
        "duracao" : 3929
}
>
```

BD LIGADO – OPERADORES LÓGICOS

Lógico

Nome	Descrição
`$and`	Unir cláusulas de consulta com uma lógica `AND` retorna todos os documentos que correspondem às condições de ambas as cláusulas.
`$not`	Inverte o efeito de uma expressão de consulta e retorna documentos que não *correspondem* à expressão de consulta.
`$nor`	Unir cláusulas de consulta com uma lógica `NOR` retorna todos os documentos que não correspondem a ambas as cláusulas.
`$or`	Une cláusulas de consulta com uma lógica `OR` que retorna todos os documentos que correspondem às condições de qualquer uma das cláusulas.

BD LIGADO – SELECIONANDO OS DOCUMENTOS CUJA DURAÇÃO SEJA MAIOR QUE 2000 E MENORES QUE 3500

```
db.albuns.find( {
    $and:
    [
       { "duracao": { $gt: 2000 } },
       { "duracao": { $lt: 3500 } }
    ]
}).pretty()
```

```
> db.albuns.find( {
...     $and:
...     [
...        { "duracao": { $gt: 2000 } },
...        { "duracao": { $lt: 3500 } }
...     ]
... }).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dc"),
        "nome" : "Peace Sells... but who's Buying?",
        "dataLancamento" : ISODate("1986-09-19T03:00:00Z"),
        "duracao" : 2172,
        "estudioGravacao" : "Music Grinder Studios"
}
>
```

BD LIGADO – SELECIONANDO OS DOCUMENTOS CUJA DURAÇÃO SEJA MAIOR QUE 2000 E MENORES QUE 3500 E EXIBINDO SOMENTE OS CAMPOS NOME E DURAÇÃO

```
db.albuns.find(
    {
        $and:
        [
        { "duracao": { $gt: 2000 } },
        { "duracao": { $lt: 3500 } }
        ]
    },
    {
        "nome":1,
        "duracao":1
    }
).pretty()
```

```
> db.albuns.find(
...     {
...         $and:
...         [
...         { "duracao": { $gt: 2000 } },
...         { "duracao": { $lt: 3500 } }
...         ]
...     },
...     {
...         "nome":1,
...         "duracao":1
...     }
... ).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dc"),
        "nome" : "Peace Sells... but who's Buying?",
        "duracao" : 2172
}
>
```

BD LIGADO – SELECIONANDO UM DOCUMENTOS PELO ObjectId

```
db.albuns.find(
    {
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da")
    }
).pretty()
```

```
> db.albuns.find(
...     {
...         "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da")
...     }
... ).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
>
```

BD LIGADO – INSERINDO UM NOVO DOCUMENTO PARA AS OPERAÇÕES DE UPDATE E DELETE

```
db.albuns.insert({
    "nome":"Este Albúm Será Editado e Depois Apagado",
    "dataLancamento": new Date(2024,6,3),
    "duracao": 3150,
    "produtor": "Eddie Kramer"
})
```

```
        "produtor" : "Eddie Kramer"
}
{
        "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df"),
        "nome" : "Este Albúm Será Editado e Depois Apagado",
        "dataLancamento" : ISODate("2024-07-03T03:00:00Z"),
        "duracao" : 3150,
        "produtor" : "Eddie Kramer"
}
```

BD LIGADO – UPDATE

Definição

```
db.collection.updateOne(filter, update, options)
```

```
db.albuns.updateOne(
    {
        "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df")
    },
    {
        $set:{"nome" : "Foi Atualizado!"}
    }
)
```

```
> db.albuns.updateOne(
...     {
...         "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df")
...     },
...     {
...         $set:{"nome" : "Foi Atualizado!"}
...     }
... )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

BD LIGADO – UPDATE

```
> db.albuns.find({}).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56db"),
        "nome" : "...And Justice for All",
        "dataLancamento" : ISODate("1988-08-25T03:00:00Z"),
        "duracao" : 3929
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dc"),
        "nome" : "Peace Sells... but who's Buying?",
        "dataLancamento" : ISODate("1986-09-19T03:00:00Z"),
        "duracao" : 2172,
        "estudioGravacao" : "Music Grinder Studios"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dd"),
        "nome" : "Reign in Blood",
        "dataLancamento" : ISODate("1986-10-07T03:00:00Z"),
        "duracao" : 1738,
        "artistaCapa" : "Larry Carroll"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56de"),
        "nome" : "Among the Living",
        "produtor" : "Eddie Kramer"
}
{
        "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df"),
        "nome" : "Foi Atualizado!",
        "dataLancamento" : ISODate("2024-07-03T03:00:00Z"),
        "duracao" : 3150,
        "produtor" : "Eddie Kramer"
}
>
```

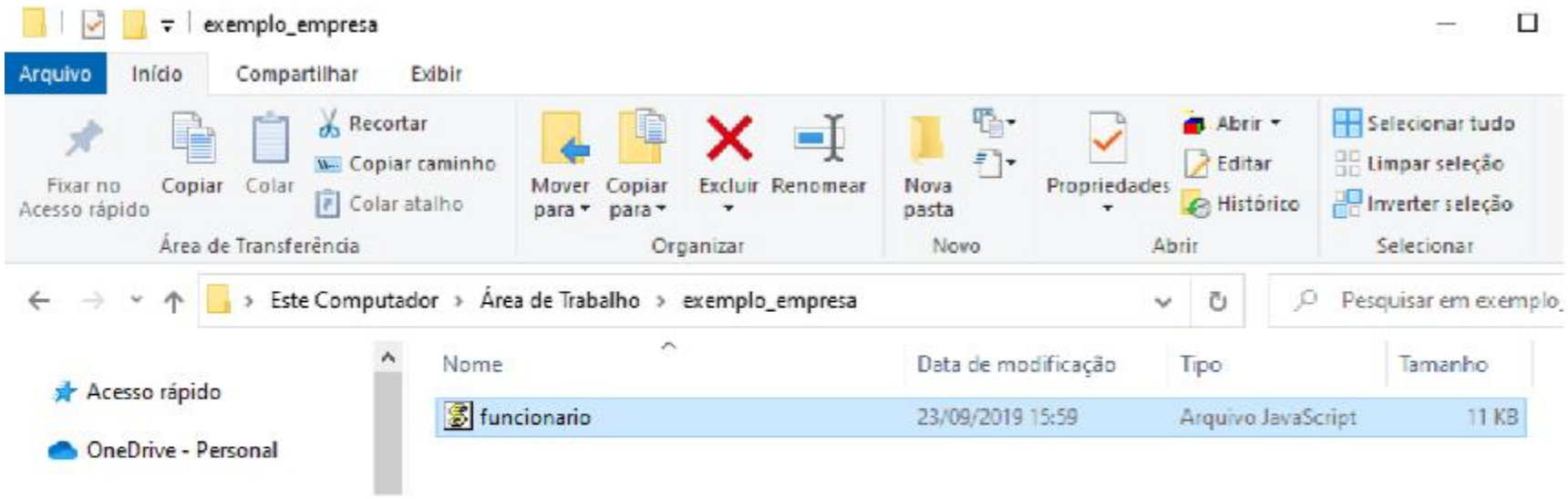
BD LIGADO – DELETE

```
db.albuns.deleteOne({
    "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df")
})
```

```
> db.albuns.deleteOne({
...     "_id" : ObjectId("65e8d6c510d1f5d7e4cb56df")
... })
{ "acknowledged" : true, "deletedCount" : 1 }
>
```

```
> db.albuns.find({}).pretty()
{
        "_id" : ObjectId("65e8c90a10d1f5d7e4cb56da"),
        "nome" : "Master of Puppets",
        "dataLancamento" : ISODate("1986-03-03T03:00:00Z"),
        "duracao" : 3286
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56db"),
        "nome" : "...And Justice for All",
        "dataLancamento" : ISODate("1988-08-25T03:00:00Z"),
        "duracao" : 3929
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dc"),
        "nome" : "Peace Sells... but who's Buying?",
        "dataLancamento" : ISODate("1986-09-19T03:00:00Z"),
        "duracao" : 2172,
        "estudioGravacao" : "Music Grinder Studios"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56dd"),
        "nome" : "Reign in Blood",
        "dataLancamento" : ISODate("1986-10-07T03:00:00Z"),
        "duracao" : 1738,
        "artistaCapa" : "Larry Carroll"
}
{
        "_id" : ObjectId("65e8cafa10d1f5d7e4cb56de"),
        "nome" : "Among the Living",
        "produtor" : "Eddie Kramer"
}
>
```

IMPORTANDO DE UM ARQUIVO JSON




```
C:\Users\Anderson\Desktop\exemplo_empresa>mongo <funcionario.js
```

Observação: Utilizar a base de dados Empresa disponível no Github aqui

* Crie uma pasta na área de trabalho e salve este arquivo nela!

IMPORTANDO DE UM ARQUIVO JSON

Após a importação do banco de dados, visualize seu conteúdo.

```
> show dbs
admin          0.000GB
avaliacaoP1    0.000GB
config         0.000GB
empresa        0.000GB
ligado         0.000GB
local          0.000GB
```

```
> use empresa
switched to db empresa
```

```
> show collections
funcionario
```

```
> db.funcionario.find().pretty()
{
        "_id" : 1,
        "nome" : "CARLOS MACEDO CERRI",
        "depto" : 3,
        "funcao" : "VENDEDOR",
        "salario" : 1530,
        "admissao" : ISODate("2010-05-25T20:06:20.123Z"),
        "casado" : true,
        "skill" : "EXCELENTE RELACIONAMENTO INTERPESSOAL",
        "endereco" : {
                "logradouro" : "AVENIDA PERNAMBUCO",
                "bairro" : "BRASIL",
                "uf" : "BA",
                "numero" : 102,
                "cidade" : "VITÓRIA DA CONQUISTA",
                "cep" : "45000-000"
        },
        "hobbies" : [
                "TÊNIS DE MESA",
                "YOGA",
                "VIAGEM"
        ],
        "notas" : [
```

IMPORTANDO DE UM ARQUIVO JSON

- Uma boa prática é conhecer o conteúdo armazenado em suas collections e se familiarizar com a estrutura do banco de dados.
- Visualize os documentos criados, avalie padrões de estrutura (documentos com campos parecidos ou iguais, etc), quantidade de documentos, etc.

Visualizando o primeiro documento


```
{
        "_id" : 1,
        "nome" : "CARLOS MACEDO CERRI",
        "depto" : 3,
        "funcao" : "VENDEDOR",
        "salario" : 1530,
        "admissao" : ISODate("2010-05-25T20:06:20.123Z"),
        "casado" : true,
        "skill" : "EXCELENTE RELACIONAMENTO INTERPESSOAL",
        "endereco" : {
                "logradouro" : "AVENIDA PERNAMBUCO",
                "bairro" : "BRASIL",
                "uf" : "BA",
                "numero" : 102,
                "cidade" : "VITÓRIA DA CONQUISTA",
                "cep" : "45000-000"
        },
        "hobbies" : [
                "TÊNIS DE MESA",
                "YOGA",
                "VIAGEM"
        ],
        "notas" : [
                {
                        "criterio" : "PRODUÇÃO",
                        "nota" : 4.8
                },
                {
                        "criterio" : "CONVÍVIO",
                        "nota" : 9.2
                }
        ],
        "avaliacoes" : [
                73,
                87,
                77
        ],
        "feedbacks" : [
                true,
                true
        ]
}
```

Tipos de estruturas de campos em documentos

A. De forma geral um documento no Mongo segue o padrão: **“chave” : “valor”**

```
"_id" : 1,
"nome" : "CARLOS MACEDO CERRI",
"depto" : 3,
"funcao" : "VENDEDOR",
```

B. Também é possível em um campo armazenar ao invés de um valor único, termos um outro documento como valor deste campo, e este documento possuir a estrutura **“chave” : “valor”**

```
"endereco" : {
        "logradouro" : "AVENIDA PERNAMBUCO",
        "bairro" : "BRASIL",
        "uf" : "BA",
        "numero" : 102,
        "cidade" : "VITÓRIA DA CONQUISTA",
        "cep" : "45000-000"
},
```

Tipos de estruturas de campos em documentos

C. Também é possível em um campo armazenar uma lista de valores que são associados à este campo.

```
"hobbies" : [
        "TÊNIS DE MESA",
        "YOGA",
        "VIAGEM"
],
```

D. Por último, podemos também, associar à um campo, uma lista de documentos.

```
"notas" : [
        {
                "criterio" : "PRODUÇÃO",
                "nota" : 4.8
        },
        {
                "criterio" : "CONVÍVIO",
                "nota" : 9.2
        }
],
```

Exemplos de consultas para cada tipo de campo

A. SELECIONAR ***NOME*** E ***FUNÇÃO*** DO FUNCIONÁRIO QUE TEM O ***_ID = 1***.

```
> db.funcionario.find({"_id": 1},{"nome":1,"funcao":1}).pretty()
{ "_id" : 1, "nome" : "CARLOS MACEDO CERRI", "funcao" : "VENDEDOR" }
```

```
db.funcionario.find(
    {"_id": 1 },
    {
        "nome":1,
        "funcao":1
    }
).pretty()

// ou

db.funcionario.find({"_id": 1},{"nome":1,"funcao":1}).pretty()
```

Exemplos de consultas para cada tipo de campo

B. SELECIONAR ***BAIRRO*** DO FUNCIONÁRIO QUE TEM O ***_ID = 1***.

```
> db.funcionario.find({"_id": 1},{"endereco.bairro":1}).pretty()
{ "_id" : 1, "endereco" : { "bairro" : "BRASIL" } }
```

```
db.funcionario.find(
    {"_id": 1 },
    {
        "endereco.bairro":1,
    }
).pretty()

// ou

db.funcionario.find({"_id": 1},{"endereco.bairro":1}).pretty()
```

Exemplos de consultas para cada tipo de campo

C. SELECIONAR ***HOBBIES*** DO FUNCIONÁRIO QUE TEM O ***_ID = 1***.

```
> db.funcionario.find({"_id": 1}, {"hobbies": 1})
{ "_id" : 1, "hobbies" : [ "TÊNIS DE MESA", "YOGA", "VIAGEM" ] }
> db.funcionario.find({"_id": 1}, {"hobbies": 1,"_id":0})
{ "hobbies" : [ "TÊNIS DE MESA", "YOGA", "VIAGEM" ] }
```

```
db.funcionario.find({"_id": 1}, {"hobbies": 1,"_id":0})
```

Exemplos de consultas para cada tipo de campo

D. SELECIONAR ***CRITERIOS*** DO CAMPO ***NOTAS*** DO FUNCIONÁRIO QUE TEM O ***_ID = 1***.

```
> db.funcionario.find({"_id": 1},{"notas.criterio": 1 }).pretty()
{
        "_id" : 1,
        "notas" : [
                {
                        "criterio" : "PRODUÇÃO"
                },
                {
                        "criterio" : "CONVÍVIO"
                }
        ]
}
```

```
db.funcionario.find({"_id": 1},{"notas.criterio": 1 })
```

Exercícios com base no banco importado empresa

1. Selecionar os nomes e os endereços dos funcionários que são programadores e trabalham no departamento 3.
2. Selecionar os funcionários com função de programador e de analista que ganham acima de R$ 2.000,00. Classificar em ordem crescente pela função e nome do funcionário.
3. Selecionar os funcionários com salário maior do que R$ 2.600,00 que sejam analistas, ou tenham exatamente um filho. Mostrar o nome, a função, o salário, a quantidade de filhos e o bairro onde mora do funcionário. Ordenar em ordem decrescente pelo bairro.

Exercícios com base no banco importado empresa

4. Quais os funcionários possuem hobby e não têm filhos? Mostrar o nome, o hobby e o salário.
5. Quais os funcionários possuem filhos, recebe salário maior do que R$ 1.500,00 e tem como hobby futebol ou xadrez? Mostrar o nome, a quantidade de filhos, os hobbies e o salário.
6. Quais os funcionários têm como hobby futebol e tênis de mesa? Não mostrar o salário e a data de admissão.
7. Quais os funcionários têm 4 hobbies, sendo que pelo menos 1 hobby seja natação? Mostrar o nome e os hobbies.

Exercícios com base no banco importado empresa

8. Quais os funcionários que não ocupem a função de secretária, não tem tênis de mesa como hobby e é do departamento 3?
9. Quantos funcionários do departamento 3 ganham entre R$ 1.200,50 e R$ 1.600,00?
10. Quantos funcionários moram em Vitória da Conquista e possuem algum hobby?
11. Recuperar os funcionários que possuem os três maiores salários, excluindo o funcionário com o maior salário. Não mostrar os hobbies, as notas, as avaliações, os feedbacks e o endereço. Utilizar os métodos limit e skip.
12. Quais os funcionários têm 3 ou 4 feedbacks? Mostrar o nome e os feedbacks ordenado em ordem decrescente pelo nome.