



IBD-016 – BANCO DE DADOS - NÃO RELACIONAL

Prof. Me. Anderson Vanin



Banco de Dados de Documentos + Objetos

Caso da Nota Fiscal



Green Leaf Design
111 Main Street
Pleasanton, CA 99999
555.555.4444
info@greenleafdesign.com

Bill To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)
555-555-2225 (Fax)

Ship To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)

Shipment Details:
Carrier: USPS
Method: Overnight Priority
Tracking#: w126a5s4321sas
Ship Date: 03/02/2009

INVOICE

Invoice: 00016
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.01
Customer PO#:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System Includes logo, layout for letterhead, 2nd sheet, A10 envelope, and business card	3,500.00	project	1	3,500.00	8.25%
Design Services - Additional Concepts includes thumbnail sketches for one additional concept	125.00	each	1	125.00	8.25%

Pre-tax Total: 3,625.00

Tax: 299.06

Shipping: 12.95

Total: 3,937.01

Payments: -2,000.00

Balance (\$): 1,937.01

Notes

Thanks for the work! Art hard copies and CD mailed to printer per instructions.

Terms and Conditions

Balance Due Net 30 days.

Banco de Dados de Documentos + Objetos

Vamos transformar o exemplo da NF mostrada em uma estrutura.



Green Leaf Design
111 Main Street
Pleasanton, CA 99999
555.555.4444
info@greenleafdesign.com

Bill To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)
555-555-2225 (Fax)

Ship To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)

Shipment Details:
Carrier: USPS
Method: Overnight Priority
Tracking#: w126a5s4321sas
Ship Date: 03/02/2009

INVOICE

Invoice: 00016
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.01
Customer PO#:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System Includes logo, layout for letterhead, 2nd sheet, A10 envelope, and business card	3,500.00	project	1	3,500.00	8.25%
Design Services - Additional Concepts includes thumbnail sketches for one additional concept	125.00	each	1	125.00	8.25%

Pre-tax Total: 3,625.00

Tax: 299.06

Shipping: 12.95

Total: 3,937.01

Payments: -2,000.00

Balance (\$): 1,937.01

Notes

Thanks for the work! Art hard copies and CD mailed to printer per instructions.

Terms and Conditions

Balance Due Net 30 days.

notafiscal:

- numero

- data

- valor

- item:

- caracterização:

- nomeproduto

- categoria

- preço

- quantidade

**Seria possível construir
essa estrutura em uma
única tabela no banco
de dados relacional?**

Banco de Dados de Documentos + Objetos

Quando lidamos com esse tipo de estrutura em banco de dados temos duas situações:

- **Armazenamento**
- **Troca de Informações entre sistemas**

Vamos ver agora sob o aspecto de intercambiar essas informações em diferentes sistemas.

Banco de Dados de Documentos + Objetos

Quando lidamos com esse tipo de estrutura em banco de dados temos duas situações:

- **Armazenamento**
- **Troca de Informações entre sistemas**

Vamos ver agora sob o aspecto de intercambiar essas informações em diferentes sistemas.

Banco de Dados de Documentos + Objetos

Public/Subscribe

ESTOQUE

BARRAMENTO

VENDAS

LOGISTICA



CLIENTE

Neste aspecto, porque o BD em tabelas é ruim neste momento?

INVOICE

Green Leaf Design
211 Main Street
Pawcatuck, CT 06460
505-555-4444
info@greenunderdesign.com

Bill To:
Alan Miller and Rosett
2000 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Ship To:
Alan Miller and Rosett
200 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Invoice Details:
Invoice: 00000
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.81
Customer POC:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System	3,000.00	project	1	3,000.00	0.25%
Design Services - Business System	125.00	each	1	125.00	0.25%
Price Due Total:				3,000.00	
Tax:				280.00	
Shipping:				12.50	
Total:				3,937.81	
Payment:				1,937.81	
Balance Due (\$):				1,937.81	

Notes:
Thanks for the work. All hard copies and CD mailed to printer per instructions.

Terms and Conditions:
Balance Due Net 30 days.

- notafiscal:**
- numero
 - data
 - valor
 - item:
 - caracterização:
 - nomeproduto
 - categoria
 - preço
 - quantidade

INVOICE

Green Leaf Design
211 Main Street
Pawcatuck, CT 06460
505-555-4444
info@greenunderdesign.com

Bill To:
Alan Miller and Rosett
2000 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Ship To:
Alan Miller and Rosett
200 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Invoice Details:
Invoice: 00000
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.81
Customer POC:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System	3,000.00	project	1	3,000.00	0.25%
Design Services - Business System	125.00	each	1	125.00	0.25%
Price Due Total:				3,000.00	
Tax:				280.00	
Shipping:				12.50	
Total:				3,937.81	
Payment:				1,937.81	
Balance Due (\$):				1,937.81	

Notes:
Thanks for the work. All hard copies and CD mailed to printer per instructions.

Terms and Conditions:
Balance Due Net 30 days.

- notafiscal:**
- numero
 - data
 - valor
 - item:
 - caracterização:
 - nomeproduto
 - categoria
 - preço
 - quantidade

INVOICE

Green Leaf Design
211 Main Street
Pawcatuck, CT 06460
505-555-4444
info@greenunderdesign.com

Bill To:
Alan Miller and Rosett
2000 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Ship To:
Alan Miller and Rosett
200 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Invoice Details:
Invoice: 00000
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.81
Customer POC:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System	3,000.00	project	1	3,000.00	0.25%
Design Services - Business System	125.00	each	1	125.00	0.25%
Price Due Total:				3,000.00	
Tax:				280.00	
Shipping:				12.50	
Total:				3,937.81	
Payment:				1,937.81	
Balance Due (\$):				1,937.81	

Notes:
Thanks for the work. All hard copies and CD mailed to printer per instructions.

Terms and Conditions:
Balance Due Net 30 days.

- notafiscal:**
- numero
 - data
 - valor
 - item:
 - caracterização:
 - nomeproduto
 - categoria
 - preço
 - quantidade

INVOICE

Green Leaf Design
211 Main Street
Pawcatuck, CT 06460
505-555-4444
info@greenunderdesign.com

Bill To:
Alan Miller and Rosett
2000 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Ship To:
Alan Miller and Rosett
200 Broadway
New York, NY 10008
505-555-2222 (Phone)
505-555-2222 (Fax)

Invoice Details:
Invoice: 00000
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.81
Customer POC:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System	3,000.00	project	1	3,000.00	0.25%
Design Services - Business System	125.00	each	1	125.00	0.25%
Price Due Total:				3,000.00	
Tax:				280.00	
Shipping:				12.50	
Total:				3,937.81	
Payment:				1,937.81	
Balance Due (\$):				1,937.81	

Notes:
Thanks for the work. All hard copies and CD mailed to printer per instructions.

Terms and Conditions:
Balance Due Net 30 days.

- notafiscal:**
- numero
 - data
 - valor
 - item:
 - caracterização:
 - nomeproduto
 - categoria
 - preço
 - quantidade

Descrevendo animais pré-históricos



MNHN A. C. 8592		
Especie	Plesiosauros dolichodeirus	
Origem	Lyme Regis	Inglaterra
Descoberto	1824	
Tamanho	5	

Descrevendo animais pré-históricos



STC223		
Especie	Plesiosauros gurgitis	
Origem	St. Crox	Suíça
Descoberto	1964	
Tamanho	3.5	

Descrevendo animais pré-históricos



MNHN 1912.20		
Especie	Triceratops horidus	
Origem	Lance Creek	EUA
Descoberto	1859	
Tamanho	9	

Descrevendo animais pré-históricos



FMNH PR2081		
Especie	Tyranosaurus rex	
Origem	Hell Creek	EUA
Descoberto	1990	
Tamanho	12.3	

Descrevendo animais pré-históricos



MNHN A. C. 8592		
<u>Especie</u>	<u>Plesiosauros dolichodeirus</u>	
<u>Origem</u>	<u>Lyme Regis</u>	Inglaterra
<u>Descoberto</u>	1824	
<u>Tamanho</u>	5	



STC223		
<u>Especie</u>	<u>Plesiosauros gurgitis</u>	
<u>Origem</u>	<u>St. Croix</u>	Suíça
<u>Descoberto</u>	1964	
<u>Tamanho</u>	3.5	



MNHN 1912.20		
<u>Especie</u>	<u>Triceratops horridus</u>	
<u>Origem</u>	<u>Lance Creek</u>	EUA
<u>Descoberto</u>	1859	
<u>Tamanho</u>	9	



FMNH PR2081		
<u>Especie</u>	<u>Tyrannosaurus rex</u>	
<u>Origem</u>	<u>Hell Creek</u>	EUA
<u>Descoberto</u>	1990	
<u>Tamanho</u>	23	

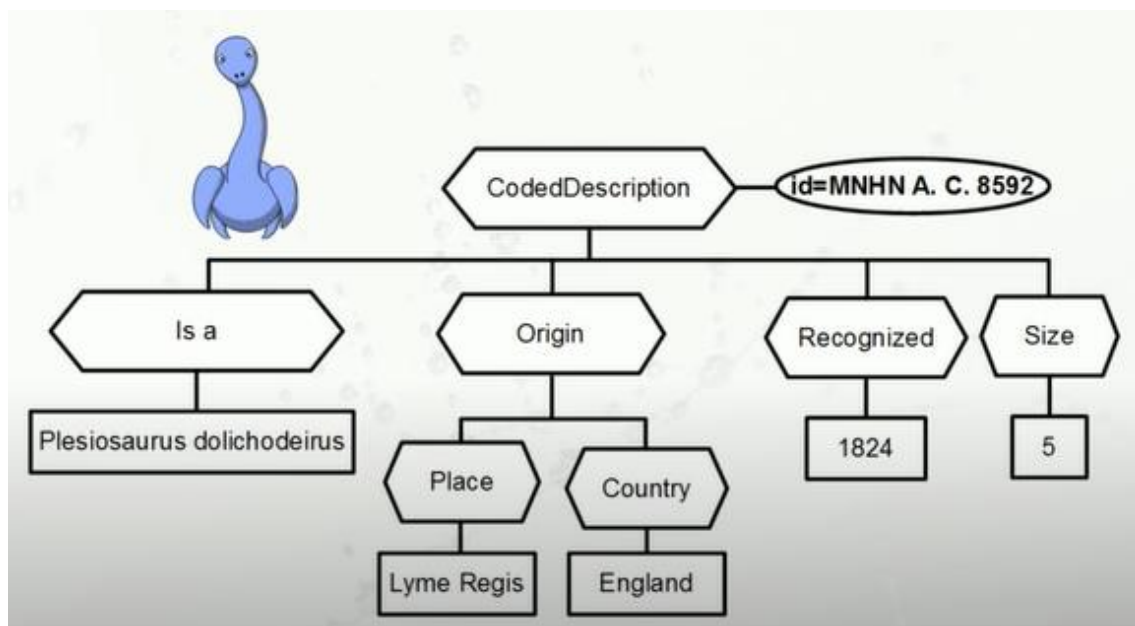
Tabela

id	Espécie	Origem Lugar	Origem Pais	Descoberto	Tamanho
MNHN A. C. 8592	Plesiosauros dolichodeirus	Lyme Regis	Inglaterra	1824	5
STC223	Plesiosauros gurgitis	St. Crox	Suíça	1964	3.5
MNHN 1912.20	Triceratops horidus	Lance Creek	EUA	1859	9
FMNH PR2081	Tyrannosaurus rex	Hell Creek	EUA	1990	12.3

Tabela

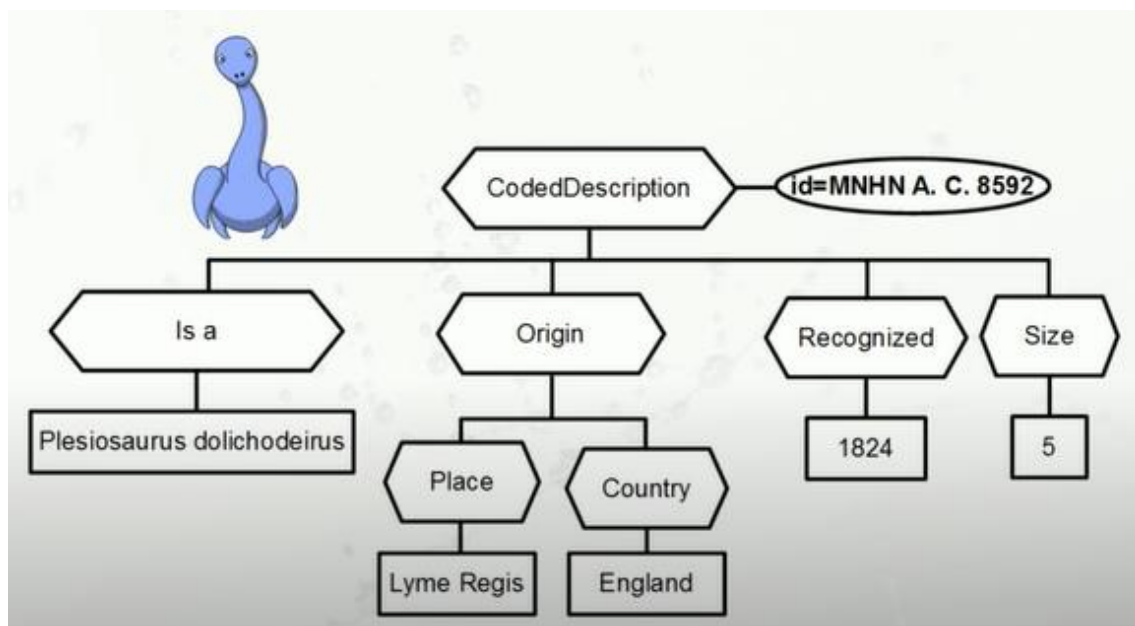
- Excelente para gerenciar dados com esquema estático e previsível.
- Mas e quanto ao compartilhamento?

Documentos e XML



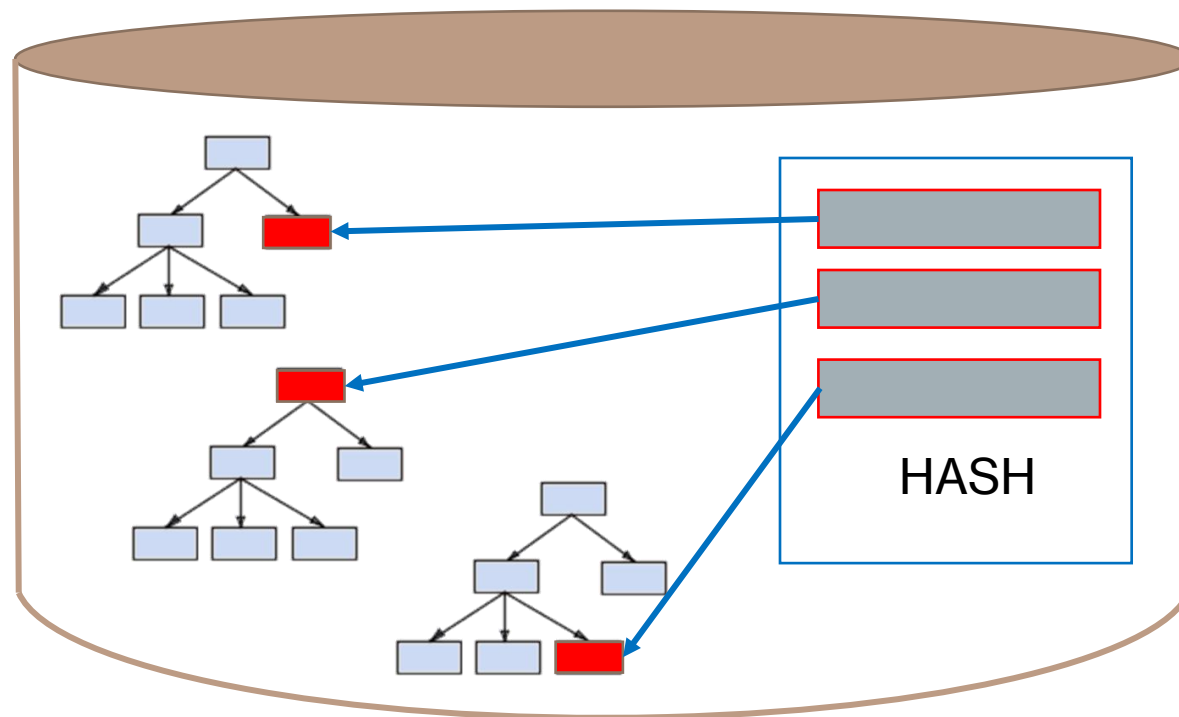
O XML é um modelo hierárquico. O XML quando foi criado foi pensado muito mais no aspecto de intercâmbio do que em armazenamento.

Documentos e XML



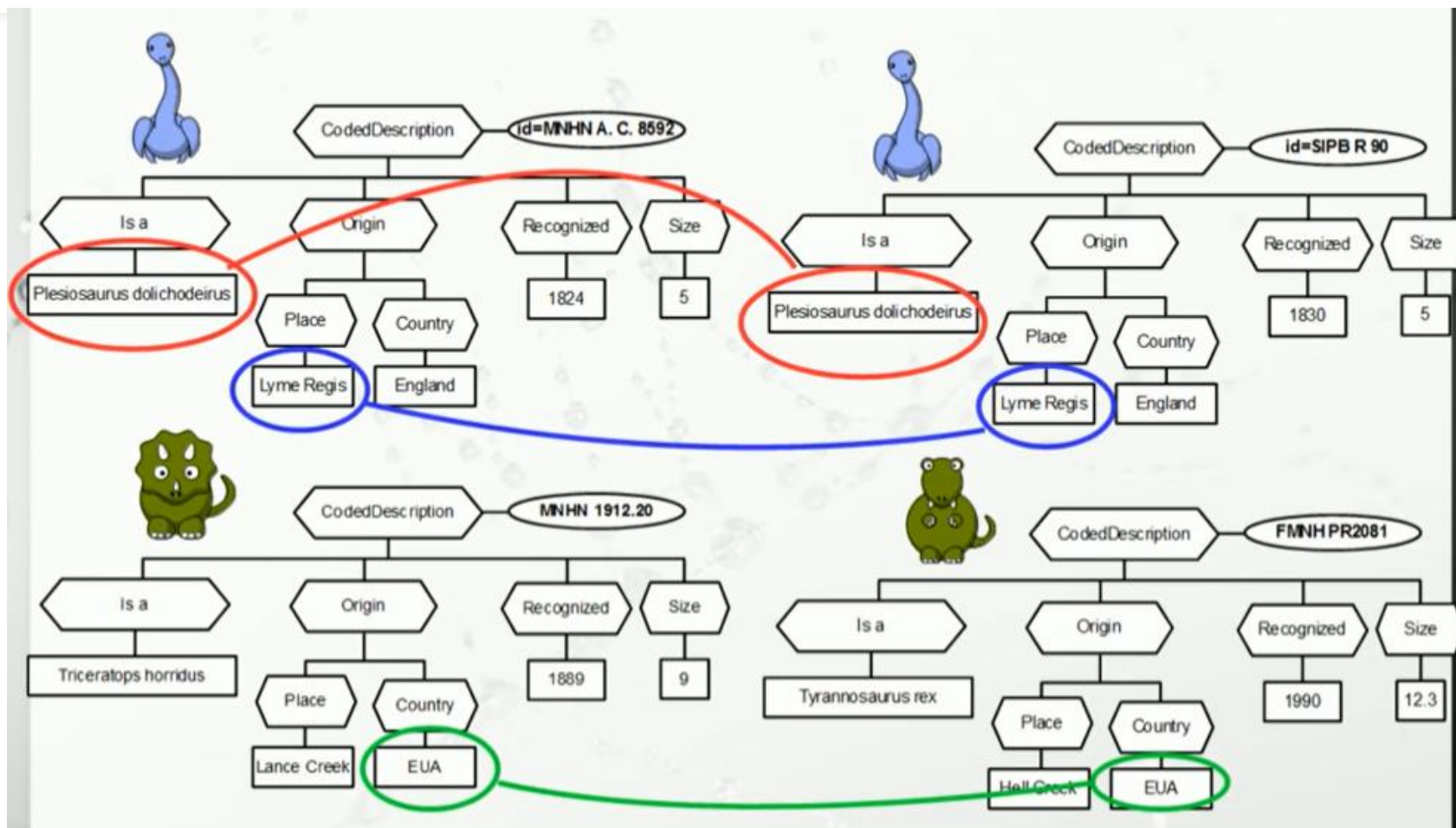
Se podemos guardar o documento com as informações, porque transformar em uma tabela para guardar?

Documentos e XML



Em uma base de dados relacional, podemos criar índices para agilizar a busca por informações, através de HASHES.

Documentos e XML



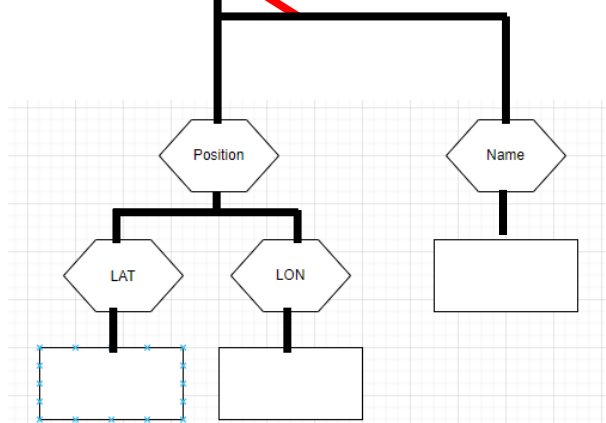
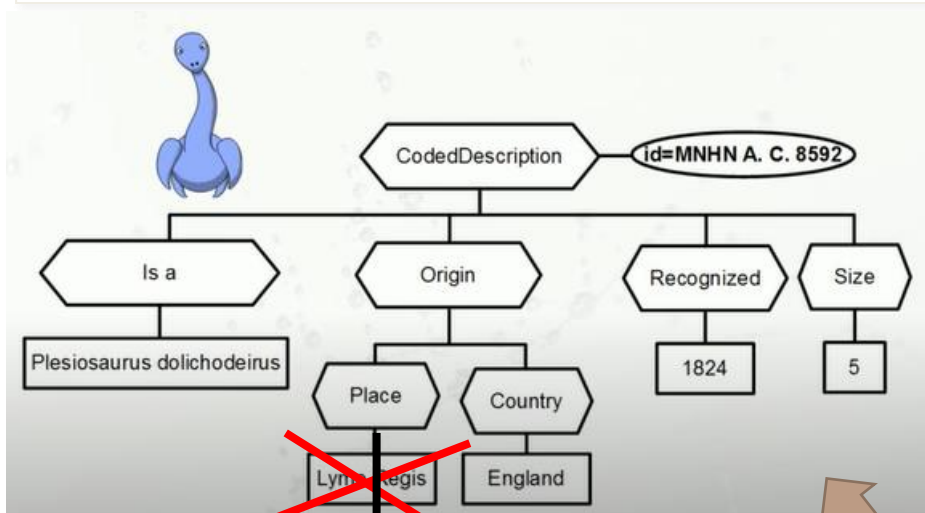
Estruturado X Semi-Estruturado

- **Estruturado**

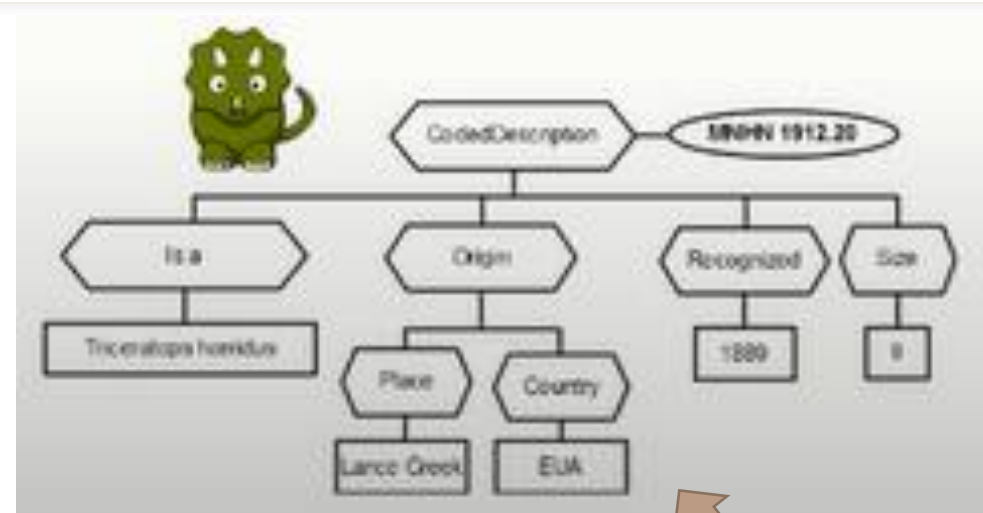
- Formato restrito
 - Ex: modelo relacional
- Cada registro segue o mesmo formato

- **Semi-estruturado**

- Itens de dados podem ter estruturas variadas
- Grupos de itens compartilham estruturas



Nova Versão do
Documento



Versão Antiga do
Documento

Banco de Dados - Documentos

- **XML-based**

- BaseX (<http://basex.org>)

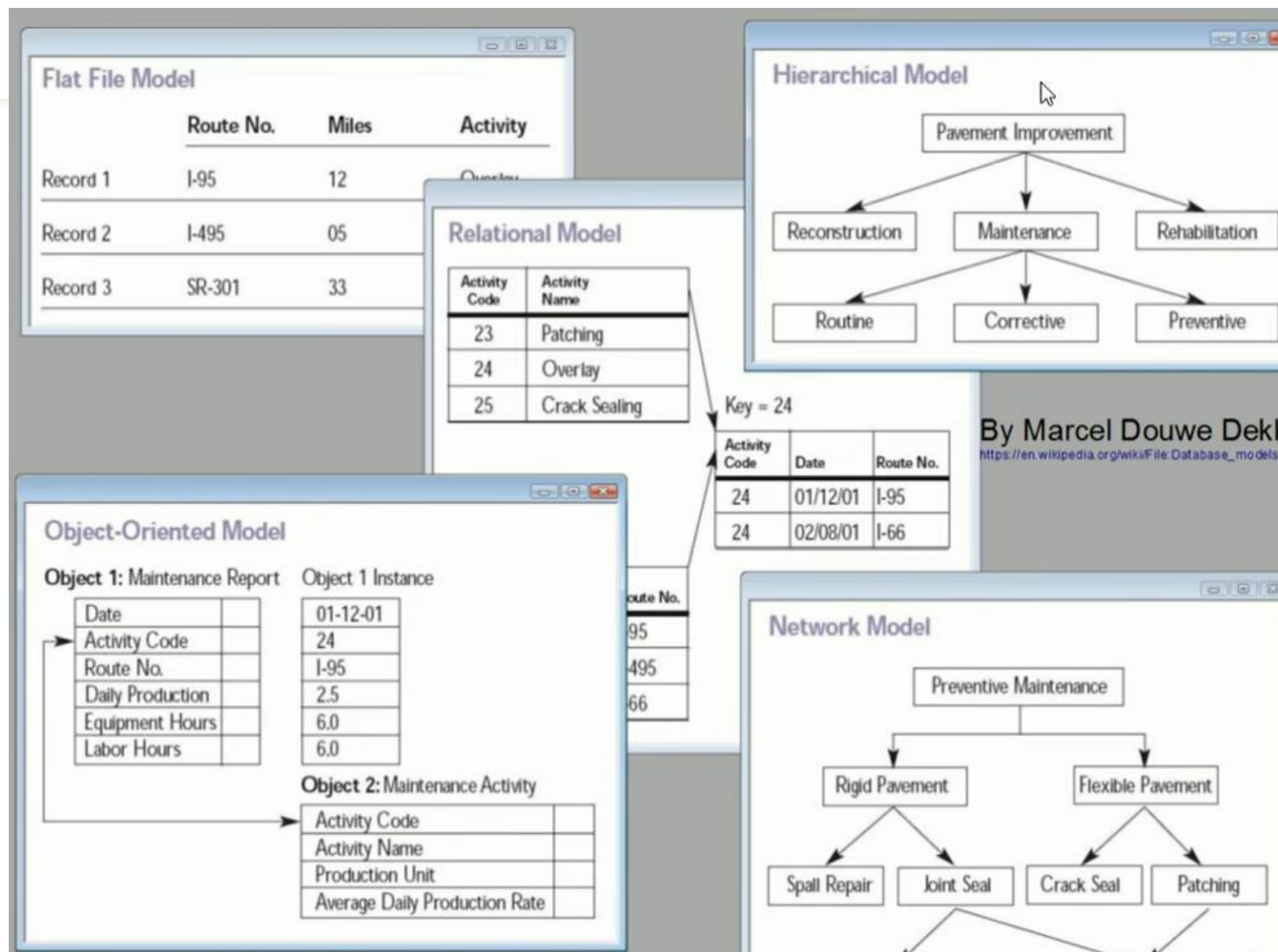
- **JSON**

- CouchDB (<http://couchdb.apache.org>)
 - MongoDB (<http://www.mongodb.org>)

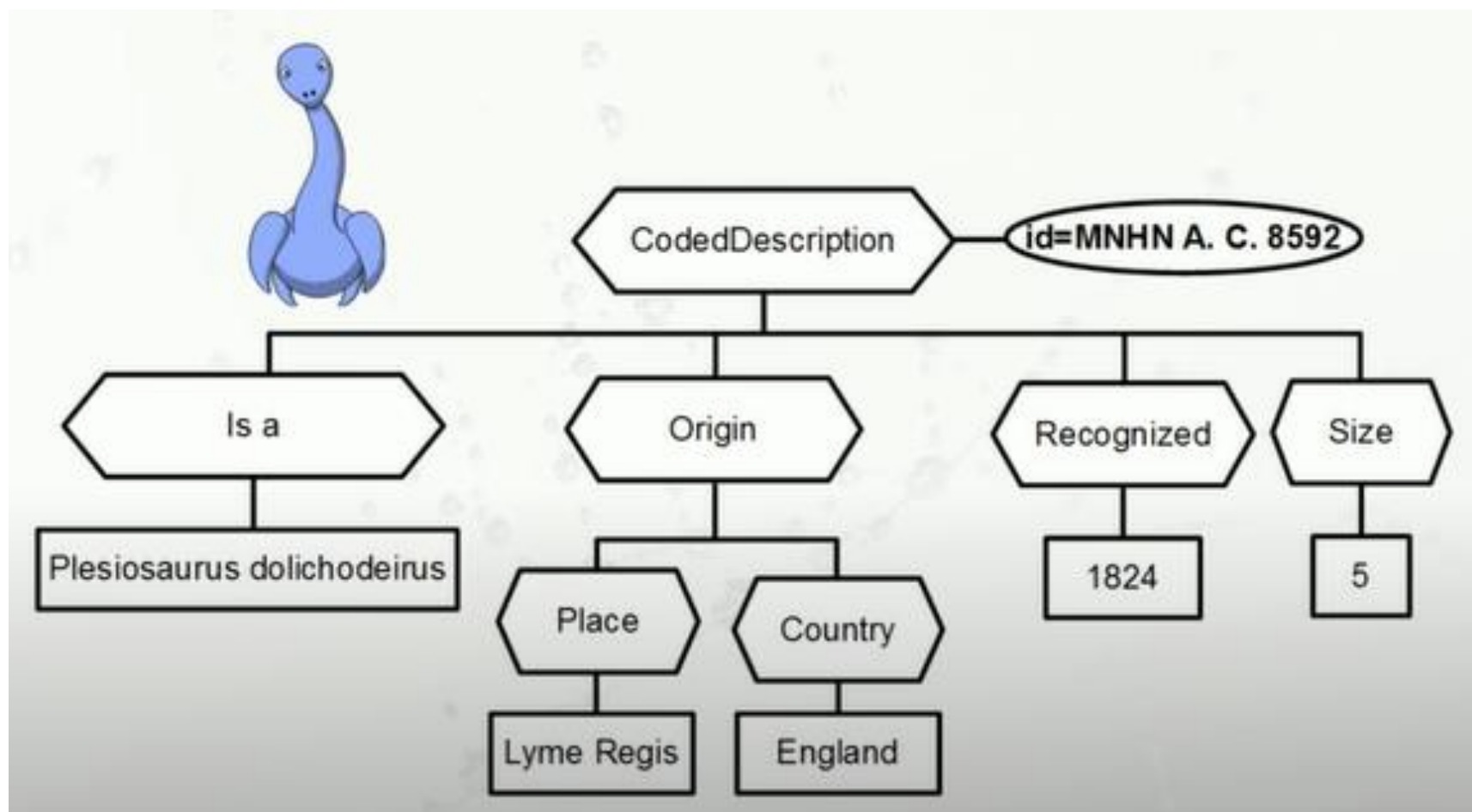
Modelo Lógico

- Em se tratando de Banco de Dados baseados em documentos qual seria a ideia do modelo lógico de BD baseados em documentos?
- **HIERARQUIA**

Modelo Lógico



Modelo Hierárquico (Documentos/XML)



Modelo Hierárquico (JSON – Documentos + OO)

```
{  
  "nome": "Asdrubal",  
  "livro": {  
    "titulo": "Horacio o dino",  
    "ano": 2010  
  }  
}
```



JavaScript

- Inventada por Brendan Eich na Netscape
- Incorporada no Internet Explorer com Jscript
- ECMA Script → padronização (ECMA, 2011)
- Originalmente para pequenos scripts no navegador
- Limitações de acesso ao hardware local
 - Segurança
 - Independência de plataforma

JSON (JavaScript Object Notation)

- Padrão aberto de intercâmbio de objetos
- Baseado na notação JavaScript
- Incorporado ao ECMA Script (Ecma, 2011)
- Adotado por diversas linguagens (<http://json.org/>)

Stringify

- **Serializando**

```
var pessoa = {  
    "nome": "Asdrubal",  
    "idade": 25  
};  
  
var pessoaStr = JSON.stringify(pessoa);
```

- **Desserializando**

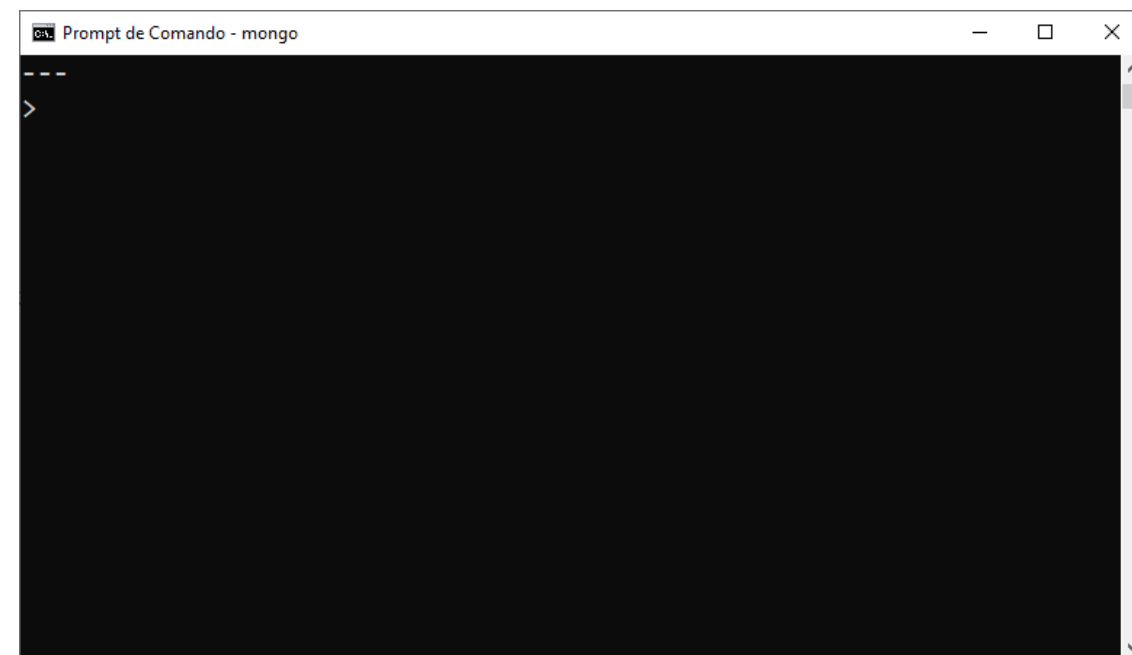
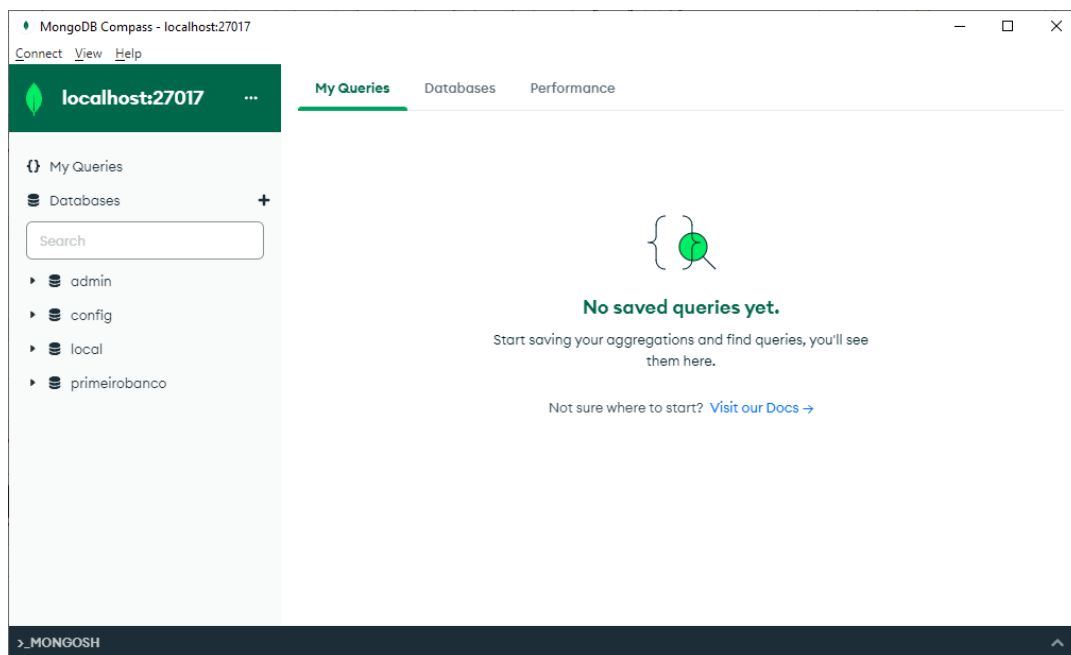
```
var pessoa2 = JSON.parse(pessoaStr);
```

MongoDB

- <http://www.mongodb.org>
- É um banco de dados que armazena nativamente objetos JSON
- Como instalar? ([Github](#))

Inicializando o Servidor MongoDB

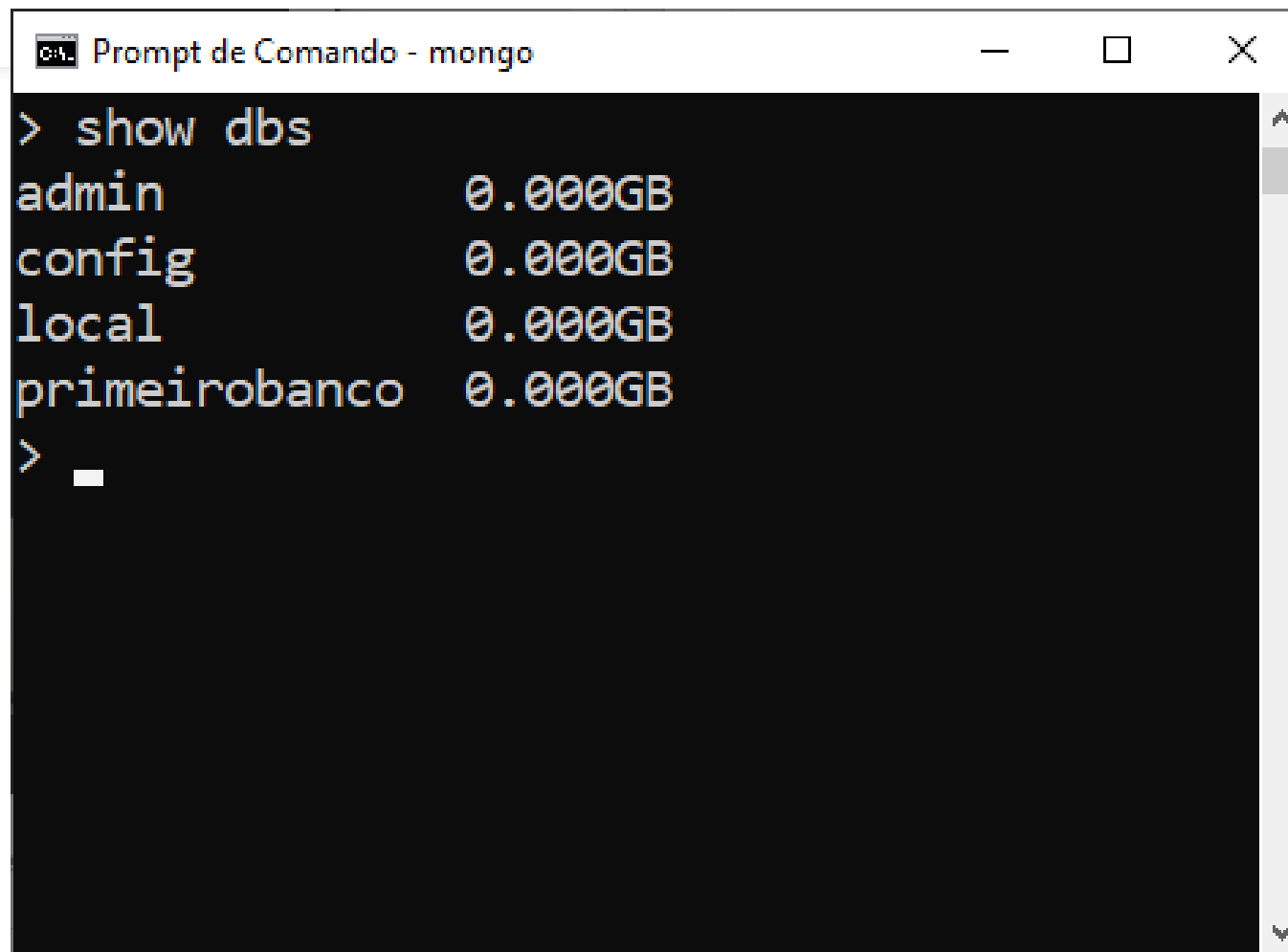
- Clique para executar o MongoDB Compass e clique em Conect
- Abra um terminal de Comando e digite mongo



Verificando todos os bancos

- Podemos verificar os bancos do sistema com: **show dbs**
- Este comando mostra todos os BDs criados até o momento
- Note que há alguns bancos do próprio Mongo já criados

Verificando todos os bancos



```
Prompt de Comando - mongo
> show dbs
admin          0.000GB
config         0.000GB
local          0.000GB
primeirobanco  0.000GB
> _
```


Criando um banco de dados

- Utilizamos para a criação do banco o comando `use nomeBanco`
- Este comando além de criar também já selecione este banco para o uso.
- Este comando também serve para realizar a troca do banco de dados.

```
> use bancofatec  
switched to db bancofatec  
>
```

Verificando o banco criado

```
> show dbs
admin          0.000GB
config         0.000GB
local          0.000GB
primeirobanco 0.000GB
> use bancofatec
switched to db bancofatec
> show dbs
admin          0.000GB
config         0.000GB
local          0.000GB
primeirobanco 0.000GB
>
```



Verificando o banco criado

O MongoDB só cria de fato o banco de dados quando há dados inseridos!



Criando uma Collection

```
Prompt de Comando - mongo
> db.primeiracollection.insertOne({ nome: "Anderson Vanin", idade: 48 })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f54be79511ac4ac8523af")
}
>
```

```
> show dbs
admin          0.000GB
bancofatec     0.000GB
config         0.000GB
local          0.000GB
primeirobanco 0.000GB
>
```

Encontrando um dado

 Prompt de Comando - mongo

```
> db.primeiracollection.findOne()
{
  "_id" : ObjectId("641f54be79511ac4ac8523af"),
  "nome" : "Anderson Vanin",
  "idade" : 48
}
> db.primeiracollection.findOne({})
{
  "_id" : ObjectId("641f54be79511ac4ac8523af"),
  "nome" : "Anderson Vanin",
  "idade" : 48
}
>
```

Exercício

- Insira um dado na última collection criada
- Faça uma seleção de dados para ver o dado inserido (comando find)

Exercício

```
> db.primeiracollection.insertOne({ nome: "Fulano da Silva", idade: 35, profissao: "Programador" })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f56bc79511ac4ac8523b0")
}
```

```
> db.primeiracollection.find()
{ "_id" : ObjectId("641f54be79511ac4ac8523af"), "nome" : "Anderson Vanin", "idade" : 48 }
{ "_id" : ObjectId("641f56bc79511ac4ac8523b0"), "nome" : "Fulano da Silva", "idade" : 35, "profissao" : "Programador" }
>
```

Exercício

```
> db.primeiracollection.find().pretty()
{
  "_id" : ObjectId("641f54be79511ac4ac8523af"),
  "nome" : "Anderson Vanin",
  "idade" : 48
}
{
  "_id" : ObjectId("641f56bc79511ac4ac8523b0"),
  "nome" : "Fulano da Silva",
  "idade" : 35,
  "profissao" : "Programador"
}
```


Visualizando em qual banco de dados estamos.

```
> db  
bancofatec
```

Criando outro Banco

- Inicie um novo banco de dados e insira um documento com algum valor e finalmente visualize o banco criado

```
> use segundobanco
switched to db segundobanco
> db.teste.insertOne({ a: 5})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f592479511ac4ac8523b1")
}
> show dbs
admin          0.000GB
bancofatec     0.000GB
config         0.000GB
local          0.000GB
primeirobanco  0.000GB
segundobanco   0.000GB
```

Criando outra Collection

```
> db  
segundobanco
```

```
> db.teste2.insertOne({ b: 10})  
{  
  "acknowledged" : true,  
  "insertedId" : ObjectId("641f5a5c79511ac4ac8523b2")  
}
```

Encontrando dados

- Para buscar dados utilizamos o comando: find
- Este comando pode receber filtros, para selecionarmos dados específicos
- Exemplo: `db.<collection>.find({nome:"João"})`
- Neste caso buscamos por um document com uma chave nome e um valor João

Encontrando dados

```
> db
segundobanco
> db.pessoas.insertOne({nome: "Maria"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5bb079511ac4ac8523b3")
}
> db.pessoas.insertOne({nome: "João"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5bb779511ac4ac8523b4")
}
```

Encontrando dados

```
> db.pessoas.find()
{ "_id" : ObjectId("641f5bb079511ac4ac8523b3"), "nome" : "Maria" }
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
>
```

```
> db.pessoas.find({nome:"João"})
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
```

Apesar de parecer muito com o comando LIKE do SQL, devemos ter um pouco mais de cuidado. Vamos inserir mais um document.

```
> db.pessoas.insertOne({nome: "João Amado"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5c8179511ac4ac8523b5")
}
```

Encontrando dados

Vamos usar o comando find para localizar o document com o valor João

```
> db.pessoas.find()
{ "_id" : ObjectId("641f5bb079511ac4ac8523b3"), "nome" : "Maria" }
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
{ "_id" : ObjectId("641f5c8179511ac4ac8523b5"), "nome" : "João Amado" }
>
```

```
> db.pessoas.find({nome:"João"})
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
>
```

Inserindo dados mais complexos

```
> db.pessoas.insertOne({nome: "Cintia", idade: 41, hobbies:["Programadora","Ler","Correr","Jogar"], profissao:
"Professora", saldo: 15000, trabalhando: true})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5e2479511ac4ac8523b6")
}
>
```

```
> db.pessoas.find()
{ "_id" : ObjectId("641f5bb079511ac4ac8523b3"), "nome" : "Maria" }
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
{ "_id" : ObjectId("641f5c8179511ac4ac8523b5"), "nome" : "João Amado" }
{ "_id" : ObjectId("641f5e2479511ac4ac8523b6"), "nome" : "Cintia", "idade" : 41, "hobbies" : [ "Programadora",
"Ler", "Correr", "Jogar" ], "profissao" : "Professora", "saldo" : 15000, "trabalhando" : true }
>
```


Inserindo dados mais complexos

```
> db.pessoas.find().pretty()
{ "_id" : ObjectId("641f5bb079511ac4ac8523b3"), "nome" : "Maria" }
{ "_id" : ObjectId("641f5bb779511ac4ac8523b4"), "nome" : "João" }
{ "_id" : ObjectId("641f5c8179511ac4ac8523b5"), "nome" : "João Amado" }
{
  "_id" : ObjectId("641f5e2479511ac4ac8523b6"),
  "nome" : "Cintia",
  "idade" : 41,
  "hobbies" : [
    "Programadora",
    "Ler",
    "Correr",
    "Jogar"
  ],
  "profissao" : "Professora",
  "saldo" : 15000,
  "trabalhando" : true
}
>
```

Criação de uma Collection implícita

- Há a possibilidade de criar a collection com um comando também;
- Exemplo: `db.createCollection("nome",{opções})`
- Podemos definir alguns parâmetros de configuração como: número máximo de registros, tamanho máximo da collection, etc.

Criação de uma Collection implícita

```
> db.createCollection("alunos",{capped: true, size: 1000, max: 3})  
{ "ok" : 1 }
```

Criação de uma Collection implícita

```
> db.alunos.insertOne({nome: "a1"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5ffa79511ac4ac8523b7")
}
> db.alunos.insertOne({nome: "a2"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f5fff79511ac4ac8523b8")
}
> db.alunos.insertOne({nome: "a3"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f600379511ac4ac8523b9")
}
>
```

```
> db.alunos.find()
{ "_id" : ObjectId("641f5ffa79511ac4ac8523b7"), "nome" : "a1" }
{ "_id" : ObjectId("641f5fff79511ac4ac8523b8"), "nome" : "a2" }
{ "_id" : ObjectId("641f600379511ac4ac8523b9"), "nome" : "a3" }
>
```

Criação de uma Collection implícita

Agora veja o que acontece quando inserirmos mais um registro, visto que a nossa coleção está limitada a 3 registros

```
> db.alunos.insertOne({nome: "a4"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f609779511ac4ac8523ba")
}
>
```

Não recebemos nenhuma mensagem de erro. Porém, veja seus registros com o comando find.

```
> db.alunos.find()
{ "_id" : ObjectId("641f5ffff79511ac4ac8523b8"), "nome" : "a2" }
{ "_id" : ObjectId("641f600379511ac4ac8523b9"), "nome" : "a3" }
{ "_id" : ObjectId("641f609779511ac4ac8523ba"), "nome" : "a4" }
>
```

Exibindo todas as Collections

- Para exibir todas as collections utilizamos: show collections
- Este comando de verificação nos ajuda a entender melhor o banco de dados
- Lembrando que para uma collection ser criada de fato, ela precisa ter algum dado inserido!

```
> db
segundobanco
> show collections
alunos
pessoas
teste
teste2
>
```

Exercício

- Crie uma collection com dados de nome de pessoas e salários
- Utilize o comando find para verificar os registros dela
- Verifique todas as collections do banco

Resolução

```
> db.salarios.insertOne({nome: "Anderson", salario: 10000})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f629c79511ac4ac8523bb")
}
> db.salarios.insertOne({nome: "Maria", salario: 7500})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f62aa79511ac4ac8523bc")
}
> db.salarios.insertOne({nome: "Fernanda", salario: 7000})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("641f62c279511ac4ac8523bd")
}
> db.salarios.find()
{ "_id" : ObjectId("641f629c79511ac4ac8523bb"), "nome" : "Anderson", "salario" : 10000 }
{ "_id" : ObjectId("641f62aa79511ac4ac8523bc"), "nome" : "Maria", "salario" : 7500 }
{ "_id" : ObjectId("641f62c279511ac4ac8523bd"), "nome" : "Fernanda", "salario" : 7000 }
>
```


Resolução

```
> show collections
alunos
pessoas
salarios
teste
teste2
>
```