



Banco de Dados NoSQL

Tiago Alves de Oliveira

Sumário

Introdução

Categorias

MongoDB

Firebase

Introdução

01

No SQL = NOT
ONLY SQL

02

O termo foi usado
pela primeira vez
em 1998.

03

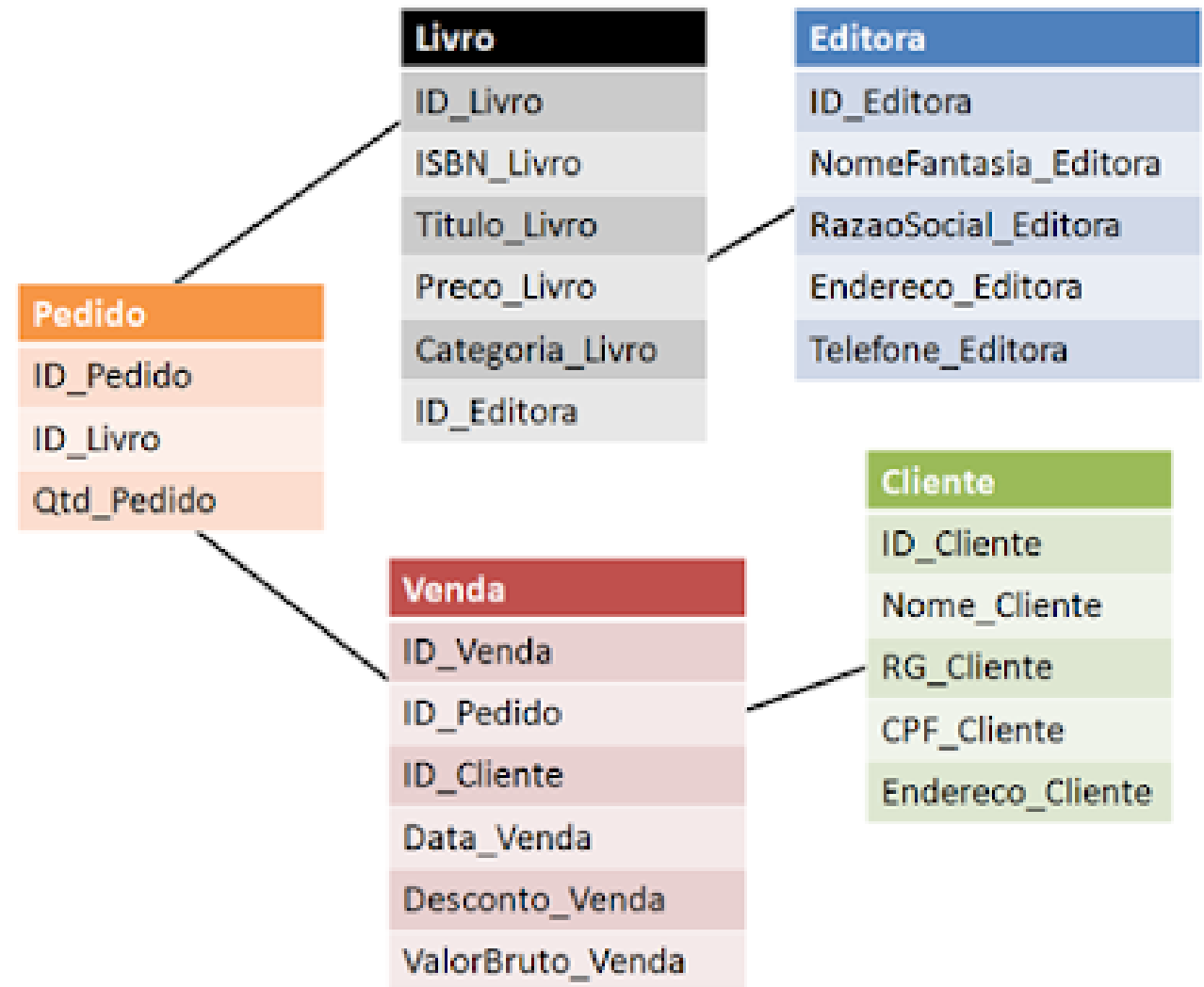
Era um nome de
banco relacional de
código aberto que
não possuía uma
interface SQL

04

SQL -> Linguagem
Estruturada para
Querys

Introdução

- SQL é baseado em tabelas e relacionamentos:



Introdução

01

Mas por que então
NoSQL?

02

Surgiram como
uma alternativa
para os banco de
dados relacionais.

03

Devido a alta
escalabilidade e
desempenho

Introdução

01

Surgiram da necessidade de uma performance superior e de uma alta escalabilidade.

02

Os bancos de dados relacionais atuais são muito restritos a isso.

03

Sendo necessária a distribuição vertical de servidores.

04

Isto é, quanto mais dados, mais memória e mais disco um servidor precisa.

Categorias dos Banco de Dados Não-Relacionais

01

Armazenamento
Chave-Valor

02

Armazenamento
column Family

03

Orientado a
Documentos

04

Orientados a
Grafos

Armazenamento Chave-Valor

Usa uma tabela hash na qual há uma chave única e um indicador de um dado ou de um item em particular



Bancos dessa categoria:

Redis

Oracle BDB

Armazenamento Column Family

Utilizado para armazenar e processar grandes quantidades de dados distribuídos em muitas máquinas.



Bancos dessa categoria:

Cassandra

HBase

Orientado a Documentos

São documentos versionados que são coleções de outras coleções de chave-valor.



Os documentos semi-estruturados são armazenados em formatos como JSON.



Bancos dessas Categorias:

MongoDB

CouchDB

Orientados a grafos

São uma de muitas categorias de bancos de dados propostas nos últimos anos.

A grande diferença para o modelo clássico está na representação explícita de relacionamentos entre os dados.

Para isso utiliza de um modelo com vértices, chamados de nós, e arcos, chamados de relações.

Bancos de Dados dessa categorias:

Neo4J

InfoGrid

Resumindo

- Com o grande crescimento do volume de dados em determinadas organizações, os bancos de dados NoSQL tem se tornado uma grande alternativa quando nos referimos a escalabilidade e disponibilidade, fatores que se tornam imprescindíveis em algumas aplicações WEB.

Porque escolher NoSQL

Dificuldades para modelar bancos complexos de forma relacional.

Como exemplo, pensem um banco de dados de uma rede social.

Necessidade de armazenamento de dados com esquemas flexíveis.



Onde obter
Informações

- <http://nosql-database.org/>



Quem usa Banco de
Dados NoSQL



The
Weather
Channel

NETFLIX

Quem usa Bancos de
Dados NoSQL



MongoDB

MongoDB é um banco de dados de código aberto, gratuito, de alta performance, sem esquemas e orientado à documentos.

Foi lançado em fevereiro de 2009 pela empresa 10gen.

Foi escrito na linguagem de programação C++ (o que o torna portátil para diferentes sistemas operacionais) .

Seu desenvolvimento durou quase 2 anos, tendo iniciado em 2007.

MongoDB

Por ser orientado à documentos JSON (armazenados em modo binário, apelidado de JSON), muitas aplicações podem modelar informações de modo muito mais natural.

Isso acontece pois os dados podem ser aninhados em hierarquias complexas e ainda serem indexáveis e fáceis de buscar, igual ao que já é feito em JavaScript.

Existem vários bancos NOSQL atualmente no mercado porque existem dezenas de problemas de persistência de dados que o SQL tradicional não resolve

MongoDB

Bancos não-relacionais document-based (que armazenam seus dados em documentos), são os mais comuns e mais proeminentes de todos.

O seu maior expoente o banco MongoDB como o gráfico da pesquisa mais recente de bancos de dados utilizados pela audiência do StackOverflow em 2018 mostra.

O link é o seguinte: <http://bit.ly/2PLKfeh> .

MongoDB

Dentre todos os bancos não relacionais, o MongoDB é o mais utilizado com um quinto de todos os respondentes alegando utilizar ele em seus projetos.

Isto é mais do que até mesmo o Oracle, um banco muito mais tradicional.

Basicamente, neste tipo de banco (document-based ou document-oriented), temos coleções de documentos, nas quais cada documento é autossuficiente.

Ele contém todos os dados que possa precisar, ao invés do conceito de não repetição + chaves estrangeiras do modelo relacional.

MongoDB

A ideia é que você não tenha de fazer JOINS pois eles prejudicam muito a performance em suas queries (são um mal necessário no modelo relacional, infelizmente).

Você modela a sua base de forma que a cada query você vai uma vez no banco e com apenas uma chave primária pega tudo que precisa.

Obviamente, isto tem um custo: armazenamento em disco.

Não é raro bancos MongoDB consumirem muitas vezes mais disco do que suas contrapartes relacionais.

MongoDB

MongoDB foi criada com Big Data em mente.

Ele suporta tanto escalonamento horizontal quanto vertical usando replica sets (instâncias espelhadas) e sharding (dados distribuídos)

Isto o torna uma opção muito interessante para grandes volumes de dados, especialmente os desestruturados.

MongoDB

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

MongoDB

Dados desestruturados são um problema para a imensa maioria dos bancos de dados relacionais, mas não tanto para o MongoDB.

Quando o seu schema é variável, é livre, usar MongoDB vem muito bem a calhar.

Os documentos BSON (JSON binário) do Mongo são schemaless e aceitam quase qualquer coisa que você quiser armazenar, sendo um mecanismo de persistência perfeito para uso com tecnologias que trabalham com JSON nativamente, como JavaScript (e consequentemente Node.js).

MongoDB

Cenários altamente recomendados e utilizados atualmente são catálogos de produtos de e-commerces.

Telas de detalhes de produto em ecommerces são extremamente complicadas devido a diversidade de informações aliada às milhares de variações de características entre os produtos que acabam resultando em dezenas de tabelas se aplicado sobre o modelo relacional.

Em MongoDB essa problemática é tratada de uma maneira muito mais simples.

MongoDB

Além do formato de documentos utilizado pelo MongoDB ser perfeitamente intercambiável com o JSON serializado do JS, MongoDB opera basicamente de maneira assíncrona em suas operações, assim como o próprio Node.js

Isto nos permite ter uma persistência extremamente veloz, aliado a uma plataforma de programação igualmente rápida.

Embora o uso de Node.js com bancos de dados relacionais não seja incomum, é com os bancos não-relacionais como MongoDB e Redis que ele mostra todo o seu poder de tecnologia para aplicações real-time e volumes absurdos de requisições na casa de 500 mil/s, com as configurações de servidor adequadas.

MongoDB

Além disso, do ponto de vista do desenvolvedor, usar MongoDB permite criar uma stack completa apenas usando JS.

Uma vez que temos JS no lado do cliente, do servidor (com Node) e do banco de dados (com Mongo), pois todas as queries são criadas usando JS também, como você verá mais à frente.

MongoDB

Nem tudo são flores e o MongoDB não é uma “bala de prata”, ele não resolve todos os tipos de problemas de persistência existentes.

Você não deve utilizar MongoDB quando relacionamentos entre diversas entidades são importantes para o seu sistema.

Se for ter de usar muitas “chaves estrangeiras” e “JOINS”, você está usando do jeito errado, ou, ao menos, não do jeito mais indicado.

MongoDB

Além disso, diversas entidades de pagamento (como bandeiras de cartão de crédito), não homologam sistemas cujos dados financeiros dos clientes não estejam em bancos de dados relacionais tradicionais.

Obviamente, isso não impede completamente o uso de MongoDB em sistemas financeiros, mas o restringe apenas a certas partes (como dados públicos).

MongoDB

Diversos players de cloud computing fornecem versões de Mongo hospedadas e prontas para uso como Umblar e mLab, no entanto, é muito importante um conhecimento básico de administração local de MongoDB para entender melhor como tudo funciona.

Não focaremos aqui em nenhum aspecto de segurança, de alta disponibilidade, de escala ou sequer de administração avançada de MongoDB.

Deixo todas estas questões para você ver junto à documentação oficial no site oficial, onde inclusive você pode estudar e tirar as certificações.

MongoDB

Não serão detalhados aqui o processo de instalação do MongoDB.

Para maiores testes acesse o site oficial do MongoDB e baixe gratuitamente a versão mais recente para o seu sistema operacional

MongoDB

Dentro dessa pasta do Mongo podem existir outras pastas, mas a que nos interessa é a pasta bin.

Nessa pasta estão uma coleção de utilitários de linha de comando que são o coração do MongoDB (no caso do Windows, todos terminam com .exe):

MongoDB

mongod: inicializa o servidor de banco de dados;

mongo: inicializa o cliente de banco de dados;

mongodump: realiza dump do banco (backup binário);

mongorestore: restaura dumps do banco (restore binário);

mongoimport: importa documentos JSON ou CSV pro seu banco;

mongoexport: exporta documentos JSON ou CSV do seu banco;

MongoDB

Para nosso minicurso iremos utilizar o mLab:



Acesse o link:

<https://mlab.com>

MongoDB

[WELCOME](#)[PLANS & PRICING](#)[DOCS & SUPPORT](#)[ACCOUNT](#)[LOG OUT](#)

{ user: "[tiagoalvesdiv](#)", account: "[tiagoalvesdiv](#)" }

[Home](#)

Your email address needs to be verified. [Send an email verification](#) again to "tiagoalvesdiv@gmail.com".

MongoDB Deployments

[Create from backup](#)[+ Create new](#)

None exist at this time. Click "Create new" to create a fully managed MongoDB deployment on AWS, Azure, or Google.

Environments

[+ Create new](#)

None exist at this time. Click "Create new" to create an [mLab Environment](#) (VPC).

[help](#)


MongoDB




[Home](#)

MongoDB Deployments


None exist at this time. Click "Create new" to create a fully managed MongoDB deployment on AWS, Azure, or Google.

 Create from backup

 Create new

Environments

None exist at this time. Click "Create new" to create an [mLab Environment](#) (VPC).

 Create new

MongoDB

The screenshot shows the mLab MongoDB deployment interface. Red annotations highlight key steps: a red circle around the 'Cloud Provider' section with 'Amazon Web Services' selected (marked with a red '1'); a red circle around the 'Plan Type' section with 'SANDBOX' selected (marked with a red '2'); and a red circle around the 'CONTINUE' button at the bottom right (marked with a red '3').

mLab

Home

Cloud Provider

amazon web services

Google Cloud Platform

Microsoft Azure

■ USE SSDS ON DEDICATED PLANS (AZURE 2) ⓘ

New Deployment Creator

PLAN TYPE
Sandbox

PROVIDER
Amazon Web Services

Plan Type

SANDBOX **FREE**
For learning, developing, or prototyping. Up to 0.5 GB storage.

SHARED **\$15 / GB**
For small datasets and light production workloads. Up to 8 GB storage.

DEDICATED **\$180 & UP**
For demanding production workloads. Supports unlimited vertical and horizontal scaling (via sharding).

AVAILABLE AWS REGIONS

- US East (Virginia) (us-east-1)
- Europe (Ireland) (eu-west-1)

cancel **CONTINUE**

© 2018 ObjectLabs Corporation Terms of Service Privacy Policy

MongoDB

mLab

[Home](#)

AWS Region


Q Search regions


US East (Virginia) (us-east-1)

Europe (Ireland) (eu-west-1)

US East (Virginia) (us-east-1)

New Deployment Creator

PLAN TYPE 
Sandbox

PROVIDER 
Amazon Web Services

REGION
US East (Virginia) (us-east-1)

cancel BACK CONTINUE

© 2018 ObjectLabs Corporation Terms of Service Privacy Policy

MongoDB



[Home](#)

Final Details

MONGODB VERSION 3.6.7 (MMAPv1)

1

DATABASE NAME

minicursomongo

New Deployment Creator

PLAN TYPE

Sandbox

PROVIDER

Amazon Web Services

REGION

US East (Virginia) (us-east-1)

PLAN LINE & SIZE

- 0.5 GB

MONGODB VERSION

3.6.7 (MMAPv1)

DATABASE NAME

minicursomongo

2

MongoDB



[Home](#)

Order Confirmation


PLAN TYPE	Sandbox	
CLOUD PROVIDER	Amazon Web Services	
REGION	US East (Virginia) (us-east-1)	
PLAN LINE & SIZE	Sandbox	FREE
	STORAGE 0.5 GB	
MONGODB VERSION	3.6.7 (MMAPv1)	
DATABASE NAME	minicursomongo	
Total Price	FREE	


MongoDB




[Home](#)

MongoDB Deployments


 Create from backup

 Create new

Development and Utility Single-node deployments intended for environments that do not require high availability.

>	DEPLOYMENT	TYPE	RAM	SIZE ⓘ	SIZE ON DISK ⓘ
>	 ds261332/minicursomongo	Sandbox	shared	0.00 KB	0.00 KB

Environments

 Create new


None exist at this time. Click "Create new" to create an [mLab Environment](#) (VPC).

MongoDB



[Home](#)

Database: minicursomongo

 Delete database


To connect using the mongo shell:

```
% mongo ds261332.mlab.com:61332/minicursomongo -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
mongodb://<dbuser>:<dbpassword>@ds261332.mlab.com:61332/minicursomongo
```

mongod version: 3.6.7 (MMAPv1)

 Sandbox databases do not have redundancy and therefore [are not suitable for production](#). Read our documentation on [how to upgrade](#).

Collections


Users

Stats

Backups

Tools

Collections

 Add collection

[None at this time]


System Collections

[None at this time]

MongoDB

[Home](#)

Database: minicursomongo

 Delete database


To connect using the mongo shell:

```
% mongo ds261332.mlab.com:61332/minicursomongo -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
mongodb://<dbuser>:<dbpassword>@ds261332.mlab.com:61332/minicursomongo
```

mongod version: 3.6.7 (MMAPv1)

 Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on [how to upgrade](#).

Collections

Users

Stats

Backups

Tools

Database Users

 Add database user

NAME	READ ONLY?
tiago	false


C:\Windows\system32\cmd.exe

```
C:\Program Files\MongoDB\Server\4.0\bin>mongo ds261332.mlab.com:61332/minicursomongo -u tiago -p 1q2w3e4r
```

MongoDB

[Home](#)

Database: minicursomongo

 Delete database


To connect using the mongo shell:

```
% mongo ds261332.mlab.com:61332/minicursomongo -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
mongodb://<dbuser>:<dbpassword>@ds261332.mlab.com:61332/minicursomongo
```

mongod version: 3.6.7 (MMAPv1)

 Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on [how to upgrade](#).

Collections

Users

Stats

Backups

Tools

Database Users

 Add database user

NAME	READ ONLY?
tiago	false


C:\Windows\system32\cmd.exe

```
C:\Program Files\MongoDB\Server\4.0\bin>mongo ds261332.mlab.com:61332/minicursomongo -u tiago -p 1q2w3e4r
```

MongoDB

[Home](#)

Database: minicursomongo

 Delete database

To connect using the mongo shell:

```
% mongo ds261332.mlab.com:61332/minicursomongo -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
|mongodb://<dbuser>:<dbpassword>@ds261332.mlab.com:61332/minicursomongo
```

mongod version: 3.6.7 (MMAPv1)



Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on [how to upgrade](#).

Collections


Users


Stats


Backups

Tools

Collections

 Delete all collections

 Add collection

NAME	DOCUMENTS	CAPPED?	SIZE 
cliente	1	false	16.20 KB



System Collections

NAME	DOCUMENTS	SIZE
system.indexes	4	0.44 KB

Firebase

O *Firebase* é uma plataforma móvel do Google que ajuda você a expandir seu negócio e desenvolver rapidamente apps de alta qualidade.

Ele nos fornece uma plataforma de ferramentas e serviços, que com apenas alguns passos, nos podemos ter uma funcionalidade como o *Push Notification* configurado e em produção.

Os serviços do firebase podem ser divididos em dois grupos: desenvolvimento e crescimento.



Firebase

- **Desenvolvimento**
 - [Realtime Database](#)
 - [Auth](#)
 - [Test Lab](#)
 - [Crashlytics](#)
 - [Cloud Functions](#)
 - [Firestore](#)
 - [Cloud Storage](#)
 - [Performance Monitoring](#)
 - [Crash Reporting](#)
 - [Hosting](#)



Firebase

- **Crescimento**
- [Firebase Analytics](#)
- [Invites](#)
- [Cloud Messaging](#)
- [Predictions](#)
- [AdMob](#)
- [Dynamic Links](#)
- [Adwords](#)
- [Remote Config](#)
- [App Indexing](#)

Firebase

Para que você possa utilizar alguns desses serviços, você precisa ter um conta google como Gmail.

Caso você já tenha uma, basta clicar no link:

<https://firebase.google.com>

E passar os dados de sua conta.

O próximo passo será clicar no botão do canto superior do site go to console.

Firestore

O Firestore tem uma versão gratuita que já atende muitos cenários, mas caso você tenha interesse e explorar os outros planos, segue um link para o Firestore Pricing:

<https://firebase.google.com/pricing/?authuser=0>

Firestore

A documentação do Firestore pode ser encontrada em:

<https://firebase.google.com/docs/?authuser=0>

Links importantes

Robot 3T: <https://robomongo.org/download>

MongoDB: <https://www.mongodb.com/>

Umbler: <https://www.umbler.com/br>

Livros: <http://www.luiztools.com.br/meus-livros/>

Livro Mongo: <http://www.luiztools.com.br/livro-mongodb/>

Firebase: <https://firebase.google.com/>

Obrigado!

Contato:

tiago.alves@uemg.br

www.tiagoalves.com.br