



# Banco de Dados

Prof. Dr. Alan Souza

[alan.souza@unama.br](mailto:alan.souza@unama.br)

2020

## Sumário - NoSQL



1. Introdução
2. Diferenças entre SQL e NoSQL
3. Usando o MongoDB
4. Exercícios

## Introdução



- NoSQL = **Not only SQL** = Não somente SQL;
- Não é composto por tabelas com linhas e colunas;
- Na maioria dos casos, não usa a linguagem SQL para definir, manipular e controlar os dados;
- Exemplos de bancos de dados NoSQL: MongoDB, Cassandra, Berkeley DB, Redis, Firebase, Neo4J e vários outros...
- A principal vantagem é a performance;
- A principal desvantagem é a inconsistência.

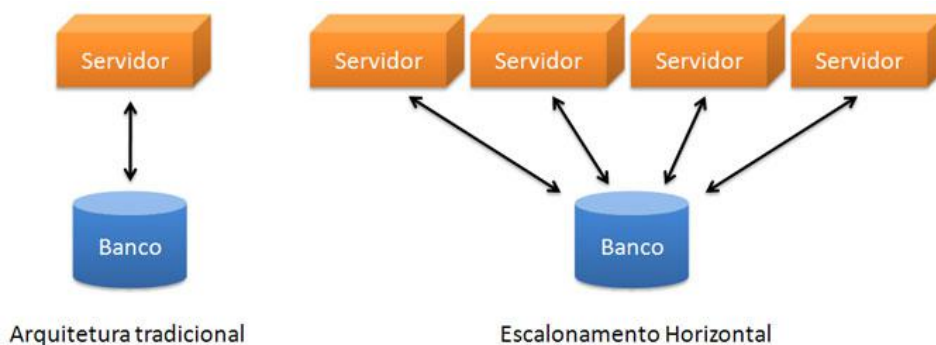
## Introdução



- Aplicações de alta disponibilidade, por exemplo:
  - Redes sociais: Facebook, Instagram, Twitter...
  - Comunicação: Whatsapp, Hangouts...
  - Cadastros: Enquete.
  - Games: CartolaFC.
- Exemplos específicos:
  - Uma celebridade realiza uma postagem no Twitter que é mostrada, compartilhada e curtida por milhares de outras contas quase em tempo real!
  - Envio de mensagens em um grupo do Whatsapp de 100 pessoas que recebem a mensagem quase em tempo real e ao mesmo tempo.

## Introdução

- Um BD NoSQL pode ser distribuído em larga escala, o que permite escalabilidade, alta disponibilidade, alto desempenho e confiabilidade.



## Diferenças entre SQL e NoSQL

- Comparação entre modelo relacional e modelo NoSQL:

Característica	Modelo relacional	NoSQL
<b>Escalação</b>	É estruturado, portanto, difícil de aumentar o escalonamento.	Não é estruturado, sendo relativamente fácil de aumentar o escalonamento. É a principal vantagem.

## Diferenças entre SQL e NoSQL



- Comparação entre modelo relacional e modelo NoSQL:

Característica	Modelo relacional	NoSQL
<b>Consistência</b>	O ACID implementa isso por padrão no BD. Rigidez, sendo o ponto forte do modelo relacional.	Característica eventual (pode ter ou não). Portanto, não há garantia de consistência. Ponto fraco.

## Diferenças entre SQL e NoSQL



- Comparação entre modelo relacional e modelo NoSQL:

Característica	Modelo relacional	NoSQL
<b>Disponibilidade</b>	O servidor pode ficar sobrecarregado com muitos acessos simultâneos.	A distribuição dos nós em vários servidores diferentes permite a alta disponibilidade do BD. Mais um ponto positivo dessa abordagem.

## Diferenças entre SQL e NoSQL



- Comparação entre modelo relacional e modelo NoSQL:

Característica	Modelo relacional	NoSQL
Tolerância e particionamento	O BD relacional é único e indivisível.	Trabalha de forma fácil e eficiente com a distribuição de dados. É capaz de suporte grandes demandas, sendo bastante tolerante à falhas.

## Diferenças entre SQL e NoSQL



- Comparação entre modelo relacional (MySQL) e modelo NoSQL (MongoDB), em relação as estruturas:

MySQL	MongoDB
tabela	coleção
índice	índice
linha	documento BSON
coluna	campo
joins	documentos incorporados e vinculação
primary key	primary key (definida automaticamente)
group by	aggregation

## Usando o MongoDB



- Banco de Dados NoSQL MongoDB
- Site oficial: [mongodb.com](https://mongodb.com)
- Instalação: <https://docs.mongodb.com/manual/installation/>
- A versão community é free
- Documentação excelente!
- Universidade MongoDB: <https://university.mongodb.com/>

## Usando o MongoDB



```

{
  na
  ag
  st
  gr
}
{
  na
  ag
  st
  gr
}
{
  name: "al",
  age: 18,
  status: "D",
  groups: [ "politics", "news" ]
}
  
```

**Coleção**

## Usando o MongoDB



- O Mongo Shell é usado para trabalhar com o MongoDB:

```
marcel@leqram: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
marcel@leqram:~$ mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
Server has startup warnings:
2020-05-18T07:36:31.903-0300 I STORAGE [initandlisten]
2020-05-18T07:36:31.903-0300 I STORAGE [initandlisten] ** WARNING: Using the XFS
filesystem is strongly recommended with
the WiredTiger storage engine
2020-05-18T07:36:31.903-0300 I STORAGE [initandlisten] ** See http://doc
hub.mongodb.org/core/prodnotes-filesystem
2020-05-18T07:36:33.127-0300 I CONTROL [initandlisten]
2020-05-18T07:36:33.127-0300 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2020-05-18T07:36:33.127-0300 I CONTROL [initandlisten] ** Read and write
access to data and configuration is unrestricted.
2020-05-18T07:36:33.127-0300 I CONTROL [initandlisten]
>
```

## Usando o MongoDB



produto					
<u>cod</u>	nome	altura	potencia	peso	diametro
1	Britadeira		300	80	
2	Cimento				
3	Empilhadeira		15	300	
4	Escada	3			
5	Tubulação				20

- Vamos comparar como faríamos para criar a tabela, inserir e manipular os dados acima em MySQL e em MongoDB.

## Usando o MongoDB



Criação da estrutura	
MySQL	MongoDB
create	-
<pre>create table produto (   cod int not null primary key   auto_increment,   nome varchar(100) not null,   altura decimal(10,2),   potencia decimal (10,2),   peso decimal (10,2),   diametro decimal(10,2) );</pre>	<p>A coleção é criada automaticamente na primeira inserção de dados. O cod (chave-primária) é adicionado automaticamente se não for especificado.</p>

## Usando o MongoDB



Destruição da estrutura	
MySQL	MongoDB
drop	drop
<pre>drop table produto;</pre>	<pre>db.produto.drop();</pre>

- *Mais detalhes sobre drop:*

<https://docs.mongodb.com/manual/reference/method/db.collection.drop/>



## Usando o MongoDB



Inserção de dados	
MySQL	MongoDB
insert	insert
<code>insert into produto</code> (nome, altura, potencia, peso, diametro) <code>values</code> ('Britadeira', null, 300, 80, null), ('Cimento', null, null, null, null), ('Empilhadeira', null, 15, 300, null), ('Escada', 3, null, null, null), ('Tubulação', null, null, null, 20);	<code>db.produto.insert ( {cod: 1, nome: "Britadeira", potencia: 300, peso: 100});</code> <code>db.produto.insert ( {cod: 2, nome: "Cimento"} );</code> <code>db.produto.insert ( {cod: 3, nome:"Empilhadeira", potencia: 15, peso: 300} );</code> <code>db.produto.insert ( {cod: 4, nome: "Escada", altura: 3} );</code>

- *Mesmo não tendo o valor (null), a linha como um todo ocupa espaço no disco (MySQL). Não temos esse problema no MongoDB. Tabelas esparsas.*

## Usando o MongoDB



Seleção de dados	
MySQL	MongoDB
select	find
<code>select * from produto;</code>	<code>db.produto.find();</code>
<code>select * from produto where nome = 'Escada';</code>	<code>db.produto.find( { nome: "Escada" } );</code>
<code>select * from produto where peso &gt; 200;</code>	<code>db.produto.find( { peso: { \$gt: 200 } } );</code> gt: <i>greater than</i> (maior que)
<code>select * from produto where potencia &lt; 100;</code>	<code>db.produto.find( { peso: { \$lt: 100 } } );</code> lt: <i>less than</i> (menor que)

- *Mais detalhes sobre find:*  
<https://docs.mongodb.com/manual/reference/method/db.collection.find/>

## Usando o MongoDB



Funções de agregação	
MySQL	MongoDB
count	count
<code>select count(*) from produto;</code>	<code>db.produto.count();</code>
<code>select count(*) from produto where peso &gt; 200;</code>	<code>db.produto.find(   { peso: { \$gt: 200 } } ) .count();</code>
avg	avg
<code>select avg(peso) from produto;</code>	<code>db.produto.aggregate( { \$group: { "_id": null, "media_peso": { \$avg: "\$peso" } } } );</code>

- *Mais operadores de agregação na documentação:*  
<https://docs.mongodb.com/manual/reference/operator/aggregation/>

## Usando o MongoDB



Atualização de dados	
MySQL	MongoDB
update	update
<code>update produto set altura = 4.5 where cod = 4;</code>	<code>db.produto.update(   { cod: 4 } ,   { \$set: { altura: 4.5 } },   { multi: false } );</code>

- *Mais detalhes sobre update:*  
<https://docs.mongodb.com/manual/reference/method/db.collection.update/>

## Usando o MongoDB



Remoção de dados	
MySQL	MongoDB
delete	remove
<code>delete from produto where cod = 1;</code>	<code>db.produto.remove(   { cod: 1 } );</code>

- *Mais detalhes sobre remove:*

<https://docs.mongodb.com/manual/reference/method/db.collection.remove/>

## Usando o MongoDB



Resumo de comandos de CRUD do MongoDB.

Inserção de dados:

```
db.users.insertOne(  ← collection
  {
    name: "sue",      ← field: value
    age: 26,          ← field: value
    status: "pending" ← field: value } document
  )
```

## Usando o MongoDB



Resumo de comandos de CRUD do MongoDB.

Seleção de dados:

```
db.users.find(
  { age: { $gt: 18 } },
  { name: 1, address: 1 }
).limit(5)
```

← collection  
← query criteria  
← projection  
← cursor modifier

## Usando o MongoDB



Resumo de comandos de CRUD do MongoDB.

Atualização de dados:

```
db.users.updateMany(
  { age: { $lt: 18 } },
  { $set: { status: "reject" } }
```

← collection  
← update filter  
← update action  
)

## Usando o MongoDB



Resumo de comandos de CRUD do MongoDB.

Remoção de dados:

```
db.users.deleteMany(  
  { status: "reject" }  
)
```

← collection  
← delete filter

## Exercícios



- 1) Cite três diferenças entre os bancos de dados relacionais e não relacionais.
- 2) Cite três exemplos de bancos de dados relacionais e três não relacionais.
- 3) Um restaurante decidiu criar uma aplicação web para que seus clientes realizem pedidos sem sair de casa. Em média, eles recebem 50 pedidos por dia. Baseado nesse cenário, é melhor usar um banco de dados relacional ou não relacional? Justifique sua resposta.

## Exercícios



4) Baseado na tabela abaixo, faça os itens do próximo slide.

aluno					
matricula	nome	nota1	nota2	nota3	situacao
260108	Maria Silva	5.5	4.5	5.0	Indefinido
260109	Caio Soares	6.7	7.3		Indefinido
260110	Fábio Mendes	7.4	8.3		Indefinido
260111	Flávia Lopes	9.4	9.1		Indefinido
260112	Laura Barata	8.3	2.9	8.7	Indefinido

## Exercícios



Continuação da 4a questão. No MongoDB, quais comandos servem para:

- Criar a coleção.
- Exibir os alunos que tiraram nota maior que 7.0 na primeira avaliação.
- Mostrar a quantidade de alunos que tiraram nota menor 5.0 na segunda avaliação.
- Revelar a média de todas as três avaliações.
- Alterar a situação de todos os alunos para Aprovado.
- Remover um aluno a sua escolha.

## Exercícios



Continuação...

- g) Revelar a maior e a menor nota da segunda avaliação.
- h) Inserir mais um aluno com dados a sua escolha.
- i) Mostrar todos os dados em ordem alfabética.
- j) Mostrar os dados ordenados da maior para a menor nota considerando a primeira avaliação.