

Curso de Ciência da Computação

Banco de Dados 2
Aula 08

Prof. Dr. rer. nat. Eros Comunello

Banco de Dados Não Relacional

NoSQL Orientado a Documentos

MongoDB

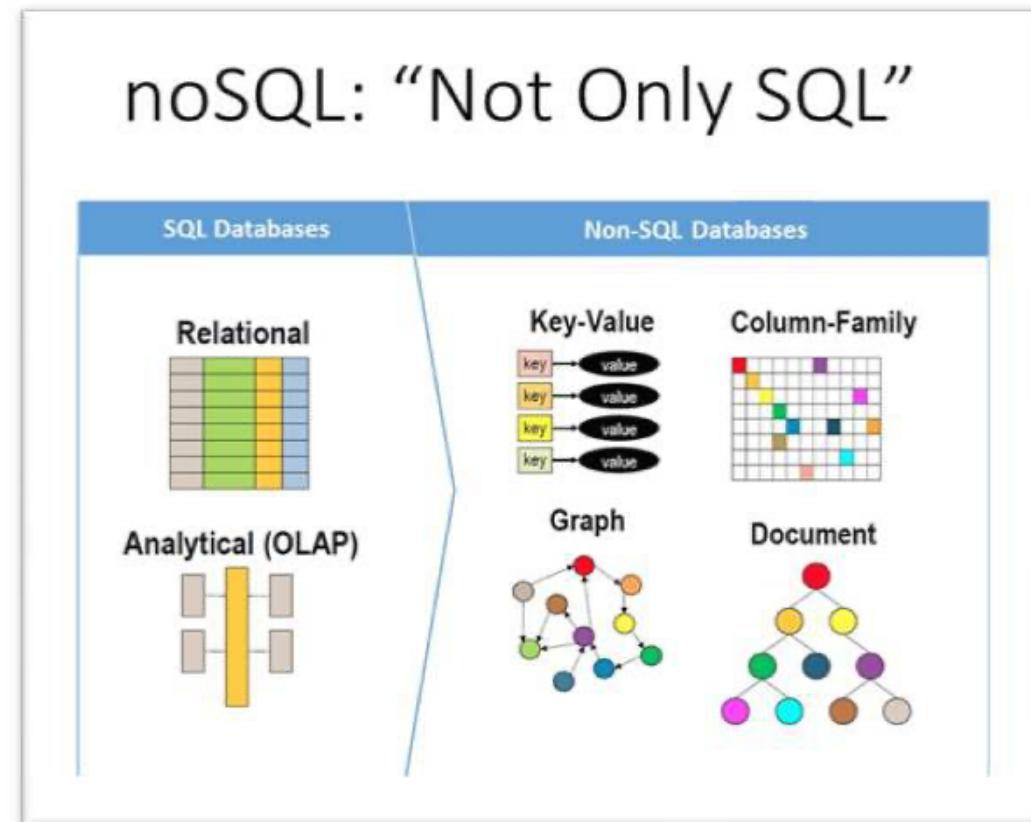
Instalação e Testes

Exemplos

Backup e Restore

Tipos de BD NoSQL

- Orientada a documentos
- Key-Value
- Família de colunas
- Grafos



NoSQL Orientados a documentos

❑ Bancos de dados orientados a documentos:

- ❑ Os documentos dos bancos dessa categoria, são **coleções de atributos e valores**, onde um atributo pode ser multivlorado;
- ❑ Em geral, eles **não possuem esquema**, ou seja, os documentos armazenados não precisam possuir estrutura em comum;
- ❑ Essa característica faz deles boas opções para o **armazenamento de dados semiestruturados**;
- ❑ Alguns bancos que utilizam esse padrão são: **MongoDB**, CouchDB, RavenDb, etc.

```
{  
    "id": 49,  
    "Pais": "Alemanha",  
    "Regiao": "Europa",  
    "Populacao":  
    "PrincipaisCidades": [  
        {  
            "NomeCidade": "Berlin",  
            "Populacao": 3610156,  
        },  
        {  
            "NomeCidade": "Hamburg",  
            "Populacao": 1746342,  
        }  
    ]  
}
```

MongoDB

- ❑ Orientado a documentos JSON (armazenados em modo binário, apelidado de BSON);
- ❑ Os dados podem ser aninhados em hierarquias complexas e continuar a ser indexáveis e fáceis de buscar, igual ao que já é feito em JavaScript;
- ❑ Dentre todos os bancos não relacionais o MongoDB é o mais utilizado;
- ❑ Temos coleções de documentos, nas quais cada documento é autossuficiente, 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 (necessários em BD relacionais);
- ❑ 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;
- ❑ Custo alto no armazenamento em disco;
 - ❑ Bancos MongoDB tendem a consumir muitas vezes mais disco do que suas contrapartes relacionais.
- ❑ **Importante:** no site oficial você pode estudar e tirar as certificações de forma gratuita.

Quando usar MongoDB?

- ❑ Ele suporta tanto escalonamento horizontal quanto vertical usando replica sets (instâncias espelhadas) e sharding (dados distribuídos);
 - ❑ Opção muito interessante para grandes volumes de dados, especialmente os desestruturados.
- ❑ Dados desestruturados são um problema para BD relacionais, mas não tanto para o MongoDB;
- ❑ Quando o seu schema é variável, é livre, utilize MongoDB;

Quando usar MongoDB?

- ❑ Os documentos BSON (JSON binário) do Mongo são schemaless e aceitam quase qualquer coisa que você quiser armazenar;
- ❑ Um mecanismo de persistência perfeito para uso com tecnologias que trabalham com JSON nativamente, como JavaScript (e consequentemente Node.js).
- ❑ Altamente recomendados em catálogos de produtos de e-commerces;
 - ❑ Telas de detalhes de produto em e-commerces são extremamente complicadas devido à diversidade de informações aliada às milhares de variações de características entre os produtos;

Quando **não** usar MongoDB?

- ❑ Não deve utilizar MongoDB quando relacionamentos entre diversas entidades são importantes para o seu sistema;
- ❑ Diversas entidades de pagamento (como bandeiras de cartão de crédito) não homologam sistemas cujos dados financeiros dos clientes não estejam em BD relacionais tradicionais;
- ❑ Não impede completamente o uso de MongoDB em sistemas financeiros, mas o restringe apenas a certas partes (como dados públicos).

MongoDB – Instalação e testes

- ❑ Diversos players de cloud computing fornecem versões de Mongo hospedadas e prontas para uso;
- ❑ Faça o download no site: <http://www.mongodb.org/>
- ❑ Baixe o arquivo compactado e, no caso do Windows, rode o executável que extrairá os arquivos na sua pasta de Arquivos de Programas;
 - ❑ não há uma instalação de verdade, apenas extração de arquivos;
- ❑ Na pasta bin estão uma coleção de utilitários de linha de comando que são o coração do MongoDB
 - ❑ no caso do Windows, todos terminam com .exe

MongoDB – Instalação e testes

- ❑ **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;
- ❑ entre outros.

MongoDB – Instalação e testes

- ❑ Para subir um servidor de Mongo:

c:\mongo\bin> mongod --dbpath C:\mongo\data

- ❑ O servidor está executando corretamente, sem segurança alguma e na porta padrão 27017;

c:\mongo\bin> mongo

- ❑ No MongoDB você não precisa construir a estrutura do seu banco previamente antes de sair utilizando ele;

MongoDB – Instalação e testes

❑ Comandos básicos:

- > *show databases ou show dbs*
- > *use nomedobanco*
- > *show collections*
- > *db.createCollection('nomeCollection')*
- > *db.nomeCollection.find()*
- > *db.nomeCollection.find().pretty()*
- > *db.nomeCollection.find().limit(NUMBER)*
- > *db.nomeCollection.find().sort({KEY:1})*
- > *db.nomeCollection.insert({ })*
- > *db.nomeCollection.update({ })*
- > *db.nomeCollection.updateOne({ }, { :set { } })*
- > *db.nomeCollection.remove({ })*
- > *db.nomeCollection.drop()*
- > *db.nomeCollection.save({_id:ObjectId(), NOVODADO})*

MongoDB – Exemplos

- `db.academico.insert({nome: "Hilson", idade: 20, curso: "Ciência da Computação"})`
- `db.academico.insert([{nome: "Douglas", idade: 21, curso: "Ciência da Computação"}, {nome: "Gaby", curso: "Ciência da Computação"}])`
- `db.academico.find().pretty()`
- `db.universidade.insert({nome: "Univali", campus: "Kobrasol"})`
- `db.universidade.updateOne({nome: "Univali"}, {$set: {campus: "Florianópolis"}})`

MongoDB – Exemplos

- `db.universidade.find({nome: "Univali"}).pretty()`
- `db.universidade.insert([{nome: "Univali", campus: "Kobrasol"}, {nome: "UFSC", campus: "Trindade"}])`
- `db.universidade.insert({nome: "UNIVALI", operando: true})`
- `db.universidade.find()`
- `db.universidade.remove({_id: ObjectId("5ecf8dc6401e67258f871719")})`

Clausula WHERE no MongoDB

Operação	Syntaxe	Exemplo	Equivalentes
Equality	{<key>:<value>}	db.mycol.find({"by":"tutorials point"}).pretty()	where by = 'tutorials point'
Less Than	{<key>:{\$lt:<value>}}	db.mycol.find({"likes":{\$lt:50}}).pretty()	where likes < 50
Less Than Equals	{<key>:{\$lte:<value>}}	db.mycol.find({"likes":{\$lte:50}}).pretty()	where likes <= 50
Greater Than	{<key>:{\$gt:<value>}}	db.mycol.find({"likes":{\$gt:50}}).pretty()	where likes > 50
Greater Than Equals	{<key>:{\$gte:<value>}}	db.mycol.find({"likes":{\$gte:50}}).pretty()	where likes >= 50
Not Equals	{<key>:{\$ne:<value>}}	db.mycol.find({"likes":{\$ne:50}}).pretty()	where likes != 50

Comando AND no MongoDB

Sintaxe:

```
> db.mycol.find({key1:value1, key2:value2}).pretty()
```

Exemplo:

```
>db.mycol.find()  
{ "_id": ObjectId("7df78ad8902c"),  
  "title": "MongoDB - Guia Rapido",  
  "description": "MongoDB - Guia Rapido",  
  "by": "MongoDBWise",  
  "url": "http://www.mongodbwise.wordpress.com",  
  "tags": ["mongodb", "database", "NoSQL"],  
  "likes": "100"}>
```

Comando OR no MongoDB

Sintaxe:

```
>db.mycol.find({  
    $or:[{key1: value1},{key2:value2}  
]}.pretty()
```

Exemplo:

```
> db.mycol.find({$or:[{"by":"tutorials point"}, {"title":"MongoDB  
Overview"}]}).pretty()  
{"_id":ObjectId("7df78ad8902c"),  
 "title":"MongoDB - Guia Rapido",  
 "description":"MongoDB - Guia Rapido",  
 "by":"MongoDBWise",  
 "url":"http://www.mongodbwise.wordpress.com",  
 "tags":["mongodb","database","NoSQL"],  
 "likes":100}>
```

Comando AND e OR juntos

Exemplo:

```
>db.mycol.find("likesgt:10}, $or: [{"by": "MongoDBWise"}, {"title": "MongoDB - Guia Rapido"}] }).pretty()  
{  
  "_id": ObjectId("7df78ad8902c"),  
  "title": "MongoDB - Guia Rapido",  
  "description": "MongoDB - Guia Rapido",  
  "by": "MongoDBWise",  
  "url": "http://www.mongodbwise.wordpress.com",  
  "tags": ["mongodb", "database", "NoSQL"],  
  "likes": "100"  
}
```

MongoDB – Backup e Restore

❑ Backup

```
./mongodump --out /Applications/mongodb-macos-x86_64-  
4.2.7/backup/
```

❑ Restore

```
./mongorestore -h <host> -d <database> -u <usuário> -p <senha>  
/Applications/mongodb-macos-x86_64-4.2.7/backup/academico.bson
```

Exercícios

- ❑ *Instale o MongoDB;*
- ❑ *Crie uma coleção de dados;*
- ❑ *Utilize todos os comandos apresentados;*
- ❑ *Faça um backup dos dados.*

Links complementares

- ❑ <https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>
- ❑ <https://micreiros.com/tipos-de-bancos-de-dados-nosql/>
- ❑ <https://www.luiztools.com.br/post/tutorial-mongodb-para-iniciantes-em-nosql/>
- ❑ <https://mongodbwise.wordpress.com/2014/05/22/mongodb-guia-rapido/>