



Denise Garcia Telli
Squad Canais Backend
LT Agibank 02/08/2021



- Lançado em 2009
- Código aberto
- Escrito em C++
- É um banco de dados NoSQL, oferece alto desempenho, alta disponibilidade e fácil escalabilidade
- Trabalha com o conceito de **coleções e documentos**
- Armazena dados em documentos do tipo JSON flexíveis - o que significa que os campos podem variar de documento para documento e a estrutura de dados pode ser alterada ao longo do tempo
- Ao utilizar esquemas dinâmicos, é possível criar registros sem predefinir nada e a estrutura de um registro pode ser alterada simplesmente adicionando novos campos ou excluindo os existentes

Terminologia

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key <code>_id</code> provided by mongodb itself)

Indexação

O MongoDB Java Driver fornece a classe *Indexes* que inclui 'static factory methods' para criar 'index specification documents' para os vários tipos de chave de índice MongoDB.

OBS: O MongoDB só cria um índice se ainda não existir um índice com a mesma especificação.

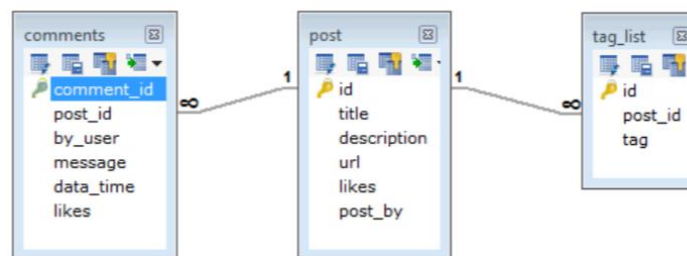
- Por padrão, o MongoDB cria um índice para o `_id` para cada coleção.
- Como os índices são armazenados na RAM, eles ocupam algum espaço e podem causar sobrecarga em cada inserção, atualização e exclusão.
- Uma coleção não pode ter mais de 64 índices.
- O comprimento do nome do índice não pode ultrapassar 125 caracteres.
- Um índice composto pode ter no máximo 31 campos indexados.

mongoDB e Java

Apesar do Java ser bem conhecido por ter a coleção mais robusta de bibliotecas Object Relational Mapping (ORM) para lidar com dados relacionais, usar documentos em Java é bem simples.

O MongoDB utiliza documentos JSON para seu armazenamento de dados, porém quando estamos integrando ele com o Java temos uma classe que abstrai o json: classe *Document* (que implementa a interface *Bson*).

Exemple of modeling



```
{
  _id: POST_ID,
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```

Setup

1. Baixar mongoDB em <https://www.mongodb.com/try/download/community>
2. Criar uma pasta separada “db” para rodar o servidor
3. Dentro da pasta bin, subir o BD mongo na pasta “db”
`>.\mongod --dbpath pathParaAPastadb`
4. Dentro da pasta bin, subir o banco
`>.\mongo`

Gradle

Adicionar ao build.gradle:

```
java compile 'org.mongodb:mongodb-driver-sync:4.0.5'
```

Maven

Adicionar nas dependências do arquivo pom.xml:

```
<dependency>  
    <groupId>org.mongodb</groupId>  
    <artifactId>mongodb-driver-sync</artifactId>  
    <version>4.0.5</version>  
</dependency>
```

Conexão

MongoClient

```
MongoClient client = new MongoClient();
```

- A instância MongoClient () representa um pool de conexões com o banco de dados; você só precisará de uma instância da classe MongoClient, mesmo com vários threads.
- É possível instanciá-lo sem parâmetros (ele fará a conexão local na porta 27017) ou especificar a porta e/ou o host.

MongoDatabase

```
MongoDatabase database = client.getDatabase( databaseName: "test");
```

- Se um banco de dados não existir, o MongoDB criará o banco de dados quando você primeiro armazenar dados para esse banco de dados.
- 'test' é o banco de dados padrão

```
database.createCollection( s: "myCollection");
```

- Se uma coleção não existe, o MongoDB cria a coleção quando você armazena os dados para essa coleção pela primeira vez

**Talk is cheap. Show
me the code.**

March

Linus Torvalds

THANKS!

Referências

- <https://pt.wikipedia.org/wiki/MongoDB>
 - <https://www.mongodb.com/pt-br/>
 - <https://docs.mongodb.com/manual/>
- https://docs.google.com/presentation/d/1gHAp-jaVSgAkix0xxGDmhQG55JFFIs_-Pi1cnhcWh4/edit#slide=id.p
- <https://docs.mongodb.com/drivers/java/sync/current/fundamentals/crud/>
- <https://www.alura.com.br/curso-online-criando-uma-webapp-com-java-e-mongodb>
- <https://mongodb.github.io/mongo-java-driver/3.4/javadoc/>