



Programação Web 1



# Objetivos de aprendizagem

- Diferenciar bancos de dados NoSQL e SQL
- Iniciar a utilização do MongoDB
- Criar conta MongoDB Atlas

# Agenda

- SQL x NoSQL
- MongoDB
- CRUD

# SQL x NoSQL

# SQL

- RDBMS (*Relational Database Management System*)
- Dados estruturados em tabelas relacionadas entre si
- Acesso e manipulação dos dados é feita utilizando SQL (*Structured Query Language*)

# SQL

## Vantagens

- Abstração de dados independente de implementação
- Dados otimizados para armazenamento em disco
- Utilização de recursos otimizada
- Maior consistência e integridade dos dados (atomicidade, transações, durabilidade, etc)

# SQL

## Desvantagens

- Necessidade de planejamento prévio da estrutura de dados, dificuldade em modificações
- Modelos de dados rígidos

# SQL

Quando usar SQL?

- Recomendado para dados com relacionamento bem definido
- Quando a integridade de dados é crítica

# NoSQL

*Not only SQL*

- Desenvolvido na década de 1990, mas evidenciado em 2000 com a evolução da Computação em Nuvem, Internet, uso de dispositivos móveis e similares
- Surgiu da necessidade de bancos de dados com menores restrições e mais escaláveis

NoSQL Oracle Brasil

---

# NoSQL

## Vantagens

- Alta escalabilidade eliminando pontos de falha únicos
- Modelos de dados flexíveis
- Criação e mudança podem ser feitas "on the fly"
- Alta performance devido à simplicidade da implementação comparada a RDBMS

# NoSQL

## Desvantagens

- Menor disponibilidade de funcionalidades relacionadas à confiabilidade dos dados

# NoSQL

Quando usar NoSQL?

- Conjuntos de dados mais flexíveis que não se enquadram ou não necessitam de relacionamentos
- Dados não estruturados
- Acesso rápido a grandes volumes de dados

# Exemplos

## SQL

- Oracle 
- PostgreSQL 
- MySQL 
- SQLite 

## NoSQL

- Redis 
- Fauna 
- CouchDB 
- Cassandra 
- Elasticsearch 
- MongoDB 

MongoDB



"MongoDB is an open source, nonrelational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store various forms of data."

## What is MongoDB?

---

# Características

- Unidade básica de dado consiste em **documentos**
- Os documentos são agrupados em **coleções**
- Formatação básica em **JSON**
- Flexibilidade total na criação/consulta dos registros
- Confiável
- Altamente escalável

"A JSON object is an unordered set of name/value pairs..."

# Base de documentos

- Um registro é chamado de **documento**
- Os documentos são estruturados em pares campo/valor, JSON

```
1  {  
2      "name" : "sue",  
3      "age" : 26,  
4      "status" : "A",  
5      "groups" : [ "news", "sports" ]  
6 }
```

# Por que utilizar documentos?

- Correspondem aos dados nativos utilizados nas linguagens, facilmente processáveis
- Favilitam a interoperabilidade
- Polimorfismo de dados é facilitado
- Reduzem a necessidade de *joins* de alto custo computacional

# Estrutura de Documentos

- Os nomes de campos são `strings`
- Os valores podem assumir diversos tipos, inclusive outros documentos e `arrays` de documentos
- Tipos BSON (*Binary Encoded JSON*)

```
1 var mydoc = {  
2     _id: ObjectId("37238237"),  
3     name: {first: "Allan", last: "Turing"},  
4     birth: new Date('Jun 23, 1912'),  
5     death: new Date('Jun 07, 1954'),  
6     contribs: ["Turing Machine", "Turing test", "Turingery"],  
7     views: NumberLong(1250000)  
8 }
```

# Estrutura de Documentos

```
1  {
2      "type": "icecream",
3      "name": "Vanilla Cone",
4      "image":
5          {
6              "url": "img/01.png",
7              "width": 200,
8              "height": 200
9          },
10     "thumbnail":
11         {
12             "url": "images/thumbnails/01.png",
13             "width": 32,
14             "height": 32
```

# BSON Types

Formato Binário de Serialização

Referência BSON

---

# ObjectId

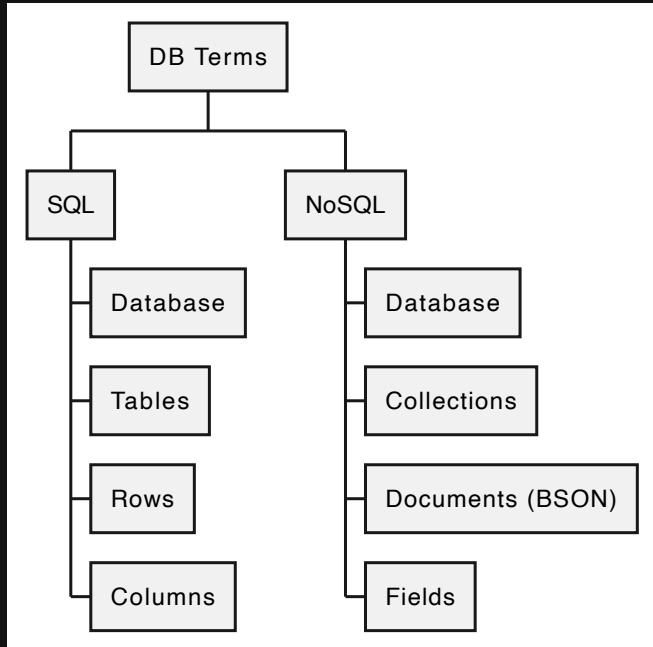
- MongoDB requer que cada documento de uma coleção possua um campo `_id` único que funciona como chave primária
- Se omitido, é gerado automaticamente
- `_id` é um nome de campo reservado, não pode ser atribuído a nenhum outro campo

```
1 var mydoc = {  
2     _id: ObjectId("37238237"),  
3     name: {first: "Allan", last: "Turing"},  
4     birth: new Date('Jun 23, 1912'),  
5     death: new Date('Jun 07, 1954'),  
6     contribs: ["Turing Machine", "Turing test", "Turingery"],  
7     views: NumberLong(1250000)  
8 }
```

# Coleções

- Um conjunto de documentos é chamado de **coleção** (*collection*)
- *Collections* são equivalentes a tabelas em RDBMS

# Resumo



# CRUD

# CRUD

- Create
- Read
- Update
- Delete

# Create

## insertOne()

```
1 db.student.insertOne(  
2     {  
3         Name: "Akshay",  
4         Marks: 500  
5     })
```

# Create

## insertMany()

```
1 db.student.insertMany(  
2   [  
3     {name:"Ajay", age:20},  
4     {name:"Bina", age:24},  
5     {name:"Ram", age:23}  
6   ])
```

# Read

## find()

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

# Read

## Exemplos

```
1 db.inventory.find({})
2 db.inventory.find({ status: "D" })
3 db.inventory.find({ $or: [ { status: "A" }, { qty: { $lt: 30 } } ] })
```

Equivalente SQL:

```
1 SELECT * FROM inventory
2 SELECT * FROM inventory WHERE status = "D"
3 SELECT * FROM inventory WHERE status = "A" OR qty < 30
```

# Update

```
1 db.collection.updateOne(<filter>, <update>, <options>)
2 db.collection.updateMany(<filter>, <update>, <options>)
3 db.collection.replaceOne(<filter>, <update>, <options>)
```

# Delete

```
1 db.collection.deleteMany()  
2 db.collection.deleteOne()
```

# Perguntas

# Exercícios



2

Criar conta MongoDB Atlas usando o email institucional através do link abaixo.

[Create Your Account](#)

---

"MongoDB Atlas is a multi-cloud database service..."

[MongoDB Atlas Docs](#)

---

# 3

Configurar sua conta no MongoDB Atlas.

1. Criar conta (já feito no exercício anterior)
2. Implantar *cluster* gratuito
3. Configurar IP para acesso
4. Configurar senha de acesso
5. Identificar sua *connection string*

Links úteis:

- [Deploy cluster](#)  
-----
- [Add IP](#)  
-----

# 4

Despois de criada a conta, execute os passos a seguir.

1. Criar bases de dados de amostra
  - Clusters -> ... -> Load Sample Dataset
2. Criar um novo banco de dados ( `sample_data` ) no seu *cluster* recém-criado e uma nova collection ( `countries` )
  - + Create Database
3. Adicionar documentos na *collection* conforme o arquivo `countries.json`

# 5

1. Criar banco de dados `db`
2. Adicionar coleção `restaurants`
3. Importar `restaurants.json` (Ver Google Classroom da Aula 9)
  - Via `mongosh` ou via MongoDB Compass
4. Após importar `restaurants.json`, resolver os exercícios:
  - MongoDB Exercises

---

# Referências

- Insert Documents
- Query Documents
- Update Documents
- Remove Documents

# Referências Adicionais

- IBM MongoDB
- MongoDB Básico
- BSON
- JSON.org

# José Roberto Bezerra

✉️ [jbroberto@ifce.edu.br](mailto:jbroberto@ifce.edu.br)

⌚ [jbroberto76](#)

⌚ [pw1-repo](#)



Powered by  Sliddev

Cover image by José Campos from 