MongoDB

BANCO DE DADOS NÃO RELACIONAL ORIENTADO A DOCUMENTOS

BANCO DE DADOS AVANÇADOS

VALÉRIA TIMES

MongoDB

- Origem: Humongous
  - Significa **Gigantesco**

- Alguém conhece MongoDB?
- Sim! Já trabalhou?

MongoDB

- O que é?
  - Banco de dados (BD) não relacional orientado a documentos
  - Não relacional?
    - Não existe a necessidade de criar uma estrutura de tabelas e dados antes de começar a inserir informações
  - Bancos de Dados relacionais – Você sabe previamente o que vai inserir
    - Normalização
    - CREATE TABLE “NOME DA TABELA”
    - Definição dos tipos de dados
    - Ex:
      - Nome varchar
      - DtNascimento date
      - Idade int

MongoDB

- Criado pelos fundadores da Doubleclick
  - Empresa de anúncios da internet comprada pelo Google
- Criada em 2007 como “10gen”. Em 2013 foi alterado o nome para MongoDB Inc;
- É um dos bancos NoSQL mais populares do mundo;
- Baixa curva de aprendizagem
  - Erros (Tentar implementar pensando relacional)

MongoDB

Bancos de Dados Relacional	Bancos de Dados Orientado a Documentos
Bancos de dados	Bancos de dados
Tabelas	Coleções (Collections)
Linhas	Documentos
Colunas	Campos

MongoDB

- Banco de Dados Relacional
  - Cada coluna continua reservado no BD aguardando um valor
- Orientado a documentos
  - Cada documento pode ser inserido na mesma Coleção com número de campos distintos

MongoDB

- Alguns usuários
  - Mercado livre
  - EasyTaxi
  - Zap Imóveis
  - Petrobrás
  - Terra
  - SAP

MongoDB

- Pra que serve?
  - Mesmo tendo um conceito diferente, é um banco de dados
    - Armazenar informações
  - Utilizado quando o modelo estrutural não é adequado
  - Cada banco de dados é mais adequado pra cada situação
    - Ex:

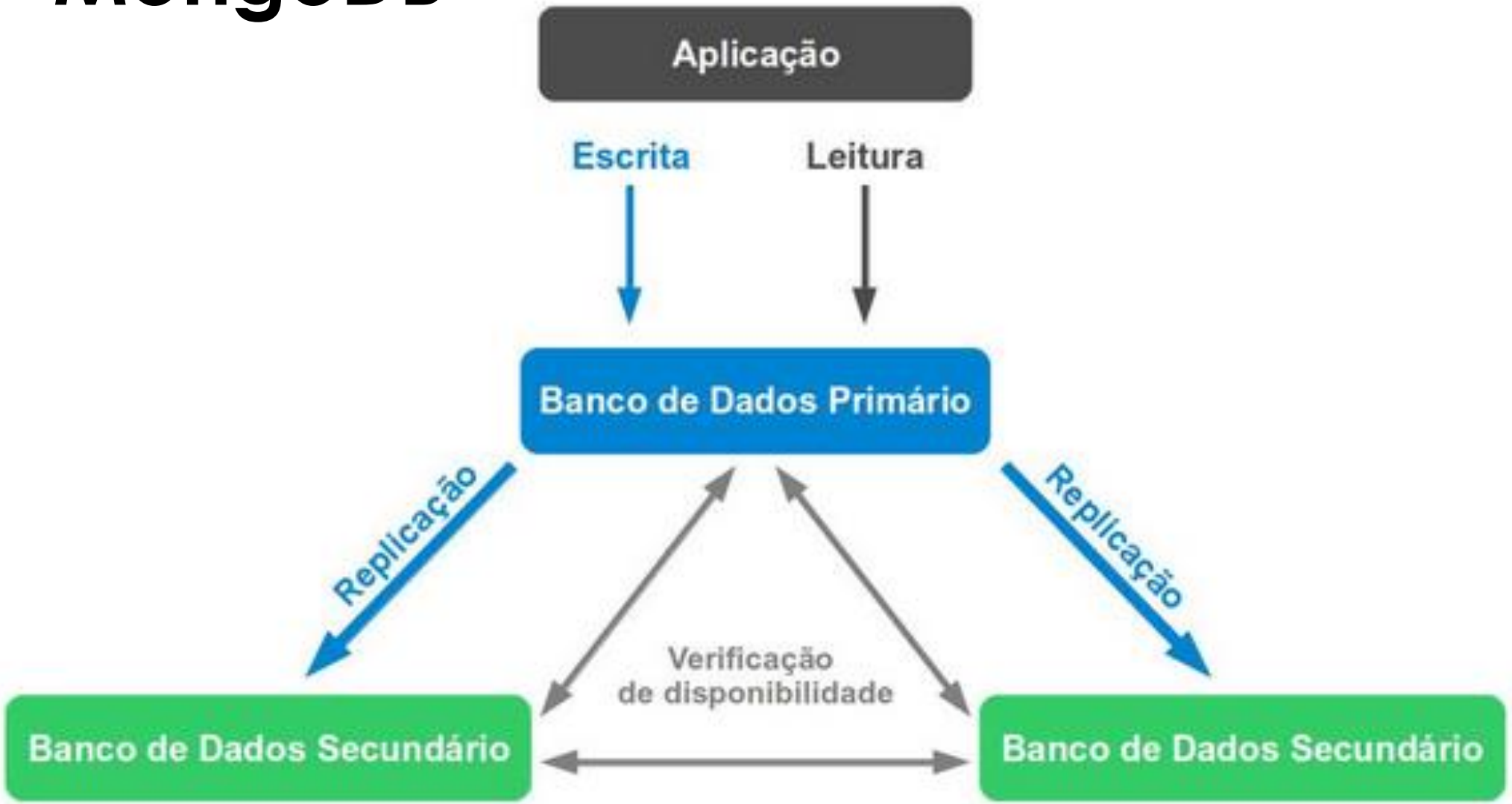
MongoDB

- Alguns Recursos
  - Alta disponibilidade
  - Escalabilidade horizontal
  - Armazenamento de arquivos (binários)
  - Agregation framework (Trabalhar com estatísticas, cruzamento de informações)
  - Dar suporte a diversas linguagens de programação
    - C#, Java, Php, Python, NodeJs
  - Joins entre coleções

MongoDB

- Query Language ou Mongo Shel
- MongoD – Serviço do MongoDB
- Replicação
  - Alta disponibilidade
    - Primário – Secundário – Secundário
    - Sempre que o primário cair, um secundário assume
  - Todas as requisições são feitas no primário

MongoDB


Aplicação
Escrita
Leitura
Banco de Dados Primário
Replicação
Replicação
Verificação
de disponibilidade
Banco de Dados Secundário
Banco de Dados Secundário

MongoDB

- Exemplo de modelagem – Alunos x Livros
  - Relacional
    - TBLIVROS
    - TBEDITORA
    - TBALUNOS
    - TBEMPRESTIMOS
  - MongoDB
    - Alunos
    - Livros

MongoDB

- Os não são estruturadas
  - Exemplo 1:
    - Para cada aluno repete-se: UF e Cidade
  - Exemplo 2:
    - {
      - “_ID”: 1,
      - “Titulo”: “Como programar em C#”,
      - “Autor”: “Joaquim F. Souza”,
      - “Emprestimos”: {
        - “Aluno_id” : 1,
        - “Data”:”ISODate(2016-11-01T15:55:57)”
      - }
    - }

MongoDB

- **Vantagem:**
  - Seria fácil saber quais livros estão lendo alugados no momento

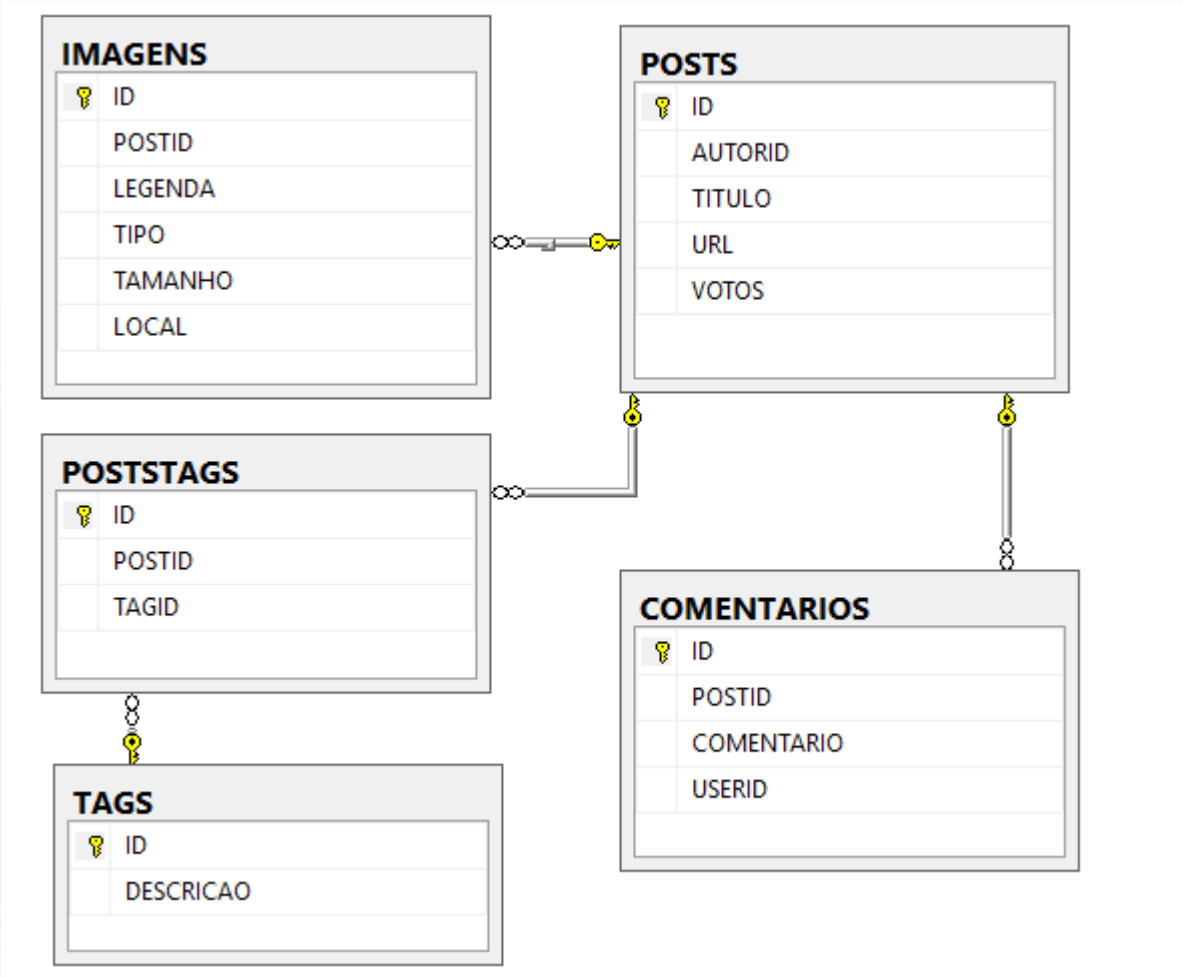
- **Problema:**
  - Saber quais alunos possuem livros alugados.
    - Necessário realizar consultas mais robustas ou diversas consultas

    - **Solução**: Armazenar o histórico de empréstimos na colação de alunos

MongoDB

- Modelagem
  - Posts
    - Id
    - AutorID
    - Titulo
    - Url
    - Votos
  - PostsTags
    - Id
    - PostID
    - TagID
  - Tags
    - Id
    - Descricao
- Imagens
  - Id
  - PostID
  - Legenda
  - Tipo
  - Tamanho
  - Local
- Comentarios
  - Id
  - PostID
  - Comentario
  - UserID

MongoDB


IMAGENS
ID
POSTID
LEGENDA
TIPO
TAMANHO
LOCAL
POSTS
ID
AUTORID
TITULO
URL
VOTOS
POSTSTAGS
ID
POSTID
TAGID
COMENTARIOS
ID
POSTID
COMENTARIO
USERID
TAGS
ID
DESCRICAO

MongoDB

```
{
  "ID":1,
  "AUTORID":10,
  "TITULO":"MongoDB - Aula de Laboratório",
  "URL":"http://aulamongo.com.br/post/1",
  "VOTOS":99,
    "COMENTARIOS": [
      {
        "ID": "1",
        "COMENTARIO": "Parabéns!!!",
        "USERID": 11
      },
      {
        "ID": "2",
        "COMENTARIO": "Comentário 2",
        "USERID": 14
      }
    ]
}
```

MongoDB

- Sintaxe
  - [Database].[coleção].[ação]( );
- Criando um Banco de Dados
  - Use [NomeBanco]
    - Cria e já começa a usar o Banco
  - DB – retorna o banco utilizado no momento
  - Show DBS – Retorna a lista de Banco de Dados
- Mongod: Inicia o serviço (No servidor)
- Mongo: Inicia o client

MongoDB

- Ferramenta online
  - https://www.tutorialspoint.com/mongodb_terminal_online.php
- Servidor Online – 500MB Free
  - https://mlab.com/
- Exemplo de Conexão servidor “mLAB”
  - mongo ds050879.mlab.com:50879/aula -u <dbuser> -p <dbpassword>

MongoDB

- **Insert**
  - Exemplo 1
    - //Inserindo – com parâmetros
    - db.teste.insert({a: true});
  - Exemplo 2
    - //Inserindo – com declaração de variável
    - var json = {b: 'TESTE'}
    - db.teste.insert(json)

- db.teste.find()
- {
  - "_id": ObjectId("546142385b9f2b586cb31d06"),
  - "a": true
- },
- {
  - "_id": ObjectId("546142665b9f2b586cb31d07"),
  - "b": "TESTE"
- }

MongoDB

- **ObjectId:**
- **Insert**
  - Insert um objeto
  - var disciplina =
    - {
      - 'descricao':‘Banco de Dados Avançados',
      - 'Categoria':‘Banco de Dados',
      - ‘Professor': ‘Valeria Times'
    - }
  - db.disciplinas.insert(disciplina)
  - db.disciplinas.find()

MongoDB

- **Insert**
  - Coleções
    - var pessoas = [
      - {‘nome’:’pessoa 1’,’idade’:27},
      - {‘nome’:’pessoa 2’,’idade’:29},
      - {‘nome’:’pessoa 3’,’idade’:40}
    -]
    - db.pessoas.insert(pessoas)
    - db.pessoas.find()

MongoDB

- **Save** – Insere e altera valores
  - Var pessoa = {‘nome’:’pessoa 5’,’idade’:27};
  - db.pessoas.save(pessoa)
  - db.pessoas.find()

  - Var query = {‘nome’: ‘Pessoa 5’}
  - var p = db.pessoas.findOne(query)
  - //É possível imprimir os valores do objeto
    - Ex:
      - p
      - p.nome ou p.idade
  - p.nome = ‘Novo nome’
  - db.pessoas.save(p)

MongoDB

- **Find**
  - Retorna um cursor que é convertido em array
    - var query = {idade:27}
    - var campos = {nome: 1, idade: 0}
      - Nos campos o valor determina que o propriedade será exibida no select
    - db.pessoas.find(query,campos)

MongoDB

- **Operadores Aritmétricos**
  - < é $lt - less than
    - db.colecao.find({ "campo" : { $lt: value } }); Retorna documentos com valores menores que value.
  - <= ou $lte - less than or equal
    - db.colecao.find({ "campo" : { $lte: value } }); Retorna documentos com valores menores ou igual que value.
  - > ou $gt - greater than
    - db.colecao.find({ "campo" : { $gt: value } }); Retorna documentos com valores maiores que value.
  - >= ou $gte - greater than or equal
    - db.colecao.find({ "campo" : { $gte: value } }); Retorna documentos com valores maiores ou igual que value.

MongoDB

- **Operadores Lógicos**
  - **$or - OU**
    - { $or : [{ campo1 : valor } , { campo2 : valor }] }
  - **$nor - Negação**
    - { $nor : [{ a : 1 } , { b : 2 }] }
  - **$and**
    - { $and: [{ a: 1 }, { a: { $gt: 5 } }] }
- **Operador de Existência**
  - **$exists**
    - db.colecao.find({ campo : { $exists : true } });

MongoDB

- **Datas**
  - {dataCad: new Date()}
  - {dataCad: new Date(2016,10,12)}

MongoDB

- **Update**
  - **$set**
  - db.colecao.update(query, mod, options);
    - var query = {"_id": ObjectId("9854670669bd5df270cc7e01")}
    - var mod = {$set: {nome: "Nome alterado"}}
    - db.pessoas.update(query, mod)
    - db.pessoas.find(query)
  - **Set com Arrays**
    - var mod = {$set: { telefones: ['(81) 9 9999-9999'] }}
  - **$unset – Remove um campo**
    - { $unset : { campo : 1} }
  - **$inc – Incrementa o valor de um campo**
    - { $inc : { campo : valor } }

MongoDB

- **$push**
  - Adiciona um valor a um campo Array (Caso não existe cria um novo)
    - { $push : { campo : valor } }
- **$pull**
  - Remove um valor de campo Array
    - db.pokemons.update(query, { $pull: { campo:valor} })
- **$each**
  - Insere um coleção de valores
    - { $push : { campo : {$each: [Array_de_valores] } } }

- **Options**
    - {
    - upsert: boolean, -- Update or Insert [FALSE]
    - multi: boolean, -- [O Banco garante que você não fará Update sem Where]
    - writeConcern: document --
    - }

MongoDB

- **Remove**
  - Apaga os dados mas a coleção continua existindo
    - db.pessoas.remove(query)

- **Drop**
  - Apaga toda a coleção
    - show collections
    - db.pessoas.drop()

- **Paginação**
  - Limit – Quantidade por página e skip – Número da página
    - .limit(10).skip(0 * 10);

MongoDB

- **Outros recursos**
  - Aggregation – Framework para querys mais elaboradas
    - https://docs.mongodb.org/manual/core/aggregation-pipeline/
  - Group – Agrupamento
    - https://docs.mongodb.com/manual/reference/method/db.collection.group/
  - Replica – Documentação sobre Replicação e disponibildade
    - https://docs.mongodb.com/manual/reference/method/rs.initiate/#rs.initiate
  - Sharding – Divisão de um grande cluster vários pequenos
    - https://docs.mongodb.com/
  - GridFs – Armazenamento de arquivos binários
    - https://docs.mongodb.com/

MongoDB

- Documentação MongoDB
  - https://docs.mongodb.com/

Obrigado a todos!!!!