2021/04/26 - AULA 06.1

Laboratório de Sistemas Computacionais Complexos

https://uclab.xyz/sistemas-complexos-2021-aula06-1

Renato Cordeiro Ferreira
renatocf@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br

Em caso de dúvidas

Acessem www.slido.com com #complexos

ou

Agenda

Tema da aula:
MongoDB

1. SQL vs. NoSQL
2. Modelos NoSQL
3. Mais sobre agregados
4. Usando o MongoDB

NoSQL vs SQL

SQL vs. NoSQL

O que é NoSQL?

NoSQL é uma classe de SGBDs cujos bancos são caracterizados por não utilizarem o modelo relacional.

Por que abrir mão dos benefícios de SQL?

Segundo Michael Stonebreaker[1] [2]

1. necessidade de melhor desempenho
2. necessidade de maior flexibilidade

[1] em “SQL Databases vs. NoSQL Databases”, Communications of the ACM, abril de 2010

[2] ganhador do prêmio Turing de 2015

Modelos NoSQL

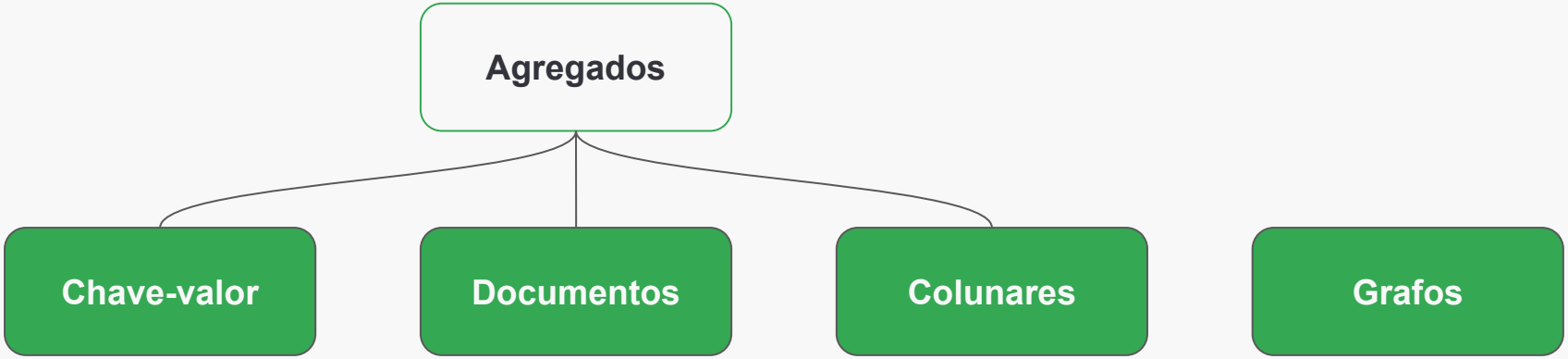
Modelos NoSQL

Outras características recorrentes:

1. Bom desempenho em operações simples de leitura e escrita
2. Particionamento e replicação de dados
3. Escalabilidade horizontal
4. Ausência de esquema fixo
5. Ausência de linguagem de consulta padronizada

Modelos NoSQL

Heterogeneidade e vários modelos:



Mais sobre agregados

Mais sobre agregados

O que é?

Definição de origem em DDD, um **Agregado** é um conjunto de objetos relacionados que deve ser tratado como uma **unidade de dados.**

1. diferente de tuplas do SQL
2. aninháveis
3. operações em agregados são atômicas

Mais sobre agregados

Questões interessantes

1. Outros agregados de alunos devem seguir a mesma estrutura?

2. Suponhamos outros alunos matriculados em MAC0475 e o nome da disciplina muda para “Agilidade e Sistemas Complexos”. O que precisa ser feito para essa atualização?

Usando o MongoDB

Usando o MongoDB

Alguns termos:

Por fins didáticos, vamos considerar uma configuração simples, sem *cluster*.

O processo	Um namespace	Um agrupamento semântico	O registro de dado

Usando o MongoDB

Primeiras interações:

1. `const uri = " … "`
2. `const client = …`
3. `await client.connect()`
4. `const database = client.db( … )`
5. `const movies = database.collection( … )`

```
const MongoDB = require("mongodb");

const uri =
  "mongodb+srv://" +
  "<user>:<password>@<cluster-url>" +
  "?retryWrites=true&writeConcern=majority";

const client = new MongoDB.MongoClient(uri, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

async function run() {
  try {
    await client.connect();

    const database = client.db('sample_mflix');
    const movies = database.collection('movies');

    const movie = await movies.findOne({
      title: 'Back to the Future'
    });

    console.log(movie);
  } catch (e) {
    console.dir(e);
  } finally {
    await client.close();
  }
}

run()
```

Usando o MongoDB

Algumas operações de CRUD:
Consultas feitas sobre uma **`collection`** (**`c`**) devolvem **`documents`**

- `c.insertOne(doc)`
- `c.insertMany(docs)`
- `c.find(query)`
- `c.findOne(query)`
- `c.updateOne(filter, update)`
- `c.updateMany(filter, update)`
- `c.deleteOne(filter)`
- `c.deleteMany(filter)`

mais informações na documentação de Collection

Usando o MongoDB

O *framework aggregate*:
Um *framework* nativo do MongoDB que realiza operações em formato de *pipeline*.

1. Considere a *collection* de artigos como na imagem
2. `articles.aggregate(pipeline)`

```
{
  "programming": 1,
  "database": 1,
  "mongodb": 1
}
```

```
{
   _id: ObjectId("..."),
   author: "...",
   title: "...",
   tags: [ "programming", "database", "mongodb" ]
}
```

```
async function run() {
  try {
    await client.connect()

    const db = client.db(...)
    const articles = db.collection('articles')

    const tagCounts = await articles.aggregate([
      { $project: { tags: 1 } },
      { $unwind: "$tags" },
      { $group: { _id: "$tags", count: { $sum : 1 } } }
   ])
  }
}
```

Licença

Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: **Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença**

Mais detalhes sobre essa licença em: creativecommons.org/licenses/by-nc-sa/3.0/

Créditos

Imagens usadas nesta apresentação são provenientes de: freepik.com

Frequência

Senha do Estudante: **ozgjuq**

2021/04/26 - AULA 06.1

Laboratório de Sistemas Computacionais Complexos

https://uclab.xyz/sistemas-complexos-2021-aula06-1

Renato Cordeiro Ferreira
renatocf@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br