

noSQL

Mestrado Integrado em Engenharia Informática

<https://hpeixoto.me/class/nosql>

Hugo Peixoto

hpeixoto@di.uminho.pt

2020/2021

noSQL

PL08 - Introdução às Bases de Dados de Grafos

Sumário

 Introdução às bases de dados multi-modelo

 Instalação de container do ArangoDB

 AQL

 Laboratório



Introdução às bases de dados multi-modelo

Introdução às bases de dados multi-modelo



ArangoDB

Introdução às bases de dados de multi-modelo

- A instalação do ArangoDB é **simples** e **fácil**;
- A modelação de dados é **flexível**: combina key-value, documentos ou grafos;
- **AQL** - Como linguagem de pesquisa;
- ArangoDB pode ser instalado como **servidor de aplicação**, fundindo os dados e a aplicação;
- As queries podem ser executadas em diferentes documentos e coleções, com **consistência** e **isolamento** opcional;
- A **replicação** e **fragmentação** (sharding) pode ser executada, escolhendo uma configuração master-slave e espalhando os dados por vários servidores;
- Tira partido de **armazenamento rápido** como os SSD;
- **Javascript** como linguagem universal;
- É **open source** (Apache License 2.0).

Introdução às bases de dados de multi-modelo

Modelação de dados:

Documentos em json;

Não tem esquema;

Suporta key-value;

Documentos armazenados em coleções;

Coleções específicas para suportar grafos.

✂ Instalação de container ArangoDB

Instalação de container ArangoDB

Instalação do container:

```
# docker run --name arangodb -p 8529:8529 -e ARANGO_RANDOM_ROOT_PASSWORD=1 -d arangodb
```

Encontrar a password de root:

```
# docker logs arangodb
automatically choosing storage engine
=====
GENERATED ROOT PASSWORD: rmw2maFU9FwIjwNv
=====
```

Aceder via browser (não esquecer abrir portas no Vbox)

<http://localhost:8529/>



AQL

Passos para a importação da BD de exemplo:

Criação de Utilizador

```
> nosql
```

Criação de Base de Dados

```
> nosqlbd
```

Criação de coleções

```
> airports (documental)
```

```
> flights (grafos)
```

Importar os ficheiros

flights.json

airports.json

Aceder ao separador "Collections" e criar um documento de exemplo.

CRUD:

Create:

```
INSERT {_key: "Test"} INTO 'airports'  
RETURN NEW
```

Read:

```
RETURN DOCUMENT('airports', ['test', 'Test'])
```

CRUD:

Update:

```
UPDATE {_key:'Test'} WITH {newVal: 1234} in 'airports'  
RETURN NEW
```

Update a vários documentos:

```
FOR docs in ['test','Test']  
  UPDATE docs WITH {newVal: 124} in 'airports'  
RETURN NEW
```

CRUD:

Delete:

```
FOR docs in ['test','Test']  
  REMOVE docs in 'airports'
```

Read com filtros:

```
FOR flight in flights  
  FILTER flight.TailNum == "N592ML"  
  RETURN flight
```

Devolve todos os voos que têm como propriedade o TailNum = “N592ML”

Joins:

```
FOR airport IN airports
  FILTER airport.city == "Dallas"
  FOR flight in flights
    FILTER flight._to == airport._id
RETURN {
  "airport": airport.name,
  "flight": flight.FlightNum
}
```

Groups:

```
FOR airport IN airports
  COLLECT state = airport.state WITH COUNT INTO total
  RETURN {
    "state":state,
    "total": total
  }
```

Group com ordenação:

```
FOR airport IN airports
  COLLECT state = airport.state WITH COUNT INTO total
  SORT total DESC
  RETURN {
    "state":state,
    "total": total
  }
```

Importar dados do IMDB:

Download do ficheiro imdb.zip

```
# docker cp imdb arangodb:/imdb
```

```
# docker exec -it arangodb arangorestore --input-directory "imdb" --server.database  
imdb --create-database true --create-collection true
```

Validar a importação:

```
FOR ver in imdb_vertices
```

```
return ver
```

Entrar na ArangoShell:

```
# docker exec -it arangodb arangosh --server.database imdb  
--server.authentication true
```

Criação de uma view

```
> db._createView('firstView','arangosearch',{})
```

Importação dos dados para a view:

```
> db._view('firstView').properties({links: { "imdb_vertices": { "includeAllFields":  
true, "fields": { "description": { analyzers: ["text_en"] }} } } } });
```

Listar todos os documentos com a palavra ninja:

```
FOR d IN firstView
  SEARCH ANALYZER(d.description == 'ninja', 'text_en')
LIMIT 20
RETURN {
  title: d.title,
  description: d.description
}
```

Listar todos os documentos com a palavra ninja:

```
FOR d IN firstView
  SEARCH PHRASE(d.description, 'art of the ninja','text_en')
LIMIT 20
RETURN {
  title: d.title,
  description: d.description
}
```

Listar todos os documentos com a frase "art of the ninja":

```
FOR d IN firstView
  SEARCH PHRASE(d.description, 'art of the ninja','text_en')
LIMIT 20
RETURN {
  title: d.title,
  description: d.description
}
```

Listar todos os documentos com a frase "art" % "ninja":

```
FOR d IN firstView
  SEARCH PHRASE(d.description, ['art', 2, 'ninja'],'text_en')
LIMIT 20
RETURN {
  title: d.title,
  description: d.description
}
```


Listar todos os documentos que contenham pelo menos 2 das frases:

```
FOR d IN firstView
  SEARCH ANALYZER(MIN_MATCH(
    PHRASE(d.description, 'feudal japan'),
    PHRASE(d.description, 'the ninja'),
    PHRASE(d.description, 'mutated turtles'), 2
  ),
  'text_en')
LIMIT 20
RETURN {
  title: d.title,
  description: d.description
}
```

Listar todos os documentos que contenham palavras que comecem por "st":

```
FOR d IN firstView
  SEARCH ANALYZER (STARTS_WITH(d.title, 'st'), 'text_en')
LIMIT 20
RETURN d.title
```

Acrescentar na view:

```
"title": {
  "analyzers": [
    "text_en"
  ]
},
```

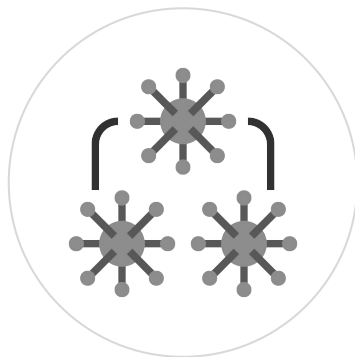


Laboratório

Laboratório

Exercício:

- 1) Filtrar todos os aeroportos que têm área vip.
- 2) Filtrar todos os aeroportos do estado "FL".
- 3) Procurar todos os voos que têm como origem o aeroporto "William B Hartsfield-Atlanta Intl".
- 4) Procurar todos os voos que têm como origem o aeroporto "William B Hartsfield-Atlanta Intl" e como destino o aeroporto "LaGuardia".



noSQL

Mestrado Integrado em Engenharia Informática

<https://hpeixoto.me/class/nosql>

Hugo Peixoto

hpeixoto@di.uminho.pt

2020/2021