



ArangoDB

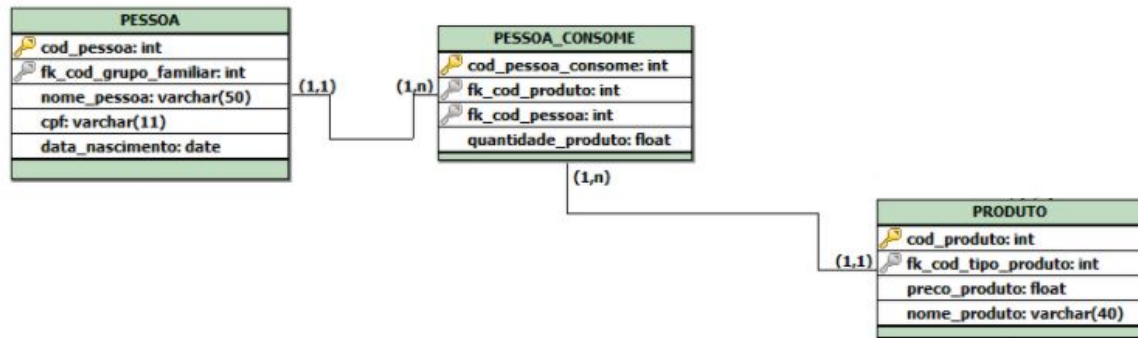
Kelvin Lehrback



ArangoDB- Características Gerais

- NoSQL do tipo multi-model (chave-valor, documentos e grafos)
- Código fonte aberto
- Utiliza uma linguagem chamada AQL(ArangoDB Query Language) que diminui a complexidade em determinadas situações
- Por ser do tipo multi-model, possui um certo grau de relacionamento
- Identificação dos dados feito obrigatoriamente de forma automática.
- Além do ID (identificador), possui também os atributos chaves do documento (key) e revisão do documento (rev)

Modelo l3gico vs Modelo ArangoDB



```
{ '_id': 'pessoa_consomme/14043640',  
  '_key': '14043640',  
  '_rev': '_bN8RD1---L',  
  'cpf': '49033310082',  
  'data_nascimento': '1934-10-09',  
  'nome_pessoa': 'Gerald de Rivia',  
  'produtos_consumidos': [{ 'nome_produto': 'Dorflex',  
    'preco_produto': 3.68,  
    'quantidade_produto': 20},  
    { 'nome_produto': 'Batata Monalisse',  
      'preco_produto': 3.85,  
      'quantidade_produto': 230} ] },
```

```
{ '_id': 'pessoa_consomme/14043641',  
  '_key': '14043641',  
  '_rev': '_bN8RD1---N',  
  'cpf': '88385078053',  
  'data_nascimento': '1923-12-31',  
  'nome_pessoa': 'Derfel da Silva',  
  'produtos_consumidos': [{ 'nome_produto': 'Farinha',  
    'preco_produto': 5.39,  
    'quantidade_produto': 150},  
    { 'nome_produto': 'Vitamina C',  
      'preco_produto': 8.59,  
      'quantidade_produto': 100} ] },
```



Sintaxe de conexão

Importando biblioteca e realizando conexão com o ArangoDB

```
[2] from arango import ArangoClient
    client = ArangoClient(hosts='https://d8e236160531.arangodb.cloud:8529/')
    db = client.db('_system', username='root', password='IKye3P1hs0bCIWvEbIdw')
```

Criando e acessando database 'provisoes_de_emergencia'

```
[3] if not db.has_database('provisoes_de_emergencia'):
    db.create_database('provisoes_de_emergencia')
```

Sintaxe de inclusão

Criando coleções

```
▶ db.create_collection('pessoa')
  db.create_collection('produto')
  db.create_collection('pessoa_consoma')
```

Inserindo um documento na coleção

```
▶ objeto_produto = {
    "nome_produto": "Lasanha",
    "preco_produto": 10.90
}
collection_produto = db.collection('produto')
collection_produto.insert(objeto_produto)
```

Inserindo vários documentos na coleção

Inserindo dados dos produtos

```
▶ produtos = [
    {
        "nome_produto": "Arroz",
        "preco_produto": 7.79
    },
    {
        "nome_produto": "Feijão",
        "preco_produto": 5.99
    }
]
```

```
    {
        "nome_produto": "Vitamina C",
        "preco_produto": 8.59
    },
    {
        "nome_produto": "Batata Monalisa",
        "preco_produto": 3.85
    }
]

collection_produto = db.collection('produto')
collection_produto.insert(produtos)
```

Sintaxe de seleção

Buscando os documentos de cada coleção

Convertemos para lista para mostrar os resultados

```
collection_produto = db.collection('produto')
list(collection_produto.all())
```

```
[{'_id': 'produto/14043612',
  '_key': '14043612',
  '_rev': '_bN8RDSu---',
  'nome_produto': 'Lasanha',
  'preco_produto': 10.9},
 {'_id': 'produto/14043613',
  '_key': '14043613',
  '_rev': '_bN8RDbG---',
  'nome_produto': 'Arroz',
  'preco_produto': 7.79},
 {'_id': 'produto/14043614',
  '_key': '14043614',
  '_rev': '_bN8RDbG--A',
  'nome_produto': 'Feijão',
  'preco_produto': 5.99},
 {'_id': 'produto/14043615',
  '_key': '14043615',
  '_rev': '_bN8RDbG--C',
  'nome_produto': 'Peixe',
  'preco_produto': 37.9},
 {'_id': 'produto/14043616',
  '_key': '14043616',
  '_rev': '_bN8RDbG--E',
  'nome_produto': 'Farinha',
```

```
collection_pessoa = db.collection('pessoa')
list(collection_pessoa.all())
```

```
[{'_rev': '_bN8RDjq--C',
  'cpf': '61576870880',
  'data_nascimento': '1977-09-03',
  'nome_pessoa': 'Valdeci G'},
 {'_id': 'pessoa/14043624',
  '_key': '14043624',
  '_rev': '_bN8RDjq--E',
  'cpf': '80663532787',
  'data_nascimento': '2007-12-20',
  'nome_pessoa': 'Clara L'},
 {'_id': 'pessoa/14043625',
  '_key': '14043625',
  '_rev': '_bN8RDjq--G',
  'cpf': '6410440979',
  'data_nascimento': '1979-07-05',
  'nome_pessoa': 'Kelen L'},
 {'_id': 'pessoa/14043626',
  '_key': '14043626',
  '_rev': '_bN8RDjq--I',
```

Buscando um documento em específico (utilizando filtros)

```
collection_pessoa = db.collection('pessoa')
list(collection_pessoa.find({'nome_pessoa': 'Merlin de Avalon'}))
```

```
[{'_id': 'pessoa/14043628',
  '_key': '14043628',
  '_rev': '_bN8RDjq--M',
  'cpf': '93397782007',
  'data_nascimento': '1960-05-17',
  'nome_pessoa': 'Merlin de Avalon'}
```

```
[12] collection_pessoa_consome = db.collection('pessoa_consome')
list(collection_pessoa_consome.all())
```

```
[{'_id': 'pessoa_consome/14043633',
  '_key': '14043633',
  '_rev': '_bN8RD06--C',
  'cpf': '61576870880',
  'data_nascimento': '1977-09-03',
  'nome_pessoa': 'Valdeci G',
  'produtos_consumidos': [{'nome_produto': 'Arroz',
    'preco_produto': 7.79,
    'quantidade_produto': 100},
    {'nome_produto': 'Peixe',
    'preco_produto': 37.9,
    'quantidade_produto': 250}]},
 {'_id': 'pessoa_consome/14043634',
  '_key': '14043634',
  '_rev': '_bN8RD1---',
  'cpf': '80663532787',
  'data_nascimento': '2007-12-20',
  'nome_pessoa': 'Clara L',
  'produtos_consumidos': [{'nome_produto': 'Feijão',
    'preco_produto': 5.99,
    'quantidade_produto': 150},
    {'nome_produto': 'Farinha', 'preco_produto': 5.39}]},
 {'_id': 'pessoa_consome/14043635',
  '_key': '14043635',
  '_rev': '_bN8RD1---B',
  'cpf': '6410440979',
  'data_nascimento': '1979-07-05',
  'nome_pessoa': 'Kelen L',
  'produtos_consumidos': [{'nome_produto': 'Cloro',
    'preco_produto': 10.79,
    'quantidade_produto': 500},
    {'nome_produto': 'Tomate Carmem',
    'preco_produto': 3.59,
    'quantidade_produto': 100}]},
 {'_id': 'pessoa_consome/14043636',
  '_key': '14043636',
```



Sintaxe de exclusão

Excluindo documentos

```
▶ collection_pessoa = db.collection('pessoa')  
collection_pessoa.delete_match({'nome_pessoa': 'Merlin de Avalon'})
```

```
▶ collection_produto = db.collection('produto')  
collection_produto.delete_match({'nome_produto': 'Vitamina C'})
```

Sintaxe de atualização

Atualizando documentos

```
[14] collection_pessoa = db.collection('pessoa')  
      collection_pessoa.update_match({'nome_pessoa': 'Valdeci G'}, {'data_nascimento': '2000-01-01'})
```



```
collection_produto = db.collection('produto')  
collection_produto.update_match({'nome_produto': 'Lasanha'}, {'preco_produto': 9.90})
```




Pontos positivos

- Boa documentação voltada para devs
- Possui um eficiente sistema de tolerâncias a falhas (de modo nativo)
- Performance simplificada (pode-se aumentar a performance de maneira rápida, escalando tanto verticalmente como horizontalmente)
- Suporte a estrutura de grafos, não se prendendo apenas no “chave-valor”
- Estrutura flexível, facilitando o “update” dos dados e coleções
-



Pontos negativos

- Não possui oferta em nuvem gratuita (apenas por um período, o que dificulta o estudo da linguagem de forma prolongada)
- Alta curva de aprendizagem, se comparado com outros bancos NoSQL
- Comunidade ainda em ascensão, o que dificulta a busca por profissionais dessa área e fóruns ativos
- Identificação dos documentos não pode ser alterada pelo desenvolvedor
- Possui atributos obrigatórios que para um sistema simples, acabam sendo irrelevantes, aumentando o consumo de memória



Referências

- <https://www.arangodb.com/why-arangodb/sql-aql-comparison/>
- <https://python-driver-for-arangodb.readthedocs.io/en/master/overview.html>
- <https://www.arangodb.com/tutorials/mongodb-to-arangodb-tutorial/>
- <https://db-engines.com/en/system/ArangoDB>