

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
UNIDADE ACADÊMICA ESPECIALIZADA EM CIÊNCIAS AGRÁRIAS  
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
COMPONENTE CURRICULAR: FUNDAMENTOS E TÉCNICAS EM CIÊNCIAS DE DADOS

Lista de Exercícios 1

1) **Verdadeiro ou Falso?** Como forma de permitir as buscas em documentos semi-estruturados, um banco de dados NoSQL do tipo orientado a documentos armazena objetos indexados por chaves utilizando tabelas de hash distribuídas (CESPE/TCU 2015).

R. **Falso.** O tipo de banco **chave-valor** tem sua estrutura baseada em tabelas hash distribuídas.

2) Considere as seguintes características de um projeto de banco de dados (IBGE/2016).

- I. O modelo de dados é conhecido a priori e é estável;
- II. A integridade dos dados deve ser rigorosamente mantida;
- III. **Velocidade e escalabilidade são preponderantes.**

Dessas características, o emprego de bancos de dados NoSQL é favorecido somente por:

- a. **III**
- b. II e III
- c. I
- d. II
- e. I e II

3) Sobre os banco de dados NoSQL, assinale a afirmativa correta.

- a) não podem ser indexados
- b) são considerados bancos de dados relacionais
- c) deve ser definido um esquema fixo antes de qualquer operação
- d) são exemplos de NoSQL: mongoDB, Firebird, DynamoDB, SQLite, Access, Azure Table
- e) **n.d.a**

4) **Verdadeiro ou Falso?** Com relação à forma como os dados são armazenados e manipulados no desenvolvimento de aplicações, julgue os itens a seguir. Em um banco de dados NoSQL do tipo grafo, cada arco é definido por um identificador único e expresso como um par chave/valor.

R. **Falso.** Um NoSQL baseado em grafos pode ter seus arcos com identificadores, mas não apenas com um par chave/valor, pode haver um conjunto de pares chave-valor em cada aresta, inclusive indicando o sentido da conexão e 'nomeando' a ação ou função semântica daquele arco.

5) Seja os comandos SQL abaixo que criaram a estrutura de BD (tabelas) na figura. Considerando que as mesmas podem estar numa mesma coleção, escreva o arquivo JSON para a representação NoSQL de documentos desta estrutura. Se possível instalar o mongoDB compass em sua máquina ou utilizar no cloud.mongodb (ATLAS), importe o arquivo JSON produzido.

MeuBanco.

```
CREATE TABLE Tab1
(Codigo int PRIMARY KEY NOT NULL,
Descricao text NULL);

CREATE TABLE Tab2
(Cod int PRIMARY KEY NOT NULL,
Descricao text NULL,
Fonte int NOT NULL,
CONSTRAINT FK_Tab2_Fonte FOREIGN KEY
(Fonte) REFERENCES Tab1 (Codigo));

CREATE TABLE Tab3
(ID_Seq int PRIMARY KEY NOT NULL,
Inscricao int UNIQUE NOT NULL,
Responsavel int NULL,
CONSTRAINT FK_Tab3_Responsavel FOREIGN KEY
(Responsavel) REFERENCES Tab3 (Inscricao));
```

Tab1

Codigo	Descricao
10	Alfa
20	Beta
30	Qui

Tab2

Cod	Descricao	Fonte
1000	Zeta	30
2000	Psi	10
3000	Xi	20
4000	Omega	10
5000	Upsilon	20
6000	Tau	30

Tab3

ID_Seq	Inscricao	Responsavel
1	201601	201603
2	201602	201602
3	201603	201603
4	201604	201604
5	201605	201603
6	201606	201602

R. As tabelas Tab1 e Tab2 estão relacionadas, com o campo Tab2.Fonte (FK) associado ao campo Tab1.Codigo. A Tab3 é independente, estando auto-relacionada. Portanto, iremos focar na Tab1 e Tab2. Embora Tab1 e Tab2 possam representar coleções separadas, onde a coleção Tab2 incluiria a chave Fonte com o respectivo valor apontando para a PK de Tab1, a questão menciona ser apenas uma coleção agregadora. Uma solução é usar documentos embutidos ou também chamados subdocumentos. Assim, ao buscar pela PK (\_id de Tab1), todos os dados são retornados numa única busca. Criaremos portanto 04 documentos:

JSON Data/URL	JSON Data/URL	JSON Data/URL
<pre>{   "codigo":10,   "descricao":"alfa",   "tab2":[     {       "codigo":2000,       "desc":"psi"     },     {       "codigo":4000,       "desc":"omega"     }   ] }</pre>	<pre>{   "codigo":20,   "descricao":"beta",   "tab2":[     {       "codigo":3000,       "desc":"xi"     },     {       "codigo":5000,       "desc":"upsilon"     }   ] }</pre>	<pre>{   "codigo":30,   "descricao":"qui",   "tab2":[     {       "codigo":1000,       "desc":"zeta"     },     {       "codigo":6000,       "desc":"tau"     }   ] }</pre>

```

{
  "tab3": [
    {
      "_id": 1,
      "inscricao": 201601,
      "responsavel": 201603
    },
    {
      "_id": 2,
      "inscricao": 201602,
      "responsavel": 201602
    },
    {
      "_id": 3,
      "inscricao": 201603,
      "responsavel": 201603
    },
    {
      "_id": 4,
      "inscricao": 201604,
      "responsavel": 201604
    },
    {
      "_id": 5,
      "inscricao": 201605,
      "responsavel": 201603
    },
    {
      "_id": 6,
      "inscricao": 201606,
      "responsavel": 201602
    }
  ]
}

```

Primeiro documento inserido no mongoDB (como exemplo). Atenção que no JSON todas chaves e valores string devem estar com aspas duplas:

```

codigo: 10
descricao: "alfa"
▼ tab2: Array
  ▼ 0: Object
    codigo: 2000
    desc: "psi"
  ▼ 1: Object
    codigo: 4000
    desc: "omega"
  _id: ObjectId("602c5f6743a6fd76cca59dc4")

```

Caso haja problemas na inserção digitando o documento diretamente no Compass (insert document) ou mesmo no insert file (criando o .json associado, como no caso do tab3.json para a terceira tabela), pode-se inserir via shell (no robo3T, por exemplo) ou via pymongo, como no código abaixo, que funciona corretamente:

```
In [2]: import json
from pymongo import MongoClient
localConn = MongoClient('mongodb://127.0.0.1:27017')

# selecionar banco
db = localConn.test #ou db = cloudConn['sample_airbnb']
# acessar collection (equivalente a uma tabela no relacional), onde os documentos estão armazenados
tb = db.test

with open('tab3.json') as f:
    fileData = json.load(f)

tb.insert_one(fileData)
print('documento inserido com sucesso...')
localConn.close()
```

documento inserido com sucesso...

6) Em uma tabela chamada Contribuinte de um banco de dados padrão SQL aberto e em condições ideais há o campo idContribuinte do tipo inteiro e chave primária. Há também o campo nomeContribuinte que é do tipo varchar. Nessa tabela, um Auditor Fiscal deseja alterar o nome do contribuinte de id 1 para 'Marcos Silva'. Para isso, terá que utilizar o comando:

- a) ALTER TABLE Contribuinte SET nomeContribuinte = 'Marcos Silva' WHERE idContribuinte = 1
- b) **UPDATE Contribuinte SET nomeContribuinte = 'Marcos Silva' WHERE idContribuinte=1**
- c) UPDATE nomeContribuinte TO 'Marcos Silva' FROM Contribuinte WHERE idContribuinte=1

7) O termo NoSQL refere-se

- a) a uma abordagem teórica que segue o princípio de não utilização da linguagem SQL em bancos de dados heterogêneos.
- b) **ao aumento da escalabilidade das bases de dados neles armazenados, aliado a um desempenho mais satisfatório no seu acesso.**
- c) à facilidade de implementação de bases de dados normalizadas, com vistas a minimização de redundâncias no conjunto de dados