

Alunos: Deivid Castagna, Lorenzo Jablonski

Nosso trabalho foi feito utilizando o MongoDB na versão 6.0.3.

Justificativa

Gostamos do visual, e comparamos as nossas opções com os seguintes argumentos:

- Chave-valor: aparenta uma fraqueza visto a repetição de muitos dados semelhantes para sua chave;
- Colunas: ineficiente por se tratar de um banco que em teoria tem muitas entradas e suas buscas por colunas específicas não são tão úteis visto que o registro total é o mais importante;
- Grafos: interconectividade e topologia não são importantes;
- Documentos: suas coleções de atributos, com várias chaves possíveis, combinam com nossa estrutura.

Dados usados

Usamos os mesmos dados do primeiro trabalho, uma base de dados com faixas extraídas do Spotify em formato csv, com a diferença de que, nesta vez, excluimos várias colunas que não utilizaríamos. O arquivo usado está nomeado como SpotifyTracks.csv no repositório.

Criptografia

Utilizamos um algoritmo de criptografia chamado "Fernet". Utilizamos python para implementar. O algoritmo de criptografia está no arquivo nomeado de cripto.py.

Inserção

A inserção dos dados aconteceu pela interface do MongoDB, o MongoDB Compass. Foi possível importar o arquivo csv dizendo os tipos de dados de cada coluna e o separador utilizado.

Consultas

Desenvolvemos quatro consultas:

1 – Esta consulta responde à pergunta: Qual é a música do gênero Indie dentro desta base de dados com a maior popularidade? Lembrando que este índice compreende a pior popularidade em 0 e a maior em 100.

```
use('SpotifyTracks');  
db.getCollection('Tracks')  
  .find({genre : "Indie"},  
  {})  
  .sort({popularity : -1})  
  .limit(1);
```

2 – Já nesta segunda consulta, listamos todas as faixas de artista Adele que tenham um índice de popularidade maior ou igual a 10. Nota-se que o banco retornou 61 documentos.

```
use('SpotifyTracks');
db.getCollection('Tracks')
  .find({artist_name : "Adele", popularity : {$gte : 10}},
  {})
  .sort({});
```

3 – Na terceira consulta, nós requisitamos ao banco os registros que tenham gênero igual a Movie e uma duração (em milissegundos) menor ou igual a 200000. Nota-se que o banco retornou 4160 documentos.

```
use('SpotifyTracks');
db.getCollection('Tracks')
  .find({genre : "Movie", duration_ms : {$lte : 200000}},
  {})
  .sort({});
```

4 – Agora, na última consulta, pedimos ao banco que liste os registros que contenham gênero igual a Pop, um índice de dançabilidade (quão dançante a faixa é) maior ou igual a 0.7 com os documentos ordenados em ordem decrescente a partir do índice de dançabilidade.

```
use('SpotifyTracks');
db.getCollection('Tracks')
  .find({genre : "Pop", danceability : {$gte : 0.7}},
  {})
  .sort({danceability : -1});
```