

Acessando Banco Access com Python

Mário Leite

...

Segundo Turban *et al* (2003) “O programa de software (ou grupo de programas) que fornece acesso a um banco de dados é conhecido como Sistema Gerenciador de Banco de Dados”. Esta definição enfatiza o fato de um SGBD (DBMS em Inglês - Data Base Management System) ser um *software* que permite acessar uma base de dados de maneira interativa e mais profunda. Outra definição mais detalhada é dada por O’Brien (2003): “Um sistema de gerenciamento de bancos de dados (DBMS em inglês) é um conjunto de programas de computador que controla a criação, manutenção e uso dos bancos de dados para uma organização e seus usuários finais”. Ou de maneira mais técnica: “uma coleção de dados inter-relacionados, representando dados sobre um domínio específico”. De acordo com O’Brien (2003) os dados estão armazenados no banco em “locais” bem determinados para uso dos tomadores de decisão da organização. Em resumo: os bancos de dados são apenas arquivos formatados com dados segundo uma análise bem estudada e de maneira a permitir que os acessos a esses dados sejam rápidos e seletivos.

Existem vários tipos e formatos de bancos de dados e que podem ser classificados de várias maneiras; por exemplo, em “Relacionais” e “Não Relacionais”. Um exemplo muito comum, e que pode ser considerado a “entrada” para esse mundo do armazenamento de dados é o **MS-Access** (ou simplesmente Access), que é relacional e criado pela Microsoft Corporation®. Esse aplicativo, atualmente integrado ao pacote MS-Office® é o SGBD mais fácil de trabalhar, uma vez que todos os recursos para acesso e manipulação dos dados estão integrados nesse pacote, fazendo com que o aprendizado seja fácil e rápido. Entretanto, para a maioria dos programadores o uso direto desse SGBD não traz muita vantagem, já que a manipulação dos dados para obter algum tipo de formação envolve uma programação muito “engessada” na chamada “Grade QBE” (grade de consulta por exemplos). Por isto, é importante que os programadores, de um modo geral, tenham opções de acessar bancos de dados através de recursos da própria linguagem em que implementam seus programas. Este é o caso de programadores Python, que possuem muitos recursos de código e que podem desenvolver aplicações que acessam bancos de dados.

O programa “**AcessaBancoAccess**”, implementado em Python, usa recursos avançados de código baseados na biblioteca “pyodbc” para mostrar um exemplo de aplicação bem interessante. Esse programa acessa um banco de dados chamado “Loterias”.mdb” e mostra o número de ocorrências de cada dezena da tabela “MegaSena”; também mostra a dezena que foi sorteada mais vezes e a que foi sorteada menos vezes até o último concurso cadastrado nessa tabela. A **figura 1** mostra a saída do programa, considerando até o último concurso cadastrado.

Relatório das dezenas sorteadas até o Concurso '2695'

01 sorteada 257 vezes	31 sorteada 252 vezes
02 sorteada 268 vezes	32 sorteada 285 vezes
03 sorteada 253 vezes	33 sorteada 290 vezes
04 sorteada 286 vezes	34 sorteada 290 vezes
05 sorteada 301 vezes	35 sorteada 291 vezes
06 sorteada 267 vezes	36 sorteada 270 vezes
07 sorteada 254 vezes	37 sorteada 288 vezes
08 sorteada 268 vezes	38 sorteada 284 vezes
09 sorteada 255 vezes	39 sorteada 255 vezes
10 sorteada 314 vezes	40 sorteada 254 vezes
11 sorteada 278 vezes	41 sorteada 286 vezes
12 sorteada 258 vezes	42 sorteada 288 vezes
13 sorteada 274 vezes	43 sorteada 283 vezes
14 sorteada 262 vezes	44 sorteada 286 vezes
15 sorteada 234 vezes	45 sorteada 262 vezes
16 sorteada 278 vezes	46 sorteada 277 vezes
17 sorteada 282 vezes	47 sorteada 256 vezes
18 sorteada 260 vezes	48 sorteada 248 vezes
19 sorteada 252 vezes	49 sorteada 272 vezes
20 sorteada 263 vezes	50 sorteada 262 vezes
21 sorteada 228 vezes	51 sorteada 273 vezes
22 sorteada 247 vezes	52 sorteada 271 vezes
23 sorteada 287 vezes	53 sorteada 308 vezes
24 sorteada 267 vezes	54 sorteada 278 vezes
25 sorteada 255 vezes	55 sorteada 227 vezes
26 sorteada 229 vezes	56 sorteada 280 vezes
27 sorteada 280 vezes	57 sorteada 257 vezes
28 sorteada 279 vezes	58 sorteada 264 vezes
29 sorteada 283 vezes	59 sorteada 261 vezes
30 sorteada 288 vezes	60 sorteada 255 vezes

Dezena mais sorteada: 10 - 314 vezes

Dezena menos sorteada: 55 - 227 vezes

Process finished with exit code 0

Figura 1 - Saída do programa

```
'''
```

AcessaBancoAccess.py

Acessa a tabela "MegaSena" do banco de dados Access "Loterias.mdb" e mostra o número de vezes de cada dezena sorteada até o último concurso cadastrado.

```
'''
```

```
import sys
```

```
sys.path.append(r'C:\Users\Usuario\AppData\Local\Programs\Python\Python310\lib\site-packages')
```

```
import pyodbc
```

```
#Cria a string de conexão com o banco de dados "Access"
```

```
strCon = (  
    r'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};'   
    r'DBQ=D:\Livros\livro11\Dados\Loterias.mdb;'   
)
```

```
#Cria o objeto de conexão
```

```
objCon = pyodbc.connect(strCon)
```

```
cursor = objCon.cursor()
```

```
#Define e acessa a tabela desejada
```

```
tabela = "MegaSena"
```

```
print()
```

```
cursor.execute(f"SELECT COUNT(*) FROM {tabela}")
```

```
nRegs = cursor.fetchone()[0]
```

```
print(f"Relatório das dezenas sorteadas até o Concurso '{nRegs}'")
```

```
print("-----")
```

```
#Cria um dicionário para contar as dezenas sorteadas
```

```
contDez = {f"{i:02d}": 0 for i in range(1, 61)}
```

```
#Obtém a contagem de cada dezena
```

```
cursor.execute(f"SELECT Dez01, Dez02, Dez03, Dez04, Dez05, Dez06 FROM {tabela}")
```

```
LstRegs = cursor.fetchall()
```

```
for reg in LstRegs:
```

```
    for dezena in reg:
```

```
        if(dezena and dezena in contDez):
```

```
            contDez[dezena] += 1
```

```
#Imprime a contagem de cada dezena em duas colunas
```

```
LstCol1 = [f"{i:02d} sorteada {contDez[f'{i:02d}']} vezes" for i in range(1,31)]
```

```
LstCol2 = [f"{i:02d} sorteada {contDez[f'{i:02d}']} vezes" for i in range(31,61)]
```

```
for c1, c2 in zip(LstCol1, LstCol2):
```

```
    print(f"{c1:<30} {c2}")
```

```
#Encontra a dezena mais sorteada e a menos sorteada
```

```
maisSort = max(contDez, key=contDez.get)
```

```
menosSort = min(contDez, key=contDez.get)
```

```
print()
```

```
print(f"Dezena mais sorteada: {maisSort} - {contDez[maisSort]} vezes")
```

```
print(f"Dezena menos sorteada: {menosSort} - {contDez[menosSort]} vezes")
```

```
objCon.close() #Fecha a conexão com o banco de dados (destrói o objeto de conexão)
```

```
#Fim do programa " AcessaBancoAccess" -----
```