

```
In [1]: # Importação das bibliotecas necessárias
import csv
import pandas as pd
import re
```

```
In [2]: # Importação do arquivo Excel
df = pd.read_excel('Dados_Brutos.xlsx')
```

```
In [3]: # Preenche os valores ausentes com 0
df.fillna(value=0, inplace=True)
```

```
In [11]: # Converte os valores da coluna 'IDADE' para int32
df['IDADE'] = df['IDADE'].astype(int)
```

```
In [8]: # Transfere os dados convertidos para um arquivo .csv
df.to_csv('Dados_Brutos.csv', index=False)
```

```
In [9]: # Função para gerar script de SQL
def scriptGenerator(nome_arquivo, limite_inferior, limite_superior):
    # Abre o arquivo 'Dados_Brutos.csv' no modo leitura
    with open('Dados_Brutos.csv', newline='') as csvfile:
        reader = csv.reader(csvfile)

        # Cria um arquivo '.sql' com o parâmetro recebido e abre o arquivo no modo de escrita
        f = open(nome_arquivo, 'w')

        counter = 0

        for line in reader:
            newline = list()
            # Remove todos os apóstrofes das palavras
            for element in line:
                if re.search("'", element):
                    index = element.find("'")
                    e_list = list(element)
                    e_list[index] = ''
                    delimiter = ''
                    element = delimiter.join(e_list)
                    element = '\\' + element + '\\'
                    newline.append(element)
                # Substitui todos os elementos 0 por 'NULL'
                elif element == '0':
                    element = 'NULL'
                    newline.append(element)
                # Coloca todas as palavras e datas entre aspas simples
                elif not (element.isdigit() or element == 'NULL'):
                    element = '\\' + element + '\\'
                    newline.append(element)
                else:
                    newline.append(element)

            # Une os elementos de cada linha, separados por vírgula
            delimiter = ', '
            row = delimiter.join(newline)

            # Cria os comandos insert para cada linha da tabela e escreve em um arquivo '.sql'
            if row.startswith('\\ID\\'): continue
            if counter >= limite_inferior and counter < limite_superior:
                f.write('INSERT INTO Dados_Brutos VALUES (' + row + ')\n\n')

            counter += 1

        f.close()
```

```
In [10]: # Gera scripts SQL com um número de linhas consecutivas pré-definido nos argumentos da função
scriptGenerator('script1.sql', 0, 100000)
scriptGenerator('script2.sql', 100000, 200000)
scriptGenerator('script3.sql', 200000, 300000)
scriptGenerator('script4.sql', 300000, 400000)
scriptGenerator('script5.sql', 400000, 500000)
scriptGenerator('script6.sql', 500000, 600000)
scriptGenerator('script7.sql', 600000, 700000)
scriptGenerator('script8.sql', 700000, 800000)
```

```
Out[10]: "scriptGenerator('script2.sql', 100000, 200000)\nscriptGenerator('script3.sql', 200000, 300000)\nscriptGenerator('script
4.sql', 300000, 400000)\nscriptGenerator('script5.sql', 400000, 500000)\nscriptGenerator('script6.sql', 500000, 600000)\ns
criptGenerator('script7.sql', 600000, 700000)\nscriptGenerator('script8.sql', 700000, 800000)\n"
```

In []: