

Popular as Tabelas SQL

Para popular as tabelas em SQL foram usados dois scripts escritos em Python, foi usado um script chamado de `bs4_population.py` que é responsável por ir buscar nomes de atletas a páginas web com listas de nomes de atletas olímpicos e modalidades oficiais de atletismo. E o script `db_populate.py` que é responsável por manipular todos os dados e gerar código SQL responsável pela população da base de dados. Este código é guardado num ficheiro designado por `db_populate.sql`.

Este comando irá gerar dois ficheiros de texto (`atletas.txt` e `modalidades.txt`) com dados necessários para executar o script `db_populate.py` :

```
> python bs4_population.py
```

De seguida, o script `db_populate.py` :

```
> python db_populate.py
```

```
Ficheiros .txt encontrados e dados importados!
```

```
Atletas: 733
```

```
Modalidades: 9
```

```
Clubes: 24
```

```
Categorias: 5
```

```
Códigos Postais: 22
```

```
Especialidades: 10
```

```
Médicos: 20
```

```
Consultas : 400
```

```
Gerando SQL...
```

```
Ficheiro db_populate.sql gerado com sucesso!
```

```
Tempo de execução: 0.101s
```

No final da execução dos dois script ficamos com um ficheiro `db_populate.sql` pronto a ser executado e dessa forma populando a nossa base de dados.

Analizando o script 'db_populate.py'

Dependências:

```
import os # Executar comandos na 'command line'.
import platform # Detetar o Sistema Operativo onde o script é corrido.
import bs4_population # Module responsável por dar scrap.
```

```

from datetime import datetime # Noção de tempo atual.
import threading # Animation.
import time # Performance.
from bs4 import BeautifulSoup # Procurar em HTML.
import requests # Fazer HTTP requests.
from pathlib import Path # Encontrar ficheiros.
import random # Muito importante para gerar números aleatórios.
import codecs # 'UTF-8' charset.

```

Dados Iniciais do Script:

```

# Dados Iniciais
current_year = datetime.now().year
current_month = datetime.now().month
current_day = datetime.now().day
random.seed() # Gerar uma nova seed.
ids_athlete = [] # A popular no futuro.
ids_doctor = [] # A popular no futuro.
categorias = ["Escolinhas", "Infantis", "Juvenis", "Juniore", "Seniores"]
cidades = ["Angra do Heroísmo", "Praia da Vitória", "Ponta Delgada", "Lisboa", "Porto", "Funchal"]
zipcodes = [f'{random.randint(1000,9999)}-{random.randint(100,999)}' for i in range(len(cidades))]
medicos = ["Asi Lucas", "Belchior Telles", "Boaventura Quintana", "Clarisse Mantas", "Collin Gouveia"]
clubes = ["SL Benfica", "FC Porto", "Sporting CP", "Boavista FC", "SC Braga", "Vitória S.C.", "F.C. Paços de Ferreira"]
modalidades_nomes = [] # A popular no futuro.
atletas_nomes = [] # A popular no futuro.
especialidades = ["Anatomia Patológica", "Anestesiologia", "Cardiologia", "Cirurgia Geral", "Neurologia"]

```

Popular os Atletas e Modalidades:

```

try:
    with codecs.open('atletas.txt', 'r', 'utf-8') as file:
        line = file.readline()
        while line[:len(line) - 1]:
            atletas_nomes.append(line.replace('\n', '').replace('\r',''))
            line = file.readline()

    with open('modalidades.txt', 'r') as file:
        line = file.readline()
        while line[:len(line) - 1]:
            modalidades_nomes.append(line.replace('\n', ''))
            line = file.readline()

except Exception:
    print("Ficheiros .txt não encontrados ou mal formatados!")
    raise # Parar o programa com erro.

else: # Alguns dados de quantidades que serão geradas.
    print("Ficheiros .txt encontrados e dados importados!")
    print(f'Atletas: {len(atletas_nomes)}')
    print(f'Modalidades: {len(modalidades_nomes)}')

```

```

print(f'Clubes: {len(clubes)}')
print(f'Categorias: {len(categorias)}')
print(f'Códigos Postais: {len(zipcodes)}')
print(f'Especialidades: {len(especialidades)}')
print(f'Médicos: {len(medicos)}')
print(f'Consultas : 400\n')
print(f'Gerando SQL...\n')

```

Animação de espera pela execução do script e Tempo de Execução:

complete = False # Variável responsável por parar a animação.

```

def animation():
    animation = "|/-\\"
    idx = 0
    while not complete:
        print(animation[idx % len(animation)], end="\r")
        idx += 1
        time.sleep(0.1)

```

```

animation_func = threading.Thread(target=animation)
animation_func.start() # Começar a thread da animação.

```

Execution Time

```

start = time.time()

```

Gerar código SQL:

Gerar SQL

```

i = 1
try:
    with codecs.open('db_populate.sql', 'w', 'utf-8') as file:
        # Clubes
        file.write("/* Inserir os Clubes na tabela \"club\" */\n")
        file.write("INSERT INTO club (idClub, nameClub)\nVALUES\n")
        for clube in clubes:
            if (i == len(clubes)):
                file.write(f'\t({i}, \"{clube}\");\n')
            else:
                file.write(f'\t({i}, \"{clube}\"),\n')

            i += 1

        # Modalidades
        i = 1
        file.write("\n/* Inserir as Modalidades na tabela \"modality\" */\n")

```

```

file.write("INSERT INTO modality (idModality, nameModality)\nVALUES\n")
for modalidade in modalidades_nomes:
    if (i == len(modalidades_nomes)):
        file.write(f'\t({i}, \'{modalidade}\');\n\n')

    else:
        file.write(f'\t({i}, \'{modalidade}\'),\n')

    i += 1

# Especialidades
i = 1
file.write("/* Inserir as Especialidades na tabela \"expertise\" */\n")
file.write("INSERT INTO expertise (idExpertise, designation)\nVALUES\n")
for especialidade in especialidades:
    if (i == len(especialidades)):
        file.write(f'\t({i}, \'{especialidade}\');\n\n')

    else:
        file.write(f'\t({i}, \'{especialidade}\'),\n')

    i += 1

# Código Postal
i = 1
file.write("/* Inserir os Códigos Postais na tabela \"zipcode\" */\n")
file.write("INSERT INTO zipcode (zipcode, city)\nVALUES\n")
for zipcode in zipcodes:
    if (i == len(zipcodes)):
        file.write(f'\t(\'{zipcode}\', \'{cidades[random.randint(0, len(zipcodes) - 1)}\');\n\n')

    else:
        file.write(f'\t(\'{zipcode}\', \'{cidades[random.randint(0, len(zipcodes) - 1)}\'),\n')

    i += 1

# Categorias
i = 1
file.write("/* Inserir as Categorias na tabela \"category\" */\n")
file.write("INSERT INTO category (idCategory, nameCategory)\nVALUES\n")
for categoria in categorias:
    if (i == len(categorias)):
        file.write(f'\t({i}, \'{categoria}\');\n\n')

    else:
        file.write(f'\t({i}, \'{categoria}\'),\n')

    i += 1

# Médicos
i = 1
file.write("/* Inserir os Médicos na tabela \"doctor\" */\n")
file.write("INSERT INTO doctor (idDoctor, birthdate, nameDoctor, idZipcode, cellphone,
for medico in medicos:
    idd = random.randint(10000000, 99999999)

```

```

ids_doctor.append(idd)
ano = random.randint(1970,1989)
mes = random.randint(1,12)
dias = random.randint(1,28)

if dias < 10 and mes < 10:
    birthdate = f'{ano}-0{mes}-0{dias}'
elif dias < 10 and mes >= 10:
    birthdate = f'{ano}-{mes}-0{dias}'
elif dias >= 10 and mes < 10:
    birthdate = f'{ano}-0{mes}-{dias}'
else:
    birthdate = f'{ano}-{mes}-{dias}'

if (i == len(medicos)):
    file.write(f'\t({idd}, \'{birthdate}\', \'{medico}\', \'{zipcodes[random.randi

else:
    file.write(f'\t({idd}, \'{birthdate}\', \'{medico}\', \'{zipcodes[random.randi

i += 1

# Atletas
i = 1
file.write("/ * Inserir os Atletas na tabela \"athlete\" */\n")
file.write("INSERT INTO athlete (idAthlete, nameAthlete, birthdate, weight, idModality
for nome in atletas_nomes:
    idd = random.randint(10000000, 99999999)
    ids_athlete.append(idd)
    ano = random.randint(1990,2019)
    mes = random.randint(1,12)
    dias = random.randint(1,28)

    if dias < 10 and mes < 10:
        birthdate = f'{ano}-0{mes}-0{dias}'
    elif dias < 10 and mes >= 10:
        birthdate = f'{ano}-{mes}-0{dias}'
    elif dias >= 10 and mes < 10:
        birthdate = f'{ano}-0{mes}-{dias}'
    else:
        birthdate = f'{ano}-{mes}-{dias}'

    if (i == len(atletas_nomes)):
        file.write(f'\t({idd}, \'{nome}\', \'{birthdate}\', {round(random.uniform(40,1

    else:
        file.write(f'\t({idd}, \'{nome}\', \'{birthdate}\', {round(random.uniform(40,1

i += 1

# Consultas
i = 1
number = 400

```

```
file.write("/ * Inserir as Consultas na tabela \"appointment\" */\n")
file.write("INSERT INTO appointment (idDoctor, idAthlete, observations, price, dateApp
for i in range(number):
    ano = random.randint(2016,2022)
    mes = random.randint(1,12)
    dias = random.randint(1,28)
    horas = random.randint(10,23)
    minutos = random.randint(0, 1) * 30
    segundos = 0

    if dias < 10 and mes < 10:
        date = f'{ano}-{mes}-0{dias}'
    elif dias < 10 and mes >= 10:
        date = f'{ano}-{mes}-0{dias}'
    elif dias >= 10 and mes < 10:
        date = f'{ano}-0{mes}-{dias}'
    else:
        date = f'{ano}-{mes}-{dias}'

    datetime = f'{date} {horas}:{minutos if minutos > 9 else "0" + str(minutos)}:{segu

    if (i == number - 1):
        file.write(f'\t({ids_doctor[random.randint(0, len(ids_doctor) -1)]}, {ids_athl

    else:
        file.write(f'\t({ids_doctor[random.randint(0, len(ids_doctor) - 1)]}, {ids_ath

    i += 1
except Exception as err:
    # Parar a thread da animação caso algo corra mal.
    complete = True
    animation_func.join()
    print(err) # Mostrar o erro.
    raise

complete = True
animation_func.join()
print("Ficheiro db_populate.sql gerado com sucesso!")
print(f'Tempo de execução: {round(time.time() - start, 3)}s')
```