Исследование зоопрака

Установка зависимостей

In [165…
```
%pip install pandas openpyxl faker psycopg2 sqlalchemy pylint xlsxwriter
```

```
Requirement already satisfied: pandas in c:\users\ab54\desktop\pythonproject\.venv\lib\s
ite-packages (2.2.1)
Requirement already satisfied: matplotlib in c:\users\ab54\desktop\pythonproject\.venv\l
ib\site-packages (3.8.3)
Requirement already satisfied: seaborn in c:\users\ab54\desktop\pythonproject\.venv\lib
\site-packages (0.13.2)
Requirement already satisfied: openpyxl in c:\users\ab54\desktop\pythonproject\.venv\lib
\site-packages (3.1.2)
Requirement already satisfied: faker in c:\users\ab54\desktop\pythonproject\.venv\lib\si
te-packages (24.3.0)
Requirement already satisfied: spicy in c:\users\ab54\desktop\pythonproject\.venv\lib\si
te-packages (0.16.0)
Requirement already satisfied: psycopg2 in c:\users\ab54\desktop\pythonproject\.venv\lib
\site-packages (2.9.9)
Requirement already satisfied: sqlalchemy in c:\users\ab54\desktop\pythonproject\.venv\l
ib\site-packages (2.0.29)
Requirement already satisfied: pylint in c:\users\ab54\desktop\pythonproject\.venv\lib\s
ite-packages (3.1.0)
Requirement already satisfied: xlsxwriter in c:\users\ab54\desktop\pythonproject\.venv\l
ib\site-packages (3.2.0)
Note: you may need to restart the kernel to use updated packages.
ERROR: Could not find a version that satisfies the requirement utf8 (from versions: non
e)
ERROR: No matching distribution found for utf8

[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Импорт библиотек

In [134…
```
from faker import Faker
import pandas as pd
import psycopg2 as psql
from sqlalchemy import create_engine
import openpyxl
```

Настройка окружения

In [135…
```
fake = Faker()
```

Создание классов для генерации данных

Создание класса генератора зоопарков

In [136…
```
class ZooparkRowGenerator:
    def gen_name(self):
        return fake.company()
```

```
    def gen_country(self):
        return fake.country()
    def gen_city(self):
        return fake.city()
    def __iter__(self):
        return self
    def __next__(self):
        name = self.gen_name()
        country = self.gen_country()
        city = self.gen_city()
        row = [name,country,city]
        return row
```

Создание класса генератора поставщиков

In [137…

```
class ProviderRowGenerator:
    def gen_name(self):
        return fake.company()
    def __iter__(self):
        return self
    def __next__(self):
        name = self.gen_name()
        row = [name]
        return row
```

Создание класса генератора типов еды

In [138…

```
class FoodTypeRowGenerator:
    def gen_name(self):
        return fake.company()
    def __iter__(self):
        return self
    def __next__(self):
        name = self.gen_name()
        row = [name]
        return row
```

Создание класса генератора еды

In [139…

```
class FoodRowGenerator:
    def __init__(self, type_foods, n_foods):
        self._type_foods = type_foods
        self._n_foods = int(n_foods)
        self._state_foods = 0
        self._state_cur_food = 1
    def gen_name(self):
        return fake.company()
    def __iter__(self):
        return self
    def __next__(self):
        name = self.gen_name()
        row = [self._state_foods, name]
        self.update_state()
        return row
    def update_state(self):
        if self._state_cur_food == self._n_foods:
            if self._state_foods == len(self._type_foods) - 1:
                self._state_cur_food = 1
                self._state_foods = 0
            else:
                self._state_cur_food = 1
```

```
                self._state_foods += 1
        else:
            self._state_cur_food += 1
```

Функция создания файла

In [140…
```
def create_file(file_name, file_path, header,  num_rows, generator):
    file = file_path + file_name + '.xlsx'
    with pd.ExcelWriter(file, engine='xlsxwriter') as writer:
        workbook = writer.book
        worksheet = workbook.add_worksheet()
        worksheet.write_row(0, 0, header)
        for i in range(num_rows):
            row_data = next(generator)
            worksheet.write_row(i + 1, 0, row_data)
```

Генерация данных

Генерация данных зоопарков

In [141…
```
create_file('Zooparks', 'C:/Users/ab54/Desktop/data/', ['Наименование зоопарка', 'Страна
```

Генерация данных поставщиков

In [142…
```
create_file('Providers', 'C:/Users/ab54/Desktop/data/', ['Наименование поставщика'], 5,
```

Генерация данных типов еды

In [143…
```
create_file('Types_of_food', 'C:/Users/ab54/Desktop/data/', ['Тип еды'], 5, FoodTypeRowG
```

Генерация данных еды

In [144…
```
dt_types = pd.read_excel('C:/Users/ab54/Desktop/data/Types_of_food.xlsx')
create_file('Food', 'C:/Users/ab54/Desktop/data/', ['Тип еды', 'Наименование'], 25, Food
```

Получение сгенерированных данных

Получение и анализ данных зоопарков

In [145…
```
dt_zooparks = pd.read_excel("C:/Users/ab54/Desktop/data/Zooparks.xlsx")
```

In [146…
```
dt_zooparks.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Наименование зоопарка  10 non-null     object
 1   Страна                 10 non-null     object
 2   Город                  10 non-null     object
```

```
dtypes: object(3)
memory usage: 372.0+ bytes
```

In [147…
```
dt_zooparks.describe()
```

Out[147]:

	Наименование зоопарка	Страна	Город
count	10	10	10
unique	10	10	10
top	Garcia-Lee	Azerbaijan	Loribury
freq	1	1	1

Получение и анализ данных зоопарков

In [148…
```
dt_providers = pd.read_excel("C:/Users/ab54/Desktop/data/Providers.xlsx")
```

In [149…
```
dt_providers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 1 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Наименование поставщика  5 non-null      object
dtypes: object(1)
memory usage: 172.0+ bytes
```

In [150…
```
dt_providers.describe()
```

Out[150]:

	Наименование поставщика
count	5
unique	5
top	Young-Beard
freq	1

Анализ данных типов еды

In [151…
```
dt_types.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Тип еды  5 non-null      object
dtypes: object(1)
memory usage: 172.0+ bytes
```

In [152…
```
dt_types.describe()
```

Out[152]:

	Тип еды
count	5
unique	5
top	Higgins, Hayden and Kelly

freq	1

Получение и анализ данных еды

In [153…
```
dt_foods = pd.read_excel("C:/Users/ab54/Desktop/data/Food.xlsx")
```

In [154…
```
dt_foods.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Тип еды       25 non-null     int64 
 1   Наименование  25 non-null     object
dtypes: int64(1), object(1)
memory usage: 532.0+ bytes
```

In [155…
```
dt_foods.describe()
```

Out[155]:

	Тип еды
count	25.000000
mean	2.000000
std	1.443376
min	0.000000
25%	1.000000
50%	2.000000
75%	3.000000
max	4.000000

Работа с БД

Подготовка данных к внесению

Создание id

In [158…
```
id_zooparks = [i for i in range(0, len(dt_zooparks))]
id_providers = [i for i in range(0, len(dt_providers))]
id_types = [i for i in range(0, len(dt_types))]
id_food = [i for i in range(0, len(dt_foods))]
```

Приводим данные к виду наших таблиц

In [159…
```
df_zooparks = pd.DataFrame({'id': id_zooparks, 'name': dt_zooparks['Наименование зоопарк
df_providers = pd.DataFrame({'id': id_providers, 'name': dt_providers['Наименование пост
df_types = pd.DataFrame({'id': id_types, 'name': dt_types['Тип еды']})
df_foods = pd.DataFrame({'id': id_food, 'type_of_food': dt_foods['Тип еды'], 'name': dt_
```

Открываем скрипт создания БД

In [163…
```
with open("C:/Users/ab54/Desktop/data/Create_zoo_script.txt", 'r', encoding='utf-8') as
    sql_create_script = file.read()
```

Создаём подключение

In [183…
```
connection = psql.connect(
    dbname="zooparkdb",
    user="postgres",
    password="postgres",
    host="localhost",
)
cursor = connection.cursor()
```

Создаём БД

In [184…
```
cursor.execute(sql_create_script)
connection.commit()
```

Внесение данных в БД

Внесение данных в таблицу зоопарки

In [185…
```
for index, row in df_zooparks.iterrows():
    query = 'INSERT INTO zoo.zoo(id, name, country, city) VALUES (%s, %s, %s, %s)'
    cursor.execute(query, (row['id'], row['name'], row['country'], row['city']))
```

Внесение данных в таблицу поставщиков

In [186…
```
for index, row in df_providers.iterrows():
    query = 'INSERT INTO zoo.provider(id, name) VALUES (%s, %s)'
    cursor.execute(query, (row['id'], row['name']))
```

Внесение данных в таблицу типы еды

In [188…
```
for index, row in df_types.iterrows():
    query = 'INSERT INTO zoo.food_type(id, name) VALUES (%s, %s)'
    cursor.execute(query, (row['id'], row['name']))
```

Внесение данных в таблицу еда

In [189…
```
for index, row in df_foods.iterrows():
    query = 'INSERT INTO zoo.food(id, type_id, name) VALUES (%s, %s, %s)'
    cursor.execute(query, (row['id'], row['type_of_food'], row['name']))
```

Проверка внесённых данных

Проверка внесённых данных в таблицу зоопарки

In [190…
```
print(df_zooparks)
```

```
query = 'SELECT * FROM zoo.zoo'
cursor.execute(query)
table = cursor.fetchall()
for row in table:
    print(row)
```

```
   id                       name                    country  \
0   0                  Garcia-Lee                  Azerbaijan
1   1              Fields-Alvarez                     Jamaica
2   2     Gould, Miller and Ross                     Belgium
3   3  Johnson, Green and Wilson                Sierra Leone
4   4               Nguyen-Foster                        Togo
5   5                Reynolds LLC                   Sri Lanka
6   6             Zimmerman-Bauer                Saint Helena
7   7   Smith, Lane and Martinez  Bouvet Island (Bouvetoya)
8   8                   Rose Ltd                       Benin
9   9                Farrell PLC      Saint Kitts and Nevis

                  city
0              Loribury
1            Hoffmanton
2             Grantport
3            Crystalton
4         East Bobbyview
5           Priscillaton
6           Matthewshire
7  South Deborahchester
8       Patriciaborough
9             Davisport
(0, 'Garcia-Lee', 'Azerbaijan', 'Loribury')
(1, 'Fields-Alvarez', 'Jamaica', 'Hoffmanton')
(2, 'Gould, Miller and Ross', 'Belgium', 'Grantport')
(3, 'Johnson, Green and Wilson', 'Sierra Leone', 'Crystalton')
(4, 'Nguyen-Foster', 'Togo', 'East Bobbyview')
(5, 'Reynolds LLC', 'Sri Lanka', 'Priscillaton')
(6, 'Zimmerman-Bauer', 'Saint Helena', 'Matthewshire')
(7, 'Smith, Lane and Martinez', 'Bouvet Island (Bouvetoya)', 'South Deborahchester')
(8, 'Rose Ltd', 'Benin', 'Patriciaborough')
(9, 'Farrell PLC', 'Saint Kitts and Nevis', 'Davisport')
```

Данные приведённые сверху показывают, что данные внесенны в нашу базу данных успешно, без потери строк и пропуска данных

Завершение работы с БД

In [194…

```
cursor.close()
connection.close()
```