

Data Science amb Python

Sprint 18

S18 T01: NoSQL database task

Cristiane de Souza da Silva

July 2021

Description

We are starting to get acquainted with NoSQL databases !!! Let's start with a few basic exercises

Nivel 1

Exercise 1

Create a NoSQL database using MongoDB. Add some sample data that allows you to check that you are able to process the information in a basic way.

Exercise 2

Connect the NoSQL database to Python using for example pymongo.

```
In [1]: ! pip install pymongo
```

```
Requirement already satisfied: pymongo in /Applications/anaconda3/lib/python3.8/site-packages (3.11.4)
```

```
In [7]: # import libraies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pymongo
```

```
In [8]: # Creating a Database

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["mydatabase"]
mydb
```

```
Out[8]: Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True), 'mydatabase')
```

```
In [9]: #Check if Database Exists

print(myclient.list_database_names())

['admin', 'config', 'local']
```

```
In [14]: # Add some sample data
mycol = mydb["customers"]

mylist = [
    {'Country': 'United States', 'First Name' : 'Marshall', 'Last Name': 'Bernadot', 'Test Score': 54},
    {'Country': 'Ghana', 'First Name' : 'Celinda', 'Last Name': 'Malkin', 'Test Score': 55},
    {'Country': 'Ukraine', 'First Name' : 'Guillermo', 'Last Name': 'Furze', 'Test Score': 56},
    {'Country': 'Greece', 'First Name' : 'Aharon', 'Last Name': 'Tunnow', 'Test Score': 57},
    {'Country': 'Russia', 'First Name' : 'Bail', 'Last Name': 'Goodwin', 'Test Score': 58},
    {'Country': 'Poland', 'First Name' : 'Cole', 'Last Name': 'Winteringham', 'Test Score': 59},
    {'Country': 'Sweden', 'First Name' : 'Emlyn', 'Last Name': 'Erricker', 'Test Score': 60},
    {'Country': 'Russia', 'First Name' : 'Cathee', 'Last Name': 'Sivewright', 'Test Score': 61},
    {'Country': 'China', 'First Name' : 'Barney', 'Last Name': 'Ingerson', 'Test Score': 62},
    {'Country': 'Uganda', 'First Name' : 'Sharla', 'Last Name': 'Papaccio', 'Test Score': 63}
]
```

```
In [15]: x = mycol.insert_many(mylist)
```

```
In [16]: #print list of the _id values of the inserted documents:
print(x.inserted_ids)

[ObjectId('60ddbb84dd473efd9b0c27b3'), ObjectId('60ddbb84dd473efd9b0c27b4'), ObjectId('60ddbb84dd473efd9b0c27b5'), ObjectId('60ddbb84dd473efd9b0c27b6'), ObjectId('60ddbb84dd473efd9b0c27b7'), ObjectId('60ddbb84dd473efd9b0c27b8'), ObjectId('60ddbb84dd473efd9b0c27b9'), ObjectId('60ddbb84dd473efd9b0c27ba'), ObjectId('60ddbb84dd473efd9b0c27bb'), ObjectId('60ddbb84dd473efd9b0c27bc')]
```

Nivel 2

Exercises 1

Load some simple queries to a Pandas Dataframe.

```
In [17]: # select data from a collection
x1 = mycol.find_one()
print(x1)

{'_id': ObjectId('60ddbb84dd473efd9b0c27b3'), 'Country': 'United States', 'First Name': 'Marshall', 'Last Name': 'Bernadot', 'Test Score': 54}
```

```
In [19]: x2 = mycol.find()
entries = list(x2)
entries[:5]
```

```
Out[19]: [{ '_id': ObjectId('60ddbb84dd473efd9b0c27b3'),
  'Country': 'United States',
  'First Name': 'Marshall',
  'Last Name': 'Bernadot',
  'Test Score': 54},
 {'_id': ObjectId('60ddbb84dd473efd9b0c27b4'),
  'Country': 'Ghana',
  'First Name': 'Celinda',
  'Last Name': 'Malkin',
  'Test Score': 51},
 {'_id': ObjectId('60ddbb84dd473efd9b0c27b5'),
  'Country': 'Ukraine',
  'First Name': 'Guillermo',
  'Last Name': 'Furze',
  'Test Score': 53},
 {'_id': ObjectId('60ddbb84dd473efd9b0c27b6'),
  'Country': 'Greece',
  'First Name': 'Aharon',
  'Last Name': 'Tunnow',
  'Test Score': 48},
 {'_id': ObjectId('60ddbb84dd473efd9b0c27b7'),
  'Country': 'Russia',
  'First Name': 'Bail',
  'Last Name': 'Goodwin',
  'Test Score': 46}]
```

```
In [43]: df_scores = pd.DataFrame(entries)
df_scores
```

```
Out[43]:
```

	_id	Country	First Name	Last Name	Test Score
0	60ddbb84dd473efd9b0c27b3	United States	Marshall	Bernadot	54
1	60ddbb84dd473efd9b0c27b4	Ghana	Celinda	Malkin	51
2	60ddbb84dd473efd9b0c27b5	Ukraine	Guillermo	Furze	53
3	60ddbb84dd473efd9b0c27b6	Greece	Aharon	Tunnow	48
4	60ddbb84dd473efd9b0c27b7	Russia	Bail	Goodwin	46
5	60ddbb84dd473efd9b0c27b8	Poland	Cole	Winteringham	49
6	60ddbb84dd473efd9b0c27b9	Sweden	Emlyn	Erricker	55
7	60ddbb84dd473efd9b0c27ba	Russia	Cathee	Sivewright	49
8	60ddbb84dd473efd9b0c27bb	China	Barny	Ingerson	57
9	60ddbb84dd473efd9b0c27bc	Uganda	Sharla	Papaccio	55

In [23]: *#Find document(s) with the country "Greece"*

```
myquery_1 = { "Country": "Greece" }

mydoc = mycol.find(myquery_1)

df_list1 = []
for x in mydoc:
    df_list1.append(x)

df_scores1 = pd.DataFrame(df_list1)
df_scores1
```

Out[23]:

	_id	Country	First Name	Last Name	Test Score
0	60ddbb84dd473efd9b0c27b6	Greece	Aharon	Tunnow	48

In [24]: *#Find documents where the First Name starts with the letter "G" or higher*

```
myquery_2 = { "First Name": { "$gt": "G" } }

mydoc = mycol.find(myquery_2)

df_list2 = []
for x in mydoc:
    df_list2.append(x)

df_scores2 = pd.DataFrame(df_list2)
df_scores2
```

Out[24]:

	_id	Country	First Name	Last Name	Test Score
0	60ddbb84dd473efd9b0c27b3	United States	Marshall	Bernadot	54
1	60ddbb84dd473efd9b0c27b5	Ukraine	Guillermo	Furze	53
2	60ddbb84dd473efd9b0c27bc	Uganda	Sharla	Papaccio	55

Nivel 3

Exercise 1

Generates a statistical summary of the information contained in the database

In this task, I will use other dataset because there are sufficient entries on it for the exercise.

In [25]: `df_scores.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   _id          10 non-null    object
1   Country      10 non-null    object
2   First Name   10 non-null    object
3   Last Name    10 non-null    object
4   Test Score   10 non-null    int64
dtypes: int64(1), object(4)
memory usage: 528.0+ bytes
```

```
In [46]: # How many unique countries there are in the dataframe
df_scores['Country'].nunique()
```

```
Out[46]: 9
```

```
In [47]: # How many entries per country
df_scores['Country'].value_counts()
```

```
Out[47]: Russia          2
Poland          1
Sweden          1
United States   1
Uganda          1
Greece          1
Ukraine         1
Ghana           1
China           1
Name: Country, dtype: int64
```

```
In [48]: score_mean = pd.DataFrame(df_scores.groupby('Country')['Test Score'].mean()
score_mean
```

```
Out[48]:
```

	Country	Test Score
0	China	57.0
1	Ghana	51.0
2	Greece	48.0
3	Poland	49.0
4	Russia	47.5
5	Sweden	55.0
6	Uganda	55.0
7	Ukraine	53.0
8	United States	54.0

```
In [72]: print('The maximum average score is', score_mean['Test Score'].max())
print('The country with the highest average score is',
      score_mean.at[score_mean['Test Score'].idxmax(), 'Country'])
```

The maximum average score is 57.0
 The country with the highest average score is China

We can see that China has the highest average test score of this dataframe.

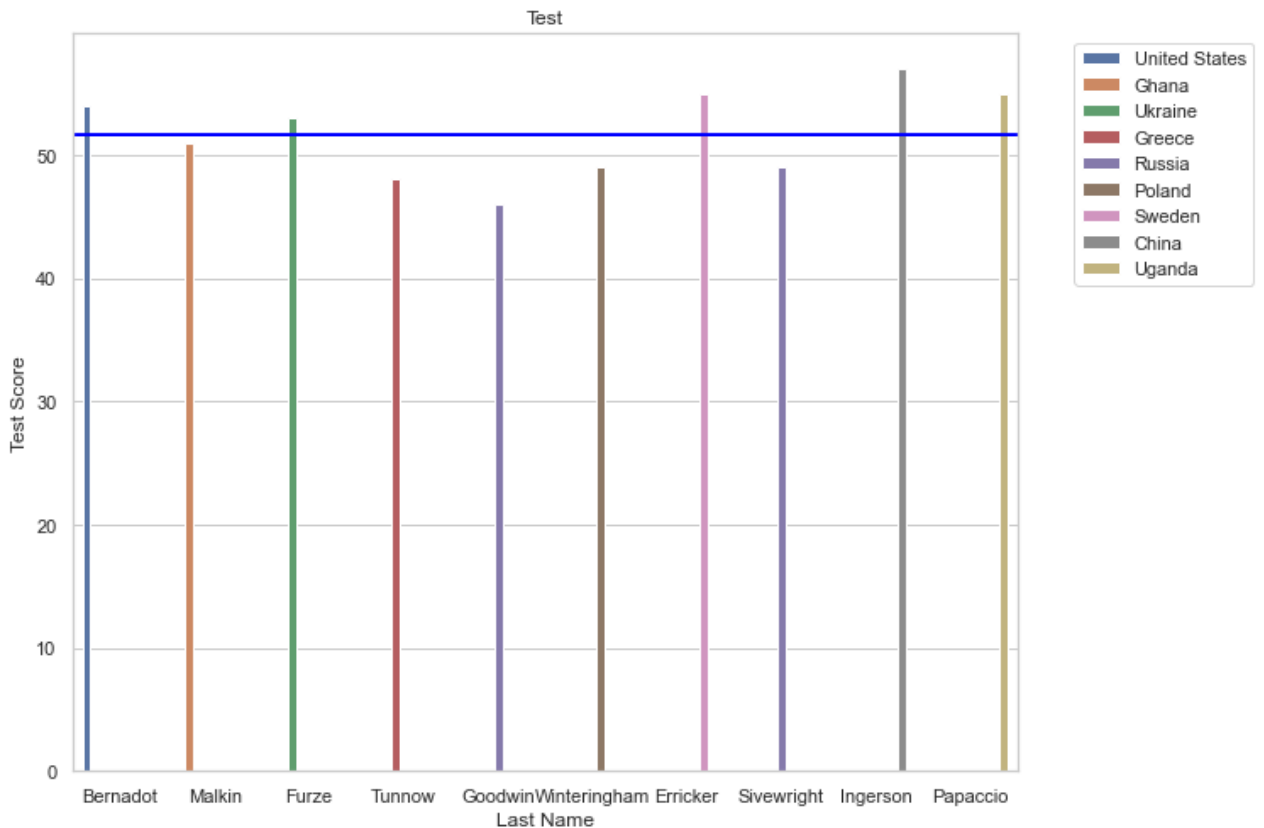
```
In [70]: score_mean.at[score_mean['Test Score'].idxmax(), 'Country']
```

```
Out[70]: 'China'
```

```
In [73]: # Average test score of all students
df_scores['Test Score'].mean()
```

```
Out[73]: 51.7
```

```
In [66]: plt.figure(figsize = (10,8))
sns.set_theme(style="whitegrid")
sns.barplot(x='Last Name', y='Test Score', hue='Country', data = df_scores)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
plt.title('Test')
plt.axhline(df_scores['Test Score'].mean(), color='blue', linewidth=2);
```



Individually, the largest note belongs to Ingerson of China, while the smallest belongs to Goodwin of Russia.

We can also see that 50% of students are above the average of 51.7.

```
In [ ]:
```