



BOSCH
Invented for life

Aula 3.9 - SQL com Python

inside.Docupedia Export

Author: Lundgren Daniel (CtP/ETS)
Date: 15-Dec-2021 15:58

Table of Contents

1	Inserção de dados	3
2	Seleção de dados	5
2.1	Selecionar os dados pelo python	5
2.2	Criação do Gráfico no Seaborn	5
2.3	Criação do gráfico em PowerBI	6
2.3.1	Abra o PowerBI Desktop.	7
2.3.2	Selecione Get Data.	8
2.3.3	Na aba Get Data selecione SQL Server database. Clique em Load.	8
2.3.4	No canto esquerdo em DATA está a tabela que você selecionou do seu Banco de dados.	9
2.3.5	Vamos separar a coluna timestamp, uma coluna para Data e outra para o Tempo.	9
2.3.6	Selecione Transform Data ou Transformar dados. Uma nova aba será aberta.	10
2.3.7	Vamos Duplicar a coluna Timestamp, com o botão direito do mouse selecione Duplicate Column ou duplicar coluna	11
2.3.8	Novamente com o botão direito do mouse em uma das colunas, mofique o tipo dos dados dessa coluna em Change Type, selecione Date. Modifique também o nome da coluna para 'Data'.	11
2.3.9	Criar o gráfico.	12
	No canto esquerdo selecione a aba Report. Vamos criar um gráfico de linha (Line chart).	12
	Vamos selecionar os eixos da seguinte forma.	13

1 Inserção de dados

Após recebermos os dados do firebase ou de algum outro método, precisamos incluí-los em um banco de dados.

Para isso será utilizado o Python como uma forma de nos conectarmos ao banco de dados para tanto alimentarmos esse banco de dados quanto para consumirmos os dados que estão no banco de dados.



Primeiro passo do nosso programa será fazer a requisição dos dados que stão alocados no Firebase.

Dados do Firebase

```
import requests
def sinal():
    proxies = {'https': 'https://<SEU_ID>:<SUA_SENHA>@rb-proxy-de.bosch.com:8080'}
    url_temperatura = 'https://esp32andfirebase-ec772-default-rtdb.firebaseio.com/Sensor/temperatura.json'
    url_umidade = 'https://esp32andfirebase-ec772-default-rtdb.firebaseio.com/Sensor/umidade.json'
    temperatura = float(requests.get(url_temperatura, proxies=proxies).content)
    umidade = float(requests.get(url_umidade, proxies=proxies).content)
    return temperatura, umidade
```

Segundo passo do nosso programa será **Entrar** no banco de dados pelo python, para isso vamos utilizar a biblioteca **pyodbc**.

Informações necessárias:

- **Server name**

Server type:	Database Engine
Server name:	CTNB1553\SQLEXPRESS01
Authentication:	SQL Server Authentication
Login:	admin
Password:	
<input type="checkbox"/> Remember password	

- **Nome do banco de dados** que você criou a sua tabela.
- **Login.**
- **Password.**

Além dessas informações precisamos do ODBC Driver 13.

Python entrando no BD

```
import pyodbc
def InserirBD(sinal):
    server = 'CTNB1553\SQLEXPRESS01'
    database = 'tempdbo'
    username = 'admin'
    password = 'admin'
    cnxn = pyodbc.connect('DRIVER={ODBC Driver 13 for SQL Server};SERVER='+server+';DATABASE='+database+';U
ID='+username+';PWD='+ password)
```

Vamos acrescentar na função acima o comando para inserirmos os dados nesse banco de dados.

Inserir

```
import pyodbc
def InserirBD(sinal):
    server = 'CTNB1553\\SQLEXPRESS01'
    database = 'tempdbo'
    username = 'admin'
    password = 'admin'
    cnxn = pyodbc.connect('DRIVER={ODBC Driver 13 for SQL Server};SERVER='+server+';DATABASE='+database+';U
ID='+username+';PWD='+ password)
    cursor = cnxn.cursor()
    cursor.execute(f"INSERT dbo.Sensor (Temperatura, Umidade) VALUES ({sinal[0]},{sinal[1]});")
    con.commit()
    print("Inserido com sucesso!")
```

Terceiro passo é criar uma função para apresentar os dados no console

Visualização

```
def apresentar(sinal):
    print(f"Temperatura: {sinal[0]}")
    print(f"Umidade: {sinal[1]}")
```

Por último vamos chamar todas as nossas funções

While

```
import time
while True:
    valores=sinal()
    apresentar(valores)
    InserirBD(valores)
    time.sleep(120)
```

Agora é só rodar o seu código para inserir os dados no banco de dados.

2 Seleção de dados

2.1 Selecionar os dados pelo python

Com os dados armazenados no banco de dados, vamos consumir esses dados e transformá-los em um gráfico através do Seaborn OU PowerBI.

Para consumir esses dados precisamos usar o comando SELECT.

Select

```
import matplotlib.pyplot as plt
import pyodbc
import numpy as np
server = 'CTNB1553\\SQLEXPRESS01'
database = 'tempdb'
username = 'admin'
password = 'admin'
driver= '{ODBC Driver 17 for SQL Server}'
conn= pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password)
cursor=conn.cursor()
cursor.execute("SELECT Temperatura, timestamp FROM dbo.Sensor")
row = cursor.fetchone()
lista=[]
listatempo=[]
while row:
    lista.append(row[0])
    listatempo.append(str(row[1]))
    row = cursor.fetchone()
```

- Fetchone - retorna uma linha da tabela por vez.
- Enquanto tiver linha eu vou separar os dados em duas listas:
 - A primeira (lista) irá receber o primeiro item da linha que é a temperatura.
 - A segunda (listatempo) irá receber o segundo item da linha que é o timestamp.

2.2 Criação do Gráfico no Seaborn

Depois que todos os dados estão devidamente separados podemos aplicá-los no gráfico SEABORN

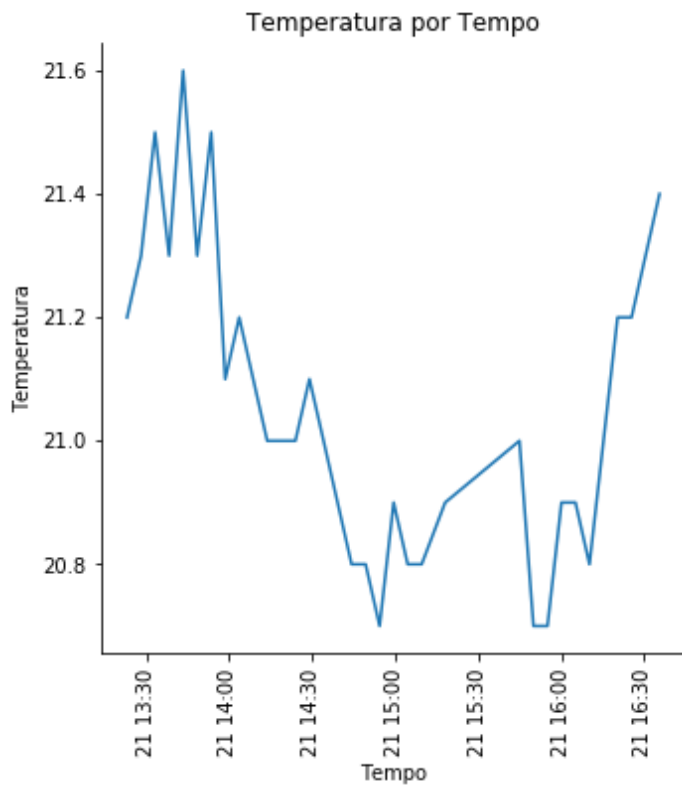
Seaborn

```
df=pd.DataFrame({"Temperatura":lista, "Tempo":listatempo})

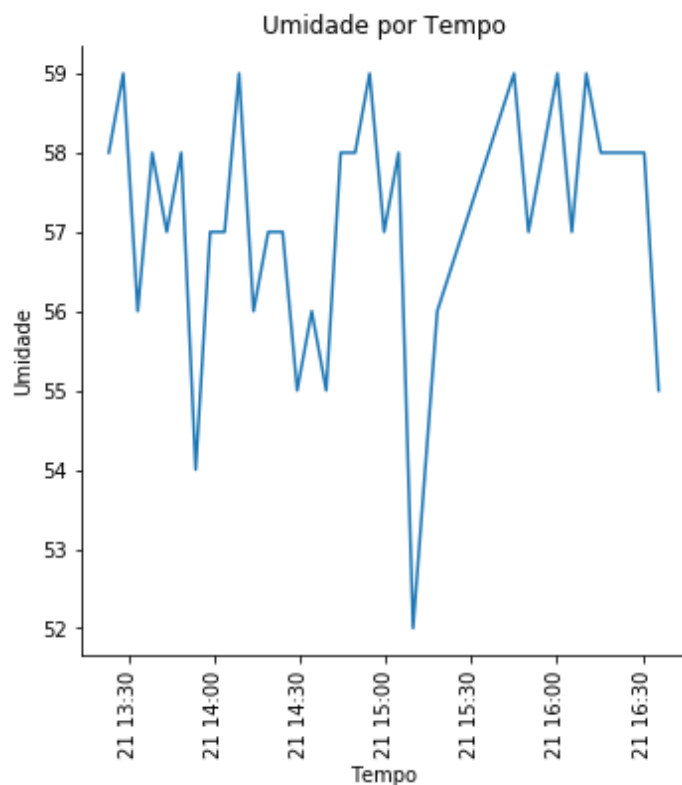
import seaborn as sns

df['Tempo'] = pd.to_datetime(df["Tempo"])
sns.relplot(x="Tempo", y="Temperatura", data=df, kind="line")
plt.title("Temperatura por Tempo")
plt.xticks(rotation=90)
```

- Criamos um dataframe para que seja possível utilizar o gráfico de Seaborn.



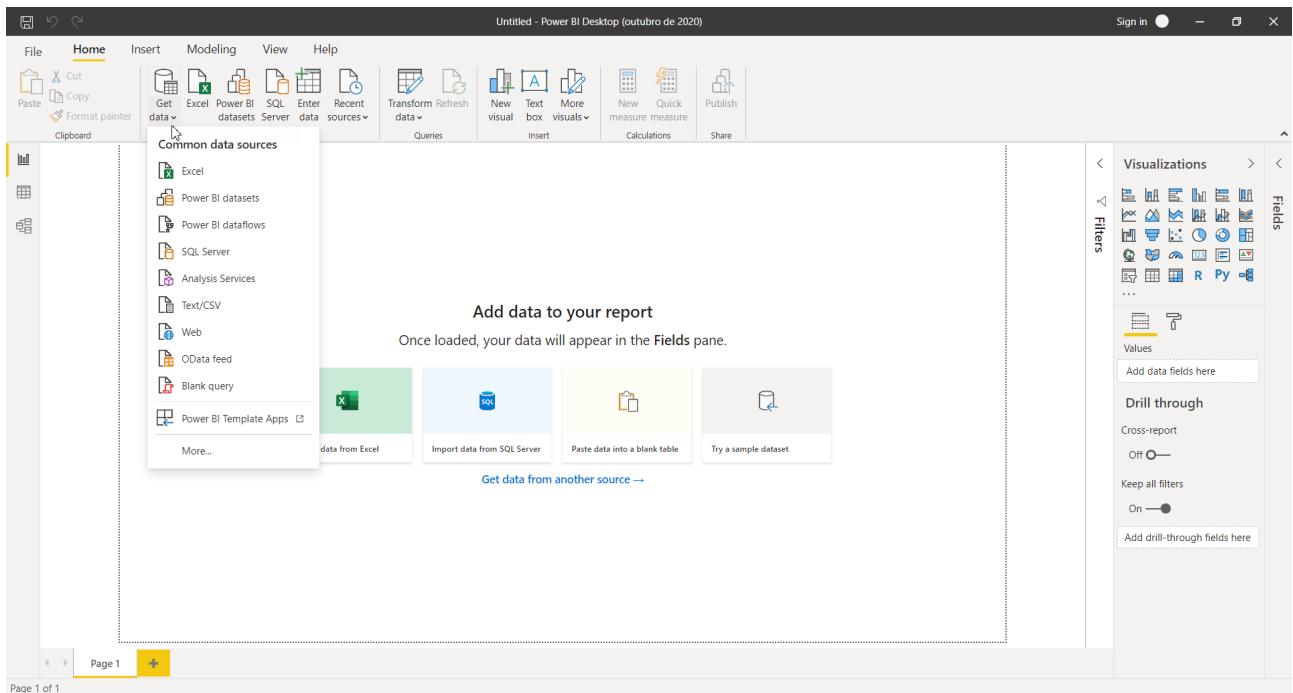
Repita os passos 2.1 e 2.2 para os dados de Umidade.



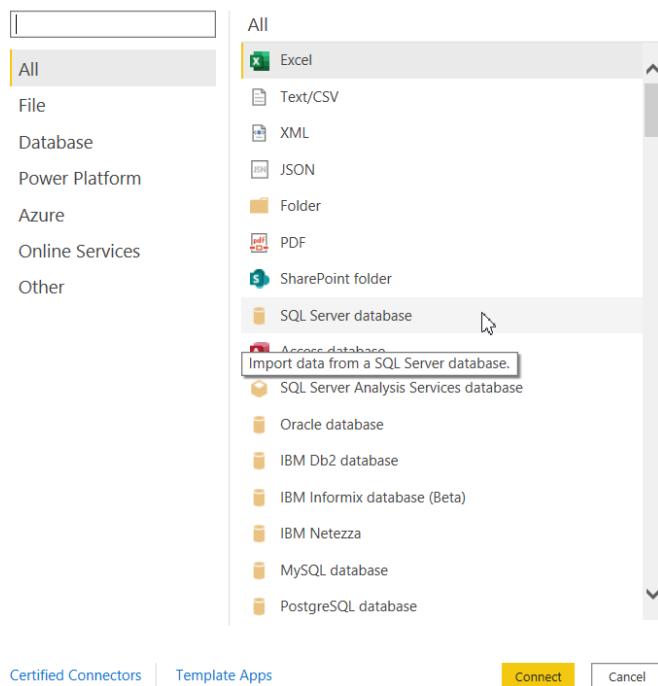
2.3 Criação do gráfico em PowerBI

Para criarmos o gráfico em PowerBi precisamos seguir os passos abaixo:

2.3.1 Abra o PowerBI Desktop.



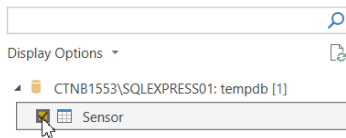
Get Data



2.3.2 Selezione Get Data.

2.3.3 Na aba Get Data selecione SQL Server database. Clique em Load.

Navigator



Sensor

Preview downloaded on sexta-feira

id	Temperatura	Umidade	timestamp
1	21,200001	58	21/05/2021 13:23:07
2	21,299999	59	21/05/2021 13:28:11
3	21,5	56	21/05/2021 13:33:15

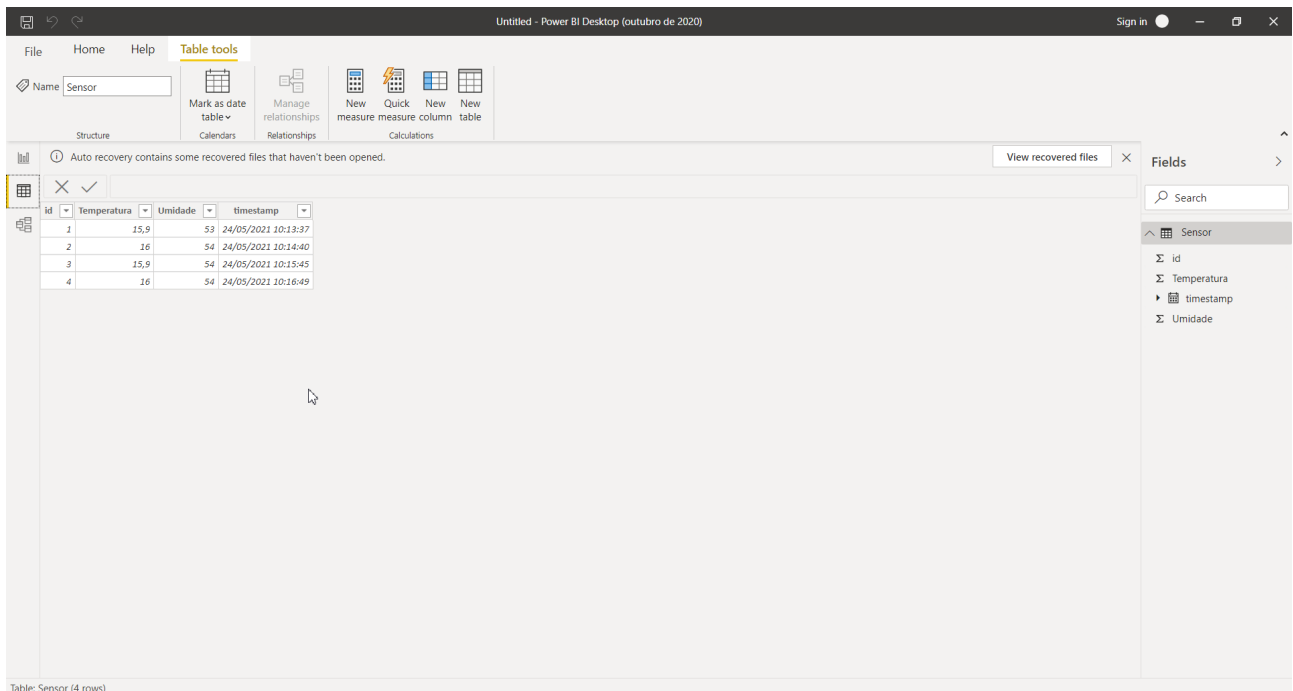
Select Related Tables

Load

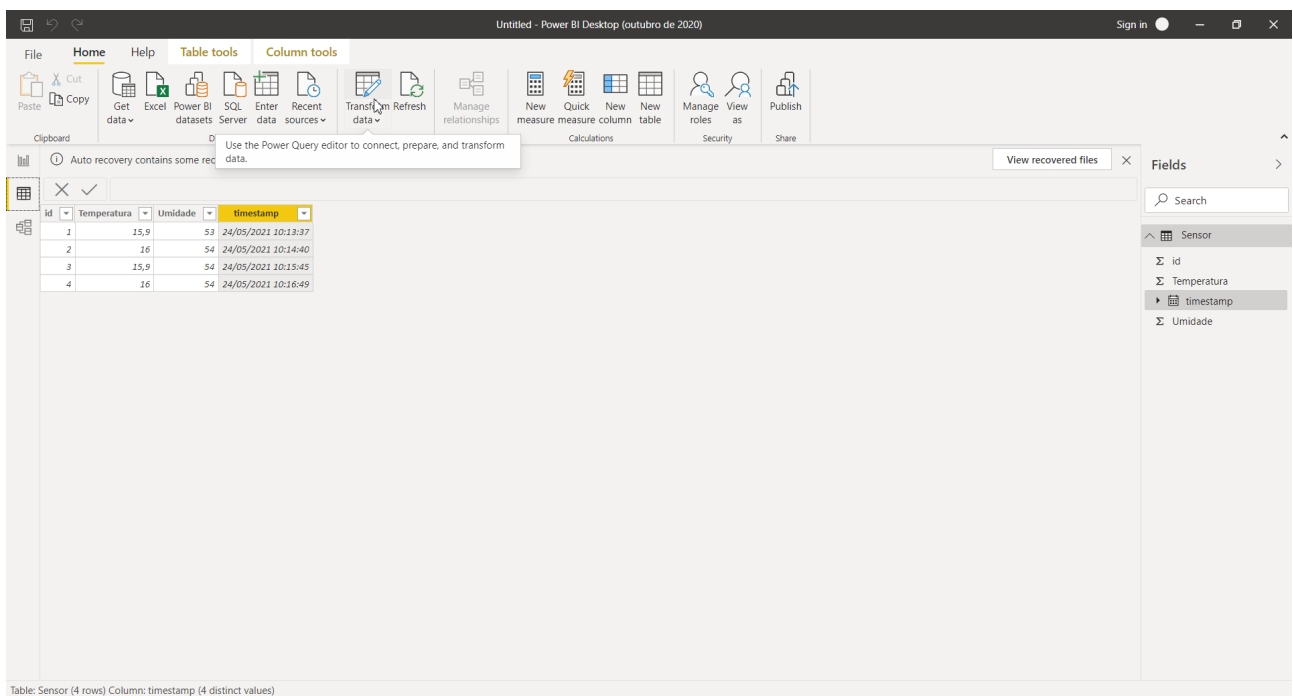
Transform Data

Cancel

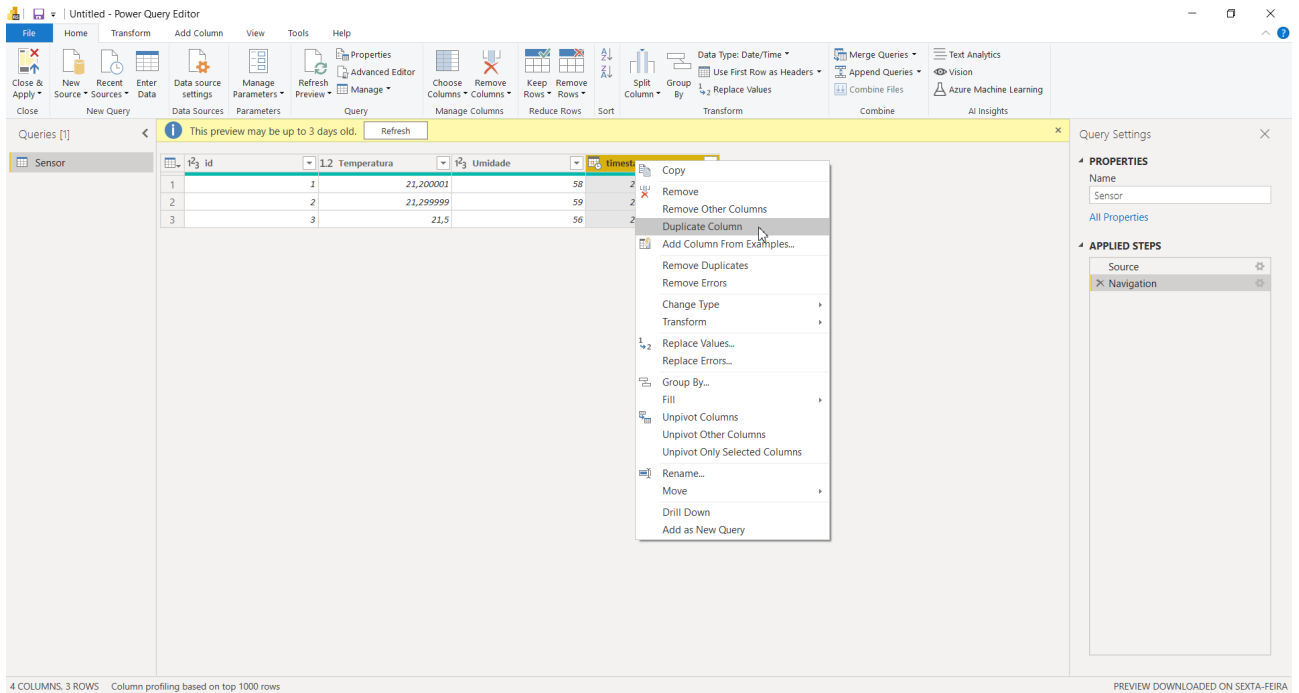
2.3.4 No canto esquerdo em DATA está a tabela que você selecionou do seu Banco de dados.



2.3.5 Vamos separar a coluna timestamp, uma coluna para Data e outra para o Tempo.



2.3.6 Selecione Transform Data ou Transformar dados. Uma nova aba será aberta.



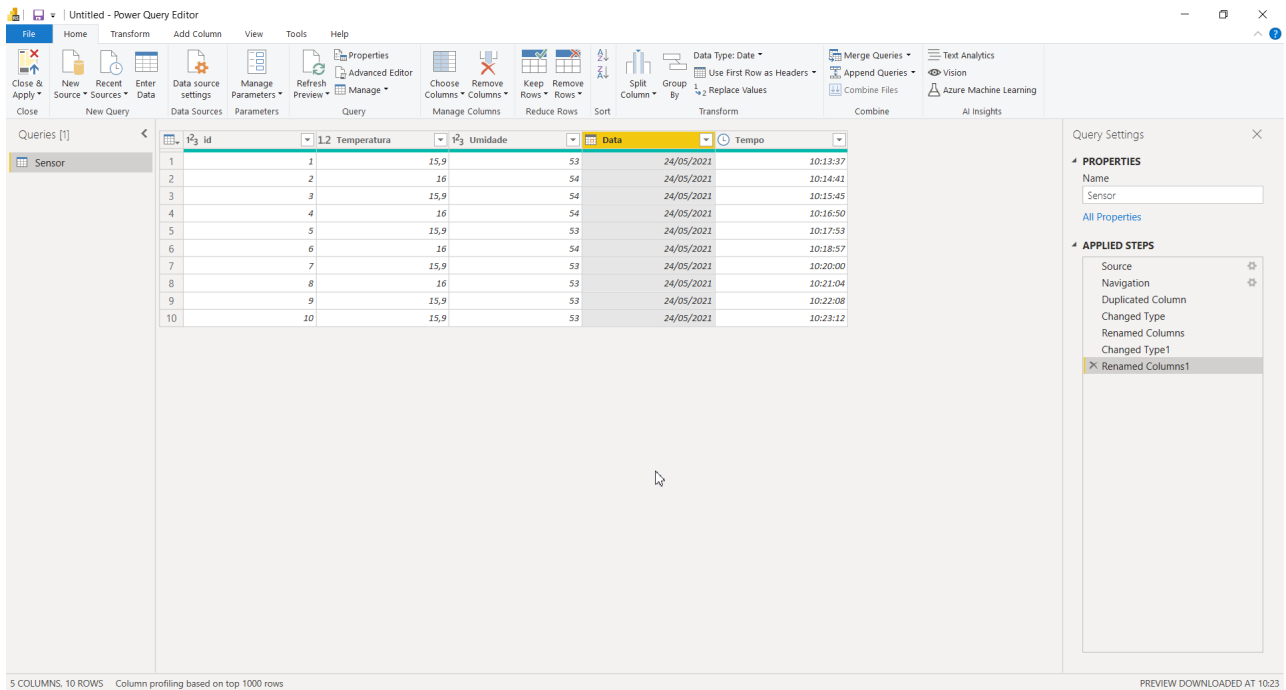
2.3.7 Vamos Duplicar a coluna Timestamp, com o botão direito do mouse selecione Duplicate Column ou duplicar coluna.

The first screenshot shows the Power Query Editor with a table containing 5 columns and 9 rows. The columns are 'id', 'Temperatura', 'Umidade', and 'timestamp'. A right-click context menu is open over the 'timestamp' column, and the 'Duplicate Column' option is selected. The 'APPLIED STEPS' pane on the right shows 'Duplicated Column' as the second step.

The second screenshot shows the same table, but now it has 5 columns and 10 rows. The 'timestamp' column has been duplicated and renamed to 'Data'. A right-click context menu is open over the 'Data' column, and the 'Change Type' option is selected. The 'APPLIED STEPS' pane on the right shows 'Changed Type' as the third step.

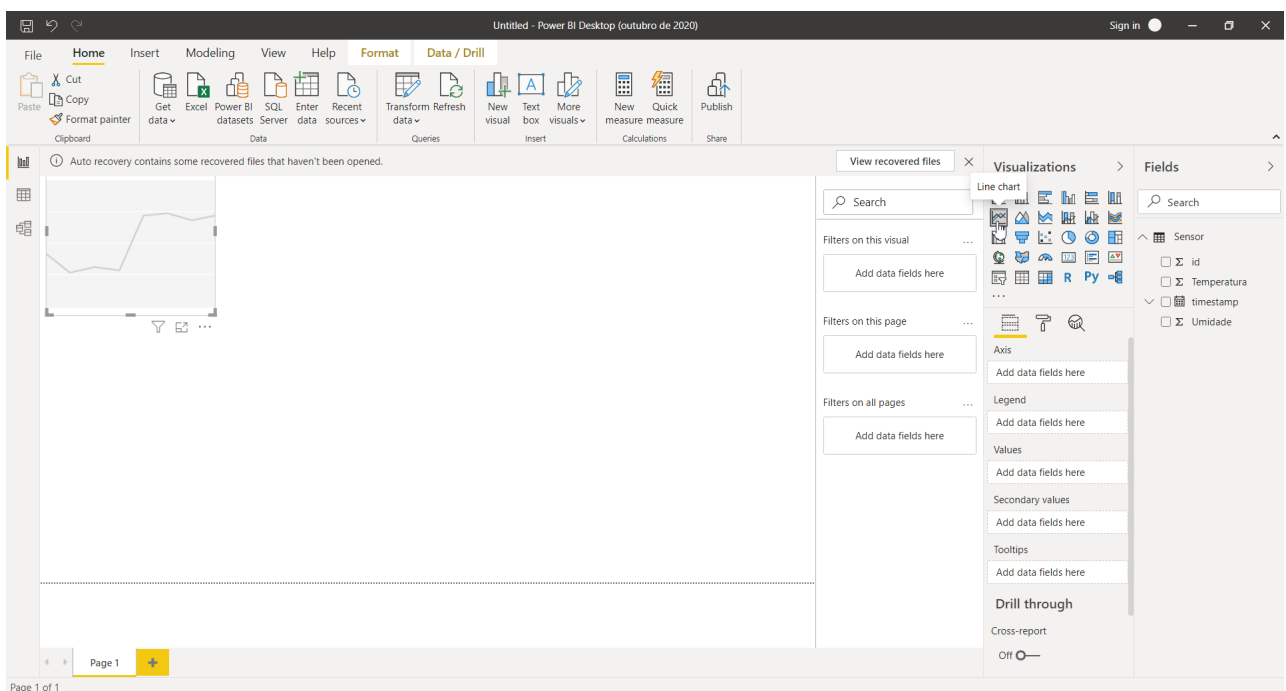
2.3.8 Novamente com o botão direito do mouse em uma das colunas, mofique o tipo dos dados dessa coluna em Change Type, selecione Date. Modifique também o nome da coluna para 'Data'.

Repita os passos 2.3.7 e 2.3.8 para modificar a segunda coluna para o tipo Time e o nome 'Tempo'.

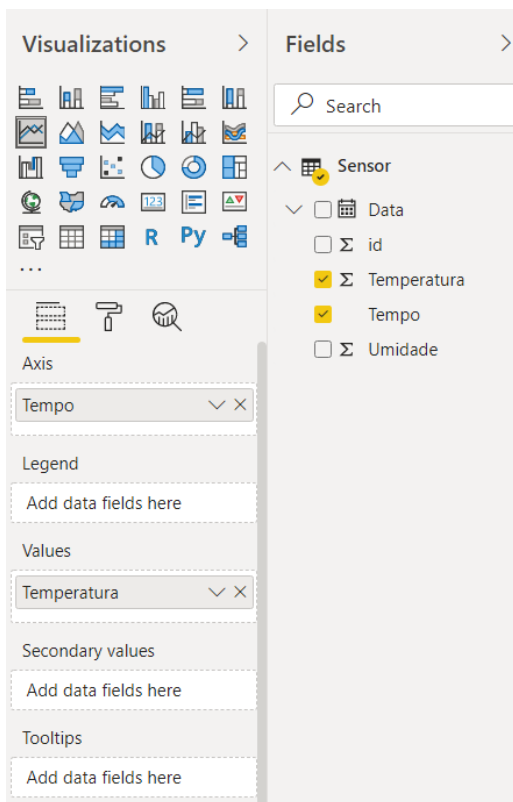


2.3.9 Criar o gráfico.

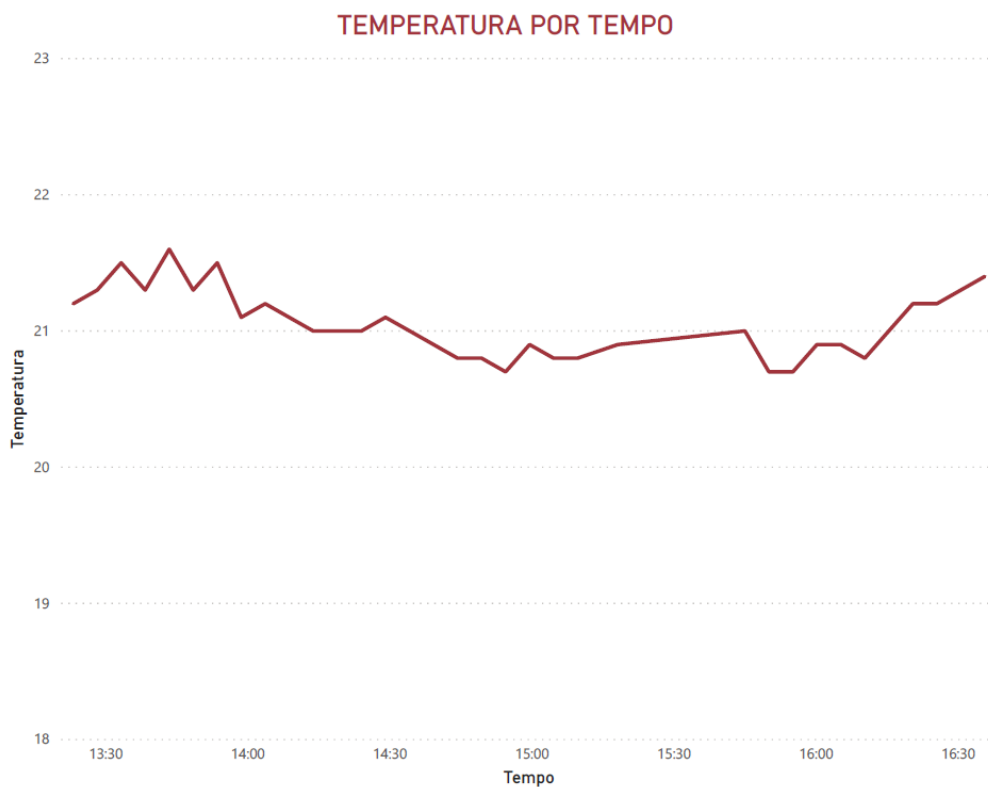
No canto esquerdo selecione a aba Report. Vamos criar um gráfico de linha (Line chart).



Vamos selecionar os eixos da seguinte forma.



Dessa forma, teremos um gráfico de temperatura por tempo.



Repita os passos 2.3.9.1 e 2.3.9.2 para criar o gráfico de Umidade por tempo.

