Carried out by:

YASSER El KARKOURI

ID : **S450915**

Cloud Computing : Instruction for the execution

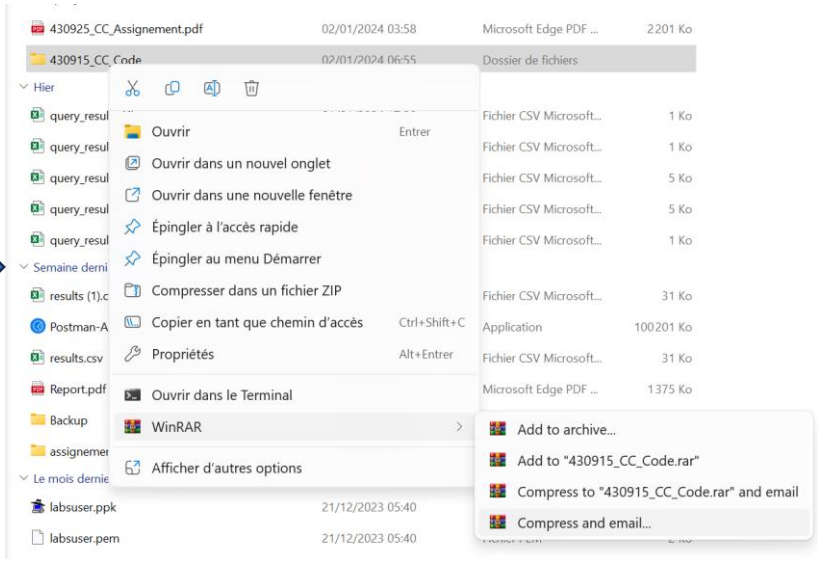
Collecting, Processing and Distributing IoT Data to Client

Module supervisor teacher:

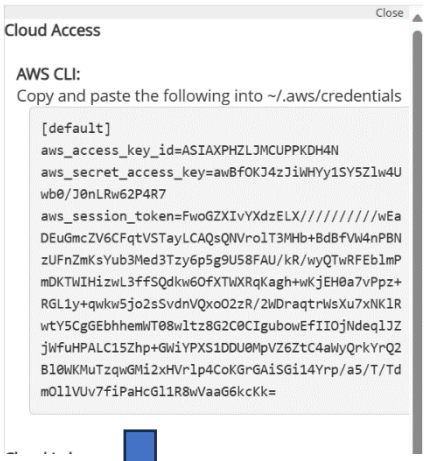
Dr Stuart Barnes

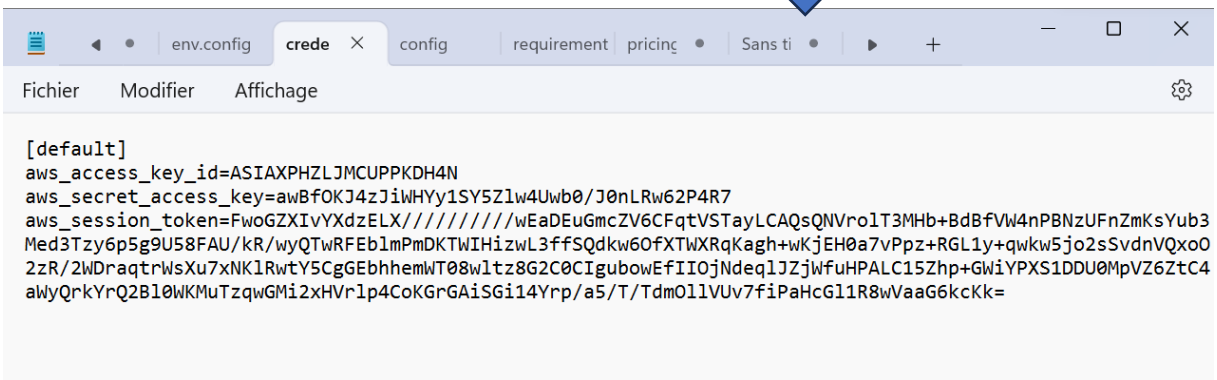
2 January 2024

After unzipping the folder of my project , it is recommended if you put it in downloads

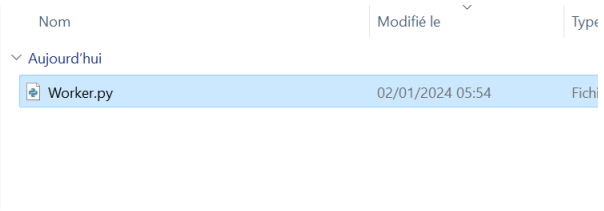


After launching the lab , copy the credentials ,and then update your credential file





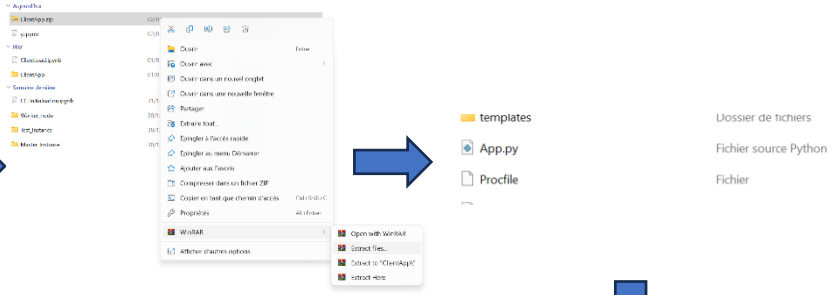
Open the file Worker.py in the folder Worker_node of the project folder



Update Credentials in the script ,

```
# Initialize AWS services
dynamodb = boto3.resource(
                          'dynamodb',
                          aws_access_key_id='ASIAXPHZLJMCUPPKDH4N',
                          aws_secret_access_key='awBfOKJ4zJiWHYy1SY5Zlw4Uwb0/J0nLRw62P4R7',
                          aws_session_token='FwoGZXIvYXdzELX/////////wEaDEuGmcZV6CFqtVSTayLCAQsQNVrolT3MH
                          region_name='us-east-1'
)

sqs = boto3.client(
                   'sqs',
                   aws_access_key_id='ASIAXPHZLJMCUPPKDH4N',
                   aws_secret_access_key='awBfOKJ4zJiWHYy1SY5Zlw4Uwb0/J0nLRw62P4R7',
                   aws_session_token='FwoGZXIvYXdzELX/////////wEaDEuGmcZV6CFqtVSTayLCAQsQNVrolT3MH
                   region_name='us-east-1'
)
```

Unzip the file ClientApp , and open the file App.py , then update credentials

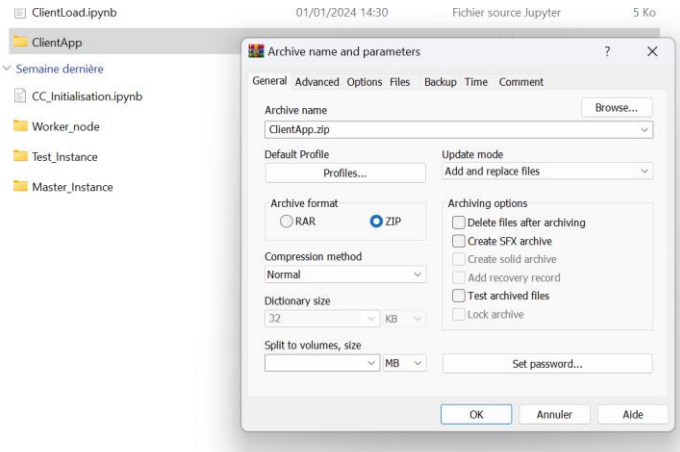



```
# Initialize the DynamoDB client
dynamodb = boto3.resource(
                          'dynamodb',
                          aws_access_key_id='ASIAXPHZLJMCUPPKDH4N',
                          aws_secret_access_key='awBfOKJ4zJiWHYy1SY5Zlw4Uwb
                          aws_session_token='FwoGZXIvYXdzELX/////////wEaDEuGmcZV6CFqtVSTayLCAQsQN
                          region_name='us-east-1'
)

sqs = boto3.client(
                   'sqs',
                   aws_access_key_id='ASIAXPHZLJMCUPPKDH4N',
                   aws_secret_access_key='awBfOKJ4zJiWHYy1SY5Zlw4Uwb0/J0nLRw62P4R7',
                   aws_session_token='FwoGZXIvYXdzELX/////////wEaDEuGmcZV6CFqtVSTayLCAQsQN
                   region_name='us-east-1'
)
```

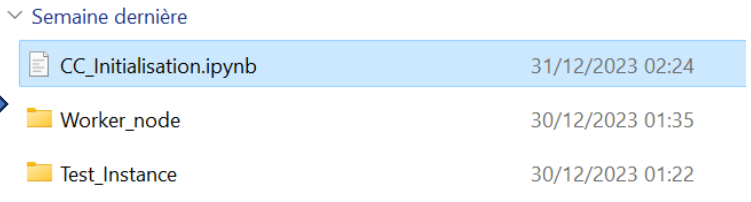
Open the file Master.py in the folder Master_Instance of the project folder


```python
# Initialize a DynamoDB client
dynamodb = boto3.resource('dynamodb',
                          aws_access_key_id='ASIAXPHZLJMCUPPKDH4N',
                          aws_secret_access_key='awBfOKJ4zJiWHYy1SY5Zlw4Uwb0/J0nLRw62P4R7',
                          aws_session_token='FwoGZXIvYXdzELX/////////wEaDEuGmcZV6CFqtVSTayLCAQsQNV
                          region_name='us-east-1')

# Define our DynamoDB table ,"QualityAirTable"
table = dynamodb.Table('QualityAirTable')

# Define a function to calculate the AQI and range label
def calculate_aqi(p1, p2):
```

After those Update compress the ClientApp in a zip file ,



Nowopen the Main program that is going to launch all our cloud elements



```python
def setup_ClientApp(instance, app_zip_path, public_ip):
    try:
        key = paramiko.RSAKey.from_private_key_file('C:/Users/elkar/Downloads/labsuser.pem')
        client = paramiko.SSHClient()
        client.set_missing_host_key_policy(paramiko.AutoAddPolicy())

        DNS = instance['PublicDnsName']
        client.connect(hostname=DNS, username='ec2-user', pkey=key)
```

Change the file path to your .pem file

```python
# Launch and set up the 'Master' EC2 instance
master_instance = create_instance(ec2, ami_id, instance_type, security_group_ids, key_name, user_data_master, 'Master')
if master_instance:
    # don't forget to change the file path to where you have downoaded the folder
    if upload_script(ec2, master_instance['InstanceId'], 'C:/Users/elkar/Downloads/430915_CC_Code/Master_Instance/master.py',
        if execute_script(ec2, master_instance['InstanceId'], key_path, 'master.py'):
            print("Script uploaded and executed on the 'Master' instance.")
        else:
            print("Failed to execute script on the 'Master' instance.")
    else:
```

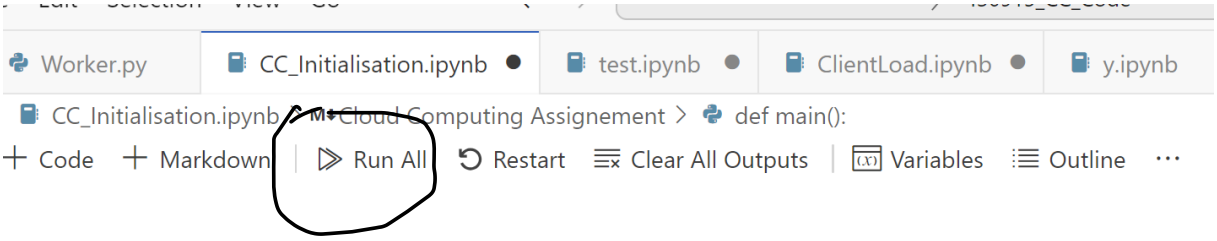
Change the file path to where the master.py is saved

```python
# Launch and set up the 'Worker' EC2 instance after 'Master' instance
worker_instance = create_instance(ec2, ami_id, instance_type, security_group_ids, key_name, user_data_client_load, 'Worker')
if worker_instance:
    # don't forget to change the file path to where you have downoaded the folder
    if upload_script(ec2, worker_instance['InstanceId'], 'C:/Users/elkar/Downloads/430915_CC_Code/Worker_node/Worker.py', key_
        if execute_script(ec2, worker_instance['InstanceId'], key_path, 'Worker.py'):
            print("Script uploaded and executed on the 'Worker' instance .")
        else:
            print("Failed to execute script on the 'Worker' instance.")
    else:
        print("Failed to upload script to the 'Worker' instance.")
```

Change the file path to where the Worker.py is saved

```
Launch and set up the 'Test_Instance' EC2 instance
ient_load_instance = create_instance(ec2, ami_id, instance_type, security_group_ids, key_name, user_data_client_load, 'Test_Instance')
 client_load_instance:
  # don't forget to change the file path to where you have downoaded the folder
  if upload_script(ec2, client_load_instance['InstanceId'], 'C:/Users/elkar/Downloads/430915_CC_Code/Test_Instance/TestPerformance.py',
      if execute_script(ec2, client_load_instance['InstanceId'], key_path, 'TestPerformance.py'):
          print("Script uploaded and executed on the 'Test_Instance' instance.")
      else:
          print("Failed to execute script on the 'Test_Instance' instance.")
  else:
      print("Failed to upload script to the 'Test_Instance' instance.")

ame__ == '__main__':
```

Change the file path to where the TestPerformance.py is saved



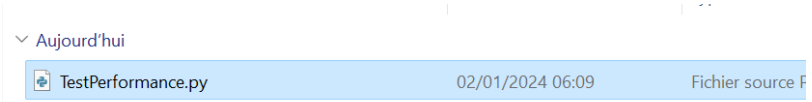
Cloud Computing Assignement

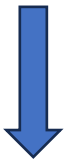
Now we can proceed to the program execution

```
Instance i-0bc2daf49d2bc42ea (Master) created, waiting for it to run...
Instance i-0bc2daf49d2bc42ea (Master) is running.
Script uploaded and executed on the 'Master' instance.
Instance i-0e260db1776b2922a (ClientApp) created, waiting for it to run...
Instance i-0e260db1776b2922a (ClientApp) is running.
b'Last metadata expiration check: 0:00:51 ago on Tue Jan  2 06:08:05 2024.\nDependencies resolved.\nNothing to do.\nComplete!\n'
b'Last metadata expiration check: 0:00:52 ago on Tue Jan  2 06:08:05 2024.\nPackage python3-3.9.16-1.amzn2023.0.6.x86_64 is already
b'Archive:  /home/ec2-user/ClientApp.zip\n   creating: /home/ec2-user/app/ClientApp/\n  inflating: /home/ec2-user/app/ClientApp/App.
b'Collecting Flask==2.0.1\n  Downloading Flask-2.0.1-py3-none-any.whl (94 kB)\nCollecting Werkzeug==2.0.0\n  Downloading Werkzeug-2.
Application deployed. Access it at http://34.238.146.179:8501
```

Once the program log shows the link to the web page you have to copy it before the program finishes the execution and then

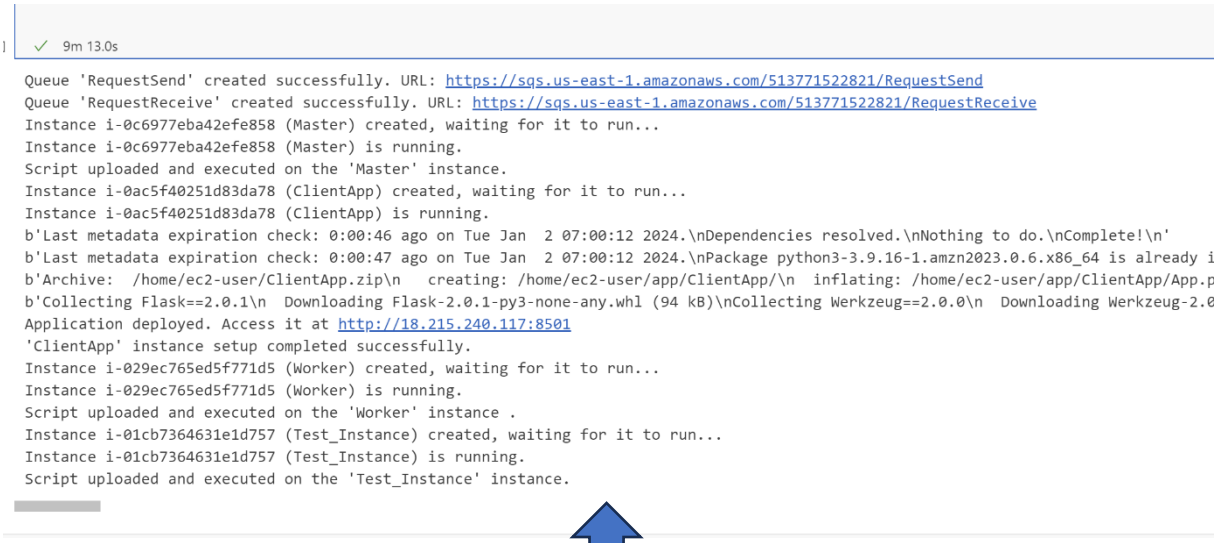
Open the file TestPerformance.py in the folder Test-Instance of the project folder





Paste the webpage link for further testing save and close file

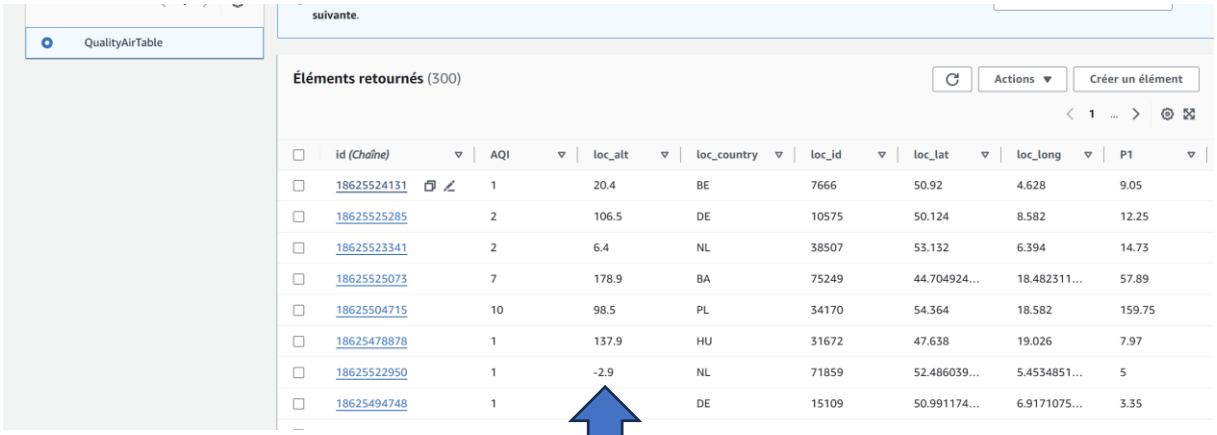
```python
import requests
from concurrent.futures import ThreadPoolExecutor
import time

# Endpoint URL of your Flask application
endpoint_url = 'http://127.0.0.1:5000/api/query'

# Sample payload for your Flask app's expected input
payload = {
    'startDate': '2024-01-01 00:00:00',
    'endDate': '2024-01-01 01:00:00',
    'country': 'DE',
```



After approximately 10 minute the program finishes setting up the elements running

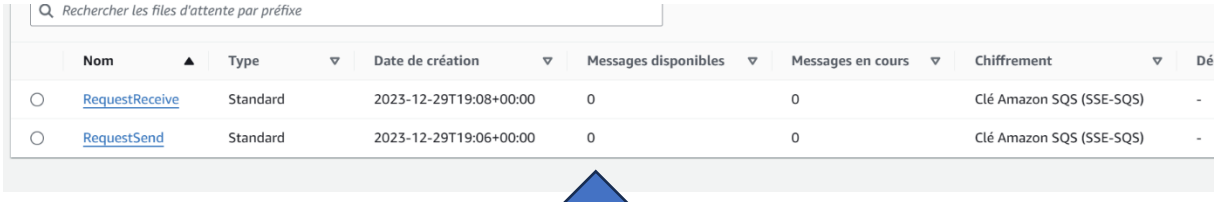
Name	ID d'instance	État de l'insta...	Type d'insta...	Contrôle des statu	Statu
ClientApp	i-0e260db1776b2922a	Résilié(e)	t2.micro	2/2 vérifications r	Aucur
Test_Instance	i-018e47341dfb1897d	Résilié(e)	t2.micro	2/2 vérifications r	Aucur
Worker	i-04051f56252e74e27	Résilié(e)	t2.micro	2/2 vérifications r	Aucur
Master	i-0bc2daf49d2bc42ea	Résilié(e)	t2.micro	2/2 vérifications r	Aucur

Instances were launched and set up correctly



id (Chaîne)	AQI	loc_alt	loc_country	loc_id	loc_lat	loc_long	P1
18625524131	1	20.4	BE	7666	50.92	4.628	9.05
18625525285	2	106.5	DE	10575	50.124	8.582	12.25
18625523341	2	6.4	NL	38507	53.132	6.394	14.73
18625525073	7	178.9	BA	75249	44.704924...	18.482311...	57.89
18625504715	10	98.5	PL	34170	54.364	18.582	159.75
18625478878	1	137.9	HU	31672	47.638	19.026	7.97
18625522950	1	-2.9	NL	71859	52.486039...	5.4534851...	5
18625494748	1		DE	15109	50.991174...	6.9171075...	3.35

DynamoDB was created and populated with the elements from the server



Nom	Type	Date de création	Messages disponibles	Messages en cours	Chiffrement	Dé
RequestReceive	Standard	2023-12-29T19:08+00:00	0	0	Clé Amazon SQS (SSE-SQS)	-
RequestSend	Standard	2023-12-29T19:06+00:00	0	0	Clé Amazon SQS (SSE-SQS)	-

2 SQS queues were created

```
wnloading Flask-2.0.1-py3-none-any.wl
 at http://18.215.240.117:8501
leted successfully.
orker) created, waiting for it to ru
orker) is running.
```

Access the webpage from the provided link in the CC_Initialisation.ipynb

Data Query Interface

Start Date: 01/01/2024 00:00:00

End Date: 02/01/2024 02:00:00

Country: DE

Minimum AQI: 0

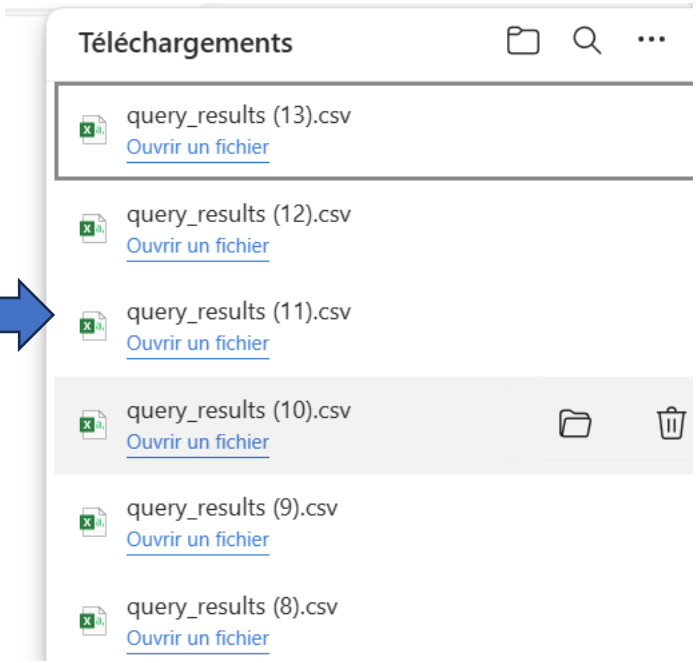
Maximum AQI: 10

Range: All ranges

Submit Query

Filter data and , pay attention to the day the data was collected in it if you have generated the code in 01/01/2024 the period between Start Date and End Date should be 24 HOURS

A csv file called query_results is being uploaded automatically after fiew seconds



id	AQI	loc_alt	loc_country	loc_id	loc_lat	loc_long	P1	P2	Range	timestamp
1,8525E+10	1	331.8	DE	39775	49.00339945	12.16260455	0.78	0.45	Low	25/12/2023 19:04
1,8525E+10	1	51.7	ES	63577	39.766	2.72	9.95	6.06	Low	25/12/2023 19:05
1,8525E+10	1	343.3	CZ	60387	49.994	14.66	0.12	0.08	Low	25/12/2023 19:03
1,8525E+10	5	170.5	IT	26989	45.524	10.938	45.59	23.61	Medium	25/12/2023 19:03
1,8525E+10	3	78.9	HU	877	46.266	20.062	31.48	20.17	Low	25/12/2023 19:03
1,8525E+10	1	598.0	DE	65695	50.65207062	13.16601669	0.72	0.24	Low	25/12/2023 19:04
1,8525E+10	1	63.5	DE	71018	50.74243273	7.093594463	2.11	0.79	Low	25/12/2023 19:03
1,8525E+10	2	178.2	RU	17573	55.922	37.54	12.22	7.86	Low	25/12/2023 19:05
1,8525E+10	10	30.4	BE	4464	50.882	4.73	1999.9	999.9	Very High	25/12/2023 19:04
1,8525E+10	3	533.8	BG	47923	42.506	24.192	32.62	18.56	Low	25/12/2023 19:03
1,8525E+10	1	435.8	DE	10202	48.474	9.138	4.01	2.15	Low	25/12/2023 19:04
1,8525E+10	1	228.8	HU	63050	47.514	18.92	8.59	6.23	Low	25/12/2023 19:01
1,8525E+10	1	341.5	DE	34289	48.572	13.378	5.49	3.02	Low	25/12/2023 19:04
1,8525E+10	1	70.4	GB	49390	51.38	0.098	2.13	1.12	Low	25/12/2023 19:04
1,8525E+10	1	539.7	DE	28767	48.12255939	11.49409174	1.32	0.68	Low	25/12/2023 19:02
1,8525E+10	1	133.2	HU	57541	47.408	18.822	3.89	1.71	Low	25/12/2023 19:05
1,8524E+10	1	84.4	PL	21624	52.29968379	21.09473168	5.74	1.87	Low	25/12/2023 17:44
1,8525E+10	1	52.2	FR	73879	48.88888890	2.322899900	4.71	3.27	Low	25/12/2023 19:05
1,8525E+10	2	61.2	RU	64402	44.708	37.644	12.63	3.11	Low	25/12/2023 19:03
1,8525E+10	1	54.6	DE	7656	51.954	7.998	2.61	1.56	Low	25/12/2023 19:01

The data structure of the query you have requested