Sathish_Kumar_Rajendiran_Week7_MDB2_Class_Activity

August 13, 2020

Name: Sathish Kumar Rajendiran Task: Week7: MongoDB Class Activity 2 Date: 8/13/2020

[2]:
```
#import libraries
from urllib import request
import json
import pymongo
from pymongo import MongoClient
from datetime import datetime
import sys
import pandas as pd
import os
import csv
print("all libraries imported successfully")
```

```
all libraries imported successfully
```

[3]:
```
#verify current directory
print('working directory: ',os.getcwd())
```

```
working directory:  /Users/sathishrajendiran/ist652-python
```

[4]:
```
ls *.csv
```

```
Border_Crossing_Entry_Data.csv*       price_of_gasoline.csv
albb.salaries.2003.Pitchers.csv       tv_life.csv
albb.salaries.2003.under-million.csv
```

[5]:
```
# define file name
infile = 'Border_Crossing_Entry_Data.csv'
# Working with file, list and sorting
try:
    data = pd.read_csv(infile)
    print("data has been processed \n")
    print('top 2 rows:\n',data.head(2))
    print('\nbottom 2 rows:\n',data.tail(2))

except:
    print("Is the file in correct directory?")
```

```
data has been processed

top 2 rows:
   Port Name State  Port Code            Border            Date  \
0     Alcan    AK       3104  US-Canada Border  2/1/2020 00:00
1     Alcan    AK       3104  US-Canada Border  2/1/2020 00:00

                     Measure  Value
0  Personal Vehicle Passengers   1414
1            Personal Vehicles    763

bottom 2 rows:
        Port Name State  Port Code            Border            Date  \
355509   Carbury    ND       3421  US-Canada Border  1/1/1996 00:00
355510   Skagway    AK       3103  US-Canada Border  1/1/1996 00:00

                     Measure  Value
355509  Truck Containers Empty      0
355510                   Buses      3
```

[6]:
```
#check index
data.index
```

[6]:
```
RangeIndex(start=0, stop=355511, step=1)
```

[7]:
```
# dropping null value columns to avoid errors
data.dropna(inplace = True)
```

[8]:
```
# Change the column names
data.columns = ['PortName','State','PortCode','Border','Date','Measure','Value']
data.head()
```

[8]:
```
  PortName State  PortCode            Border            Date  \
0    Alcan    AK      3104  US-Canada Border  2/1/2020 00:00
1    Alcan    AK      3104  US-Canada Border  2/1/2020 00:00
2    Alcan    AK      3104  US-Canada Border  2/1/2020 00:00
3    Alcan    AK      3104  US-Canada Border  2/1/2020 00:00
4    Alcan    AK      3104  US-Canada Border  2/1/2020 00:00

                     Measure  Value
0  Personal Vehicle Passengers   1414
1            Personal Vehicles    763
2       Truck Containers Empty    412
3        Truck Containers Full    122
4                       Trucks    545
```

[9]:
```
# storing dtype before converting
before = data.dtypes
print('dtype before converting:\n',before)
```

```
dtype before converting:
 PortName    object
State       object
PortCode     int64
Border      object
Date        object
Measure     object
Value        int64
dtype: object
```

[10]:
```
#convert Area and Population values to float
data['Date'] = data['Date'].astype('datetime64[ns]')
# data['Value'] = data['Value'].astype('float64')
data.head()
```

[10]:
```
  PortName State  PortCode            Border       Date  \
0    Alcan    AK      3104  US-Canada Border 2020-02-01
1    Alcan    AK      3104  US-Canada Border 2020-02-01
2    Alcan    AK      3104  US-Canada Border 2020-02-01
3    Alcan    AK      3104  US-Canada Border 2020-02-01
4    Alcan    AK      3104  US-Canada Border 2020-02-01

                      Measure  Value
0  Personal Vehicle Passengers   1414
1           Personal Vehicles    763
2       Truck Containers Empty    412
3        Truck Containers Full    122
4                      Trucks    545
```

[11]:
```
# storing dtype before converting
after = data.dtypes

# printing to compare
print('Before Conversion:\n', before)
print('\nAfter Conversion:\n', after)
```

```
Before Conversion:
 PortName    object
State       object
PortCode     int64
Border      object
Date        object
Measure     object
Value        int64
```

```
dtype: object

After Conversion:
 PortName            object
State               object
PortCode             int64
Border              object
Date        datetime64[ns]
Measure             object
Value                int64
dtype: object
```

[12]:
```
#data frame shape
data.shape
```

[12]:
```
(355511, 7)
```

[13]:
```
# the url request was successful - convert the response to a string
data_json = json.loads(data.to_json(orient='records'))
```

[14]:
```
print(data_json[:10])
```

```
[{'PortName': 'Alcan', 'State': 'AK', 'PortCode': 3104, 'Border': 'US-Canada
Border', 'Date': 1580515200000, 'Measure': 'Personal Vehicle Passengers',
'Value': 1414}, {'PortName': 'Alcan', 'State': 'AK', 'PortCode': 3104, 'Border':
'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Personal Vehicles',
'Value': 763}, {'PortName': 'Alcan', 'State': 'AK', 'PortCode': 3104, 'Border':
'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Truck Containers Empty',
'Value': 412}, {'PortName': 'Alcan', 'State': 'AK', 'PortCode': 3104, 'Border':
'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Truck Containers Full',
'Value': 122}, {'PortName': 'Alcan', 'State': 'AK', 'PortCode': 3104, 'Border':
'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Trucks', 'Value': 545},
{'PortName': 'Alexandria Bay', 'State': 'NY', 'PortCode': 708, 'Border': 'US-
Canada Border', 'Date': 1580515200000, 'Measure': 'Bus Passengers', 'Value':
1174}, {'PortName': 'Alexandria Bay', 'State': 'NY', 'PortCode': 708, 'Border':
'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Buses', 'Value': 36},
{'PortName': 'Alexandria Bay', 'State': 'NY', 'PortCode': 708, 'Border': 'US-
Canada Border', 'Date': 1580515200000, 'Measure': 'Personal Vehicle Passengers',
'Value': 68630}, {'PortName': 'Alexandria Bay', 'State': 'NY', 'PortCode': 708,
'Border': 'US-Canada Border', 'Date': 1580515200000, 'Measure': 'Personal
Vehicles', 'Value': 31696}, {'PortName': 'Alexandria Bay', 'State': 'NY',
'PortCode': 708, 'Border': 'US-Canada Border', 'Date': 1580515200000, 'Measure':
'Truck Containers Empty', 'Value': 1875}]
```

[15]:
```
print('Border Crossing List', len(data_json))
```

```
Border Crossing List 355511
```

[16]:
```
# Connection to Mongo DB
try:
    client = MongoClient('localhost', 27017)
    print ("Connected successfully!!!")
except pymongo.errors.ConnectionFailure as e:
    print ("Could not connect to MongoDB: %s" % e )

else:
    # use database named usgs or create it if not there already
    borderdb = client['borderentry']
    # create collection named borderentries or create it if not there already
    bordercoll = borderdb['borderentries']
    # add all the border entries to the list
    bordercoll.insert_many(data_json)
    print("Added", len(data_json), "to Border Crossing entries collection in␣
 ↪borderentry database")
    # close the database connection
    client.close()
```

```
Connected successfully!!!
Added 355511 to Border Crossing entries collection in borderentry database
```

[20]:
```
#search the first item from the collection
bordercoll.find_one()
```

[20]:
```
{'_id': ObjectId('5f35d5a631efdcbd009a7422'),
 'PortName': 'Alcan',
 'State': 'AK',
 'PortCode': 3104,
 'Border': 'US-Canada Border',
 'Date': 1580515200000,
 'Measure': 'Personal Vehicle Passengers',
 'Value': 1414}
```

[18]:
```
#print the number of docs from db
print('Total Number of Documents: ',bordercoll.count_documents({}))
```

```
Total Number of Documents:  355511
```

[19]:
```
# Aggregation
cursor = bordercoll.aggregate([{"$group":
       {"_id":"$Measure",
       "Crossings":{"$sum":'$Value'}}}])

for document in cursor:
       print(document)
```

```
{'_id': 'Personal Vehicles', 'Crossings': 2651535415}
```

```
{'_id': 'Pedestrians', 'Crossings': 1090067964}
{'_id': 'Truck Containers Full', 'Crossings': 185463194}
{'_id': 'Bus Passengers', 'Crossings': 146027374}
{'_id': 'Buses', 'Crossings': 8754394}
{'_id': 'Trucks', 'Crossings': 264731943}
{'_id': 'Truck Containers Empty', 'Crossings': 67036035}
{'_id': 'Rail Containers Empty', 'Crossings': 22386399}
{'_id': 'Rail Containers Full', 'Crossings': 40492650}
{'_id': 'Train Passengers', 'Crossings': 6472717}
{'_id': 'Personal Vehicle Passengers', 'Crossings': 5629526756}
{'_id': 'Trains', 'Crossings': 933270}
```

[21]:
```
#search the first item from the collection
bordercoll.find_one()
```

[21]:
```
{'_id': ObjectId('5f35d5a631efdcbd009a7422'),
 'PortName': 'Alcan',
 'State': 'AK',
 'PortCode': 3104,
 'Border': 'US-Canada Border',
 'Date': 1580515200000,
 'Measure': 'Personal Vehicle Passengers',
 'Value': 1414}
```

[22]:
```
#list db names
borderdb.list_collection_names()
```

[22]:
```
['borderentries']
```

[23]:
```
#drop db
borderdb.borderentries.drop()
```

[24]:
```
#empty database
borderdb.list_collection_names()
```

[24]:
```
[]
```