Bankit - append di tutti i files csv e trasformazione in un unico parquet

```
import pandas as pd
import os
import glob
import numpy as np
pd.options.display.float_format = "{:,.2f}".format
```

creazione unico parquet

```
date_column = ['DATA_OSS']
df_append = pd.DataFrame() #append all files together

os.chdir('D:/files/csv/Bankit/STAFINRA')
csv_files = glob.glob('*TFR*.{}'.format('csv'))
for file in csv_files:
            df_temp = pd.read_csv( file, encoding='utf-8',sep=';',
                                  dtype={'VALORE':'float', 'SET_CTP':'str','LOC_SPORT':'str'
                                  parse_dates=date_column, dayfirst=False,decimal=",")
            df_temp['tabella'] = file.split('.')[0].split('-')[-1]
            df_temp['fonte'] = "STAFINRA"
            df_append = pd.concat([df_append, df_temp],ignore_index=True)
#df_append = df_append.fillna(value=np.nan)
#df_append = df_append.replace([None], ['VOID'], regex=True)
df_append.to_parquet('D:/files/csv/Bankit/STAFINRA.parquet', engine='fastparquet') # pip inst
df_append.shape

# df = pd.concat([df, pd.DataFrame([new_row])], ignore_index=True)
```

```
(2551968, 26)
```

```
date_column = ['DATA_OSS']
df_append = pd.DataFrame() #append all files together

os.chdir('D:/files/csv/Bankit/STABOL')
csv_files = glob.glob('*TDB*.{}'.format('csv'))
for file in csv_files:
            df_temp = pd.read_csv(file, sep=';',dtype={'VALORE':'float'},parse_dates=date_co
            df_temp['tabella'] = file.split('.')[0].split('-')[-1]
            df_temp['fonte'] = "STABOL"
            df_append = pd.concat([df_append, df_temp],ignore_index=True)
df_append.shape
```

```
(7044671, 36)
```

```
date_column = ['DATA_OSS']
df_append = pd.DataFrame() #append all files together

os.chdir('D:/files/csv/Bankit/STACORIS')
csv_files = glob.glob('*TRI*.{}'.format('csv'))
for file in csv_files:
            df_temp = pd.read_csv(file, sep=';',dtype={'VALORE':'float'},parse_dates=date_co
            df_temp['tabella'] = file.split('.')[0].split('-')[-1]
            df_temp['fonte'] = "STACORIS"
            df_append = pd.concat([df_append, df_temp],ignore_index=True)
df_append.shape
```

```
date_column = ['DATA_OSS']
df_append = pd.DataFrame() #append all files together

os.chdir('D:/files/csv/Bankit/STAMEN')
csv_files = glob.glob('*TDB*.{}'.format('csv'))
for file in csv_files:
            df_temp = pd.read_csv(file, sep=';',dtype={'VALORE':'float'},parse_dates=date_co
            df_temp['tabella'] = file.split('.')[0].split('-')[-1]
            df_temp['fonte'] = "STAMEN"
            df_append = pd.concat([df_append, df_temp],ignore_index=True)
df_append.shape
```

```
date_column = ['DATA_OSS']
df_append = pd.DataFrame() #append all files together
```

```
os.chdir('D:/files/csv/Bankit/STAATER')
csv_files = glob.glob('*TDB*.{}'.format('csv'))
for file in csv_files:
            df_temp = pd.read_csv(file, sep=';',dtype={'VALORE':'float'},parse_dates=date_col
            df_temp['tabella'] = file.split('.')[0].split('-')[-1]
            df_temp['fonte'] = "STAATER"
            df_append = pd.concat([df_append, df_temp],ignore_index=True)
df_append.shape
```

```
df_append[['fonte','tabella','DATA_OSS','VALORE']].sort_values(by='VALORE', ascending=False)
```

```
df_append.to_parquet('D:/Bankit.parquet')
```

creazione DB

```
import pyarrow.parquet as pq # pip install pyarrow
import duckdb
con = duckdb.connect()
con.execute("PRAGMA threads=8") # enable automatic query parallelization
con.execute("PRAGMA enable_object_cache") # enable caching of parquet metadata
```

```
<duckdb.duckdb.DuckDBPyConnection at 0x203b22fd370>
```

```
# tabella dati
conn = duckdb.connect(database='D:/Bankit.duckdb', read_only=False)
```

```
df = (con.execute("SELECT * FROM 'D:/Bankit.parquet' where tabella = 'TFR10194'").df())
## read parquet file
df_append = pd.read_parquet('d:/Bankit.parquet')
df.shape
```

```
(444698, 41)
```

tabelle

```
# crezione tabella completa
conn.execute(f'''CREATE TABLE tabelle AS SELECT * FROM read_parquet('D:/Bankit.parquet');''')
```

```
# creazione tabella dimensioni stafinra
conn.execute(f'''CREATE TABLE IF NOT EXISTS  stafinra (Dominio VARCHAR, Elemento VARCHAR, Des
conn.execute(f'''COPY stafinra FROM 'D:/files/csv/Bankit/stafinra/20240403_125435-DOMAIN-STAF
```

```
# creazione tabella dimensioni stamen
conn.execute(f'''CREATE TABLE IF NOT EXISTS  stamen (Dominio VARCHAR, Elemento VARCHAR, Descr
conn.execute(f'''COPY stamen FROM 'D:/files/csv/Bankit/stamen/DOMAIN-stamen-MULTICUBE.csv' WI
```

	Count
0	211

conn.execute(f’‘’drop table stafinra;’’’).df()

conn.execute(f’‘’show tables;’’’).df()

estrazione TFR10194

```
TFR10194 = conn.execute(f'''Select DATA_OSS,  VALORE from tabelle where tabella = 'TFR10194'
TFR10194.shape
```

```
conn.execute(f'''Select * from stafinra limit 10;''').df()
```

	Dominio	Elemento	Descrizione
0	ATECO	000000	Informazione non prevista o non applicabile
1	ATECO	1000055	Prodotti chimici e farmaceutici
2	ATECO	1000060	Fabbricazione di autoveicoli e altri mezzi di ...
3	ATECO	1000061	Industrie alimentari, delle bevande e del tabacco
4	ATECO	1000062	Industrie tessili, abbigliamento e articoli i...
5	ATECO	1000063	Carta, articoli di carta e prodotti della stampa
6	ATECO	1000065	Attività manifatturiera residuale (divisioni 1...
7	ATECO	1000074	Attività residuali (sezioni O P Q R S T)
8	ATECO	1004999	Totale ateco al netto della sez. U
9	ATECO	1005001	Attività industriali

```
conn.execute(f'''select count(*) from tabelle;''').df()
```

	count_star()
0	17093341

Query

TFR10194 = df_append.query(‘tabella==“TFR10194”’) TFR10194 = TFR10194.dropna(how= “all”, axis=1)

TFR10194.shape

df = conn.execute(f’’’Select tabella,data_oss, ENTE_SEGN, ATECO_CTP, LOC_CTP, set_ctp, valore, b.descrizione NUT,c.descrizione target
from tabelle a left JOIN stafinra b ON a.LOC_CTP = b.Elemento
left JOIN stafinra c ON a.set_CTP = c.Elemento where LOC_CTP = ’IT’ order by VALORE desc;’’’).df()

```
df = conn.execute(f'''Select data_oss, ENTE_SEGN, ATECO_CTP, LOC_CTP, set_ctp, valore, b.desc
from tabelle a left JOIN stafinra b ON a.LOC_CTP = b.Elemento \
left JOIN stafinra c ON a.set_CTP = c.Elemento where tabella = 'TFR20232' and data_oss = '20
df['DATA_OSS'] = pd.to_datetime(df['DATA_OSS'])
df['VALORE'] = df['VALORE'].astype(int)

TFR20232 = df
df.shape
```

```
(1890, 8)
```

```
pd.set_option('display.max_colwidth', 500)
```

```
# TDB20295
df = conn.execute(f'''Select data_oss, ENTE_SEGN, ATECO_CTP, LOC_CTP, set_ctp, valore, b.desc
left JOIN stamen b ON a.LOC_CTP = b.Elemento left JOIN stamen c ON a.set_CTP = c.Elemento whe
and NUT IN ('Roma');''').df()   # and SET_CTP IN ('SBI59','600','','','','','')
df['DATA_OSS'] = pd.to_datetime(df['DATA_OSS'])
df['VALORE'] = df['VALORE'].astype(int)
TDB20295 = df
df.sort_values(by='VALORE', ascending = False)
```

	DATA_OSS	ENTE_SEGN	ATECO_CTP	LOC_CTP	SET_CTP	VALORE	NUT	target
0	2011-12-31	1070001	1005009	ITI43	SBI42	366712344	Roma	Totale res
2	2011-12-31	1070001	1004999	ITI43	SBI25	95685337	Roma	Società n
4	2011-12-31	1070001	1005003	ITI43	SBI25	55692482	Roma	Società n
3	2011-12-31	1070001	1005001	ITI43	SBI25	20472203	Roma	Società n
1	2011-12-31	1070001	F	ITI43	SBI25	18769779	Roma	Società n

```
df.to_excel('D:/butta.xlsx')
```

```
conn.execute(f'''COPY (Select data_oss, ENTE_SEGN, ATECO_CTP, LOC_CTP, set_ctp, cast(valore a
TO 'D:\butta.xlsx' WITH (FORMAT GDAL, DRIVER 'xlsx');''').df()
```

```
CatalogException: Catalog Error: Copy Function with name gdal does not exist!
Did you mean "parquet"?
---------------------------------------------------------------------------
CatalogException                          Traceback (most recent call last)
Cell In[9], line 1
----> 1 conn.execute(f'''COPY (Select data_oss, ENTE_SEGN, ATECO_CTP, LOC_CTP, set_ctp, cast
      2 TO 'D:\butta.xlsx' WITH (FORMAT GDAL, DRIVER 'xlsx');''').df()
CatalogException: Catalog Error: Copy Function with name gdal does not exist!
Did you mean "parquet"?
```