PROJECT: CLEANING BANK MARKETING CAMPAIGN DATA

Personal loans are a lucrative revenue stream for banks. The typical interest rate of a two-year loan in the United Kingdom is **around 10%**. This might not sound like a lot, but in September 2022 alone UK consumers borrowed **around £1.5 billion**, which would mean approximately £300 million in interest generated by banks over two years!

You have been asked to work with a bank to clean the data they collected as part of a recent marketing campaign, which aimed to get customers to take out a personal loan. They plan to conduct more marketing campaigns going forward so would like you to ensure it conforms to the specific structure and data types that they specify so that they can then use the cleaned data you provide to set up a PostgreSQL database, which will store this campaign's data and allow data from future campaigns to be easily imported.

They have supplied you with a csv file called `"bank_marketing.csv"`, which you will need to clean, reformat, and split the data, saving three final csv files. Specifically, the three files should have the names and contents as outlined below:

`client.csv`

column	data type	description	cleaning requirements
`client_id`	`integer`	Client ID	N/A
`age`	`integer`	Client's age in years	N/A
`job`	`object`	Client's type of job	Change `"."` to `"_"`
`marital`	`object`	Client's marital status	N/A
`education`	`object`	Client's level of education	Change `"."` to `"_"` and `"unknown"` to `np.NaN`
`credit_default`	`bool`	Whether the client's credit is in default	Convert to `boolean` data type: `1` if `"yes"`, otherwise `0`

column	data type	description	cleaning requirements
`mortgage`	`bool`	Whether the client has an existing mortgage (housing loan)	Convert to boolean data type: `1` if `"yes"`, otherwise `0`

`campaign.csv`

column	data type	description	cleaning requirements
`client_id`	`integer`	Client ID	N/A
`number_contacts`	`integer`	Number of contact attempts to the client in the current campaign	N/A
`contact_duration`	`integer`	Last contact duration in seconds	N/A
`previous_campaign_contacts`	`integer`	Number of contact attempts to the client in the previous campaign	N/A
`previous_outcome`	`bool`	Outcome of the previous campaign	Convert to boolean data type: `1` if `"success"`, otherwise `0`.
`campaign_outcome`	`bool`	Outcome of the current campaign	Convert to boolean data type: `1` if `"yes"`, otherwise `0`.
`last_contact_date`	`datetime`	Last date the client was contacted	Create from a combination of `day`, `month`, and a newly created `year` column (which should have a value of `2022`); **Format =** `"YYYY-MM-DD"`

`economics.csv`

column	data type	description	cleaning requirements
`client_id`	`integer`	Client ID	N/A
`cons_price_idx`	`float`	Consumer price index (monthly indicator)	N/A
`euribor_three_months`	`float`	Euro Interbank Offered Rate (euribor) three-month rate (daily indicator)	N/A

```
import pandas as pd
import numpy as np
import os
import calendar

# Start coding here...
df = pd.read_csv('bank_marketing.csv')

df['job'] = df['job'].str.replace('.', '_')
df['education'] = df['education'].str.replace('.', '_')
df['education'] = df['education'].replace('unknown', np.NaN)

df['credit_default'] = np.where(df['credit_default'] == 'yes', 1, 0)
df['mortgage'] = np.where(df['mortgage'] == 'yes', 1, 0)

df['previous_outcome'] = np.where(df['previous_outcome'] == 'success', 1, 0)
df['campaign_outcome'] = np.where(df['campaign_outcome'] == 'yes', 1, 0)

df['credit_default'] = df['credit_default'].astype(bool)
df['mortgage'] = df['mortgage'].astype(bool)
df['previous_outcome'] = df['previous_outcome'].astype(bool)
df['campaign_outcome'] = df['campaign_outcome'].astype(bool)

df['month'] = np.where(df['month'] == 'may', 5, 0)
df['day'] = pd.to_numeric(df['day'], errors='coerce')
#df['month'] = pd.to_numeric(df['month'], errors='coerce')
df['year'] = 2022
df['last_contact_date'] = pd.to_datetime(df[['year', 'month', 'day']],
errors='coerce').dt.strftime('%Y-%m-%d')

client_df = df[['client_id', 'age', 'job', 'marital', 'education',
'credit_default', 'mortgage']]
client_df.to_csv('client.csv', index=False)

campaign_df = df[['client_id', 'number_contacts', 'contact_duration',
'previous_campaign_contacts', 'previous_outcome', 'campaign_outcome',
'last_contact_date']]
campaign_df.to_csv('campaign.csv', index=False)

economics_df = df[['client_id', 'cons_price_idx', 'euribor_three_months']]
economics_df.to_csv('economics.csv', index=False)

#col1_type = df['credit_default'].dtype
#print(col1_type)

#display(df)
```

```
df = pd.read_csv("bank_marketing.csv")

for col in ["credit_default", "mortgage", "previous_outcome",
"campaign_outcome"]:
    print(col)
    print("--------------")
    print(df[col].value_counts())
```

```
credit_default
--------------
no         32588
unknown     8597
yes            3
Name: credit_default, dtype: int64
mortgage
--------------
yes        21576
no         18622
unknown      990
Name: mortgage, dtype: int64
previous_outcome
--------------
nonexistent    35563
failure         4252
success         1373
Name: previous_outcome, dtype: int64
campaign_outcome
--------------
no     36548
yes     4640
Name: campaign_outcome, dtype: int64
```