# Bank marketing database

Bank Marketing Campaign Data Cleaning and Storage Project, completed as part of my learning journey on DataCamp. The project involved cleaning and storing data from a bank marketing campaign using pandas, PostgreSQL, and Python, and provided insights into campaign effectiveness and revenue generation. Developed skills in data cleaning and manipulation, SQL database design and management, and Python programming.

***Karim Sherif Taha***

*The objective of this project is to clean and store data from a marketing campaign conducted by a bank in order to extract useful insights and optimize future campaigns. Specifically, the project aims to:*

- *Load the raw data from a CSV file into pandas DataFrames*
- *Clean and reformat the data*
- *Split the data into separate DataFrames that correspond to different tables in a PostgreSQL database schema*
- *Save the cleaned data as CSV files*
- *Generate SQL code to create the corresponding tables in PostgreSQL and populate them with the data*

# *Now, let's do it:*

```
import pandas as pd
import numpy as np
```

## *1. Load the raw data from a CSV file:*

```
bank_markting= pd.read_csv('bank_marketing.csv')
```

## *2. Split the data into three DataFrames based on the desired table structure:*

```
# create a DataFrame for client data
client_df = bank_markting[['client_id', 'age', 'job', 'marital', 'education', 'credit_default', 'housing',
loan'] .rename(columns={'client_id': 'id'})
```

```
# create a DataFrame for campaign data
campaign_df = bank_markting[['client_id', 'campaign', 'month', 'day', 'duration', 'pdays', 'previous',
'poutcome','y']].rename(columns={'previous':'previous_campaign_contacts','duration':'contact_duration',
'y':'campaign_outcome','poutcome':'previous_outcome','campaign':'number_contacts'})
```

```
# create a DataFrame for economics data
 economics_df=bank_markting[['client_id','emp_var_rate','cons_price_idx','euribor3m','nr_employed']]
.rename(columns={'euribor3m':'euribor_three_months','nr_employed':'number_employed'})
```

## 3. Clean and reformat the data according to the instructions provided:

```
client_df["education"] = client_df["education"].str.replace(".","_")
client_df["education"] = client_df["education"].replace("unknown",np.nan)
client_df['job'] = client_df['job'].str.replace('.', '')
campaign_df['campaign_outcome'] = campaign_df['campaign_outcome'].replace({'success': 1,
'failure': 0})
campaign_df['previous_outcome'] = campaign_df['previous_outcome'].replace({'success': 1,
'failure': 0, 'nonexistent': np.nan})
campaign_df = campaign_df.assign(campaign_id=1)
campaign_df['last_contact_date'] = pd.to_datetime('2022-' + campaign_df['month'].astype(str) + '- '
+ campaign_df['day'].astype(str))
data_frames = [economics_df, campaign_df, client_df]
for df in data_frames:
df.drop_duplicates(inplace=True)
```

## 4. Save the cleaned data as CSV files:

```
client_df.to_csv('client.csv', index=False)
campaign_df.to_csv('campaign.csv', index=False)
economics_df.to_csv('economics.csv', index=False)
```

## 5. Generate SQL code to create the corresponding tables in PostgreSQL and populate them with the data:

```
client_table = """CREATE TABLE client
(
id SERIAL PRIMARY KEY,
age INTEGER,
job TEXT,
marital TEXT,
education TEXT,
credit_default BOOLEAN,
housing BOOLEAN,
loan BOOLEAN
);
\copy client from 'client.csv' DELIMITER ',' CSV HEADER
"""

campaign_table = """CREATE TABLE campaign
(
campaign_id SERIAL PRIMARY KEY,
client_id SERIAL references client (id),
number_contacts INTEGER,
contact_duration INTEGER,
pdays INTEGER,
previous_campaign_contacts INTEGER,
previous_outcome BOOLEAN,
last_contact_date DATE
);
\copy campaign from 'campaign.csv' DELIMITER ',' CSV HEADER
"""

economics_table = """CREATE TABLE economics
(
client_id SERIAL references client (id),
emp_var_rate FLOAT,
cons_price_idx FLOAT,
euribor_three_months FLOAT,
number_employed FLOAT
);
\copy economics from 'economics.csv' DELIMITER ',' CSV HEADER
"""
```