```
#pip install fugue[sql]


import os
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from google.colab import drive
#from fugue_sql import fsql
drive.mount('/content/drive')
%matplotlib inline
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Problem Statement 1: Your colleague Jane Hopper, the Business Analyst in charge of analyzing customer behavior, who directly consumes data from the Data Warehouse Environment, needs to get all the account's monthly balances between Jan/2020 and Dec/2020. She wasn't able to do it alone, and asked for your help. Add to your solution the SQL query (.sql file) used to retrieve the data needed (the necessary tables were sent in csv format along with this pdf, on folder tables/). Feel free to use the dialect of your choice, but please specify the SQL engine.

[Valeska] I klnow a method to write sql through pandas using the fuge_sql library and this could one of the alternatives to implement this solution, however as I have more skill with python and due the available time to deliver this test, I implemented the code below

```
accounts= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/accounts/part
transfer_outs= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/transfer
transfer_ins= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/transfer_
pix= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/pix_movements/part
d_time= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/d_time/part-000
```

```
#Selecting relevant column and renaming it to merge with other df
accounts = accounts[['account_id']]
accounts= accounts.rename(columns={'account_id':'id'})
```

```
#Filtering all pix in and status completed
pix_in = pix[(pix['status'] =='completed') & (pix['in_or_out'] =='pix_in')]
#Selecting relevant columns
pix_in = pix_in[['account_id', 'pix_amount', 'pix_completed_at']]
#Converting to numeric
pix_in['pix_completed_at'] = pd.to_numeric(pix_in['pix_completed_at'])
#Merge df with d_time dt to calculate the month_year
pix_in = pd.merge(pix_in, d_time, how='left', left_on='pix_completed_at', right_on='time_i
pix_in = pix_in.rename(columns={'account_id':'id', 'pix_amount':'pix_in', 'action_timestam
#Creating column month_year
```

```
pix_in['pix_in_month_year'] = pd.to_datetime(pix_in['pix_date_in']).dt.to_period('M')
pix_in = pix_in[['id', 'pix_in', 'pix_in_month_year']]
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arrays/datetimes.py:1146: UserWarr
  UserWarning,
```

```
#Filtering all pix out and status completed
pix_out = pix[(pix['status'] =='completed') & (pix['in_or_out'] =='pix_out')]
#Selectin relevant columns
pix_out = pix_out[['account_id', 'pix_amount', 'pix_completed_at']]
#Converting to numeric
pix_out['pix_completed_at'] = pd.to_numeric(pix_out['pix_completed_at'])
#Merge df with d_time df to calculate the month_year
pix_out = pd.merge(pix_out, d_time, how='left', left_on='pix_completed_at', right_on='time
pix_out = pix_out.rename(columns={'account_id':'id', 'pix_amount':'pix_out', 'action_times
#Creating column month_year
pix_out['pix_out_month_year'] = pd.to_datetime(pix_out['pix_date_in']).dt.to_period('M')
pix_out = pix_out[['id', 'pix_out', 'pix_out_month_year']]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
```

```
#Mergint pix with accounts based on account id
pix_in = pd.merge(pix_in, accounts, how='left',on='id')
pix_out = pd.merge(pix_out, accounts, how='left', on='id')
pix = pd.merge(pix_in, pix_out, how='outer',on='id')
#If pix_in_month_year is empty, consider pix_out_month_year
pix['pix_month'] = np.where(pix['pix_in_month_year'].isna(), pix['pix_out_month_year'], pi
pix = pix[['pix_month', 'id', 'pix_in', 'pix_out']]
#Sum pix of the month by account id
pix = pix.groupby(['id','pix_month'])['pix_in','pix_out'].sum().reset_index()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: FutureWarning: Indexi
  if __name__ == '__main__':
```

```
#Filtering transf_outs with status completed and convert to numeric
transfer_outs = transfer_outs[(transfer_outs['status'] =='completed')]
transfer_outs['transaction_completed_at'] = pd.to_numeric(transfer_outs['transaction_compl
#merging with d_time df to calculate the date
transfer_out = pd.merge(transfer_outs, d_time, how='left', left_on='transaction_completed_
transfer_out = transfer_out[['id', 'amount', 'transaction_completed_at', 'action_timestamp
transfer_out = transfer_out.rename(columns={'amount':'transfer_out', 'transaction_complete
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
#Filtering transf_ins with status completed and convert to numeric
transfer_ins = transfer_ins[(transfer_ins['status'] =='completed')]
transfer_ins['transaction_completed_at'] = pd.to_numeric(transfer_ins['transaction_complet
#Merging with d_time df to calculate the date
transfer_ins = pd.merge(transfer_ins, d_time, how='left', left_on='transaction_completed_a
transfer_ins = transfer_ins[['id', 'amount', 'transaction_completed_at', 'action_timestamp
transfer_ins = transfer_ins.rename(columns={'amount':'transfer_in', 'transaction_completed
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
#Merge transfers dfs with accounts based on id
df1 = pd.merge(transfer_ins, accounts, how='left',on='id')
df2 = pd.merge(transfer_out, accounts, how='left', on='id')
#Create month_year columns
df1['in_month_year'] = pd.to_datetime(df1['date_in']).dt.to_period('M')
df2['out_month_year'] = pd.to_datetime(df2['date_out']).dt.to_period('M')
#Select relevant columns
df1 = df1[['in_month_year', 'id', 'transfer_in']]
df2 = df2[['out_month_year', 'id', 'transfer_out']]
#Sum the transfers aggregating by account id
df1 = df1.groupby(['id','in_month_year'])['transfer_in'].sum().reset_index()
df2 = df2.groupby(['id','out_month_year'])['transfer_out'].sum().reset_index()
df = pd.merge(df1, df2, how='outer', on='id')
#Creating month_transf column
df['month_transf'] = np.where(df['in_month_year'].isna(), df['out_month_year'], df['in_mon
df = df[['month_transf', 'id', 'transfer_in', 'transfer_out']]
#Aggregating transfers by month and id
df = df.groupby(['id','month_transf'])['transfer_in','transfer_out'].sum().reset_index()
#Merging with pix df
df = pd.merge(df, pix, how='outer', on='id')
df['month'] = np.where(df['pix_month'].isna(), df['month_transf'], df['pix_month'])
df = df[['id', 'month', 'transfer_in', 'transfer_out', 'pix_in', 'pix_out']]
df['pix_in'] = df['pix_in'].fillna(0)
df['pix_out'] = df['pix_out'].fillna(0)
df['transfer_in'] = df['transfer_in'].fillna(0)
df['transfer_out'] = df['transfer_out'].fillna(0)
#Calculating the account monthly balance with the transfers and pix
df['Account Monthly Balance'] = (df['transfer_in'] + df['pix_in']) - (df['transfer_out']+
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:18: FutureWarning: Index
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:23: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:25: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:26: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:28: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
```

```
df.head()
```

	id	month	transfer_in	transfer_out	pix_in	pix_out	Account Monthly Balance
0	15468786971758	2020-12	1866.63	0.00	0.0	0.0	1866.63
1	21047734532096	2020-11	448.48	0.00	0.0	0.0	448.48
2	21538059756947	2020-01	0.00	1797.07	0.0	0.0	-1797.07

This following csv file is the result for Jane Hopper. It has all the account's monthly balances between Jan/2020 and Dec/2020.

```
df.to_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Problem_Statement_1.csv')
```

*- *Problem Statement 2:* * Now that you were able to work with these tables, you might have noticed that we could improve the data model somehow and want to suggest some changes. You may also consider that Nubank is always evolving with new products and it is also expanding to new countries, so our data warehouse model needs to accommodate all these incoming changes. Keep in mind that the new products sometimes are not related to peer-to-peer transactions – for example: life insurance, lending, rewards and other products – and some of them might be available only in some countries. Knowing all of that, which modifications would you propose to the current data model and why? Remember that other analysts will be using the same structure, so it should be as clear as possible. Feel free to change, remove and/or add tables and fields to generate a better data model design.

[Valeska] Considering the requirement to expand to new countries, there is no need for a change that brings a significant gain in the architecture of the tables, since it already has the Country table as a dimension.

[Valeska] A possible change could be related to investments, as some yields require some follow-up fields that are not mapped in the Investiments table, however, only for the purposes of each client's monthly balance, the Investments table is sufficient to support other products such as life insurance, lending, etc.

*- *Problem Statement 3:* * Since many people are already consuming data from the current model, we need to come up with a migration plan in order to change our data warehouse above with your suggestions. Which strategy would you propose in order to implement those changes?

[Valeska] First, you need to have two environments, one environment being consumed by users and another environment as a replica for development. Any change must be made in the development environment, including the entire ETL process, testing and data validation. Once this change is approved, this development is applied to the production environment. This is a strategy to ensure minimal impact to the systems and users that consume the data.

*- *Problem Statement 4:* * On another note, Jane's friend, Pepino, wants to know how well our PIX product is doing inside Nubank. For that, he wants your help to come up with indicators that can be used to track the technical and business performance of the product. Which metrics would you suggest to track and why?

[Valeska] There are some important KPIs to be monitored and combined can bring valuable insights for the Jane"s business area.

- Quantity of transactions by month
- Quantity of transaction by customer
- Quantity of transactions by customer and by month
- Quantity of transactions by input by customer and by month
- Quantity of transactions by country/state/city
- Quantity of transactions trasações over the days per month
- Quantity of transaction over the time
- Quantity of transaction by status
- Ammount of money transactioned by month
- Ammount of money transactioned by country/state/city
- Ammount of money transactioned by customer
- Correlation between customers that make transfers and customer that makes pix

*- *Problem Statement 5:* *

An example scenario for customer A is given below, starting to invest in this fixed income product on day 16 of month 1. This is the first time this customer is depositing money into this investment, so the previous balance was null. His first deposit was 1000 (one thousand) and at the end of the day this amount has generated a 0.01% income rate to his balance. This customer continues to invest other times throughout the month in this same product. Keep in mind that this is a mock sample of the transaction log with calculations applied on a daily basis. Keep in mind that, in case of negative Movements, the income for that respective day should be set to zero. $Movements = PreviousDayBalance + Deposit - Withdrawal$ $End\ of\ Day\ Income = Movements * Income\ Rate$ $AccountDailyBalance = Movements + EndofDayIncome$ Another business analyst, Sophia, would like your help to analyze how much money Nubank's customers' have on their investment account on a daily basis.

```
accounts= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/accounts/part
transfer_outs= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/transfer
transfer_ins= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/transfer_
pix= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/pix_movements/part
d_time= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Tables/d_time/part-000
investment_accounts_to_send = pd.read_csv('/content/drive/My Drive/Colab Notebooks/Nubank/
```

```
#Filtering the pix
pix_in = pix[(pix['status'] =='completed') & (pix['in_or_out'] =='pix_in')]
pix_in = pix_in[['account_id', 'pix_amount', 'pix_completed_at']]
pix_in['pix_completed_at'] = pd.to_numeric(pix_in['pix_completed_at'])
#Merging with d_time df and creating the pix_in_date column
pix_in = pd.merge(pix_in, d_time, how='left', left_on='pix_completed_at', right_on='time_i
pix_in = pix_in.rename(columns={'account_id':'id', 'pix_amount':'pix_in', 'action_timestam
pix_in['pix_in_date'] = pd.to_datetime(pix_in['pix_date_in']).dt.to_period('D')
pix_in = pix_in[['id', 'pix_in', 'pix_in_date']]
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arrays/datetimes.py:1146: UserWarr
  UserWarning,
```

```
#Filtering the pix
pix_out = pix[(pix['status'] =='completed') & (pix['in_or_out'] =='pix_out')]
pix_out = pix_out[['account_id', 'pix_amount', 'pix_completed_at']]
pix_out['pix_completed_at'] = pd.to_numeric(pix_out['pix_completed_at'])
#Merging with d_time df and creating pix_out_date column
pix_out = pd.merge(pix_out, d_time, how='left', left_on='pix_completed_at', right_on='time
pix_out = pix_out.rename(columns={'account_id':'id', 'pix_amount':'pix_out', 'action_times
pix_out['pix_out_date'] = pd.to_datetime(pix_out['pix_date_in']).dt.to_period('D')
pix_out = pix_out[['id', 'pix_out', 'pix_out_date']]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  after removing the cwd from sys.path.
```

```
#Merging pix with accounts based on id and creating the day column
pix_in = pd.merge(pix_in, accounts, how='left',left_on='id', right_on='account_id')
pix_out = pd.merge(pix_out, accounts, how='left',left_on='id', right_on='account_id')
pix = pd.merge(pix_in, pix_out, how='outer',on='id')
pix['pix_day'] = np.where(pix['pix_in_date'].isna(), pix['pix_out_date'], pix['pix_in_date
pix = pix[['pix_day', 'id', 'pix_in', 'pix_out']]
#Sum of pix grouped by id and day
pix = pix.groupby(['id','pix_day'])['pix_in','pix_out'].sum().reset_index()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: FutureWarning: Indexi
```

```
#Filtering the transfers outs and formating the columns
transfer_outs = transfer_outs[(transfer_outs['status'] =='completed')]
transfer_outs['transaction_completed_at'] = pd.to_numeric(transfer_outs['transaction_compl
transfer_out = pd.merge(transfer_outs, d_time, how='left', left_on='transaction_completed_
transfer_out = transfer_out[['id', 'amount', 'transaction_completed_at', 'action_timestamp
transfer_out = transfer_out.rename(columns={'amount':'transfer_out', 'transaction_complete
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
#Filtering the transfers ins and formating the columns
transfer_ins = transfer_ins[(transfer_ins['status'] =='completed')]
transfer_ins['transaction_completed_at'] = pd.to_numeric(transfer_ins['transaction_complet
transfer_ins = pd.merge(transfer_ins, d_time, how='left', left_on='transaction_completed_a
transfer_ins = transfer_ins[['id', 'amount', 'transaction_completed_at', 'action_timestamp
transfer_ins = transfer_ins.rename(columns={'amount':'transfer_in', 'transaction_completed
```

```
#Merging the transfers and creating the date columns
df1 = pd.merge(transfer_ins, accounts, how='left', left_on='id', right_on='account_id')
df2 = pd.merge(transfer_out, accounts, how='left', left_on='id', right_on='account_id')
df1['in_date'] = pd.to_datetime(df1['date_in']).dt.to_period('D')
df2['out_date'] = pd.to_datetime(df2['date_out']).dt.to_period('D')
df1 = df1[['in_date', 'id', 'transfer_in']]
df2 = df2[['out_date', 'id', 'transfer_out']]
#Sum the transfers by id and date
df1 = df1.groupby(['id','in_date'])['transfer_in'].sum().reset_index()
df2 = df2.groupby(['id','out_date'])['transfer_out'].sum().reset_index()
df = pd.merge(df1, df2, how='outer', on='id')
df['date_transf'] = np.where(df['in_date'].isna(), df['out_date'], df['in_date'])
df = df[['date_transf', 'id', 'transfer_in', 'transfer_out']]
df = df.groupby(['id','date_transf'])['transfer_in','transfer_out'].sum().reset_index()
```

```
#Merging transfers with pix and selecting relevant columns
df = pd.merge(df, pix, how='outer', on='id')
df['date'] = np.where(df['pix_day'].isna(), df['date_transf'], df['pix_day'])
df = df[['id', 'date', 'transfer_in', 'transfer_out', 'pix_in', 'pix_out']]
df['pix_in'] = df['pix_in'].fillna(0)
df['pix_out'] = df['pix_out'].fillna(0)
df['transfer_in'] = df['transfer_in'].fillna(0)
df['transfer_out'] = df['transfer_out'].fillna(0)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: FutureWarning: Inde

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:19: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:20: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:21: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:22: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
```

```
#Merging the investiment with df and creating the calculus to reply this statement
df_invest = pd.merge(investment_accounts_to_send, df, left_on='account_id', right_on='id',
#Sum of all transactions (pix and transfers)
df_invest['Account Daily Balance'] = (df_invest['transfer_in'] + df_invest['pix_in']) - (d
df_invest['last_day'] = df_invest.sort_values(['date']).groupby(df_invest.account_id)['Acc
#Calculate the movements based on the last day balance
df_invest['Movements'] = df_invest['last_day'] + df_invest['Account Daily Balance']
df_invest = df_invest[['account_id', 'date', 'Account Daily Balance', 'Movements']]
df_invest = df_invest.groupby(['account_id','date']).sum().reset_index()
#Calculating the end of day income based on daily balance
df_invest['End of Day Income'] = df_invest['Account Daily Balance'] * 0.01


#Reset the income if there is no positive transactions
df_invest['End of Day Income'] = np.where(df_invest['End of Day Income'] < 0, 0,  df_inve


#Calculate the final result
df_invest['Account Daily Balance'] = df_invest['Account Daily Balance'] + df_invest['End of


df_invest.head()
```

	account_id	date	Account Daily Balance	*Movements*	*End of Day Income*	*Account Daily Balance*
0	752997682765132288	2020-01-14	-5837.37	0.00	0.0000	-5837.3700
1	752997682765132288	2020-01-20	8277.15	2439.78	82.7715	8359.9215

The following csv file has the result for the statement 5

```
df_invest.to_csv('/content/drive/My Drive/Colab Notebooks/Nubank/Problem_Statement_5.csv')
```