3. Atividade extra

O objetivo da atividade extra, é exportar arquivos flat file, para análise de comportamento de compra, inadimplência e churn dos associados.

Importação de bibliotecas

In [14]:
```
import os
import shutil
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_unixtime, col, to_timestamp, coalesce
from pyspark.sql.types import StringType, IntegerType, LongType, DecimalType, DateType
```

Variaveis do projeto

In [15]:
```
#Variaveis de conexao com postgres
v_caminho_jar_postgres='/home/jovyan/work/jars/postgresql-9.4.1207.jar'
v_url_jdbc='jdbc:postgresql://postgres/projeto'
v_user_jdbc='airflow'
v_pass_jdbc='airflow'

#Diretorio de export do arquivo de flatfile
v_diretorio_export='/home/jovyan/work/export'
```

Criando sessao e contexto

In [16]:
```
spark = (SparkSession
         .builder
         .master('local')
         .appName('load-postgres')
         # Add postgres jar
         .config('spark.driver.extraClassPath', v_caminho_jar_postgres)
         .getOrCreate())
sc = spark.sparkContext
```

In [17]:
```
#Funcao para exportar um dataframe para arquivo csv
def f_exporta_flat_file(df, arquivo_csv):
    #Gera arquivo flat
    df.coalesce(1).write.mode('overwrite').options(header='True', delimiter=';').csv(v_diretori

    #Renomeia arquivo e move para a pasta export
    for arquivo in os.listdir(v_diretorio_export + '/flatfile/'):
        a_arquivo=arquivo.split('.')
        if(len(a_arquivo)==2 and a_arquivo[1]=='csv'):
            shutil.move(v_diretorio_export + '/flatfile/' + arquivo, v_diretorio_export + '/' +
            print('Arquivo flat gerado: ' + v_diretorio_export + '/' + arquivo_csv)
```

In [21]:
```
#Carregando dados no Dataframe
df_flat_file = (
    spark.read
    .format('jdbc')
    .option('url', v_url_jdbc)
    .option('query', '''
        select
            ass.id as id_associado,
            ass.nome,
            ass.sobrenome,
            ass.idade,
            ass.email,
            case when compras_inad.data_encerramento_conta is null then 'Sim' else 'Não' end as
            case when coalesce(compras_inad.qtd_dias_atraso_pgto, 0)=0 then 'Não' else 'Sim' er
```

```
            compras_inad.data_primeira_compra,
            compras_inad.data_ultima_compra,
            (compras_inad.data_ultima_compra - compras_inad.data_primeira_compra)/30::integer a
            compras_inad.data_encerramento_conta,
            compras_inad.dia_ult_compra,
            compras_inad.dia_ult_compra/30 as mes_ult_compra,
            trim(replace(compras_inad.vlr_total_compras::varchar(100), '.', ',')) as vlr_total_
            compras_inad.qtd_compras,
            trim(replace(compras_inad.vlr_medio_compras::varchar(100), '.', ',')) as vlr_medio_
            compras_inad.qtd_dias_atraso_pgto

        from target.associado ass

        left join (
            select
                car.id_associado,
                min(mov.data_movimento) as data_primeira_compra,
                max(mov.data_movimento) as data_ultima_compra,
                enc.data_encerramento as data_encerramento_conta,
                coalesce(enc.data_encerramento, current_date) - max(mov.data_movimento) as dia_
                sum(mov.vlr_transacao) as vlr_total_compras,
                count(distinct mov.id) as qtd_compras,
                avg(mov.vlr_transacao)::decimal(10,2) as vlr_medio_compras,
                max(fat.qtd_dias_atraso_pgto) as qtd_dias_atraso_pgto

            from target.movimento mov

            inner join target.cartao car
            on mov.id_cartao=car.id

            inner join target.encerramento_conta enc
            on enc.id=car.id_conta

            inner join (
                select
                    id_cartao,
                    max(qtd_dias_atraso_pgto) as qtd_dias_atraso_pgto
                from target.fatura

                group by 1
            ) fat
            on fat.id_cartao=car.id

            group by car.id_associado, enc.data_encerramento
        ) as compras_inad
        on compras_inad.id_associado=ass.id

        where ass.id>0
    ''')
    .option('user', v_user_jdbc)
    .option('password', v_pass_jdbc)
    .load()
)

#Gerando flat file
f_exporta_flat_file(df_flat_file, 'associado_compras_fatura_inad.csv')
```

```
[Stage 6:>                                                          (0 + 1) / 1]
Arquivo flat gerado: /home/jovyan/work/export/associado_compras_fatura_inad.csv
```

In [22]:

```
#Carregando dados no Dataframe
df_flat_file = (
    spark.read
    .format('jdbc')
    .option('url', v_url_jdbc)
    .option('query', '''
        select
```

```
            ass.id as id_associado,
            ass.nome,
            ass.sobrenome,
            ass.idade,
            ass.email,
            compras_inad.des_transacao,
            trim(replace(compras_inad.vlr_total_compras::varchar(100), '.', ',')) as vlr_total_
            compras_inad.qtd_compras,
            trim(replace(compras_inad.vlr_medio_compras::varchar(100), '.', ',')) as vlr_medio_

        from target.associado ass

        left join (
            select
                car.id_associado,
                mov.des_transacao,
                sum(mov.vlr_transacao) as vlr_total_compras,
                count(distinct mov.id) as qtd_compras,
                avg(mov.vlr_transacao)::decimal(10,2) as vlr_medio_compras
            from target.movimento mov

            inner join target.cartao car
            on mov.id_cartao=car.id

            group by 1, 2
        ) as compras_inad
        on compras_inad.id_associado=ass.id

        where ass.id>0
    ''')
    .option('user', v_user_jdbc)
    .option('password', v_pass_jdbc)
    .load()
)

#Gerando flat file
f_exporta_flat_file(df_flat_file, 'associado_comportamento_compras.csv')
```

```
[Stage 7:>                                                          (0 + 1) / 1]
Arquivo flat gerado: /home/jovyan/work/export/associado_comportamento_compras.csv
```

In [ ]: