In [179]:

```
import os, sys, glob
from pyspark.sql import Window
from pyspark.sql.functions import *
import pandas as pd
import datetime as dt
from dateutil.relativedelta import relativedelta
pd.set_option('max_columns', None)
os.environ['SPARK_HOME']='/opt/mapr/spark/spark'
os.environ['PYSPARK_PYTHON']='/opt/python/python37/bin/python'
spark_python = os.path.join(os.environ.get('SPARK_HOME', None), 'python')
py4j = glob.glob(os.path.join(spark_python, 'lib', 'py4j-*.zip'))[0]
sys.path[:0]=[spark_python, py4j]
os.environ['PYTHONPATH']=py4j
from pyspark.sql import SparkSession
sqlContext = SparkSession \
.builder \
.appName("spark model19") \
.config("spark.executor.memory", "25g") \
.config("spark.master", "yarn") \
.config("spark.submit.deployMode", "client") \
.config("spark.executor.instances","25") \
.config("spark.driver.memory", "25g") \
.config("spark.speculation", "true") \
.config("spark.driver.maxResultSize", "25g") \
.config("spark.sql.parquet.binaryAsString", "true") \
.config("spark.sql.broadcastTimeout", "20000000ms") \
.config("spark.sql.crossJoin.enabled", "true") \
.config("spark.driver.extraJavaOptions", "-XX:MaxDirectMemorySize=99999998m -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=30 -XX:+ScavengeBeforeFullGC -XX:+CMSScavengeBeforeRemark") \
.config("spark.executor.extraJavaOptions", "-XX:MaxDirectMemorySize=99999998m -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=30 -XX:+ScavengeBeforeFullGC -XX:+CMSScavengeBeforeRemark") \
.config("spark.yarn.dist.files", "/opt/mapr/hive/hive/conf/hive-site.xml") \
.config("spark.sql.catalogImplementation", "hive") \
.config("spark.jars", "/axp/platform/cloak/lib/cloak-spark.jar,/axp/platform/mlplat/app/spy/hive-contrib.jar") \
.config("spark.sql.legacy.allowCreatingManagedTableUsingNonemptyLocation","true") \
.config("spark.yarn.archive","/axp/platform/cloak/lib/spark-jars.zip") \
.config("spark.driver.extraJavaOptions=-Dlog4j.configuration","/var/tmp/log4j.properties") \
.config("spark.driver.extraJavaOptions=-Dhadoop.login","hybrid") \
.config("spark.driver.extraJavaOptions=-Dhadoop.login","hybrid") \
.config("spark.sql.parquet.writeLegacyFormat","true") \
.config("spark.driver.allowMultipleContexts", "true") \
.config("spark.sql.parquet.enableVectorizedReader","false") \
.config("spark.dynamicAllocation.enabled","false") \
.config("spark.shuffle.service.enabled","false") \
.config("num-executors","800") \
.config("spark.shuffle.compress","true") \
.config("spark.shuffle.spill.compress","true") \
.config("spark.io.compression.codec","zstd") \
.config("spark.io.compression.zstd.level","6") \
.config("spark.sql.shuffle.partitions","1200") \
.config("SQLContext.sql.execution.arrow.enabled", "true") \
.config("hive.server2.enable.doAs=false") \
.config("spark.rpc.message.maxSize", "512") \
.config("spark.sql.legacy.replaceDatabricksSparkAvro.enabled", "true") \
.config("spark.sql.avro.compression.codec", "snappy") \
.config("spark.sql.avro.deflate.level", "-1") \
.config("spark.sql.orc.impl", "native") \
.config("spark.sql.orc.enableVectorizedReader", "true") \
.enableHiveSupport() \
.config("spark.sql.parquet.binaryAsString", "FALSE") \
.config("spark.sql.parquet.int96AsTimestamp", "TRUE") \
.config("spark.sql.parquet.compression.codec", "snappy") \
.config("spark.sql.parquet.filterPushdown", "TRUE") \
.config("spark.sql.hive.convertMetastoreParquet", "TRUE") \
.config("spark.sql.parquet.mergeSchema", "FALSE") \
.config("spark.sql.parquet.writeLegacyFormat", "FALSE") \
.getOrCreate()
sc = sqlContext.sparkContext
print(sc)
print(sqlContext)
```

```
<SparkContext master=yarn appName=spark model19>
<pyspark.sql.session.SparkSession object at 0x7f462dd5ea20>
```

In [180]:

```
sqlContext.sql("""CREATE DATABASE IF NOT EXISTS AWB19 location 'maprfs:/axp/gms/ramptnt/de
v/cbhan3/AWB19.db' """)
```

Out[180]:

```
DataFrame[]
```

In [181]:

```
def table_isExist_drop(database,table):
    if sqlContext.catalog._jcatalog.tableExists(f"{database}.{table}"):
        sqlContext.sql(f"""drop table {database}.{table}""")
```

In [182]:

```
def customUnion(df1, df2):
    cols1 = [x.lower() for x in df1.columns]
    cols2 = [x.lower() for x in df2.columns]
    total_cols = sorted(cols1 + list(set(cols2) - set(cols1)))
    def expr(mycols, allcols):
        def processCols(colname):
            if colname in mycols:
                return colname
            else:
                return lit(None).alias(colname)
        cols = map(processCols, allcols)
        return list(cols)
    appended = df1.select(expr(cols1, total_cols)).union(df2.select(expr(cols2, total_cols
)))
    return appended
```

In [183]:

```
def drop_duplicate_columns(df):
    cols_new = []
    seen = set()
    dict1 = {}
    for c in df.columns:
        c = c.lower()
        if c in dict1.keys():
            cols_new.append(f'{c}_{dict1[c]}')
            dict1[c]= dict1[c]+1
        else:
            cols_new.append(c)
            dict1[c]=1
    df1 = df.toDF(*cols_new)
    print(cols_new)
    for i in dict1:
        i = i.lower()
        if dict1[i.lower()]>1:
            for x in range(1,dict1[i.lower()]):
                str1 = ""
                str1 = str1+","+(f"{i.lower()}_{x}")
            print(str1)
            str1 = f"COALESCE({i.lower()}{str1}) as {i.lower()}"
            df1 = df1.withColumn(i.lower(),expr(str1))
    column_lower = [i for i in df.columns]
    column_lower = list(map(lambda x: x.lower(), column_lower))
    return df1.select(*[c for c in set(column_lower)])
```

In [184]:

```
r3_month = (dt.date.today() + relativedelta(months=-3)).strftime("%Y%m")
r6_month = (dt.date.today() + relativedelta(months=-6)).strftime("%Y%m")
print(r3_month, r6_month)
```

```
202207 202204
```

In [185]:

```
ACPT_STA_UPDT_DT = str(dt.date.today().replace(day=1) + relativedelta(months=-2,days=-1))
cur_month = (dt.date.today() + relativedelta(months=-1)).strftime("%Y%m")
print(ACPT_STA_UPDT_DT, cur_month)
```

```
2022-07-31 202209
```

In [186]:

```
GLBL_MER_DIM=sqlContext.sql(""" Select mer_id, mer_no, MER_SRCE_SYS_CD, MER_DBA_NM, MER_LGL_NM, MER_HIER_LVL_NO, SEIMS_ACCT_CLASS_4_CD, SEIMS_ACCT_CLASS_6_CD, SEIMS_ACCT_CLASS_8_CD,
MER_CTY_CD, DMA_CD, PHYS_AD_PSTL_AREA_CD, seims_inus_cd, MER_SETUP_DT, MER_ORIG_CD, BRND_TOC_MER_IN, MER_STA_CD, MER_ACT_DT, MER_CANC_RSN_CD, MER_CANC_DT, PHONE_NO, CREAT_TS, PHYS_AD_LINE_1_TX, PHYS_AD_LINE_2_TX, PHYS_AD_LINE_3_TX, PRIM_CLNT_MGR_FULL_NM, PRIM_CLNT_MGMT_DIV_NM, PRIM_LGCY_SALE_SRVC_ID, MER_REIN_DT, MER_REIN_RSN_CD, SRVC_BUS_CTR_CD, SRVC_BUS_CTR_NM, SRVC_SALE_PERS_ID, PHYS_AD_POST_TOWN_NM, PHYS_AD_RGN_AREA_CD, SCF_CD, SMSA_CD, MCC_INUS_CD, BRND_TOC_MER_ID, BRND_TOC_MER_NO, BRND_TOC_MER_SRCE_SYS_CD, BASE_DISC_RT,
CNTRCT_DISC_RT, SUBM_CPBL_TYPE_1_IN, srvc_prvd_cd, mer_tenure_cd from cstonedb3.gmar_glbl_mer_dim
                        where MER_CTY_CD in ('840','850','630') """)
```

In [187]:

```
table_isExist_drop("AWB19","crmd_ind_base")
crmd_ind_base=sqlContext.sql("""
create table awb19.crmd_ind_base as
  select trim(se10) as SE_NO ,
         trim(se_crmd_prim_level3_ctgy_cd) as crmc ,
         trim(se_crmd_prim_vrfy_indus_cd) as sic8 ,
                 trim(se_crmd_mer_onln_offln_cd) as MER_ONLN_OFFLN_CD
  from cstonedb3.GMS_MERCHANT_CRMD_ANALYTICAL""")
```

In [188]:

```
crt_postal_geo_summary=sqlContext.sql("""select
  trim(post_cd) as post_cd,
  trim(fk_mkt_areamkt_are) as DMA_MKT_AREA_ID
  from cstonedb3.crt_postal_geo_summary where trim(fk_regionfk_ctryct)='840'""")
```

In [189]:

```
table_isExist_drop("AWB19","dma_mapping")
```

In [190]:

```
sqlContext.sql("""create table AWB19.dma_mapping as select distinct dma_ds as DMA_DS_TX, dma_cd as DMA_MKT_AREA_ID
from cstonedb3.gmar_glbl_mer_info where trim(mer_ctry_cd)='840'
and trim(dma_ds) != '' and trim(dma_cd) != '' """)
```

In [191]:

```
GLBL_MER_DIM.createOrReplaceTempView("GLBL_MER_DIM")
```

In [192]:

```
all_mer_temp1 = sqlContext.sql("""
select  a.*, b.MKT_TOC_MER_NO as managed_toc ,b.GCG_IN, b.PRIM_CLNT_MGMT_DIV_NM
from GLBL_MER_DIM as a
left outer join awb19.gmar_glbl_mer_info as b on
a.mer_id = b.mer_id
where a.MER_CTY_CD in ('840','850','630')
and (a.MER_STA_CD in ("A","R") or (a.MER_STA_CD="C" and YEAR(a.MER_CANC_DT)>=2011))
and REPLACE(a.mer_no, ' ', '') > '' and REPLACE(a.mer_no, ' ', '') <> '0' """)
```

In [193]:

```
all_mer_temp1 = all_mer_temp1.orderBy(col("MER_SRCE_SYS_CD").desc())
```

In [194]:

```
all_mer_temp1 = all_mer_temp1.orderBy(col("mer_no")).dropDuplicates(["mer_no"])
```

In [196]:

```
all_mer_temp1 = drop_duplicate_columns(all_mer_temp1)
```

```
['mer_id', 'mer_no', 'mer_srce_sys_cd', 'mer_dba_nm', 'mer_lgl_nm', 'mer_hier
_lvl_no', 'seims_acct_class_4_cd', 'seims_acct_class_6_cd', 'seims_acct_class
_8_cd', 'mer_cty_cd', 'dma_cd', 'phys_ad_pstl_area_cd', 'seims_inus_cd', 'mer
_setup_dt', 'mer_orig_cd', 'brnd_toc_mer_in', 'mer_sta_cd', 'mer_act_dt', 'me
r_canc_rsn_cd', 'mer_canc_dt', 'phone_no', 'creat_ts', 'phys_ad_line_1_tx',
'phys_ad_line_2_tx', 'phys_ad_line_3_tx', 'prim_clnt_mgr_full_nm', 'prim_clnt
_mgmt_div_nm', 'prim_lgcy_sale_srvc_id', 'mer_rein_dt', 'mer_rein_rsn_cd', 's
rvc_bus_ctr_cd', 'srvc_bus_ctr_nm', 'srvc_sale_pers_id', 'phys_ad_post_town_n
m', 'phys_ad_rgn_area_cd', 'scf_cd', 'smsa_cd', 'mcc_inus_cd', 'brnd_toc_mer_
id', 'brnd_toc_mer_no', 'brnd_toc_mer_srce_sys_cd', 'base_disc_rt', 'cntrct_d
isc_rt', 'subm_cpbl_type_1_in', 'srvc_prvd_cd', 'mer_tenure_cd', 'managed_to
c', 'gcg_in', 'prim_clnt_mgmt_div_nm_1']
,prim_clnt_mgmt_div_nm_1
```

In [197]:

```
all_mer_temp1.createOrReplaceTempView("all_mer_temp1")
all_mer_temp2 = sqlContext.sql("""
select case when b.mer_id is not null then b.MER_ID else 0 end as mer_id,
        case when b.mer_no is not null then b.mer_no end as se_no,
        case when b.mer_no is null then 0
        end as MMDB_MATCH,
        b.MER_SRCE_SYS_CD as MER_SRCE_SYS_CD,
        b.MER_DBA_NM as DBA_NM,
        b.MER_LGL_NM as Legal_NM,
        case when b.MER_HIER_LVL_NO=2 then "S"
             when b.MER_HIER_LVL_NO=6 then "C"
             when b.MER_HIER_LVL_NO=8 then "M"
             end as mer_hier_lvl_no,
        b.SEIMS_ACCT_CLASS_4_CD as ACCT_class_4_CD,
        b.SEIMS_ACCT_CLASS_6_CD as ACCT_class_6_CD,
        b.SEIMS_ACCT_CLASS_8_CD as ACCT_class_8_CD,
        b.MER_CTY_CD as PHYS_AD_CTRY_CD,
        b.DMA_CD as DMA_MKT_AREA_ID,
        b.PHYS_AD_PSTL_AREA_CD as PHYS_AD_POST_CD_TX,
        b.SEIMS_INUS_CD as SEIMS_INDUS_DS_CD,
        b.MER_SETUP_DT as MER_SETUP_DT,
        b.MER_ORIG_CD as MER_ACCT_ORIG_CD,
        b.BRND_TOC_MER_IN as TOP_OF_CHAIN_IN,
        b.MER_STA_CD as CUR_MER_STA_CD,

        b.mer_tenure_cd as MER_TENURE_MO_CT,

        b.MER_ACT_DT as MER_ACT_DT,
        b.MER_CANC_RSN_CD as CUR_MER_CANC_CD,
        b.MER_CANC_DT as CUR_MER_CANC_DT,
        b.PHONE_NO as PHONE_NO,

        b.CREAT_TS,

        b.PHYS_AD_LINE_1_TX,
        b.PHYS_AD_LINE_2_TX,
        b.PHYS_AD_LINE_3_TX,
        b.PRIM_CLNT_MGR_FULL_NM,
        b.PRIM_CLNT_MGMT_DIV_NM,
        PRIM_LGCY_SALE_SRVC_ID,
        b.MER_REIN_DT,
        b.MER_REIN_RSN_CD,
        b.SRVC_BUS_CTR_CD,
        b.SRVC_BUS_CTR_NM,
        b.SRVC_SALE_PERS_ID,
        b.PHYS_AD_POST_TOWN_NM as city_nm,
        b.PHYS_AD_RGN_AREA_CD,
        b.SCF_CD,
        b.SMSA_CD,
        b.MCC_inus_CD,
        b.BRND_TOC_MER_ID,
        b.BRND_TOC_MER_NO,
        b.BRND_TOC_MER_SRCE_SYS_CD,
        b.BASE_DISC_RT,
        b.CNTRCT_DISC_RT,
        b.SUBM_CPBL_TYPE_1_IN,
        b.managed_toc,
        b.GCG_IN ,
        b.srvc_prvd_cd
from    ALL_MER_TEMP1 as b """)
```

In [198]:

```
from pyspark.sql.functions import to_date
all_mer_temp2 = all_mer_temp2.withColumn('CUR_MER_CANC_DT',to_date(all_mer_temp2.CUR_MER_C
ANC_DT, 'yyyy-MM-dd')) \
                             .withColumn('MER_ACT_DT',to_date(all_mer_temp2.MER_ACT_DT, 'y
yyy-MM-dd'))
```

In [199]:

```
all_mer_temp2.createOrReplaceTempView("all_mer_temp2")
```

In [200]:

```
temp1 = sqlContext.sql(""" select se_no, count(*) from all_mer_temp2 group by 1""")
```

In [201]:

```
temp1.createOrReplaceTempView("temp1")
```

In [202]:

```
temp3 = sqlContext.sql("select BRND_TOC_MER_NO as aff_toc, count(*) from all_mer_temp2 gro
up by 1")
```

In [203]:

```
temp3.createOrReplaceTempView("temp3")
```

In [204]:

```
Missing_TOCs=sqlContext.sql("""select distinct aff_toc from temp3 a
     left join temp1 b on a.aff_toc=b.se_no
where b.se_no is null """)
```

In [205]:

```
Missing_TOCs.createOrReplaceTempView("Missing_TOCs")
```

In [206]:

```
GMAR_MISSING_TOCs= sqlContext.sql("""select  b.mer_no as se_no,
        b.mer_id,
        4 as MMDB_MATCH,
        b.MER_SRCE_SYS_CD,
        b.MER_DBA_NM as DBA_NM,
        b.MER_LGL_NM as Legal_NM,
        Case when b.MER_HIER_LVL_NO=2 then "S"
            when b.MER_HIER_LVL_NO=6 then "C"
            when b.MER_HIER_LVL_NO=8 then "M"
            else cast(b.MER_HIER_LVL_NO as string)
        end as MER_HIER_LVL_NO  ,
        b.SEIMS_ACCT_CLASS_4_CD as ACCT_class_4_CD,
        b.SEIMS_ACCT_CLASS_6_CD as ACCT_class_6_CD,
        b.SEIMS_ACCT_CLASS_8_CD as ACCT_class_8_CD,
        b.MER_CTY_CD as PHYS_AD_CTRY_CD,
        b.DMA_CD as DMA_MKT_AREA_ID,
        b.PHYS_AD_PSTL_AREA_CD as PHYS_AD_POST_CD_TX,
        b.SEIMS_INUS_CD as SEIMS_INDUS_DS_CD,
        b.MER_SETUP_DT as MER_SETUP_DT,
        b.MER_ORIG_CD as MER_ACCT_ORIG_CD,
        b.BRND_TOC_MER_IN as TOP_OF_CHAIN_IN,
        b.MER_STA_CD as CUR_MER_STA_CD,
        b.MER_ACT_DT as MER_ACT_DT,
        b.MER_CANC_RSN_CD as CUR_MER_CANC_CD,
        b.MER_CANC_DT as CUR_MER_CANC_DT,
        b.PHONE_NO,
        b.CREAT_TS,
        b.PHYS_AD_LINE_1_TX,
        b.PHYS_AD_LINE_2_TX,
        b.PHYS_AD_LINE_3_TX,
        b.PRIM_CLNT_MGR_FULL_NM,
        C.PRIM_CLNT_MGMT_DIV_NM,
        b.MER_REIN_DT,
        b.MER_REIN_RSN_CD,
        b.SRVC_BUS_CTR_CD,
        b.SRVC_BUS_CTR_NM,
        b.SRVC_SALE_PERS_ID,
        b.PHYS_AD_RGN_AREA_CD,
        b.SCF_CD,
        b.SMSA_CD,
        b.MCC_INUS_CD,
        b.BRND_TOC_MER_ID,
        b.BRND_TOC_MER_NO,
        b.BRND_TOC_MER_SRCE_SYS_CD,
        0 as DISC_FULL_RT,
        b.BASE_DISC_RT,
        b.CNTRCT_DISC_RT,
        c.MKT_TOC_MER_NO as managed_toc,
        c.GCG_IN ,
        b.srvc_prvd_cd
from    GLBL_MER_DIM as b
left outer join cstonedb3.gmar_glbl_mer_info as c
on b.mer_id=c.mer_id
where   b.mer_no in (select * from Missing_TOCs where replace(aff_toc,' ','') > '')
        and b.MER_CTY_CD in ('840','850','630') """)
```

In [207]:

```
GMAR_MISSING_TOCs.createOrReplaceTempView("GMAR_MISSING_TOCs")
```

In [208]:

```
GMAR_MISSING_TOCs = GMAR_MISSING_TOCs.sort(col("se_no").asc(), col("MER_SRCE_SYS_CD").desc())
```

In [209]:

```
GMAR_MISSING_TOCs = GMAR_MISSING_TOCs.dropDuplicates(["se_no"])
```

In [210]:

```
# GMAR_MISSING_TOCs.show()
```

In [211]:

```
ALL_MER_TEMP = all_mer_temp2.join(GMAR_MISSING_TOCs,["se_no"],"fullouter")
```

In [212]:

```
ALL_MER_TEMP.createOrReplaceTempView("ALL_MER_TEMP")
```

In [213]:

```
prev_month = dt.date.today().replace(day=1) + relativedelta(months=0,days=-1)
```

In [214]:

```
# ALL_MER_TEMP.show()
```

In [215]:

```
ALL_MER_TEMP = drop_duplicate_columns(ALL_MER_TEMP)
```

```
['se_no', 'mer_id', 'mmdb_match', 'mer_srce_sys_cd', 'dba_nm', 'legal_nm', 'm
er_hier_lvl_no', 'acct_class_4_cd', 'acct_class_6_cd', 'acct_class_8_cd', 'ph
ys_ad_ctry_cd', 'dma_mkt_area_id', 'phys_ad_post_cd_tx', 'seims_indus_ds_cd',
'mer_setup_dt', 'mer_acct_orig_cd', 'top_of_chain_in', 'cur_mer_sta_cd', 'mer
_tenure_mo_ct', 'mer_act_dt', 'cur_mer_canc_cd', 'cur_mer_canc_dt', 'phone_n
o', 'creat_ts', 'phys_ad_line_1_tx', 'phys_ad_line_2_tx', 'phys_ad_line_3_t
x', 'prim_clnt_mgr_full_nm', 'prim_clnt_mgmt_div_nm', 'prim_lgcy_sale_srvc_i
d', 'mer_rein_dt', 'mer_rein_rsn_cd', 'srvc_bus_ctr_cd', 'srvc_bus_ctr_nm',
'srvc_sale_pers_id', 'city_nm', 'phys_ad_rgn_area_cd', 'scf_cd', 'smsa_cd',
'mcc_inus_cd', 'brnd_toc_mer_id', 'brnd_toc_mer_no', 'brnd_toc_mer_srce_sys_c
d', 'base_disc_rt', 'cntrct_disc_rt', 'subm_cpbl_type_1_in', 'managed_toc',
'gcg_in', 'srvc_prvd_cd', 'mer_id_1', 'mmdb_match_1', 'mer_srce_sys_cd_1', 'd
ba_nm_1', 'legal_nm_1', 'mer_hier_lvl_no_1', 'acct_class_4_cd_1', 'acct_class
_6_cd_1', 'acct_class_8_cd_1', 'phys_ad_ctry_cd_1', 'dma_mkt_area_id_1', 'phy
s_ad_post_cd_tx_1', 'seims_indus_ds_cd_1', 'mer_setup_dt_1', 'mer_acct_orig_c
d_1', 'top_of_chain_in_1', 'cur_mer_sta_cd_1', 'mer_act_dt_1', 'cur_mer_canc_
cd_1', 'cur_mer_canc_dt_1', 'phone_no_1', 'creat_ts_1', 'phys_ad_line_1_tx_
1', 'phys_ad_line_2_tx_1', 'phys_ad_line_3_tx_1', 'prim_clnt_mgr_full_nm_1',
'prim_clnt_mgmt_div_nm_1', 'mer_rein_dt_1', 'mer_rein_rsn_cd_1', 'srvc_bus_ct
r_cd_1', 'srvc_bus_ctr_nm_1', 'srvc_sale_pers_id_1', 'phys_ad_rgn_area_cd_1',
'scf_cd_1', 'smsa_cd_1', 'mcc_inus_cd_1', 'brnd_toc_mer_id_1', 'brnd_toc_mer_
no_1', 'brnd_toc_mer_srce_sys_cd_1', 'disc_full_rt', 'base_disc_rt_1', 'cntrc
t_disc_rt_1', 'managed_toc_1', 'gcg_in_1', 'srvc_prvd_cd_1']
,mer_id_1
,mmdb_match_1
,mer_srce_sys_cd_1
,dba_nm_1
,legal_nm_1
,mer_hier_lvl_no_1
,acct_class_4_cd_1
,acct_class_6_cd_1
,acct_class_8_cd_1
,phys_ad_ctry_cd_1
,dma_mkt_area_id_1
,phys_ad_post_cd_tx_1
,seims_indus_ds_cd_1
,mer_setup_dt_1
,mer_acct_orig_cd_1
,top_of_chain_in_1
,cur_mer_sta_cd_1
,mer_act_dt_1
,cur_mer_canc_cd_1
,cur_mer_canc_dt_1
,phone_no_1
,creat_ts_1
,phys_ad_line_1_tx_1
,phys_ad_line_2_tx_1
,phys_ad_line_3_tx_1
,prim_clnt_mgr_full_nm_1
,prim_clnt_mgmt_div_nm_1
,mer_rein_dt_1
,mer_rein_rsn_cd_1
,srvc_bus_ctr_cd_1
,srvc_bus_ctr_nm_1
,srvc_sale_pers_id_1
,phys_ad_rgn_area_cd_1
,scf_cd_1
```

```
,smsa_cd_1
,mcc_inus_cd_1
,brnd_toc_mer_id_1
,brnd_toc_mer_no_1
,brnd_toc_mer_srce_sys_cd_1
,base_disc_rt_1
,cntrct_disc_rt_1
,managed_toc_1
,gcg_in_1
,srvc_prvd_cd_1
```

In [216]:

```
ALL_MER_TEMP.createOrReplaceTempView("ALL_MER_TEMP")
```

In [217]:

```
all_mer_out = sqlContext.sql("""
select
a.*,
month(a.CUR_MER_CANC_DT) as CANCEL_MONTH,
year(a.CUR_MER_CANC_DT) as CANCEL_YEAR,
month(a.MER_SETUP_DT) as OPEN_MONTH,
year(a.MER_SETUP_DT) as OPEN_YEAR,
(case when a.CUR_MER_STA_CD ='A' then 1
when a.CUR_MER_STA_CD ='R' and a.MER_REIN_DT<='{}' then 1
when a.CUR_MER_STA_CD ='C' and (a.CUR_MER_CANC_DT)>'{}' then 1
else 0 end) as active_ind,
(case when a.MER_TENURE_MO_CT >= 0 and a.MER_TENURE_MO_CT <=11 then "<1yr"
when a.MER_TENURE_MO_CT >= 12 and a.MER_TENURE_MO_CT <=23 then "1-2yrs"
when a.MER_TENURE_MO_CT >= 24 and a.MER_TENURE_MO_CT <=35 then "2-3yrs"
when a.MER_TENURE_MO_CT >= 36 then "3yrs+"
end) as mer_tenure_per,
case when a.SEIMS_INDUS_DS_CD in ('591' , '682' , '900' , '654' , '665' ,
'666' , '668' , '669' , '672' , '673' , '674' , '678' ,
'680' , '681' , '683' , '699' , '710' , '725' , '922' ,
'927' , '931' , '932' , '946' , '646' ,
'648' , '687' , '696' , '865' , '872' , '880' ,
'630' , '644' , '645' , '647' , '649' , '659' ,
'660' , '661' , '689' , '694' , '866' , '656' ,
'657' , '658' , '688' , '869' , '916' , '947' ,
'677' , '945' , '971' , '605' , '608' , '642' ,
'664' , '667' , '684' , '735' , '911' , 'A01' ,
'640' , '641' , '653' , '873' , '877' , '349' ,
'486' , '521' , '662' , '663' , '686' , '690' ,
'691' , '692' , '693' , '695' , '861' , '862' ,
'863' , '864' , '867' , '868' , '870' , '871' ,
'875' , '876' , '878' , '879' , '881' , '883' , '948' )
then 1 else 0 end as B2B_ind_old,
case when a.SEIMS_INDUS_DS_CD in ('644','645','648','649','659','660','661',
'687','689','694','880','688','947')
then 1 else 0 end as B2B_ind,
'PROP' as portfolio_type,
(case
when GCG_IN ='Y' then "GCG"
when  PRIM_CLNT_MGMT_DIV_NM like '%GCG%' then "GCG"
when  PRIM_CLNT_MGMT_DIV_NM like '%US NATIONAL%' then "NCG"
when  PRIM_CLNT_MGMT_DIV_NM like '%US REGIONAL%' then "RCG"
when  PRIM_CLNT_MGMT_DIV_NM like ('%US SMALL MERCHANTS CCLM%') then "CCLM"
when  PRIM_CLNT_MGMT_DIV_NM like ('%US SMALL MERCHANTS AGGREGATORS%') then "Small Merchants Aggregators"
when  replace(ACCT_class_8_CD, ' ','') >' ' and substr(ACCT_class_8_CD,1,2) <> '10' then "OTH_SM AMEX Internal"
when  replace(MER_SRCE_SYS_CD, ' ', '') <> "USD" or PHYS_AD_CTRY_CD not in ('001','840') then "OTH_SM INTL"
else "OTH_SM"
 end ) as partner_typ,
(CASE
WHEN REPLACE(a.BRND_TOC_MER_NO, ' ', '')<'' or a.BRND_TOC_MER_NO is null then a.se_no
else a.BRND_TOC_MER_NO end ) as AFF_TOC
from ALL_MER_TEMP as a """.format(str(prev_month),str(prev_month)))
```

In [218]:

```
all_mer_out = drop_duplicate_columns(all_mer_out)
```

```
['mcc_inus_cd', 'phys_ad_line_1_tx', 'phys_ad_ctry_cd', 'prim_lgcy_sale_srvc_
id', 'srvc_sale_pers_id', 'phys_ad_rgn_area_cd', 'dba_nm', 'cur_mer_sta_cd',
'subm_cpbl_type_1_in', 'phys_ad_line_3_tx', 'srvc_bus_ctr_cd', 'mer_act_dt',
'city_nm', 'mer_tenure_mo_ct', 'base_disc_rt', 'phys_ad_line_2_tx', 'mmdb_mat
ch', 'acct_class_8_cd', 'prim_clnt_mgmt_div_nm', 'mer_hier_lvl_no', 'acct_cla
ss_6_cd', 'mer_id', 'legal_nm', 'dma_mkt_area_id', 'phys_ad_post_cd_tx', 'acc
t_class_4_cd', 'srvc_prvd_cd', 'cur_mer_canc_cd', 'mer_rein_rsn_cd', 'brnd_to
c_mer_no', 'managed_toc', 'cntrct_disc_rt', 'smsa_cd', 'brnd_toc_mer_srce_sys
_cd', 'brnd_toc_mer_id', 'top_of_chain_in', 'se_no', 'mer_setup_dt', 'seims_i
ndus_ds_cd', 'srvc_bus_ctr_nm', 'cur_mer_canc_dt', 'mer_srce_sys_cd', 'mer_ac
ct_orig_cd', 'gcg_in', 'creat_ts', 'disc_full_rt', 'mer_rein_dt', 'scf_cd',
'prim_clnt_mgr_full_nm', 'phone_no', 'cancel_month', 'cancel_year', 'open_mon
th', 'open_year', 'active_ind', 'mer_tenure_per', 'b2b_ind_old', 'b2b_ind',
'portfolio_type', 'partner_typ', 'aff_toc']
```

In [219]:

```
# email_temp.createOrReplaceTempView("email_temp")
all_mer_out.createOrReplaceTempView("all_mer_out")
```

In [220]:

```
# ##confirm from krishna for tpa_se_noall
all_mer_out = sqlContext.sql("""
select b.*,a.*,
row_number() over (order by "se_no") as index
from all_mer_out a left join AWB19.seller_hier_tpa b
on a.managed_toc = b.tpa_se_no where ob_consider_ind = 5  """)
```

In [221]:

```
all_mer_out.createOrReplaceTempView("all_mer_out")
```

In [222]:

```
table_isExist_drop("AWB19", "MER_GLBL_SUBM_MO_F_ALL")
MER_GLBL_SUBM_MO_F_ALL=sqlContext.sql("""create table AWB19.MER_GLBL_SUBM_MO_F_ALL as SELE
CT mer_dim_id,
mer_id,
idy_cl_cd_genesis,
iso_alpha_cd,
mo_id,
adj_lcl_am ,
adj_usd_am,
cr_lcl_am,
cr_usd_am,
soc_lcl_vol,
soc_usd_vol,
db_usd_am,
db_lcl_am,
db_roc_lcl_am,
db_roc_usd_am,
dcn_rev_lcl_am,
dcn_rev_usd_am,
div_lcl_am,
div_usd_am,
take_one_cmsn_lcl_am,
take_one_cmsn_usd_am,
tot_bsn_usd_vol,
dcn_rev_ct,
db_ct,
cr_ct,
adj_ct,
net_roc_ct,
db_soc_ct,
db_roc_ct,
cr_roc_ct,
cr_soc_ct,
tot_soc_ct,
src_stm_id FROM cstonedb3.gmar_mer_glbl_subm_mo_fact""")
```

In [223]:

```
GLBL_CV = sqlContext.sql("""select mer_id, SRC_STM_ID,mo_id, sum(tot_bsn_usd_vol) as tot_b
sn_usd_vol, sum(db_roc_ct) as db_roc_ct,
sum(net_roc_ct) as net_roc_ct,
sum(cr_roc_ct) as cr_roc_ct,
sum(dcn_rev_usd_am) as dcn_rev_usd_am
from AWB19.MER_GLBL_SUBM_MO_F_ALL
where mo_id >= 201101
group by mer_id, SRC_STM_ID,mo_id
order by mer_id, SRC_STM_ID,mo_id """)
```

In [224]:

```
DATE = dt.date.today()
yr_ly3 = DATE.year-3
yr_ly = DATE.year-1
yr_ly2 = DATE.year-2
last_r12_month = (dt.date.today() + relativedelta(months=-13)).strftime("%Y%m")
cur_year_begin = (dt.date.today().replace(month=1,day=1)).strftime("%Y%m")
r12_month = (dt.date.today() + relativedelta(months=-12)).strftime("%Y%m")
print(yr_ly3,yr_ly, yr_ly2,last_r12_month,cur_year_begin,r12_month)
```

```
2019 2021 2020 202109 202201 202110
```

In [225]:

```
# all_mer_out1.createOrReplaceTempView("all_mer_out1")
GLBL_CV.createOrReplaceTempView("GLBL_CV")
```

In [226]:

```
ALL_MER_CV = sqlContext.sql("""
Select a.*,
sum(case when mo_id >= cast('{yr_ly3}01' as int) and mo_id <= cast('{yr_ly3}12' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) ly3_cv_fy,
sum(case when mo_id >= cast('{yr_ly2}01' as int) and mo_id <= cast('{yr_ly2}12' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) ly2_cv_fy,
sum(case when mo_id >= cast('{yr_ly}01' as int) and mo_id <= cast('{yr_ly}12' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) ly_cv_fy,
sum(case when mo_id = cast('{last_r12_month}' as int) then tot_bsn_usd_vol else 0 end ) over(partition by a.mer_id) LY_CV_CurM,
sum(case when mo_id >= (cast('{cur_year_begin}' as int )-100) and mo_id <= (cast('{cur_month}' as int)-100) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) LY_CV_YTD,

sum(case when mo_id >=(cast('{r12_month}' as int)-100) and mo_id <=(cast('{cur_month}' as int)-100) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) LY_CV_R12,
sum(case when mo_id >=(cast('{r3_month}' as int)-100) and mo_id <=(cast('{cur_month}' as int)-100) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) LY_CV_R3,
sum(case when mo_id >=(cast('{r6_month}' as int)-100) and mo_id <=(cast('{cur_month}' as int)-100) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) LY_CV_R6,

sum(case when mo_id = cast('{cur_month}' as int) then tot_bsn_usd_vol else 0 end ) over(partition by a.mer_id) CV_CurM,
sum(case when mo_id >=cast('{cur_year_begin}' as int ) and mo_id <= cast('{cur_month}' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) CV_YTD,

sum(case when mo_id >=cast('{r12_month}' as int) and mo_id <=cast('{cur_month}' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) CV_R12,
sum(case when mo_id >=cast('{r3_month}' as int) and mo_id <=cast('{cur_month}' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) CV_R3,
sum(case when mo_id >=cast('{r6_month}' as int) and mo_id <=cast('{cur_month}' as int) then tot_bsn_usd_vol else 0 end) over(partition by a.mer_id) CV_R6,

sum(case when mo_id >=cast('{yr_ly3}01' as int) and mo_id <= cast('{yr_ly2}12' as int)  then net_roc_ct else 0 end) over(partition by a.mer_id) ly3_roc_fy,
sum(case when mo_id >=cast('{yr_ly2}01' as int) and mo_id <= cast('{yr_ly2}12' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ly2_roc_fy,
sum(case when mo_id >=cast('{yr_ly}01' as int) and mo_id <= cast('{yr_ly2}12' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ly_roc_fy,
sum(case when mo_id =cast('{last_r12_month}' as int) then net_roc_ct else 0 end ) over(partition by a.mer_id) LY_roc_CurM,
sum(case when mo_id >=(cast('{cur_year_begin}' as int )-100) and mo_id <=(cast('{cur_month}' as int)-100) then net_roc_ct else 0 end) over(partition by a.mer_id) LY_ROC_YTD,

sum(case when mo_id >=(cast('{r12_month}' as int)-100) and mo_id <=(cast('{cur_month}' as int)-100)then net_roc_ct else 0 end) over(partition by a.mer_id) LY_ROC_R12,
sum(case when mo_id >=(cast('{r3_month}' as int)-100) and mo_id <= (cast('{cur_month}' as int)-100) then net_roc_ct else 0 end) over(partition by a.mer_id) LY_ROC_R3,
sum(case when mo_id >=(cast('{r6_month}' as int)-100) and mo_id <= (cast('{cur_month}' as int)-100) then net_roc_ct else 0 end) over(partition by a.mer_id) LY_ROC_R6,

sum(case when mo_id =cast('{cur_month}' as int) then net_roc_ct else 0 end ) over(partition by a.mer_id) ROC_CurM,
sum(case when mo_id >=cast('{cur_year_begin}' as int ) and mo_id <= cast('{cur_month}' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ROC_YTD,
sum(case when mo_id >=(cast('{r12_month}' as int)-100) and mo_id <=cast('{cur_month}' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ROC_R12,
sum(case when mo_id >=(cast('{r3_month}' as int)-100) and mo_id <=cast('{cur_month}' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ROC_R3,
sum(case when mo_id >=(cast('{r6_month}' as int)-100) and mo_id <=cast('{cur_month}' as int) then net_roc_ct else 0 end) over(partition by a.mer_id) ROC_R6,

sum(case when mo_id >=cast('{r12_month}' as int) and mo_id <=cast('{cur_month}' as int) then db_roc_ct else 0 end) over(partition by a.mer_id) DBROCS_R12,
sum(case when mo_id =cast('{cur_month}' as int) then DCN_REV_USD_AM else 0 end ) over(partition by a.mer_id) DCN_REV_CurM,
sum(case when mo_id =cast('{r12_month}' as int) then DCN_REV_USD_AM else 0 end ) over(partition by a.mer_id) DCN_REV_LY_CurM,
sum(case when mo_id >=cast('{cur_year_begin}' as int ) and mo_id <=cast('{cur_month}' as int) then DCN_REV_USD_AM else 0 end) over(partition by a.mer_id) DCN_REV_YTD,
sum(case when mo_id >=cast('{r12_month}' as int) and mo_id <=cast('{cur_month}' as int) then DCN_REV_USD_AM else 0 end) over(partition by a.mer_id) DCN_REV_R12,
sum(case when mo_id >=(cast('{r12_month}' as int)-100) and mo_id <= (cast('{cur_month}' as int)-100) then DCN_REV_USD_AM else 0 end) over(partition by a.mer_id) LY_DCN_REV_R12,
sum(case when mo_id >=(cast('{cur_year_begin}' as int )-100) and mo_id <=(cast('{cur_month}' as int)-100) then DCN_REV_USD_AM else 0 end) over(partition by a.mer_id) LY_DCN_REV_YTD

from all_mer_out as a left join GLBL_CV as b on
a.mer_id=b.mer_id
""".format(yr_ly3 = yr_ly3,yr_ly2=yr_ly2,yr_ly=yr_ly,cur_year_begin = cur_year_begin, cur_month = cur_month, r12_month=r12_month, r3_month=r3_month,r6_month=r6_month, last_r12_month=last_r12_month ))
```

In [227]:

```
ALL_MER_CV1 = ALL_MER_CV.withColumn("cv_active_ind", when((col("DBROCS_R12")>=1)&(col("CV_
R12")>=0.000005),1).otherwise(0)) \
                        .withColumn("cur_cv_r12_band", when(col("cv_r12")<=0,"01.<=0
") \
                                   .when((col("cv_r12")>0)&(col("cv_r12")<50),'02.$1-<$50
') \
                                   .when((col("cv_r12")>=50)&(col("cv_r12")<250),'03.$50-<
250 ') \
                                   .when((col("cv_r12")>=250)&(col("cv_r12")<500),'04.$250
-<500 ') \
                                   .when((col("cv_r12")>=500)&(col("cv_r12")<1000),'05.$50
0-<1K  ') \
                                   .when((col("cv_r12")>=1000)&(col("cv_r12")<2000),'06.$1
K-<2K   ') \
                                   .when((col("cv_r12")>=2000)&(col("cv_r12")<30000),'0
7.$2K-<3K   ') \
                                   .when((col("cv_r12")>=3000)&(col("cv_r12")<5000),'08.$3
K-<5K ') \
                                   .when((col("cv_r12")>=5000)&(col("cv_r12")<10000),'0
9.$5K-<10K ') \
                                   .when((col("cv_r12")>=10000)&(col("cv_r12")<15000),'1
0.$10K-<15K   ') \
                                   .when((col("cv_r12")>=15000)&(col("cv_r12")<25000),'1
1.$15K-<25K   ') \
                                   .when((col("cv_r12")>=25000)&(col("cv_r12")<50000),'1
2.$25K-<50K   ') \
                                   .when((col("cv_r12")>=50000)&(col("cv_r12")<100000),'1
3.$50K-<100K  ')\
                                   .when((col("cv_r12")>=100000)&(col("cv_r12")<250000),'1
4.$100K-<250K ')\
                                   .when((col("cv_r12")>=250000)&(col("cv_r12")<500000),'1
5.$250K-<500K ')\
                                   .when((col("cv_r12")>=500000)&(col("cv_r12")<750000),'1
6.$500K-<750K ')\
                                   .when((col("cv_r12")>=750000)&(col("cv_r12")<1000000),
'17.$750K-<1MM  ')\
                                   .when((col("cv_r12")>=1000000)&(col("cv_r12")<2000000),
'18.$1MM-<2MM   ')\
                                   .when((col("cv_r12")>=2000000)&(col("cv_r12")<3000000),
'19.$2MM-<3MM   ')\
                                   .when((col("cv_r12")>=3000000)&(col("cv_r12")<10000000
),'20.$3MM-<10MM  ')\
                                   .when(col("cv_r12")>=10000000,'21.$10MM+      ')) \
                        .withColumn("cur_cv_r3_band", when(col("cv_r3")<=0,'01.<=0        '
) \
                                   .when((col("cv_r3")>0)&(col("cv_r3")<50),'02.$1-<$50
 ') \
                                   .when((col("cv_r3")>=50)&(col("cv_r3")<250),'03.$50-<25
0 ') \
                                   .when((col("cv_r3")>=250)&(col("cv_r3")<500),'04.$250-<
500 ') \
                                   .when((col("cv_r3")>=500)&(col("cv_r3")<1000),'05.$500-
<1K  ') \
                                   .when((col("cv_r3")>=1000)&(col("cv_r3")<2000),'06.$1K-
<2K   ') \
                                   .when((col("cv_r3")>=2000)&(col("cv_r3")<30000),'07.$2K
-<3K    ') \
                                   .when((col("cv_r3")>=3000)&(col("cv_r3")<5000),'08.$3K-
<5K ') \
                                   .when((col("cv_r3")>=5000)&(col("cv_r3")<10000),'09.$5K
-<10K ') \
                                   .when((col("cv_r3")>=10000)&(col("cv_r3")<15000),'10.$1
0K-<15K    ') \
                                   .when((col("cv_r3")>=15000)&(col("cv_r3")<25000),'11.$1
5K-<25K    ') \
                                   .when((col("cv_r3")>=25000)&(col("cv_r3")<50000),'12.$2
5K-<50K    ') \
                                   .when((col("cv_r3")>=50000)&(col("cv_r3")<100000),'1
3.$50K-<100K  ')\
                                   .when((col("cv_r3")>=100000)&(col("cv_r3")<250000),'1
4.$100K-<250K ')\
                                   .when((col("cv_r3")>=250000)&(col("cv_r3")<500000),'1
5.$250K-<500K ')\
                                   .when((col("cv_r3")>=500000)&(col("cv_r3")<750000),'1
6.$500K-<750K ')\
                                   .when((col("cv_r3")>=750000)&(col("cv_r3")<1000000),'1
7.$750K-<1MM  ')\
                                   .when((col("cv_r3")>=1000000)&(col("cv_r3")<2000000),'1
8.$1MM-<2MM   ')\
                                   .when((col("cv_r3")>=2000000)&(col("cv_r3")<3000000),'1
9.$2MM-<3MM   ')\
                                   .when((col("cv_r3")>=3000000)&(col("cv_r3")<10000000),
'20.$3MM-<10MM  ')\
                                   .when(col("cv_r3")>=10000000,'21.$10MM+      ')) \
                        .withColumn("cur_cv_r6_band", when(col("cv_r6")<0, "01.<=0        "
) \
                                   .when((col("cv_r6")>0)&(col("cv_r6")<50),'02.$1-<$50
 ') \
                                   .when((col("cv_r6")>=50)&(col("cv_r6")<250),'03.$50-<25
0 ') \
                                   .when((col("cv_r6")>=250)&(col("cv_r6")<500),'04.$250-<
500 ') \
                                   .when((col("cv_r6")>=500)&(col("cv_r6")<1000),'05.$500-
<1K  ') \
                                   .when((col("cv_r6")>=1000)&(col("cv_r6")<2000),'06.$1K-
<2K   ') \
                                   .when((col("cv_r6")>=2000)&(col("cv_r6")<30000),'07.$2K
-<3K    ') \
                                   .when((col("cv_r6")>=3000)&(col("cv_r6")<5000),'08.$3K-
<5K ') \
                                   .when((col("cv_r6")>=5000)&(col("cv_r6")<10000),'09.$5K
-<10K ') \
                                   .when((col("cv_r6")>=10000)&(col("cv_r6")<15000),'10.$1
0K-<15K    ') \
                                   .when((col("cv_r6")>=15000)&(col("cv_r6")<25000),'11.$1
5K-<25K    ') \
                                   .when((col("cv_r6")>=25000)&(col("cv_r6")<50000),'12.$2
5K-<50K    ') \
                                   .when((col("cv_r6")>=50000)&(col("cv_r6")<100000),'1
3.$50K-<100K  ')\
                                   .when((col("cv_r6")>=100000)&(col("cv_r6")<250000),'1
4.$100K-<250K ')\
                                   .when((col("cv_r6")>=250000)&(col("cv_r6")<500000),'1
5.$250K-<500K ')\
                                   .when((col("cv_r6")>=500000)&(col("cv_r6")<750000),'1
6.$500K-<750K ')\
                                   .when((col("cv_r6")>=750000)&(col("cv_r6")<1000000),'1
7.$750K-<1MM  ')\
                                   .when((col("cv_r6")>=1000000)&(col("cv_r6")<2000000),'1
8.$1MM-<2MM   ')\
                                   .when((col("cv_r6")>=2000000)&(col("cv_r6")<3000000),'1
9.$2MM-<3MM   ')\
                                   .when((col("cv_r6")>=3000000)&(col("cv_r6")<10000000),
'20.$3MM-<10MM  ')\
                                   .when(col("cv_r6")>=10000000,'21.$10MM+'))
```

In [228]:

```
ALL_MER_CV1=ALL_MER_CV1.distinct().sort(["mer_id"])
```

In [229]:

```
ALL_MER_CV1 = ALL_MER_CV1.withColumn('creat_ts',to_date(ALL_MER_CV1.creat_ts, 'yyyy-MM-dd'
))
```

In [230]:

```
ob_char_temp=sqlContext.sql("""select * from cstonedb3.gmdl_mer_seller  where sell_ctry_cd
in ('US','PR','VI') """).filter(~col("tpa_se10").isin('3343897567','2044727287','104909217
4','2044439040','4452822323','3042881599','4105150783','')).sort(["subm_se10", "SELL_ID"])
.dropDuplicates(["subm_se10","SELL_ID"])
```

In [231]:

```
ob_char_temp = ob_char_temp.withColumn('lst_spnsr_mer_updt_ts',to_date(ob_char_temp.lst_sp
nsr_mer_updt_ts, 'yyyy-MM-dd')) \
                           .withColumn('creat_ts',to_date(ob_char_temp.creat_ts, 'yyyy-MM-
dd'))
```

In [232]:

```
ob_char_temp.createOrReplaceTempView("ob_char_temp")
ob_char_temp= sqlContext.sql("""
select *,
to_date(creat_ts) as create_dt,
to_date(lst_spnsr_mer_updt_ts) as status_dt,
to_date(creat_ts) as sign_date,
to_date(creat_ts) as open_date
from ob_char_temp """).sort(["subm_se10","sell_id"])
```

In [234]:

```
ob_char_temp = drop_duplicate_columns(ob_char_temp)
```

```
['cstone_feed_key', 'cstone_last_updatetm', 'sell_se10', 'srce_sys_id', 'bus_
ctr_cd', 'subm_se10', 'sell_id', 'subm_id', 'subm_type_cd', 'payee_se10', 'tp
a_se10', 'rel_grp_type_no', 'rel_grp_type_updt_dt', 'rel_subgrp_type_cd', 'mt
ch_tpa_se10', 'mtch_sell_subm_se10', 'mtch_sell_id', 'mtch_sell_srce_sys_id',
'mtch_sell_bus_ctr_cd', 'mkt_elig_cd', 'init_subm_dt', 'init_rec_srce_cd', 'c
reat_ts', 'sell_lgl_nm', 'sell_dba_nm', 'sell_bus_tax_id', 'sell_mer_ctgy_c
d', 'sell_st_ad_line1', 'sell_st_ad_line2', 'sell_st_ad_line3', 'sell_st_ad_l
ine4', 'sell_st_ad_line5', 'sell_city_nm', 'sell_rgn_area_cd', 'sell_pstl_c
d', 'sell_ctry_cd', 'sell_bus_phone_no', 'sell_url', 'sell_email_ad', 'subm_c
urr_cd', 'settle_curr_cd', 'sgfnt_own_first_nm', 'sgfnt_own_lst_nm', 'sgfnt_o
wn_nat_id', 'sgfnt_own_birth_dt', 'sgfnt_own_ctry_cd', 'sgfnt_own_st_ad_line
1', 'sgfnt_own_st_ad_line2', 'sgfnt_own_st_ad_line3', 'sgfnt_own_st_ad_line
4', 'sgfnt_own_st_ad_line5', 'sgfnt_own_city_nm', 'sgfnt_own_rgn_area_cd', 's
gfnt_own_pstl_cd', 'lst_spnsr_mer_updt_ts', 'sale_chan_cd', 'sale_chan_nm',
'sale_repr_id', 'iso_regis_no', 'seims_indus_ds_cd', 'genesis_se_indus_ds_c
d', 'sell_acct_open_dt', 'sell_prtr_cd', 'lang_pref_cd', 'jcb_acpt_in', 'sell
_sta_cd', 'sell_strt_dt', 'sell_trmn_dt', 'prim_auth_sign_first_nm', 'prim_au
th_sign_lst_nm', 'prim_auth_sign_birth_dt', 'own_type_cd', 'sell_acpt_sta_c
d', 'sell_acpt_sta_updt_dt', 'bank_card_act_in', 'mtch_sell_mer_id', 'scnd_sg
fnt_own_first_nm', 'scnd_sgfnt_own_lst_nm', 'scnd_sgfnt_own_birth_dt', 'scnd_
sgfnt_own_ctry_cd', 'scnd_sgfnt_own_st_ad_line1', 'scnd_sgfnt_own_st_ad_line
2', 'scnd_sgfnt_own_st_ad_line3', 'scnd_sgfnt_own_st_ad_line4', 'scnd_sgfnt_o
wn_st_ad_line5', 'scnd_sgfnt_own_city_nm', 'scnd_sgfnt_own_rgn_area_cd', 'scn
d_sgfnt_own_pstl_cd', 'third_sgfnt_own_first_nm', 'third_sgfnt_own_lst_nm',
'third_sgfnt_own_birth_dt', 'third_sgfnt_own_ctry_cd', 'third_sgfnt_own_ad_li
ne1', 'third_sgfnt_own_ad_line2', 'third_sgfnt_own_ad_line3', 'third_sgfnt_ow
n_ad_line4', 'third_sgfnt_own_ad_line5', 'third_sgfnt_own_city_nm', 'third_sg
fnt_own_rgn_area_cd', 'third_sgfnt_own_pstl_cd', 'fourth_sgfnt_own_first_nm',
'fourth_sgfnt_own_lst_nm', 'fourth_sgfnt_own_birth_dt', 'fourth_sgfnt_own_ctr
y_cd', 'fourth_sgfnt_own_ad_line1', 'fourth_sgfnt_own_ad_line2', 'fourth_sgfn
t_own_ad_line3', 'fourth_sgfnt_own_ad_line4', 'fourth_sgfnt_own_ad_line5', 'f
ourth_sgfnt_own_city_nm', 'fourth_sgfnt_own_rgn_area_cd', 'fourth_sgfnt_own_p
stl_cd', 'est_chrg_vol_dr_usd_am', 'est_chrg_vol_dr_loc_am', 'est_chrg_vol_cr
_usd_am', 'est_chrg_vol_cr_loc_am', 'est_chrg_vol_dr_trans_ct', 'est_chrg_vol
_cr_trans_ct', 'est_chrgbk_ct', 'est_chrgbk_usd_am', 'est_chrgbk_loc_am', 'cr
eat_ts1', 'spnsr_mer_fl_lst_updt_ts', 'tax_id', 'auth_sign_ttl', 'auth_sign_n
at_id_no', 'auth_sign_phys_ad_line1', 'auth_sign_phys_ad_line2', 'auth_sign_p
hys_ad_line3', 'auth_sign_phys_ad_line4', 'auth_sign_phys_ad_line5', 'auth_si
gn_phys_ad_post_town_nm', 'auth_sign_phys_ad_rgn_area_cd', 'auth_sign_phys_ad
_pstl_cd', 'auth_sign_phys_ad_ctry_cd', 'scnd_sgfnt_own_nat_id_no', 'third_sg
fnt_own_nat_id_no', 'fourth_sgfnt_own_nat_id_no', 'mer_prop_in', 'create_dt',
'status_dt', 'sign_date', 'open_date']
```

In [235]:

```
ob_char_temp.createOrReplaceTempView("ob_char_temp")
```

In [236]:

```
QUERY_FOR_SPNSR_MER_LINK = sqlContext.sql("""
SELECT T1.NEW_SUBM_NTRL_MER_ID AS NEW_SUBM_SE_NO,
T1.NEW_SELL_ID,
T1.ORIG_SUBM_NTRL_MER_ID AS ORIG_SUBM_SE_NO,
T1.CHNG_RSN_CD,
trim(T1.SELL_CHNG_STA_CD) as SELL_CHNG_STA_CD,
trim(T1.ORIG_SUBM_NTRL_MER_ID) as ORIG_SUBM_NTRL_MER_ID,
trim(T1.ORIG_SELL_ID) as ORIG_SELL_ID,
trim(T2.SELL_ID) as SELL_ID,
trim(T2.subm_se10) as subm_se_no,

T2.sign_date AS ORIG_SIGN_DATE,
T2.init_subm_dt AS ORIG_INIT_SUBM_DT
FROM cstonedb3.gmar_sponsor_merchant_linkage T1
left join OB_CHAR_TEMP  T2
on (T1.ORIG_SUBM_NTRL_MER_ID) = (T2.SUBM_SE10) and (T1.ORIG_SELL_ID) = (T2.SELL_ID)
where (T1.SELL_CHNG_STA_CD) = 'C'  """).drop("SELL_CHNG_STA_CD","ORIG_SUBM_NTRL_MER_ID",
"SELL_ID","subm_se_no").sort(["ORIG_SUBM_SE_NO", "orig_sell_id", "NEW_SUBM_SE_NO", "NEW_SE
LL_ID"])
```

In [237]:

```
CROSS_XREF_TABLE = QUERY_FOR_SPNSR_MER_LINK.withColumnRenamed("NEW_SUBM_SE_NO", "NEW_SUBM_
SE_NO1").withColumnRenamed("ORIG_SUBM_SE_NO", "ORIG_SUBM_SE_NO1")
```

In [238]:

```
from pyspark.sql.types import StringType
CROSS_XREF_TABLE = CROSS_XREF_TABLE.withColumn("NEW_SUBM_SE_NO",col("NEW_SUBM_SE_NO1").cas
t(StringType())).withColumn("ORIG_SUBM_SE_NO",col("ORIG_SUBM_SE_NO1").cast(StringType())).
drop("NEW_SUBM_SE_NO1","ORIG_SUBM_SE_NO1")
```

In [239]:

```
CROSS_XREF_TABLE.createOrReplaceTempView("CROSS_XREF_TABLE")
# table_isExist_drop("AWB19", "CROSS_XREF_TABLE")
# sqlContext.sql("create table AWB19.CROSS_XREF_TABLE as select * from CROSS_XREF_TABLE")
```

In [240]:

```
OLD_PRTR_INFO = CROSS_XREF_TABLE.select("ORIG_SUBM_SE_NO","ORIG_SELL_ID", "NEW_SUBM_SE_NO"
, "NEW_SELL_ID", "CHNG_RSN_CD") \
                                .sort(["ORIG_SUBM_SE_NO","ORIG_SELL_ID"]) \
                                .withColumnRenamed("ORIG_SUBM_SE_NO", "SUBM_SE_NO").withCo
lumnRenamed("ORIG_SELL_ID", "SELL_ID")
```

In [241]:

```
NEW_PRTR_INFO = CROSS_XREF_TABLE.select("NEW_SUBM_SE_NO","NEW_SELL_ID", "ORIG_SIGN_DATE", "ORIG_INIT_SUBM_DT") \
                                .sort(["NEW_SUBM_SE_NO","NEW_SELL_ID"]) \
                                .withColumnRenamed("NEW_SUBM_SE_NO", "SUBM_SE_NO").withColumnRenamed("NEW_SELL_ID", "SELL_ID")
```

In [242]:

```
OLD_PRTR_INFO.createOrReplaceTempView("OLD_PRTR_INFO")
NEW_PRTR_INFO.createOrReplaceTempView("NEW_PRTR_INFO")
```

In [243]:

```
ob_old_inner = ob_char_temp.join(OLD_PRTR_INFO,["SELL_ID"],"left").withColumn('SELL_ACPT_STA_CD', OLD_PRTR_INFO['CHNG_RSN_CD']) \
                                .withColumn("SELL_ACPT_STA_UPDT_DT", when((col("SELL_ACPT_STA_UPDT_DT").isNull()),ACPT_STA_UPDT_DT) \
                                .otherwise(col("SELL_ACPT_STA_UPDT_DT")))


ob_new_inner = ob_char_temp.join(NEW_PRTR_INFO,["SELL_ID"],"left").withColumn('SIGN_DATE', NEW_PRTR_INFO['ORIG_SIGN_DATE']) \
                                .withColumn('open_date', NEW_PRTR_INFO['ORIG_SIGN_DATE']) \
                                .withColumn("INIT_SUBM_DT", when((col("ORIG_INIT_SUBM_DT").isNotNull()),col("ORIG_INIT_SUBM_DT")) \
                                .otherwise(col("INIT_SUBM_DT")))

NEW_OB_CHAR_TEMP = ob_old_inner.join(ob_new_inner,["SELL_ID"], "fullouter").drop("NEW_SUBM_SE_NO", "NEW_SELL_ID", "CHNG_RSN_CD", "ORIG_SIGN_DATE", "ORIG_INIT_SUBM_DT")
```

In [244]:

```
NEW_OB_CHAR_TEMP = drop_duplicate_columns(NEW_OB_CHAR_TEMP)
```

```
['sell_id', 'third_sgfnt_own_city_nm', 'sgfnt_own_lst_nm', 'auth_sign_nat_id_
no', 'sell_email_ad', 'scnd_sgfnt_own_st_ad_line4', 'status_dt', 'subm_id',
'cstone_last_updatetm', 'sell_mer_ctgy_cd', 'scnd_sgfnt_own_nat_id_no', 'four
th_sgfnt_own_birth_dt', 'sell_acpt_sta_updt_dt', 'lst_spnsr_mer_updt_ts', 're
l_grp_type_no', 'sell_acpt_sta_cd', 'scnd_sgfnt_own_st_ad_line1', 'third_sgfn
t_own_rgn_area_cd', 'sell_st_ad_line5', 'genesis_se_indus_ds_cd', 'tax_id',
'jcb_acpt_in', 'mtch_sell_subm_se10', 'payee_se10', 'fourth_sgfnt_own_ad_line
2', 'creat_ts1', 'sell_st_ad_line3', 'fourth_sgfnt_own_ctry_cd', 'est_chrg_vo
l_cr_loc_am', 'open_date', 'third_sgfnt_own_nat_id_no', 'auth_sign_phys_ad_ct
ry_cd', 'rel_subgrp_type_cd', 'scnd_sgfnt_own_first_nm', 'scnd_sgfnt_own_st_a
d_line3', 'settle_curr_cd', 'fourth_sgfnt_own_rgn_area_cd', 'third_sgfnt_own_
ad_line3', 'third_sgfnt_own_ad_line2', 'mtch_sell_id', 'scnd_sgfnt_own_lst_n
m', 'sell_rgn_area_cd', 'third_sgfnt_own_pstl_cd', 'subm_se10', 'scnd_sgfnt_o
wn_birth_dt', 'bus_ctr_cd', 'sgfnt_own_st_ad_line5', 'scnd_sgfnt_own_st_ad_li
ne2', 'sgfnt_own_pstl_cd', 'fourth_sgfnt_own_first_nm', 'auth_sign_phys_ad_po
st_town_nm', 'auth_sign_phys_ad_line4', 'sell_se10', 'sgfnt_own_st_ad_line3',
'est_chrgbk_loc_am', 'spnsr_mer_fl_lst_updt_ts', 'third_sgfnt_own_ad_line1',
'init_rec_srce_cd', 'bank_card_act_in', 'sell_sta_cd', 'sell_acct_open_dt',
'scnd_sgfnt_own_rgn_area_cd', 'sell_bus_phone_no', 'scnd_sgfnt_own_city_nm',
'seims_indus_ds_cd', 'third_sgfnt_own_lst_nm', 'sgfnt_own_st_ad_line4', 'four
th_sgfnt_own_ad_line1', 'creat_ts', 'sell_lgl_nm', 'sale_repr_id', 'sell_ctry
_cd', 'sgfnt_own_ctry_cd', 'auth_sign_phys_ad_line2', 'mtch_sell_bus_ctr_cd',
'third_sgfnt_own_ctry_cd', 'fourth_sgfnt_own_lst_nm', 'third_sgfnt_own_birth_
dt', 'scnd_sgfnt_own_st_ad_line5', 'mtch_tpa_se10', 'scnd_sgfnt_own_pstl_cd',
'fourth_sgfnt_own_nat_id_no', 'sell_strt_dt', 'own_type_cd', 'est_chrgbk_ct',
'mkt_elig_cd', 'sgfnt_own_nat_id', 'est_chrg_vol_cr_trans_ct', 'fourth_sgfnt_
own_ad_line4', 'sgfnt_own_st_ad_line2', 'fourth_sgfnt_own_ad_line5', 'auth_si
gn_ttl', 'third_sgfnt_own_ad_line5', 'fourth_sgfnt_own_city_nm', 'est_chrg_vo
l_dr_loc_am', 'sell_trmn_dt', 'prim_auth_sign_birth_dt', 'sell_bus_tax_id',
'auth_sign_phys_ad_line5', 'sign_date', 'sgfnt_own_first_nm', 'sell_pstl_cd',
'srce_sys_id', 'mtch_sell_srce_sys_id', 'subm_type_cd', 'sgfnt_own_rgn_area_c
d', 'sell_dba_nm', 'est_chrg_vol_cr_usd_am', 'auth_sign_phys_ad_line3', 'thir
d_sgfnt_own_first_nm', 'mer_prop_in', 'tpa_se10', 'prim_auth_sign_lst_nm', 's
gfnt_own_city_nm', 'prim_auth_sign_first_nm', 'sell_st_ad_line4', 'est_chrg_v
ol_dr_trans_ct', 'sgfnt_own_st_ad_line1', 'init_subm_dt', 'lang_pref_cd', 'se
ll_url', 'sell_prtr_cd', 'est_chrgbk_usd_am', 'sale_chan_nm', 'est_chrg_vol_d
r_usd_am', 'cstone_feed_key', 'sell_st_ad_line2', 'subm_curr_cd', 'sale_chan_
cd', 'iso_regis_no', 'mtch_sell_mer_id', 'sgfnt_own_birth_dt', 'fourth_sgfnt_
own_pstl_cd', 'fourth_sgfnt_own_ad_line3', 'auth_sign_phys_ad_rgn_area_cd',
'create_dt', 'third_sgfnt_own_ad_line4', 'scnd_sgfnt_own_ctry_cd', 'auth_sign
_phys_ad_line1', 'rel_grp_type_updt_dt', 'sell_st_ad_line1', 'sell_city_nm',
'auth_sign_phys_ad_pstl_cd', 'subm_se_no', 'third_sgfnt_own_city_nm_1', 'sgfn
t_own_lst_nm_1', 'auth_sign_nat_id_no_1', 'sell_email_ad_1', 'scnd_sgfnt_own_
st_ad_line4_1', 'status_dt_1', 'subm_id_1', 'cstone_last_updatetm_1', 'sell_m
er_ctgy_cd_1', 'scnd_sgfnt_own_nat_id_no_1', 'fourth_sgfnt_own_birth_dt_1',
'sell_acpt_sta_updt_dt_1', 'lst_spnsr_mer_updt_ts_1', 'rel_grp_type_no_1', 's
ell_acpt_sta_cd_1', 'scnd_sgfnt_own_st_ad_line1_1', 'third_sgfnt_own_rgn_area
_cd_1', 'sell_st_ad_line5_1', 'genesis_se_indus_ds_cd_1', 'tax_id_1', 'jcb_ac
pt_in_1', 'mtch_sell_subm_se10_1', 'payee_se10_1', 'fourth_sgfnt_own_ad_line2
_1', 'creat_ts1_1', 'sell_st_ad_line3_1', 'fourth_sgfnt_own_ctry_cd_1', 'est_
chrg_vol_cr_loc_am_1', 'open_date_1', 'third_sgfnt_own_nat_id_no_1', 'auth_si
gn_phys_ad_ctry_cd_1', 'rel_subgrp_type_cd_1', 'scnd_sgfnt_own_first_nm_1',
'scnd_sgfnt_own_st_ad_line3_1', 'settle_curr_cd_1', 'fourth_sgfnt_own_rgn_are
a_cd_1', 'third_sgfnt_own_ad_line3_1', 'third_sgfnt_own_ad_line2_1', 'mtch_se
ll_id_1', 'scnd_sgfnt_own_lst_nm_1', 'sell_rgn_area_cd_1', 'third_sgfnt_own_p
stl_cd_1', 'subm_se10_1', 'scnd_sgfnt_own_birth_dt_1', 'bus_ctr_cd_1', 'sgfnt
_own_st_ad_line5_1', 'scnd_sgfnt_own_st_ad_line2_1', 'sgfnt_own_pstl_cd_1',
'fourth_sgfnt_own_first_nm_1', 'auth_sign_phys_ad_post_town_nm_1', 'auth_sign
_phys_ad_line4_1', 'sell_se10_1', 'sgfnt_own_st_ad_line3_1', 'est_chrgbk_loc_
am_1', 'spnsr_mer_fl_lst_updt_ts_1', 'third_sgfnt_own_ad_line1_1', 'init_rec_
srce_cd_1', 'bank_card_act_in_1', 'sell_sta_cd_1', 'sell_acct_open_dt_1', 'sc
nd_sgfnt_own_rgn_area_cd_1', 'sell_bus_phone_no_1', 'scnd_sgfnt_own_city_nm_
1', 'seims_indus_ds_cd_1', 'third_sgfnt_own_lst_nm_1', 'sgfnt_own_st_ad_line4
_1', 'fourth_sgfnt_own_ad_line1_1', 'creat_ts_1', 'sell_lgl_nm_1', 'sale_repr
_id_1', 'sell_ctry_cd_1', 'sgfnt_own_ctry_cd_1', 'auth_sign_phys_ad_line2_1',
'mtch_sell_bus_ctr_cd_1', 'third_sgfnt_own_ctry_cd_1', 'fourth_sgfnt_own_lst_
nm_1', 'third_sgfnt_own_birth_dt_1', 'scnd_sgfnt_own_st_ad_line5_1', 'mtch_tp
a_se10_1', 'scnd_sgfnt_own_pstl_cd_1', 'fourth_sgfnt_own_nat_id_no_1', 'sell_
strt_dt_1', 'own_type_cd_1', 'est_chrgbk_ct_1', 'mkt_elig_cd_1', 'sgfnt_own_n
at_id_1', 'est_chrg_vol_cr_trans_ct_1', 'fourth_sgfnt_own_ad_line4_1', 'sgfnt
_own_st_ad_line2_1', 'fourth_sgfnt_own_ad_line5_1', 'auth_sign_ttl_1', 'third
_sgfnt_own_ad_line5_1', 'fourth_sgfnt_own_city_nm_1', 'est_chrg_vol_dr_loc_am
_1', 'sell_trmn_dt_1', 'prim_auth_sign_birth_dt_1', 'sell_bus_tax_id_1', 'aut
h_sign_phys_ad_line5_1', 'sign_date_1', 'sgfnt_own_first_nm_1', 'sell_pstl_cd
_1', 'srce_sys_id_1', 'mtch_sell_srce_sys_id_1', 'subm_type_cd_1', 'sgfnt_own
_rgn_area_cd_1', 'sell_dba_nm_1', 'est_chrg_vol_cr_usd_am_1', 'auth_sign_phys
_ad_line3_1', 'third_sgfnt_own_first_nm_1', 'mer_prop_in_1', 'tpa_se10_1', 'p
rim_auth_sign_lst_nm_1', 'sgfnt_own_city_nm_1', 'prim_auth_sign_first_nm_1',
'sell_st_ad_line4_1', 'est_chrg_vol_dr_trans_ct_1', 'sgfnt_own_st_ad_line1_
1', 'init_subm_dt_1', 'lang_pref_cd_1', 'sell_url_1', 'sell_prtr_cd_1', 'est_
chrgbk_usd_am_1', 'sale_chan_nm_1', 'est_chrg_vol_dr_usd_am_1', 'cstone_feed_
key_1', 'sell_st_ad_line2_1', 'subm_curr_cd_1', 'sale_chan_cd_1', 'iso_regis_
no_1', 'mtch_sell_mer_id_1', 'sgfnt_own_birth_dt_1', 'fourth_sgfnt_own_pstl_c
d_1', 'fourth_sgfnt_own_ad_line3_1', 'auth_sign_phys_ad_rgn_area_cd_1', 'crea
te_dt_1', 'third_sgfnt_own_ad_line4_1', 'scnd_sgfnt_own_ctry_cd_1', 'auth_sig
n_phys_ad_line1_1', 'rel_grp_type_updt_dt_1', 'sell_st_ad_line1_1', 'sell_cit
y_nm_1', 'auth_sign_phys_ad_pstl_cd_1', 'subm_se_no_1']
,third_sgfnt_own_city_nm_1
,sgfnt_own_lst_nm_1
,auth_sign_nat_id_no_1
,sell_email_ad_1
,scnd_sgfnt_own_st_ad_line4_1
,status_dt_1
,subm_id_1
,cstone_last_updatetm_1
,sell_mer_ctgy_cd_1
,scnd_sgfnt_own_nat_id_no_1
,fourth_sgfnt_own_birth_dt_1
,sell_acpt_sta_updt_dt_1
,lst_spnsr_mer_updt_ts_1
,rel_grp_type_no_1
,sell_acpt_sta_cd_1
,scnd_sgfnt_own_st_ad_line1_1
,third_sgfnt_own_rgn_area_cd_1
,sell_st_ad_line5_1
,genesis_se_indus_ds_cd_1
,tax_id_1
,jcb_acpt_in_1
,mtch_sell_subm_se10_1
,payee_se10_1
,fourth_sgfnt_own_ad_line2_1
,creat_ts1_1
,sell_st_ad_line3_1
,fourth_sgfnt_own_ctry_cd_1
```

```
    ,est_chrg_vol_cr_loc_am_1
    ,open_date_1
    ,third_sgfnt_own_nat_id_no_1
    ,auth_sign_phys_ad_ctry_cd_1
    ,rel_subgrp_type_cd_1
    ,scnd_sgfnt_own_first_nm_1
    ,scnd_sgfnt_own_st_ad_line3_1
    ,settle_curr_cd_1
    ,fourth_sgfnt_own_rgn_area_cd_1
    ,third_sgfnt_own_ad_line3_1
    ,third_sgfnt_own_ad_line2_1
    ,mtch_sell_id_1
    ,scnd_sgfnt_own_lst_nm_1
    ,sell_rgn_area_cd_1
    ,third_sgfnt_own_pstl_cd_1
    ,subm_se10_1
    ,scnd_sgfnt_own_birth_dt_1
    ,bus_ctr_cd_1
    ,sgfnt_own_st_ad_line5_1
    ,scnd_sgfnt_own_st_ad_line2_1
    ,sgfnt_own_pstl_cd_1
    ,fourth_sgfnt_own_first_nm_1
    ,auth_sign_phys_ad_post_town_nm_1
    ,auth_sign_phys_ad_line4_1
    ,sell_se10_1
    ,sgfnt_own_st_ad_line3_1
    ,est_chrgbk_loc_am_1
    ,spnsr_mer_fl_lst_updt_ts_1
    ,third_sgfnt_own_ad_line1_1
    ,init_rec_srce_cd_1
    ,bank_card_act_in_1
    ,sell_sta_cd_1
    ,sell_acct_open_dt_1
    ,scnd_sgfnt_own_rgn_area_cd_1
    ,sell_bus_phone_no_1
    ,scnd_sgfnt_own_city_nm_1
    ,seims_indus_ds_cd_1
    ,third_sgfnt_own_lst_nm_1
    ,sgfnt_own_st_ad_line4_1
    ,fourth_sgfnt_own_ad_line1_1
    ,creat_ts_1
    ,sell_lgl_nm_1
    ,sale_repr_id_1
    ,sell_ctry_cd_1
    ,sgfnt_own_ctry_cd_1
    ,auth_sign_phys_ad_line2_1
    ,mtch_sell_bus_ctr_cd_1
    ,third_sgfnt_own_ctry_cd_1
    ,fourth_sgfnt_own_lst_nm_1
    ,third_sgfnt_own_birth_dt_1
    ,scnd_sgfnt_own_st_ad_line5_1
    ,mtch_tpa_se10_1
    ,scnd_sgfnt_own_pstl_cd_1
    ,fourth_sgfnt_own_nat_id_no_1
    ,sell_strt_dt_1
    ,own_type_cd_1
    ,est_chrgbk_ct_1
    ,mkt_elig_cd_1
    ,sgfnt_own_nat_id_1
    ,est_chrg_vol_cr_trans_ct_1
    ,fourth_sgfnt_own_ad_line4_1
    ,sgfnt_own_st_ad_line2_1
    ,fourth_sgfnt_own_ad_line5_1
    ,auth_sign_ttl_1
    ,third_sgfnt_own_ad_line5_1
    ,fourth_sgfnt_own_city_nm_1
    ,est_chrg_vol_dr_loc_am_1
    ,sell_trmn_dt_1
    ,prim_auth_sign_birth_dt_1
    ,sell_bus_tax_id_1
    ,auth_sign_phys_ad_line5_1
    ,sign_date_1
    ,sgfnt_own_first_nm_1
    ,sell_pstl_cd_1
    ,srce_sys_id_1
    ,mtch_sell_srce_sys_id_1
    ,subm_type_cd_1
    ,sgfnt_own_rgn_area_cd_1
    ,sell_dba_nm_1
    ,est_chrg_vol_cr_usd_am_1
    ,auth_sign_phys_ad_line3_1
    ,third_sgfnt_own_first_nm_1
    ,mer_prop_in_1
    ,tpa_se10_1
    ,prim_auth_sign_lst_nm_1
    ,sgfnt_own_city_nm_1
    ,prim_auth_sign_first_nm_1
    ,sell_st_ad_line4_1
    ,est_chrg_vol_dr_trans_ct_1
    ,sgfnt_own_st_ad_line1_1
    ,init_subm_dt_1
    ,lang_pref_cd_1
    ,sell_url_1
    ,sell_prtr_cd_1
    ,est_chrgbk_usd_am_1
    ,sale_chan_nm_1
    ,est_chrg_vol_dr_usd_am_1
    ,cstone_feed_key_1
    ,sell_st_ad_line2_1
    ,subm_curr_cd_1
    ,sale_chan_cd_1
    ,iso_regis_no_1
    ,mtch_sell_mer_id_1
    ,sgfnt_own_birth_dt_1
    ,fourth_sgfnt_own_pstl_cd_1
    ,fourth_sgfnt_own_ad_line3_1
    ,auth_sign_phys_ad_rgn_area_cd_1
    ,create_dt_1
    ,third_sgfnt_own_ad_line4_1
    ,scnd_sgfnt_own_ctry_cd_1
    ,auth_sign_phys_ad_line1_1
    ,rel_grp_type_updt_dt_1
    ,sell_st_ad_line1_1
    ,sell_city_nm_1
```

```
,auth_sign_phys_ad_pstl_cd_1
,subm_se_no_1
```

In [245]:

```
NEW_OB_CHAR_TEMP.createOrReplaceTempView("NEW_OB_CHAR_TEMP")
ob_char_temp = sqlContext.sql("""Select *,year(open_date) as open_year,
month(open_date) as open_month
from NEW_OB_CHAR_TEMP """)
```

In [246]:

```
# ob_char_temp.createOrReplaceTempView("ob_char_temp")
# sqlContext.sql("""select open_year,open_month, open_date from ob_char_temp""").show()
```

In [247]:

```
ob_char_temp.createOrReplaceTempView("ob_char_temp")
```

In [248]:

```
seller_temp = sqlContext.sql("""
select
case when b.sell_id is not null then b.sell_id else '' end as mer_id,
b.sell_se10 as se_no,
b.subm_se10,
b.payee_se10,
b.tpa_se10,
b.mtch_tpa_se10,
b.mtch_sell_id,
b.mtch_sell_subm_se10,
b.srce_sys_id as MER_SRCE_SYS_CD,
b.sell_dba_nm as DBA_NM,
b.sell_lgl_nm as Legal_NM,
b.sell_city_nm as city_nm,
b.REL_GRP_TYPE_NO,
b.REL_GRP_TYPE_UPDT_DT,
b.rel_subgrp_type_cd,

CASE
WHEN B.SELL_CTRY_CD = 'US' THEN '840'
WHEN B.SELL_CTRY_CD = 'CA' THEN '124'
WHEN B.SELL_CTRY_CD = 'MX' THEN '484'
WHEN B.SELL_CTRY_CD = 'PR' THEN '630'
WHEN B.SELL_CTRY_CD = 'VI' THEN '850'
END AS PHYS_AD_CTRY_CD,

b.sell_pstl_CD as PHYS_AD_POST_CD_TX,
b.SEIMS_INDUS_DS_CD as SEIMS_INDUS_DS_CD,
b.open_date as MER_SETUP_DT,
b.init_subm_dt as MER_ACT_DT,
b.SELL_BUS_PHONE_NO as PHONE_NO,

b.SELL_MER_CTGY_CD as MCC_INDUS_CD,
b.SELL_RGN_AREA_CD as PHYS_AD_RGN_AREA_CD,
b.sell_st_ad_line1 as AD_LINE_1_TX,
b.sell_st_ad_line2 as AD_LINE_2_TX,
b.sell_st_ad_line3 as AD_LINE_3_TX,
b.sell_email_ad as EMAIL_AD_TX,
b.CREAT_TS,
b.CREAT_TS,
b.sell_url,
b.INIT_REC_SRCE_CD,
b.ISO_REGIS_NO,
b.MKT_ELIG_CD,
b.BUS_CTR_CD as SRVC_BUS_CTR_CD,
b.LANG_PREF_CD,
b.sale_chan_cd,
b.sale_chan_nm,
b.subm_type_cd,
b.SALE_REPR_ID,

b.SELL_ACPT_STA_UPDT_DT,
b.SELL_ACPT_STA_CD as cur_mer_sta_cd,
b.SELL_ACPT_STA_CD,
b.bank_card_act_in,
b.JCB_ACPT_IN,
b.SGFNT_OWN_FIRST_NM,
b.SGFNT_OWN_LST_NM,
b.sign_Date
from  ob_char_temp as b""")
```

In [249]:

```
seller_temp.createOrReplaceTempView("seller_temp")
```

In [250]:

```
month_end = (dt.date.today().replace(day=1) + relativedelta(months=0,days=-1))
```

In [251]:

```
seller_temp1 = sqlContext.sql("""
Select a.*,
(case when PHYS_AD_CTRY_CD in ('840') then 'SELL' else 'SELL_INTL' end) as se_type,
year(mer_setup_dt) as open_year,
month(mer_setup_dt) as open_month,
year(MER_ACT_DT) as SUBM_YEAR,
month(MER_ACT_DT)  as SUBM_MONTH,
ceil(abs(months_between(last_day(add_months('{a}',-1)), mer_setup_dt))) as mer_tenure_mo_c
t,
(case when MER_ACT_DT is not null and  MER_ACT_DT <= {a} then ceil(abs(months_between(mer_
setup_dt, MER_ACT_DT))) else 9999 end) as activated_per
from seller_temp as a""".format(a=str(month_end)))
```

In [252]:

```
SELLER_MO_FIN_SMRY = sqlContext.sql("""select * from cstonedb3.gmar_seller_mo_fin_smry""")
```

In [253]:

```
SELLER_MO_FIN_SMRY = SELLER_MO_FIN_SMRY.sort(["SUBM_SE_NO","SELL_ID"])
```

In [256]:

```
NEW_SELLER_MO_FIN_SMRY =  SELLER_MO_FIN_SMRY.join(OLD_PRTR_INFO,["sell_id","subm_se_no"],
"left") \
                        .withColumn('SUBM_SE_NO', OLD_PRTR_INFO['NEW_SUBM_SE_NO']) \
                        .withColumn('SELL_ID', OLD_PRTR_INFO['NEW_SELL_ID']).drop("NEW_SUB
M_SE_NO","NEW_SELL_ID","CHNG_RSN_CD")
```

In [259]:

```
NEW_SELLER_MO_FIN_SMRY = drop_duplicate_columns(NEW_SELLER_MO_FIN_SMRY)
```

```
['sell_id', 'subm_se_no', 'cstone_feed_key', 'cstone_last_updatetm', 'srce_sy
s_id', 'bus_ctr_cd', 'mo_id', 'subm_mer_id', 'smry_mo_dt', 'payee_se_no', 'pa
yee_mer_id', 'tpa_se_no', 'tpa_mer_id', 'rel_grp_type_no', 'rel_subgrp_type_c
d', 'chrg_vol_dr_usd_am', 'chrg_vol_dr_loc_am', 'chrg_vol_cr_usd_am', 'chrg_v
ol_cr_loc_am', 'chrg_vol_net_usd_am', 'chrg_vol_net_loc_am', 'chrg_vol_dr_tra
ns_ct', 'chrg_vol_cr_trans_ct', 'avg_trans_usd_am', 'avg_trans_loc_am', 'adj_
usd_am', 'adj_loc_am', 'disc_usd_am', 'disc_loc_am', 'fee_usd_am', 'fee_loc_a
m', 'inq_ct', 'inq_usd_am', 'inq_loc_am', 'chrgbk_ct', 'chrgbk_usd_am', 'chrg
bk_loc_am', 'chrgbk_rvs_ct', 'chrgbk_rvs_usd_am', 'chrgbk_rvs_loc_am', 'creat
_ts', 'lst_updt_ts', 'sell_se_no', 'settle_curr_cd', 'subm_curr_cd', 'fin_c
d', 'fee_tax_usd_am', 'fee_tax_loc_am', 'disc_tax_usd_am', 'disc_tax_loc_am',
'plan_n_subm_in', 'mer_prop_in']
```

In [260]:

```
NEW_SELLER_MO_FIN_SMRY = NEW_SELLER_MO_FIN_SMRY.sort(["SUBM_SE_NO","SELL_ID"])
```

In [261]:

```
seller_temp1 = drop_duplicate_columns(seller_temp1)
```

```
['mer_id', 'se_no', 'subm_se10', 'payee_se10', 'tpa_se10', 'mtch_tpa_se10',
'mtch_sell_id', 'mtch_sell_subm_se10', 'mer_srce_sys_cd', 'dba_nm', 'legal_n
m', 'city_nm', 'rel_grp_type_no', 'rel_grp_type_updt_dt', 'rel_subgrp_type_c
d', 'phys_ad_ctry_cd', 'phys_ad_post_cd_tx', 'seims_indus_ds_cd', 'mer_setup_
dt', 'mer_act_dt', 'phone_no', 'mcc_indus_cd', 'phys_ad_rgn_area_cd', 'ad_lin
e_1_tx', 'ad_line_2_tx', 'ad_line_3_tx', 'email_ad_tx', 'creat_ts', 'creat_ts
_1', 'sell_url', 'init_rec_srce_cd', 'iso_regis_no', 'mkt_elig_cd', 'srvc_bus
_ctr_cd', 'lang_pref_cd', 'sale_chan_cd', 'sale_chan_nm', 'subm_type_cd', 'sa
le_repr_id', 'sell_acpt_sta_updt_dt', 'cur_mer_sta_cd', 'sell_acpt_sta_cd',
'bank_card_act_in', 'jcb_acpt_in', 'sgfnt_own_first_nm', 'sgfnt_own_lst_nm',
'sign_date', 'se_type', 'open_year', 'open_month', 'subm_year', 'subm_month',
'mer_tenure_mo_ct', 'activated_per']
,creat_ts_1
```

In [262]:

```
seller_temp1.createOrReplaceTempView("seller_temp1")
NEW_SELLER_MO_FIN_SMRY.createOrReplaceTempView("NEW_SELLER_MO_FIN_SMRY")
```

In [263]:

```
seller_cv_smry = sqlContext.sql("""
select a.*,
min(case when CHRG_VOL_NET_USD_AM>0.000005 and CHRG_VOL_DR_TRANS_CT>=1 then mo_id end) over(partition by a.subm_se10, a.mer_id) first_cv_mo_id,
sum(case when mo_id>=cast('{yr_ly3}01' as int) and mo_id<=cast('{yr_ly3}12' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) ly3_cv_fy,
sum(case when mo_id>=cast('{yr_ly2}01' as int) and mo_id<=cast('{yr_ly2}12' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) ly2_cv_fy,
sum(case when mo_id>=cast('{yr_ly}01' as int) and mo_id<=cast('{yr_ly}12' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) ly_cv_fy,
sum(case when MO_ID = cast('{last_r12_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end ) over(partition by a.subm_se10, a.mer_id) LY_CV_CurM,
sum(case when mo_id>=(cast('{cur_year_begin}' as int )-100) and mo_id <=(cast('{cur_month}' as int)-100) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) LY_CV_YTD,


sum(case when mo_id>=(cast('{r12_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) LY_CV_R12,
sum(case when mo_id>=(cast('{r3_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) LY_CV_R3,
sum(case when mo_id>=(cast('{r6_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) LY_CV_R6,

sum(Case when MO_ID = cast('{cur_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CV_curm,
sum(case when mo_id>=cast('{cur_year_begin}' as int ) and mo_id<= cast('{cur_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CV_YTD,



sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CV_R12,
sum(case when mo_id>=cast('{r3_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CV_R3,
sum(case when mo_id>=cast('{r6_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_NET_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CV_R6,

sum(case when mo_id>=cast('{yr_ly3}01' as int) and mo_id<=cast('{yr_ly3}12' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_ly3,
sum(case when mo_id>=cast('{yr_ly3}01' as int) and mo_id<=cast('{yr_ly3}12' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_ly3,
sum(case when mo_id>=cast('{yr_ly2}01' as int) and mo_id<=cast('{yr_ly2}12' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_ly2,
sum(case when mo_id>=cast('{yr_ly2}01' as int) and mo_id<=cast('{yr_ly2}12' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_ly2,
sum(case when mo_id>=cast('{yr_ly}01' as int) and mo_id<=cast('{yr_ly}12' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_ly,
sum(case when mo_id>=cast('{yr_ly}01' as int) and mo_id<=cast('{yr_ly}12' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_ly,
sum(Case when MO_ID = cast('{last_r12_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_LY_CurM,
sum(Case when MO_ID = cast('{last_r12_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_LY_CurM,
sum(case when mo_id>=(cast('{cur_year_begin}' as int )-100) and mo_id <=(cast('{cur_month}' as int)-100) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_CR_ROC_YTD,
sum(case when mo_id>=(cast('{cur_year_begin}' as int )-100) and mo_id <=(cast('{cur_month}' as int)-100) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_DR_ROC_YTD,



sum(case when mo_id>=(cast('{r12_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_CR_ROC_R12,
sum(case when mo_id>=(cast('{r6_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_CR_ROC_R6,
sum(case when mo_id>=(cast('{r3_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_CR_ROC_R3,

sum(case when mo_id>=(cast('{r12_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_DR_ROC_R12,
sum(case when mo_id>=(cast('{r6_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_DR_ROC_R6,
sum(case when mo_id>=(cast('{r3_month}' as int)-100) and mo_id<=(cast('{cur_month}' as int)-100) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) YOY_DR_ROC_R3,

sum(Case when MO_ID = cast('{cur_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_curm,
sum(Case when MO_ID = cast('{cur_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_curm,
sum(case when mo_id>=cast('{cur_year_begin}' as int ) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_YTD,
sum(case when mo_id>=cast('{cur_year_begin}' as int ) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_YTD,



sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_R12,
sum(case when mo_id>=cast('{r3_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_R3,
sum(case when mo_id>=cast('{r6_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_CR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CR_ROC_R6,

sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_R12,
sum(case when mo_id>=cast('{r3_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_R3,
sum(case when mo_id>=cast('{r6_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRG_VOL_DR_TRANS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) DR_ROC_R6,
sum(Case when MO_ID = cast('{cur_month}' as int) then DISC_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) DCN_REV_curM,
sum(Case when MO_ID = cast('{last_r12_month}' as int) then DISC_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) DCN_REV_LY_CurM,
sum(case when mo_id>=cast('{cur_year_begin}' as int ) and mo_id<=cast('{cur_month}' as int) then DISC_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) DCN_REV_YTD,
sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then DISC_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) DCN_REV_R12,
sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRGBK_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CHRGBK_CT_R12,
sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRGBK_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CHRGBK_USD_AM_R12,
sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRGBK_RVS_CT else 0 end) over(partition by a.subm_se10, a.mer_id) CHRGBK_RVS_CT_R12,
sum(case when mo_id>=cast('{r12_month}' as int) and mo_id<=cast('{cur_month}' as int) then CHRGBK_RVS_USD_AM else 0 end) over(partition by a.subm_se10, a.mer_id) CHRGBK_RVS_USD_AM_R12
from seller_temp1 as a
left join NEW_SELLER_MO_FIN_SMRY as b
on a.mer_id=b.sell_id
and a.SUBM_SE10 = b.SUBM_SE_NO
""".format(yr_ly3 = yr_ly3,yr_ly2=yr_ly2,yr_ly=yr_ly,cur_year_begin = cur_year_begin, cur_month = cur_month, r12_month=r12_month, r3_month=r3_month,r6_month=r6_month, last_r12_month=last_r12_month ))
```

In [264]:

```
seller_cv_smry = seller_cv_smry.withColumn("FIRST_CV_MO_ID",col("FIRST_CV_MO_ID").cast(StringType()))
```

In [265]:

```
seller_cv_smry = seller_cv_smry.withColumn('FIRST_CV_MO_ID',to_date(seller_cv_smry.FIRST_CV_MO_ID, 'yyyy-MM-dd'))
```

In [266]:

```
from pyspark.sql.types import IntegerType
yr_ly = DATE.year-1
yr_cy = (dt.date.today().replace(day=1) + relativedelta(months=0,days=-1)).year
seller_cv_smry = seller_cv_smry.withColumn("ROC_YTD", col("DR_ROC_YTD")-col("CR_ROC_YTD"))
\
            .withColumn("ROC_curm", col("DR_ROC_curm")-col("CR_ROC_curm")) \
            .withColumn("LY_ROC_R12", col("YOY_DR_ROC_R12")-col("YOY_CR_ROC_R12")) \
            .withColumn("LY_ROC_R3", col("YOY_DR_ROC_R3")-col("YOY_CR_ROC_R3")) \
            .withColumn("LY_ROC_R6",col("YOY_DR_ROC_R6")-col("YOY_CR_ROC_R6")) \
            .withColumn("ROC_R12",col("DR_ROC_R12")-col("CR_ROC_R12")) \
            .withColumn("ROC_R6",col("DR_ROC_R6")-col("CR_ROC_R6")) \
            .withColumn("ROC_R3",col("DR_ROC_R3")-col("CR_ROC_R3")) \
            .withColumn("LY_ROC_YTD",col("YOY_DR_ROC_YTD")-col("CR_ROC_R3")) \
            .withColumn("LY_ROC_CURM",col("DR_ROC_LY_CurM")-col("CR_ROC_LY_CurM")) \
            .withColumn("LY_ROC_FY",col("DR_ROC_ly")-col("CR_ROC_ly")) \
            .withColumn("LY2_ROC_FY",col("DR_ROC_ly2")-col("CR_ROC_ly2")) \
            .withColumn("LY3_ROC_FY",col("DR_ROC_ly3")-col("CR_ROC_ly3")) \
            .withColumn("BCV_OPEN_DATE", when(col("FIRST_CV_MO_ID") !='', 'FIRST_CV_MO_ID'
)) \
            .withColumn("FIRST_CV_MO_ID1", when(col("FIRST_CV_MO_ID") !='',(expr("concat(c
ast(FIRST_CV_MO_ID as string),'01')")))) \
            .withColumn("BCV_END_DATE", when(col("FIRST_CV_MO_ID") !='',(expr("add_months
(FIRST_CV_MO_ID1,12)")))) \
            .withColumn("BCV_END_MONTH",when(col("FIRST_CV_MO_ID") !='',(expr("mod(cast(BC
V_END_DATE as INT), 100)")))) \
            .withColumn("BCV_END_YEAR",when(col("FIRST_CV_MO_ID") !='',(expr("cast(BCV_END
_DATE as INT)/100 ")))) \
            .withColumn("BCV_END_MOID", when(col("FIRST_CV_MO_ID") !='', 'BCV_END_DATE'))
 \
            .withColumn("YTD_ALIF_BCV", when((col("FIRST_CV_MO_ID").cast(IntegerType())/10
0 == yr_cy) & (col("mer_setup_dt")<= col("FIRST_CV_MO_ID")) & (col("FIRST_CV_MO_ID") < exp
r("add_months(mer_setup_dt,12)")),1).otherwise(0)) \
            .withColumn("CARRY_OVER_BCV",when((col("FIRST_CV_MO_ID").cast(IntegerType())/1
00 == yr_ly) & (col("MER_SETUP_DT")<= col("FIRST_CV_MO_ID")) & (col("FIRST_CV_MO_ID") < ex
pr("add_months(MER_SETUP_DT,12)")) \
                                              & (0<=expr("ceil(abs(months_between(concat(c
ast(FIRST_CV_MO_ID as string),'01'), '{month_end}')))".format(month_end=str(month_end))))
& (expr("ceil(abs(months_between(concat(cast(FIRST_CV_MO_ID as string),'01'), '{month_end}
')))".format(month_end=str(month_end)))<13),1).otherwise(0)) \
            .drop("CR_ROC_ly3", "DR_ROC_ly3", "CR_ROC_ly2", "DR_ROC_ly2", "CR_ROC_ly", "DR
_ROC_ly", "CR_ROC_LY_CurM", "DR_ROC_LY_CurM", "YOY_CR_ROC_YTD", "YOY_DR_ROC_YTD", "YOY_CR_
ROC_R12", "YOY_CR_ROC_R6", "YOY_CR_ROC_R3", "YOY_DR_ROC_R12", "YOY_DR_ROC_R6", "YOY_DR_ROC
_R3", "CR_ROC_curm", "DR_ROC_curm", "CR_ROC_YTD", "DR_ROC_YTD", "CR_ROC_R12", "CR_ROC_R6",
"CR_ROC_R3")
```

In [267]:

```
seller_cv_smry = seller_cv_smry.distinct().sort("mer_id")
```

In [268]:

```
seller_cv_smry.createOrReplaceTempView("seller_cv_smry")
# all_mer_cv2.createOrReplaceTempView("all_mer_cv2")
```

In [269]:

```
crt_postal_geo_summary.createOrReplaceTempView("crt_postal_geo_summary")
```

In [270]:

```
ob_all = sqlContext.sql("""
select m.*,
(case when mer_tenure_mo_ct <=11 then "<1yr"
when mer_tenure_mo_ct >=12 and mer_tenure_mo_ct <=23 then "1-2yrs"
else "2yrs+"
end) as mer_tenure_per,
geo.DMA_MKT_AREA_ID,
dma.DMA_DS_TX,

(case when DR_ROC_R12>=1 and cv_r12>0.000005 then 1 else 0 end) as cv_active_ind,
(case when cv_r12>=50 then 1 else 0 end) as cv_active_50,
case when cv_r12<=0 then '01.<=0'
when cv_r12>0 and cv_r12 < 50 then  '02.$1-<$50'
when cv_r12 >= 50 and cv_r12 <250 then '03.$50-<250'
when cv_r12 >= 250 and cv_r12 <500 then '04.$250-<500 '
when cv_r12 >= 500 and cv_r12 <1000 then '05.$500-<1K'
when cv_r12 >= 1000 and cv_r12 <2000 then  '06.$1K-<2K'
when cv_r12 >= 2000 and cv_r12 <3000 then  '07.$2K-<3K'
when cv_r12 >= 3000 and cv_r12 <5000 then  '08.$3K-<5K'
when cv_r12 >= 5000 and cv_r12 <10000 then '09.$5K-<10K'
when cv_r12 >= 10000 and cv_r12 < 15000 then '10.$10K-<15K'
when cv_r12 >= 15000 and cv_r12 < 25000 then  '11.$15K-<25K'
when cv_r12 >= 25000 and cv_r12 < 50000 then  '12.$25K-<50K'
when cv_r12 >= 50000 and cv_r12 < 100000 then  '13.$50K-<100K'
when cv_r12 >= 100000 and cv_r12 < 250000 then  '14.$100K-<250K'
when cv_r12 >= 250000 and cv_r12 < 500000 then  '15.$250K-<500K'
when cv_r12 >= 500000 and cv_r12 < 750000 then   '16.$500K-<750K'
when cv_r12 >= 750000 and cv_r12 < 1000000 then  '17.$750K-<1MM'
when cv_r12 >= 1000000 and cv_r12 < 2000000 then '18.$1MM-<2MM'
when cv_r12 >= 2000000 and cv_r12 < 3000000 then '19.$2MM-<3MM'
when cv_r12 >= 3000000 and cv_r12 < 10000000 then '20.$3MM-<10MM'
when cv_r12 >= 10000000 then '21.$10MM+'
end as cur_cv_r12_band,

case when cv_r6<=0 then                                                 '01.<=0'
when cv_r6>0 and cv_r6 < 50 then                                '02.$1-<$50'
when cv_r6 >= 50 and cv_r6 <250 then                    '03.$50-<250'
when cv_r6 >= 250 and cv_r6 <500 then                   '04.$250-<500'
when cv_r6 >= 500 and cv_r6 <1000 then                  '05.$500-<1K'
when cv_r6 >= 1000 and cv_r6 <2000 then                 '06.$1K-<2K'
when cv_r6 >= 2000 and cv_r6 <3000 then                 '07.$2K-<3K'
when cv_r6 >= 3000 and cv_r6 <5000 then                 '08.$3K-<5K'
when cv_r6 >= 5000 and cv_r6 <10000 then                        '09.$5K-<10K'
when cv_r6 >= 10000 and cv_r6 < 15000 then              '10.$10K-<15K'
when cv_r6 >= 15000 and cv_r6 < 25000 then              '11.$15K-<25K'
when cv_r6 >= 25000 and cv_r6 < 50000 then              '12.$25K-<50K'
when cv_r6 >= 50000 and cv_r6 < 100000 then             '13.$50K-<100K'
when cv_r6 >= 100000 and cv_r6 < 250000 then            '14.$100K-<250K'
when cv_r6 >= 250000 and cv_r6 < 500000 then            '15.$250K-<500K'
when cv_r6 >= 500000 and cv_r6 < 750000 then            '16.$500K-<750K'
when cv_r6 >= 750000 and cv_r6 < 1000000 then   '17.$750K-<1MM'
when cv_r6 >= 1000000 and cv_r6 < 2000000 then  '18.$1MM-<2MM'
when cv_r6 >= 2000000 and cv_r6 < 3000000 then  '19.$2MM-<3MM'
when cv_r6 >= 3000000 and cv_r6 < 10000000 then         '20.$3MM-<10MM'
when cv_r6 >= 10000000 then                                                     '21.$10MM+
'
end as cur_cv_r6_band,

case when cv_r3<=0 then
'01.<=0'
when cv_r3>0 and cv_r3 < 50 then                               '02.$1-<$50'
when cv_r3 >= 50 and cv_r3 <250 then                    '03.$50-<250'
when cv_r3 >= 250 and cv_r3 <500 then                   '04.$250-<500'
when cv_r3 >= 500 and cv_r3 <1000 then                  '05.$500-<1K'
when cv_r3 >= 1000 and cv_r3 <2000 then                 '06.$1K-<2K'
when cv_r3 >= 2000 and cv_r3 <3000 then                 '07.$2K-<3K'
when cv_r3 >= 3000 and cv_r3 <5000 then                 '08.$3K-<5K'
when cv_r3 >= 5000 and cv_r3 <10000 then                        '09.$5K-<10K'
when cv_r3 >= 10000 and cv_r3 < 15000 then              '10.$10K-<15K'
when cv_r3 >= 15000 and cv_r3 < 25000 then              '11.$15K-<25K'
when cv_r3 >= 25000 and cv_r3 < 50000 then              '12.$25K-<50K'
when cv_r3 >= 50000 and cv_r3 < 100000 then             '13.$50K-<100K'
when cv_r3 >= 100000 and cv_r3 < 250000 then            '14.$100K-<250K'
when cv_r3 >= 250000 and cv_r3 < 500000 then            '15.$250K-<500K'
when cv_r3 >= 500000 and cv_r3 < 750000 then            '16.$500K-<750K'
when cv_r3 >= 750000 and cv_r3 < 1000000 then   '17.$750K-<1MM  '
when cv_r3 >= 1000000 and cv_r3 < 2000000 then  '18.$1MM-<2MM   '
when cv_r3 >= 2000000 and cv_r3 < 3000000 then  '19.$2MM-<3MM   '
when cv_r3 >= 3000000 and cv_r3 < 10000000 then         '20.$3MM-<10MM  '
when cv_r3 >= 10000000 then                                                     '21.$10MM+
'
end as cur_cv_r3_band
from  seller_cv_smry as m
left join crt_postal_geo_summary as geo
on substr(PHYS_AD_POST_CD_TX,1,5) =substr(geo.post_cd,1,5)
left join AWB19.dma_mapping as dma
on geo.DMA_MKT_AREA_ID = dma.DMA_MKT_AREA_ID
""")
```

In [271]:

```
ob_all = ob_all.withColumnRenamed("DR_ROC_R12","DBROCS_R12")
```

In [272]:

```
ob_all.createOrReplaceTempView("ob_all")
```

In [273]:

```
ALL_MER_CV1 = drop_duplicate_columns(ALL_MER_CV1)
```

```
['partner_nm', 'portfolio_type', 'partner_typ', 'tpa_se_no', 'ob_consider_in
d', 'cntry_cd', 'lst_updt_dt', 'mcc_inus_cd', 'phys_ad_line_1_tx', 'phys_ad_c
try_cd', 'prim_lgcy_sale_srvc_id', 'srvc_sale_pers_id', 'phys_ad_rgn_area_c
d', 'b2b_ind_old', 'portfolio_type_1', 'dba_nm', 'cur_mer_sta_cd', 'subm_cpbl
_type_1_in', 'phys_ad_line_3_tx', 'cancel_year', 'srvc_bus_ctr_cd', 'mer_act_
dt', 'city_nm', 'mer_tenure_mo_ct', 'base_disc_rt', 'phys_ad_line_2_tx', 'mmd
b_match', 'acct_class_8_cd', 'prim_clnt_mgmt_div_nm', 'mer_hier_lvl_no', 'acc
t_class_6_cd', 'mer_id', 'legal_nm', 'dma_mkt_area_id', 'phys_ad_post_cd_tx',
'acct_class_4_cd', 'active_ind', 'open_year', 'srvc_prvd_cd', 'b2b_ind', 'mer
_tenure_per', 'cur_mer_canc_cd', 'mer_rein_rsn_cd', 'brnd_toc_mer_no', 'manag
ed_toc', 'cancel_month', 'cntrct_disc_rt', 'smsa_cd', 'brnd_toc_mer_srce_sys_
cd', 'open_month', 'partner_typ_1', 'brnd_toc_mer_id', 'top_of_chain_in', 'af
f_toc', 'se_no', 'mer_setup_dt', 'seims_indus_ds_cd', 'srvc_bus_ctr_nm', 'cur
_mer_canc_dt', 'mer_srce_sys_cd', 'mer_acct_orig_cd', 'gcg_in', 'creat_ts',
'disc_full_rt', 'mer_rein_dt', 'scf_cd', 'prim_clnt_mgr_full_nm', 'phone_no',
'index', 'ly3_cv_fy', 'ly2_cv_fy', 'ly_cv_fy', 'ly_cv_curm', 'ly_cv_ytd', 'ly
_cv_r12', 'ly_cv_r3', 'ly_cv_r6', 'cv_curm', 'cv_ytd', 'cv_r12', 'cv_r3', 'cv
_r6', 'ly3_roc_fy', 'ly2_roc_fy', 'ly_roc_fy', 'ly_roc_curm', 'ly_roc_ytd',
'ly_roc_r12', 'ly_roc_r3', 'ly_roc_r6', 'roc_curm', 'roc_ytd', 'roc_r12', 'ro
c_r3', 'roc_r6', 'dbrocs_r12', 'dcn_rev_curm', 'dcn_rev_ly_curm', 'dcn_rev_yt
d', 'dcn_rev_r12', 'ly_dcn_rev_r12', 'ly_dcn_rev_ytd', 'cv_active_ind', 'cur_
cv_r12_band', 'cur_cv_r3_band', 'cur_cv_r6_band']
,portfolio_type_1
,partner_typ_1
```

In [274]:

```
ALL_MER_CV1.createOrReplaceTempView("ALL_MER_CV1")
```

In [275]:

```
ob_output = sqlContext.sql("""
select m.*,
n.active_ind as mtch_se_active_ind,
n.ob_consider_ind as ob_consider_ind,
n.cv_active_ind as mtch_se_cv_active_ind,
n.srvc_prvd_cd  as mtch_se_srvc_prvd_cd,
n.partner_typ,
n.portfolio_type as mtch_se_portfolio_type,
n.aff_toc as mtch_se_aff_toc
from ob_all as m left join ALL_MER_CV1 as n
on m.MTCH_TPA_SE10=n.se_no
""")
```

In [276]:

```
ob_output.createOrReplaceTempView("ob_output")
```

In [277]:

```
ob_output2 = sqlContext.sql("""
select m.*,
(case when REL_GRP_TYPE_NO=3 then 'New Signings'
when REL_GRP_TYPE_NO=1 then 'Non Accepting'
when REL_GRP_TYPE_NO=2 then 'Accepting Amex, New to Partner'
when REL_GRP_TYPE_NO=4 then 'Amex Base Conversion'
else 'other'
end) as relation_type
from ob_output as m
""")
```

In [279]:

```
ob_output2 = ob_output2.withColumn("sell_acpt_sta_updt_dt",col("sell_acpt_sta_updt_dt").ca
st(StringType()))
```

In [280]:

```
ob_output2=ob_output2.withColumn('sell_acpt_sta_updt_dt',to_date(ob_output2.sell_acpt_sta_
updt_dt, 'yyyy-MM-dd'))
```

In [281]:

```
ob_output2.createOrReplaceTempView("ob_output2")
```

In [ ]:

```
ob_output3 = sqlContext.sql("""select * from ob_output2 where relation_type != 'other'
and (mer_setup_dt between ('2013-11-01') and '{month_end}') """.format(month_end=str(month
_end))).filter("""creat_ts > ('2018-01-01') and sell_acpt_sta_updt_dt = creat_ts and cur_m
er_sta_cd IN ('D','N')""").filter("cur_mer_sta_cd = 'E'")
```

In [ ]:

```
ob_output3.createOrReplaceTempView("ob_output3")
```

In [ ]:

```
delta4 = relativedelta(months=-1)
cur_month = (DATE + delta4).strftime("%Y%m")
prev_run_yr = cur_month[0:4]
prev_run_mn = cur_month[4:6]
print("prev_run_yr ",prev_run_yr," prev_run_mn ",prev_run_mn)
```

In [ ]:

```
# Need to put loop here
```

In [ ]:

```
# Need to put loop here for all mer seller
```

In [ ]:

```
mid_output=sqlContext.sql("""select * from AWB19.all_mer_seller_2022_09_17""")
mid_output=mid_output.filter(upper(col("se_type")).contains("SELL")).select("mer_id", "sub
m_se_no", "cancel_year", "cancel_month", "reinstate_year", "reinstate_month","OB_PRTR_NM")
\
        .withColumnRenamed("cancel_year","cnc_year_old").withColumnRenamed("cancel_month",
"cnc_mnth_old") \
        .withColumnRenamed("reinstate_year", "rein_year_old").withColumnRenamed("reinstate
_month", "rein_mnth_old")
```

In [ ]:

```
output3 = ob_output3.sort(["mer_id", "subm_se10"])
```

In [ ]:

```
mid_output = mid_output.sort(["mer_id", "subm_se_no"])
```

In [ ]:

```
output3.createOrReplaceTempView("output3")
mid_output.createOrReplaceTempView("mid_output")
```

In [ ]:

```
mid_ob_output = sqlContext.sql(""" select a.*, b.* from output3 a left join mid_output b o
n a.mer_id=b.mer_id
and a.SUBM_SE10 = b.SUBM_SE_NO""")
```

In [ ]:

```
from pyspark.sql.types import StringType
mid_ob_output = mid_ob_output.withColumn("SELL_ACPT_STA_UPDT_DT",col("SELL_ACPT_STA_UPDT_D
T").cast(StringType()))
```

In [ ]:

```
EX_CANDT = (dt.datetime.today().replace(day=1) + relativedelta(months=-2,days=-1)).strftim
e("%d%m%Y").upper()
print(EX_CANDT)
cxl_month=EX_CANDT
a= "SELL_ACPT_STA_UPDT_DT"
b = "01-09-2013"
c = "{cxl_month}".format(cxl_month=cxl_month)
d =  "year(SELL_ACPT_STA_UPDT_DT)"
e = "month(SELL_ACPT_STA_UPDT_DT)"
f = "{month_end}".format(month_end=month_end)
g = "{EX_CANDT}".format(EX_CANDT=EX_CANDT)
h = "year('{month_end}')".format(month_end=month_end)
ob_output4 = mid_ob_output.withColumn("active_ind", lit(1)).withColumn("active_ind", when
(((((expr(a)) > (expr(b))) & ((expr(a)) <= (expr(c))) | ((expr(a))==None))

& ((col("cur_mer_sta_cd")).isin('D','N')))

| ((col("cur_mer_sta_cd")).isin('T','I','E')),0)) \
                                                        .withColumn("cancel_year", when
(((((expr(a)) > (expr(b))) & ((expr(a)) <= (expr(c))) | ((expr(a))==None))

& ((col("cur_mer_sta_cd")).isin('D','N')))

| ((col("cur_mer_sta_cd")).isin('T','I','E')), expr(d)) \
                                                                .when((((expr(a)) > (expr(
b))) & ((expr(a) <= (expr(f))))) | ((col("cur_mer_sta_cd")).isin('R')), col("cnc_year_old"
))) \
                                                        .withColumn("cancel_month", when
(((((expr(a)) > (expr(b))) & ((expr(a)) <= (expr(c))) | ((expr(a))==None))

& ((col("cur_mer_sta_cd")).isin('D','N')))

| ((col("cur_mer_sta_cd")).isin('T','I','E')),expr(e)) \
                                                                       .when((((expr(a)) >
(expr(b))) & ((expr(a) <= (expr(f))))) | ((col("cur_mer_sta_cd")).isin('R')), \
                                                                                 col("cnc_
mnth_old"))) \
                                                        .withColumn("reinstate_year", when
(((((expr(a)) > (expr(b))) & ((expr(a)) <= (expr(c))) | ((expr(a))==None))

& ((col("cur_mer_sta_cd")).isin('D','N')))

| ((col("cur_mer_sta_cd")).isin('T','I','E')),col("rein_year_old")) \
                                                                       .when((((expr(a)) >
(expr(b))) & ((expr(a) <= (expr(f))))) | ((col("cur_mer_sta_cd")).isin('R')), \
                                                                                  expr(d
))) \
                                                        .withColumn("reinstate_month", whe
n(((((expr(a)) > (expr(b))) & ((expr(a)) <= (expr(c))) | ((expr(a))==None))

& ((col("cur_mer_sta_cd")).isin('D','N')))

| ((col("cur_mer_sta_cd")).isin('T','I','E')),col("rein_mnth_old")) \
                                                                       .when((((expr(a)) >
(expr(b))) & ((expr(a) <= (expr(f))))) | ((col("cur_mer_sta_cd")).isin('R')), \
                                                                                 expr(e)))
```

In [ ]:

```
ob_output4=ob_output4.withColumn("total_lif", lit(1)) \
          .withColumn("cv_alif", expr("""case when cv_active_ind=1 then 1 else 0 end"""))
 \
          .withColumn("cv50_alif", expr("""case when cv_active_50=1 then 1 else 0 end"""))
\
          .withColumn("onbook_alif50", expr("""case when active_ind=1 and cv_active_50=1 t
hen 1 else 0 end"""))
```

In [ ]:

```
ob_output4.createOrReplaceTempView("ob_output4")
ob_output6 = sqlContext.sql(""" select m.*,
(case when REL_GRP_TYPE_NO =3 then 'Ph1- Net New'
when REL_GRP_TYPE_NO =2 and mtch_se_srvc_prvd_cd>'0000' then 'Ph1- FldCnvOP'
when REL_GRP_TYPE_NO =2 and mtch_se_srvc_prvd_cd='0000' then 'Ph1- FldCnvPr'
when REL_GRP_TYPE_NO =2 and MTCH_TPA_SE10 is null and MTCH_SELL_ID is not null and MTCH_SE
LL_SUBM_SE10 is not null then 'Ph1- FldCnvOB'
when REL_GRP_TYPE_NO =2 then 'Ph1- FldCnvUNK'

when REL_GRP_TYPE_NO =1 then 'Ph2- PartnrNAB'

when REL_GRP_TYPE_NO = 4 AND REL_SUBGRP_TYPE_CD='A' then 'Ph3- OPBsCnvA'
when REL_GRP_TYPE_NO = 4 AND REL_SUBGRP_TYPE_CD='B' then 'Ph4- PrBsCnvB'
when REL_GRP_TYPE_NO = 4 AND REL_SUBGRP_TYPE_CD='C' then 'Ph4- PrBsCnvC'
when REL_GRP_TYPE_NO = 4 AND REL_SUBGRP_TYPE_CD='' then 'Ph4- UNKBsCnv'
end) as PHASE
from  ob_output4 as m""")
```

In [ ]:

```
ob_output_final = ob_output6.withColumn("dual_ALIF_ind", when((col("active_ind")==1) & \
                                                            (col("cv_active_ind")==1) & \
                                                            (col("mtch_se_active_ind")==1)
& \
                                                            (col("mtch_se_cv_active_ind")==
1),'Y').otherwise('N')) \
                        .withColumn("dual_LIF_ind", when((col("active_ind")==1) & (col("mt
ch_se_active_ind") ==1),'Y') \
                                   .when((col("active_ind")==0) \
                                        & (col("mtch_se_active_ind")==1) & (col("mtch_se_
cv_active_ind")==0) & \
                                        (expr("SELL_ACPT_STA_UPDT_DT") \
                                         >= expr("('01JUN2017') ")), 'Z').otherwise('N'))
\
                        .withColumn("dual_active_ind", when((col("dual_ALIF_ind")=="Y") &
(col("dual_LIF_ind") == "Y"),'Y') \
                                   .when((col("dual_ALIF_ind")=='N') & (col("dual_LIF_ind"
)=='Y') & (col("mtch_se_cv_active_ind")==1),'X') \
                                   .when((col("dual_ALIF_ind")=='N') & ((col("dual_LIF_in
d") == 'Y') | (col("dual_LIF_ind")=='Z')),'P') \
                                   .otherwise('N'))
```

In [ ]:

```
ob_output_final = ob_output_final.withColumn("onbook_lif", when((col("active_ind")==1) & (
col("dual_active_ind") != 'X'),1).otherwise(0)) \
                                    .withColumn("onbook_alif", when((col("active_ind")==1)
& (col("cv_active_ind")==1) & (col("dual_active_ind") != 'X'), 1).otherwise(0))
```

In [ ]:

```
ob_output_final = ob_output_final.sort(["se_no"]).dropDuplicates(["se_no"])
```

In [ ]:

```
ob_output_final.createOrReplaceTempView("ob_output_final")
```

In [ ]:

```
# ob_output_final.printSchema()
```

In [ ]:

```
all_mer_final = sqlContext.sql("""
select a.*, b.mtch_tpa_se10, b.dual_active_ind from
all_mer_cv1 as a left join ob_output_final as b on
a.se_no=b.mtch_tpa_se10""")
```

In [ ]:

```
all_mer_final = all_mer_final.sort(col("se_no").desc(), col("dual_active_ind").desc())
```

In [ ]:

```
all_mer_final = all_mer_final.sort(col("se_no")).dropDuplicates(["se_no"])
```

In [ ]:

```
all_mer_final_mid = all_mer_final.withColumn("onbook_lif", when((col("active_ind")==1) & \
                                                       (col("SUBM_CPBL_TYPE_1_IN"
) == 'Y') \
                                                       & (~col("Dual_active_ind")
.isin('Y' ,'P')),1).otherwise(0)) \
                                .withColumn("onbook_alif", when((col("active_ind")==1) \
                                                       &(col("cv_active_ind")==1)
\
                                                       &(col("SUBM_CPBL_TYPE_1_I
N")=='Y') \
                                                       &(~col("Dual_active_ind").
isin('Y' ,'P')),1).otherwise(0))
```

In [ ]:

```
all_mer_final1 = all_mer_final_mid.withColumn("id", col("mer_id").cast(StringType())).drop
("mer_id").withColumnRenamed("id", "mer_id")
```

In [ ]:

```
all_mer_sorted = all_mer_final1.sort(["se_no"]).dropDuplicates(["se_no"])
```

In [ ]:

```
ob_sorted = ob_output_final.sort(["se_no"]).dropDuplicates(["se_no"])
```

In [ ]:

```
ob_sorted.createOrReplaceTempView("ob_sorted")
all_mer_sorted.createOrReplaceTempView("all_mer_sorted")
```

In [ ]:

```
combined_iclic_info = sqlContext.sql("""select a.*,b.* ,
a.PHYS_AD_LINE_1_TX as line1,
a.PHYS_AD_LINE_2_TX as line2,
a.PHYS_AD_LINE_3_TX as line3,
b.ad_line_1_tx as PHYS_AD_LINE_1_TX,
b.ad_line_2_tx as PHYS_AD_LINE_2_TX,
b.ad_line_3_tx as PHYS_AD_LINE_3_TX,

AD_LINE_1_TX as line1,
AD_LINE_2_TX as line2,
AD_LINE_3_TX as line3
from all_mer_sorted as a full join ob_sorted as b
on a.se_no=b.se_no""")
```

In [ ]:

```
crmd_ind_base.createOrReplaceTempView("crmd_ind_base")
crmd_ind = sqlContext.sql("""
  select b.SE_NO,
         b.crmc,
         b.sic8
  from crmd_ind_base b""")
```

In [ ]:

```
crmd_ind = crmd_ind.sort("se_no").dropDuplicates(["se_no"])
```

In [ ]:

```
crmd_ind = crmd_ind.withColumn("sic1", substring('sic8', 1,1)).withColumn("sic2", substrin
g('sic8', 1,2)) \
                .withColumn("sic3", substring('sic8', 1,3)).withColumn("sic4", substring(
'sic8', 1,4)) \
                .withColumn("sic6", substring('sic8', 1,6)).withColumn("MF_Ind_Cd", expr(
"""case
when CRMC="1039" then "M0110"
when CRMC="1057" then "M0120"
when CRMC="0174" then "M0130"
when CRMC="1086" then "M0130"
when CRMC="1087" then "M0130"
when CRMC="1101" then "M0130"
when CRMC="1244" then "M0130"
when CRMC="1310" then "M0130"
when CRMC="1414" then "M0130"
when CRMC="1103" then "M0140"
when CRMC="1428" then "M0140"
when CRMC="1626" then "M0140"
when CRMC="0519" then "M0150"
when CRMC="0100" then "M0160"
when CRMC="0123" then "M0160"
when CRMC="0163" then "M0160"
when CRMC="1008" then "M0160"
when CRMC="1097" then "M0160"
when CRMC="1106" then "M0160"
when CRMC="1227" then "M0160"
when CRMC="1462" then "M0160"
when CRMC="1250" then "M0170"
when CRMC="1476" then "M0170"
when CRMC="0150" then "M0180"
when CRMC="0180" then "M0190"
when CRMC in (' ',"0000") then "M0200"
when CRMC="0106" then "M0200"
when CRMC="0166" then "M0200"
when CRMC="1112" then "M0200"
when CRMC="1165" then "M0200"
when CRMC="1224" then "M0200"
when CRMC="1257" then "M0200"
when CRMC="1319" then "M0200"
when CRMC="1679" then "M0200"
when CRMC="0195" then "M0210"
when CRMC="1185" then "M0210"
when CRMC="1348" then "M0210"
when CRMC="1660" then "M0210"
when CRMC="1661" then "M0210"
when CRMC="1663" then "M0210"
when CRMC="0951" then "M0220"
when SIC6="599912" then "M1230"
when SIC6="599911" then "M1270"
when SIC6="599906" then "M2260"
when SIC6="599913" then "M2270"
when SIC6="754903" then "M3230"
when SIC4="5813" then "M0230"
when SIC4="5411" then "M1110"
when SIC4="5541" then "M1140"
when SIC4="5621" then "M1150"
when SIC4="5651" then "M1170"
when SIC4="5921" then "M1210"
when SIC4="5992" then "M1220"
when SIC4="5961" then "M1250"
when SIC4="5211" then "M2110"
when SIC4="5511" then "M2130"
when SIC4="5531" then "M2150"
when SIC4="5712" then "M2180"
when SIC4="5932" then "M2220"
when SIC4="5941" then "M2230"
when SIC4="5944" then "M2240"
when SIC4="5947" then "M2250"
when SIC4="7011" then "M3110"
when SIC4="7216" then "M3130"
when SIC4="7231" then "M3160"
when SIC4="7241" then "M3170"
when SIC4="7538" then "M3200"
when SIC4="7542" then "M3220"
when SIC4="7911" then "M3280"
when SIC4="7933" then "M3280"
when SIC4="7991" then "M3280"
when SIC4="7992" then "M3280"
when SIC4="7997" then "M3280"
when SIC4="8111" then "M3380"
when SIC4="8011" then "M4110"
when SIC4="8021" then "M4120"
when SIC4="8041" then "M4130"
when SIC4="8042" then "M4140"
when SIC4="8721" then "M5170"
when SIC4="5047" then "M6120"
when SIC4="5085" then "M6150"
when SIC4="0742" then "M7110"
when SIC4="1711" then "M7150"
when SIC4="4121" then "M8110"
when SIC3="596" then "M1260"
when SIC3="571" then "M2190"
when SIC3="721" then "M3140"
when SIC3="753" then "M3210"
when SIC3="804" then "M4150"
when SIC3="504" then "M6130"
when SIC3="508" then "M6160"
when SIC3="542" then "M1120"
when SIC3="546" then "M1130"
when SIC3="549" then "M1131"
when SIC3="563" then "M1160"
when SIC3="566" then "M1180"
when SIC3="591" then "M1200"
when SIC3="594" then "M1201"
when SIC3="523" then "M2120"
when SIC3="525" then "M2120"
when SIC3="526" then "M2120"
when SIC3="552" then "M2140"
when SIC3="557" then "M2160"
when SIC3="572" then "M2200"
when SIC3="573" then "M2210"
when SIC3="722" then "M3150"
when SIC3="726" then "M3180"
when SIC3="792" then "M3270"
when SIC3="821" then "M3300"
when SIC3="822" then "M3310"
when SIC3="866" then "M3350"
when SIC3="803" then "M4150"
when SIC3="806" then "M4160"
when SIC3="731" then "M5110"
when SIC3="733" then "M5120"
when SIC3="734" then "M5130"
when SIC3="735" then "M5140"
when SIC3="737" then "M5150"
when SIC3="874" then "M5180"
when SIC3="503" then "M6110"
when SIC3="507" then "M6110"
when SIC3="506" then "M6140"
when SIC3="509" then "M6170"
when SIC3="078" then "M7120"
when SIC3="271" then "M7170"
when SIC3="272" then "M7170"
when SIC3="273" then "M7170"
when SIC3="274" then "M7170"
when SIC3="421" then "M8120"
when SIC3="422" then "M8130"
when SIC3="472" then "M8140"
when SIC3="481" then "M8150"
when SIC2="72" then "M3190"
when SIC2="75" then "M3240"
when SIC2="76" then "M3250"
when SIC2="78" then "M3260"
when SIC2="79" then "M3290"
when SIC2="86" then "M3390"
when SIC2="80" then "M4170"
when SIC2="73" then "M5160"
when SIC2="87" then "M5190"
when SIC2="17" then "M7160"
when SIC2="27" then "M7180"
when SIC2="83" then "M3330"
when SIC2="84" then "M3340"
when SIC2="51" then "M6190"
when SIC2="56" then "M1190"
when SIC2="53" then "M1240"
when SIC2="52" then "M1280"
when SIC2="54" then "M1280"
when SIC2="57" then "M1280"
when SIC2="59" then "M1280"
when SIC2="55" then "M2170"
when SIC2="70" then "M3120"
when SIC2="82" then "M3320"
when SIC2="64" then "M3360"
when SIC2="60" then "M3370"
when SIC2="61" then "M3370"
when SIC2="62" then "M3370"
when SIC2="63" then "M3370"
when SIC2="65" then "M3370"
when SIC2="67" then "M3370"
when SIC2="89" then "M5200"
when SIC2="50" then "M6180"
when SIC2="15" then "M7140"
when SIC2="16" then "M7140"
when SIC2="20" then "M7190"
when SIC2="21" then "M7190"
when SIC2="22" then "M7190"
when SIC2="23" then "M7190"
when SIC2="24" then "M7190"
when SIC2="25" then "M7190"
when SIC2="26" then "M7190"
when SIC2="28" then "M7190"
when SIC2="29" then "M7190"
when SIC2="30" then "M7190"
when SIC2="31" then "M7190"
when SIC2="32" then "M7190"
when SIC2="33" then "M7190"
when SIC2="34" then "M7190"
when SIC2="35" then "M7190"
when SIC2="36" then "M7190"
when SIC2="37" then "M7190"
when SIC2="38" then "M7190"
when SIC2="39" then "M7190"
when SIC2="40" then "M8160"
when SIC2="41" then "M8160"
when SIC2="42" then "M8160"
when SIC2="43" then "M8160"
when SIC2="44" then "M8160"
when SIC2="45" then "M8160"
when SIC2="46" then "M8160"
when SIC2="47" then "M8160"
when SIC2="48" then "M8170"
when SIC2="49" then "M8170"
when SIC2="91" then "M3400"
when SIC2="92" then "M3400"
when SIC2="93" then "M3400"
when SIC2="94" then "M3400"
when SIC2="95" then "M3400"
when SIC2="96" then "M3400"
when SIC2="97" then "M3400"
when SIC2="99" then "M3400"
when SIC2="01" then "M7130"
when SIC2="02" then "M7130"
when SIC2="07" then "M7130"
when SIC2="08" then "M7130"
when SIC2="09" then "M7130"
when SIC2="10" then "M7130"
when SIC2="12" then "M7130"
when SIC2="13" then "M7130"
when SIC2="14" then "M7130"
end""")).withColumn("INDUSTRY", expr("""
case
when substr(MF_Ind_Cd,1,2)='M0' then '1)Eating and Drinking Places'
when substr(MF_Ind_Cd,1,2)='M1' then '2)Retailers, Non-Durable and General Goods'
when substr(MF_Ind_Cd,1,2)='M2' then '3)Retailers, Durable Goods'
when substr(MF_Ind_Cd,1,2)='M3' or MF_Ind_Cd='M7170' then '4)Consumer and General Services'
when substr(MF_Ind_Cd,1,2)='M4' or MF_Ind_Cd='M7110' then '5)Health and Medical Services'
when substr(MF_Ind_Cd,1,2)='M5' or MF_Ind_Cd='M7180' then '6)Business Services'
when substr(MF_Ind_Cd,1,2)='M6' then '7)Wholesalers'
when substr(MF_Ind_Cd,1,2)='M7' then '8)Manufacturing, Construction, Agriculture and Mining'
when substr(MF_Ind_Cd,1,2)='M8' then '9)Transportation, Communications, Public Utilities'
end"""))
```

In [ ]:

```
crmd_ind = crmd_ind.withColumn("SUBINDUSTRY", expr("""
case
when MF_Ind_Cd="M0110" then "Asian Restaurant"
when MF_Ind_Cd="M0120" then "Bar & Grill Restaurant"
when MF_Ind_Cd="M0130" then "Casual Restaurant"
when MF_Ind_Cd="M0140" then "Caterers"
when MF_Ind_Cd="M0150" then "Delicatessen"
when MF_Ind_Cd="M0160" then "Ethnic Cuisine, Other"
when MF_Ind_Cd="M0170" then "Fast Food Restaurant"
when MF_Ind_Cd="M0180" then "Mexican Restaurant"
when MF_Ind_Cd="M0190" then "Pizza Restaurant"
when MF_Ind_Cd="M0200" then "Specialty Restaurant"
when MF_Ind_Cd="M0210" then "Specialty Snack/Beverage Bar"
when MF_Ind_Cd="M0220" then "Varied Menu Restaurant"
when MF_Ind_Cd="M0230" then "Drinking Places (Alcoholic Beverages)"
when mf_ind_cd="M3360" then "Finance, Insurance and Real Estate"
when mf_ind_cd="M3370" then "Finance, Insurance and Real Estate"
when mf_ind_cd="M3400" then "Public Administration"
when sic2 in ('07','80') then "Health and Medical Services"
when sic2="27" then "Printing, Publishing, And Allied Industries"
when sic2="50" then "Wholesale Trade-durable Goods"
when sic2="51" then "Wholesale Trade-non-durable Goods"
when sic2="52" then "Building Materials, Hardware, Garden Supply, And Mobile Home Dealer"
when sic2="53" then "General Merchandise Stores"
when sic2="54" then "Food Stores"
when sic2="55" then "Automotive Dealers And Gasoline Service Stations"
when sic2="56" then "Apparel And Accessory Stores"
when sic2="57" then "Home Furniture, Furnishings, And Equipment Stores"
when sic2="59" then "Miscellaneous Retail"
when sic2="70" then "Hotels, Rooming Houses, Camps, And Other Lodging Places"
when sic2="72" then "Personal Services"
when sic2="73" then "Business Services"
when sic2="75" then "Automotive Repair, Services, And Parking"
when sic2="76" then "Miscellaneous Repair Services"
when sic2="78" then "Motion Pictures"
when sic2="79" then "Amusement And Recreation Services"
when sic2="81" then "Legal Services"
when sic2="82" then "Educational Services"
when sic2="83" then "Social Services"
when sic2="84" then "Museums, Art Galleries, And Botanical And Zoological Gardens"
when sic2="86" then "Membership Organizations"
when sic2="87" then "Engineering, Accounting, Research, Management, And Related Services"
when sic2="89" then "Miscellaneous Services"
when substr(MF_Ind_Cd,1,2)='M7' then 'Manufacturing, Construction, Agriculture and Mining'
when substr(MF_Ind_Cd,1,2)='M8' then 'Transportation, Communications, Public Utilities'
end"""))
```

In [ ]:

```
combined_iclic_info = drop_duplicate_columns(combined_iclic_info)
crmd_ind = drop_duplicate_columns(crmd_ind)
```

In [ ]:

```
crmd_ind.createOrReplaceTempView("crmd_ind")
```

In [ ]:

```
# table_isExist_drop("AWB19", "crmd_ind")
# sqlContext.sql("""create table awb19.crmd_ind as select * from crmd_ind""")
```

In [202]:

```
combined_iclic_info.createOrReplaceTempView("combined_iclic_info")
crmd_ind.createOrReplaceTempView("crmd_ind")
all_mer_seller2 = sqlContext.sql("""
select a.*,
case when a.MER_ACCT_ORIG_CD in ('06','07','08','12','17','20','22','37')
then 'Y' end as SMA_origin,
case when a.MER_ACCT_ORIG_CD in ('03','04','09','14','19','24','28','34',
'02','31','32','33' ) then "Prop"
when a.MER_ACCT_ORIG_CD in ('06','07','20','22') then "SMA-ESA"
when a.MER_ACCT_ORIG_CD in ('08') then "SMA-WTH PROP"
when a.MER_ACCT_ORIG_CD in ('12','17') then "SMA-WTH Other"
when a.MER_ACCT_ORIG_CD= '37' then "SMA-One Point"
when a.MER_ACCT_ORIG_CD= '38' then "SMA-OptBlue"
else "OTH"
end as ACQ_CHANNEL,
f.industry as INDUSTRY_CRMD,
f.subindustry as  SUBINDUSTRY_CRMD
from combined_iclic_info as a
left join crmd_ind as f
on a.se_no=f.se_no
""")
```

In [203]:

```
all_mer_seller2.createOrReplaceTempView("all_mer_seller2")
all_mer_seller3 = sqlContext.sql("""
Select
a.*,
c.industry as OptBlue_industry,
c.mcc_desc as OB_MCC_DESC
from all_mer_seller2 as a
left join model19.OptBlue_MCC_industry as c
on a.MCC_INDUS_CD = c.mcc_cd
""")
```

In [204]:

```
all_mer_seller4 = all_mer_seller3.sort("se_no").dropDuplicates(["se_no"])
```

In [205]:

```
all_mer_seller_new = all_mer_seller4.withColumnRenamed("DMA_DS_TX","DMA_DS_TX1")
```

In [ ]:

```
crmd_ind_base.createOrReplaceTempView("crmd_ind_base")
```

In [206]:

```
crmd_park_dtl_onln=sqlContext.sql("""select se_no, MER_ONLN_OFFLN_CD from crmd_ind_base"""
)
```

In [207]:

```
crmd_park_dtl_onln.createOrReplaceTempView("crmd_park_dtl_onln")
```

In [208]:

```
all_mer_seller_new.createOrReplaceTempView("all_mer_seller_new")
all_mer_seller_temp = sqlContext.sql("""
Select a.*,
(case when d.se_no is not null then 'Y' else 'N' end) as online_ind     ,
(case when a.DMA_MKT_AREA_ID is not null then dma.DMA_DS_TX else a.DMA_DS_TX1 end) as DMA_
DS_TX,
case
when GCG_IN ='Y' then "GCG"/*GCG Reconciliation*/
when PORTFOLIO_TYPE in ("CCLM") and replace(PRIM_CLNT_MGMT_DIV_NM, ' ','') in ("USSMALLMER
CHANTSCCLM") and PRIM_LGCY_SALE_SRVC_ID in ('06M0', '06MA', '06MB', '06MC') then "CCLM-CM
T"
when PORTFOLIO_TYPE in ("CCLM") and replace(PRIM_CLNT_MGMT_DIV_NM, ' ','') in ("USSMALLMER
CHANTSCCLM") then "CCLM-Managed"

when PORTFOLIO_TYPE in ("OTH_SM", "OTH_SM INTL","OTH_SM AMEX Internal") then "OTH_SM"
when PORTFOLIO_TYPE in ("OptBlue") then "OptBlue"
when PORTFOLIO_TYPE in ("AGGREGATORS") then "AGGREGATORS"
else PRIM_CLNT_MGMT_DIV_NM
end as sub_segment

from  all_mer_seller_new as a
left join  model19.CRMD_DTL as d
on a.se_no=d.se_no
and d.MER_ONLN_OFFLN_CD = '1'
                      left join
                          AWB19.dma_mapping as dma
                             on a.DMA_MKT_AREA_ID = dma.DMA_MKT_AREA_ID""")
```

In [209]:

```
a1 = all_mer_seller_temp.filter(~upper(col("se_type")).contains("SELL"))
```

In [210]:

```
a2 = all_mer_seller_temp.filter(upper(col("se_type")).contains("SELL")).drop("sub_segment"
)
```

In [211]:

```
a1.createOrReplaceTempView("a1")
a2.createOrReplaceTempView("a2")
a3 = sqlContext.sql("""
select distinct a.subm_se_no, b.sub_segment from
a2 as a left join a1 as b on
a.subm_se_no=b.se_no
""")
```

In [212]:

```
a3.createOrReplaceTempView("a3")
a4 = sqlContext.sql("""
select a.*, b.sub_segment from
a2 as a left join a3 as b on
a.subm_se_no=b.subm_se_no""")
```

In [213]:

```
a5 = a1.sort("se_no").dropDuplicates(["se_no"])
```

In [214]:

```
a4= a4.sort("se_no").dropDuplicates(["se_no"])
```

In [215]:

```
a4.createOrReplaceTempView("a4")
a5.createOrReplaceTempView("a5")
optblue_detail = sqlContext.sql("""
select a.*, b.mer_setup_dt as prop_dt, b.active_ind as prop_ind from
a4 as a left join a5 as b on
a.mtch_tpa_se_no=b.se_no""")
```

In [216]:

```
optblue_detail.createOrReplaceTempView("optblue_detail")
optblue_detail1 = sqlContext.sql("""
select *, case
when prop_ind=1 then {month_end}
when prop_ind=0 and active_ind=1 then {month_end}
when prop_ind=0 and active_ind=0 then concat(cancel_month,'-','01','-',cancel_year)
end as can_dt
from optblue_detail""".format(month_end=month_end))
```

In [217]:

```
optblue_detail2 = optblue_detail1.withColumn("tot_tenure", when(col("mtch_tpa_se_no")!='', \
                                                               expr("""ceil(abs(months_between(prop_dt, can_dt))""")))
```

In [218]:

```
opd = optblue_detail2.drop("prop_dt", "prop_ind", "can_dt").withColumn("total_tenure", max
(col("tot_tenure")).over(windowSpecAgg))
```

In [219]:

```
a5 = dropDuplicateColumns(a5)
opd = dropDuplicateColumns(opd)
```

In [220]:

```
prop = a5.alias("a").join(opd.alias("b"),col("a.se_no") == col("b.mtch_tpa_se_no"),"left")
.selectExpr("a.*","b.total_tenure")
```

In [221]:

```
prop1 = prop.withColumn("total_tenure",when((col("mtch_tpa_se_no")=="") &  \
                                            (col("active_ind")==1),expr("""ceil(abs(months
_between(mer_setup_dt,'{month_end}')))""".format(month_end=str(month_end)))) \
                        .when((col("mtch_tpa_se_no")=="") &  \
                                            (col("active_ind")==0), expr("""ceil(abs(month
s_between(mer_setup_dt, concat(cancel_month,'-','01','-',cancel_year)))""")))
```

In [222]:

```
opd = opd.sort("se_no").dropDuplicates(["se_no"])
```

In [223]:

```
prop1 = prop1.sort("se_no").dropDuplicates(["se_no"])
```

In [224]:

```
all_mer_seller6 = opd.join(prop1, ["se_no"], "full").drop("tot_tenure")
```

In [225]:

```
dtt = (DATE + relativedelta(months=-1)).strftime("%Y%m")
prev_dt= (DATE + relativedelta(months=-2)).strftime("%Y%m")
EX_CANDT = (dt.datetime.today().replace(day=1) + relativedelta(months=-2,days=-1)).strftim
e("%d%b%Y").upper()
CANDT = (dt.datetime.today().replace(day=1) + relativedelta(months=0,days=-1)).strftime("%
d%b%Y").upper()
SGDT = (dt.datetime.today().replace(day=1) + relativedelta(months=0,days=-1)).strftime("%d
%b%Y").upper()
r12start = (DATE + relativedelta(months=-12)).strftime("%Y%m")
r12end = (DATE + relativedelta(months=-1)).strftime("%Y%m")
print(dtt ,prev_dt, EX_CANDT, CANDT, SGDT, r12start, r12end)
```

```
202208 202207 30JUN2022 31AUG2022 31AUG2022 202109 202208
```

In [226]:

```
seller_vol = sqlContext.sql("""
select subm_se_no,sell_id as mer_id, chrg_vol_net_loc_am/100 as amt_{dtt},(chrg_vol_dr_tra
ns_ct - chrg_vol_cr_trans_ct) as cnt_{dtt},
chrg_vol_dr_trans_ct as drcnt_{dtt}
from NEW_SELLER_MO_FIN_SMRY
where mo_id = {dtt} and
(tpa_se_no in (select distinct tpa_Se_no from  NEW_SELLER_MO_FIN_SMRY) and
subm_Se_no in (select distinct subm_se_no from  NEW_SELLER_MO_FIN_SMRY))""".format(dtt=dtt
))
```

In [227]:

```
seller_vol.createOrReplaceTempView("seller_vol")
```

In [228]:

```
seller_vol=seller_vol.sort(["subm_se_no", "mer_id"])
```

In [229]:

```
seller_activation_temp = sqlContext.sql("select * from AWB19.seller_activation").sort("sub
m_se_no", "mer_id")
```

In [230]:

```
OLD_PRTR_INFO = OLD_PRTR_INFO.drop("CHNG_RSN_CD").withColumnRenamed("SELL_ID", "MER_ID").s
ort("subm_se_no", "mer_id")
```

In [231]:

```
NEW_PRTR_INFO = NEW_PRTR_INFO.drop("ORIG_SIGN_DATE", "ORIG_INIT_SUBM_DT").withColumnRename
d("SELL_ID", "MER_ID").sort("subm_se_no", "mer_id")
```

In [232]:

```
OLD_TO_NEW_SE_SELL_ACT = seller_activation_temp.join(OLD_PRTR_INFO, ["SUBM_SE_NO", "MER_I
D"], "inner") \
                                                .withColumn("SUBM_SE_NO", col("NEW_SUBM_SE
_NO")) \
                                                .withColumn("MER_ID", col("NEW_SELL_ID")).
drop("NEW_SUBM_SE_NO", "NEW_SELL_ID")
```

In [233]:

```
OLD_TO_NEW_SE_SELL_ACT = dropDuplicateColumns(OLD_TO_NEW_SE_SELL_ACT)
```

In [234]:

```
OLD_TO_NEW_SE_SELL_ACT = OLD_TO_NEW_SE_SELL_ACT.sort("SUBM_SE_NO", "MER_ID")
```

In [235]:

```
NEW_SE_SELL_ACT = seller_activation_temp.join(NEW_PRTR_INFO, ["SUBM_SE_NO", "MER_ID"], "inner")
```

In [236]:

```
NEW_SE_SELL_ACT = dropDuplicateColumns(NEW_SE_SELL_ACT)
```

In [237]:

```
NEW_SE_SELL_ACT = NEW_SE_SELL_ACT.sort("SUBM_SE_NO", "MER_ID")
```

In [240]:

```
FINAL_NEW_SE_SELL_ACT = dropDuplicateColumns(OLD_TO_NEW_SE_SELL_ACT.join(NEW_SE_SELL_ACT.w
ithColumnRenamed("PTD_CV_ACT_IND","NPTD_CV_ACT_IND").withColumnRenamed("ACT_DT","NACT_DT")
\
                                .withColumnRenamed("ACT_5_IND","NACT_5_IND").withColumnRen
amed("ACT_5_DT","NACT_5_DT") \
                                .withColumnRenamed("ACT_50_IND","NACT_50_IND").withColumnR
enamed("ACT_50_DT","NACT_50_DT") \
                                .withColumnRenamed("PTD_CHRG_AM","NPTD_CHRG_AM").withColum
nRenamed("PTD_CHRG_CT","NPTD_CHRG_CT") \
                                .withColumnRenamed("AMT_{dtt}".format(dtt=dtt),"AMT_{dtt}"
.format(dtt=dtt)) \
                                .withColumnRenamed("CNT_{dtt}".format(dtt=dtt),"CNT_{dtt}"
.format(dtt=dtt)) \
                                .withColumnRenamed("DRCNT_{dtt}".format(dtt=dtt),"DRCNT_{d
tt}".format(dtt=dtt)),["SUBM_SE_NO", "MER_ID"], "full")) \
                                                .withColumn("PTD_CV_ACT_IND", when(col("PT
D_CV_ACT_IND") > col("NPTD_CV_ACT_IND"),col("PTD_CV_ACT_IND")) \
                                                            .otherwise(col("NPTD_CV_ACT_IN
D"))) \
                                                .withColumn("ACT_DT", when((col("ACT_DT")
!= "")  \
                                                                           & (col("NACT_D
T") != "") \
                                                                           & (col("ACT_DT"
) < col("NACT_DT")), col("ACT_DT"))\
                                                            .when((col("ACT_DT") != "")  \
                                                                           & (col("NACT_D
T") != "") \
                                                                           & (col("ACT_DT"
) > col("NACT_DT")),col("NACT_DT")) \
                                                            .when((col("ACT_DT") != "")  \
                                                                           & (col("NACT_D
T") == ""),col("ACT_DT")).otherwise(col("NACT_DT"))) \
                                                .withColumn("ACT_5_IND", when(col("ACT_5_I
ND") > col("NACT_5_IND"),col("ACT_5_IND")) \
                                                            .otherwise(col("NACT_5_IND")))
\
                                                .withColumn("ACT_5_DT", when((col("NACT_5_
DT") != "")  \
                                                                           & (col("NACT_5_
DT") != "") \
                                                                           & (col("ACT_5_D
T") < col("NACT_5_DT")), col("ACT_5_DT"))\
                                                            .when((col("ACT_5_DT") != "")
\
                                                                           & (col("NACT_5_
DT") != "") \
                                                                           & (col("ACT_5_D
T") > col("NACT_5_DT")),col("NACT_5_DT")) \
                                                            .when((col("ACT_5_DT") != "")
\
                                                                           & (col("NACT_5_
DT") == ""),col("ACT_5_DT")).otherwise(col("NACT_5_DT"))) \
                                                .withColumn("ACT_50_IND", when(col("ACT_50
_IND") > col("NACT_50_IND"),col("ACT_50_IND")) \
                                                            .otherwise(col("NACT_50_IND"
))) \
                                                .withColumn("ACT_50_IND", when((col("NACT_
50_IND") != "")  \
                                                                           & (col("NACT_50
_IND") != "") \
                                                                           & (col("ACT_50_
IND") < col("NACT_50_IND")), col("ACT_50_IND"))\
                                                            .when((col("ACT_50_IND") != ""
)  \
                                                                           & (col("NACT_50
_IND") != "") \
                                                                           & (col("ACT_50_
IND") > col("NACT_50_IND")),col("NACT_5_DT")) \
                                                            .when((col("ACT_50_IND") != ""
)  \
                                                                           & (col("NACT_50
_IND") == ""),col("ACT_50_IND")).otherwise(col("NACT_50_IND"))) \
                                                .withColumn("PTD_CHRG_AM", col("PTD_CHRG_A
M")+col("NPTD_CHRG_AM")) \
                                                .withColumn("PTD_CHRG_CT", col("PTD_CHRG_C
T")+col("NPTD_CHRG_CT")) \
                                                .withColumn("AMT_{dtt}".format(dtt=dtt), \
                                                           col("AMT_{dtt}".format(dtt=dtt
))+col("AMT_{dtt}".format(dtt=dtt))) \
                                                .withColumn("CNT_{dtt}".format(dtt=dtt), \
                                                           col("CNT_{dtt}".format(dtt=dtt
))+col("CNT_{dtt}".format(dtt=dtt))) \
                                                .withColumn("DRCNT_{dtt}".format(dtt=dtt),
\
                                                           col("DRCNT_{dtt}".format(dtt=dt
t))+col("DRCNT_{dtt}".format(dtt=dtt))) \
                                                .drop("NPTD_CV_ACT_IND", "NACT_DT", "NACT_
5_IND", "NACT_5_DT", "NACT_50_IND", "NACT_50_DT", "NPTD_CHRG_AM", "NPTD_CHRG_CT", "NAMT_{d
tt}".format(dtt=dtt), "NCNT_{dtt}".format(dtt=dtt), "NDRCNT_{dtt}".format(dtt=dtt))
```

In [241]:

```
ACT_ORIG_NEW_INACT = OLD_TO_NEW_SE_SELL_ACT.join(NEW_SE_SELL_ACT.withColumnRenamed("PTD_CV
_ACT_IND","NPTD_CV_ACT_IND").withColumnRenamed("ACT_DT","NACT_DT") \
                                .withColumnRenamed("ACT_5_IND","NACT_5_IND").withColumnRen
amed("ACT_5_DT","NACT_5_DT") \
                                .withColumnRenamed("ACT_50_IND","NACT_50_IND").withColumnR
enamed("ACT_50_DT","NACT_50_DT") \
                                .withColumnRenamed("PTD_CHRG_AM","NPTD_CHRG_AM").withColum
nRenamed("PTD_CHRG_CT","NPTD_CHRG_CT") \
                                .withColumnRenamed("AMT_{dtt}".format(dtt=dtt),"AMT_{dtt}"
.format(dtt=dtt)) \
                                .withColumnRenamed("CNT_{dtt}".format(dtt=dtt),"CNT_{dtt}"
.format(dtt=dtt)) \
                                .withColumnRenamed("DRCNT_{dtt}".format(dtt=dtt),"DRCNT_{d
tt}".format(dtt=dtt)),["SUBM_SE_NO","MER_ID"], "full") \
                    .drop("NPTD_CV_ACT_IND", "NACT_DT", "NACT_5_IND", "NACT_5_DT", "NACT_5
0_IND", "NACT_50_DT", "NPTD_CHRG_AM", "NPTD_CHRG_CT", "NAMT_{dtt}".format(dtt=dtt), "NCNT_
{dtt}".format(dtt=dtt), "NDRCNT_{dtt}".format(dtt=dtt))
```

In [242]:

```
NEW_NOTIN_SELL_ACT = NEW_SE_SELL_ACT.withColumnRenamed("PTD_CV_ACT_IND","NPTD_CV_ACT_IND").withColumnRenamed("ACT_DT","NACT_DT") \
                                .withColumnRenamed("ACT_5_IND","NACT_5_IND").withColumnRenamed("ACT_5_DT","NACT_5_DT") \
                                .withColumnRenamed("ACT_50_IND","NACT_50_IND").withColumnRenamed("ACT_50_DT","NACT_50_DT") \
                                .withColumnRenamed("PTD_CHRG_AM","NPTD_CHRG_AM").withColumnRenamed("PTD_CHRG_CT","NPTD_CHRG_CT") \
                                .withColumnRenamed("AMT_{dtt}".format(dtt=dtt),"AMT_{dtt}".format(dtt=dtt)) \
                                .withColumnRenamed("CNT_{dtt}".format(dtt=dtt),"CNT_{dtt}".format(dtt=dtt)) \
                                .withColumnRenamed("DRCNT_{dtt}".format(dtt=dtt),"DRCNT_{dtt}".format(dtt=dtt)).join(OLD_TO_NEW_SE_SELL_ACT,["SUBM_SE_NO","MER_ID"], "full").select("SUBM_SE_NO","MER_ID")
```

In [243]:

```
FINAL_NEW_SE_SELL_ACT = FINAL_NEW_SE_SELL_ACT.sort("SUBM_SE_NO","MER_ID")
ACT_ORIG_NEW_INACT = ACT_ORIG_NEW_INACT.sort("SUBM_SE_NO","MER_ID")
NEW_NOTIN_SELL_ACT = NEW_NOTIN_SELL_ACT.sort("SUBM_SE_NO","MER_ID")
```

In [244]:

```
OTHER_SELL_ACT=seller_activation_temp.join(FINAL_NEW_SE_SELL_ACT.select("SUBM_SE_NO","MER_ID"),["SUBM_SE_NO","MER_ID"], "leftanti")
```

In [245]:

```
OTHER_SELL_ACT=OTHER_SELL_ACT.sort("SUBM_SE_NO","MER_ID")
```

In [247]:

```
ACT_ORIG_NEW_INACT = drop_duplicate_columns(ACT_ORIG_NEW_INACT)
```

In [249]:

```
SELLER_ACTIVATION1 = customUnion(customUnion(OTHER_SELL_ACT,FINAL_NEW_SE_SELL_ACT),ACT_ORIG_NEW_INACT)
```

In [250]:

```
SELLER_ACTIVATION1 = SELLER_ACTIVATION1.sort("SUBM_SE_NO","MER_ID")
```

In [251]:

```
seller_activation = SELLER_ACTIVATION1.select("subm_se_no", "mer_id", "ptd_cv_act_ind", "a
ct_5_ind", "act_50_ind", "ptd_chrg_am", "ptd_chrg_ct", "act_dt", "act_5_dt", "act_50_dt")
 \
                                        .join(seller_vol, ["subm_se_no","mer_id"],"right")
.withColumn("ptd_cv_act_ind", when((col("ptd_cv_act_ind")!=1) \

& (col("ptd_chrg_am")+col("amt_{dtt}".format(dtt=dtt))> 0.000005)  \

& (col("ptd_chrg_ct")+col("drcnt_{dtt}".format(dtt=dtt)) >=1),1)) \
                                                                                        .w
ithColumn("act_dt", when((col("ptd_cv_act_ind")!=1) \

& (col("ptd_chrg_am")+col("amt_{dtt}".format(dtt=dtt))> 0.000005)  \

& (col("ptd_chrg_ct")+col("drcnt_{dtt}".format(dtt=dtt)) >=1),dtt)) \
                                                                                        .w
ithColumn("act_5_ind", when((col("act_5_ind")!=1)&(col("ptd_chrg_am")+col("amt_{dtt}".form
at(dtt=dtt))>5),1)) \
                                                                                        .w
ithColumn("act_5_dt", when((col("act_5_ind")!=1)&(col("ptd_chrg_am")+col("amt_{dtt}".forma
t(dtt=dtt))>5),dtt)) \
                                                                                        .w
ithColumn("act_50_ind", when((col("act_50_ind")!=1)&(col("ptd_chrg_am")+col("amt_{dtt}".fo
rmat(dtt=dtt))>50),1)) \
                                                                                        .w
ithColumn("act_50_dt", when((col("act_50_ind")!=1)&(col("ptd_chrg_am")+col("amt_{dtt}".for
mat(dtt=dtt))>50),dtt)) \
                                                                                        .w
ithColumn("ptd_chrg_am",col("ptd_chrg_am")+col("amt_{dtt}".format(dtt=dtt))) \
                                                                                        .w
ithColumn("ptd_chrg_ct",col("ptd_chrg_ct")+col("cnt_{dtt}".format(dtt=dtt)))
```

In [252]:

```
seller_activation = seller_activation.sort("SUBM_SE_NO","MER_ID").dropDuplicates(["SUBM_SE
_NO","MER_ID"])
```

In [253]:

```
seller_activation = dropDuplicateColumns(seller_activation)
seller_activation.createOrReplaceTempView("seller_activation")
sqlContext.sql("create table if not exists AWB19.seller_activation as select * from seller
_activation")
```

Out[253]:

```
DataFrame[]
```

In [254]:

```
opd1 = opd.sort("SUBM_SE_NO","MER_ID").dropDuplicates(["SUBM_SE_NO","MER_ID"])
```

In [255]:

```
# EX_CANDT = (dt.datetime.today().replace(day=1) + relativedelta(months=-2,days=-1)).strftime("%d-%m-%Y").upper()
# print(EX_CANDT)
```

```
30-06-2022
```

In [257]:

```
optblue_seller_report = opd1.join(sqlContext.sql("select * from awb19.seller_activation"),
["SUBM_SE_NO","MER_ID"], "fullouter") \
                                .withColumn("act_months", when(col("act_dt") != "",expr(
"""ceil(abs(months_between(mer_setup_dt,substr(act_dt,5,2))))"""))) \
                                .withColumn("act_30", when((col("act_dt") != "") & (col("a
ct_months").isin(-1,0)),1)) \
                                .withColumn("act_60", when((col("act_dt") != "") & (col("a
ct_months").isin(-1,0,1)),1)) \
                                .withColumn("act_90", when((col("act_dt") != "") & (col("a
ct_months").isin(-1,0,1,2)),1)) \
                                .withColumn("act_120", when((col("act_dt") != "") & (col(
"act_months").isin(-1,0,1,2,3)),1)) \
                                .withColumn("act_150", when((col("act_dt") != "") & (col(
"act_months").isin(-1,0,1,2,3,4)),1)) \
                                .withColumn("act_180", when((col("act_dt") != "") & (col(
"act_months").isin(-1,0,1,2,3,4,5)),1)) \
                                .withColumn("act5_months", when(col("act_5_dt") != "",expr
("""ceil(abs(months_between(mer_setup_dt,substr(act_5_dt,5,2))))"""))) \
                                .withColumn("act5_30", when((col("act_5_dt") != "") & (col
("act5_months").isin(-1,0)),1)) \
                                .withColumn("act5_60", when((col("act_5_dt") != "") & (col
("act5_months").isin(-1,0,1)),1)) \
                                .withColumn("act5_90", when((col("act_5_dt") != "") & (col
("act5_months").isin(-1,0,1,2)),1)) \
                                .withColumn("act5_120", when((col("act_5_dt") != "") & (co
l("act5_months").isin(-1,0,1,2,3)),1)) \
                                .withColumn("act5_150", when((col("act_5_dt") != "") & (co
l("act5_months").isin(-1,0,1,2,3,4)),1)) \
                                .withColumn("act5_180", when((col("act_5_dt") != "") & (co
l("act5_months").isin(-1,0,1,2,3,4,5)),1)) \
                                .withColumn("act50_months", when(col("act_50_dt") != "",ex
pr("""ceil(abs(months_between(mer_setup_dt,substr(act_50_dt,5,2))))"""))) \
                                .withColumn("act50_30", when((col("act_50_dt") != "") & (c
ol("act50_months").isin(-1,0)),1)) \
                                .withColumn("act50_60", when((col("act_50_dt") != "") & (c
ol("act50_months").isin(-1,0,1)),1)) \
                                .withColumn("act50_90", when((col("act_50_dt") != "") & (c
ol("act50_months").isin(-1,0,1,2)),1)) \
                                .withColumn("act50_120", when((col("act_50_dt") != "") & (
col("act50_months").isin(-1,0,1,2,3)),1)) \
                                .withColumn("act50_150", when((col("act_50_dt") != "") & (
col("act50_months").isin(-1,0,1,2,3,4)),1)) \
                                .withColumn("act50_180", when((col("act_50_dt") != "") & (
col("act50_months").isin(-1,0,1,2,3,4,5)),1)) \
                                .withColumn("seller_status_exec", when((((expr("(SELL_ACPT
_STA_UPDT_DT)")) \
                                                                       < (expr("'{EX_CAND

T}'".format(EX_CANDT=EX_CANDT)))) & \
                                                                       (col("cur_mer_sta_c

d").isin('D','N'))) | \
                                                                       (col("cur_mer_sta_c

d").isin('T','I','E')),'C').otherwise('A')) \
                                 .withColumn("cur_mer_sta_cd", when((((expr("(SELL_ACPT_ST
A_UPDT_DT)")) \
                                                                       < (expr("'{CANDT}'"
.format(CANDT=CANDT)))) & \
                                                                        (col("cur_mer_sta_c
d").isin('D','N'))) | \
                                                                        (col("cur_mer_sta_c
d").isin('T','I','E')),'C').otherwise('A')).drop("amt_{dtt}".format(dtt=dtt), "cnt_{dtt}".
format(dtt=dtt), "drcnt_{dtt}".format(dtt=dtt))
```

In [258]:

```
# sqlContext.sql("""select * from AWB19.seller_hier_tpa""").printSchema()
```

In [259]:

```
optblue_seller_report.createOrReplaceTempView("optblue_seller_report")
optblue_seller_report=sqlContext.sql("""select b.*,a.*
from optblue_seller_report a left join AWB19.optblue_mcc_industry b
on a.MCC_INDUS_CD = b.mcc_cd""")
```

In [260]:

```
optblue_seller_report.createOrReplaceTempView("optblue_seller_report")
optblue_seller_report=sqlContext.sql("""select b.*,a.*
from optblue_seller_report a left join AWB19.seller_hier_tpa b
on a.tpa_se_no = b.tpa_se_no""")
```

In [261]:

```
optblue_seller_report = drop_duplicate_columns(optblue_seller_report)
```

In [262]:

```
optblue_seller_report.createOrReplaceTempView("optblue_seller_report")
```

In [263]:

```
optblue_seller_report=sqlContext.sql(""" select *, (case when trim(Partner_Typ)='' and sub
m_type_cd = 'T' then 'OPTBLUE' end) as PORTFOLIO_TYPE,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'T' then 'OPTBLUE TRADITIONAL' end) as partner_typ,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'B' then 'OPTBLUE' end) as PORTFOLIO_TYPE,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'B' then 'OPTBLUE PAYFAC' end) as partner_typ,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'C' then 'AGGREGATOR' end) as PORTFOLIO_TYPE,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'C' then 'AGGREGATOR REPORTED' end) as partner_typ,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'A' then 'AGGREGATOR' end) as PORTFOLIO_TYPE,
                                                   (case when trim(Partner_Typ)='' and sub
m_type_cd = 'A' then 'AGGREGATOR NOT-REPORTED' end) as partner_typ,

                                                   (case when subm_se_no in ('6569612666')
then 'WEPAYCHASE' end) as OB_PRTR_NM,
                                                   (case  when subm_se_no in ('656961266
6') then 'AGGREGATOR' end) as portfolio_type,
                                                   (case  when subm_se_no in ('656961266
6') then 'AGGREGATOR NOT-REPORTED' end) as partner_typ,
                                                   (case  when subm_se_no in ('656961266
6') then 4 end) as ob_consider_ind,

                                                   (case when  subm_se_no in ('656961262
5','6569612583','6569612641','6569612591','6569612609','6569612567','6569612575','65696126
33','6569612658')
                                                   then 'WEPAYCHASE' end) as OB_PRTR_NM,
                                                   (case when  subm_se_no in ('656961262
5','6569612583','6569612641','6569612591','6569612609','6569612567','6569612575','65696126
33','6569612658')
                                                   then 'AGGREGATOR' end) as portfolio_typ
e,
                                                   (case when  subm_se_no in ('656961262
5','6569612583','6569612641','6569612591','6569612609','6569612567','6569612575','65696126
33','6569612658')
                                                   then 'AGGREGATOR REPORTED' end) as part
ner_typ,
                                                   (case when  subm_se_no in ('656961262
5','6569612583','6569612641','6569612591','6569612609','6569612567','6569612575','65696126
33','6569612658')
                                                   then 3 end) as ob_consider_ind ,

                                                   (case when  subm_se_no in ('305694707
0','3056947054','3056947047','3056947039','3056947021','3056947013','3056947005','30569469
99','3056946981')
                                                   then 'WAVE INC' end) as OB_PRTR_NM,
                                                   (case when  subm_se_no in ('305694707
0','3056947054','3056947047','3056947039','3056947021','3056947013','3056947005','30569469
99','3056946981')
                                                   then 'AGGREGATOR' end) as portfolio_typ
e,
                                                   (case when  subm_se_no in ('305694707
0','3056947054','3056947047','3056947039','3056947021','3056947013','3056947005','30569469
99','3056946981')
                                                          then 'AGGREGATOR REPORTED' end) as part
ner_typ,
                                                          (case when  subm_se_no in ('305694707
0','3056947054','3056947047','3056947039','3056947021','3056947013','3056947005','30569469
99','3056946981')
                                                          then 3 end) as ob_consider_ind,



                                                           (case when  subm_se_no in ('554834665
9','1548293893','1548293885','1546508706','6542457700','6544840812','6544840861','65448408
53',
'6544840838','6544840820','6544840804','6544840796','6544840788','6544840770','654246294
0','6542460035',
'6542457692','6542457684','6542457676','6542457668','6542457627')
                                                          then 'WEPAY' end) as OB_PRTR_NM,
                                                          (case when  subm_se_no in ('554834665
9','1548293893','1548293885','1546508706','6542457700','6544840812','6544840861','65448408
53',
'6544840838','6544840820','6544840804','6544840796','6544840788','6544840770','654246294
0','6542460035',
'6542457692','6542457684','6542457676','6542457668','6542457627')
                                                          then 'AGGREGATOR' end) as portfolio_typ
e,
                                                          (case when  subm_se_no in ('554834665
9','1548293893','1548293885','1546508706','6542457700','6544840812','6544840861','65448408
53',
'6544840838','6544840820','6544840804','6544840796','6544840788','6544840770','654246294
0','6542460035',
'6542457692','6542457684','6542457676','6542457668','6542457627')
                                                          then 'AGGREGATOR NOT REPORTED' end) as
 partner_typ,
                                                          (case when  subm_se_no in ('554834665
9','1548293893','1548293885','1546508706','6542457700','6544840812','6544840861','65448408
53',
'6544840838','6544840820','6544840804','6544840796','6544840788','6544840770','654246294
0','6542460035',
'6542457692','6542457684','6542457676','6542457668','6542457627')
                                                          then 4 end) as ob_consider_ind,

                                                          (case when  subm_se_no in ('536191410
4','1213504780','3219653557','3219653540','3214523953','3214523862','3214523854','32145238
39',
'3214522831','3214522815','3214522773','3214522765','5368343919','3214523946')
                                                          then 'GLOBAL' end) as OB_PRTR_NM,
                                                          (case when  subm_se_no in ('536191410
4','1213504780','3219653557','3219653540','3214523953','3214523862','3214523854','32145238
39',
'3214522831','3214522815','3214522773','3214522765','5368343919','3214523946')
                                                          then 'AGGREGATOR' end) as portfolio_typ
e,
                                                          (case when  subm_se_no in ('536191410
4','1213504780','3219653557','3219653540','3214523953','3214523862','3214523854','32145238
39',
'3214522831','3214522815','3214522773','3214522765','5368343919','3214523946')
                                                          then 'AGGREGATOR REPORTED' end) as part
```

```
ner_typ,
                                                   (case when  subm_se_no in ('536191410
4','1213504780','3219653557','3219653540','3214523953','3214523862','3214523854','32145238
39',
'3214522831','3214522815','3214522773','3214522765','5368343919','3214523946')
                                                   then 3 end) as ob_consider_ind,

                                                   (case when  subm_se_no in ('143306409
4','1433064219','1433064177','1433064169','1433064151','1433064144','1433064136','14330641
28','1433064110','1433064102')
                                                   then 'PROPAY' end) as OB_PRTR_NM,
                                                   (case when  subm_se_no in ('143306409
4','1433064219','1433064177','1433064169','1433064151','1433064144','1433064136','14330641
28','1433064110','1433064102')
                                                   then 'OPTBLUE' end) as PORTFOLIO_TYPE,
                                                   (case when  subm_se_no in ('143306409
4','1433064219','1433064177','1433064169','1433064151','1433064144','1433064136','14330641
28','1433064110','1433064102')
                                                   then 'OPTBLUE PAYFAC' end) as partner_t
yp,



                                                   (case when  subm_se_no in ('105796912
3','1052648524','1052648516','1052648508','1052648490','1052487998','1052487980','10524879
64','1052487287','1052648532','1052487279',
'1052487261','1052487253','1052487246','2053159406','2053159356','2053159349','105248723
8','1052487220','1052487212','1052487204','1052487196',
'1052487188','1052487170','1052487162','1052487154','1052487147','1052487121','105248706
3','1052487105','1052487097','1052487089','1430745760',
'1430745604','1052487113','1341210805','1341210789','1058091232','1058091216','105809120
8','6561619339','1058091190','1058091182','1058091174',
'1341210748','1341210755')
                                                   then 'TRANSFIRST' end) as OB_PRTR_NM,
                                                   (case when  subm_se_no in ('105796912
3','1052648524','1052648516','1052648508','1052648490','1052487998','1052487980','10524879
64','1052487287','1052648532','1052487279',
'1052487261','1052487253','1052487246','2053159406','2053159356','2053159349','105248723
8','1052487220','1052487212','1052487204','1052487196',
'1052487188','1052487170','1052487162','1052487154','1052487147','1052487121','105248706
3','1052487105','1052487097','1052487089','1430745760',
'1430745604','1052487113','1341210805','1341210789','1058091232','1058091216','105809120
8','6561619339','1058091190','1058091182','1058091174',
'1341210748','1341210755')
                                                   then 'OPTBLUE' end) as PORTFOLIO_TYPE,
                                                   (case when  subm_se_no in ('105796912
3','1052648524','1052648516','1052648508','1052648490','1052487998','1052487980','10524879
64','1052487287','1052648532','1052487279',
'1052487261','1052487253','1052487246','2053159406','2053159356','2053159349','105248723
8','1052487220','1052487212','1052487204','1052487196',
'1052487188','1052487170','1052487162','1052487154','1052487147','1052487121','105248706
3','1052487105','1052487097','1052487089','1430745760',
'1430745604','1052487113','1341210805','1341210789','1058091232','1058091216','105809120
8','6561619339','1058091190','1058091182','1058091174',
'1341210748','1341210755')
                                                   then 'OPTBLUE TRADITIONAL' end) as part
ner_typ,
                                                   (case when  subm_se_no in ('207186188
4','2071050199','2071050181','2070896097','2070896089','2070896071','2070896063','20708960
55','2070896048','2070896030','2070896022','2070896014')
                                                   then 'EPX' end) as OB_PRTR_NM,

                                                   (case when  subm_se_no in ('207186188
4','2071050199','2071050181','2070896097','2070896089','2070896071','2070896063','20708960
55','2070896048','2070896030','2070896022','2070896014')
                                                   then 'AGGREGATOR' end) as PORTFOLIO_TYP
E,


                                                   (case when  subm_se_no in ('207186188
4','2071050199','2071050181','2070896097','2070896089','2070896071','2070896063','20708960
55','2070896048','2070896030','2070896022','2070896014')
                                                   then 'AGGREGATOR REPORTED' end) as part
ner_typ,

                                                   (case when  subm_se_no in ('207186188
4','2071050199','2071050181','2070896097','2070896089','2070896071','2070896063','20708960
55','2070896048','2070896030','2070896022','2070896014')
                                                   then 3 end) as ob_consider_ind,

                                                   (case when  subm_se_no in ('656961385
4','6569613813','6569613805','6569613797','6569613789','6569613771','6569613730','65696137
22','6569613714')
                                                   then 'PAYPAL PRO' end) as OB_PRTR_NM,

                                                   (case when  subm_se_no in ('656961385
4','6569613813','6569613805','6569613797','6569613789','6569613771','6569613730','65696137
22','6569613714')
                                                   then 'AGGREGATOR' end) as PORTFOLIO_TYP
E,


                                                   (case when  subm_se_no in ('656961385
4','6569613813','6569613805','6569613797','6569613789','6569613771','6569613730','65696137
22','6569613714')
                                                   then 'AGGREGATOR NOT-REPORTED' end) as
 partner_typ,

                                                   (case when  subm_se_no in ('656961385
4','6569613813','6569613805','6569613797','6569613789','6569613771','6569613730','65696137
22','6569613714')
                                                   then 4 end) as ob_consider_ind from opt
blue_seller_report """)
```

In [266]:

```
optblue_seller_report = optblue_seller_report.sort("se_no").dropDuplicates(["se_no"])
```

In [267]:

```
all_mer_seller6 = all_mer_seller6.sort("se_no").dropDuplicates(["se_no"])
```

In [268]:

```
all_mer_seller7 = optblue_seller_report.join(all_mer_seller6,["se_no"], "fullouter").filte
r(col("se_no") != "").drop("tot_tenure","DMA_DS_TX1")
```

In [269]:

```
all_mer_seller7 = dropDuplicateColumns(all_mer_seller7)
all_mer_seller7.createOrReplaceTempView("all_mer_seller7")
all_mer_seller8 = sqlContext.sql("""
Select m.*,
case when open_year > {yr_ly3} then 2
when cancel_year > {yr_ly3} then 1
else active_ind end as ly3_active_ind,
case when open_year > {yr_ly2} then 2
when cancel_year > {yr_ly2} then 1
else active_ind end as ly2_active_ind,
case when open_year > {yr_ly} then 2
when cancel_year > {yr_ly} then 1
else active_ind end as ly_active_ind
from all_mer_seller7 as m
where trim(se_no) > '' and replace(se_no, ' ','') <>'0'
""".format(yr_ly3=yr_ly3,yr_ly2=yr_ly2,yr_ly=yr_ly))
```

In [270]:

```
all_mer_seller8.createOrReplaceTempView("all_mer_seller8")
all_mer_seller_bfr =sqlContext.sql("""select * from all_mer_seller8""")
```

In [271]:

```
all_mer_seller_bfr.createOrReplaceTempView("all_mer_seller_bfr")
```

In [273]:

```
tpsp_agg=sqlContext.sql("""select distinct  concat("'", tpa_se_no, "'") from awb19.seller_
hier_tpa where portfolio_type='PROP-TPSP' """)
```

In [ ]:

```
# tpsp_agg.show()
```

In [276]:

```
TPSP=sqlContext.sql("""select * from cstonedb3.gmar_glbl_mer_dim where mer_no IN  ("{a}
")""".format(a=tpsp_agg))
```

In [277]:

```
TPSP.createOrReplaceTempView("TPSP")
```

In [278]:

```
MAP=sqlContext.sql("""select * from TPSP where MER_HIER_LVL_NO=8""")
```

In [279]:

```
CAP=sqlContext.sql("""select MER_ID,MER_NO from TPSP where MER_HIER_LVL_NO=6""")
```

In [280]:

```
SELL=sqlContext.sql("""select MER_ID,MER_NO from TPSP where MER_HIER_LVL_NO !=6 & 8""")
```

In [281]:

```
CAP.createOrReplaceTempView("CAP")
CAP= sqlContext.sql("""select *,  MER_NO as brnd_toc_mer_no from CAP""")
```

In [282]:

```
TPSP_CAPS=sqlContext.sql("""select * from cstonedb3.gmar_glbl_mer_dim where MER_HIER_LVL_N
O=6 AND brnd_toc_mer_no IN ("{tpsp_agg}")""".format(tpsp_agg=tpsp_agg))
```

In [283]:

```
TPSP_CAPS.createOrReplaceTempView("TPSP_CAPS")
TPSP_CAPS_LIF=sqlContext.sql("""SELECT A.MER_ID AS MER_ID,A.PRIM_CAP_MER_NO, A.MER_NO AS M
ER_NO, B.brnd_toc_mer_no
        FROM cstonedb3.mira_mer_affiliation_dimension_na AS A
        INNER JOIN TPSP_CAPS AS B
        ON A.PRIM_CAP_MER_NO=B.MER_NO""")
```

In [284]:

```
# CAP.printSchema()
```

In [285]:

```
CAP.createOrReplaceTempView("CAP")
CAPS_LIF=sqlContext.sql("""SELECT A.MER_ID AS MER_ID,A.PRIM_CAP_MER_NO, A.MER_NO AS MER_N
O, B.brnd_toc_mer_no
 FROM AWB19.mira_mer_affiliation_dimension_na AS A INNER JOIN CAP AS B ON A.PRIM_CAP_MER_N
O=B.MER_NO""")
```

In [286]:

```
MAP.createOrReplaceTempView("MAP")
MAP_LIF=sqlContext.sql("""SELECT B.MER_ID, B.MER_NO, B.brnd_toc_mer_no
        FROM MAP AS A
        LEFT JOIN cstonedb3.gmar_glbl_mer_dim AS B
        ON A.MER_NO = B.mkt_toc_mer_no or A.MER_NO = B.brnd_toc_mer_no""")
```

In [287]:

```
TPSP_2= customUnion(customUnion(customUnion(MAP_LIF,TPSP_CAPS_LIF), SELL),CAPS_LIF).filter
(col("MER_ID").isNull())
```

In [288]:

```
TPSP_2.sort(["MER_ID"]).dropDuplicates(["MER_ID"])
TPSP_2.createOrReplaceTempView("TPSP_2")
table_isExist_drop("AWB19","TPSP_2")
sqlContext.sql("""create table awb19.TPSP_2 as select * from TPSP_2""")
```

In [289]:

```
# TPSP_2_v2=sqlContext.sql("""select *, "Y" as tpsp_flag from awb19.TPSP_2""")
```

In [290]:

```
TPSP_2_v2.createOrReplaceTempView("TPSP_2_v2")
```

In [292]:

```
table_isExist_drop("AWB19","all_mer_seller_tpsp_flag")
all_mer_seller_with_tpsp_flag=sqlContext.sql("""create table awb19.all_mer_seller_tpsp_fla
g as select a.*,b.tpsp_flag from
all_mer_seller_bfr a left join TPSP_2_v2 b on a.se_no=b.mer_no""")
```

In [1]:

```
all_mer_seller=sqlContext.sql("""create table awb19.all_mer_seller as select *, case when
 tpsp_flag="Y" then 'PROP' end as portfolio_type
                                                when tpsp_flag="Y" then 'PROP-TPSP' end as
partner_typ  from awb19.all_mer_seller_with_tpsp_flag""").drop("tpsp_flag")
```

In [187]:

```
seller_hier_tpa=sqlContext.sql(""" select brnd_toc_mer_no, partner_typ, PORTFOLIO_TYPE, pa
rtner_nm, ob_consider_ind,tpa_se_no as brnd_toc_mer_no from awb19.seller_hier_tpa where ob
_consider_ind = '5' """)
```

In [188]:

```
seller_hier_tpa = seller_hier_tpa.sort(["brnd_toc_mer_no"]).dropDuplicates(["brnd_toc_mer_
no"])
```

In [2]:

```
all_mer_seller.createOrReplaceTempView("all_mer_seller")
all_mer_seller=sqlContext.sql("""create table awb19.all_mer_seller as select * from all_me
r_seller a left join seller_hier_tpa b on a.brnd_toc_mer_no=b.brnd_toc_mer_no""")
```

```
[NbConvertApp] Converting notebook model19putty.ipynb to python
[NbConvertApp] Writing 145756 bytes to model19putty.py
```

In [ ]:

```
```