from pyspark\_llap import HiveWarehouseSession

from pyspark.sql.functions import unix\_timestamp,from\_unixtime, to\_date

from pyspark.sql.functions import year, month, dayofmonth, quarter

from pyspark.sql.types import StringType

from pyspark.sql.functions import concat, col, lit

from pyspark.sql import functions as func

from pyspark.sql.functions import array, col, explode, struct, lit

hive = HiveWarehouseSession.session(spark).build()

hive.setDatabase("rnmpetromkt\_fleetmgmt")

card\_trans\_product = hive.executeQuery('select vartransid,varproductcode,intqty,varuom from rnmpetromkt\_fleetmgmt.tblmanualcardtransproducts where varproductcode = "512" or varproductcode = "1" or varproductcode = "32" or varproductcode = "29"')

manual\_card\_trans = hive.executeQuery('select \* from rnmpetromkt\_fleetmgmt.tblmanualcardtrans')

manual\_card\_trans = manual\_card\_trans.filter(manual\_card\_trans['createddate'] > '2016-01-01')

''' Inner Join between card\_trans\_product and manual\_card\_trans'''

inner\_join = card\_trans\_product.join(manual\_card\_trans, card\_trans\_product.vartransid == manual\_card\_trans.vartransid).select(card\_trans\_product.vartransid,manual\_card\_trans.createddate,manual\_card\_trans.intqty,manual\_card\_trans.decamount,manual\_card\_trans.varcardno,manual\_card\_trans.varmerchantno,manual\_card\_trans.varcarrierid)

df\_trans = inner\_join.select("vartransid","intqty","decamount","varcardno","varmerchantno","varcarrierid",year("createddate").alias('YearMonth'), month("createddate").alias('month'),quarter("createddate").alias('YearQuarter'),to\_date(('createddate')).alias('ts').cast("date")).sort(col("ts"))

df\_trans1 = df\_trans.withColumn("YearMonth", df\_trans["YearMonth"].cast(StringType()))

df\_trans1 = df\_trans1.withColumn("month", df\_trans["month"].cast(StringType()))

df\_trans1 = df\_trans1.withColumn("YearQuarter", df\_trans["YearQuarter"].cast(StringType()))

df\_trans1 = df\_trans1.select("vartransid","varcarrierid","intqty","decamount","varcardno","varmerchantno",concat(col("YearMonth"), lit(" "), col("month")).alias('ym'),concat(col("YearMonth"), lit(" "), col("YearQuarter")).alias('yq'),"ts" )

df\_trans2 = (df\_trans1.groupBy("varcarrierid","ym").agg(func.sum("decamount").alias('TotalAmount'),func.count("vartransid").alias('NoOfTrans'),func.countDistinct("varcardno").alias('NoOfCard'),func.countDistinct("varmerchantno").alias("UniqueMerchant")))

from\_pattern = 'yyyy mm'

to\_pattern = 'yyyy-mm'

df\_trans2 = df\_trans2.withColumn('YearMonth1', from\_unixtime(unix\_timestamp(df\_trans2['ym'], from\_pattern), to\_pattern))

month\_list = df\_trans2.select('YearMonth1').distinct().sort('YearMonth1')

min\_month = month\_list[0]

max\_month = month\_list[-1]

month\_list = month\_list.withColumn("temp", lit(1))

df\_carrier = df\_trans2.select('varcarrierid').distinct()

df\_carrier = df\_carrier.withColumn("temp", lit(1))

dx = df\_carrier.crossJoin(month\_list)

from pyspark import SparkContext

import pyspark

from pyspark.sql.functions import broadcast

dx1 = dx.select('varcarrierid', 'YearMonth1').distinct()

dx2 = df\_trans2.select('varcarrierid', 'YearMonth1', 'TotalAmount', 'NoOfTrans', 'NoOfCard', 'UniqueMerchant').distinct()

hive = HiveWarehouseSession.session(spark).build()

hive.setDatabase("sandbox\_dscoe")

df\_part\_2 = hive.executeQuery('SELECT c.varcarrierid, c.yearmonth1, o.totalamount, o.nooftrans, o.noofcard, o.uniquemerchant FROM stagingtable1master c LEFT OUTER JOIN stagingtable2transaction o ON (c.varcarrierid == o.varcarrierid AND c.yearmonth1 == o.yearmonth1)')

from pyspark.sql import functions as F

first\_date = df\_trans.select('ts','varcarrierid').sort('ts')

from\_pattern = 'yyyy-MM-dd'

to\_pattern = 'yyyy-MM'

first\_date = first\_date.withColumn("ts", first\_date["ts"].cast(StringType()))

first\_date = first\_date.withColumn('YearMonth', from\_unixtime(unix\_timestamp(first\_date['ts'], from\_pattern), to\_pattern))

first\_date = first\_date.groupBy(first\_date['varcarrierid']).agg(F.first(first\_date['YearMonth']).alias('Period'))

from datetime import datetime,timedelta

from dateutil.relativedelta import relativedelta

import pyspark.sql.functions as f

#2016-01 to curmonth-5

data = [("2017-01","2017-02")]

date = spark.createDataFrame(data, ["minDate", "maxDate"])

date = date.withColumn("monthsDiff", f.months\_between("maxDate", "minDate")).\

withColumn("repeat", f.expr("split(repeat(',', monthsDiff), ',')")).\

select("\*", f.posexplode("repeat").alias("date", "val")).\

withColumn("date", f.expr("add\_months(minDate, date)")).\

select('date')\

from pyspark.sql.types import \*

field = [StructField("varcarrierid",StringType(), True),StructField("yearmonth1", StringType(), True),StructField("totalamount", IntegerType(), False),

StructField("nooftrans", IntegerType(), True), StructField("noofcard", IntegerType(), True), StructField("uniquemerchant", IntegerType(), False), StructField("Period", TimestampType(), True),StructField("yearmonth", TimestampType(), True)]

schema = StructType(field)

transaction = sqlContext.createDataFrame(sc.emptyRDD(), schema)

from pyspark.sql.functions import asc

from pyspark.sql.window import Window

df\_part\_3 = df\_part\_2.join(first\_date, 'varcarrierid','left')

df\_part\_3 = df\_part\_3.fillna(0)

import pyspark.sql.functions as f

from pyspark.sql import Window

from pyspark.sql.functions import monotonically\_increasing\_id

from pyspark.sql.functions import to\_timestamp, expr

from pyspark.sql.functions import rank

from pyspark.sql.functions import row\_number, lit

df\_part\_3 = df\_part\_3.withColumn('yearmonth', to\_timestamp("yearmonth1", "yyyy-MM"))

df\_part\_3 = df\_part\_3.withColumn('period', to\_timestamp("period", "yyyy-MM"))

date = date.withColumn("month\_4", expr("add\_months(date,4)")).\

withColumn("month\_5", expr("add\_months(date,5)")).\

withColumn("month\_7", expr("add\_months(date,7)"))

date = date.withColumn('month4\_1', to\_timestamp("month\_4", "yyyy-MM"))

date = date.withColumn('month5\_1', to\_timestamp("month\_5", "yyyy-MM"))

date = date.withColumn('month7\_1', to\_timestamp("month\_7", "yyyy-MM"))

for row in date.rdd.collect():

df\_part\_4 = df\_part\_3[df\_part\_3['period'] < row[0]]

df\_part\_4 = df\_part\_4[(df\_part\_4['yearmonth'] >= row[0]) & (df\_part\_4['yearmonth'] <= row[2])]

w = Window.orderBy('yearmonth').partitionBy('varcarrierid')

df\_part\_5 = df\_part\_4.withColumn('sum', f.sum('totalamount').over(w)).where(f.col('sum') != 0.0).drop('sum')

transaction = transaction.union(df\_part\_5)

from pyspark.sql.functions import monotonically\_increasing\_id

from pyspark.sql.functions import row\_number

from pyspark.sql.functions import dense\_rank

transaction\_1 = transaction.withColumn("id", row\_number().over(Window.partitionBy("varcarrierid").orderBy("varcarrierid")))

df2 = transaction\_1[(transaction\_1.id % 5 == 0)]

from pyspark.sql.functions import expr

from pyspark.sql.functions import \*

df2 = df2.withColumn("date1", expr("add\_months(yearmonth1,1)")).\

withColumn("date2", expr("add\_months(yearmonth1,2)")).\

withColumn("date3", expr("add\_months(yearmonth1,3)"))

from\_pattern = 'yyyy-MM-dd'

to\_pattern = 'yyyy-MM'

df2 = df2.withColumn('month1', to\_timestamp("date1", "yyyy-MM")).withColumn('month2', to\_timestamp("date2", "yyyy-MM")).withColumn('month3', to\_timestamp("date3", "yyyy-MM"))

df\_part\_2 = df\_part\_2.withColumn('yearmonth', to\_timestamp("yearmonth1", "yyyy-MM"))

df\_join = df2.join(df\_part\_2, (df2.varcarrierid == df\_part\_2.varcarrierid) & (df2.month1 == df\_part\_2.yearmonth)).select(df2.varcarrierid, df2.yearmonth, df2.totalamount,df\_part\_2.totalamount.alias('A1'), df2.Period, df2.month1, df2.month2, df2.month3)

df\_join\_1 = df\_join.join(df\_part\_2, (df\_join.varcarrierid == df\_part\_2.varcarrierid) & (df\_join.month2 == df\_part\_2.yearmonth)).select(df\_join.varcarrierid,df\_join.yearmonth,df\_join.totalamount,df\_part\_2.totalamount.alias('A2'),df\_join.A1,df\_join.month1, df\_join.month2, df\_join.month3)

df\_join\_2 = df\_join\_1.join(df\_part\_2, (df\_join\_1.varcarrierid == df\_part\_2.varcarrierid) & (df\_join\_1.month3 == df\_part\_2.yearmonth)).select(df\_join\_1.varcarrierid,df\_join.yearmonth,df\_join\_1.totalamount,df\_join\_1.A1,df\_join\_1.A2,df\_part\_2.totalamount.alias('A3'), df\_join\_1.month1, df\_join\_1.month2, df\_join\_1.month3)

df\_join\_2 = df\_join\_2.dropDuplicates()

df\_join\_2 = df\_join\_2.na.fill(0)

df\_join\_2 = df\_join\_2.withColumn('flag', when(df\_join\_2.A1+df\_join\_2.A2+df\_join\_2.A3 == 0, 1).otherwise(0))

df\_join\_3 = df\_join\_2.withColumn("month4", expr("date\_sub(yearmonth,31)")).\

withColumn("month3", expr("date\_sub(yearmonth,2)")).\

withColumn("month2", expr("date\_sub(yearmonth,3)")).\

withColumn("month1", expr("date\_sub(yearmonth,4)"))

from\_pattern = 'yyyy-MM-dd'

to\_pattern = 'yyyy-MM'

df3 = df\_join\_3.withColumn('month1', to\_timestamp("month1", "yyyy-MM")).withColumn('month2', to\_timestamp("month2", "yyyy-MM")).withColumn('month3', to\_timestamp("month3", "yyyy-MM")).withColumn('month4', to\_timestamp("month4", "yyyy-MM"))