10/3/2017:

update: change thruput: if have account in loanmast, but thruput = 0 ( because thruput\_in = thruput\_out, or not arising thruput (currrent )) => thruput = 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Current status** | **Updated status** | **Macro** | **Code** | **Note** |
| 10/3 | Have account, but not have thruput, thruput = NA ( cannot specific with not have account | Have account, but not have thruput, thruput = 0 ( equal with case thruput\_in = thruput\_out | %MACRO THRUPUT  (added code :  proc sql;  create table &out as select  a.cif\_number,  b.sum\_transaction,  b.sum\_transaction\_tru\_ln,  b.sum\_transaction\_tru\_dd,  b.diff  from  ews\_lc.DDMAST a  left join &out b on a.cif\_number = b.cif\_number;  quit;  data &out;  set &out;  if diff =. then diff = 0;  run;  ) to the end of macro | %MACRO THRUPUT(STARTDATE,ENDDATE,masterfile,OUT);  /\*TAO LIBNAME\*/  libname MISDB OLEDB USER=msb\_qlrr PASS=msb@qlrr datasource='10.1.17.222\mssqlserver\_2008'  provider=sqloledb schema=dbo;  /\*lAY DU LIEU DDMAST NHUNG DA LOAI DI CAC SP KY QUY\*/  /\*CIF KHACH HANG NAM TRONG TRONG DANH SACH KHACH HANG EWS (DANH SACH DUOC TAO THEO TUNG YEU CAU)\*/  PROC SQL;  CREATE TABLE ews\_lc.DDMAST AS SELECT DISTINCT  CIF\_NUMBER,  ACCOUNT  FROM MISDB.DDMAST WHERE IMPORT\_DATE >=&startdate and IMPORT\_DATE <=&enddate AND ACCOUNT\_TYPE NOT IN (  'CA11B',  'CA11FB',  'CA11I',  'CA11ODPFI',  'CA11SD',  'CA11STAFF',  'CA11TCTD',  'CA12CP',  'CA12GT',  'CA12LC',  'CA12LS',  'CA12NV',  'CA12OP',  'CA12OPI',  'CA131',  'CA22ODP',  'CAFCB',  'CAGOPVON',  'CAM1',  'CAM1FC',  'CAMBASIC',  'CAMBASICFC',  'CAMCOM',  'CAMFAMILY',  'CA-RM',  'F-CA12GT',  'F-CA12LC',  'F-CA12NV',  'F-CA12OP',  'KYQUYT',  'L-CA12GT',  'L-CA12LC',  'L-CA12OP',  'MC-ADVANCE',  'MC-FREE100',  'MCPLUS',  'M-INVEST01',  'R-CA11IUL',  'R-CA12OP',  'S-CA11B',  'SLC01',  'SV11',  'SVVIP'  )  AND CIF\_NUMBER IN (select cif\_number from &masterfile);  QUIT;  /\*LAY DU LIEU BANG TY GIA\*/  PROC SQL;  CREATE TABLE ews\_lc.SSFXRT AS SELECT DISTINCT  MID\_RATE,  INPUT(import\_date, yymmdd10.) AS import\_date FORMAT DATE9.,  CURRENCY\_CODE  FROM MISDB.SSFXRT  WHERE IMPORT\_DATE >= &STARTDATE AND IMPORT\_DATE <= &ENDDATE;  QUIT;  /\*LAY DU LIEU DDHIST\*/  /\*LAY SO DU CUA KHACH HANG DUOC GHI CO (CREDIT)HAY C TREN DDHIST\*/  /\*KHI TAI KHAON CUA KHACH HANG DUOC GHI C CO = DONG TIEN VAO\*/  PROC SQL;  CREATE TABLE ews\_lc.DDHIST AS SELECT  ACCOUNT,  TRANSACTION\_AMOUNT,  CURRENCY\_TYPE,  DEBIT\_CREDIT\_CODE,  INPUT(POSTED\_DATE, yymmdd10.) AS POSTED\_DATE FORMAT DATE9.,  USER\_ID,  SEQUENCE\_NUMBER,  AUX\_TRANSACTION\_CODE,  AFFECTS\_CODE  FROM MISDB.DDHIST  WHERE POSTED\_DATE >= &STARTDATE AND POSTED\_DATE <=&ENDDATE AND DEBIT\_CREDIT\_CODE ="C"  AND ACCOUNT IN (SELECT ACCOUNT FROM ews\_lc.DDMAST) ;  QUIT;  /\*QUY DOI TRANSACTION\_AMOUNT TRONG DDHIST SAU KHI MAPPING LOAI DONG TIEN\*/  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_1 AS SELECT  A.ACCOUNT,  A.TRANSACTION\_AMOUNT,  A.TRANSACTION\_AMOUNT\*C.MID\_RATE AS TRANSACTION\_AMOUNT\_QD,  A.CURRENCY\_TYPE,  A.DEBIT\_CREDIT\_CODE ,  A.POSTED\_DATE,  A.USER\_ID,  A.SEQUENCE\_NUMBER,  A.AFFECTS\_CODE,  A.AUX\_TRANSACTION\_CODE,  B.CIF\_NUMBER,  C.MID\_RATE  FROM ews\_lc.DDHIST A  LEFT JOIN ews\_lc.DDMAST B  ON A.ACCOUNT=B.ACCOUNT  LEFT JOIN ews\_lc.SSFXRT C ON A.CURRENCY\_TYPE = C.CURRENCY\_CODE AND A.POSTED\_DATE =C.IMPORT\_DATE  ORDER BY B.CIF\_NUMBER, A.ACCOUNT, A.POSTED\_DATE;QUIT;  /\*TINH TONG SO TIEN TRANSACTION\_AMOUNT SAU KHI QUY DOI\*/  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_SUM AS SELECT DISTINCT  CIF\_NUMBER,  SUM(TRANSACTION\_AMOUNT\_QD) AS SUM\_TRANSACTION  FROM ews\_lc.DDHIST\_1  GROUP BY CIF\_NUMBER;  QUIT;  /\*LAY DU LIEU CAC KHOAN GIAI NGAN LOAN TREN DD\*/  /\*TRUOC TIEN LAY CAC KHOAN DA GIAI NGAN TREN LNMAST\*/  /\* CHI LAY CIF\_NUMBER VA ACCOUNT\*/  PROC SQL;  CREATE TABLE ews\_lc.LNMAST AS SELECT DISTINCT  CIF\_NUMBER,  ACCOUNT  FROM MISDB.LNMAST WHERE IMPORT\_DATE >=&STARTDATE and import\_date <=&enddate AND CIF\_NUMBER IN (select CIF\_NUMBER from &masterfile);  QUIT;  /\*TAI DU LIEU LNHIST  CHI LAY NHUNG KHOAN GIAI NGAN BANG TIEN MAT, CHUYEN KHOAN TRONG CHI NHANH/TRONG HE THONG/NGOAI HE THONG (4320,4360)  NGOAI RA DAY LA NHUNG GIAO DICH ANH HUONG DEN GOC(P)\*/  PROC SQL;  CREATE TABLE ews\_lc.LNHIST AS SELECT  ACCOUNT,  TRANSACTION\_AMOUNT,  CURRENCY\_TYPE,  DEBIT\_CREDIT\_CODE,  INPUT(POSTING\_DATE, yymmdd10.) AS POSTING\_DATE FORMAT DATE9.,  USER\_ID,  SEQUENCE\_NUMBER,  LHAXTC,  TRANSACTION\_CODE,  AFFECTS\_CODE  from misdb.lnhist  where posting\_date >= &startdate and posting\_date <= &enddate AND LHAXTC IN ( '4320', '4360', '4364') AND AFFECTS\_CODE = 'P'  AND ACCOUNT IN (SELECT ACCOUNT FROM ews\_lc.LNMAST) ;  QUIT;  /\*JOIN LNHIST VOI LNAMST DE LAY CIF\_NUMBER CHO NHUNG KHOAN GIAI NGAN\*/  proc sql;  create table ews\_lc.LNHIST\_1 as select  a.cif\_number,  b.\*  from ews\_lc.LNMAST a right join ews\_lc.LNHIST b  on a.account = b.account;  quit;  /\*JOIN DDHIT VOI LNHIST DE TIM RA CAC KHOAN GIAI NGAN\*/  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_TRU\_QD\_LN AS SELECT distinct  a.\*,  b.account as account\_ln,  b.posting\_date as posting\_date\_ln,  b.TRANSACTION\_AMOUNT as TRANSACTION\_AMOUNT\_ln,  b.cif\_number as cifno  FROM ews\_lc.DDHIST\_1 A  left JOIN ews\_lc.LNHIST\_1 b  ON b.POSTING\_DATE =a.POSTED\_DATE  and b.user\_id = a.user\_id  and b.SEQUENCE\_NUMBER = a.SEQUENCE\_NUMBER  and b.cif\_number = a.cif\_number;  QUIT;  /\*TNH TONG CAC KHOAN DA GIAI NGAN DA CHUYEN VAO DD THEO CIF\*/  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_TRU\_LN AS SELECT DISTINCT  CIFNO,  SUM(TRANSACTION\_AMOUNT\_QD) AS SUM\_TRANSACTION  FROM ews\_lc.DDHIST\_TRU\_QD\_LN  GROUP BY CIFno;  QUIT;  /\*LOAI CAC GIAO DICH CUA CUNG KHACH HANG CHUYEN VAO DD TRONG NOI BO MSB\*/  /\*LAY CAC TRANSACTION NOI BO CUA CIF HOAC THEO DANH SACH KHACH HANG\*/  data ews\_lc.ddhist\_1\_c;  set ews\_lc.DDHIST\_1;  if AUX\_TRANSACTION\_CODE in (  'BP1321',  'EB1321',  'EB2320',  'EB2321',  'IB1321',  'IB2321',  'SMS1321',  'V1321',  '1320',  '1321',  '2320',  '2321',  '9614',  '9615'  );  if AFFECTS\_CODE = "B";  if DEBIT\_CREDIT\_CODE ="C";  run;  PROC SQL;  CREATE TABLE ews\_lc.DDMAST\_D AS SELECT DISTINCT  CIF\_NUMBER,  ACCOUNT  FROM MISDB.DDMAST WHERE IMPORT\_DATE >=&startdate and IMPORT\_DATE <=&enddate  AND CIF\_NUMBER IN (select cif\_number from &masterfile);  QUIT;  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_D AS SELECT  ACCOUNT,  TRANSACTION\_AMOUNT,  CURRENCY\_TYPE,  DEBIT\_CREDIT\_CODE,  INPUT(POSTED\_DATE, yymmdd10.) AS POSTED\_DATE FORMAT DATE9.,  USER\_ID,  SEQUENCE\_NUMBER,  AUX\_TRANSACTION\_CODE,  AFFECTS\_CODE  FROM MISDB.DDHIST  WHERE POSTED\_DATE >= &STARTDATE AND POSTED\_DATE <=&ENDDATE AND DEBIT\_CREDIT\_CODE ="D"  AND ACCOUNT IN (SELECT ACCOUNT FROM ews\_lc.DDMAST\_D) and AFFECTS\_CODE = "B"  and AUX\_TRANSACTION\_CODE in ('BP1321',  'EB1321',  'EB2320',  'EB2321',  'IB1321',  'IB2321',  'SMS1321',  'V1321',  '1320',  '1321',  '2320',  '2321',  '9614',  '9615');  QUIT;  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_1\_D AS SELECT distinct  A.ACCOUNT,  A.TRANSACTION\_AMOUNT,  A.TRANSACTION\_AMOUNT\*C.MID\_RATE AS TRANSACTION\_AMOUNT\_QD,  A.CURRENCY\_TYPE,  A.DEBIT\_CREDIT\_CODE ,  A.POSTED\_DATE,  A.USER\_ID,  A.SEQUENCE\_NUMBER,  A.AFFECTS\_CODE,  B.CIF\_NUMBER,  C.MID\_RATE  FROM ews\_lc.DDHIST\_D A LEFT JOIN ews\_lc.DDMAST\_D B  ON A.ACCOUNT = B.ACCOUNT  LEFT JOIN ews\_lc.SSFXRT C ON A.CURRENCY\_TYPE = C.CURRENCY\_CODE AND A.POSTED\_DATE =C.IMPORT\_DATE  ORDER BY B.CIF\_NUMBER, A.ACCOUNT, A.POSTED\_DATE;  QUIT;  proc sql;  create table ews\_lc.DD\_DD\_TRU as select distinct  a1.\*,  b1.account as account\_dd,  b1.posted\_date as posted\_date\_tkgui,  b1.cif\_number as cifno  from ews\_lc.ddhist\_1\_c a1 left join ews\_lc.ddhist\_1\_d b1  on a1.USER\_ID = b1.user\_id  and a1.SEQUENCE\_NUMBER = b1.Sequence\_Number  and a1.posted\_date = b1.posted\_date  and a1.cif\_number = b1.CIF\_NUMBER;  quit;  PROC SQL;  CREATE TABLE ews\_lc.DDHIST\_TRU\_DD AS SELECT DISTINCT  CIFno,  SUM(TRANSACTION\_AMOUNT\_QD) AS SUM\_TRANSACTION  FROM ews\_lc.DD\_DD\_TRU  GROUP BY Cifno;  QUIT;  proc sql;  create table &out as select distinct  a.cif\_number,  a.SUM\_TRANSACTION,  b.SUM\_TRANSACTION as SUM\_TRANSACTION\_tru\_ln,  c.SUM\_TRANSACTION as SUM\_TRANSACTION\_tru\_dd  from ews\_lc.DDHIST\_SUM a left join ews\_lc.DDHIST\_TRU\_ln b  on a.cif\_number = b.cifno  left join ews\_lc.DDHIST\_TRU\_DD c  on a.cif\_number = c.cifno;  quit;  data &out;  set &out;  if SUM\_TRANSACTION\_tru\_ln = '.' then SUM\_TRANSACTION\_tru\_ln = 0;  if SUM\_TRANSACTION\_tru\_dd = '.' then SUM\_TRANSACTION\_tru\_dd = 0;  diff = SUM\_TRANSACTION - SUM\_TRANSACTION\_tru\_ln - SUM\_TRANSACTION\_tru\_dd;  run;  proc sql;  create table &out as select  a.cif\_number,  b.sum\_transaction,  b.sum\_transaction\_tru\_ln,  b.sum\_transaction\_tru\_dd,  b.diff  from  ews\_lc.DDMAST a  left join &out b on a.cif\_number = b.cif\_number;  quit;  data &out;  set &out;  if diff =. then diff = 0;  run;  /\*dm 'output;clear;log;clear;'\*/  %MEND; | Note: kiểm tra lại code này, do 19/03 chạy bị lỗi, nhưng không hiểu vì sao |
| 11/03 | Have 2 specified file code, and copy code from file excel must be done in many time, because macro , and sql precedure, command run code in mix to each other | Combine two file code to a consitency one. Add all marco to one area, And add to command run code (sql procedure) in file excel |  |  |  |
| 19/3/  2017 | Code to calculate Facility is not updated to facility tree. |  | + Macro avgbalance add more facility code: 211,180,181,….  + Macro HM: | /\*============================================================\*/  /\*============================ macro NAY DUNG TRONG MACRO TINH HM================================\*/  /\*============================ macro add colum level to lnappf table================================\*/  %macro recusive\_add\_level\_HM(IN\_lnappf,OUT\_lnappf\_with\_level);  %local level nlevel;  %let level = 1;  proc sql;  /\* Get first level as table level\_1. \*/  CREATE table Level\_1 as  SELECT  cpno\_parent,  cpno,  1 as level,  Facilityname,  facilityLimit,  Cifno,  FacilityExpiryDate,  CurrencyType,  DateApproved,  Date  FROM &IN\_lnappf  WHERE &IN\_lnappf..cpno\_parent in ('.','');  /\* Recursively get successive levels. \*/  %do %while(&SQLOBS > 0); /\* biáº¿n SQLOBS nÃ y lÃ  biáº¿n tá»± sinh sau má»—i cÃ¢u lá»‡nh query trong sql = number of columns in query results\*/    %let nlevel = %eval(&level + 1);  CREATE table Level\_&nlevel as  SELECT  &IN\_lnappf..cpno\_parent,  &IN\_lnappf..cpno,  &nlevel as level,  &IN\_lnappf..Facilityname,  &IN\_lnappf..facilityLimit,  &IN\_lnappf..Cifno,  &IN\_lnappf..FacilityExpiryDate,  &IN\_lnappf..CurrencyType,  &IN\_lnappf..DateApproved,  &IN\_lnappf..Date /\* have colum similar with level 1\*/  FROM &IN\_lnappf as child INNER JOIN Level\_&level as parent  /\* trong bang tong lnappf, lay ra nhung ban ghi co cpno\_parent == cpno trong bang level 1 thi nhung ban ghi do la level 2 \*/  ON child.cpno\_parent=parent.cpno;  %let level = &nlevel;  %end;  /\* Concatenate all levels together \*/  data &OUT\_lnappf\_with\_level;  set %do ii = 1 %to %eval(&level);  Level\_&ii  %end;  ;  run;  %mend recusive\_add\_level\_HM;  /\*============================================================\*/  /\*============================ macro NAY DUNG TRONG MACRO TINH HM================================\*/  /\*============================ macro update han muc theo min (sum\_han\_muc\_con, han\_muc\_cha) ================================\*/  %macro recusive\_sum\_HM(IN\_lnappf\_leave, IN\_lnappf\_with\_level,OUT\_lnappf\_level1\_final);  %local level low\_level ;  /\* create 5 table with level respectively \*/  %let i=1;  %do i=1 %to 5;  data lnappf\_leave\_&i;  set &IN\_lnappf\_leave;  if level = &i;  run;  %end;  %do i=1 %to 5;  %let level = %eval(6 -&i ); /\* coi nhu vong lap giam dan tu 5 -> 0 \*/  %let low\_level = %eval(&level - 1 );  proc sql;  create table lnappf\_leave\_sum\_&low\_level as select  parent.CPNo\_parent,  parent.CPNO,parent.level,  parent.Facilityname,  parent.facilityLimit ,  parent.FacilityExpiryDate,  parent.cifno,  parent.CurrencyType ,  parent.DateApproved,  parent.Date,  sum(child.facilityLimit) as sum\_limit\_child  from &IN\_lnappf\_with\_level as parent join lnappf\_leave\_&level as child  on parent.CPNo = child.CPNo\_parent  group by parent.CPNo\_parent,  parent.CPNO,  parent.level,  parent.Facilityname,  parent.facilityLimit ,  parent.FacilityExpiryDate,  parent.cifno,  parent.CurrencyType,  parent.DateApproved,  parent.Date  ;  quit;  /\* l?y min (sum (h?n m?c con), h?n m?c cha)\*/  data lnappf\_leave\_sum\_&low\_level;  set lnappf\_leave\_sum\_&low\_level;  facilityLimit = min (sum\_limit\_child,facilityLimit);  drop sum\_limit\_child;  run;  /\* t?ng h?p 2 h?n m?c c?p th?p hon\*/  data lnappf\_leave\_&low\_level;  set lnappf\_leave\_&low\_level lnappf\_leave\_sum\_&low\_level;  run;    %end;  data &OUT\_lnappf\_level1\_final ;  set lnappf\_leave\_1 ;  run;  %mend recusive\_sum\_HM;  /\*============================================================\*/  /\*============================ TINH HM================================\*/  /\*TINH HAN MUC CHO VAY DC CAP\*/  /\*LAY HAM MUC MOI CON HIEU LUC\*/  %macro HM(IN\_lnappf,IN\_lnmast\_lnappf,IN\_day1,IN\_name\_of\_Tong\_HM\_quydoi,Out\_HMSD\_01);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INPUT:input1: lnappf\_01: lnappf of month report \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INPUT:input2: lnmast\_lnappf\_01: lnmast gÃ¡n vá»›i lnappf of month report vá»›i nhá»¯ng háº¡n má»©c náº±m trong cÃ¢y HM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INPUT: day1: date end of month report \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INPUT: &IN\_Name\_of\_Tong\_HM\_quydoi: name of column Tong\_HM\_qui\_doi \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\* filter table Input acorrding to facility\_code, and revol = R (ews\_lc.lnappf\_01,ews\_lc.LNMAST\_LNAPPF\_q\_211R\_01,MDY(1,31,2017)\*/  data LNMAST\_LNAPPF\_q\_211R\_01;  set &IN\_lnmast\_lnappf ;  if FACILITY\_CODE in (001,004,005,006,007,211,180,181,902,828,900,182) and AFREVL = 'R';  run;  data lnappf\_01;  set &IN\_lnappf;  if FACILITYNAME in (001,004,005,006,007,211,180,181,902,828,900,182) and AFREVL = 'R';  run;  /\* create column cpno\_parent in table Lnappf \*/  proc sql;  create table lnappf\_with\_cpno\_parent as SELECT  in1.\*,  in2.cpno as cpno\_parent  FROM lnappf\_01 in1 /\* as child table\*/  LEFT JOIN ews\_lc.lnappf\_01 in2 /\* as parent table\*/  ON in1.FacilityAAno = in2.ApplicationNumber  AND in1.FacilityCode = in2.FacilityName  AND in1.FacilitySeq = in2.SequenceNumber;  quit;  /\* create column level of facility by recusive query \*/  %recusive\_add\_level\_HM(lnappf\_with\_cpno\_parent,lnappf\_with\_level);  /\* lay ra nhung han muc cuoi cung, gan lien voi khoan vay, va con hieu luc \*/  proc sql;  create table lnappf\_leave\_efective as SELECT \*  FROM lnappf\_with\_level  WHERE cpno not in (select cpno\_parent from lnappf\_with\_level) and FacilityExpiryDate >= &IN\_day1 ;  quit;  /\* lay ra nhung han muc gan voi khoan vay\*/  proc sql;  create table lnappf\_leave\_with\_lnmast as select \*  from lnappf\_with\_level where cpno in (select distinct cpno from LNMAST\_LNAPPF\_q\_211R\_01) ;  quit;  /\*consolidate two table lnapp\_leave( it will remove duplicate) \*/  data lnappf\_leave;  set lnappf\_leave\_efective lnappf\_leave\_with\_lnmast;  run;  /\*remove duplicate row\*/  proc sort data=lnappf\_leave noduprecs;  by \_all\_ ;  Run;  /\* create table lnappf\_level1\_final, with facilitylimit is updated = min(....)\*/  %recusive\_sum\_HM(lnappf\_leave, lnappf\_with\_level,lnappf\_level1\_final);  /\* map exchange\*/  PROC SQL;  CREATE TABLE lnappf\_level1\_final\_map\_exchange  AS SELECT  IN1.\*,  IN2.MID\_RATE as mid\_rate\_appf  FROM lnappf\_level1\_final AS IN1 LEFT JOIN ews\_lc.ssfxrt AS IN2  ON IN1.CurrencyType = IN2.CURRENCY\_CODE  where in2.import\_date = in1.Date  ;  QUIT;  data lnappf\_level1\_final\_map\_exchange;  set lnappf\_level1\_final\_map\_exchange;  HM\_quydoi = mid\_rate\_appf\*FacilityLimit;  if DateApproved = "." then delete;  run;  proc sql;  create table &Out\_HMSD\_01 as select  cifno,  sum(HM\_quydoi) as &IN\_name\_of\_Tong\_HM\_quydoi  from lnappf\_level1\_final\_map\_exchange  group by cifno;  quit;  data &Out\_HMSD\_01;  set &Out\_HMSD\_01;  if cifno = '.' then delete; run;  proc sort data= &Out\_HMSD\_01;  By Cifno;run;  %mend HM; | Kết quả chạy ra hạn mức không thay đổi, nhưng avg balance thay đổi nên là utilization thay đổi.  Ý tưởng code hạn mức: ---------------============================================  --============= flow để tính hạn mức cho khách hàng====---  -- đầu tiên, bảng lnappf chỉ lấy những hạn mức có facility name nằm trong phần quan tâm,  -- bảng lnmast được ghép với lnppf để lấy cpno  --tại bảng lnappf tìm cpno parent của nó, levlel trong cây hạn mức  -- lọc ra bảng lnappf-leave, với hạn mức được giải ngân( gắn liền với lnmast) hoặc những hạn mức có level cao nhất( điều kiện lấy là: có thể lấy hạn mức mà không là cha của 1 hạn mức nào cả và còn hiệu lực )  --tạo các bảng dữ liệu hạn mức dữ liệu theo các level từ bảng lnppf leave . ( macro recusive\_add\_level\_HM trong code)  -- [bắt đầu 1 vòng lặp] bắt đấu với bange lnappf leave với level cao nhất, VD: 5, map lên hạn cha của nó, lấy min(sum(hạn mức level 5 cùng cha), hạn mức cha) để được hạn mức level 4 ( map qua cpno parent)(1)  -- tiếp tục union hạn mức level 4 mới được tạo ra (1) với hạn mức lnappf leave level 4. [kết thúc vòng lặp]  -- tiếp tục vòng lặp. đến khi tìm được toàn bộ hạn mức level 1 (macro recusive\_sum\_HM)  sau đó dùng sum và group by thep cifno.  **Chú ý trong đoạn code.**  Cú pháp, |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |